A comparative study of bug algorithms for robot navigation*

Mykhaylo Strembitskyi[1,*,†], Myroslava Yavorska[1,†], Andriy Palamar[1,*,†], Roman Kochan[2,†] and Valeriy Yeromenko[3,†]

[1] Ternopil Ivan Puluj National Technical University, Ruska str., 56, 46001, Ternopil, Ukraine
[2] Lviv Polytechnic National University, 79013, 12 Stepan Bandera Street, Lviv, Ukraine
[3] West Ukrainian National University, Lvivska str. 11, Ternopil, 46009, Ukraine

Abstract

An important stage in the design of robotic cyber-physical immunosensors systems is the development and research of their mathematical models, which would adequately reflect the important, from the point of view of the problems of research, aspects of the spatial structure of immunopixels. After all, the quality of the model determines the effectiveness of their processing methods in immunosensors systems.
The peculiarities of the laser distance sensors application for the detection of obstacles and positioning of a mobile robotic platform in space is considered.
The work developed a robotic platform using laser distance sensors. A method of determining dynamic obstacles and a method of positioning are proposed. The calculation of non-contact sensors of the distance of the robotic platform, the scheme of placement of sensors and positioning of the mobile robotic platform in space are described
There is a mobile robotic platform in which a complex of infrared distance sensors is used to determine parallelism, distance to an obstacle and positioning.

Keywords

mobile robotic platform, sensor, algorithm, positioning, linearization

1. Introduction

The planning the movement of a mobile robotic platform, determining obstacles and positioning in space are one of the most important problems of the operation of robotic platforms and one of the areas of modern scientific and practical knowledge that is being investigated. Robot navigation is the problem of finding a path for a robot to move from a start location to a goal location in an environment, while avoiding obstacles and minimizing costs [1]. The problem can be complicated because unknown or dynamic environments, where the robot has limited or uncertain information about its surroundings. One of the approaches to solve this problem is to use bug algorithms, which are simple and efficient techniques that rely on local sensor information and do not require global maps or planning [1, 2]. Bug algorithms are inspired by the behavior of insects, such as ants or cockroaches, that can navigate complex environments using simple rules. Their basic idea is to make the robot move toward the goal until it encounters an obstacle, then follow the obstacle boundary until it finds a point closer to the goal than the point where it hit the obstacle, and then resume moving toward the goal. This process is repeated until the robot reaches the goal or determines that the goal is unreachable [3, 4].

Bug algorithms can be classified into two categories: contact-based and range-based. Contact-based bug algorithms use only tactile sensors, such as bumpers or whiskers, to detect obstacles. Range-based bug algorithms use distance sensors, such as sonar or laser, to measure the distance to

* CITI'2025: 3rd International Workshop on Computer Information Technologies in Industry 4.0, June 11–12, 2025, Ternopil, Ukraine
1* Corresponding author.
† These authors contributed equally.
m.strembitskyy@gmail.com (M. Strembitskyi); myavorska@gmail.com (M. Yavorska); palamar.andrij@gmail.com (A. Palamar); roman.v.kochan@lpnu.ua (R. Kochan); v.yeromenko@wunu.edu.ua (V. Yeromenko)
0000-0002-5713-1672 (M. Strembitskyi); 0000-0001-8033-7348 (M. Yavorska); 0000-0003-2162-9011 (A. Palamar); 0000-0003-1254-1982 (R. Kochan); 0000-0003-1369-8287 (V. Yeromenko)

obstacles and the goal [4]. Contact-based bug algorithms are simpler and more robust, but they may require more time and distance to reach the goal. Range-based bug algorithms are faster and more efficient, but they may be more sensitive to sensor noise and errors [5]

The main features, advantages, and disadvantages of different approaches to bug algorithms constructions, the discuss their performance and applicability in various scenarios and also some examples of practical applications are presented in [6-8]. We were review the main features, advantages, and disadvantages of different bug algorithms [9,10], and discuss their performance and applicability in various scenarios. The development of mathematical models based on difference and differential equations on lattices, has demonstrated high accuracy in simulating spatial and dynamic interactions in complex systems [11-13]. These modeling techniques, which ensure operational stability and precise parameter identification, can inform the design of sensor-based robotic systems where spatial awareness and real-time response are critical [14,15].

2. Range-Based Bug Algorithms

Range-based bug algorithms are more advanced and efficient type of bug algorithms. They were first proposed by Khatib (1986), who introduced a variant called Potential Field Method (PFM). These algorithms assume that the robot has a perfect compass and a distance sensor that can measure the distance to obstacles and the goal. Range-based bug algorithms use the distance sensor information to create a potential field, which is determined with a function that assigns some values to each point in the environment. The potential field is composed of two components: an attractive component, which pulls the robot toward the goal, and a repulsive component, which pushes the robot away from obstacles. The robot's motion is determined by the gradient of the potential field, which indicates the direction of the steepest ascent or descent. The robot moves along the direction of the negative gradient, which leads to the minimum of the potential field.

