JOURNAL OF THEORETICAL
AND APPLIED MECHANICS
58, 1, pp. 127-142, Warsaw 2020
DOI: 10.15632/jtam-pl/115332

MECHATRONIC DESIGNING AND PROTOTYPING OF A MOBILE WHEELED ROBOT DRIVEN BY A MICROCONTROLLER

Andrzej Burghardt, Dariusz Szybicki, Krzysztof Kurc, Magdalena Muszyńska
Rzeszow University of Technology, Faculty of Mechanical Engineering and Aeronautics, Rzeszow, Poland
e-mail: andrzejb@prz.edu.pl, dszybicki@prz.edu.pl, kkurc@prz.edu.pl, magdaw@prz.edu.pl

The paper presents an existing example of the process of mechatronic designing and prototyping of a mobile wheeled robot. Selection and specification of the mathematical model from a suitable class precedes the presentation of kinematic and dynamic equations describing the dynamic object of the analysis. Further sections contain the methods of designing, the CAD model of the device, software and electronics design. The methodology of designing and prototyping of the mobile wheeled robot is identified and presented in this paper as a solution that may be used for the purpose of designing mechatronic hardware of this kind.

Keywords: mechatronics, wheeled robots, prototyping, wheeled robot control

1. Introduction

The mechatronic design of robots has been treated in many papers in recent times. Studies (Bishop, 2006; Gawrysiak, 1997) deal with components of mechatronic systems, such as actors, sensors, signals and their processing, methods of data processing, issues connected with multiple-body system modelling, path planning and controlling. Bishop (2006) presents a comprehensive mathematical description of mechatronic systems. Turowski (2008) calls attention to the systems approach in the designing of mechatronic equipment. Contrary to Bishop (2006), the aforementioned paper highlights the role of quick prototyping and analyses the economic aspect of mechatronic designing in detail. It lists mechatronic components and brings to light the importance of designing smart machines and indispensable IT tools. It presents the method of physical field quick modelling, system motion quick designing and, additionally, matters connected with mechatronic element reliability and quality management. Other studies (Acar *et al.*, 1994; Amerongen *et al.*, 2000) explain fundamental mechatronic notions and systems expression as well as indicate the role of models and structures thereof.

Further Sections discuss the problems of path planning, languages and systems for robot programming, and give detailed descriptions of actors and sensors. The paper gives examples of mechatronic equipment and an example of a robot as a mechatronic device. A very interesting case of mechatronic designing can be found in studies by Duda *et al.* (2016) and Mężyk (2010). The said papers focus on a device for support of re-learning of walking as well as on special purpose track-laying vehicles. An example of mechatronic designing from the idea up to the prototype is given by Giergiel *et al.* (2014). In that case, however, a track-laying robot is the object of designing and prototyping whereas the kinetics and the control system employed is considerably different from what is discussed in the present paper.

The review of bibliographic references allows the conclusion that the list of available publications may be extended by a paper that will illustrate the process of mechatronic design of a wheeled robot in a comprehensive way, including the construction of prototype and software implementation. A discussion of successive design phases will come with the presentation of necessary mechatronic designing tools. Such tools include software and equipment applied for the purpose of prototype construction and testing. Advanced IT tools which allow the designing of structural elements and adapting necessary electronic elements (Giergiel *et al.*, 2014) may be included in the said mechatronic design instrumentation. Simulations which are necessary phases in describing the kinetics of the future device will become possible thanks to the resulting virtual models. Simulation software constitutes a separate group of tools. Such applications permit e.g. modelling of the phenomena of motion. They allow validation of the equations that describe the kinematics and dynamics of the object of analysis. Moreover, they ensure the selection and simulation of the operations of the control system.

2. Kinetics of a mobile wheeled robot

A physical model and description of kinetics of an object are key elements of the mechatronic designing process (Buratowski *et al.*, 2012; Giergiel *et al.*, 2000; Kurc *et al.*, 2011; Kurc *et al.*, 2016; Mężyk *et al.*, 2011). They enable simulations and are required to design control systems. Kinematics and dynamics of mobile wheeled robots have been presented in many research works (Giergiel *et al.*, 2002; Giergiel and Małka, 2006; Hendzel *et al.*, 2008) consequently, such publications will be given moderate attention in the present paper.

2.1. Kinematics

Mobile robot kinematical problems allow description of the parameters of motion. Hence, kinematical inverse problems should be determined for a future robot, by making assumptions of motion of a defined point and by determining kinematical parameters of the driving wheel. Figure 1 shows the model adopted as a robot representation (Giergiel *et al.*, 2002; Giergiel and Małka, 2006; Hendzel *et al.*, 2008).

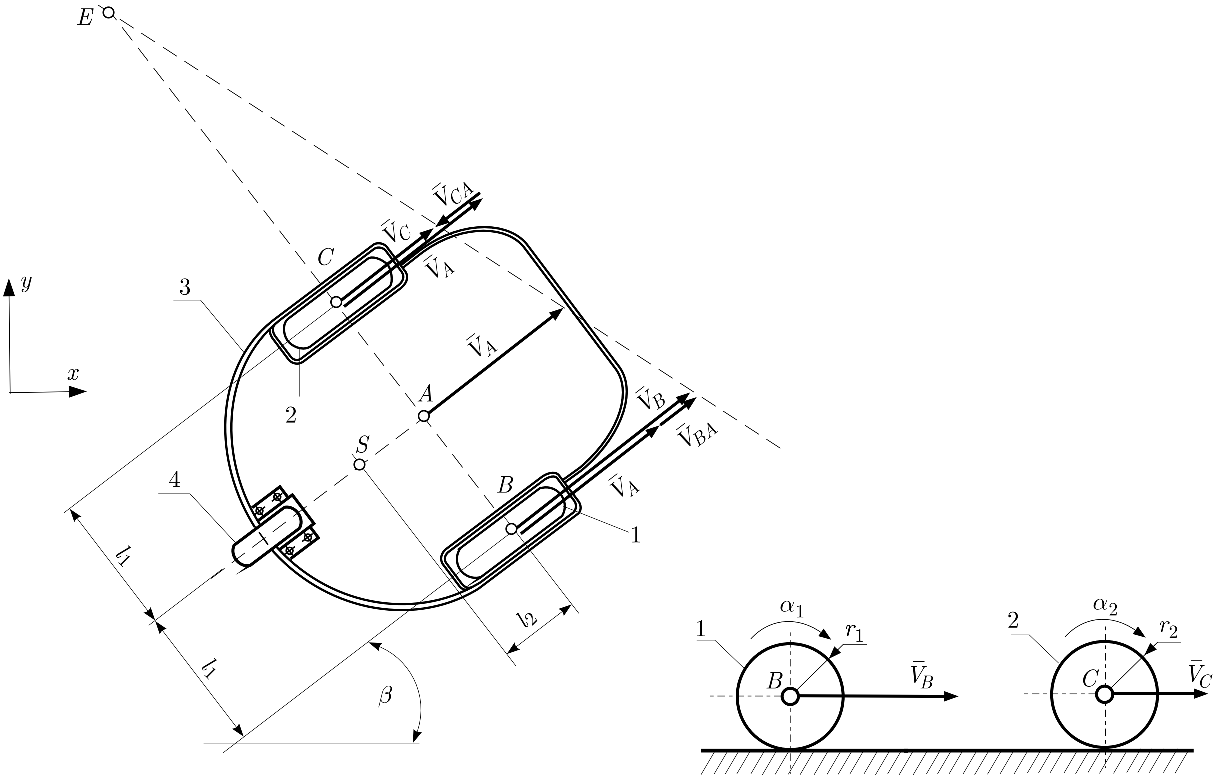

Fig. 1. Model robot kinematics

Fundamental elements to be distinguished in the model are: robot frame 3; self-adjusting supporting wheel 4, as well as driving wheels 1 and 2. The said wheels rotate around their own axes which are stationary relative to the frame. The angles of rotation of the wheels are identified as α_1 and α_2, whereas the radiuses are marked $r_1 = r_2 = r$, respectively. Points B and C are the geometrical centres of wheels 1 and 2. For example, if $V_B > V_C$, point E will become an instantaneous centre of rotation of the robot frame and β will mean an instantaneous angle of frame rotation. Point S represents the centre of robot mass and point A is a point of symmetry of segment BC. The point A velocity vector is set in a direction perpendicular to BC and its projections on the plane XY satisfy the following equation

$$\dot{y}_A = \dot{x}_A \tan\beta \tag{2.1}$$

The above relationship means that the constraints imposed on point A are non-holonomic, or that the system is non-holonomic. The system of equations produced by means of the distribution of velocities of points A, B, C allows for the determination of all motion parameters. Further analysis leads to the determination of wheel angular velocities as (Giergiel *et al.*, 2002; Hendzel *et al.*, 2008)

$$\dot{\alpha}_1 = \frac{v_A}{r_1} + \dot{\beta}\frac{l_1}{r_1} \qquad \dot{\alpha}_2 = \frac{v_A}{r_2} - \dot{\beta}\frac{l_1}{r_2} \tag{2.2}$$

The above-mentioned kinematical description which creates the relationship between the preset velocity of point A and the frame rotation angle through equations (2.2) and (2.3) will be used in further Sections of the present paper.

