UDC 004.896
DOI: 10.15587/1729-4061.2023.277840

DEVELOPMENT OF CONTROL SOFTWARE FOR SELF-ORGANIZING INTELLIGENT MOBILE ROBOTS

Daulet Toibazarov
Associate Professor
Department of Education and Science
National Defense University named after the First President of the Republic of Kazakhstan
Turan ave., 72, Astana, Republic of Kazakhstan, 010000

Gani Baiseitov
Candidate of Technical Sciences, Colonel
Department of Education and Science
LLP "Research & Development Center "Kazakhstan Engineering"
M. Auezova str., 2, Astana, Republic of Kazakhstan, 010000

Abzal Kyzyrkanov
Master
Department of Computer Engineering
Astana IT University
Rysqulbekov str., 8/2, Astana, Republic of Kazakhstan, 010000

Shadi Aljawarneh
Professor
Department of Software Engineering
Jordan University of Science and Technology
Irbid, Jordan, 3030

Sabyrzhan Atanov
Corresponding author
Professor
Department of Computer and Software Engineering
L. N. Gumilyov Eurasian National University
Satpayev str., 2, Astana, Republic of Kazakhstan, 010008
E-mail: atanov5@mail.ru

Maintaining a specific geometric formation during the movement is crucial for multiagent systems of mobile robots in various applications. Proper coordination can lead to reduced system costs, increased reliability and efficiency, and system adaptability and flexibility.

This research proposes a novel movement coordination method for self-governing multiagent systems of intelligent mobile robots. The proposed method uses a leader-follower technique with a virtual leader to maintain a specific geometric structure. Additionally, the epsilon greedy algorithm is utilized to avoid loops. To reduce power consumption, it is proposed to turn on only a few robots' lidars at a time. They could drive all the robots in the group, allowing them to reach the goal without colliding with obstacles.

Experiments on a complex map with nine robots were conducted to test the method's effectiveness. The success rate of the swarm reaching the target position and the number of steps needed were evaluated. Testing varied angular velocities of 1 to 20 degrees and linear velocities of 0.1 to 5.5 m/s. Results show the method effectively guides the robots without collisions.

This method enables a group of self-governing multiagent systems of intelligent mobile robots to maintain a desired formation while avoiding obstacles and reducing power consumption. The results of the experimental study demonstrate the method's potential to be implemented in real-world missions and traffic management systems to increase efficiency and reduce costs.

The proposed method can be utilized in military missions and traffic management systems, where maintaining a specific geometric formation is crucial. The method's ability to avoid obstacles and reduce power consumption can also lead to reduced costs and increased efficiency

Keywords: multiagent system, mobile robots, formation control, pattern formation

Received date 06.02.2023
Accepted date 16.04.2023
Published date 28.04.2023

How to Cite: *Toibazarov, D., Baiseitov, G., Kyzyrkanov, A., Aljawarneh, S., Atanov, S. (2023). Development of control software for self-organizing intelligent mobile robots. Eastern-European Journal of Enterprise Technologies, 2 (9 (122)), 46–58. doi: https://doi.org/10.15587/1729-4061.2023.277840*

1. Introduction

The use of autonomous multiagent systems of intelligent mobile robots has become increasingly prevalent in recent years, with a wide range of applications, from military missions to traffic management systems. However, proper coordination among these robots is crucial for efficient system operation, which can lead to reduced costs, improved adaptability, and increased flexibility. A particular challenge arises when the robots need to maintain a specific geometric formation while following a designated path and avoiding obstacles.

The relevance of this scientific topic is that the development of self-governing multiagent systems of intelligent mobile robots has the potential to increase efficiency and reduce costs in various applications.

Therefore, research devoted to the development of coordination algorithms for autonomous multiagent systems of intelligent mobile robots is relevant and necessary, as it can lead to increased efficiency, cost savings, and improved adaptability and flexibility of the system.

