Pomiary Automatyka Robotyka, ISSN 1427-9126, R. 28, Nr 4/2024, 47–54, DOI: 10.14313/PAR_254/47

Hardware Implementation of Sliding Mode Control of Mobile Robot with Four Mecanum Wheels

Damian Nagajek
Centrum Badań Kosmicznych Polskiej Akademii Nauk (CBK PAN), Branch Zielona Góra, Space Robot Dynamics Laboratory

Abstract: This paper proposes the design of a control system for a mobile robot capable of calculating a real-time control algorithm. The design of the mobile robot is presented with a description of the components from which it was made both in terms of hardware and software. The vision system used to analyze its motion is described. A simulation and an experiment were performed and their results are included in the paper. The paper is crowned with a comparison of the obtained results and plans for further development of the system.

Keywords: mobile robot, unstructured disturbance forces, robust finite-time task space control, control system

1. Introduction

The number of robots used in industries such as factories and warehouses is growing every year. They are most often used as automated guided vehicles or mobile platforms used, for example, to transport parts in narrow warehouses where mobility is a key element. Therefore, the use of mecanum wheels in such robots is often used because of their improved mobility. A robot equipped with them can maintain a preset orientation throughout the task (even on a curved path) and move laterally without changing the angle of its position. The four mecanum wheeled mobile robot (FMWMR) presented here is a special class of multidirectional mobile platforms, belonging to the class of holonomic dynamic systems. In order to effectively control such a robot, it is necessary to create a control system that can process data on the robot's movement, obstacles and calculate the control signals.

This article focuses on the design of the control system for this case of a mobile robot. Following the predefined trajectory requires high precision and stability of the control system. This task is very complicated due to the connection of four separate actuators and the control of each of them. In addition, the entire system must be correlated with a vision system to accurately receive information about terrain and obstacles allowing mobile robot to follow the predefined trajectory. In addition, the control algorithm should be fast, fault tolerant, optimal and require as little data as possible (e.g., full knowledge of the dynamics equations, etc.). Due to the complexity of this problem, only a few FMWMR control algorithms can meet the expectations of modern controllers.

As the most commonly used algorithms that allow trajectory tracking can be mentioned [1, 5, 7, 8, 11]. Robust algorithms [1, 5, 8] and adaptive algorithms [7, 11]. A PID (proportional-integral-derivative) controller [5] with fuzzy technique is used to adjust the gains of the PID controller, but this solution does not present a stability analysis. With full knowledge of FMWMR dynamics, it is possible to use robust adaptive control [1]. The robust algorithm [8] requires full knowledge of the dynamics equations. Algorithm [11] is complicated and suboptimal in terms of time savings. In addition, this control can be disturbed by external factors. In this paper, terminal sliding mode (TSM) is used to control FMWMR. Usually, TSM is associated with chattering effect, which is undesirable. In this case, to reduce the influence of this factor (or eliminate it completely), robust sliding mode controllers based on the transposition of the Jacobian matrix were used. With full knowledge of the Jacobian matrix, the control scheme is stable in finite time. In addition, it is robust to unlimited disturbances despite uncertain dynamic equations. This control algorithm uses the boundary layer technique to eliminate the undesirable effect of chattering. Control system is built on single-board computer (SBC) connected to motors.

The paper consists of six sections. Section 2 describes the theory of FMWMR kinematics and dynamics, problem formulation and basic assumptions. The design of the control system is presented in Section 3. Section 4 presents the course of the experiment. Discussion is presented in Section 5. The last Section 6 contains conclusions and opportunities for expanding of the overall control system.

Autor korespondujący:
Damian Nagajek, D.Nagajek@cbk.waw.pl

Artykuł recenzowany
nadesłany 09.04.2024 r., przyjęty do druku 24.09.2024 r.

2. Problem formulation

The problem considered in this article is to build the hardware part of a control system from commercially availa-

ble components, such as the SBC, for the FMWMR robot, which is capable of computing the control algorithm in real time. Commercially available components are much cheaper than dedicated computing chips (e.g., FPGAs). Such a solution allows large financial savings allowing further expansion of the hardware part of the system. In addition, the task, which involves following predefined trajectory, requires a vision system to track the robot's predefined trajectory and correct it using a control algorithm to minimize control error. The robot's task is to follow the predefined trajectory shown in Fig. 1 [2].

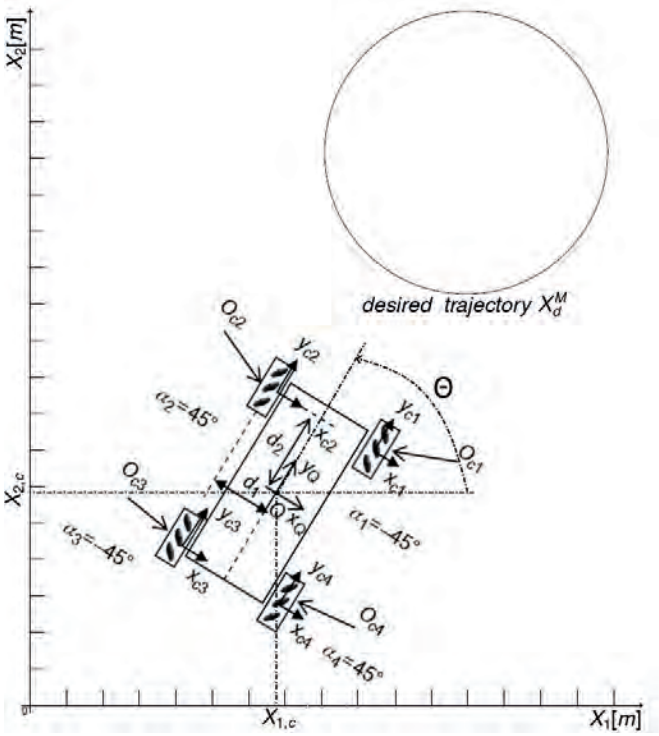

Fig. 1. A kinematic scheme of the mobile robot and the task to be accomplished [10]
Rys. 1. Schemat kinematyczny robota mobilnego i zadania do wykonania [10]