2.2. Dynamics

Advanced algorithms of robot controlling are built with the assistance of object dynamic models. Modelling is always a compromise between the complexity and the potential of implementation. Lagrange's equations of the second kind (Giergiel *et al.*, 2002; Hendzel *et al.*, 2008) are formalisms to be often used in the process of dynamic system modelling. Given that our dynamic system is a non-holonomic system, the constraints imposed on velocities (Giergiel *et al.*, 2002) will be expressed as

$$\dot{x}_A - (r\dot{\alpha}_1 - l_1\dot{\beta})\cos\beta = 0 \qquad \dot{y}_A - (r\dot{\alpha}_2 - l_1\dot{\beta})\sin\beta = 0 \tag{2.3}$$

The matrix form of equations (2.4), (2.5) will be the following

$$\mathbf{J}(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0} \tag{2.4}$$

and, in this case, the Jacobian will be defined as

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} 1 & 0 & l_1\cos\beta & -r\cos\beta & 0 \\ 0 & 1 & -l_1\sin\beta & 0 & -\sin\beta \\ 0 & 0 & 1 - \dfrac{r}{2l_1} & -\dfrac{r}{2l_1} & \end{bmatrix} \tag{2.5}$$

whereas the generalised coordinate vector will be defined as

$$\dot{\mathbf{q}} = [\dot{x}_A, \dot{y}_A, \dot{\beta}, \dot{\alpha}_1, \dot{\alpha}_2]^{\mathrm{T}} \tag{2.6}$$

Lagrange's equations for a non-holonomic system are expressed as

$$\frac{d}{dt}\Big(\frac{\partial \mathbf{E}}{\partial \dot{\mathbf{q}}}\Big)^{\mathrm{T}} - \Big(\frac{\partial \mathbf{E}}{\partial \dot{\mathbf{q}}}\Big)^{\mathrm{T}} = \mathbf{Q} + \mathbf{J}(\mathbf{q})\boldsymbol{\lambda} \tag{2.7}$$

where: $\mathbf{E} = \mathbf{E}(\mathbf{q}, \dot{\mathbf{q}})$ – kinetic energy of the system, $\mathbf{Q}$ – generalised force vector, $\boldsymbol{\lambda}$ – vector of Lagrange's multipliers. In this case, the multipliers are the projections of dry friction forces on the axes of the adopted reference system.

The dynamic equations obtained from relationships (2.7) cannot be implemented in control algorithms since they contain unknown Lagrange's multipliers (Hendzel *et al.*, 2008). Maggie's equations (Giergiel and Małka, 2006; Hendzel *et al.*, 2008) whose advantage consists in the elimination of Lagrange's multipliers by means of their removal, are alternative mathematical formalisms to be applied for the purpose of description of mobile wheeled robot dynamics. Maggie's equations describe motion of the system in generalised coordinates; they are expressed as

$$\sum_{j=1}^{n} C_{ij}\Big[\frac{d}{dt}\Big(\frac{\partial E}{\partial \dot{q}_j}\Big) - \frac{\partial E}{\partial q_j}\Big] = \theta_i \qquad i = 1,\dots,s \tag{2.8}$$

where: s – number of independent parameters of the system in generalised coordinates q_j $(j = 1,\dots,n)$, equal to the number of degrees of freedom of the system. Consequently, all the generalised velocities will be expressed as

$$\dot{q}_j = \sum_{i=1}^{s} C_{ij}\dot{e}_i + G_j \tag{2.9}$$

The quantities $\dot{e}_i$ are called characteristics or kinetic parameters of the system in generalised coordinates. On the other hand, the right-side members of system (2.8) are coefficients at variations δe_i in the expression of the prepared work of external forces of the system. The said coefficients are defined by means of relationship (2.9)

$$\sum_{i=1}^{s} \theta_i \delta e_i = \sum_{i=1}^{s} \delta e_i \sum_{i=1}^{n} C_{ij} Q_j \tag{2.10}$$

Maggie's equations employed to describe motion of a two-wheel mobile robot may be expressed in the form of (Giergiel and Małka, 2006; Hendzel *et al.*, 2008)

$$\begin{aligned}
&(2m_1 + m_4)\Big(\frac{r}{2}\Big)^2(\ddot{\alpha}_1 + \ddot{\alpha}_2) + 2m_4\Big(\frac{r}{2l_1}\Big)^2 rl_2(\dot{\alpha}_2 - \dot{\alpha}_1)\dot{\alpha}_2 + I_{z1}\dot{\alpha}_1\\
&\qquad + (2m_1l_1^2 + m_4l_1^2 + 2I_{x1} + I_{z4})\frac{r}{2l_1}(\ddot{\alpha}_1 - \ddot{\alpha}_2) = M_1 - N_1f_1\\
&(2m_1 + m_4)\Big(\frac{r}{2}\Big)^2(\ddot{\alpha}_1 + \ddot{\alpha}_2) + 2m_4\Big(\frac{r}{2l_1}\Big)^2 rl_2(\dot{\alpha}_1 - \dot{\alpha}_2)\dot{\alpha}_2 + I_{z2}\dot{\alpha}_2\\
&\qquad + (2m_1l_1^2 + m_4l_2^2 + 2I_{x1} + I_{z4})\frac{r}{2l_1}(\ddot{\alpha}_1 - \ddot{\alpha}_2) = M_2 - N_2f_2
\end{aligned} \tag{2.11}$$

Having selected the vector of independent generalised coordinates as $\mathbf{q}_d = [\alpha_1, \alpha_2]^{\mathrm{T}}$, we may express equations (2.11) as (Giergiel and Małka, 2006; Hendzel *et al.*, 2008)

$$\mathbf{M}\ddot{\mathbf{q}}_d + \mathbf{C}(\dot{\mathbf{q}}_d)\dot{\mathbf{q}}_d + \mathbf{F}(\dot{\mathbf{q}}_d) = \mathbf{U} \tag{2.12}$$

where the matrices and vectors are

$$\begin{aligned}
\mathbf{M} &= \begin{bmatrix} a_1 + a_2 + a_3 & a_1 - a_2 \\ a_1 - a_2 & a_1 + a_2 + a_3 \end{bmatrix} & \qquad \mathbf{F}(\dot{\mathbf{q}}_d) &= \begin{bmatrix} a_5 \operatorname{sgn}\dot{\alpha}_1 \\ a_6 \operatorname{sgn}\dot{\alpha}_2 \end{bmatrix}\\
\mathbf{C}(\dot{\mathbf{q}}_d) &= \begin{bmatrix} 0 & 2a_4(\dot{\alpha}_2 - \dot{\alpha}_1) \\ -2a_4(\dot{\alpha}_2 - \dot{\alpha}_1) & 0 \end{bmatrix} & \qquad \mathbf{U} &= \begin{bmatrix} M_1 \\ M_2 \end{bmatrix}
\end{aligned} \tag{2.13}$$

The elements of vector $\mathbf{a} = [a_1, a_2, a_3, a_4, a_5, a_6]^{\mathrm{T}}$ are described by relationships which result from (2.11)

$$\begin{array}{ll} a_1 = (2m_1 + m_4)\Big(\dfrac{r}{2}\Big)^2 & a_2 = (2m_1 l_1^2 + m_4 l_2^2 + I_s)\dfrac{r}{2l_1} \\ a_3 = I_{z1} = I_{z2} & a_4 = m_4\Big(\dfrac{r}{2}\Big)^2 \dfrac{rl_2}{l_1^2} \\ a_5 = N_1 f_1 & a_6 = N_2 f_2 \end{array} \tag{2.14}$$

where: $m_1 = m_2$ – wheel masses, m_4 – frame substitute mass, I_s – solid moment of inertia of the robot frame relative to the axis passing through the centre of mass, I_{z1}, I_{z2} – substitute solid moments of inertia of relevant wheels, determined with respect to the axes of rotation of the said wheels, N_1, N_2 – forces of pressure on relevant wheels, and f_1, f_2 – coefficients of relevant wheel rolling friction.