Range-based bug algorithms are faster and more efficient than contact-based bug algorithms, as they permit to avoid obstacles without touching them and find shorter or optimal paths to the goal. However, range-based bug algorithms may suffer from some drawbacks, concerning local minima, oscillations, and sensitivity to sensor noise and errors.

Bug navigation algorithms are based on some mathematical concepts that describe the robot's motion, sensing, and decision making. For example:

1. The robot's position is represented by a vector $p=(x, y)$ in a Cartesian coordinate system, where x and y are the horizontal and vertical coordinates, respectively.
2. The robot's orientation is represented by an angle θ measured from the positive x-axis in a counterclockwise direction.
3. The robot's goal position is represented by another vector $g=(x_g, y_g)$, which is assumed to be known and fixed.
4. The robot's motion is controlled by two inputs: a linear velocity v and an angular velocity ω. The robot's kinematic model is given by:

$$\frac{dp}{dt}=v\cos\theta\, i+v\sin\theta\, j, \tag{1}$$

$$\frac{d\theta}{dx}=\omega, \tag{2}$$

where i and j are the unit vectors along the x-axis and y-axis, respectively. The robot's distance to the goal is given by the Euclidean norm of the difference between p and g:

$$d(p, g)=\|p-g\|=\sqrt{(x-x_g)^2+(y-y_g)^2}, \tag{3}$$

The robot's direction to the goal is given by the angle between the vector p-g and the positive x - axis:

$$\alpha(p,g)=\arctan\frac{y_g-y}{x_g-x}, \tag{4}$$

The robot's heading error is given by the difference between θ and $\alpha(p,g)$:

$$e(p,\theta,g)=\theta-\alpha(p,g), \tag{5}$$

The robot's distance sensor measures the distance to the nearest obstacle in a given direction β, which is relative to the robot's orientation. The sensor model is given by:

$$r(\beta)=d(p,o(\beta))+n(\beta), \tag{6}$$

where $o(\beta)$ is the position of the obstacle point in direction β, and $n(\beta)$ is the sensor noise, which is assumed to be zero-mean Gaussian with variance σ^2. The robot's intensity sensor measures the signal strength from the goal, which depends on the distance and the signal intensity function $f(d)$. The sensor model is given by:

$$s=f(d(p,g))+m, \tag{7}$$

where m is the sensor noise, which is assumed to be zero-mean Gaussian with variance τ^2.

The robot's decision making is based on switching between two modes: move-to-goal (MTG) and follow-boundary (FB). The switching conditions are given by:

$$\text{MTG} \rightarrow \text{FB}: r(0)<d_s, \tag{8}$$

$$\text{FB} \rightarrow \text{MTG}: d(p,g)<d(l,g) \text{ or } s>s_{max}, \tag{9}$$

where d_s is a safety distance threshold, l is the leave point, and s_{max} is a maximum intensity threshold.

3. Control system implementation method for robotic platform

The control unit considered is designed to control the robotic platform shown in Figure 1.

The first action after the initialization of the components is to poll the distance sensors and process the data that will arrive from the ADC to the controller. The next step will be to determine the direction of rotation of the DC motor, at the same time the position of the servo is set. After these operations are completed, a control signal is sent to the DC motor, causing it to start.

When the control is carried out, it is possible to change the position of the servo drive, control the speed and direction of rotation of the direct current motor using an encoder.

4. The principle of operation of non-contact infrared distance sensors

The principle of operation of the infrared sensor Sharp GP2Y0A02YK0F is based on the use of an infrared beam that is reflected from an object to measure the distance.

The distance is calculated using the triangulation method [16, 17]. The sensor consists of an infrared LED and PSD element [18]. The emitted light beam is reflected from the object, the reflected beam reaches the PSD of the element on which a "light spot" is formed. When the position of the object changes, the angle of reflection of the beam changes Figure 2.