2.1. Kinematics and dynamics of a mobile robot

Kinematics of the robot with four mecanum wheels with 45° rollers is equal [4]:

$$x_M = f_M(q) \tag{1}$$

where:
x_M – vector of task coordinates equal $(x_{1,c},\ x_{2,c},\ \theta)^{\mathrm{T}}$;
$x_{1,c}$, $x_{2,c}$ – location of the center of mobile robot in global coordinate system;
θ – orientation of the mobile robot;
f_M – $\mathbb{R}^4 \rightarrow \mathbb{R}^3$;
q – vector of angular displacements of the wheels qual to $q = (q_1,\ q_2,\ q_3,\ q_4)^{\mathrm{T}}$;

$$f_M(q) = \left(f_{M,1}(q), f_{M,2}(q), f_{M,3}(q), f_{M,4}(q)\right) = \int_{q(0)}^{q} \mathcal{J}(q')\,dq';$$

$$dq' = \left(dq'_1, dq'_2, dq'_3, dq'_4\right) = \int_{q(0)}^{q} \mathcal{J}(q')\,dq'.$$

Jacobian matrix $\left(\mathcal{J} \in \mathbb{R}^{3\times 4}\right)$ of FMWMR is equal [4]:

$$\mathcal{J}(q') = \begin{bmatrix} \mathcal{J}_1(q') \\ \mathcal{J}_2(q') \\ \mathcal{J}_3(q') \end{bmatrix} = R\left(\theta = \theta(0) + \frac{r}{W+L}\left(-q'_1 + q'_2 + q'_3 - q'_4\right)\right) \frac{r}{4} \begin{bmatrix} -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ -\frac{1}{W+L} & \frac{1}{W+L} & \frac{1}{W+L} & -\frac{1}{W+L} \end{bmatrix} \tag{2}$$

where:

$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$ – velocity vector of mobile robot;

W – mobile platform width;
L – mobile platform length.

Dynamics of the whole system represented in generalized coordinates q can be described by equation [4]:

$$M(q)\left(\ddot{q}\right) + W\left(t, q, \dot{q}\right) = v \tag{3}$$

$$W\left(t, q, \dot{q}\right) = F\left(q, \dot{q}\right) + D\left(t, q, \dot{q}\right) \tag{4}$$

where:
M – 4 × 4 inertia matrix;
F – 4-dimensional vector representation of centrifugal and Coriolis forces;
D – 4-dimensional external disturbance signal;
v – 4-dimensional vector of control (torques/forces).

Using all parameters of the mobile robot (Section 3, Tab. 1) we can formulate the inertia matrix [4]:

$$M(q) = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \\ M_{41} & M_{42} & M_{43} & M_{44} \end{bmatrix}$$

where:

$$M_{11} = \frac{1}{8} m_p r^2 + \frac{1}{16\left(W+L\right)^2} I r^2 + I_w + m_w r^2;$$

$$M_{12} = -\frac{1}{16\left(W+L\right)^2} Ir^2;$$

$$M_{13} = \frac{1}{8} m_p r^2 - \frac{1}{16\left(W+L\right)^2} Ir^2;$$

$$M_{14} = \frac{1}{16\left(W+L\right)^2} Ir^2;$$

$$M_{21} = M_{12};\ M_{22} = M_{11};\ M_{23} = -M_{12};\ M_{24} = M_{13};$$

$$M_{31} = M_{13};\ M_{32} = M_{23};\ M_{33} = -M_{11};\ M_{34} = M_{12};$$

$$M_{41} = M_{14};\ M_{42} = M_{24};\ M_{43} = -M_{34};\ M_{44} = M_{11}.$$

Vector $F = \frac{1}{8} m_p r^2 \dot{\theta}\left(\dot{q}_2 + \dot{q}_4, -\dot{q}_1 - \dot{q}_3, \dot{q}_2 + \dot{q}_4, -\dot{q}_1 - \dot{q}_3\right)^{\mathrm{T}}$.
Initial configuration $q(0)$ and velocity $\dot{q}\left(0\right)$ are equal to $q\left(0\right) = \dot{q}\left(0\right) = \left[0 \quad 0 \quad 0 \quad 0\right]^{\mathrm{T}}$.

The task of the mobile robot is to track a desired trajectory $x_d^m\left(t\right) \in \mathbb{R}^2$, where: $t \in \left(0, \infty\right)$. In order to solve it, a sliding technique approach seem to be suitable. Control signal is composed of three variables t, x, s. Variable t denotes time. Variable x denotes position. Variable s is a sliding vector variable equal $s = (s_1, s_2, s_3, s_4)^{\mathrm{T}}$ and its defined in task coordinates as [4]:

$$s = \dot{e} + \int_0^t \left(\lambda_0 e^{\alpha_1} + \lambda_1 \left(\dot{e}\right)^{\alpha_2} \right) d\tau \tag{5}$$

where:

e – task tracking error equal $e = f_M\left(q\right) - x_d^M\left(t\right)$;

$\alpha_1 = \frac{a}{b}$, a and b are positive odd numbers ($a < b < 2a$);

$\alpha_2 = \frac{2\alpha_1}{a + \alpha_1}$;

λ_0 and λ_1 – controller gains.

To solve kinematic task control law of FMWMR we used equation below [4]:

$$u\left(t,s\right) = \begin{cases} -\frac{c}{a}\frac{s}{\left\|s\right\|}\left(\mathcal{U} + c'\right) & \text{for } s \neq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where:

u – control signal;
c and c' – controller gains;
a – variable that depending on mobile robot construction;
s – sliding vector variable;
$\mathcal{U}$ – construction parameter of the FMWMR.

Finite stability of the control law was performed. As can be seen from the above formula, the control is discontinuous. In order to avoid the effect of chattering, an extension of the above formula to the form was used [4]:

$$u\left(t,s\right) = \begin{cases} -\frac{c}{a}\frac{s}{\left\|s\right\|}\left(\mathcal{U} + c'\right) & \text{for } \left\|s\right\| \geq \varepsilon \\ -\frac{c}{a}\frac{s}{\varepsilon}\left(\mathcal{U} + c'\right) & \text{otherwise,} \end{cases} \tag{7}$$

where: ε – size of boundary layer (small positive real number defined by user).