The mathematical model of the mobile two-wheel robot obtained in this way is easy to use in the synthesis of mobile robot control. Determination of precise parameters describing the dynamic properties of the unit is difficult and, hence, smart adaptive systems with dispersed logic, neural networks, or genetic algorithms are often applied for the purpose of control. The mathematical model of the object allows for conducting the initial analysis of relationships between the required wheel driving moments and the adopted masses as well as the parameters imposed on the motion. It allows for initial selection of the type of drive to be used and for definition of mass restrictions.

3. Control of a mobile wheeled robot

The paper focuses on the task of moving by means of the robot. A hierarchical system (Fig. 2) has been adopted as the control architecture.

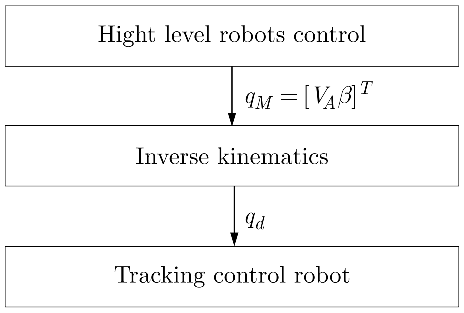

Fig. 2. The hierarchical control system

The role of the source of data on the present robot motion path falls to the master control system, using e.g. external sensors, or to the operator that delivers information in the form of the vector $\mathbf{q}_M = [V_A, \beta]^{\mathrm{T}}$ to the control system shown in Fig. 2. On this basis, an inverse kinematical problem is implemented to generate velocities of the driving wheels in the form of $\mathbf{q}_d = [\alpha_1, \alpha_2]^{\mathrm{T}}$ in the middle layers of the control system. The lowest level of the control system produces velocities of the driving wheels. The algorithms of control of this robot model were simulated by means of Matlab/Simulink package, a mechatronic designing tool. It enables system modelling, simulation, automatic code generation, testing, and validation. Matlab is a digital calculation, visualisation, and programming environment. It allows for developing algorithms, creating applications, conducting parallel calculations, and preparing reports. Simulink enables construction of simulation models by means of a graphical interface.

3.1. Tracking control of a mobile robot

The type of control system depends on whether a linear or nonlinear, stationary or dynamic object is controlled, or on the awareness of the class and accuracy of the mathematical model (Giergiel and Małka, 2006; Hendzel *et al.*, 2008). In the case of the present robot, we are dealing with a dynamic nonlinear object described by Maggie's equations. Control of such an object on the basis of linear methods brings no satisfactory results; this is why more advanced and accurate control methods were intensively developed at the turn of the last decade. Many papers (de Wit *et al.*, 2012; Giergiel and Małka, 2006; Hendzel *et al.*, 2008) refer to object control methods of this kind. The characteristics of available methods of wheeled robot control exceed the scope of the present paper. A follow-up neural control algorithm ensuring creation of preset values of velocity and displacement of robot driving wheels is shown below.

The issues concerning mobile wheeled robot modelling and control are complex, and no systematic approach to the analysis and synthesis of such nonlinear dynamic systems has been developed so far. Due to properties of neural networks such tools are often applied for the purpose of modelling complex nonlinear systems.

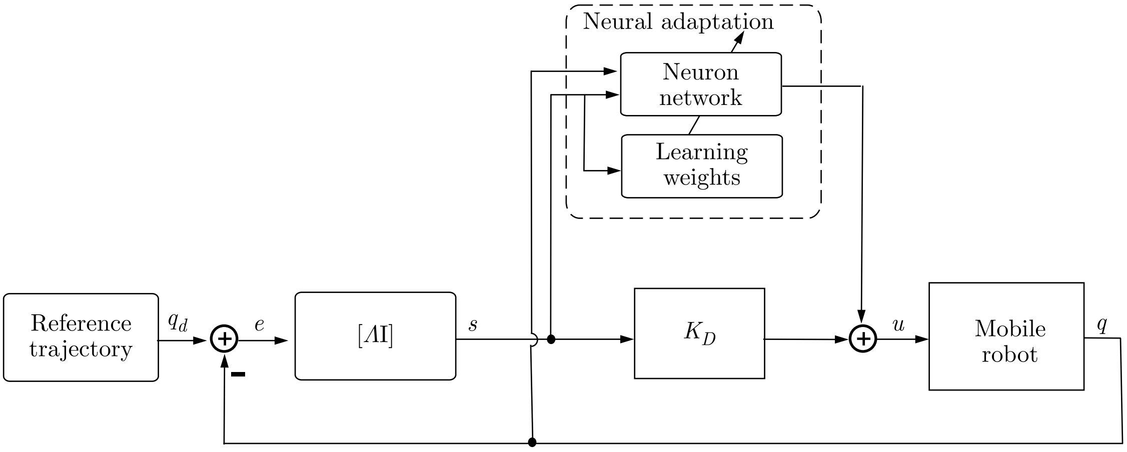

Fig. 3. Structure of a neural control system

As a part of the synthesis of an adaptive neural algorithm, the object nonlinearity has been compensated for with the use of a neural network (Fig. 3). A single layer neural network has been applied to an approximate nonlinear function $f(\mathbf{x})$, where the weights of the input layer of the said network are constant, whereas the weights of the output layer are submitted to the process of learning. A network of this kind is linear due to its parameters. If we select functions of neuron activation $s(\mathbf{x})$ in the form of a set of elementary functions, the neural network with functional extension and ideal weights $\mathbf{W}$ will have the property of approximation of any continuous function defined in a compact set. In this case, the function $f(\mathbf{x})$ will be expressed as

$$f(\mathbf{x}) = \mathbf{W}^{\mathrm{T}}\mathbf{S}(\mathbf{x}) + \boldsymbol{\varepsilon} \tag{3.1}$$

where $\boldsymbol{\varepsilon}$ means the limited approximation error satisfying the relationship $\|\boldsymbol{\varepsilon}\| < \varepsilon_N, \varepsilon_N = \mathrm{const}$. Assuming that ideal weights of the network are limited, the estimate of the function $f(\mathbf{x})$ may be expressed as

$$\widehat{f}(\mathbf{x}) = \widehat{\mathbf{W}}^{\mathrm{T}}\mathbf{S}(\mathbf{x}) \tag{3.2}$$

where $\widehat{\mathbf{W}}$ means the neural network weight estimate determined by means of the adaptation algorithm. The control principle will then become

$$\mathbf{u} = \widehat{\mathbf{W}}^{\mathrm{T}}\mathbf{S}(\mathbf{x}) + \mathbf{K}_D\mathbf{s} \tag{3.3}$$

The description of the closed control system will be expressed as

$$\mathbf{M}(\mathbf{q})\mathbf{s} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{s} + \mathbf{K}_D\mathbf{s} = \widehat{\mathbf{W}}^{\mathrm{T}}\mathbf{S}(\mathbf{x}) + \boldsymbol{\varepsilon} \tag{3.4}$$

The synthesis of a neural system for mobile robot motion control has been conducted on the basis of Lyapunov's stability theory according to the function below

$$L = \frac{1}{2}[\mathbf{s}^{\mathrm{T}}\mathbf{M}(\mathbf{q})\mathbf{s} + \operatorname{tr}(\widehat{\mathbf{W}}^{\mathrm{T}}\mathbf{F}^{-1}\widehat{\mathbf{W}})] \tag{3.5}$$

where: $\widehat{\mathbf{W}}$ – weights estimation error, $\mathbf{F}$ – diagonal matrix with positive elements.