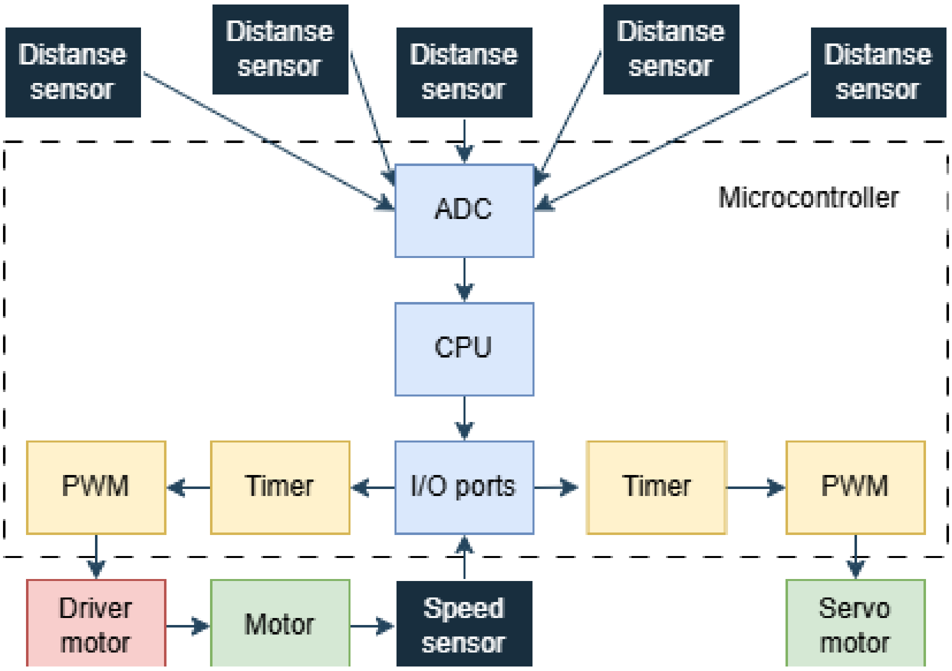


Figure 1: Block diagram for the robotic platform control.

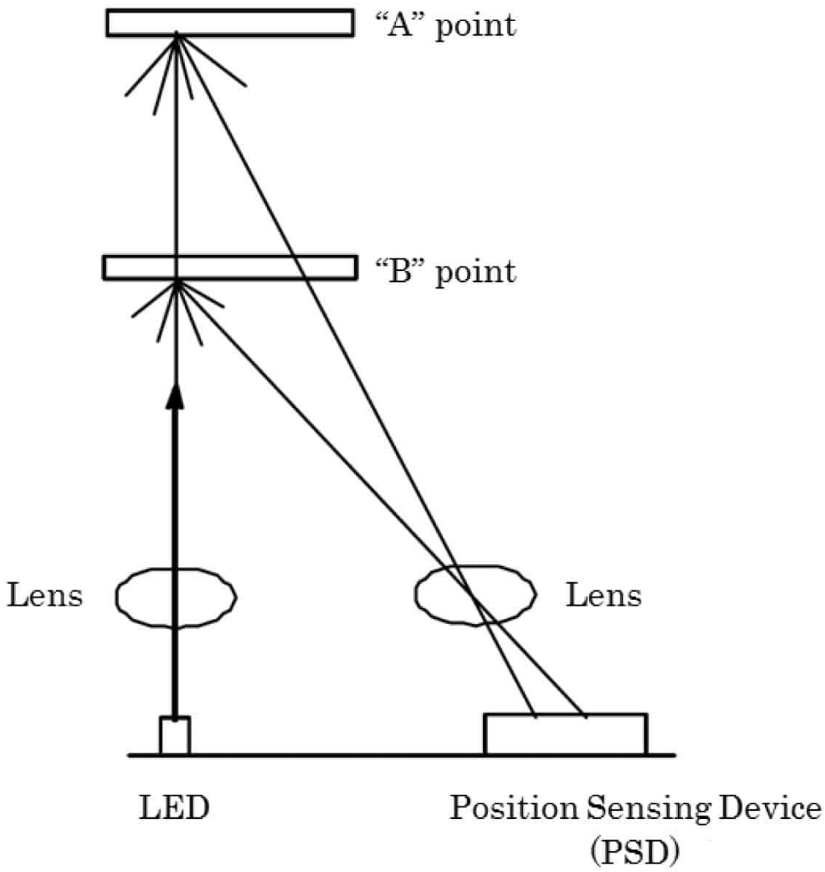


Figure 2: Infrared beam tracway to the element PSD.

The sensor has a built-in circuit for processing the signal, which allows you to determine the distance to the object. The measured distance is represented as an analog signal.

5. Calculation of non-contact distance sensors of the robotic platform

To determine the calibration characteristic of a non-contact distance sensor, we turn to the experimentally determined dependences of the output voltage on the distance to the reflecting surface. The dependencies for the surfaces with different reflective characteristics are shown in Figure 3.

