UDC 62-529
DOI: 10.15587/2706-5448.2023.274846

**Chengjian Dong,
Oleksii Povorozniuk,
Andriy Topalov,
Kai Wang,
Zhicong Chen**

# DEVELOPMENT OF THE CONTROL SYSTEM FOR LEGO MINDSTORMS EV3 MOBILE ROBOT BASED ON MATLAB/SIMULINK ELEMENTS

*The Mindstorms EV3 robot, developed by LEGO, is one of the popular robots that has been widely used in various fields. Unlike previous versions of mobile LEGO robots, EV3 allows the development of real-time applications for teaching a variety of subjects, as well as for conducting research experiments. The object of research in this case is the Mindstorms EV3 robot connected to MATLAB/Simulink. The design consists of a controller, one color sensor, two servo motors and one support wheel. Each servo motor is built on a DC collector motor with a matching gearbox and has the ability to measure the number of revolutions corresponding. A digital sensor with a sampling frequency of 1 kHz is used as a color sensor, which can determine the color or brightness of light. Despite its popularity, the EV3 robot control system in interaction with the MATLAB/Simulink programming environment is a rather complex solution and therefore requires further research. The scientific part of the research focuses on discovering the regularities of the Mindstorms EV3 control system, developing a control system model, and exploring the potential of MATLAB/Simulink to expand the robot's capabilities. An analysis of the main elements of the control system, such as sensors and servos, was carried out. The graphs of the dependences of the characteristics of the servo drives were built and the efficiency of the robot movement was checked depending on the parameters set in the program. The result of the development of the mobile robot control system was the adjustment of the mobile robot movement regulators along a given trajectory in the form of a drawn line, which allowed estimating the maximum permissible speed of the robot movement. The presented research and development of a control system based on MATLAB/Simulink elements allows using the proposed method to control a mobile robot with high precision, analyze and verify the robot's electromechanical parameters in real time. This control system has a high potential and can practically be integrated into industrial objects of mobile robotics, provided types the sensors and executive mechanisms of the mobile robot match.*

**Keywords:** *mobile robot, LEGO Mindstorms EV3, MATLAB/Simulink, controller, control system, line-following.*

*Received date: 23.01.2023*
*Accepted date: 27.02.2023*
*Published date: 28.02.2023*

***How to cite***
*Dong, C., Povorozniuk, O., Topalov, A., Wang, K., Chen, Z. (2023). Development of the control system for LEGO Mindstorms EV3 mobile robot based on MATLAB/Simulink elements. Technology Audit and Production Reserves, 1 (2 (69)), 30–35. doi: https://doi.org/10.15587/2706-5448.2023.274846*

## 1. Introduction

During the last few years there have seen significant advances in robotics both in terms of hardware capabilities as well as in terms of artificial intelligence. Arduino, Raspberry Pi, and Kinect are the most common low-cost devices for developing application control systems [1–4]. In terms of robotics, a variety of low-cost mobile robots such as Adept, E-Puck, Moway, LEGO Mindstorms, etc. have emerged in the last few years [5–7]. These devices do not offer the same precision as industrial robots, but they are sufficient for educational purposes in engineering or even for experiments and research tests.

In the field of robotics, LEGO platforms are useful at various application levels, which are confirmed by a number of works. For example, in [8], LEGO Mindstorms was used in mobile robotics to implement Kalman filter (EKF). In [9], describes a set of projects that implement artificial intelligence techniques using LEGO Mindstorms NXT. In [10], LEGO NXT Mindstorms was used in mechatronics, to teach the basics of mobile robot control, remote control, line tracing, fuzzy logic control techniques, and reactive navigation. In [11], cational hands-on project subsystems, a web camera, and convenient tools for finding and mapping indoor obstacles were proposed, and LEGO robot tracking was implemented using color marker detection and EKF.

LEGO Mindstorms EV3 is a robot building set that allows creating and programming your own autonomous robots. There are various programming languages (C, Java, ADA, Phyton, LabView) that can be used to program these robots, but in this article, let's focus on using MATLAB [12, 13]. MATLAB is a popular software for numerical computation and data analysis, which can be used for robot development and programming.