2. Literature review and problem statement

Nowadays, the field of robotics is advancing rapidly, with numerous strategies proposed throughout its history to enhance robot capabilities and overcome specific challenges in the industry. For example, work [1] presents a Bluetooth beacon-based positioning system that utilizes filtering and data smoothing to enhance accuracy, which could prove useful in mobile robotics applications requiring precise indoor localization. However, this system is limited to short-range communication, thereby restricting its usage in larger environments. In another study [2], a new spectral domain approach is presented to address the problem of radiation from a vertical short dipole above flat lossy ground. This approach yields closed-form analytical solutions for the received electromagnetic field vectors, which may prove useful in mobile robotics applications. However, the scope of this study is limited to ground-based objects and does not consider the unique characteristics of radio control in airspace environments, which may have varying implications and outcomes. Additionally, work [3] proposes a multi-agent intrusion detection system for a fleet of UAVs, which addresses the security issue by utilizing autonomous, communicating, learning, and intelligent agents that can detect known and unknown cyber-attacks in real-time. However, the proposed system is limited in its ability to detect sophisticated attacks and lacks a comprehensive evaluation of its performance in real-world scenarios.

As robots have evolved, various approaches have been proposed to enhance their performance, including the use of multiple autonomous robots or multiagent systems to complete complex tasks.

One of the most important issues in multi-agent systems is coordinating the motion of multiple agents working towards a common goal with limited knowledge, resources, and sensory and computational capabilities. To avoid collisions with each other or obstacles and achieve the goal more quickly, effective coordination is crucial. Lowering system costs, increasing reliability and efficiency, and creating a more flexible system are benefits of effective coordination.

Formation control is a control method that aims to achieve specific formations by a group of robots and allows them to move without collisions. To coordinate the motion of the robots, a particular geometric structure can be formed, maintaining the distance between the agents in the group. Many works have been done in this area. For example, Formation control for multiple mobile agents using relative positions and local landmarks is proposed in [4]. However, this paper does not consider cases when one or more agents in the system are knocked down. Similarly, the work in [5] proposes an optimal control method for the formation of multi-agent systems with energy constraints based on distance using the theory of rigid graphs and the method of the Riccati equation. But it does not address issues related to the scalability and reliability of the system. Another approach is a behavioral approach, as described in [6], which coordinates the motion of multi-agent robotic systems to maintain a certain geometric pattern. This paper does not consider how the system must react if an obstacle appears in the way.

The leader-follower method is often used in formation control as it is simple to implement and efficient. For instance, a fuzzy adaptive optimized leader-follower formation control strategy for second-order stochastic multi-agent systems using actor-critic architecture and Lyapunov stability theory for control design is presented in [7]. Likewise, [8] proposes a decentralized leader-follower approach for the formation control of unmanned ground vehicles subject to time-varying external disturbances and parametric uncertainties. However, the stability of the system is at risk if any robot from a multi-agent system is assigned as a leader, as seen in [7, 8]. One possible solution to this issue is defining a virtual leader and maintaining a distance from it, as suggested in [9], where the position of the virtual leader can be marked in the center of a maintained geometric pattern.

Another major problem in group robotics is energy consumption, which is exacerbated in group robotics as each robot has its own CPU for calculations and sensors. In contrast, in the case of one robot, one decision-making center and one sensor are sufficient. Task allocation algorithms have been proposed to minimize energy consumption, such as the algorithm presented in [10]. However, estimating the energy cost of individual tasks accurately can be challenging, leading to sub-optimal task allocation and reduced overall energy efficiency. Furthermore, optimization using genetic algorithms can be computationally expensive and not suitable for large groups of robots or more complex tasks. In our algorithm, the motion of the multi-agent system turns on the lidars only as necessary, reducing energy consumption, whereas all the above-mentioned formation control algorithms [4–10] require turning on the sensors of all robots during motion.

Deadlock is another problem in multi-agent systems that arises when robots arrive at the same location while searching for the optimal path and making decisions. If the external environment does not change, robots moving by the same algorithm and making the same decisions again and again can get into a loop, causing a deadlock. Several solutions have been proposed to solve this problem, including the ORCCAD control architecture proposed in [11], which includes formal verification and synchronous programming. However, implementing this architecture can be challenging. The genetic algorithm presented in [12] is also costly. In [13], a method is proposed to address the deadlock problem by identifying necessary sub-targets on a map and generating a collision-free path using a hybrid scheduler based on the potential field method and the Voronoi diagram. However, this method is reliant on accurate and current environmental maps, which may not be available in practical applications.