3. Control system design

The entire structure is built from commercially available components (Fig. 2). Base of the platform is made of aluminum profiles 30 cm × 30 cm equipped with four mecanum wheels. Each wheel is connected to a motor with an encoder. Mounted on top are computers with overlays responsible for controlling the motors and processing the control algorithm. Power is provided by two 12 V batteries mounted in housings. This solution provides equal weight distribution. Housings for bateries and control boards were designed and printed on 3D printer. In that case a key element of the mobile platform is the use of mecanum wheels.

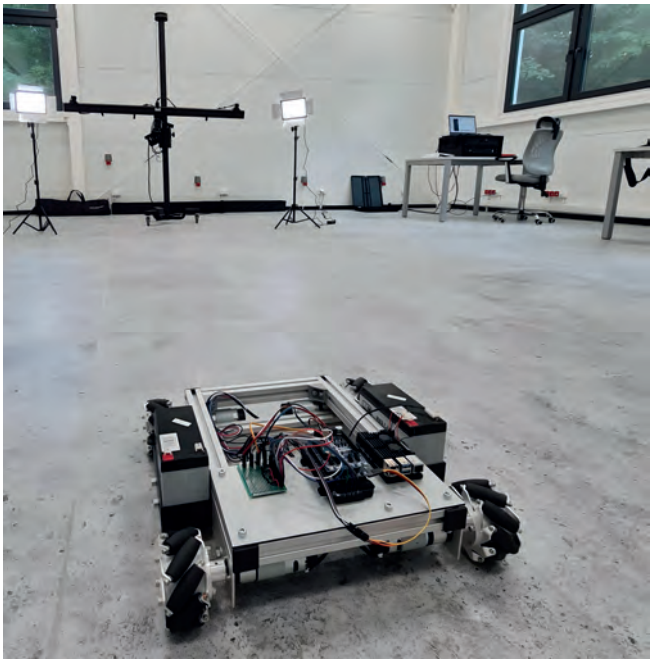

Fig. 2. Mobile robot with ARAMIS system
Rys. 2. Robot mobilny z systemem ARAMIS

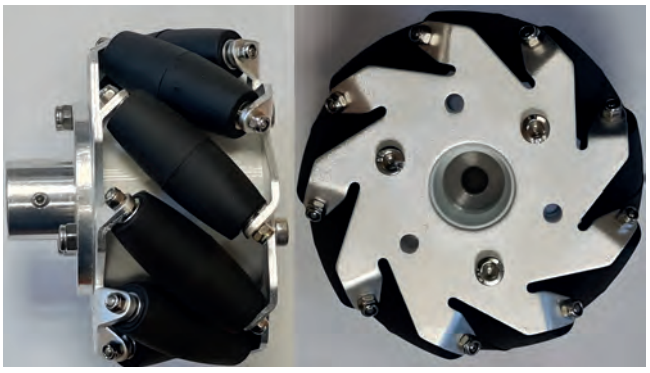

Fig. 3. Mecanum wheel
Rys. 3. Koło typu Mecanum

Mecanum wheels are a special type of omnidirectional wheels invented by Bengt Erland in 1973 [3]. They are wheels equipped with movable rollers that are located on the circumference of the wheel. The difference between them and Swedish wheels is the ratio of the angle of the rollers to the wheel. In Swedish wheels rollers are mounted at angle of 90°. In mecanum wheels (Fig. 3), the rollers are mounted at an angle of 45°. This configuration allows the platform to move in any direction, such as forward; backward; left; right; diagonal; arc and around the center point of one axis, maintaining a preset angle throughout the movement (Fig. 4) [9]. Parameters of the mobile robot system are shown in Tab 1.

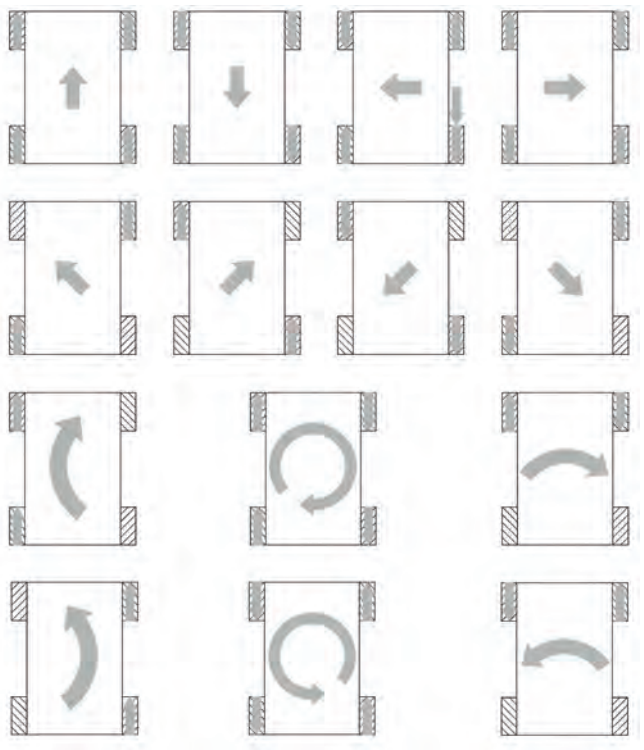

Fig. 4. All possible motions
Rys. 4.Wszystkie możliwe ruch

Tab. 1. Parameters of the system
Tab. 1. Parametry systemu

Parameter	Value	Unit	Symbol
Length of mobile platform	0.4	m	$2L$
Width of the mobile platform	0.3	m	$2W$
Mass of mobile platform	10.375	kg	m_p
Moment of interia of the platform	0.21614	kg × m^2	I
Moment of interia of the wheel	0.09375	kg × m^2	I_w
Mass of the wheel	0.375	kg	m_w
Radius of the wheel	0.05	m	r

3.1. Hardware

The prototype hardware mobile robot is built with two main components specifically Raspberry Pi and Arduino. Each of them is responsible for different tasks. The entire system is controlled by a Raspberry Pi 4B as the main computing unit. This single board computer (SBC) has enough computing power to execute the control algorithm. The SBC is connected via USB to an Arduino Mega 2560. The Mega version, unlike the Uno version, provides more connection ports, which were required to connect Cytron Motorshield MDD10 boards. This is an overlay (shield) that extends the capabilities of the Arduino. This particular one allows you to control two DC motors, specifically its speed and direction, using a Pulse With Modulation (PWM) signal, which is converted to the voltage supplied to the motors. Due to the use of four motors, two Cytron boards were used. Each was responsible for controlling the motors on one side of the mobile platform. The design used DC CHP-36GP motors equipped with encoders that allow reading direction and speed. All data from the encoders are sent to the Arduino MEGA 2560 board. The whole system provides good data transfer with frequency 10–30 Hz between its components. An external ARAMIS system was used as the vision system. It is used to track the mobile robot position during the experiment and is described in detail in Section 3.3.

