applied sciences

MDPI

Article

Neural Tracking Control of a Four-Wheeled Mobile Robot with Mecanum Wheels

Mateusz Szeremeta and Marcin Szuster *

Faculty of Mechanical Engineering and Aeronautics, Rzeszow University of Technology, 35-959 Rzeszow, Poland; m.szeremeta@prz.edu.pl

* Correspondence: mszuster@prz.edu.pl

Abstract: This study designed an algorithm for the intelligent control of the motion of a mobile robot with mecanum wheels. After reviewing the model kinematics and dynamics of the robot, we conducted a synthesis of the neural control algorithm to determine network weight adaptation, according to Lyapunov stability theory. Using a MATLAB/Simulink computing environment, we developed a numerical simulation for the implementation of the robot's motion path with parametric disturbances acting on the control object. To determine the quality of the implementation of the desired motion path, a numerical test of the robot's motion, controlled with the use of a PD controller, was conducted. The proposed control algorithm was verified on a laboratory stand equipped with a dSpace DS1103 controller board and a Husarion Panther four-wheeled mobile robot with mecanum wheels. The conducted research confirmed the improved implementation of the desired motion path by a robot controlled with the use of an intelligent control system.

Keywords: wheeled mobile robot; intelligent control; tracking control; neural networks; mecanum wheels; Lyapunov stability theory

Citation: Szeremeta, M.; Szuster, M. Neural Tracking Control of a Four-Wheeled Mobile Robot with Mecanum Wheels. *Appl. Sci.* **2022**, *12*, 5322. https://doi.org/10.3390/app12115322

Academic Editor: Antonio Fernández-Caballero

Received: 31 March 2022
Accepted: 20 May 2022
Published: 24 May 2022

1. Introduction

The expanding possibilities of wheeled mobile robots (WMR), such as maneuverability, communication, battery life, and the possibility of using sensors that enable data collection from the environment, has led to a growing interest in their industrial, medical, and scientific applications. The challenge is in designing robots for a dynamic work environment and meeting the required safety, accuracy, reliability, and maneuverability. Solutions such as mecanum wheels [1], which enable freedom of movement and advanced intelligent control systems such as artificial neural networks (NNs) [2–6] that provide real-time approximation of nonlinearities in a mathematical model of a robot using network weight adaptation, may prove helpful in this process.

The use of omnidirectional wheels, such as mecanum wheels, in the construction of a robot allows direct control of all degrees of freedom, which leads to its classification as a holonomic object. A description of a kinematics model of a holonomic robot with mecanum wheels can be found, inter alia, in [7–14] These previous studies have resolved the inverse kinematics problem, determining the angular velocities of the wheels based on the set velocity of the characteristic point of the robot's platform in relation to a stationary reference system [7,8], and in relation to a moving coordinate system [9,10]. The next stage of the mathematical analysis of an object has involved describing its dynamics [15–20]. This task was performed in [15–17], using Lagrange equations of the second kind [15] and Lagrange equations of the second kind with multipliers [16,17]; as a result, the simple and inverse dynamics problems of the WMR were solved. In [21], a description of a WMR dynamics model using Lagrange equations of the second kind was identified with the use of a real object. It was confirmed that the WMR could map a desired motion path based on the values of the wave forms of the driving moment determined by solving the inverse

dynamics problem. The effect of losing control over the robot was also considered under conditions of a low friction coefficient, which led to skidding, which has been a typical problem for the control of vehicles equipped with mecanum wheels.

The control of mechatronic objects had been widely studied [22–26]. Traditional mechatronic systems have commonly used intelligent control methods [27–37], while for holonomic mobile robots, traditional control algorithms have often been used [38–43]. A small number of studies have combined both topics by presenting intelligent control algorithms for holonomic (mecanum) mobile robots [44–50]. The use of intelligent control enables the synthesis of a control law with the adaptation of the control system parameters. The implementation of such functionality has been possible due to the use of an NN, which is a universal approximator of function [51,52]. The studies in [53,54] presented the concept of WMR control based on a neuro-fuzzy control system with wavelet networks. A numerical simulation was conducted to confirm that a WMR could implement the desired motion path. However, these works did not provide information on the quality of control expressed by the tracking errors or the adopted quality indicators. In addition, the control method was not compared with other known robot control algorithms. For this reason, it was not clear to what extent the proposed method differed from known control algorithms.

The study in [55] presented an adaptive sliding control system containing a radial basis function (RBF) network. The nonlinear function, which is part of the robot dynamics description, was approximated using a three-layer radial NN. Based on the analysis of the Lyapunov stability, the law of network weight adaptation was determined. Numerical simulations and tests were conducted on a real object. The conducted research showed that the proposed adaptive-sliding algorithm provided the best control quality among the tested algorithms (e.g., PID controller and sliding control with linearization).

The studies in [56–58] presented methods of motion control of a holonomic WMR with the use of a fuzzy control algorithm. In [56], the orientation angle of a robot (i.e., yaw heading) was determined using an inertial measurement unit (IMU LSM303DLH). Then, a fuzzy WMR motion control algorithm was applied. The fuzzy controller (FLC) contained a base of 25 rules and had two inputs for the state vector error and the error between current and the previous iteration of the state vector; a Sugeno model with a Mamdani implication was used. The output of the FLC control algorithm determined the work cycle of the PWM signal that controlled the angular velocity of the mecanum wheel motor. A simulation of controlling the robot's motion along a path in the shape of a circle, an ellipse, and a spiral was conducted. The results were presented in the form of a graph of the motion path implemented by the WMR, using only the solution of the inverse kinematics problem, and for a waveform using the IMU system and fuzzy control algorithm. Unfortunately, it was difficult to assess the quality of control due to the lack of state vector error graphs and quality indicators. Additionally, the presented results did not provide a comparison between the fuzzy control algorithm and another algorithm.

In [57], a control algorithm with a fuzzy PID controller was used. The authors focused a significant portion of the research on the description of the hardware structure of the test platform used and the communication procedures between the components of the robot. The rules for the behavior of the FLC, in the absence of measurements of the angular motion parameters of a given mecanum wheel, were considered. There was no description of the quality of the implementation of the desired motion path. Only the conceptual implementation of the obstacle avoidance behavior by the WMR and the possible translation and rotation of the robot platform in the case of controlling the rotation of two mecanum wheels were presented.

In [58], as opposed to the traditional method of a mathematical description of a robot model by solving the kinematics and dynamics problems, simulation modelling was applied. A physical model, built in Solidworks, was transferred to the Adams environment, where the robot dynamics model was determined by defining constraints, actuators, and sensors. Then, the simulation was performed using both the Adams program and the MATLAB/Simulink development environment. A fuzzy control algorithm was used that

consisted of three FLCs (one controller each for two translations and one rotation), each using a base of 25 rules with two inputs for error and error difference from the current and previous iteration, one output, and five membership functions for each variable. To implement the FLC, the MATLAB fuzzy logic toolbox with the Mamdani implication was used. The results of the simulation tests were presented; however, no quality indicators were used to assess the quality of motion implementation by the WMR control algorithm.

To verify the correctness of the operation of the proposed method of intelligent control for a mobile robot with mecanum wheels and to assess the quality of the motion implementation, a synthesis of a neural algorithm was conducted. An RVFL (random vector functional link) NN was used [59–62]; the law of weight adaptation was determined based on the mathematical model of a robot using the Lyapunov stability theory. Next, numerical simulations and verification tests were conducted, the results of which enabled the evaluation of the control quality of the tested method in comparison to a PD controller.

Our study evaluated a neural tracking control algorithm of a WMR with the use of the RVFL NNs, demonstrating the accuracy of the motion via numerical tests and studies on a real object, particularly when disturbed. The use of the adaptive nonlinearity compensator for the description of the dynamics of the control object made it possible to adjust the parameters of the control signal with parametric disturbances affecting the motion of the control object, which was confirmed by both simulation studies and tests on the real object. The most unfavorable case of selecting zero initial values of the weights of the hidden network layer was used, which corresponded to the lack of preliminary knowledge regarding the controlled process contained in the weights. Moreover, an analysis of the algorithm's stability was conducted with the use of the Lyapunov stability theory. This study confirmed the correctness of the tracking control in verification tests conducted on a real object, the Husarion Panther four-wheeled mobile robot with mecanum wheels, which was modified for real-time experiments using the dSpace DS1103 control and measurement card. In addition, a significant contribution was the comparison of the results of the proposed algorithm with another algorithm, in this case, the PD controller, both in terms of simulation tests and verification with a real object.

The control algorithm presented in this study will be used in future research in the tracking control layer of the hierarchical control system of the four-wheeled mobile robot with mecanum wheels, which performed the task of a large-size transport in cooperation with the second robot. The cooperation of robots was necessary for the correct implementation of transporting large objects in tight storage spaces, where the advantages of the mecanum wheels and the robots maneuverability could be examined. Considering the complexity of the parent layer of the control algorithm coordinating the robot formation and the desired trajectories of movement as well as the forces of the object's impact on the assembly points of the robot frame and the distance to obstacles, a neural control algorithm was used in the tracking control layer. This algorithm, as shown in the research, ensured accuracy of the movement, measured by the quality indicators used, and had low computational demand while adapting the control signal to the changing work conditions of the robot, which is particularly important for transport tasks involving large objects.

2. Description of the Control Object

To synthesize the control algorithm, the mathematical model of the holonomic motion of the WMR was analyzed. The waveforms of motion parameters were determined and served as the predicted values in the control algorithm. For this purpose, the inverse kinematics problem was solved by determining the angular velocities of the mecanum wheels based on the desired motion path of the selected point of the robot's frame.

Based on the analysis of the kinematics of the WMR with mecanum wheels in a moving coordinate system [7], it was possible to determine the robot's kinematics in a fixed XYZ system. The WMR diagram is shown in Figure 1.

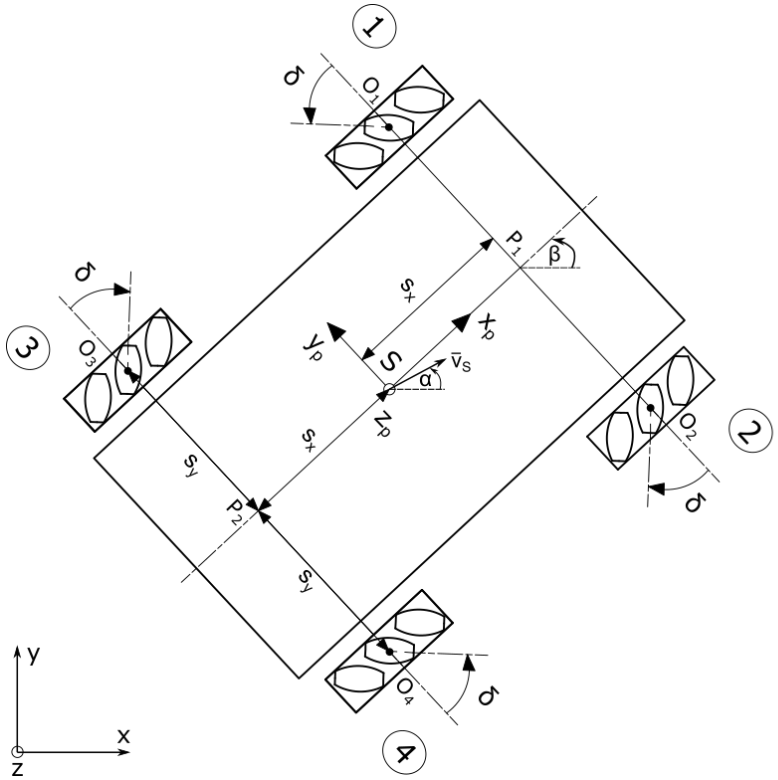

Figure 1. Diagram of the wheeled mobile robot.