Assuming that the network weight learning algorithm is

$$\widehat{\mathbf{W}}(t) = \mathbf{F}\mathbf{S}(\mathbf{x})\mathbf{s}^{\mathrm{T}} \tag{3.6}$$

We may prove that

$$L = -\mathbf{s}^{\mathrm{T}}\mathbf{K}_D\mathbf{s} + \mathbf{s}^{\mathrm{T}}\boldsymbol{\varepsilon} \leqslant -K_{D\,min}\|\mathbf{s}\|^2 + \varepsilon_N\|\mathbf{s}\| \tag{3.7}$$

where $K_{D\,min}$ is the smallest characteristic value of matrix $\mathbf{K}_D$. Given that ε_N is a constant value $\dot{V} \leqslant 0$, if the condition below is met

$$\boldsymbol{\Psi} = \{\mathbf{s}: \quad \|\mathbf{s}\| > \varepsilon_N/K_{D\,min} \equiv b_s\} \tag{3.8}$$

Relationship (3.8) allows for the conclusion that the generalised follow-up error $\mathbf{s}$ is uniformly limited, at last finally limited, to the set $\boldsymbol{\Psi}$, with a practical limit identified by b_s. Furthermore, the increase in the coefficient $K_{D\,min}$ of the regulator PD will make the follow-up error $\mathbf{s}$ decrease and, consequently, decrease error e and $\dot{e}$ which are limited on the basis of (3.8). This presentation of neural control synthesis ensures faultless operations of a closed control system as a result of the action of an internal loop with the regulator PD until the moment the network starts learning. This means that the initial process of network weight learning is no longer required and that the network weights are estimated in real time. A single layer neural network has been applied for the purpose of approximation of the nonlinear function $f(\mathbf{x})$ to make the network linear due to the parameters, RVFL type (Random Vector Functional Link). It is a network with a random vector of the network first layer weights. Such an approach to the problem means that the neuron activation function may be of the sigma type. A solution of this kind has been used in further Sections of this paper. The algorithm presented above allows for creating driving wheel motion with a limited convergence error and will be used as a low-level control system in further Sections of this paper.

3.2. Master control system

Various algorithms may be employed in the master control system (Giergiel and Małka, 2006; Hendzel *et al.*, 2013, 2015; Szuster *et al.*, 2014). This paper presents Braitenberg's algorithm which ensures access to the target location by driving past obstacles. Such solutions originate from the attempt to replicate the behaviour of living organisms, e.g. insects. The behaviour of a robot depends on the scheme of selection of the parameters. In our case, the mobile robot is equipped with ultrasonic sensors located on the perimeter (Fig. 4). Classification of the sensors into groups allows for obtaining information in the form of the distance from the obstacle on the left side d_L, on the front side d_F and on the right side d_R.

Our control system fulfils the task which consists in arriving at the centre of free space in compliance with the relationship below

$$\begin{bmatrix} u_{VS} \\ u_{\beta S} \end{bmatrix} = \begin{bmatrix} \dfrac{r}{2} & \dfrac{r}{2} \\ \dfrac{r}{d} & -\dfrac{r}{d} \end{bmatrix} \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{32} \end{bmatrix} \begin{bmatrix} d_L \\ d_R \\ d_F \end{bmatrix} \tag{3.9}$$

where: $w_{ij} =$ const – parameters selected through experimentation.

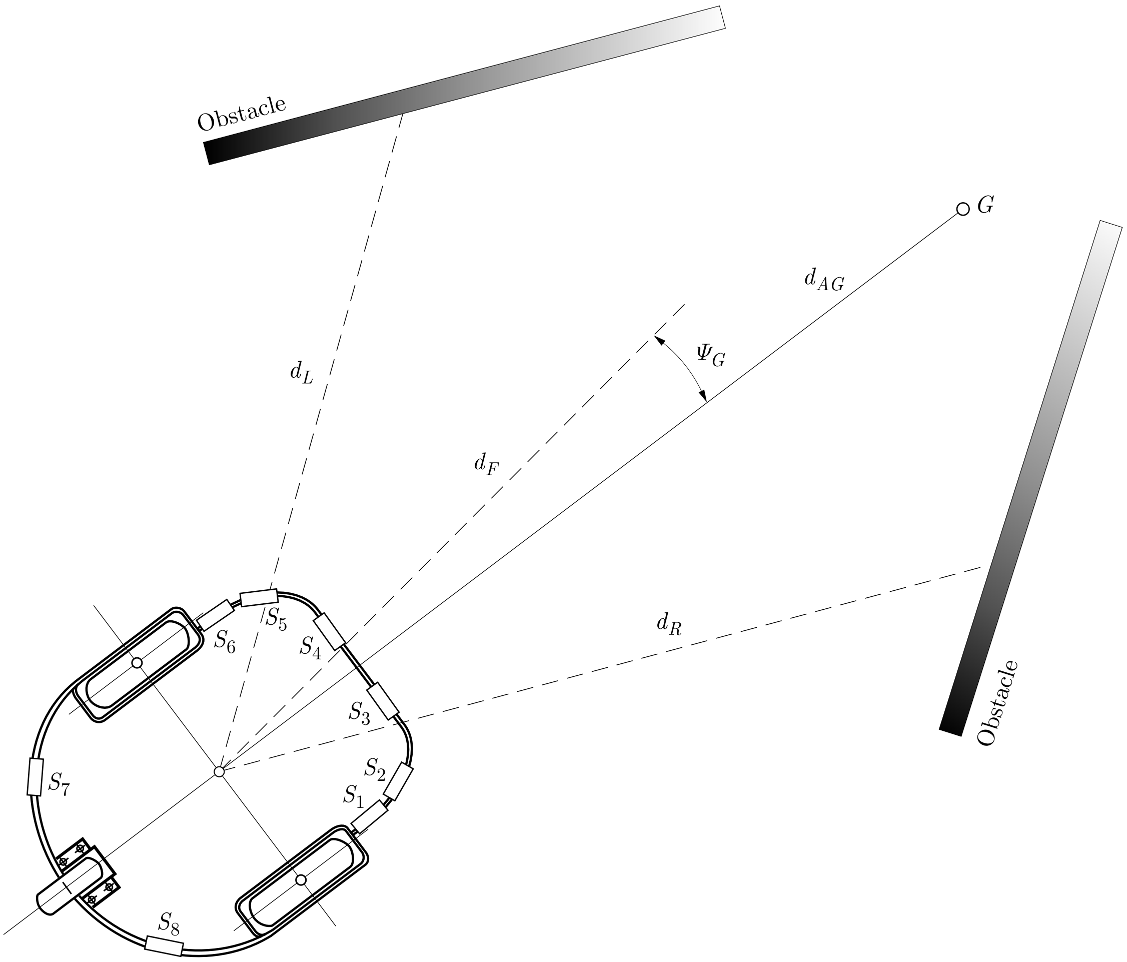

Fig. 4. Diagram of a mobile robot with sensors

The task of "go to the target place" has been carried out in compliance with the relationship

$$\begin{bmatrix} u_{VG} \\ u_{\beta G} \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{d} & -\frac{r}{d} \end{bmatrix} \begin{bmatrix} w_{11} & w_{11} \\ w_{21} & w_{22} \end{bmatrix} \begin{bmatrix} \Psi_G \\ d_{AG} \end{bmatrix} \tag{3.10}$$

where: Ψ_G – angle of robot deviation from the target place, d_{AG} – distance between point A and target point G, w_{ij} = const – parameters selected through experimentation.

Co-operative compilation of the behaviours has been conducted in compliance with the relationship

$$u_\beta = b_1 u_{\beta G} + b_2 u_{\beta S} \qquad\qquad u_v = \min(u_{VS} + u_{VG}) \tag{3.11}$$

where: coefficients $b_{(\cdot)}$ should be selected through experimentation.

Finally, the robot preset motion in the form of point A velocity and frame angular velocity is defined as

$$\begin{bmatrix} V_A \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} V_{A\,max} & 0 \\ 0 & \dot{\beta}_{max} \end{bmatrix} \begin{bmatrix} u_V \\ u_\beta \end{bmatrix} \tag{3.12}$$

The quantities $V_{A\,max}$, $\dot{\beta}_{max}$ in relationship (3.12) are, respectively: the maximum velocity of point A and the maximum velocity of the mobile robot frame, whereas u_V and u_β are standardised elements of the control vector, such that

$$u_v \in \langle 0,1\rangle \qquad u_\beta \in \langle -1,1\rangle \tag{3.13}$$

The algorithm presented above has been implemented in the mobile wheeled robot designed and built, and allowed achievement of the preset target in an unknown environment.