Pros of using MATLAB for programming LEGO Mindstorms EV3 robots:

1. Ease of use: MATLAB has a simple and intuitive interface that makes robot programming easy.

2. Rich functionality: MATLAB provides a rich set of functions for numerical computations and data analysis, which can be useful when designing and programming robots.

3. Wide support: MATLAB has broad support and a large user community that can help with troubleshooting.

4. Integration with other tools: MATLAB can integrate with other tools and programming languages, allowing for expanded robot capabilities.

5. Support for various operating systems: MATLAB supports various operating systems, making it possible to use it on different devices.

Cons of using MATLAB for programming LEGO Mindstorms EV3 robots:

1. High cost: MATLAB is paid software, which can be a barrier for some users.

2. High system requirements: MATLAB requires a high-performance system, which can be an issue for users with weaker computers.

3. Limited capabilities: MATLAB may be limited in certain aspects of robot programming, especially when high speed and accuracy are required.

4. Limited selection of add-on modules: MATLAB has a limited selection of add-on modules for robots, which can be problematic for users who want to use a specific module or sensor.

5. Additional tools needed: Programming a robot with MATLAB may require additional equipment and tools, which can increase the overall cost of using the software.

The MATLAB Support Package for LEGO Mindstorms EV3 provides MATLAB functions to control the motors and interface with hardware input sensors and output capabilities. This support package enables the following tasks to be performed directly from MATLAB:

- Motor control.
- Reading color and light intensity.
- Measuring distance and proximity.
- Detecting touch.
- Acquiring rotation angle and speed.
- Displaying text on the EV3 Brick LCD.
- Emitting sound signals and playing tones on the EV3 speaker.
- Detecting button presses.

The LEGO EV3 robot can be connected to a MATLAB desktop using Bluetooth, Wi-Fi, or USB, and then MATLAB and the LEGO EV3 sensors can be used to program the robot to perform tasks such as line following, object detection, object sorting and collision avoidance.

Simulink support package for LEGO Mindstorms EV3. The Simulink support package for LEGO Mindstorms EV3 enables programming and running algorithms on LEGO Mindstorms EV3 robots. The support package includes a library of Simulink blocks (Fig. 1) for configuring and accessing the robot's sensors, actuators, and communication interfaces.

It is possible to develop algorithms in Simulink, simulate them to ensure they function correctly, and load the completed algorithm for autonomous execution on the device (via USB, Bluetooth, Wi-Fi, or Ethernet connections). It is also possible to adjust parameters in real-time from the Simulink model, and the algorithm executes on the EV3 block.

The support package includes:

- A library of Simulink blocks that connect to LEGO Mindstorms EV3 I/O, including buttons and speakers; numerous input sensors, including color, touch, gyroscopic, ultrasonic, infrared, and encoders; and output sensors such as servo motors, status indicators, and displays.
- Support for LEGO Mindstorms NXT sensors on LEGO Mindstorms EV3 Bricks, including ultrasonic, touch, sound, battery, timer, and gyroscope.
- Interactive configuration of output parameters and monitoring of sensor input signals for applications running on LEGO Mindstorms EV3.
- UDP and TCP/IP blocks that allow communication with LEGO Mindstorms EV3 from Arduino, Raspberry Pi™, and mobile devices (Android and iOS).
- Deployment of the model for autonomous operation.