In Figure 1, the following notation is introduced: β, robot frame rotation angle; α, angle between instantaneous velocity vector $\bar{v}_S$ and the x-axis; $2s_y$, robot width, $s_y = |O_3P_2| = |P_2O_4|$; s_x, distance between the characteristic point S (platform center of mass) and the front or rear robot axis, $s_x = |SP_1| = |P_2S|$; δ, angle between the roller axis and the mecanum wheel axis ($\delta = 45°$); $x_py_pz_p$, moving coordinate system associated with the frame of the mobile robot at point S; and xyz, stationary coordinate system.

A view of the WMR model in the x_pz_p plane of the moving coordinate system is presented in Figure 2.

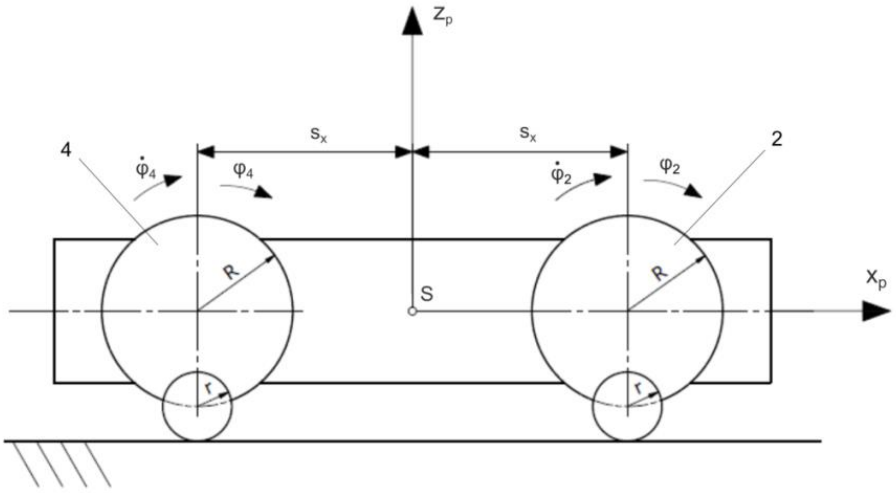

Figure 2. View of the WMR model in the x_pz_p plane.

The description of WMR kinematics in the stationary coordinate system was formulated in the form of equations [7]:

$$\begin{aligned} -\dot{x}_s(\cos\beta+\sin\beta)+\dot{y}_s(\cos\beta-\sin\beta)+\dot{y}(s_x+s_y)+\dot{\varphi}_1(R+r)=0\\ \dot{x}_s(\cos\beta-\sin\beta)+\dot{y}_s(\cos\beta+\sin\beta)+\dot{y}(s_x+s_y)-\dot{\varphi}_2(R+r)=0\\ \dot{x}_s(\cos\beta-\sin\beta)+\dot{y}_s(\cos\beta+\sin\beta)-\dot{y}(s_x+s_y)-\dot{\varphi}_3(R+r)=0\\ -\dot{x}_s(\cos\beta+\sin\beta)+\dot{y}_s(\cos\beta-\sin\beta)-\dot{y}(s_x+s_y)+\dot{\varphi}_4(R+r)=0 \end{aligned} \quad (1)$$

where $\dot{x}_s, \dot{y}_s$ are the projections of the velocity vector of point S on the x,y axes of the stationary coordinate system, $\dot{\beta}$ is the angular velocity of the robot frame, R is the wheel radius, r is the roller radius, and $\dot{\varphi}_1$, $\dot{\varphi}_2$, $\dot{\varphi}_3$, $\dot{\varphi}_4$ are the angular velocities of the mecanum wheels.

Using the transforming system of Equation (2), the solution of the inverse kinematics problem was determined using the following:

$$\begin{aligned} \dot{\varphi}_1=\tfrac{1}{(R+r)}\left[\dot{x}_s(\cos\beta+\sin\beta)+\dot{y}_s(-\cos\beta+\sin\beta)-\dot{\beta}(s_x+s_y)\right]\\ \dot{\varphi}_2=\tfrac{1}{(R+r)}\left[\dot{x}_s(\cos\beta-\sin\beta)+\dot{y}_s(\cos\beta+\sin\beta)+\dot{\beta}(s_x+s_y)\right]\\ \dot{\varphi}_3=\tfrac{1}{(R+r)}\left[\dot{x}_s(\cos\beta-\sin\beta)+\dot{y}_s(\cos\beta+\sin\beta)-\dot{\beta}(s_x+s_y)\right]\\ \dot{\varphi}_4=\tfrac{1}{(R+r)}\left[\dot{x}_s(\cos\beta+\sin\beta)+\dot{y}_s(-\cos\beta+\sin\beta)+\dot{\beta}(s_x+s_y)\right] \end{aligned} \quad (2)$$

The system of Equation (2) calculated the angular velocity waveforms of individual wheels on the basis of the desired motion path and the velocity profile of a selected point on the robot's frame.

A robot dynamics model was analyzed in [15]. Lagrange equations of the second kind were used to describe the WMR dynamics in the form

$$\frac{d}{dt}\left(\frac{\partial E}{\partial \dot{q}_j}\right)-\frac{\partial E}{\partial q_j}=Q_j \quad (3)$$

where j = 1, 2, . . . , s; s is the number of the object's degrees of freedom, q_j is the value of the jth generalized coordinate, $\dot{q}_j$ is the value of the jth generalized velocity, E is the kinetic energy of the solid system, and Q_j is the value of the jth generalized force.

The solution of the inverse dynamics problem was determined using the following:

$$M\ddot{q}+N(\dot{q})=\tau \quad (4)$$

where τ is the vector of moment of force, $\ddot{q}=\ddot{\varphi}$ is the vector of generalized angular accelerations, N is the vector of resistance to motion, and M is the inertia matrix.

The values of matrices and vectors were introduced to Equation (4), obtaining the solution of the inverse dynamics problem for the ith mecanum wheel:

$$M\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_2\\ \ddot{q}_3\\ \ddot{q}_4\end{bmatrix}+\begin{bmatrix}N_1f_1\mathrm{sgn}(\dot{q}_1)\\ N_2f_2\mathrm{sgn}(\dot{q}_2)\\ N_3f_3\mathrm{sgn}(\dot{q}_3)\\ N_4f_4\mathrm{sgn}(\dot{q}_4)\end{bmatrix}=\begin{bmatrix}\tau_1\\ \tau_2\\ \tau_3\\ \tau_4\end{bmatrix} \quad (5)$$

where τ_i is the moment of force acting on the ith wheel, $\ddot{q}_i=\ddot{\varphi}_i$ is the angular acceleration of the ith mecanum wheel, N_i is the pressure force of the ith wheel, and f_i is the coefficient of rolling friction of the ith wheel.

The inertia matrix M was determined using the following:

$$M = \begin{bmatrix} (A+B+C) & (-B) & B & (A-B) \\ (-B) & (A+B+C) & (A-B) & B \\ B & (A-B) & (A+B+C) & (-B) \\ (A-B) & B & (-B) & (A+B+C) \end{bmatrix} \quad (6)$$

where

$$\begin{aligned} A &= \frac{m_{pc}(R+r)^2}{8} \\ B &= \frac{I_{pc}(R+r)^2}{16(s_x+s_y)^2} \\ C &= I_k \\ m_{pc} &= m_p + 4m_k \\ I_{pc} &= I_{p_{zp}} + 4I_{k_{zp}} \end{aligned} \quad (7)$$

and m_p is the mass of the platform, m_k is the mass of the mecanum wheel, I_k is the mass moment of inertia of the wheel in relation to the axis of self-rotation, $I_{k_{zp}}$ is the mass moment of inertia of the mecanum wheel in relation to the axis z_p of the movable system $x_p y_p z_p$, and $I_{p_{zp}}$ is the mass moment of platform inertia in relation to the z_p axis.

In order to solve the simple task of dynamics, the relationship of (5) was transformed to the form

$$\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \\ \ddot{q}_4 \end{bmatrix} = M^{-1} \begin{bmatrix} \tau_1 - N_1 f_1 sgn(\dot{q}_1) \\ \tau_2 - N_2 f_2 sgn(\dot{q}_2) \\ \tau_3 - N_3 f_3 sgn(\dot{q}_3) \\ \tau_4 - N_4 f_4 sgn(\dot{q}_4) \end{bmatrix} \quad (8)$$

The matrix M^{-1} is written in the form

$$M^{-1} = \begin{bmatrix} D_1 & D_2 & D_3 & D_4 \\ D_2 & D_1 & D_4 & D_3 \\ D_3 & D_4 & D_1 & D_2 \\ D_4 & D_3 & D_2 & D_1 \end{bmatrix} \quad (9)$$

where auxiliary variables are described by the following equations:

$$\begin{aligned} D_1 &= \frac{2AB+AC+3BC+C^2}{(8AB+2AC+4BC+C^2)C} \\ D_2 &= \frac{B}{(4B+C)C} \\ D_3 &= \frac{(-B)}{(4B+C)C} \\ D_4 &= \frac{-(2AB+AC-BC)}{(8AB+2AC+4BC+C^2)C} \end{aligned} \quad (10)$$

On the basis of relationships in (5), it was possible to determine the driving moments τ_i of each of the mecanum wheels for a desired motion path while Equation (8) determined the generalized accelerations of each of the wheels for the assumed waveforms of driving moments.

On the basis of the mathematical model of the robot, a synthesis of the robot's motion control system was conducted, and the motion of the object was simulated.

3. Synthesis of the Neural Control Algorithm

NNs are used in neural control algorithms and act as universal approximators of functions. They enable the approximation of the nonlinearity of the control object and its compensation in the control signal. The output from the network is determined in real time, which enables NN weight adaptation in response to changes in the parameters of the robot model. An additional advantage of this method is that it does not require detailed

knowledge of the parameters of the mathematical model of the controlled object. The disadvantage of using NNs in the control algorithm is the large number of calculations that must be performed by the microprocessor robot control system in a given iteration step of the algorithm, which limits their application in real-time control algorithms. This problem has been especially visible in extensive network structures, such as those with a large number of neurons and many input signals.