3.2. Software

The used algorithm is computed on a hardware part of the control system, which consists of a Raspberry Pi as the central processing unit (CPU) used to compute the control algorithm. The CPU is connected to an Arduino Mega 2560, which is equipped with two Cytron motor controllers and is responsible for communication between them and the CPU, which is directly connected to motors equipped with encoders. The solution is divided into two paths. In the first path, Raspberry Pi calculates the control signals and sends data to the motors via Arduino. In the second path, the encoders collect data (rotations, orientation angles, etc.) and send it back to the CPU. This solution provides a good data flow. The entire process is completed with data from the ARAMIS vision system, which sends data to the Raspberry to analyze the current position and possibly correct the position and orientation of the mobile robot.

3.3. ARAMIS system

A mobile robot requires information about its location and orientation. This information can be obtained by using a vision system. The ARAMIS Adjustable base 12M Professional system from GOM (ger. *Gesellschaft für Optische Meßtechnik*) was used to take measurements during the experiment. It is an optical system for calculating and analyzing the actual coordinates, displacement, acceleration and velocity of the measured object. The measurement is made by tracking the displacement of special measurement points (markers). The resolution of the cameras is 12 Mpx and their spacing is 1800 mm. The measurement area ranges from 20 mm × 15 mm to 5000 mm × 4000 mm. Four markers were used to track the trajectory of the platform, which were placed at its angles. This arrangement allows tracking both the deviation from the predefined trajectory and the potential rotation of the platform. It also makes it possible to calculate the geometric center of the platform. Before the experiment, the system was calibrated using a calibration cross. The cameras were then positioned at an appropriate distance so that the markers could be tracked along the entire trajectory of the platform's movement. In addition, two spotlights were used during the experiment to provide adequate illumination. Two lenses with linear polarizing filters were used, with a focal length of 50 mm to enable marker detection [6].

4. Simulation and experimental results

The purpose of the study was to test the control system in simulation and through experiment and then compare these results.

4.1. Simulation

The simulated trajectory of the mobile robot deviated slightly from the predefined trajectory. The desired trajectory equation is equal (t – given time sample):

$$x_d^M = \left(2 + \cos\left(0.5t\right), 3 + \sin\left(0.5t\right), 0\right)^{\mathrm{T}} \tag{8}$$

MATLAB was used for simulation purposes. In its environment, the equations of dynamics and kinematics of the mobile robot were implemented based on the actual parameters of the platform (showed in Section 3). The linear velocity along the X

and Y axes and the angular velocity of the mecanum wheels were calculated. The calculations also took into account the centrifugal and Coriolis force vector. In addition, the entire control algorithm was checked for external disturbances of different strengths for different compartments: $D = (0,\ 0,\ 0,\ 0)^{\mathrm{T}}$ for $t \in \left[0,20\right]$, $D = (1.5,\ 0,\ 0,\ 0)^{\mathrm{T}}$ for $t \in \left[20,26\right]$, $D = (0,\ 0,\ 0,\ 0)^{\mathrm{T}}$ for $t \in \left[26,40\right]$, $D = (0,\ 0.5\ \sin(2t),\ 0,\ 0)^{\mathrm{T}}$ for $t \in \left[40,46\right]$, $D = (0,\ 0.5\ \sin(2t),\ 0.5\ \sin(2t),\ 0)^{\mathrm{T}}$ for $t \geq 60$.

4.2. Simulation results

The simulated trajectory of the mobile robot (Fig. 5) deviated slightly from the set trajectory. Logarithm of the control errors reached a value of less than 10^{-4} m in the peak, which occurred during the largest disturbances in the time interval of 20–26 s. At the same simulation time, the torques of individual motors in the peaks did not exceed the value of 2 Nm. In other time intervals with smaller disturbances, or their complete absence, the values were even smaller. The simulation results are shown in Figs. 6–10.

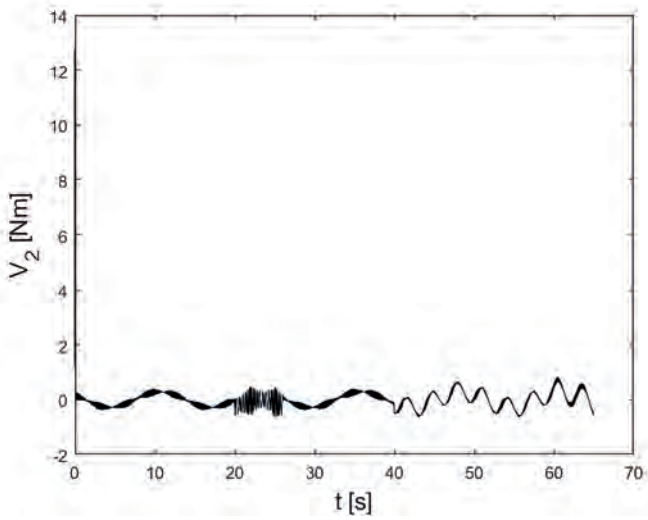

Fig. 7. Torque v_2 on motor m_2 versus time (simulation results)
Rys. 7. Moment obrotowy v_2 na silnik m_2 w funkcji czasu (wynik symulacji)