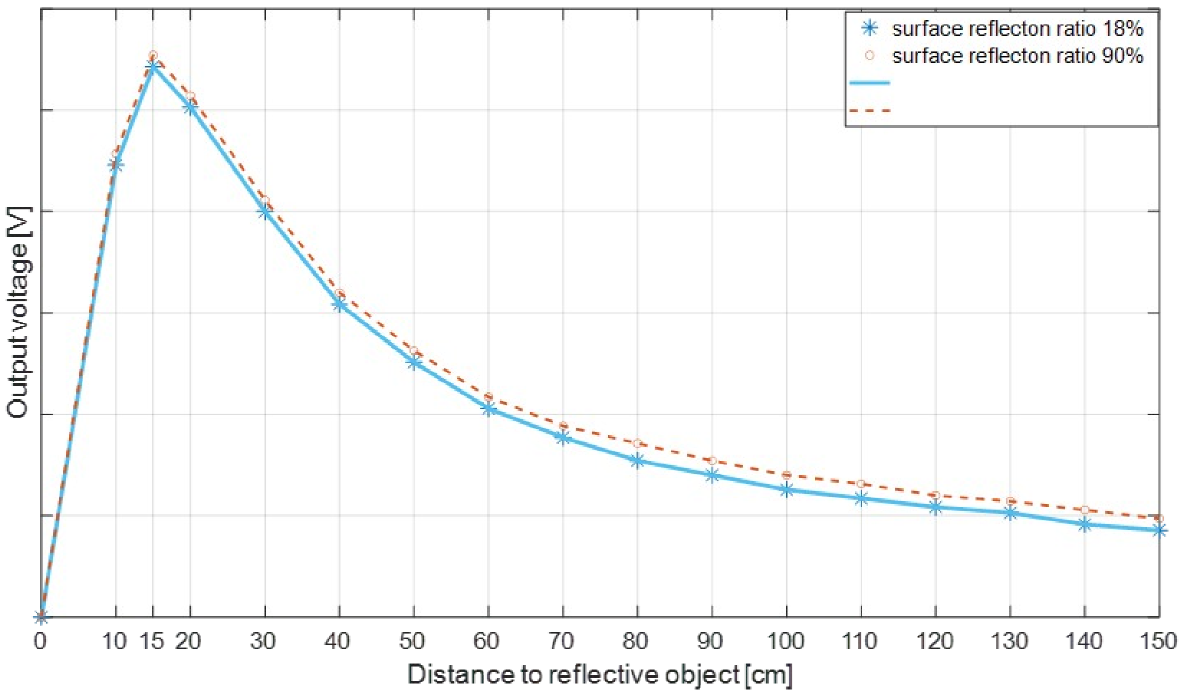


Figure 3: The voltage on distance dependencies for the surfaces with different reflective characteristics.

As one can see from Figure 3, it is possible to specify the operating ranges for which the dependence of the sensor voltage from the distance measured are acceptable.

An example of an approximate distance determination based on information received from a sensor is shown in Figure 4.

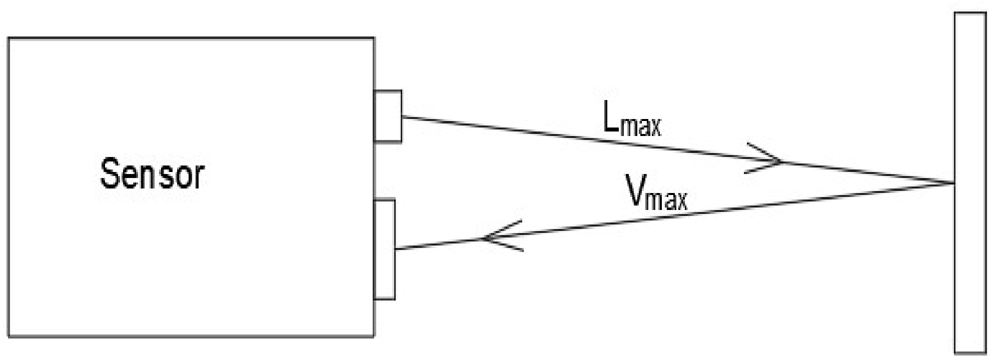


Figure 4: An example of the distance measuring to an object.

Lmax – is the maximum distance from the sensor to the measurement object.

Vmax – is the voltage we get when measuring the maximum possible distance to the object.

For the most optimal distance measurement modes we will operate with the next data: 150 centimeters to the measuring object will equal 0 volts of the output signal, and 0 centimeters equals 2.2 volts.