Adaptive intelligent control was based on the model of the control system resulting from the combination of nonlinearity compensating control implemented with the use of NN and the PD controller. The scheme of the algorithm is shown in Figure 3.

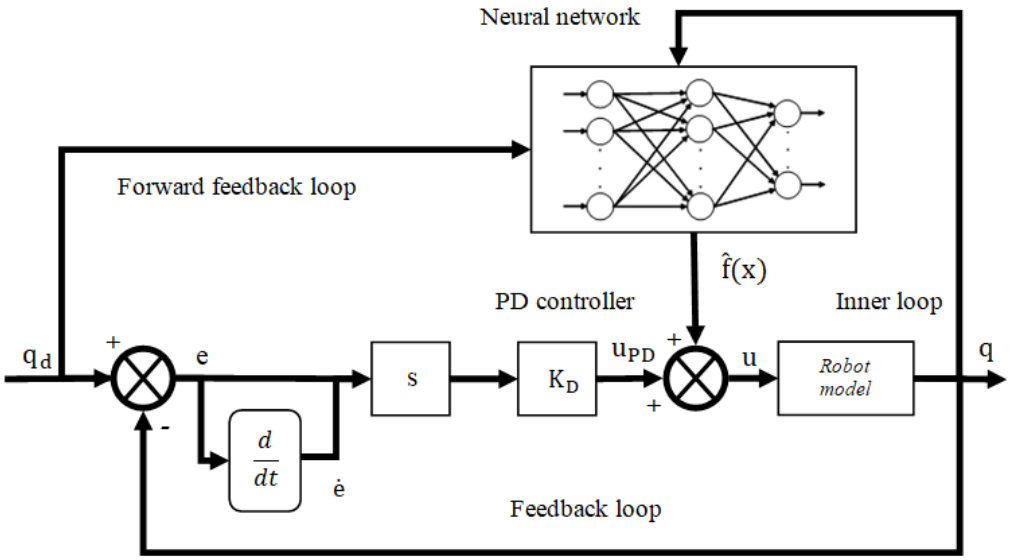

Figure 3. Diagram of the intelligent control algorithm.

The WMR nonlinearity compensation algorithm consisted of four RVFL NNs, one for each of the mecanum wheels. A bipolar sigmoidal neuron activation function was assumed in the form of the following equation:

$$\Phi(x) = S\left(V^T x\right) = \frac{2}{1 + e^{-\gamma V^T x}} - 1 \tag{11}$$

where γ is the inclination factor of the activation function, V is the constant weight matrix of the input layer selected randomly in the network initialization process, $x = [1, \dot{v}, \dot{q},]$ is the vector of network input values, $\dot{v} = \ddot{q}_d - \lambda \dot{e}$ is the auxiliary variable simplifying the notation, and a value of 1 in vector x represents the constant input signal used to determine the threshold value.

The output from the NN was described by the relationship

$$y(x) = W^T \Phi(x) \tag{12}$$

where W is the matrix of weights of the output layer with zero initial values.

The control signal of the PD controller had the following form:

$$u_{PD} = K_D s = K_D(\dot{e} + \Lambda e) \tag{13}$$

where K_D is a diagonal matrix of the gain coefficients of the derivative term, $K_P = K_D \Lambda$ is a diagonal matrix of the gain coefficients of the proportional term, and Λ is a diagonal matrix with positive elements $\lambda_{i,i}$.

The tracking error e and the generalized error s were defined as

$$e = q_d - q \tag{14}$$

$$s = \dot{e} + \Lambda e \tag{15}$$

where q_d is the vector of desired rotation angles of mecanum wheels, and q is the vector of implemented rotation angles.

Based on the above model, the law of NN weight adaptation was determined so that the analyzed system was stable.

The WMR dynamics Equation (4) was adopted in the following form:

$$M\ddot{q} + N(\dot{q}) + \eta_d(t) = u \tag{16}$$

where $\eta_d(t)$ is a vector of bounded, time-varying disturbance signals, and u is a vector of control signals.

By differentiating Equation (15) with respect to time and considering the dynamics of Equation (16), a description of the closed control system as a function of the generalized error was obtained as the following:

$$M\dot{s} = -u + f(x) + \eta_d(t) \tag{17}$$

where $f(x) = M\dot{v} + N(\dot{q})$ is the nonlinear function approximated by the NN.

The control law was adopted as the following form:

$$u = \hat{f}(x) + K_D s \tag{18}$$

where $\hat{f}(x)$ is an estimate of the function f(x).

Assuming an approximation of the nonlinearity of the object dynamics using NN, the following was obtained:

$$f(x) = W^T \Phi(x) + \varepsilon \tag{19}$$

where W is the constant matrix of ideal NN output weights and ε is the approximation error.

The estimate of the function f(x) was determined by the following:

$$\hat{f}(x) = \hat{W}^T \Phi(x) \tag{20}$$

where $\hat{W}$ is an estimate of the matrix of ideal weights.

Inserting Equation (20) into (18), we obtained

$$u = \hat{W}^T \Phi(x) + K_D s \tag{21}$$

The first part of the equation is implemented by an NN and approximates the function f(x), while the second part is the PD controller equation according to the notation (13). Inserting (21) into (17), the following relationship was obtained:

$$M\dot{s} = -K_D s + \widetilde{f}(x) + \eta_d(t) \tag{22}$$

where

$$\widetilde{f}(x) = f(x) - \hat{f}(x) = \left(W - \hat{W}\right)^T \Phi(x) + \varepsilon = \widetilde{W}^T \Phi(x) + \varepsilon \tag{23}$$

Using Lyapunov stability theory, the algorithm of weight adaptation $\hat{W}$ was determined. A positively defined candidate for the Lyapunov function was assumed in the following form:

$$V(t) = \frac{1}{2} s^T M s + \frac{1}{2} tr\left(\widetilde{W}^T F^{-1} \widetilde{W}\right) \tag{24}$$

where F^{-1} is a constant diagonal matrix with positive elements.

The derivative of V(t) over time is the following:

$$\dot{V} = s^T M \dot{s} + tr\left(\widetilde{W}^T F^{-1} \dot{W}\right) \tag{25}$$

Substituting (22) into (25) and simplifying the equation, we obtained

$$\dot{V} = -s^T K_D s + tr\left\{\widetilde{W}^T\left(F^{-1}\dot{W} + \Phi(x)s^T\right)\right\} + s^T(\varepsilon + \eta_d(t)) \quad (26)$$

The law of weight adaptation was adopted using the following:

$$\dot{\hat{W}} = F\Phi(x)s^T \quad (27)$$

Assuming that the ideal weights W = const., the derivative $\dot{W} = 0$. Inserting (27) into (26), we obtained

$$\dot{V} = -s^T K_D s + s^T(\varepsilon + \eta_d(t)) \quad (28)$$

Therefore, for $\dot{V} \leq 0$:

$$\|s\| \geq \frac{\varepsilon_{max} + \eta_{dmax}}{K_{Dmin}} \quad (29)$$

The above analysis showed that V was positively defined, and $\dot{V}$ was negatively defined for the satisfied condition (29). According to the Lyapunov theory, the signals s and $\widetilde{W}$ were limited, and the analyzed system was stable for the applied control (21) and the law of NN weight adaptation (27).

4. Numerical Tests of the Operation of the Algorithm

The results of the simulation of WMR motion controlled by the proposed algorithm are presented below. The use of a robot dynamics model enabled approximate mapping of the response of the dynamic system to a set excitation, which would verify the accuracy of the control system. In addition, numerical tests analyzed the quality of the dynamic object motion with the use of various control methods, using the adopted quality indicators.

The simulation tests were conducted in the MATLAB/Simulink package. The Runge-Kutta integration with a constant time discretization step equal to $t_d = 0.01$ [s] was used. The disturbances introduced in the simulation were implemented by changes in the value of the rolling friction coefficient from the initial value $f_n = 0.046$ [m] to the value $f_d = 0.055$ [m] during $6.5 \leq t \leq 11.2$ [s]. The coefficient was changed for all of the mecanum wheels.

In the simulation, the controlled object parameters were used to replicate the real parameters of Husarion's Panther mobile robot. The following geometric parameters of the robot were adopted: R = 0.085 [m], r = 0.0165 [m], s_x = 0.22 [m], and s_y = 0.351 [m]. Dynamic parameters of the robot were the following: A = 0.01404 $\left[kg{\cdot}m^2\right]$, B = 0.03767 $\left[kg{\cdot}m^2\right]$, and C = 0.01193 $\left[kg{\cdot}m^2\right]$.

Numerical tests of the WMR motion were conducted using a PD controller (Test 1) with the gain of the proportional term $K_{Pi,i} = 2$ and the derivative term $K_{Di,i} = 1$. In the intelligent control algorithm test (Test 2), the same PD controller gain values were used as in Test 1 while in the nonlinearity compensation term of the controlled object, four NNs were used, each of which consisted of h = 4 inputs (including the threshold value), m = 10 neurons in the hidden layer, and one output. The choice of m = 10 neurons in the hidden layer resulted from numerous simulation studies for a different number of neurons, in which a slight increase in the quality of tracking was observed with more than 10 neurons while the NN complexity increased proportionally to the number of neurons. A coefficient value of γ = 4 and matrix of learning gain coefficients F = diag(9) were adopted. The parameters of the PD and NN controllers were selected by trial and error so as to obtain the best quality of motion execution.

The waveform of the velocity vector of the characteristic point S of the robot was adopted in accordance with the following relationship:

$$v_S = v_{max}\left(\frac{1}{1+e^{-c(t-t_r)}} - \frac{1}{1+e^{-c(t-t_h)}}\right)[m/s] \tag{30}$$

where $v_{max} = 0.5$ [m/s] is the maximum velocity in steady state, c = 9 [1/s] is the rate of velocity change in the acceleration and deceleration phases, $t_r = 2$ [s] is the mean acceleration time, and $t_h = 13$ [s] is the mean braking time.

To assess the quality of motion implementation with the use of a given control algorithm, the following quality indicators were adopted:

(a) The maximum value of the rotation angle error of the mecanum wheel:

$$e_{imax} = \max|e_{ik}|[rad] \tag{31}$$

where i = 1,2,3,4 is the mecanum wheel number and k is the number of consecutive discrete measurements.

(b) The maximum value of the angular velocity error of the mecanum wheel:

$$\dot{e}_{imax} = \max\left|\dot{e}_{ik}\right|[rad/s] \tag{32}$$

(c) Root-mean-square error (RMSE) of tracking the desired rotation angle:

$$\varepsilon_i = \sqrt{\frac{1}{n}\sum_{k=1}^{n} e_{ik}^2}[rad] \tag{33}$$

where n is the number of discrete measurements.