To overcome this limitation, this paper proposes the use of the epsilon greedy approach, which is commonly used in reinforcement learning [14] to balance exploration and exploitation. It is also used in optimization [15] and game theory. The proposed algorithm utilizes a leader-follower technique with a virtual leader to maintain a desired geometric structure, and the epsilon greedy algorithm is employed to avoid loops. To conserve power, only a few robots' lidars are activated at a time to navigate the group towards the goal while avoiding obstacles.

In conclusion, the literature review has highlighted the current state of research on multiagent systems of intelligent mobile robots and the challenges associated with maintaining a specific geometric formation while following a designated path. While there are many studies that suggest potential solutions to these challenges, there is still a need for an integrated approach that can ensure system scalability, reliability, accurate estimation of individual task energy consumption, and avoidance of infinite loops during motion.

Therefore, through a critical literature review in the field of multi-agent control of mobile robot systems, the following unresolved problems have been identified:

– in traditional systems, the use of a virtual leader works effectively when moving in open spaces, but in complex environments – such as a forest with many trees or mountains with high hills, or in urban environments with many buildings – it cannot ensure goal achievement and preservation of the system topology;

– the monitoring of obstacles on both the right and left sides becomes more relevant when moving in complex environments, e.g., in a city with many buildings. However, the active use of lidar rangefinders by all agents in multi-agent systems often leads to control failures due to false obstacle detection;

– one of the most significant problems when moving multi-agent systems in autonomous mode is the high energy consumption by scanners, rangefinders, and sensors.

3. The aim and objectives of the study

The aim of this study is to develop a comprehensive method for enabling autonomous multiagent systems of intelligent mobile robots to maintain a specific geometric formation while following a designated path.

To achieve this aim, the following objectives are accomplished:

– to design a leader-follower technique with a virtual leader;

– to develop a method for selecting observer robots that activate lidars during movement and calculating the turn direction depending on the lidar data;

– to apply the epsilon greedy algorithm to avoid infinite loops during motion;

– to design and evaluate the method by integrating all of the above algorithms and conducting simulation experiments on a complex and light map to demonstrate its effectiveness.

4. Materials and methods

4. 1. Object and hypothesis of the study

This study aims to address this challenge by developing a movement coordination method that enables autonomous multiagent systems of intelligent mobile robots to maintain a specific geometric formation while avoiding obstacles and reducing power consumption.

To address this challenge, a new approach is proposed for controlling a multi-agent system of robots to move towards a target location while maintaining a desired formation and avoiding obstacles. The technique used is based on leader-follower mechanism with a virtual leader. The method initiates movement towards the destination, where each robot updates its velocity regularly. The velocity vector is expressed in terms of a pair (v, w), where v represents the linear velocity in meters per second, and w represents the angle of rotation. The method only changes the direction of motion while the linear velocity remains constant. The first step is to determine the position of the virtual leader, and then observer agents with their lidars turned on are identified based on the direction of motion. Other robots turn off their lidars to conserve power. Three observers with lidars are required: one leading robot in the center acts as the middle observer, and two robots positioned on the left and right sides of the multi agent system, respectively act as left and right observers. To avoid collisions during direction changes, the side lidars are slightly directed to either side. The selection of these observers depends on the direction of the multi-agent system. Based on their detection, the chosen observers activate their lidar and determine the direction of movement, i.e., straight, left, or right, and then calculate the angular velocity. If the direction is forward, the angular velocity remains the same, else the angular velocity is adjusted to the selected direction.

4. 2. Leader-follower approach

The method uses a leader-follower approach to coordinate the movements of the multiagent system, and is designed to optimize the use of lidars. By reducing their usage, the method helps to increase the lifespan of the lidars and reduce maintenance costs.

The leader-follower approach is based on the principle that follower agents strive to maintain a specific distance from the leader at all times. This is achieved by representing the leader and followers as points on a surface, with their respective poses used to calculate the distance between them. Specifically, assuming the follower pose is (x_F, y_F) and the leader pose is (x_L, y_L), the distance is computed as follows:

$$D=\sqrt{(x_F-x_L)^2+(y_F-y_L)^2}. \quad (1)$$

The computed distance can be interpreted as the range that the follower aims to maintain, as illustrated in Fig. 1.