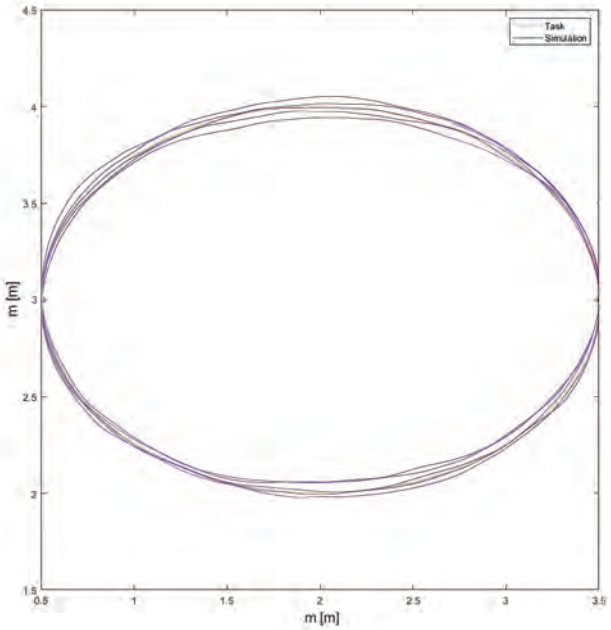

Fig. 5. Task trajectory (yellow line) and simulation results (red line)
Rys. 5. Trajektoria zadania (żółta linia) i wyniki symulacji (czerwona linia)

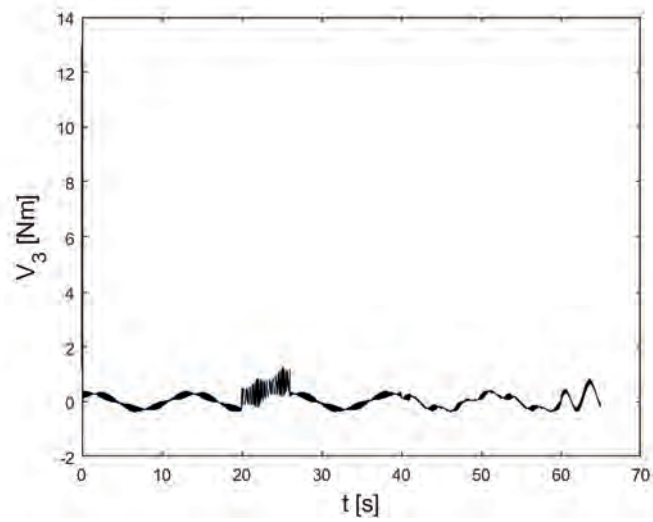

Fig. 8. Torque v_3 on motor m_3 versus time (simulation results)
Rys. 8. Moment obrotowy v_3 na silnik m_3 w funkcji czasu (wynik symulacji)

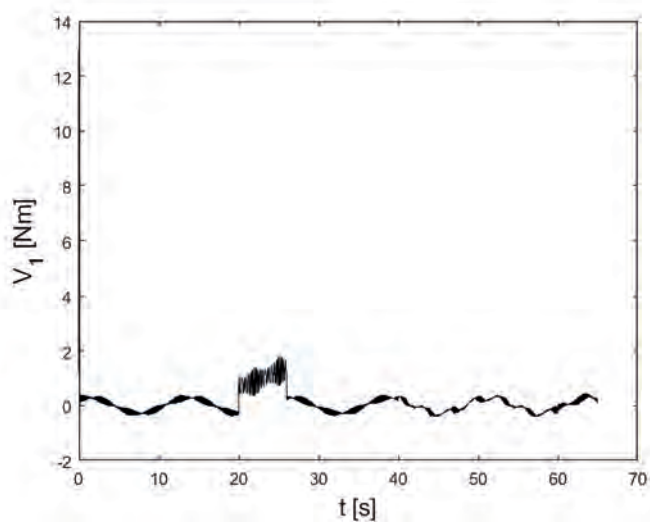

Fig. 6. Torque v_1 on motor m_1 versus time (simulation results)
Rys. 6. Moment obrotowy v_1 na silnik m_1 w funkcji czasu (wynik symulacji)

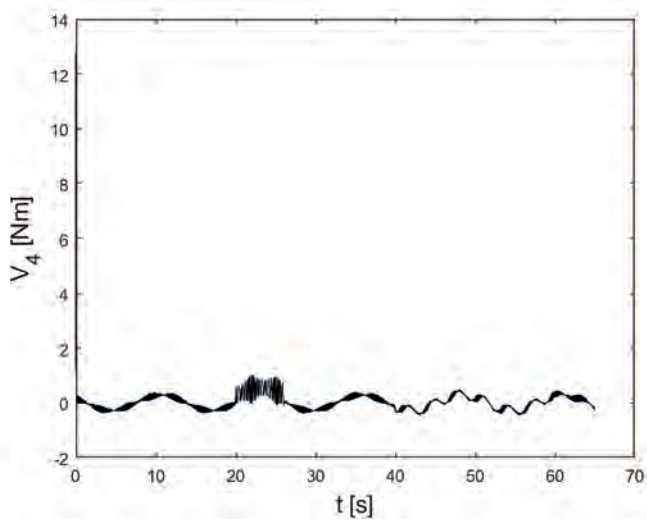

Fig. 9. Torque v_4 on motor m_4 versus time (simulation results)
Rys. 9. Moment obrotowy v_4 na silnik m_4 w funkcji czasu (wynik symulacji)

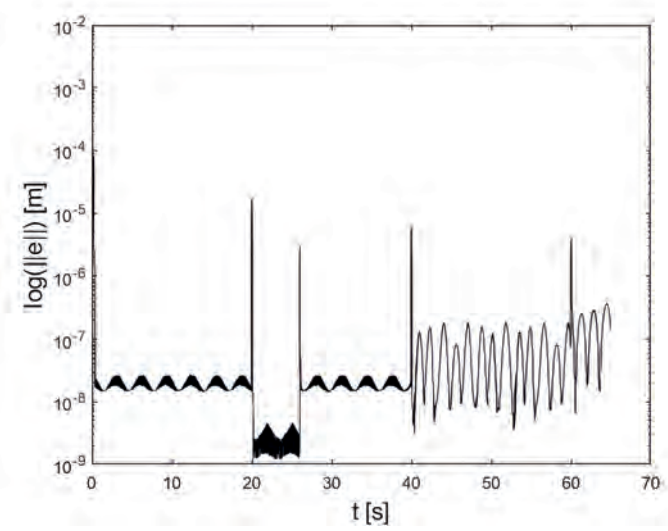

Fig. 10. Control errors in simulation
Rys. 10. Błędy sterowania w symulacji

4.3. Experiment

During the experiment the mobile robot was required to follow predefined trajectory x_d^M (a circle – 3 m diameter), while consistently facing the circle's center. The equation for the desired trajectory is presented in chapter 4.1 (Equation 8). During the experiment, the ARAMIS system tracks the mobile robot that is trying to follow a predefined trajectory and provides information about its position, orientation and displacement from predefined trajectory. The control algorithm aimed to maintain the desired trajectory by adjusting the velocity of each wheel via PWM signals. The goal was to achieve the most accurate representation of the task trajectory in reality.