(d) RMSE of tracking the desired angular velocity:

$$\dot{\varepsilon}_i = \sqrt{\frac{1}{n}\sum_{k=1}^{n} \dot{e}_{ik}^2}[rad/s] \tag{34}$$

(e) The maximum distance between the desired and implemented position of the characteristic point S of the robot on the xy plane:

$$d_{max} = \max d_k[m] \tag{35}$$

where $d_k = \sqrt{(x_{Sk} - x_{dSk})^2 + (y_{Sk} - y_{dSk})^2}$, x_S, y_S are the coordinates of point S in a stationary coordinate system xyz, and x_{dS}, y_{dS} are the desired coordinates of point S.

(f) RMSE of the distance between the desired and implemented position of the robot's characteristic point S:

$$\rho = \sqrt{\frac{1}{n}\sum_{k=1}^{n} d_k^2}[m] \tag{36}$$

(g) The distance between the desired and implemented point S position after the end of the motion:

$$d_n = \sqrt{(x_{Sn} - x_{dSn})^2 + (y_{Sn} - y_{dSn})^2}[m] \tag{37}$$

During the implementation of the desired motion path, the orientation of the robot's frame remained constant $\beta = 0$[rad]. Point S of the robot, in the first stage, moved along a straight-line path according to the direction of the instantaneous velocity vector $\overline{v}_S$, deviating from the x-axis by an angle $\alpha = \frac{\pi}{2}$[rad]. In the second stage, point S moved along a circular path, traversing $\frac{3}{4}$ of a full circle (vector $\overline{v}_S$ changed its direction by an angle $\frac{\pi}{2} \leq \alpha \leq 2\pi$[rad]). The movement along a curvilinear path of the robot began at $t_{op} \approx 4$ [s] and ended when $t_{ok} \approx 11$ [s] (the mean time of the start and the end of vector $\overline{v}_S$

direction change). The shape of the angular velocity profile $\dot{\alpha}$, defining the change in the motion direction of the point S, was analogous to the shape of the velocity profile of point S (Equation (30)). In the last stage, the robot moved along a straight-line path again, at angle $\alpha = 2\pi[\text{rad}]$, as shown in Figure 4.

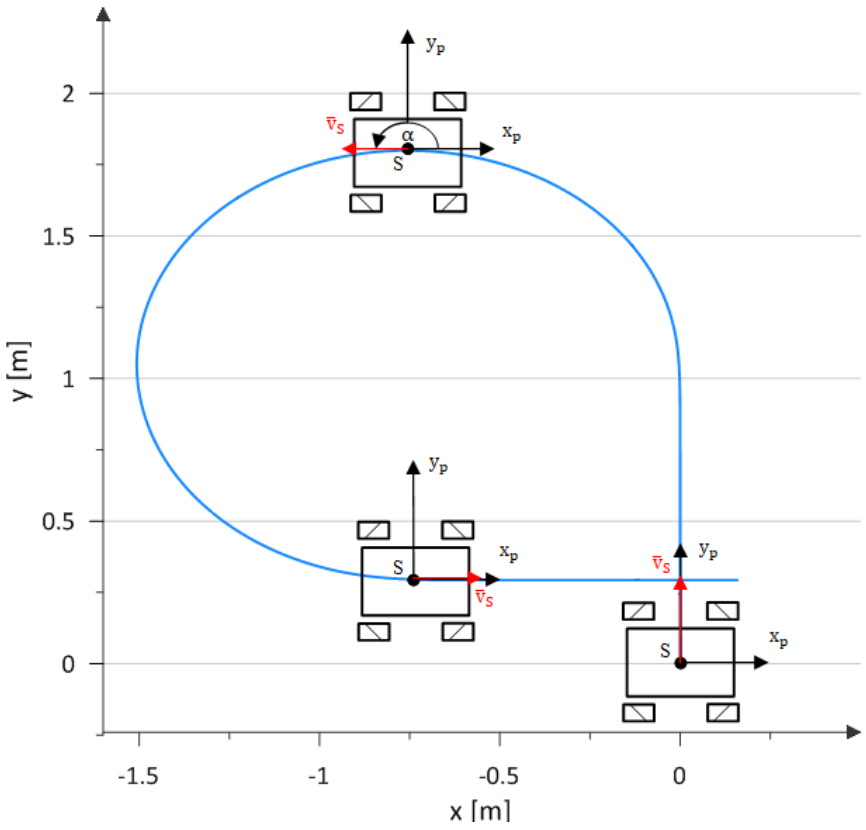

Figure 4. The desired path of point S of the mobile robot's frame.

4.1. Simulation Test 1 (PD Controller)

Figure 5 presents the results of the simulation showing the waveforms of the implemented angles of rotation of individual mecanum wheels (Figure 5a), implemented angular velocities (Figure 5b), tracking errors of the rotation angles, and angular velocities (Figure 5c,d, respectively). The control signals generated by the PD controller are shown in Figure 5e. The diagram of the desired and implemented motion path is shown in Figure 5f. All of the graphs use the color marking of the mecanum wheel number: red, wheel 1; yellow, wheel 2; blue, wheel 3; and green, wheel 4.

The values of angular velocity vectors of the wheels (Figure 5b) were the same for the pairs of wheels, 1, 4 and 2, 3, due to the constant angle of the robot's frame orientation. The waveforms of the control signals (Figure 5e) reflected this phenomenon by having the same values for the aforementioned wheels pairs. The influence of the disturbance on the waveform of the control signal occurring at $t_{z1} = 6.5$ [s] and ending at $t_{z2} = 11.2$ [s] was also noticeable (the time of the occurrence of the disturbance in the images is marked with a dashed line).

The angular velocity error (Figure 5d) was limited, and after stabilization, it tended toward zero. The rotation angle error (Figure 5c) in steady states had a non-zero value. The error value decreased and tended toward zero as a result of the start of the braking process and the end of the motion. The value of the error of mapping the desired motion path (Figure 5f) was small.

The obtained values of quality indicators were used to compare the quality of the control using a neural control algorithm.

The quality indicators obtained as a result of the simulation are presented in Table 1.

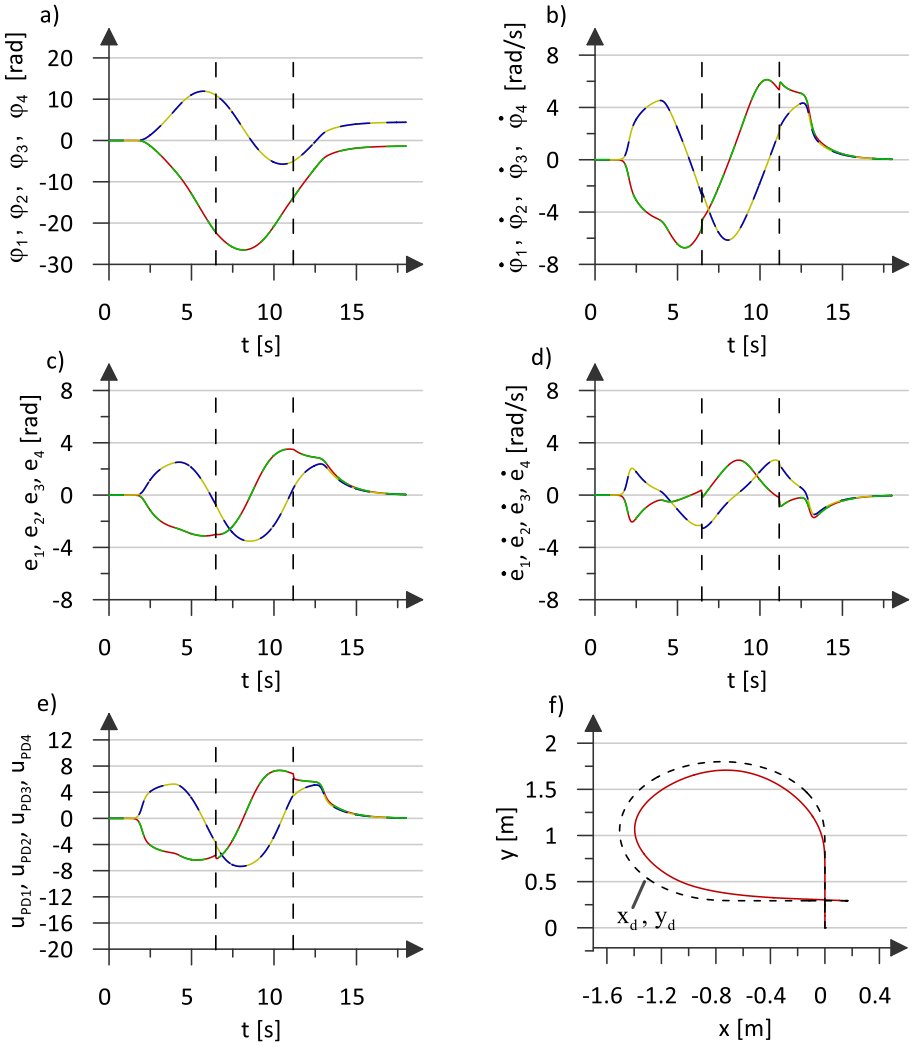

Figure 5. Results of simulation 1: (**a**) waveforms of implemented rotation angles $\varphi_1, \varphi_2, \varphi_3, \varphi_4$, (**b**) angular velocities $\dot{\varphi}_1$, $\dot{\varphi}_2$, $\dot{\varphi}_3$, $\dot{\varphi}_4$, (**c**) tracking errors of the rotation angle e_1, e_2, e_3, e_4, (**d**) tracking errors of the angular velocity $\dot{e}_1$, $\dot{e}_2$, $\dot{e}_3$, $\dot{e}_4$, (**e**) control signals of the PD controller u_{PD1}, u_{PD2}, u_{PD3}, u_{PD4}, (**f**) graph of the desired (x_d, y_d) and implemented path.

Table 1. Quality indicators for PD control simulation.

Indicator	e_{imax} [rad]	$\dot{e}_{imax}$ [rad/s]	ε_i [rad]	$\dot{\varepsilon}_i$ [rad/s]	d_{max} [m]	d_n [m]	ρ
wheel 1, i = 1	3.5175	2.6729	2.0919	1.0540			
wheel 2, i = 2	3.5244	2.6855	1.7741	1.2389			
wheel 3, i = 3	3.5244	2.6855	1.7741	1.2389	0.2740	0.0043	0.1969
wheel 4, i = 4	3.5175	2.6729	2.0919	1.0540			
mean value	3.5209	2.6792	1.9330	1.1465			

4.2. Simulation Test 2 (Intelligent Control)

The same motion path was used from Test 1. The arrangement of the graphs showing the waveforms of individual motion parameters, errors, and control signals in Figure 6 corresponded to that in Figure 5. Additionally, Figure 6 shows the waveforms of control signals generated by NN (Figure 6f) and waveforms of total control signals (Figure 6g). The waveforms of the NN output weights are presented for the first wheel in Figure 6h (for the remaining wheels, the waveforms of the weights had a similar shape). The diagram of the

desired and implemented motion path is shown in Figure 6i. For all the graphs (except for Figure 6h,i), the color markings of the mecanum wheel numbers were used: red, wheel 1; yellow, wheel 2; blue, wheel 3; and green, wheel 4.