4. Structure prototyping

The analysis of the mathematical model of the mobile robot will be followed by the process of designing a real object. In compliance with the assumptions, the robot should have two driving wheels and one supporting wheel. To give the structure an appropriate shape, we need to consider which modules should be located inside the unit. The interior of the robot will be mainly filled with electronic circuits that perform the preset functionality of the platform. Modern designing processes are conducted with the use of applications such as CAD/CAX in the role of mechatronic design tools. Autodesk Inventor 2013 was used to build the elements of the robot. Apart from the basic body design module, the said application contains many building-assisting modules. Our building process employed an extensive base of standardised elements. The main assumptions about the structural properties of the housing were: simple shape, a small number of components, and the potential for expansion. The design includes no fixing elements, such as, e.g. bolts or nuts. Such elements have no major impact on the shape of the structure. Many ideas for the prototype were generated in the period of robot designing. They were modified parallel to the acquisition of results of simulations and the production of new solutions. Modifications and simulations of structural solutions were possible thanks to the use of advanced CAD/CAX software.

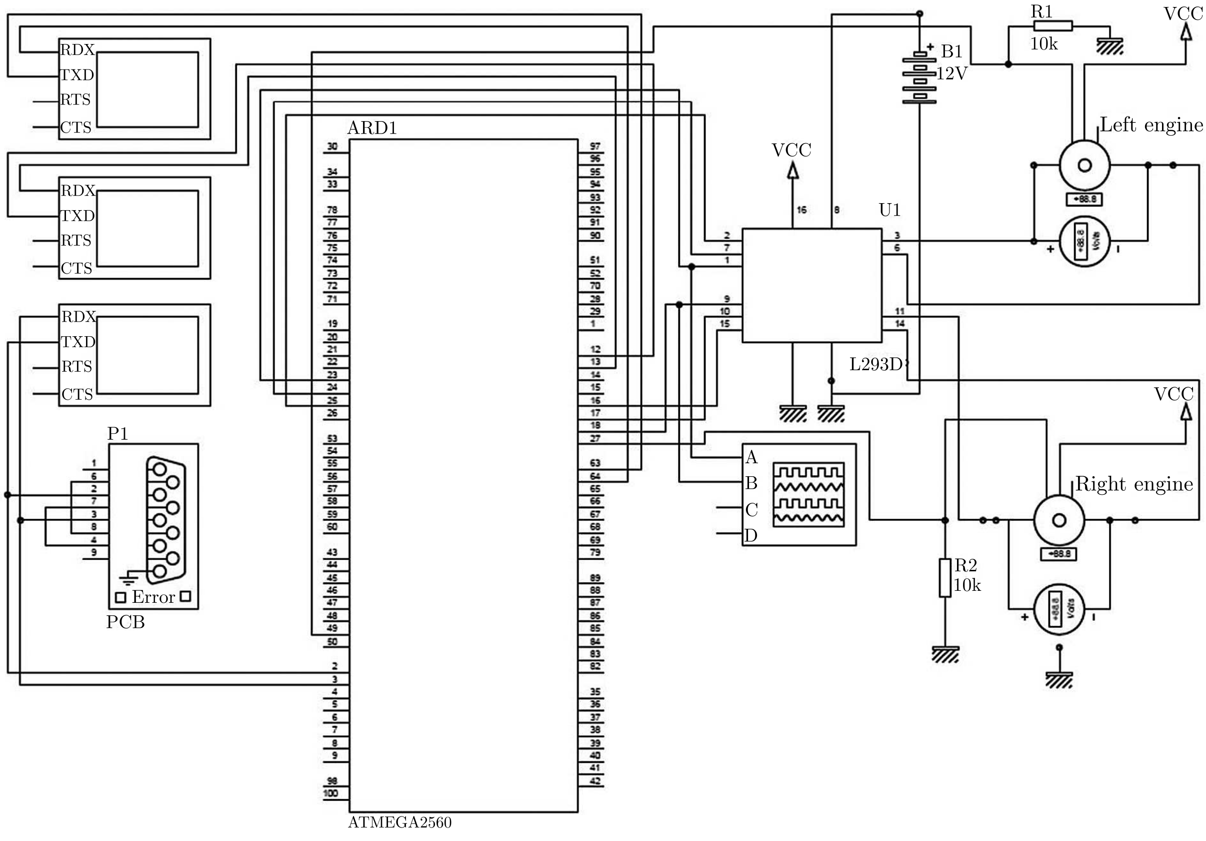

Fig. 5. Diagram of the electronic system designed in the ISIS program

There are many methods of controlling robots and other devices. Most control systems are built of electronic elements thanks to the evolution of digital technology. Typically, a control system includes a logical part and actors. In our robot, the logical circuit is created by an ARDUINO MEGA system with integrated ATmega2560 processor. The electronic circuit with microcontroller ensures the complete control of sensors and allows generation of suitable low-current control signals (Fig. 5). Control of the selected drive system requires a power stage. In our case, the power stage is based on an L293D circuit. This circuit contains two H bridges which ensure complete control of PWM signals and DC engines in two senses of rotation. The control system was simulated by means of an ISIS application, another mechatronic design tool. Electric engine models with encoders, power stage and microcontroller with a simple control program were applied as elements of the flowchart.

Two communication modules employing 433 MHz wave AM radio and Bluetooth transmission were selected for the purpose of robot design. All additional modules are located on the board adapted to the terminals of the ARDUINO board; this solution allows elimination of the need to install many physical wires. The unit contains a DHT11 module for temperature and humidity measurements as well as a BMP180 module responsible for delivery of information on temperature, air pressure and altitude with regard to the reference level. The electronic circuit contains I2C, UART, SPI, and OneWire interfaces which ensure the connection of any number of various measurement modules. The first software tests were carried out on a universal plate and in a simulator. This ensured that there were no inconsistencies in the design of the system and writing of the source code.

5. Software prototyping

Our structure contains four processor circuits. A PIC16F628A microchip is responsible for encoding and decoding the 433 MHz radio transmission. The Bluetooth module contains a circuit for control of this module and of the transmission. The Arduino R3 board contains two microcontrollers: ATmega16AU for conversion of UART transmission into USB transmission, and ATmega2560 whose memory is used for the implementation of the main robot control software. Apart from many microcontroller series, various programming environments are also available. Developed by MPLAB IDE, the software for the PIC system allows for compilation of the software created in C/C++ language. Given a considerable number of examples and comprehensive technical support, we decided to create the microcontroller software using Arduino 1.5.8 application (Kardaś, 2011). An ATmega2560 microprocessor is the main robot controller. It contains the software for handling two robot drives, two encoders coupled with the drives, a set of five ultrasonic sensors, two motion sensors as well as data transmission through Bluetooth, ATmega16 and PIC16f628A microprocessors. The robot software contains the path implementation generator, follow-up controller and elementary behaviour generator. All the processes were written in C language. The entire software structure relies on two command loops. The first loop is responsible for control vector acquisitions. It contains a number of equations presented in Section 2 and operates in compliance with the algorithm presented in Fig. 6.

The analysis of the algorithm shown in Fig. 6 allows one to conclude that the loop will be executed if the status of the variable “interr” is set as “high”. The setup of this variable is followed by calculations according to the order of precedence embedded in the algorithm. The last elements of the loop are responsible for sending the PWM signal to the controlling inputs of the drive system power bridge. The call frequency of the main loop may be adjusted by changing the period of the interruption in whose cycle the system updates the robot status vector.