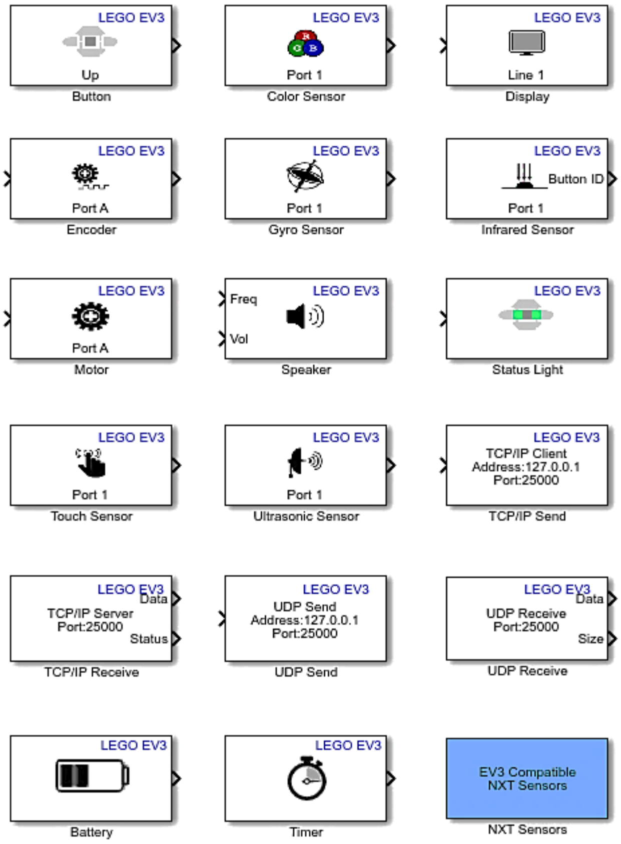

**Fig. 1.** Simulink block library for EV3

Despite its popularity, the EV3 robot control system in interaction with the MATLAB programming environment is a rather complex solution and therefore requires further research. *The purpose of this study* is to develop an effective control system for the EV3 robot using MATLAB with the study of the performance of the executive mechanisms (motors) of the EV3 robot.

## 2. Materials and Methods

*The object of research* in this case is the EV3 robot developed by LEGO. The research was carried out using MATLAB software. The theoretical methods employed in this study include system identification and control theory.

In this case, the mobile robot is a differential vehicle with two servomotors, each connected to one of the wheels B and C. Each servo motor is built on a DC collector motor with a matching gearbox and has the ability to measure the number of revolutions corresponding to the wheel sensor. The mobile robot also has a color sensor that is mounted on the front of the vehicle and pointed directly at the ground.

The color sensor is a digital sensor that can detect the color or brightness of light entering a small window on the front of the sensor. This sensor can operate in three different modes: Color Mode, Reflected Light Intensity Mode, and Ambient Light Intensity Mode. The sampling rate of the color sensor is 1 kHz. In Reflected Light Intensity mode, the color sensor detects the brightness of light reflected from a red lamp. The sensor uses a scale from 0 (very dark) to 100 (very light). This means that the robot can be programmed to move on a white surface until a black line is detected.

Equations that describe the electric processes in a DC motor. In the armature circuit of the motor, a current $I_A$ flows under the influence of the DC voltage $U_A$ of the power supply and the counter-EMF of the motor $E$. This circuit is characterized by the parameters: the active resistance $R_A$ [Ohm] and inductance $L_A$ [H] of the armature winding, as well as the active resistances and inductances of the connected series choke ($R_{CH}$ [Ohm], $L_{CH}$ [H]) and transformer ($R_{TR}$ [Ohm], $L_{TR}$ [H]). The rotating rotor, which has a moment of inertia $J_A$ [kg·m$^2$], and the working element connected to it with the moment of inertia $J_{WE}$ are set in motion simultaneously by the action of the electromagnetic moment of the motor $M_{EMM}$ and the moment of resistance of external forces $M_N$ applied to the working element.

The output differential equations of the DC motor are based on the laws of physics. The second Kirchhoff's law is used for the electric circuit, according to which the equation can be written as:

$$U_A - E = R_{A\Sigma} I_A + L_{A\Sigma}\frac{dI_A}{dt}, \tag{1}$$

where $t$ is the current time; $R_{A\Sigma}I_A$ characterizes the voltage drop on the active resistance of the armature circuit according to Ohm's law, and $L_{A\Sigma}$ $(dI_A/dt)$ reflects the presence of the counter-EMF of self-induction that arises in the winding when the armature current changes. The voltage drop on the brushes, which depends nonlinearly on the armature current, is not taken into account in equation (1), but usually has a relatively small value compared to the voltage $U_A$.