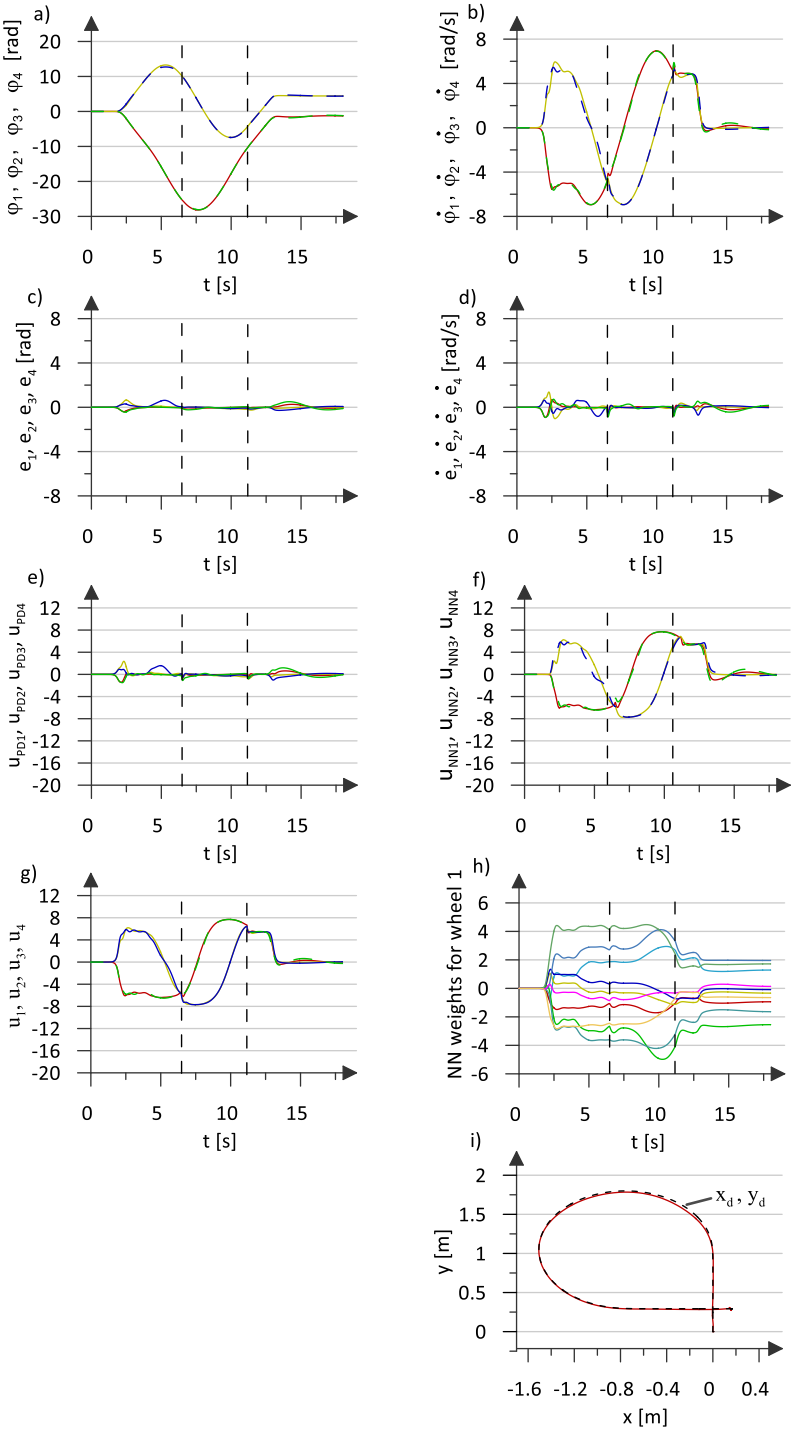

Figure 6. Results of simulation 2: (**a**) waveforms of implemented rotation angles $\varphi_1, \varphi_2, \varphi_3, \varphi_4$, (**b**) angular velocities $\dot{\varphi}_1$, $\dot{\varphi}_2$, $\dot{\varphi}_3$, $\dot{\varphi}_4$, (**c**) tracking errors of the rotation angle e_1, e_2, e_3, e_4, (**d**) tracking errors of the angular velocity $\dot{e}_1$, $\dot{e}_2$, $\dot{e}_3$, $\dot{e}_4$, (**e**) control signals of the PD controller u_{PD1}, u_{PD2}, u_{PD3}, u_{PD4}, (**f**) control signals from the NN u_{NN1}, u_{NN2}, u_{NN3}, u_{NN4}, (**g**) overall control signals u_1, u_2, u_3, u_4, (**h**) NN output weights for wheel 1, (**i**) graph of the desired (x_d, y_d) and implemented path.

The quality indicators obtained as a result of the simulation are presented in Table 2.

Table 2. Quality indicators for neural control simulation.

Indicator	e_{imax} [rad]	$\dot{e}_{imax}$ [rad/s]	ε_i [rad]	$\dot{\varepsilon}_i$ [rad/s]	d_{max} [m]	d_n [m]	ρ
wheel 1, i = 1	0.3933	0.9179	0.1038	0.1828			
wheel 2, i = 2	0.6514	1.3743	0.1359	0.2611			
wheel 3, i = 3	0.6264	0.8208	0.1805	0.2529	0.0442	0.0175	0.0162
wheel 4, i = 4	0.5039	0.9180	0.1825	0.2579			
mean value	0.5438	1.0078	0.1507	0.2387			

The implemented motion parameters had similar waveforms to those obtained in Test 1 (Figure 6a,b). The tracking errors were limited and had the greatest values at the beginning and ending phases of motion, when the angular velocity vector of the wheel changed to the opposite value and, to a lesser extent, at the time of the disturbance. The error was minimized to the vicinity of zero for both the angular velocity and the angle of rotation error (Figure 6c,d).

The control signal PD assumed a non-zero value during the beginning of the acceleration and deceleration phases, during changes in the direction of the rotation of the wheels pairs, and as a result of the disturbance acting on the object (Figure 6e). Neural control (Figure 6f), after partially determining the output weights of the network, intervened as the controller. The values of the NN weights were relatively stable after the initial stabilization (shown as the first wheel in Figure 6h). As found in Test 1, the error of mapping the desired motion path (Figure 6i) was small.

All quality indicators, with the exception of d_n, had lower values for the control algorithm when the NNs were used, as compared to a PD controller. This difference was apparent, especially for the RMSE of tracking the desired rotation angle ε_i, where the index values decreased by approximately 90% when using intelligent control, as compared to the PD controller itself. Additionally, the results of the neural control were obtained under the most unfavorable conditions, during which the NN output weights had zero initial values. The intelligent algorithm, due to the process of NN weight adaptation, provided more accurate mapping of the motion path, as compared to a classic PD controller.

5. Verification Tests of the Control Algorithm

The verification tests of the control algorithm were conducted on the laboratory stand, as shown in Figure 7, consisting of a computer with a dSpace DS1103 control, a measurement card, and a Husarion Panther WMR with mecanum wheels. The dimensions of the robot were 805 × 840 × 290 mm (length × width × height), its weight was 55 kg, and the maximum load capacity was 80 kg. The robot control system was built on the basis of an internal Raspberry Pi 4B control computer with a Broadcom BCM2711 processor (Broadcom, San Jose, CA, USA) and an additional Intel NUC10i7FNK computer (Intel, Santa Clara, CA, USA) for the implementation of computationally demanding tasks, such as mapping the environment, generating collision-free trajectories in an unknown environment, and the implementation of computationally demanding artificial intelligence algorithms. The robot was also equipped with a Slamtec RPLIDAR S1 laser 2D scanner (Slamtec, Shanghai, China). The robot used four 80 PMB800K.80RBL BLDC motors (rated power 473 W each) with planetary gears and incremental encoders. The rated torque of each of the drive modules was 34.5 Nm. The power source was a package of lithium-ion batteries with a voltage of 36 V and a capacity of 740 Wh. The maximum speed of the robot was 2 m/s. For the purposes of scientific research with the use of the dSpace control and the measurement card, the design of the robot had been modified by the manufacturer to provide measurement signals for the robot's motion parameters and direct control of the actuator systems, bypassing the internal controller.

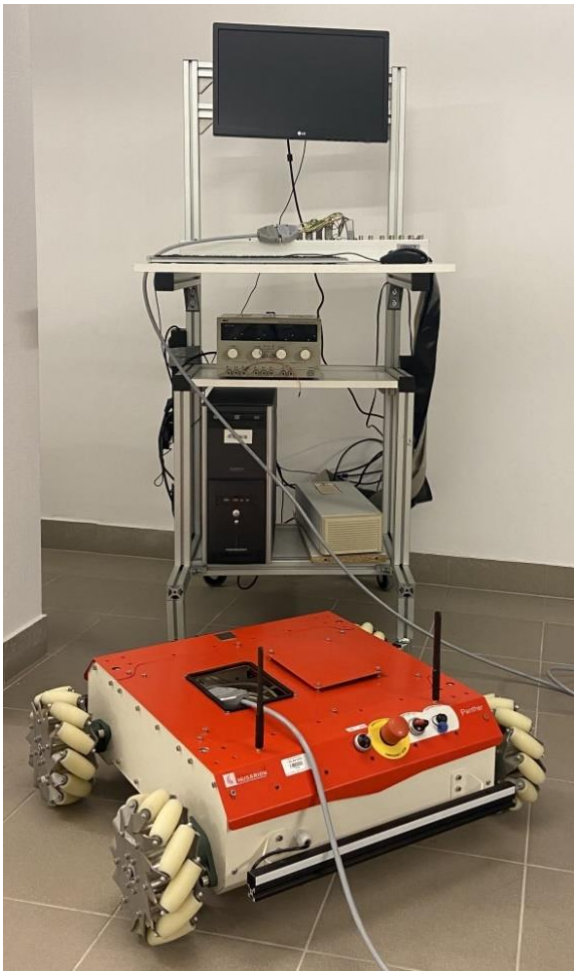

Figure 7. The laboratory stand.

The methods of rapid prototyping of control systems with the use of dSpace control and measurement cards provided quick and easy testing on a real object. During the verification tests, the dSpace DS1103 card was used to obtain the measurement data from the encoders and to generate motor control signals. The control algorithm was programmed in the MATLAB/Simulink environment, compiled to the level of optimized C code and then to the machine code and implemented in real time by the dSpace DS1103 card. The scheme of the laboratory stand is shown in Figure 8.