During the experiment on a substrate without a mat, slippage prevented the platform from working effectively. To ensure proper adhesion conditions, the entire experiment was carried out on a special mat, which provided adequate friction due to its rough surface. The parameters in Equation 7 (c, c', a) were tuned manually by trial and error to obtain the most accurate trajectory possible.

4.4. Experiment results

The mobile robot trajectory obtained during the experiment deviated significantly from the predefined trajectory, but still reached similar values. The highest torque on the motor was 25 Nm, while the lowest was –23 Nm. During the rest of the experiment, the torque values remained at similar values in the range of –8 Nm to 8 Nm. Logarithm of control errors during the experiment did not exceed 10^{-3}. The results of the experiment are shown in Figs. 12–16.

5. Discussion

By analyzing the simulation run and experiment results and the magnitude of errors in the simulated trajectory, it can be assumed

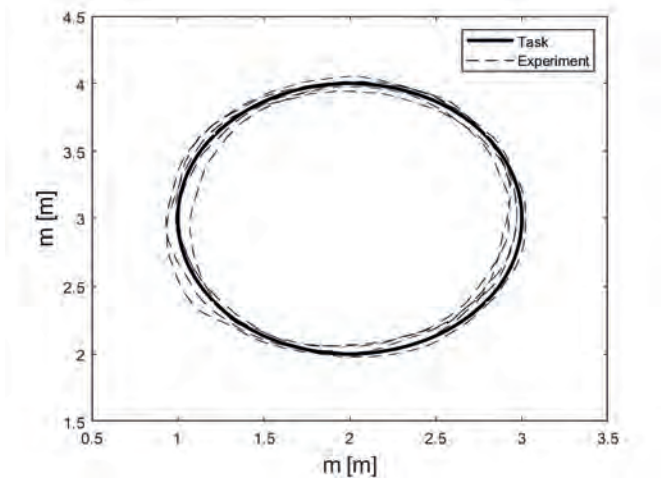

Fig. 11. Task trajectory and experiment results
Rys. 11. Trajektoria zadania i wyniki eksperymentu

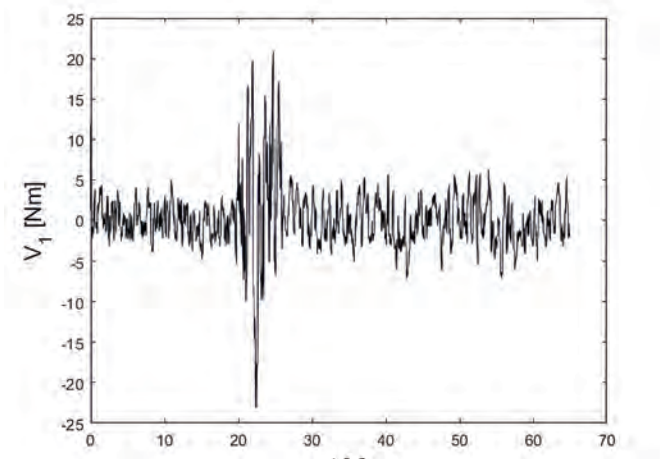

Fig. 12. Torque v_1 on motor m_1 during the experiment versus time
Rys. 12. Moment obrotowy v_1 na silnik m_1 podczas eksperymentu w funkcji czasu

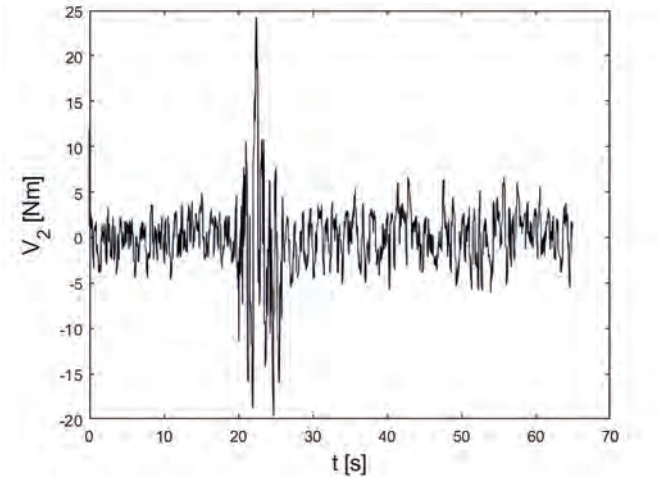

Fig. 13. Torque v_2 on motor m_2 during the experiment versus time
Rys. 13. Moment obrotowy v_2 na silnik m_2 podczas eksperymentu w funkcji czasu

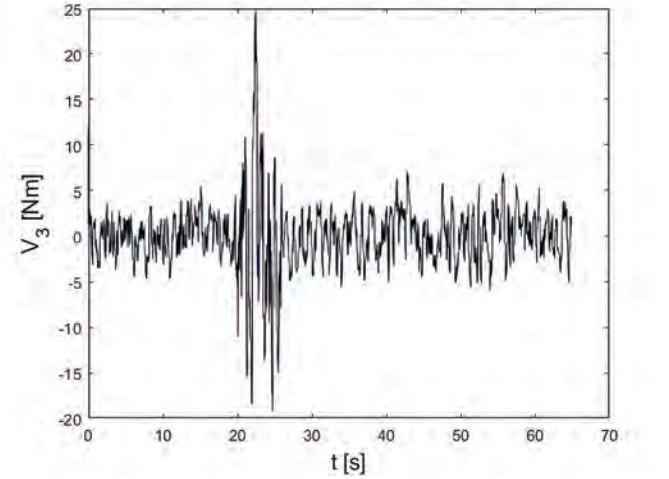

Fig. 14. Torque v_3 on motor m_3 during the experiment versus time
Rys. 14. Moment obrotowy v_3 na silnik m_3 podczas eksperymentu w funkcji czasu