The differential equation that characterizes the processes in the mechanical part of the motor is formed on the basis of the second law of Newton:

$$J_A\frac{d\omega_M}{dt} = M_{EMM} - M_N, \tag{2}$$

where $\omega_M$ is angular speed of rotation of the motor shaft.

This equation does not take into account the action of friction forces that arise during the rotation of the rotor and have a relatively weak effect on the acceleration of the DC motor shaft.

Using the formulas provided above and bringing the differential equations to the normal Cauchy form, let's obtain the description of the DC motor in the form:

$$\begin{cases} \dfrac{dI_A}{dt} = \dfrac{U_A - R_{A\Sigma}I_A - k_E\omega_M}{L_{A\Sigma}}; \\ \dfrac{d\omega_M}{dt} = \dfrac{k_M I_A - M_N}{J_A}. \end{cases} \tag{3}$$

The development of the control program for a mobile robot system that provides movement of the robot along a line of an arbitrary trajectory (Fig. 2) using an automaton approach. The system consists of two parts: the robot executor and a user interface for a personal computer (PC). The robot executor and the interface communicate with each other through a USB channel. The robot executor works in two modes: line movement (autonomous) and movement controlled by the PC user. To enable the robot to find and recognize a line on the ground, a color sensor must be used. For the color sensor to work correctly, the line thickness should be approximately 2 cm.

The control program algorithm is shown in Fig. 3. At the start, the color sensor determines the color of the object it is pointed at. Then, the value obtained from the color sensor is compared to a set threshold value using a comparator. If the value obtained from the color sensor is greater or less than the threshold value, the comparator will output the corresponding binary signal «1» or «0». The binary signal obtained from the comparator can be used to decide on further action. For example, if the binary signal is equal to «1», it means that the object's color meets a certain condition, and if the binary signal is equal to «0», the condition is not met.

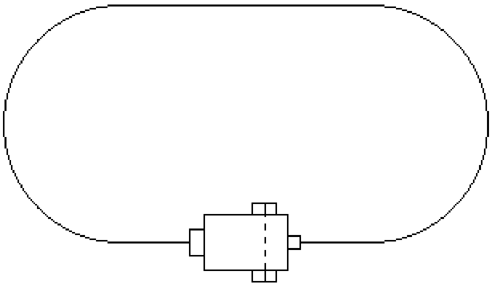

**Fig. 2.** Trajectory of the mobile robot's movement

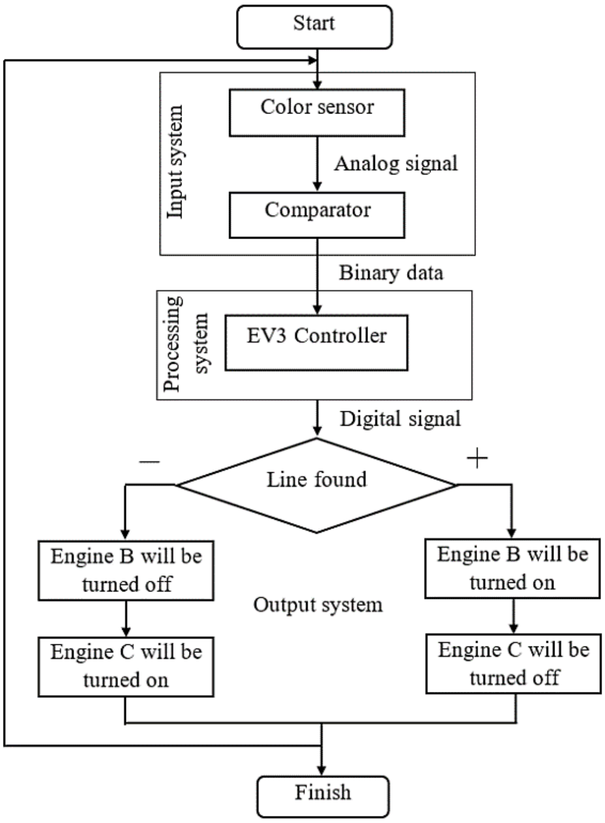

**Fig. 3.** Control program algorithm

The EV3 Controller processes the binary signal and makes the appropriate decision about operating the servos based on the set parameters. Depending on the decision made by the EV3 Controller, a digital signal is generated, which will be sent to the servos.