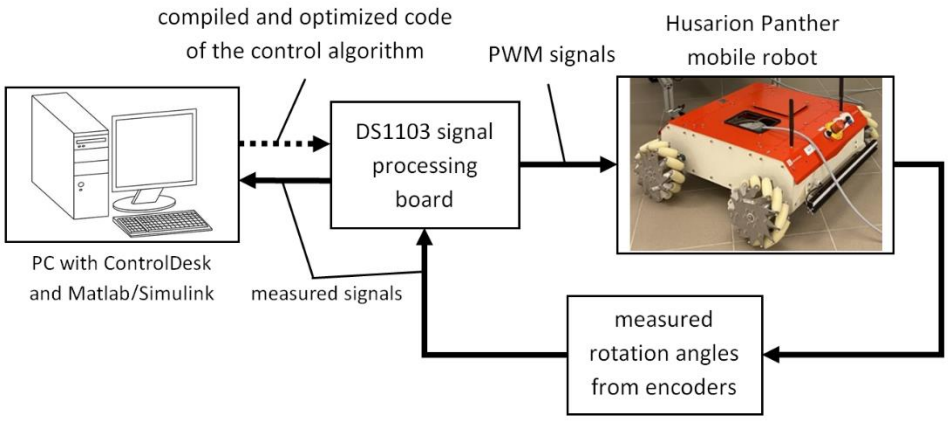

Figure 8. Diagram of the laboratory stand.

The same motion trajectory was used in the verification experiments as in the numerical tests, and the same settings of the control algorithms were also adopted. The parametric disturbance of the robot's motion was introduced by changing the texture of the test track surface (at t ∈ < 6.5, 11.2 > [s]), generating greater rolling resistance in the robot's wheels. The same set of quality indicators (31–37) as in the simulation studies were used to assess the quality of tracking.

5.1. Verification Test 1 (PD Controller)

Figure 9 presents the results of the verification test, including the waveforms of the implemented angles of rotation of individual mecanum wheels (Figure 9a), the implemented angular velocities (Figure 9b), the tracking errors of the rotation angles, and the angular velocities (Figure 9c,d, respectively). The control signals generated by the PD controller are shown in Figure 9e. The diagram of the desired and implemented motion path is shown in Figure 9f.

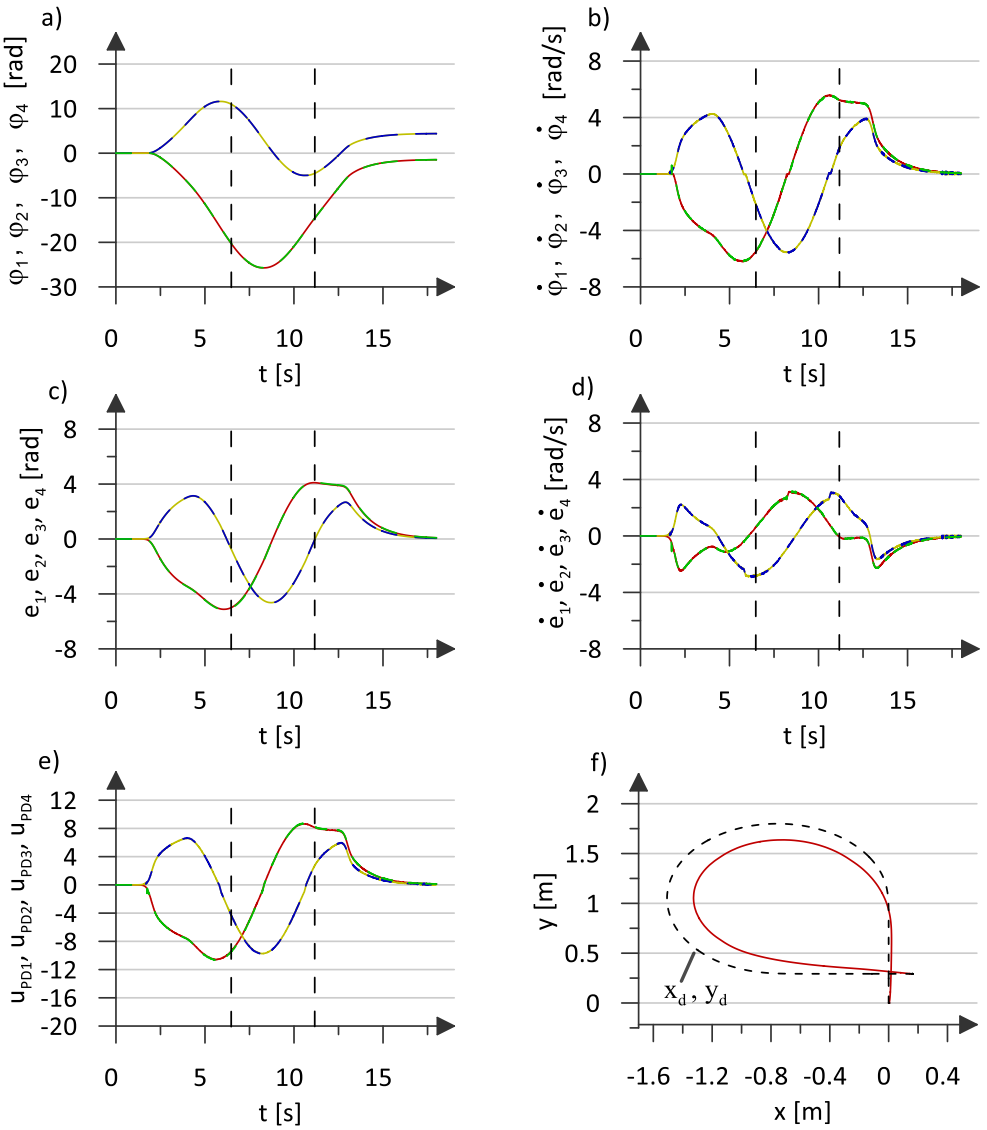

Figure 9. Results of verification 1: (**a**) waveforms of implemented rotation angles $\varphi_1, \varphi_2, \varphi_3, \varphi_4$, (**b**) angular velocities $\dot{\varphi}_1, \dot{\varphi}_2, \dot{\varphi}_3, \dot{\varphi}_4$, (**c**) tracking errors of the rotation angle e_1, e_2, e_3, e_4, (**d**) tracking errors of the angular velocity $\dot{e}_1, \dot{e}_2, \dot{e}_3, \dot{e}_4$, (**e**) control signals of the PD controller u_{PD1}, u_{PD2}, u_{PD3}, u_{PD4}, (**f**) graph of the desired (x_d, y_d) and implemented path.

The individual waveforms were similar to the results obtained in the simulation tests. Due to the occurrence of unmodeled disturbances, the tracking errors in the verification tests had higher values than in the numerical test, and the realized path differed to a greater extent from the desired one.

The quality indicators obtained as a result of the verification are presented in Table 3.

Table 3. Quality indicators for PD control verification.

Indicator	e_{imax} [rad]	$\dot{e}_{imax}$ [rad/s]	ε_i [rad]	$\dot{\varepsilon}_i$ [rad/s]	d_{max} [m]	d_n [m]	ρ
wheel 1, i = 1	5.1201	3.1432	2.8616	1.3841			
wheel 2, i = 2	4.6355	3.0982	2.1873	1.4994			
wheel 3, i = 3	4.6579	3.0943	2.1949	1.4994	0.3751	0.0124	0.2606
wheel 4, i = 4	5.1606	3.2050	2.8842	1.3841			
mean value	4.8935	3.1352	2.5320	1.4380			

5.2. Verification Test 2 (Intelligent Control)

The arrangement of the graphs showing the waveforms of individual motion parameters, errors, and control signals in Figure 10 corresponds to that in Figure 9. Additionally, Figure 10 shows the waveforms of the control signals generated by the NN (Figure 10f) and the waveforms of the total control signals (Figure 10g). The waveforms of the NN output weights are presented for the first wheel in Figure 10h. The diagram of the desired and implemented motion path is shown in Figure 10i.

The waveforms of the realized motion parameters and control signals presented in Figure 10 had a shape similar to the waveforms obtained in the simulation tests. The tracking error values were slightly higher, which was confirmed by the values of the determined quality indicators. It may have been a result of the occurrence of unmodeled phenomena and disturbances during motion. The tracking errors were the highest in the initial phase of the movement, and then they were minimized to a range close to zero, which resulted from the presence of an adaptive element in the control system in the form of an NN. Such a phenomenon did not occur when only the PD controller had been used in the control system, as in Verification Test 1. The values of the NN weights from zero initial values changed their values during the adaptation process and stabilized at certain values. The comparison of the desired and realized trajectory of the selected WMR point indicated the high accuracy of the tracking and mapping of the desired trajectory.

The quality indicators obtained as a result of the verification are presented in Table 4.

Table 4. Quality indicators for neural control verification.

Indicator	e_{imax} [rad]	$\dot{e}_{imax}$ [rad/s]	ε_i [rad]	$\dot{\varepsilon}_i$ [rad/s]	d_{max} [m]	d_n [m]	ρ
wheel 1, i = 1	0.7718	4.1174	0.1371	0.3763			
wheel 2, i = 2	0.7149	3.8033	0.2099	0.4677			
wheel 3, i = 3	0.9735	4.8162	0.1834	0.4891	0.0846	0.0319	0.0278
wheel 4, i = 4	1.1244	4.0510	0.2383	0.4718			
mean value	0.8961	4.1970	0.1922	0.4512			

The values of the quality indicators obtained during verification with the use of the neural control algorithm indicated significantly improved accuracy of the tracking, as compared to the use of the PD controller alone.