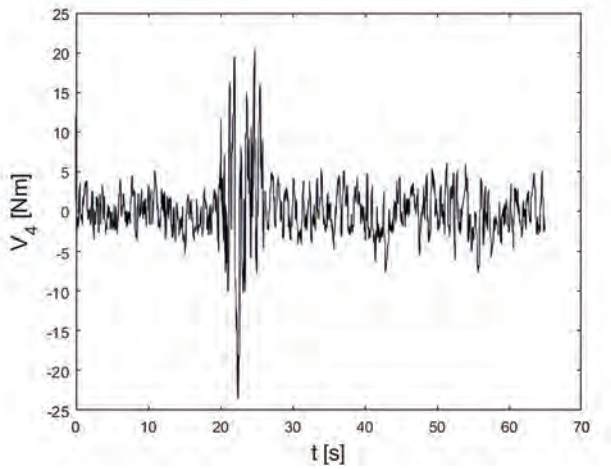

Fig. 15. Torque v_4 on motor m_4 during the experiment versus time
Rys. 15. Moment obrotowy v_4 na silnik m_4 podczas eksperymentu w funkcji czasu

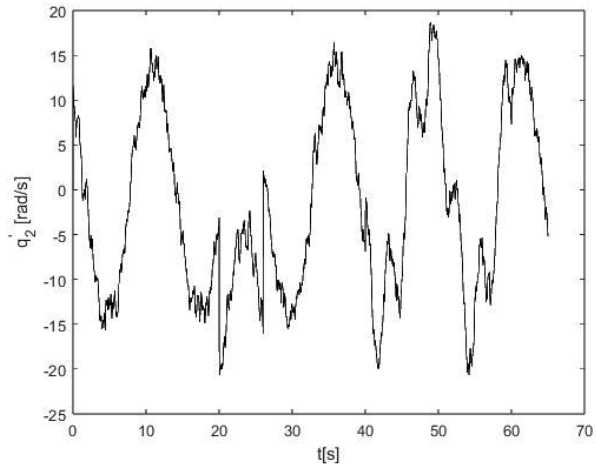

Fig. 18. Angular velocity of wheel q'_2
Rys. 18.Prędkość kątowa koła q'_2

that it is possible to build a working control system for a mobile robot from available shop components. During the largest disturbance, which was in the period of 20–26 s of the simulation, logarithm of the control errors reached a value of no more than 10^{-4}. The torque reached a value of 2 Nm in this interval. The actual trajectory deviated slightly from the predefined trajectory (the differences in the simulation were smaller). Wheel torques in the peak reached 20 Nm, while the lowest logarithm of control error during the experiment was less than 10^{-3}. Most of the time the errors oscillate around $10^{-2} - 10^{-1}$. The use of a precision vision system has made it possible to generate accurate measurements, but the use of such an expensive system in a commercial application is impossible. Future research could focus on building a vision system created from off-the-shelf components like individual parts of a mobile robot control system and analyzing its accuracy. The algorithm provides theoretically finite stable tracking, but because the integration is in discrete form it introduces numerical errors and makes the integration algorithm a discrete

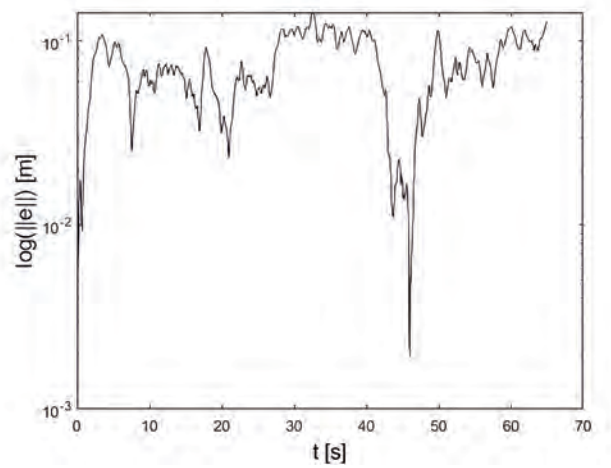

Fig. 16. Control errors during the experiment
Rys. 16. Błędy sterowania podczas eksperymentu