## 3. Results and Discussion

Line following is a classic robotics program that combines algorithms for sensing, actuation, and control robots designed for line following are relatively simple to create and can navigate along a predetermined path autonomously without human intervention. This model (Fig. 4) implements a line-following algorithm on a two-wheeled EV3 robot with a single color sensor.

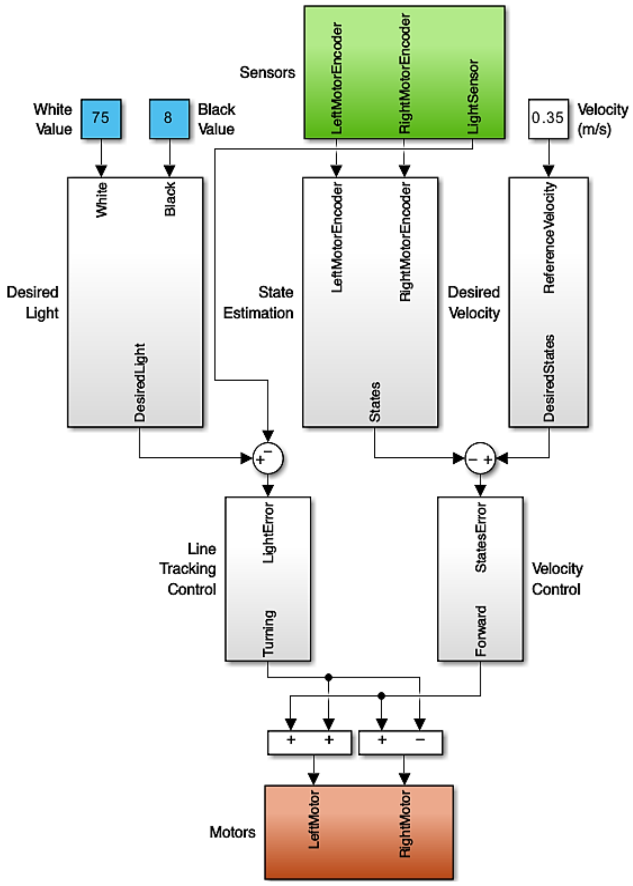

**Fig. 4.** MATLAB control system model for a mobile robot

The robot uses feedback from the color sensor to follow the line and feedback from the motor sensors to maintain forward motion speed. Two controllers control speed and tracking. The controller is a common technique used in various kinds of machineries such as vehicles, robots and even rockets, etc. The controller calculates the difference between the output and a required set-point, and attempts to minimize this difference by adjusting the inputs of system [14].

The sensors and motors contain blocks that interact with the EV3 hardware. State estimation uses the value of each motor sensor to calculate the position and speed of the robot. The desired speed takes the user-specified speed (m/s) and converts it to the required state values for the speed controller. The desired light source takes the white and black color sensor values to select the appropriate reference value for the light source.

The robot is capable of following a line with smooth turns, but on sharp turns, the robot loses the line. The speed of the track is quite high. The proportional-integral-derivative (PID) controller allows achieving the best results, but it is quite complex to tune due to the presence of three coefficients. Thus, a PID controller can be recommended for the best routing, but careful tuning of the controller in changing conditions is required. If the speed of the robot along the line is not high, then it is better to use simpler algorithms, for example, PI or PD. The speed control is implemented with a PI controller to control the speed of the movement. The line tracking controller is implemented with a PD controller to control the turn.