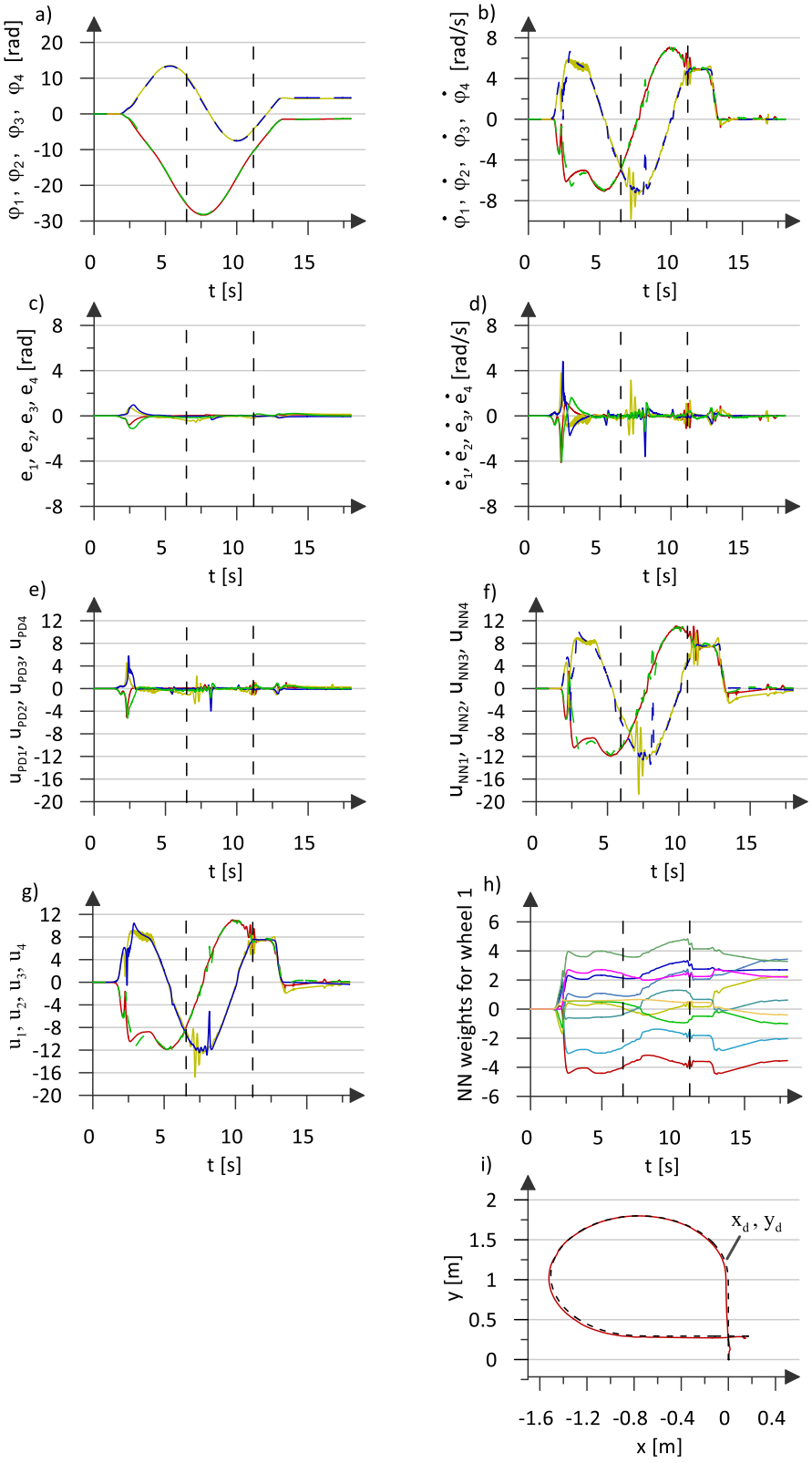

Figure 10. Results of verification 2: (**a**) waveforms of implemented rotation angles $\varphi_1, \varphi_2, \varphi_3, \varphi_4$, (**b**) angular velocities $\dot{\varphi}_1$, $\dot{\varphi}_2$, $\dot{\varphi}_3$, $\dot{\varphi}_4$, (**c**) tracking errors of the rotation angle e_1, e_2, e_3, e_4, (**d**) tracking errors of the angular velocity $\dot{e}_1$, $\dot{e}_2$, $\dot{e}_3$, $\dot{e}_4$, (**e**) control signals of the PD controller u_{PD1}, u_{PD2}, u_{PD3}, u_{PD4}, (**f**) control signals from the NN u_{NN1}, u_{NN2}, u_{NN3}, u_{NN4}, (**g**) overall control signals u_1, u_2, u_3, u_4, (**h**) NN output weights for wheel 1, (**i**) graph of the desired (x_d, y_d) and implemented path.

6. Conclusions

To test the accuracy of a WMR motion implementation with the use of an intelligent (neural) control algorithm, simulation tests were conducted in a MATLAB/Simulink environment, and verification tests were conducted using the presented laboratory stand. The results of a neural control algorithm and an algorithm with the structure of a classic PD controller were compared.

Based on the collected data, presented in the form of graphs and designated quality indicators, an analysis was conducted of the control strategies on the desired robot motion. The tests confirmed that both control methods provided stability of motion executed by the object due to limiting tracking errors. In addition, the neural control algorithm, as compared to PD control, provided better accuracy for the desired motion path under uncertainty conditions, which were related to the model parameters (disturbances). This conclusion was drawn based on the values of the quality indicators.

A synthesis of an intelligent neural control algorithm for the motion of a mobile robot with mecanum wheels was conducted as well as a qualitative analysis of the control method used. Based on the results, we concluded that neural control increased the quality of motion execution.

Author Contributions: Conceptualization, M.S. (Marcin Szuster); Formal analysis, M.S. (Marcin Szuster); Investigation, M.S. (Mateusz Szeremeta); Project administration, M.S. (Marcin Szuster); Software, M.S. (Mateusz Szeremeta); Supervision, M.S. (Marcin Szuster); Visualization, M.S. (Mateusz Szeremeta); Writing—original draft, M.S. (Mateusz Szeremeta); Writing—review & editing, M.S. (Marcin Szuster). All authors contributed equally to this article and accepted the final report. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: All authors declare no conflict of interest.

References

1. Abd Mutalib, M.A.; Azlan, N.Z. Prototype development of mecanum wheels mobile robot: A review. *Appl. Res. Smart Technol.* **2020**, *1*, 71–82. [CrossRef]
2. Velagic, J.; Osmic, N.; Lacevic, B. Neural network controller for mobile robot motion control. *World Acad. Sci. Eng. Technol.* **2008**, *47*, 193–198.
3. Watanabe, K.; Tang, J.; Nakamura, M.; Koga, S.; Fukuda, T. A fuzzy-Gaussian neural network and its application to mobile robot control. *IEEE Trans. Control Syst. Technol.* **1996**, *4*, 193–199. [CrossRef]
4. Vo, A.T.; Kang, H.J. An adaptive neural non-singular fast-terminal sliding-mode control for industrial robotic manipulators. *Appl. Sci.* **2018**, *8*, 2562. [CrossRef]
5. Bae, H.; Kim, G.; Kim, J.; Qian, D.; Lee, S. Multi-robot path planning method using reinforcement learning. *Appl. Sci.* **2019**, *9*, 3057. [CrossRef]
6. Szuster, M. Dual-Heuristic Dynamic Programming in the Three-Wheeled Mobile Transport Robot Control. In Proceedings of the International Conference on Artificial Intelligence and Soft Computing, Zakopane, Poland, 3–7 June 2018; pp. 763–776.
7. Hendzel, Z.; Rykała, Ł. Description of kinematics of a wheeled mobile robot with mecanum wheels. *Model. Eng.* **2015**, *26*, 5–12.
8. Hendzel, Z. A Description of the Motion of a Mobile Robot with Mecanum Wheels–Kinematics. In Proceedings of the Conference on Automation, Warsaw, Poland, 27–29 March 2019; pp. 346–355.
9. Li, Y.; Dai, S.; Zheng, Y.; Tian, F.; Yan, X. Modeling and kinematics simulation of a Mecanum wheel platform in RecurDyn. *J. Robot.* **2018**, *2018*, 1–7. [CrossRef]
10. Taheri, H.; Qiao, B.; Ghaeminezhad, N. Kinematic model of a four mecanum wheeled mobile robot. *Int. J. Comput. Appl.* **2015**, *113*, 6–9. [CrossRef]
11. Li, Y.; Ge, S.; Dai, S.; Zhao, L.; Yan, X.; Zheng, Y.; Shi, Y. Kinematic Modeling of a Combined System of Multiple Mecanum-Wheeled Robots with Velocity Compensation. *Sensors* **2020**, *20*, 75. [CrossRef]
12. Röhrig, C.; Heß, D.; Künemund, F. Motion Controller Design for a Mecanum Wheeled Mobile Manipulator. In Proceedings of the 2017 IEEE Conference on Control Technology and Applications (CCTA), Maui, HI, USA, 27–30 August 2017; pp. 444–449.