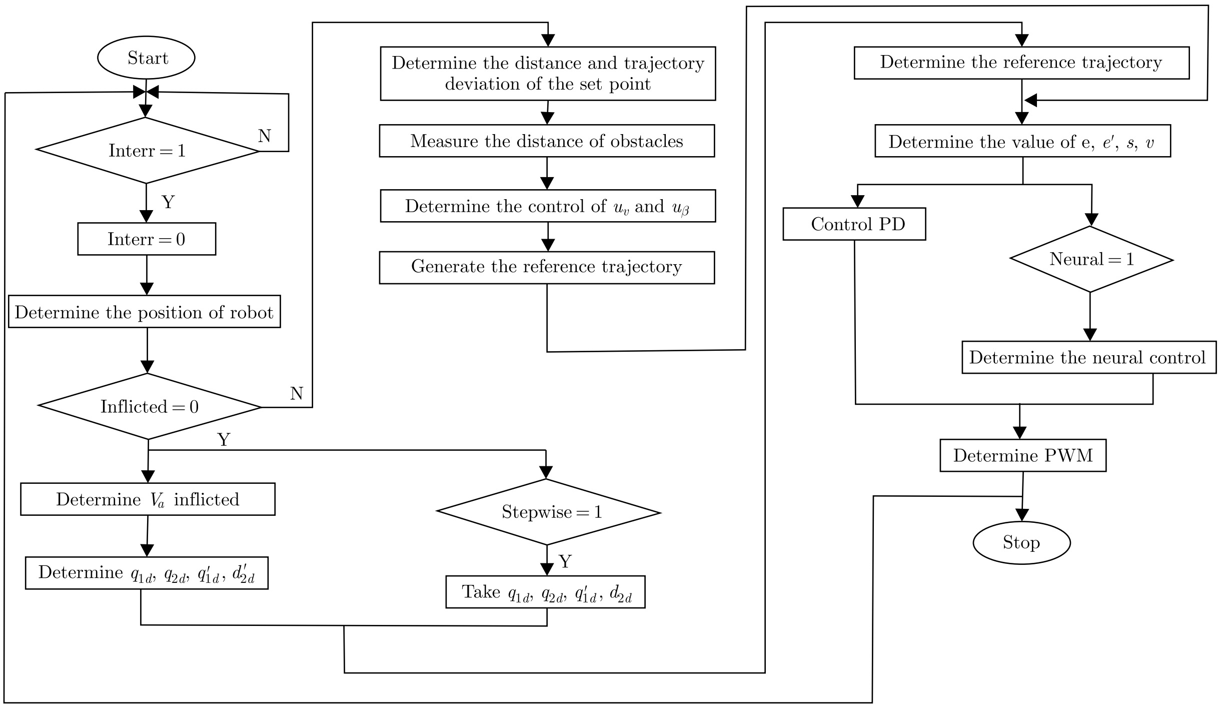
Start
Interr = 1
N
Y
Interr = 0
Determine the position of robot
Inflicted = 0
N
Y
Determine V_a inflicted
Determine q_{1d}, q_{2d}, q'_{1d}, d'_{2d}
Stepwise = 1
Y
Take q_{1d}, q_{2d}, q'_{1d}, d_{2d}
Determine the distance and trajectory deviation of the set point
Measure the distance of obstacles
Determine the control of u_v and u_β
Generate the reference trajectory
Determine the reference trajectory
Determine the value of e, e', s, v
Control PD
Neural = 1
Determine the neural control
Determine PWM
Stop

Fig. 6. The robot software main loop algorithm

6. Real object prototyping

Having developed a 3D model and completed the software elements, we began to build the real object for the purpose of validation. The components, such as: base, sensor bows and angle sections for engine installation were built on the basis of the CAD model. In that case, we opted for methods of quick prototyping using 3D printing technologies.

The accuracy of our printer was 0.5 mm which, combined with the same value (again, 0.5 mm) of the nozzle diameter, had an impact on a considerable inaccuracy of positioning the process holes. Minor fitting and drilling adjustments were made at the final stage of production. All the above-mentioned works resulted in satisfactory precision of component production and permitted proceeding to the final assembling works. Figure 7 shows a comparison of the CAD model (a) and its real object (b).

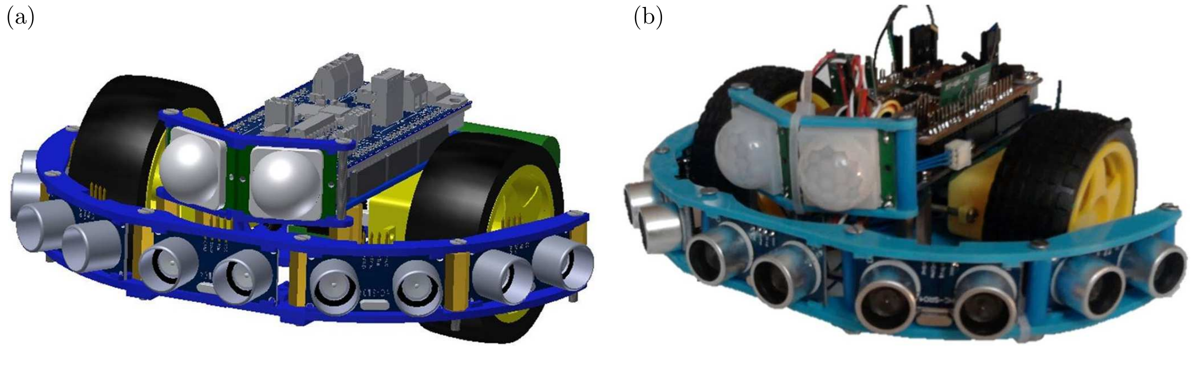

Fig. 7. (a) Comparison of CAD model (b) with a picture of the real prototype

As illustrated above, the visualisation of the robot shows minor differences from the real object. Such differences result from some necessary installation and adjustment operations during the assembling works. Furthermore, the CAD model was not equipped with electronic circuit wiring. The differences shown in the real object have no major impact on the robot functionality. Robot software was developed as a parametric application. This means that the variables connected with physical characteristics of the object of control may be modified at random. Robot software should be supplied with the values of several mechanical and utility features. Structural parameters are: driving wheel track l_2 or driving wheel radius r. Moreover, the system requires data on the resolution of the actual encoder. The parameters which result from dynamic properties of the structure rather than from its constitution are as follows: robot maximum velocity $V_{A\,max}$, frame maximum angular velocity $\dot{\beta}_{max}$, and maximum limitations on the pulse-width modulation signal PWM_{max}. The simulation allowed determination of the parameters of low- and high-level control circuits. The characteristic of wheel angular velocity as a function of the control signal may be obtained by changing the robot PWM signal from 0 up to 50% and from 50 down to 0% as well as by receiving information from the robot on the said signal and on values of the measured angular velocity. In this case, the measurement of robot acceleration and braking will be required. This results from the existence of the static torque the drive must overcome during acceleration. This phenomenon is absent when the velocity is reduced. The hysteresis of the system is visible on the characteristics in Fig. 8. Limitation on the PWM signal results from the fact that the maximum engine voltage is lower than the power supply source voltage.

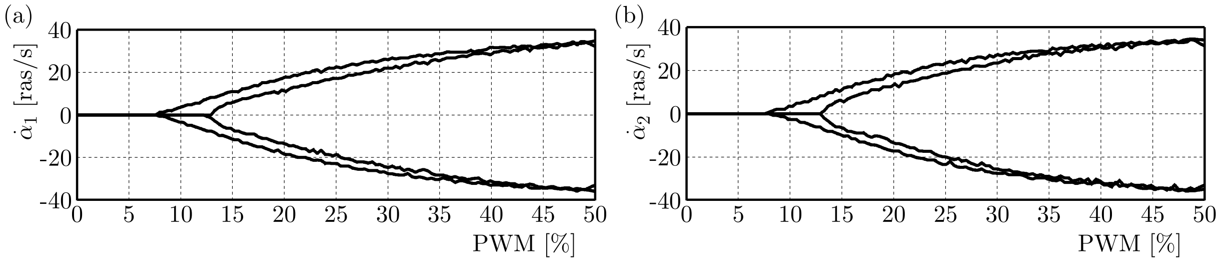

Fig. 8. Characteristic of wheel angular velocity as a function of PWM signal

The characteristic allows determination of three parameters. The maximum PWM signal is located at the point where the PWM signal derivative has a very small value. The second parameter is the robot maximum velocity which may be calculated from a simple formula

$$V_{A\,max} = \frac{\dot{\alpha}_{1\,max} + \dot{\alpha}_{2\,max}}{2} r \tag{6.1}$$

The last parameter determined from the characteristic shown in Fig. 8 is the frame maximum angular velocity

$$\dot{\beta}_{max} = \frac{r}{2l_1}(\dot{\alpha}_{1\,max} + \dot{\alpha}_{2\,max}) \tag{6.2}$$

The frame angular velocity achieves its maximum value during rotation of the robot at the wheel maximum angular velocity relative to the centre of rotation which, at a given moment, falls to the characteristic point A of the robot. Another stage of determination of the parameters necessary for robot control is the tuning of the conventional regulator PD. For the purpose of tuning, Ziegler-Nichols methods were employed. The first way of selection of setup values uses the step response of the regulator with decoupled feedback and gain $K_p = 1$. Relevant setup values may be calculated by means of determination of the maximum slope of the curve linear derivative and of the response delay time. Another method is the experimental way. Having decoupled the differentiating element, we must set such a gain K_{kr} of the proportional regulator that ensures the occurrence of non-extinguishing responses to the step input function. At the next stage, we assume $K_p = 0.6K_{kr}$ and select the gain K_d for shortening the response delay time. The results of simulation are shown in Fig. 9.