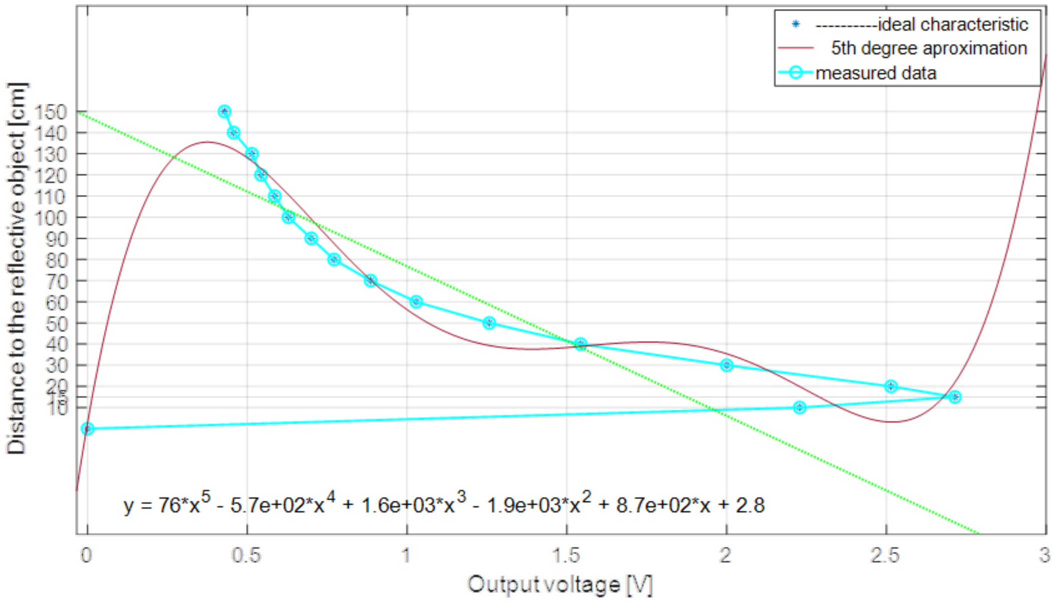


Figure 5: Approximation of the sensor characteristic by a polynomial of the 5th order.

Steepness of the linearized characteristic:

$$\frac{2.8}{150}=0.0187\frac{V}{cm}, \tag{10}$$

Sensor characteristic coefficients after approximation by a 5th order polynomial: [-3. 83, -645.2215, -3849.3362, -1230, 150]

Minimal error of Approximation: 0.0014 cm

Mean error of Approximation: 2.6278 cm

Maximal error of Approximation: 9.1994 cm

Blue Pill ADC resolution = 12-bits

These error metrics are comparable to those achieved in complex systems modeling using numerical simulation and parameter identification techniques, where mean approximation errors below 5% are considered acceptable for real-time applications [19]. The use of polynomial approximation and signal linearization in sensor calibration mirrors approaches used in the analysis of complex systems dynamics, where accurate signal reconstruction is essential [20].

ADC resolution of the microcontroller:

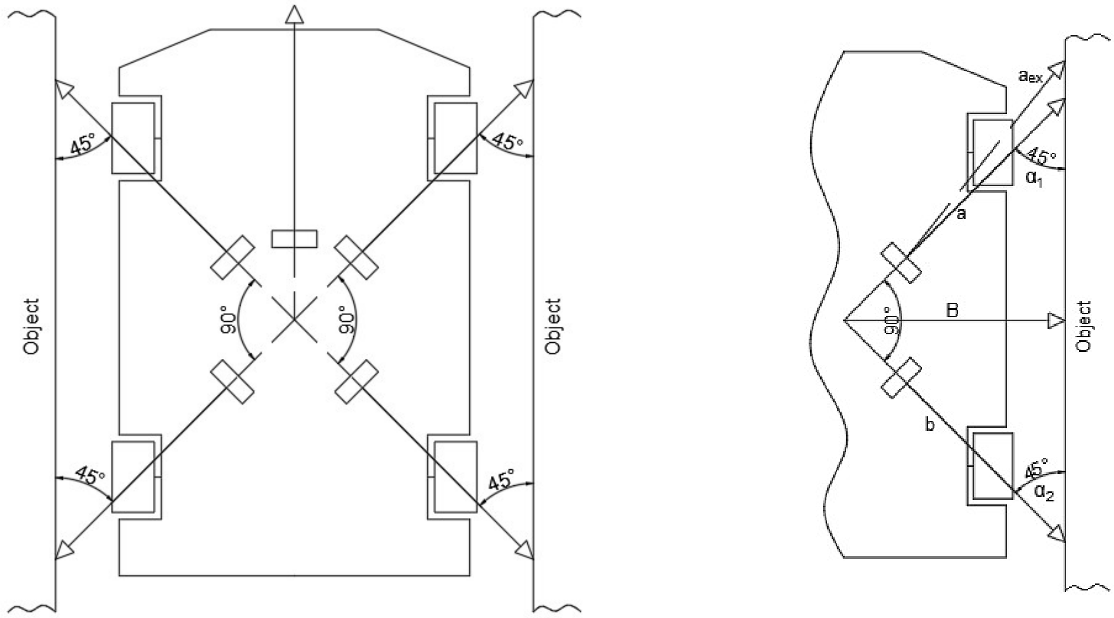
$$\frac{3.3}{2^{12}}=0.0008\frac{V}{bit}, \tag{11}$$

Resolution of the linearized sensor

$$\left(\frac{0.0008}{0.0187}\right)=0.043\,cm, \tag{12}$$

Methods of accurate distance measurement and determination of the parallelism of mobile robotic platforms to obstacles and objects in real time are an important component in the tasks set for modern robotic platforms [21].