13. Maulana, E.; Muslim, M.; Hendrayawan, A.V. Inverse Kinematic Implementation of Four-WHEELS mecanum Drive Mobile Robot Using Stepper Motors. In Proceedings of the 2015 International Seminar on Intelligent Technology and Its Applications (ISITIA), Surabaya, Indonesia, 20–21 May 2015; pp. 51–56.
14. Jia, Q.; Wang, M.; Liu, S.; Ge, J.; Gu, C. Research and Development of Mecanum-Wheeled Omnidirectional Mobile Robot Implemented by Multiple Control Methods. In Proceedings of the 2016 23rd International Conference on Mechatronics and Machine Vision in Practice (M2VIP), Nanjing, China, 28–30 November 2016; pp. 1–4.
15. Hendzel, Z.; Rykała, Ł. Modelling of Dynamics of a Wheeled Mobile Robot with Mecanum Wheels with the use of Lagrange Equations of the Second Kind. *Int. J. Appl. Mech. Eng.* **2017**, *22*, 81–99. [CrossRef]
16. Zeidis, I.; Zimmermann, K. Dynamics of a four wheeled mobile robot with Mecanum wheels. *J. Appl. Math. Mech. Z. Angew. Math. Mech.* **2019**, *99*, e201900173. [CrossRef]
17. Becker, F.; Bondarev, O.; Zeidis, I.; Zimmermann, K.; Abdelrahman, M.; Adamov, B. An approach to the kinematics and dynamics of a four-wheel Mecanum vehicle. *Sci. J. IFToMM Probl. Mech.* **2014**, *2*, 27–37.
18. Adamov, B.I.; Saypulaev, G.R. A Study of the Dynamics of an Omnidirectional Platform, Taking into Account the Design of Mecanum Wheels and Multicomponent Contact Friction. In Proceedings of the 2020 International Conference Nonlinearity, Information and Robotics (NIR), Innopolis, Russia, 3–6 December 2020; pp. 1–6.
19. Vlantis, P.; Bechlioulis, C.P.; Karras, G.; Fourlas, G.; Kyriakopoulos, K.J. Fault Tolerant Control for Omni-Directional Mobile Platforms with 4 Mecanum Wheels. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 2395–2400.
20. Hendzel, Z. A Description of the Motion of a Mobile Robot with Mecanum Wheels–Dynamics. In Proceedings of the Conference on Automation, Warsaw, Poland, 27–29 March 2019; pp. 337–345.
21. Tătar, M.O.; Popovici, C.; Mândru, D.; Ardelean, I.; Pleşa, A. Design and Development of an Autonomous Omni-Directional Mobile Robot with Mecanum Wheels. In Proceedings of the 2014 IEEE International Conference on Automation, Quality and Testing, Robotics, Cluj-Napoca, Romania, 22 May 2014; pp. 1–6.
22. Ma'arif, A.; Vera, M.A.M.; Mahmoud, M.S.; Ladaci, S.; Çakan, A.; Parada, J.N. Backstepping Sliding Mode Control for Inverted Pendulum System with Disturbance and Parameter Uncertainty. *J. Robot. Control* **2022**, *3*, 86–92. [CrossRef]
23. Reguii, I.; Hassani, I.; Rekik, C. Mobile Robot Navigation Using Planning Algorithm and Sliding Mode Control in a Cluttered Environment. *J. Robot. Control* **2022**, *3*, 166–175. [CrossRef]
24. Zhang, H.; Li, B.; Xiao, B.; Yang, Y.; Ling, J. Nonsingular recursive-structure sliding mode control for high-order nonlinear systems and an application in a wheeled mobile robot. *ISA Trans.* 2022; *in press*. [CrossRef]
25. Xin, L.; Wang, Q.; She, J.; Li, Y. Robust adaptive tracking control of wheeled mobile robot. *Robot. Auton. Syst.* **2016**, *78*, 36–48. [CrossRef]
26. Huynh, V.N.S.; Ngo, H.Q.T.; Nguyen, T.P.; Nguyen, H. High Performance of an Adaptive Sliding Mode Controller under Varying Loads for Lifting-Type Autonomous Grounded Robot. *Appl. Sci.* **2020**, *10*, 5858. [CrossRef]
27. Jabeur, C.B.; Seddik, H. Optimized Neural networks-PID Controller with Wind Rejection Strategy for a Quad-rotor. *J. Robot. Control* **2022**, *3*, 62–72. [CrossRef]
28. Ginting, A.H.; Doo, S.Y.; Pollo, D.E.; Djahi, H.J.; Mauboy, E.R. Attitude Control of a Quadrotor with Fuzzy Logic Controller on SO (3). *J. Robot. Control* **2022**, *3*, 101–106. [CrossRef]
29. Muni, M.K.; Parhi, D.R.; Kumar, P.B.; Sahu, C.; Kumar, S. Towards motion planning of humanoids using a fuzzy embedded neural network approach. *Appl. Soft Comput.* **2022**, *119*, 108588. [CrossRef]
30. Štefek, A.; Pham, V.T.; Krivanek, V.; Pham, K.L. Optimization of Fuzzy Logic Controller Used for a Differential Drive Wheeled Mobile Robot. *Appl. Sci.* **2021**, *11*, 6023. [CrossRef]
31. Wu, X.; Huang, Y. Adaptive fractional-order non-singular terminal sliding mode control based on fuzzy wavelet neural networks for omnidirectional mobile robot manipulator. *ISA Trans.* **2022**, *121*, 258–267. [CrossRef] [PubMed]
32. Wu, Q.; Wang, X.; Chen, B.; Wu, H. Development of an RBFN-based neural-fuzzy adaptive control strategy for an upper limb rehabilitation exoskeleton. *Mechatronics* **2018**, *53*, 85–94. [CrossRef]
33. Fang, H.; Zhu, Y.; Dian, S.; Xiang, G.; Guo, R.; Li, S. Robust tracking control for magnetic wheeled mobile robots using adaptive dynamic programming. *ISA Trans.* 2021; *in press*. [CrossRef] [PubMed]
34. Muszynska, M.; Burghardt, A.; Kurc, K.; Szybicki, D. Verification hybrid control of a wheeled mobile robot and manipulator. *Open Eng.* **2016**, *6*, 64–72. [CrossRef]
35. Szuster, M.; Gierlak, P. Approximate dynamic programming in tracking control of a robotic manipulator. *Int. J. Adv. Robot. Syst.* **2016**, *13*, 16. [CrossRef]
36. Szuster, M.; Hendzel, Z. *Intelligent Optimal Adaptive Control for Mechatronic Systems*; Springer: Berlin, Germany, 2018.
37. Penar, P.; Hendzel, Z. Biologically Inspired Neural Behavioral Control of the Wheeled Mobile Robot. In Proceedings of the Conference on Automation, Warsaw, Poland, 18–19 March 2021; Springer: Cham, Switzerland, 2021; pp. 96–105.
38. Sun, Z.; Hu, S.; He, D.; Zhu, W.; Xie, H.; Zheng, J. Trajectory-tracking control of Mecanum-wheeled omnidirectional mobile robots using adaptive integral terminal sliding mode. *Comput. Electr. Eng.* **2021**, *96*, 107500. [CrossRef]
39. Sun, Z.; Xie, H.; Zheng, J.; Man, Z.; He, D. Path-following control of Mecanum-wheels omnidirectional mobile robots using nonsingular terminal sliding mode. *Mech. Syst. Signal Process.* **2021**, *147*, 107128. [CrossRef]

40. Yadav, P.S.; Agrawal, V.; Mohanta, J.C.; Ahmed, M.F. A robust sliding mode control of mecanum wheel-chair for trajectory tracking. *Mater. Today Proc.* **2022**, *56*, 623–630. [CrossRef]
41. Pizá, R.; Carbonell, R.; Casanova, V.; Cuenca, Á.; Salt Llobregat, J.J. Nonuniform Dual-Rate Extended Kalman-Filter-Based Sensor Fusion for Path-Following Control of a Holonomic Mobile Robot with Four Mecanum Wheels. *Appl. Sci.* **2022**, *12*, 3560. [CrossRef]
42. Alakshendra, V.; Chiddarwar, S.S. Adaptive robust control of Mecanum-wheeled mobile robot with uncertainties. *Nonlinear Dyn.* **2017**, *87*, 2147–2169. [CrossRef]
43. Hendzel, Z.; Kołodziej, M. Robust Tracking Control of Omni-Mecanum Wheeled Robot. In Proceedings of the Conference on Automation, Warsaw, Poland, 18–19 March 2021; pp. 219–229.
44. Zhao, T.; Zou, X.; Dian, S. Fixed-time observer-based adaptive fuzzy tracking control for Mecanum-wheel mobile robots with guaranteed transient performance. *Nonlinear Dyn.* **2022**, *107*, 921–937. [CrossRef]
45. Qin, Y.; Dian, S.; Guo, B.; Xiang, G.; Fang, H.; Wang, H.; Zhang, X. Trajectory Tracking Control for Mecanum-Wheel Cambered Mobile Robots Based on Online Adaptive Critic Optimal Controller. In Proceedings of the 2021 40th Chinese Control Conference (CCC), Shanghai, China, 26–28 July 2021; pp. 314–319.
46. Jiang, M.; Chen, L.; Wang, Y.; Wu, H. Adaptive Backstepping Control for Mecanum-Wheeled Omnidirectional Vehicle Using Neural Networks. *IEEJ Trans. Electr. Electron. Eng.* **2022**, *17*, 378–386. [CrossRef]
47. Wang, D.; Wei, W.; Yeboah, Y.; Li, Y.; Gao, Y. A robust model predictive control strategy for trajectory tracking of omni-directional mobile robots. *J. Intell. Robot. Syst.* **2020**, *98*, 439–453. [CrossRef]
48. Malayjerdi, E.; Kalani, H.; Malayjerdi, M. Self-Tuning Fuzzy Pid Control of a Four-Mecanum Wheel Omni-Directional Mobile Platform. In Proceedings of the 29th Iranian Conference on Electrical Engineering (ICEE), Mashhad, Iran, 8–10 May 2021; pp. 816–820.
49. Cao, G.; Zhao, X.; Ye, C.; Yu, S.; Li, B.; Jiang, C. Fuzzy adaptive PID control method for multi-mecanum-wheeled mobile robot. *J. Mech. Sci. Technol.* **2022**, *36*, 2019–2029. [CrossRef]
50. Zijie, N.; Qiang, L.; Yonjie, C.; Zhijun, S. Fuzzy control strategy for course correction of omnidirectional mobile robot. *Int. J. Control Autom. Syst.* **2019**, *17*, 2354–2364. [CrossRef]
51. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst.* **1989**, *2*, 303–314. [CrossRef]
52. Csáji, B.C. Approximation with Artificial Neural Networks. Master's Thesis, Eotvos Loránd University, Budapest, Hungary, 2001.
53. Tsai, C.C.; Wu, H.L. Nonsingular Terminal Sliding Control Using Fuzzy Wavelet Networks for Mecanum Wheeled Omni-Directional Vehicles. In Proceedings of the International Conference on Fuzzy Systems, Barcelona, Spain, 18–23 July 2010; pp. 1–6.
54. Tsai, C.C.; Wu, H.L.; Lee, Y.R. Intelligent adaptive motion controller design for mecanum wheeled omnidirectional robots with parameter variations. *Int. J. Nonlinear Sci. Numer. Simul.* **2010**, *11*, 91–96. [CrossRef]
55. Lu, X.; Zhang, X.; Zhang, G.; Fan, J.; Jia, S. Neural network adaptive sliding mode control for omnidirectional vehicle with uncertainties. *ISA Trans.* **2019**, *86*, 201–214. [CrossRef]
56. Kuo, C.H. Trajectory and Heading Tracking of a Mecanum Wheeled Robot Using Fuzzy LOGIC control. In Proceedings of the 2016 International Conference on Instrumentation, Control and Automation (ICA), Bandung, Indonesia, 29–31 August 2016; pp. 54–59.
57. Kumile, C.M.; Tlale, N.S. Intelligent Distributed Fuzzy Logic Control System (IDFLCS) of a Mecanum Wheeled Autonomous Guided Vehicle. In Proceedings of the IEEE International Conference Mechatronics and Automation, Niagara Falls, ON, Canada, 29 July–1 August 2005; Volume 1, pp. 131–137.
58. Jamali, P.; Tabatabaei, S.M.; Sohrabi, O.; Seifipour, N. Software Based Modeling, Simulation and Fuzzy Control of a Mecanum Wheeled Mobile Robot. In Proceedings of the 2013 First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM), Tehran, Iran, 13–15 February 2013; pp. 200–204.
59. Hendzel, Z.; Szuster, M.; Gierlak, P. *Sieci Neuronowe i Systemy Rozmyte*; Oficyna Wydawnicza Politechniki Rzeszowskiej: Rzeszów, Poland, 2010.
60. Gao, W.; Selmic, R.R. Neural network control of a class of nonlinear systems with actuator saturation. *IEEE Trans. Neural Netw.* **2006**, *17*, 147–156. [CrossRef]
61. Zhou, Z.; Wu, B. Adaptive sliding mode control of manipulators based on fuzzy random vector function links for friction compensation. *Optik* **2021**, *227*, 166055. [CrossRef]
62. Dian, S.; Fang, H.; Zhao, T.; Wu, Q.; Hu, Y.; Guo, R.; Li, S. Modeling and trajectory tracking control for magnetic wheeled mobile robots based on improved dual-heuristic dynamic programming. *IEEE Trans. Ind. Inform.* **2020**, *17*, 1470–1482. [CrossRef]