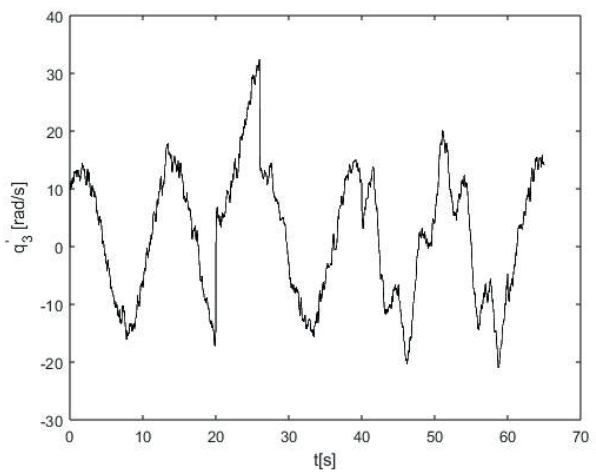

Fig. 19. Angular velocity of wheel q'_3
Rys. 19. Prędkość kątowa koła q'_3

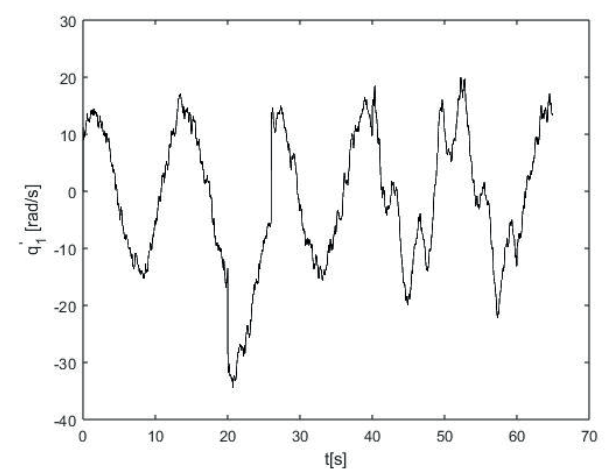

Fig. 17. Angular velocity of wheel q'_1
Rys. 17.Prędkość kątowa koła q'_1

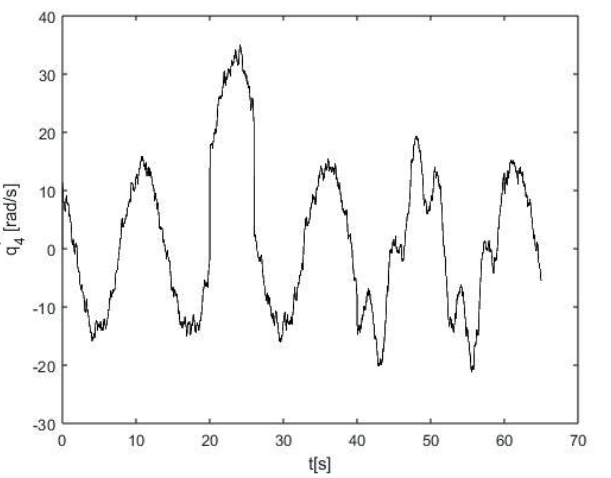

Fig. 20. Angular velocity of wheel q'_4
Rys. 20. Prędkość kątowa koła q'_4

algorithm. This makes it stable with the ability to reduce tracking error along with the ability to enhance tracking (parameter c'). Fluctuations in the speed of the robot were quite significant (Figs 17—20), but thanks to the use of an algorithm with the ability to reduce the effect of chattering, the robot was able to maintain the set trajectory.

6. Conclusion

The results of this study demonstrate that a mobile robot control system can be effectively constructed using commercially available components. It is capable of executing the robot's control algorithm in real time and transmitting the control signals to the motors. The differences between the predefined trajectory and the actual trajectory were small enough that, in the author's opinion, the system can be called accurate. It is planned to extend the entire mobile robot system to include coupling with an integrated vision system on a mobile platform, and to extend the algorithm in such a way that it will be able to recognize obstacles and correct the trajectory to avoid collisions. Moreover the next step will be to expand the system with dedicated chips (e.g. Field Programmable Gate Array) with further optimization of the control algorithm for better tracking desired trajectory.

Bibliography

1. Alakshendra V., Chiddarwar S., *Adaptive robust control of mecanum-wheeled mobile robot with uncertainties*, "Nonlinear Dynamics", Vol. 87, 2017, 2147–2169, DOI: 10.1007/s11071-016-3179-1.
2. Banaszkiewicz M., Węgrzyn M., Basmadji F.L., Galicki M., *Robust trajectory tracking control of space manipulators in extended task space*, "Pomiary Automatyka Robotyka", Vol. 26, No. 4, 2022, 27–35, DOI: 10.14313/PAR_246/27.
3. Diegel O., Badve A., Bright G., Potgieter J., Tlale S., *Improved mecanum wheel design for omnidirectional robots*, Australasian Conference on Robotics and Automation, 2012, 117–121.
4. Galicki M., Banaszkiewicz M., *Optimal trajectory tracking control of omni-directional mobile robots*, 12th International Workshop on Robot Motion and Control (RoMoCo), 2019, 137–142, DOI: 10.1109/RoMoCo.2019.8787377.
5. Han K.-L., Choi O.-K., Kim J., Kim H., Lee J.S., *Design and control of mobile robot with mecanum wheel*, 2009, 2932–2937.
6. Józwik J., Ostrowski D., *Wybrane problemy badawcze robotów przemysłowych*. Politechnika Lubelska, 2016.
7. Lin L.-C., Shih H.-Y., *Modeling and adaptive control of an omni-mecanum-wheeled robot*, "Intelligent Control and Automation", Vol. 4, No. 2, 2013, 166–179, DOI: 10.4236/ica.2013.42021.
8. Sahoo S.R., Chiddarwar S.S., Alakshendra V., *Intuitive dynamic modeling and flatness-based nonlinear control of a mobile robot*, "Simulation", Vol. 94, No. 9, 2018, 797–820, DOI: 10.1177/0037549717741192.
9. Taheri H., Qiao B., Ghaeminezhad N., *Kinematic model of a four mecanum wheeled mobile robot*, "International Journal of Computer Applications", Vol. 113, No. 3, 2015, 6–9, DOI: 10.5120/19804-1586.
10. Tzafestas S.G., *Introduction to mobile robot control*, Elsevier, 2013.
11. Yang Y.-C., Cheng C.-C., *Robust adaptive trajectory control for an omnidirectional vehicle with parametric uncertainty*, "Transactions of the Canadian Society for Mechanical Engineering", Vol. 37, No. 3, 2015, 405–413, DOI: 10.1139/tcsme-2013-0030.

Sprzętowa implementacja sterowania ślizgowego robota mobilnego z czterema kołami typu mecanum

Streszczenie: W artykule omówiono projekt systemu sterowania dla robota mobilnego zdolnego do obliczania algorytmu sterowania w czasie rzeczywistym. Przedstawiono budowę robota mobilnego z opisem komponentów, z których został wykonany zarówno pod względem hardwareowym, jak i softwareowym. Opisano zastosowany system wizyjny w celu analizy jego ruchu. Wykonano symulację oraz eksperyment i ich wyniki zostały zamieszczone w pracy. Całość zwieńczona jest porównaniem uzyskanych wyników oraz planami dalszego rozwoju systemu.

Słowa kluczowe: robot mobilny, nieustrukturyzowane siły zakłócające, odporne skończone czasowo sterowanie w przestrzeni zadaniowej, system sterowania

Damian Nagajek, MSc Eng.
D.Nagajek@cbk.waw.pl
ORCID: 0000-0001-7598-2449

Gradute of Mechanics and Mechanical Engineering and Computer Science at the University of Zielona Góra. He is a PhD student. Since 2019 he has been with the Space Robot Dynamics Lab in CBK PAN (branch in Zielona Góra).