Analysis of the interference of mobile robotic platforms in space is carried out with the help of four infrared sensors Sharp GP2Y0A02YK0F. The arrangement of sensors is shown in Figure 6.



a) b)

Figure 6: An example of object distance measurement (a), scheme for calculating parallelism and distance to the object (b).

The distance to the obstacles and parallelism measurement of the mobile robotic platform to objects is determined from an isosceles right triangle. The distance is determined by finding the bisector (*B*) according to the formula (13)

$$B=\sqrt{2}\frac{ab}{a+b}, \tag{13}$$

Since, while maintaining the required accuracy based on the characteristics of the sensor, we can measure the distance (a, b) in the range from 15 to 150 cm, the considered scheme allows us to determine the distance to the obstacle and the parallelism from 0 to106 cm.

To determine parallelism, we define (a_{ex}) the experimental distance and compare with the real (a)

$$a=\frac{B}{\cos 45°}, \tag{14}$$

If $(a_{ex}) \neq (a)$ or, we determine the angle α_1

The angle α is determined using the formula

$$\alpha_1 = atan\left(\frac{a}{b}\right), \tag{15}$$

If $(\alpha_1) > (\alpha)$ then the desired angle is α_1

$$\alpha_1 = atan\frac{2B-a}{\sqrt{3B}}, \tag{16}$$

If $(\alpha_1) < (\alpha)$ then the desired angle is α_1

$$\alpha_1 = atan\frac{a-2B}{\sqrt{3B}}, \tag{17}$$

where a_{ex} is the experimentally determined leg length of an isosceles right triangle.
a – the actual length of the leg of the studied triangle.
α_1 – angle of deviation from the object.
B – bisector of the studied triangle.
α – valid angle to the object is 45°.