In this section, the controller is designed using MATLAB/SIMULINK, where the parameters of the coefficients should be adjusted, for the best distance covered by the robot. There are several ways to adjust the parameters: the gradient descent method, the Ziegler-Nichols method [15], based on the obtained nonlinear dependencies of the robot's behavior. In our case, the behavior of the actuators was studied in detail on the basis of physical modeling of the operation of direct current motors. The following charts (Fig. 5–8) show the characteristics of the EV3 servomotor versus applied load. EV3 was powered at 9 V. Power level is 100 % for all charts.

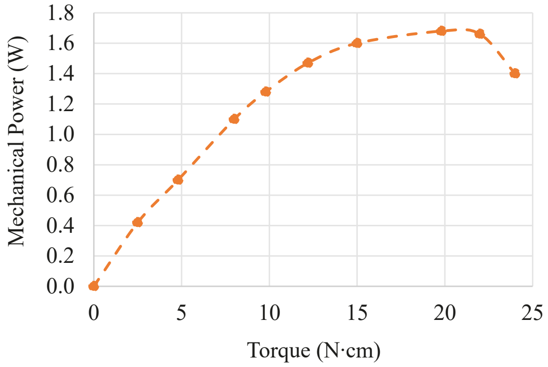

**Fig. 5.** Mechanical Power vs. Torque

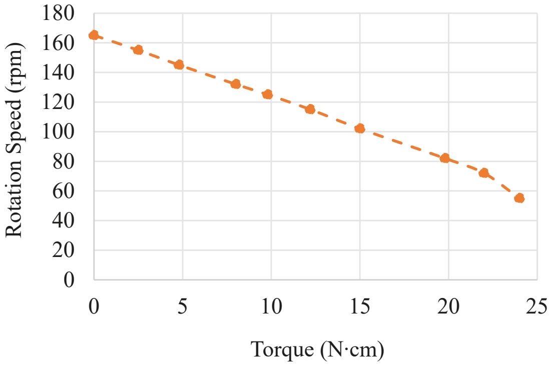

**Fig. 6.** Rotation speed vs. Torque

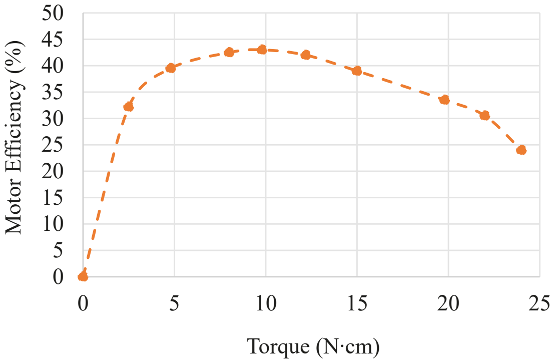

**Fig. 7.** Motor Efficiency vs. Torque

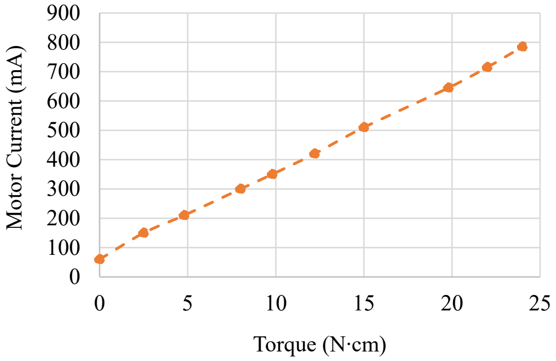

**Fig. 8.** Motor Current vs. Torque

Fig. 5 shows that the maximum mechanical power of 1.7 W is achieved with a torque load of approximately 21 N·cm after which it sharply decreases. In Fig. 6, the speed of rotation in relation to the torque decreases with the load and has an almost linear character. Fig. 7 shows that the maximum efficiency of 43 % of the engine is achieved at a torque load of approximately 10 N·cm. Fig. 8 shows the dependence of the current on the torque, and the current increases linearly with the load, which corresponds to the laws of electromechanics. All these studies were used to effectively configure the mobile robot control system. The results of the system are summarized in Table 1.