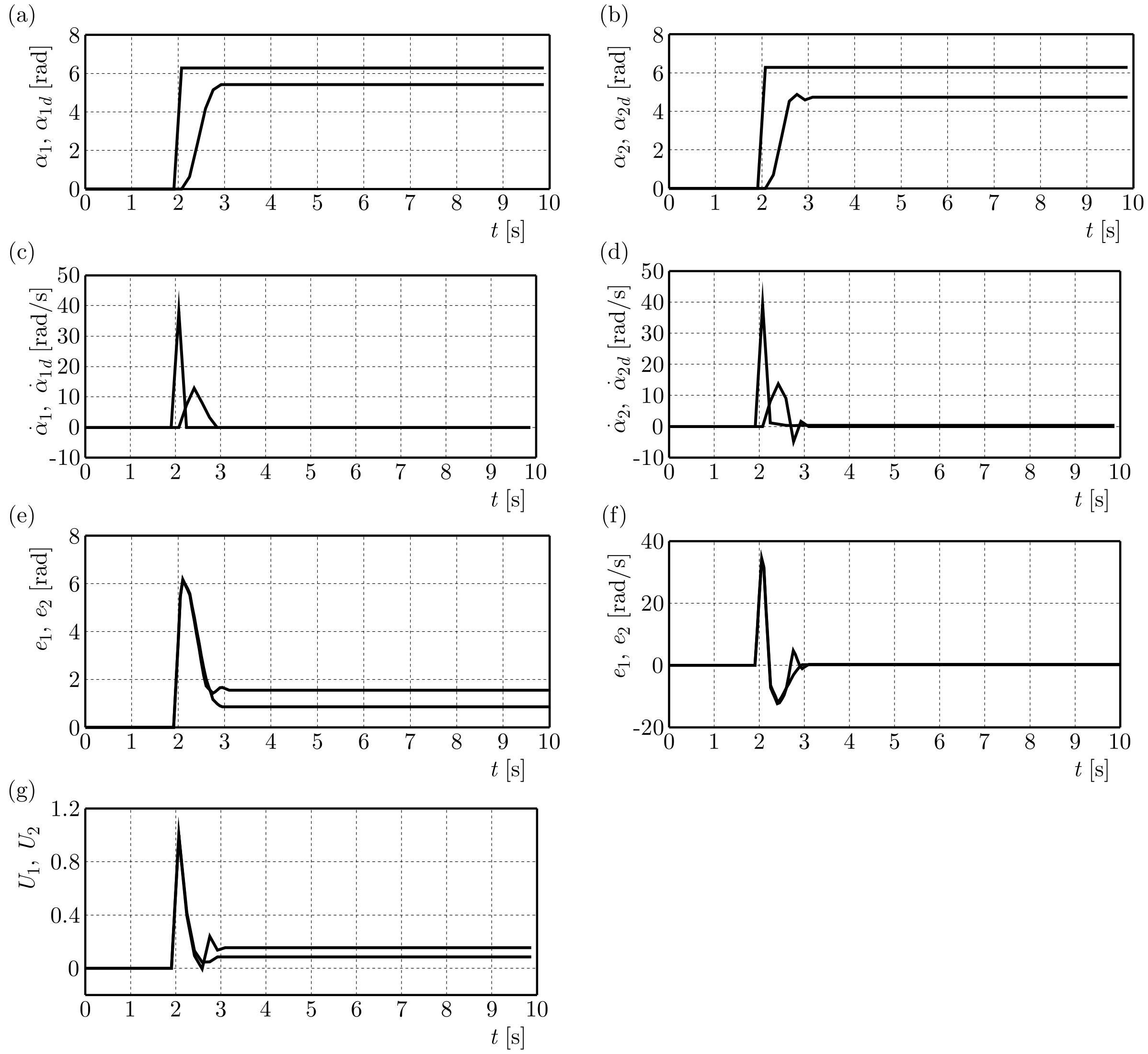

Fig. 9. The results of PD simulation control: (a) angle of rotation of wheel 1, (b) angle of rotation of wheel 2, (c) angular velocity of wheel 1, (d) angular velocity of wheel 2, (e) errors to keep up the angle of rotation of the wheels, (f) errors to keep up the angular speed of the wheels, (g) control

The analysis of the above-mentioned diagrams allows one to conclude that statistical deviations of control as well as any other over-regulations are acceptable with regard to the accuracy of control. We adopted the quality markers as the mean-square errors of angular velocity and wheel rotation angle: $RMSE_{e1} = 1.4245$, $RMSE_{e1p} = 5.3511$, $RMSE_{e2} = 1.7798$, $RMSE_{e2p} = 5.4488$. A further step in the simulation of PD control adopted in this way consisted in the addition of nonlinearity offset control in the form of a neural network RVFL.

The output weights W are determined by an appropriate learning algorithm with zero initial values of weights. The input weights V were generated at random and adopted as a statistical matrix. The results of neural control analysis are shown in the diagrams in Fig. 10.

The diagrams in Fig. 10 allow us to see an improvement in the accuracy of control. Quality markers for neural control are as follows: $RMSE_{e1} = 1.3356$, $RMSE_{e1p} = 4.8893$, $RMSE_{e2} = 1.3012$, $RMSE_{e2p} = 5.0191$. The pattern of neural network weight adaptation is shown in Fig. 11.

It can be noticed that the weights move towards certain values and oscillate around such values. Having completed the simulation, we assumed that the average values of the weights adopted in the steady state would be the initial weights of the process of network learning.