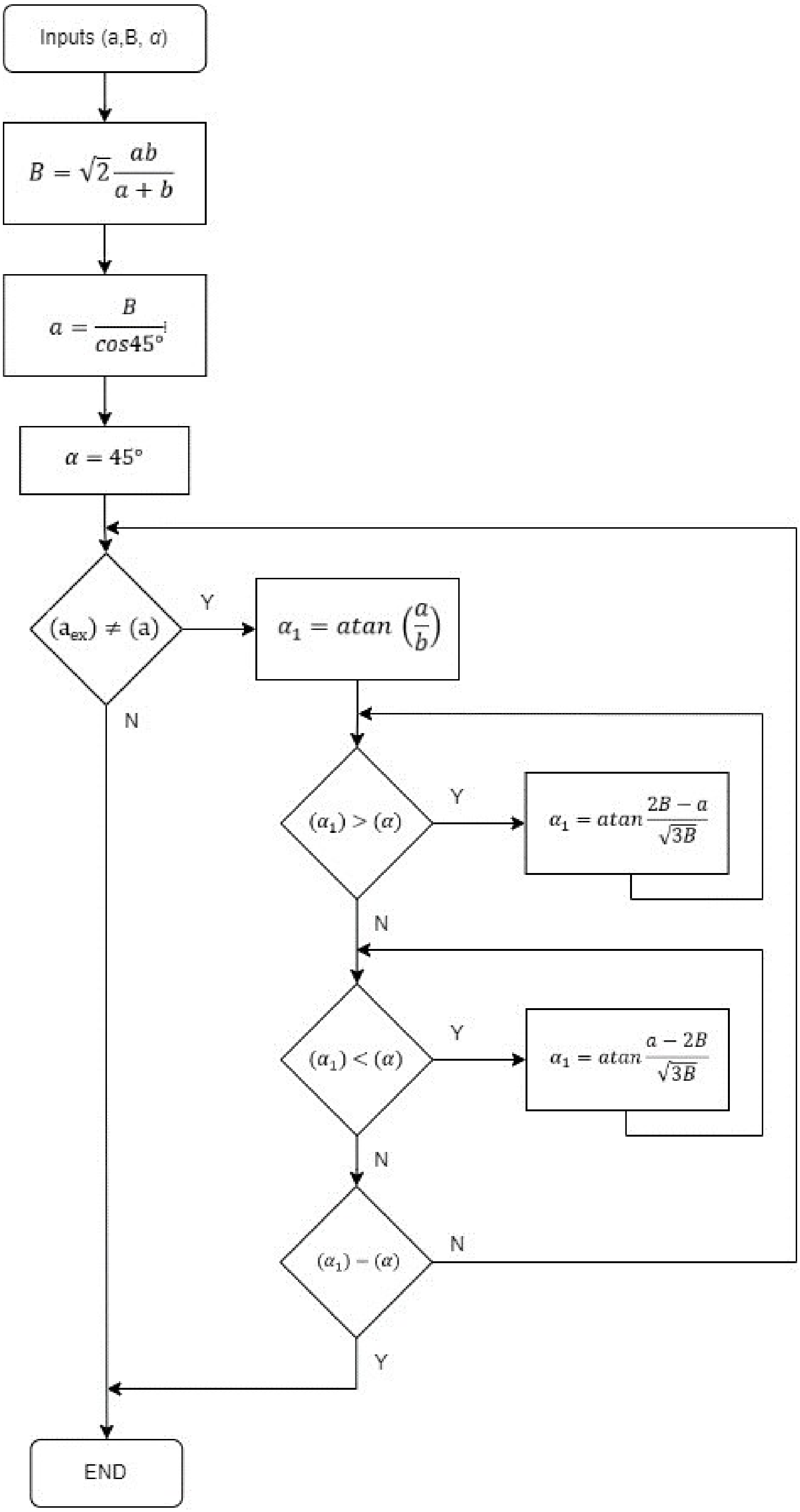


Figure 7: Block diagram of the parallelism calculation algorithm.

Thus, the work algorithm is as follows. The mobile robotic platform distance to the obstacle and parallelism calculation is carried out from the measured distances of the isosceles right triangle as in figure 6 b). The distance to the object is determined by the bisector estimating. After finding the bisector the experimental dynamic distance in motion to the object and compare it with the real one. If the experimental distance differs from the real one, then the angle α_1 is determined, if $(\alpha_1) > (\alpha)$ then the angle α_1 is determined. If $(\alpha_1) < (\alpha)$ then we determine the angle α_1. When performing all angle determinations, we compare it with the actual angle to object 45°.

The proposed method of measuring distance and parallelism allows to carry out these operations using a small number of sensors, and also allows to make measurements quite accurately based on the basic characteristics of the sensors. The obtained results are useful for the tasks described in [22,23].

Conclusions

A method of determining dynamic obstacles and a method of robotic platform positioning are proposed considered. The approach proposed is established on the laser distance sensors using for the obstacles detecting and positioning a mobile robotic platform in the space. The calculation of non-contact sensors of the distance of the robotic platform, the scheme of placement of sensors and positioning of the mobile robotic platform in space are described.

The method proposed of using laser distance sensors for determining obstacles and positioning a mobile robotic platform in the space allows to determine the coordinates of the location and course of a mobile robotic platform using infrared distance sensors. The above functions in various combinations can be performed by a positioning system, an inertial navigation system capable of calculating the distance to obstacles, calculating the parallelism of the mobile robotic platform to the plane, and calculating the course of the mobile robotic platform.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] A. Al-Kaff, F. García, D. Martín, A. De La Escalera, and J. M. Armingol, Obstacle detection and avoidance system based on monocular camera and size expansion algorithm for UAVs, 2017.

[2] V. A. Bhavesh, Comparison of various obstacle avoidance algorithms. Int. J. Eng. Res. Technol.4, (2015) 629–632.

[3] M. Coppola, K. N. McGuire, K. Y. W. Scheper, G. C. H. E. de Croon, On-board communication-based relative localization for collision avoidance in micro air vehicle teams, Autonomous Robots, 2018.

[4] J. P. Fuentes, D. Maravall, J. de Lope, Entropy-based search combined with a dual feedforward/feedback controller for landmark search and detection for the navigation of a UAV using visual topological maps, in Proc. Robot 2013: First Iberian Robotics Conference. Springer Series on Advances in Intelligent Systems and Computing, Heidelberg: Springer, volume 233, 2014, pp. 65–76.

[5] I. Noreen, A. Khan, Z. Habib, Optimal path planning using RRT* based approaches: a survey and future directions. Int. J. Adv. Comput. Sci., 2016, pp. 97–107.

[6] J. M. Lopez-Guede, M. Graña, Neural modeling of hose dynamics to speedup reinforcement learning experiments, in Bioinspired Computation in Artificial Systems (Elche), 2015, pp. 311–319.

[7] B. Gfeller, M. Mihalak, S. Suri, E. Vicari, P. Widmayer, Counting targets with mobile sensors in an unknown environment. In ALGOSENSORS, 2007.