**Table 1**

The results of the experiment

| Speed | Result of the experiment |
|---|---|
| 0.1 m/s | The robot drives accurately and smoothly along the track |
| 0.2 m/s | The robot drives accurately and smoothly along the track |
| 0.3 m/s | The robot drives with slight errors on turns |
| 0.35 m/s | The robot drives with a large error after the turn, but always returns to the track |
| 0.4 m/s | The robot deviates from the track after the first turn and does not return |

The proposed mobile robot control system showed high results of following the Mindstorms EV3 mobile robot along the line. At a movement speed of up to 0.2 m/s, the robot travels the full distance without errors. When the speed increases, the controller does not have time to react to the disturbance, which leads to the robot moving away from the distance. In further research, it is planned to improve robot motion controllers based on the use of a fuzzy controller that will process the nonlinear characteristics obtained in this article and issue a control signal taking into account the theory of fuzzy sets.

## 4. Conclusions

In this paper, LEGO Mindstorms EV3 can potentially be used as a robust robotic platform for learning theoretical algorithms. The platform allows implementing control algorithms by simply programming on a high-level language interface.

The results showed that this control system allows ensuring the desired behavior of the robot depending on the color of the object to which the sensor is directed. Speed control has the implementation of a PI controller to control the speed of movement. Line tracking control has a PD controller implementation for position control.

Experimental results show that at a speed of 0.1, 0.2 m/s – the robot drives precisely and smoothly along the track; 0.3 m/s – the robot moves with small errors on turns; 0.35 m/s – the robot drives with a large error after turning, but always returns to the track; 0.4 m/s – the robot goes off track after the first turn and does not return.

The obtained results can be useful for the further development of mobile robot control systems that use servos and a color sensor. For example, based on the obtained results, it is possible to develop control algorithms that will ensure efficient and accurate movement of the robot in various conditions. Positive results were obtained regarding the performance and accuracy of the control system based on MATLAB elements.

## Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

## Financing

This study is financially supported by the National High Level Foreign Experts Introduction Project (G2022014116L) and Yancheng Key Technology Unveiling Project (Research, development and application of intelligent unmanned boat and cluster control technology).

Presentation of research in the form of publication through financial support in the form of a grant from SUES (Support to Ukrainian Editorial Staff).

## Data availability

The manuscript has no associated data.

### References

1. Irigoyen, E., Larzabal, E., Priego, R. (2013). Low-cost platforms used in Control Education: An educational case study. *IFAC Proceedings Volumes, 46 (17),* 256–261. doi: https://doi.org/10.3182/20130828-3-uk-2039.00058
2. Pedersen, B. K. M. K., Larsen, J. C., Nielsen, J. (2019). The Effect of Commercially Available Educational Robotics: A Systematic Review. *Advances in Intelligent Systems and Computing,* 14–27. doi: https://doi.org/10.1007/978-3-030-26945-6_2
3. Zhang, M., Wan, Y. (2020). Improving learning experiences using LEGO Mindstorms EV3 robots in control systems course. *The International Journal of Electrical Engineering & Education.* doi: https://doi.org/10.1177/0020720920965873
4. Ding, J., Li, Z., Pan, T. (2017). Control System Teaching and Experiment Using LEGO MINDSTORMS NXT Robot. *International Journal of Information and Education Technology, 7 (4),* 309–313. doi: https://doi.org/10.18178/ijiet.2017.7.4.886
5. Maharuddin, M. F., Ghani, N. M., Jamin, N. F. (2018). Two-Wheeled LEGO EV3 Robot Stabilisation Control Using Fuzzy Logic Based PSO Algorithm. *Journal of Telecommunication, Electronic and Computer Engineering, 10 (2-5),* 149–153. Available at: https://jtec.utem.edu.my/jtec/article/view/4402
6. Rosillo, N., Montés, N., Alves, J. P., Ferreira, N. M. F. (2019). A Generalized Matlab/ROS/Robotic Platform Framework for Teaching Robotics. *Advances in Intelligent Systems and Computing, 1023,* 159–169. doi: https://doi.org/10.1007/978-3-030-26945-6_15