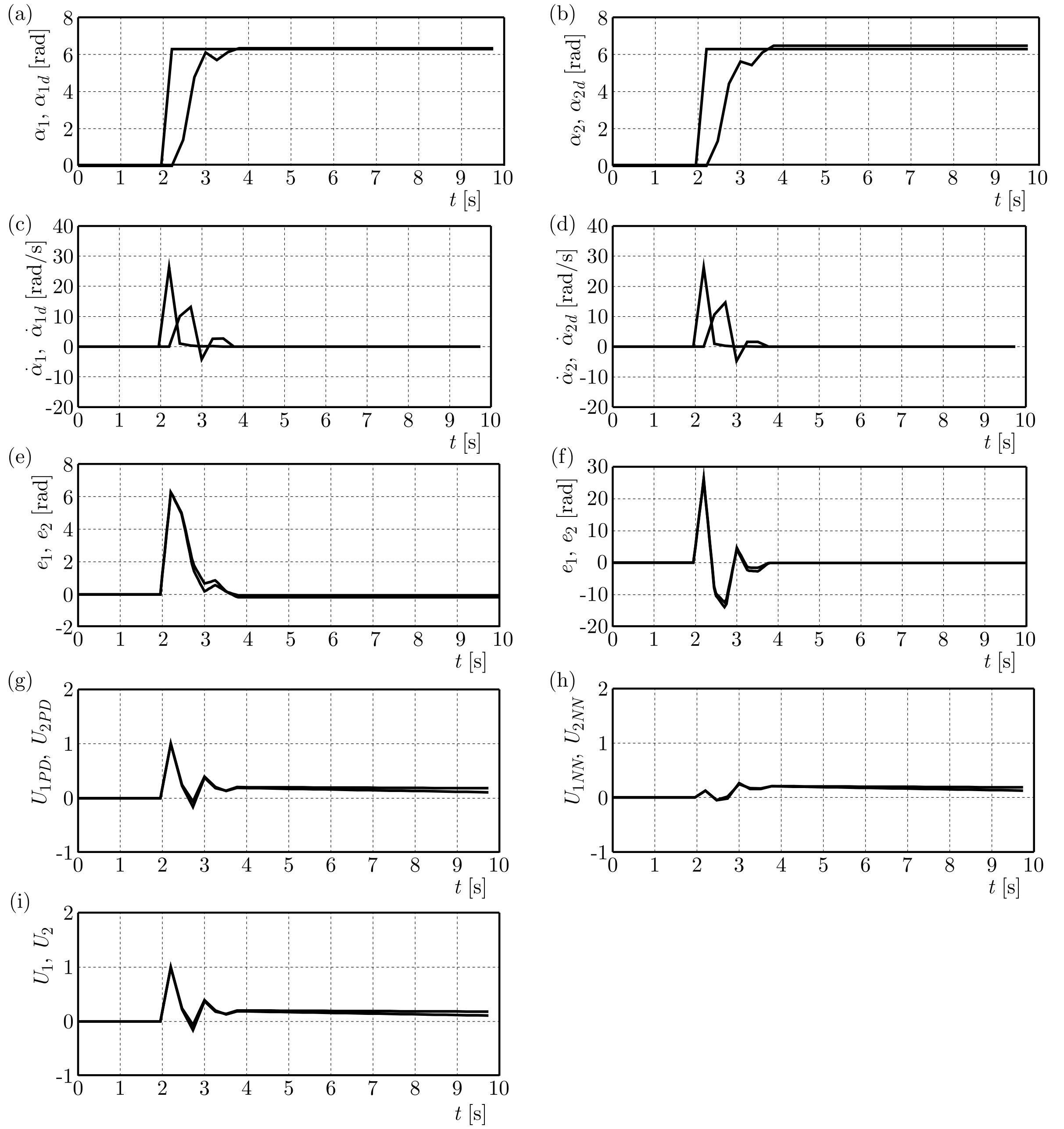

Fig. 10. Results of neuronal simulation control: (a) angle of rotation of wheel 1, (b) angle of rotation of wheel 2, (c) angular velocity of wheel 1, (d) angular velocity of wheel 2, (e) errors to keep up the angle of rotation of the wheels, (f) errors to keep up the angular speed of the wheels, (g) control PO, (h) neural control, (i) control

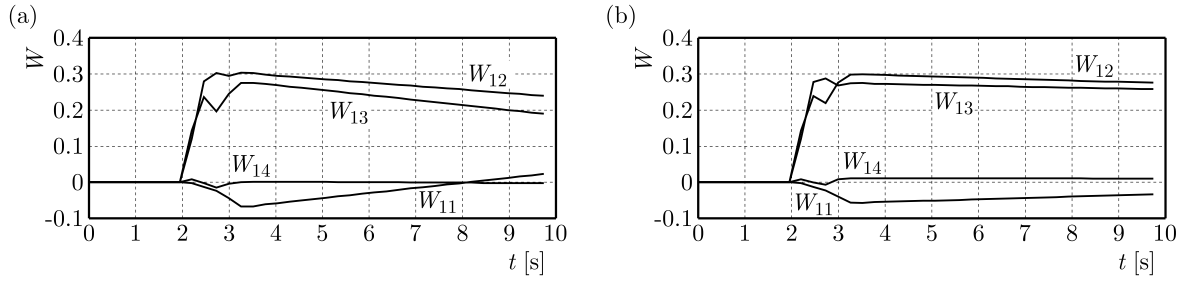

Fig. 11. The process of adaptation of the weights of the neural network: (a) weights *RVFL* wheel 1, (b) weights *RVFL* wheel 2

Quality markers for neural control have better values than the markers for PD control. The use of the neural network compensated for the differences in drive dynamics and decreased the value of statistical deviation.

7. Conclusions

The paper presents a real example of the methodology of mechatronic designing. It gives a summary of all the stages of this process. Given the comprehensive bibliographic references to kinematics, dynamics and control of mobile wheeled robots, the report on this stage of mechatronic designing is reduced. We focused on the prototyping of the structure, electronic circuits and software. Further Sections listed the tools applied in mechatronic designing along with short descriptions. Thanks to the use of CAD software and of electronic circuit design and simulation applications, as well as a 3D printer, the time to the end of prototype production was reduced to the minimum. The robot we obtained may be designed for implementation and validation of control algorithms thanks to its sensor circuits and software. The paper presents the methodology applied for selection of PD regulator setup values as well as the process of weight learning by the neural network which compensates for nonlinearities of the object. Further works will focus on validation of driving along a preset path, Braitenberg's algorithm functioning and a robot driving towards a preset target location.

References

1. Acar M., Makara J., Penney E., 1994, *Mechatronics. The Basis of New Industrial Development*, Computational Mechanics Inc. Ashurst Lodge, Southampton UK
2. Amerongen van J., Coelingh E., de Vries T.J.A., 2000, Computer support for mechatronic control system design, *Robotics and Autonomous Systems*, **30**, 3, 249-260
3. Bishop R.H., 2006, *Mechatronics: An Introduction*, CRC Press Taylor & Francis Group
4. Buratowski T., Cieślak P., Giergiel M., Uhl T., 2012, A self-stabilising multipurpose single-wheel robot, *Journal of Theoretical and Applied Mechanics*, **50**, 1, 99-118
5. de Wit C.C., Siciliano B., Bastin G., 2012, *Theory of Robot Control*, Springer Science & Business Media
6. Duda S., Gąsiorek D., Gembalczyk G., Kciuk S., Mężyk A., 2016, Mechatronic device for locomotor training, *Acta Mechanica et Automatica*, **10**, 4, 310-315
7. Gawrysiak M., 1997, *Mechatronics and Mechatronics Design* (in Polish), Wyd. Politechniki Białostockiej
8. Giergiel J., Hendzel Z., Żylski W., 2000, *Kinematics, Dynamics and Control of Mobile Robots in Mechatronics* (in Polish), Monografia Wydz. IMiR, AGH Kraków
9. Giergiel M., Hendzel Z., Żylski W., 2002, *Modeling and Control of Mobile Wheel Robots* (in Polish), PWN, Warszawa
10. Giergiel J., Kurc K., Szybicki D., 2014, *Mechatronics Crawler Inspection Robots* (in Polish), Oficyna Wydawnicza Politechniki Rzeszowskiej
11. Giergiel M., Małka P., 2006, Modeling of kinematics and dynamics of mobile mini-robots (in Polish), *Modelowanie Inżynierskie*, **32**, 157-162
12. Hendzel Z., Burghardt A., Szuster M., 2013, Adaptive critic designs in control of robots formation in unknown environment, [In:] *International Conference on Artificial Intelligence and Soft Computing*, Springer, Berlin, Heidelberg, 351-362

13. Hendzel Z., Burghardt A., Szuster M., 2015, Artificial intelligence algorithms in behavioural control of wheeled mobile robots formation, [In:] *Computational Intelligence*, Springer, Cham, 263-277

14. Hendzel Z., Żylski W., Burghardt A., 2008, *Autonomous Mobile Robotics, Mechatronic Design and Control* (in Polish), Oficyna Wydawnicza Politechniki Rzeszowskiej, Rzeszów

15. Kardaś M., 2011, *AVR Microcontrollers Language C Fundamentals of Programming* (in Polish), Wydawnictwo Atnel, Szczecin

16. Kurc K., Szybicki D., 2011, Kinematics of a robot with crawler drive, *Mechanics and Mechanical Engineering*, **15**, 4, 93-101

17. Kurc K., Szybicki D., Burghardt A., Muszyńska M., 2016, The application of virtual prototyping methods to determine the dynamic parameters of mobile robot, *Open Engineering*, **6**, 1

18. Mężyk A., 2010, Mechatronics in the design of special tracked vehicles (in Polish), *Transport Przemysłowy i Maszyny Robocze*, **4**

19. Mężyk A., Świtoński E., Kciuk S., Klein W., 2011, Modelling and investigation of dynamic parameters of tracked vehicles, *Mechanics and Mechanical Engineering*, **15**, 4, 115-130

20. Szuster M., Hendzel Z., Burghardt, A., 2014, Fuzzy sensor-based navigation with neural tracking control of the wheeled mobile robot, [In:] *International Conference on Artificial Intelligence and Soft Computing*, Springer, Cham, 302-313

21. Turowski J., 2008, *Basics of Mechatronics* (in Polish), Wyd. Akademii Humanistyczno-Ekonomicznej w Łodzi, Łódź

Manuscript received April 24, 2017; accepted for print July 24, 2019