[8] L. Keming, A comprehensive review of bug algorithms in path planning, Applied and Computational Engineering, 33, 2024, pp. 259-265. doi:10.54254/2755-2721/33/20230278.
[9] Automaticaddison, The Bug2 Algorithm for Robot Motion Planning, 2020. URL: https://automaticaddison.com/the-bug2-algorithm-for-robot-motion-planning/.
[10] K. N. McGuire, G. C. de Croon, K. Tuyls, A comparative study of bug algorithms for robot navigation. Robotics and Autonomous Systems, 121, 2019, 103261. doi:10.1016/j.robot.2019.103261.
[11] Martsenyuk, V., Klos-Witkowska, A., & Sverstiuk, A. (2020). Stability Investigation of Biosensor Model Based on Finite Lattice Difference Equations. In Springer Proceedings in Mathematics & Statistics, 297–321. Springer International Publishing. https://doi.org/10.1007/978-3-030-35502-9_13
[12] Martsenyuk, V., Sverstiuk, A., Gvozdetska, I. (2019). Using Differential Equations with Time Delay on a Hexagonal Lattice for Modeling Immunosensors. In Cybernetics and Systems Analysis (Vol. 55, Issue 4, pp. 625–637). Springer Science and Business Media LLC. https://doi.org/10.1007/s10559-019-00171-2.
[13] Martsenyuk V., Andrushchak I., Zinko, P., Sverstiuk, A. (2018). On Application of Latticed Differential Equations with a Delay for Immunosensor Modeling. In Journal of Automation and Information Sciences, Vol. 50, Issue 6, 55–65. Begell House. https://doi.org/10.1615/jautomatinfscien.v50.i6.50.
[14] Martsenyuk, V., Soldatkin, O., Klos-Witkowska, A., Sverstiuk, A., & Berketa, K. (2024). Operational stability study of lactate biosensors: modeling, parameter identification, and stability analysis. In Frontiers in Bioengineering and Biotechnology (Vol. 12). Frontiers Media SA. https://doi.org/10.3389/fbioe.2024.1385459.
[15] Saiapina, O. Y., Berketa, K., Sverstiuk, A. S., Fayura, L., Sibirny, A. A., Dzyadevych, S., & Soldatkin, O. O. (2024). Adaptation of Conductometric Monoenzyme Biosensor for Rapid Quantitative Analysis of L-arginine in Dietary Supplements. Sensors, Vol. 24, Issue 14, 4672. https://doi.org/10.3390/s24144672.
[16] M. Palamar, M. Yavorska, M. Strembitskyi, I. Zelinskyi, The device for remote measurements of geometric dimensions and positions 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), September 22-23, 2017. Bucharest, Romania, 2017, volume 2, pp. 524–527.
[17] I. Zelinskyi, M. Palamar, M. Yavorska, The optical device for coordinate measurement, XIII International Scientific Conference Coordinate Measuring Technique, Bielsko-Biala, 11th – 13th of April, 2018.
[18] M. Palamar, M. Yavorska, M. Strembitskyi, V. Strembitskyi, Selection of the efficient video data processing strategy based on the analysis of statistical digital images characteristics, Scientific Journal of the Ternopil National Technical University, 2018, pp. 107-114.
[19] Nakonechnyi A., Martsenyuk V., Sverstiuk A., Arkhypova V., Dzyadevych S. (2020). Investigation of the mathematical model of the biosensor for the measurement of α-chaconine based on the impulsive differential system. CEUR Workshop Proceedings, 2762, 209 – 217.
[20] Martsenyuk V., Bahrii-Zaiats O., Sverstiuk A., Dzyadevych S., Shelestovskyi B. (2023). Mathematical and Computer Simulation of the Response of a Potentiometric Biosensor for the Determination of α-chaconine. CEUR Workshop Proceedings, 3468, 1 – 11.
[21] W. G. Aguilar, V. P. Casaliglla, J. L. Polit, Obstacle avoidance based-visual navigation for micro aerial vehicles, Electronics, 6(10), 2017.
[22] Maruschak, P. O., Panin, S. V., Zakiev, I. M., Poltaranin, M. A., Sotnikov, A. L. (2016). Scale levels of damage to the raceway of a spherical roller bearing. In Engineering Failure Analysis (Vol. 59, pp. 69–78). https://doi.org/10.1016/j.engfailanal.2015.11.019
[23] Konovalenko, I.; Maruschak, P.; Brevus, V.; Prentkovskis, O. Recognition of Scratches and Abrasions on Metal Surfaces Using a Classifier Based on a Convolutional Neural Network. Metals 2021, 11, 549.