7. Akmal, M. A., Jamin, N. F., Ghani, N. M. A. (2017). Fuzzy logic controller for two wheeled EV3 LEGO robot. *2017 IEEE Conference on Systems, Process and Control (ICSPC)*. Melaka, 134–139. doi: https://doi.org/10.1109/spc.2017.8313035
8. Serrano, V., Thompson, M., Tsakalis, K. (2017). Learning Multivariable Controller Design: a Hands-on Approach with a Lego Robotic Arm. *Lecture Notes in Networks and Systems,* 271–278. doi: https://doi.org/10.1007/978-3-319-54377-2_23
9. Pinto, M., Moreira, A. P., Matos, A. (2012). Localization of Mobile Robots Using an Extended Kalman Filter in a LEGO NXT. *IEEE Transactions on Education, 55 (1),* 135–144. doi: https://doi.org/10.1109/te.2011.2155066
10. Cuéllar, M. P., Pegalajar, M. C. (2011). Design and implementation of intelligent systems with LEGO Mindstorms for undergraduate computer engineers. *Computer Applications in Engineering Education, 22 (1),* 153–166. doi: https://doi.org/10.1002/cae.20541
11. Canale, M., Casale-Brunet, S. (2014). A multidisciplinary approach for Model Predictive Control Education: A Lego Mindstorms NXT-based framework. *International Journal of Control, Automation and Systems, 12 (5),* 1030–1039. doi: https://doi.org/10.1007/s12555-013-0282-7
12. Montés, N., Rosillo, N., Mora, M. C., Hilario, L. (2018). Real-Time Matlab-Simulink-Lego EV3 Framework for Teaching Robotics Subjects. *Advances in Intelligent Systems and Computing,* 230–240. doi: https://doi.org/10.1007/978-3-319-97085-1_23
13. Montes, N., Rosillo, N., Mora, M. C., Hilario, L. (2021). A Novel Real-Time MATLAB/Simulink/LEGO EV3 Platform for Academic Use in Robotics and Computer Science. *Sensors, 21 (3),* 1006. doi: https://doi.org/10.3390/s21031006
14. Basilio, J. C., Matos, S. R. (2002). Design of PI and PID controllers with transient performance specification. *IEEE Transactions on Education, 45 (4),* 364–370. doi: https://doi.org/10.1109/te.2002.804399
15. Ünal, M., Erdal, H., Topuz, V. (2010). Trajectory tracking performance comparison between genetic algorithm and ant colony optimization for PID controller tuning on pressure process. *Computer Applications in Engineering Education, 20 (3),* 518–528. doi: https://doi.org/10.1002/cae.20420

***Chengjian Dong,*** *School of Automobile and Transportation, Yancheng Polytechnic College, Yancheng, China, ORCID: https://orcid.org/0000-0003-3529-6529*

✉***Oleksii Povorozniuk,*** *Postgraduate Student, Department of Computerized Control Systems, Admiral Makarov National University of Shipbuilding, Mykolaiv, Ukraine, e-mail: alexey99195@gmail.com, ORCID: https://orcid.org/0000-0002-0455-9915*

***Andriy Topalov,*** *PhD, Associate Professor, Department of Computerized Control Systems, Admiral Makarov National University of Shipbuilding, Mykolaiv, Ukraine, ORCID: https://orcid.org/0000-0003-2745-7388*

***Kai Wang,*** *Yunzhou (Yancheng) Innovation Technology Co., Ltd., Yancheng, China, ORCID: https://orcid.org/0009-0008-1175-6045*

***Zhicong Chen,*** *Yunzhou (Yancheng) Innovation Technology Co., Ltd., Yancheng, China, ORCID: https://orcid.org/0009-0001-2511-8289*

✉*Corresponding author*