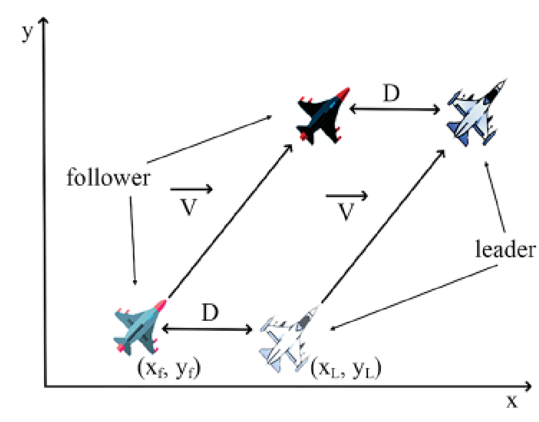

Fig. 1. Leader-Follower approach

Assigning one agent from a multi agent system as the leader can pose a significant risk to the stability of the entire group, as a failure of the leader can render the multi agent system inoperable. To address this issue, a virtual leader can be designated and a distance maintained in relation to it. The virtual leader's position can be easily marked at the center of the multi agent system.

4. 3. Turning lidars on and off

To address the challenge of reducing power consumption, the proposed algorithm for movement of multiagent system in this article utilizes lidars only as necessary. So, mobile robots can turn off their lidars whenever possible and instead rely on their neighbors' movements to orient themselves. For instance, agents moving behind or in the middle can be guided by the movements of their neighbors ahead of them. By turning on the lidars only when required, power consumption is minimized while maintaining effective obstacle detection.

4. 4. Epsilon greedy algorithm

Another movement problem of multi-agent systems is deadlock. While moving to the target and bypassing the obstacle and making decisions on the spot, depending on the external environment, robots can arrive at the same location. And if the external environment does not change then the robots moving by the same algorithm and making the same decisions again and again the robots can get into a loop. To cope with this problem an approach called epsilon greedy or shortly e-greedy has been used. This algorithm can be used to add randomness to the robots' movements, which can help them avoid following the same way over and over. This approach is widely used in machine learning problems so that the trained system is not too much adapted to local optima and to add to its knowledge more diverse. The essence of the method is that the system chooses an "illogical" solution when making a decision with a certain very small probability equal to epsilon. Usually in machine learning algorithms the epsilon is 5–10 percent. The epsilon value equal to 0.1 percent is chosenwhen an obstacle is detected and 1 percent when no obstacle is detected. When an obstacle is detected, the epsilon is less because near obstacles the risk of collision with these obstacles increases.

4. 5. Simulation setup

The proposed method will be evaluated and validated through simulations and experiments to demonstrate its effectiveness and applicability. The findings of this study can contribute to the advancement of swarm intelligence and robotics technology, and benefit the scientific community and industries related to robotics and control.

To evaluate the effectiveness of the method, a Python simulator was created using the PyGame library to provide a visual representation of the multiagent system motion. The simulation was conducted on two types of maps, one with simple obstacles (Fig. 2) and another with complex obstacles (Fig. 3). Obstacles were represented by black color and the target point by red. In both maps experiments with a multiagent system consisting of nine robots were performed. The robots were depicted in purple and the observers were colored green to track their movement.

In the experiment, different values of linear velocity (v) and angular velocity (w) were tested to see their effects on the number of steps needed to reach the target location and the occurrence of collisions during the movement.

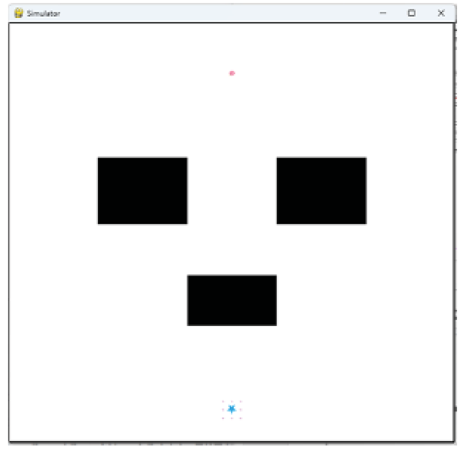

Fig. 2. Initial state of the multi agent system in the simple map

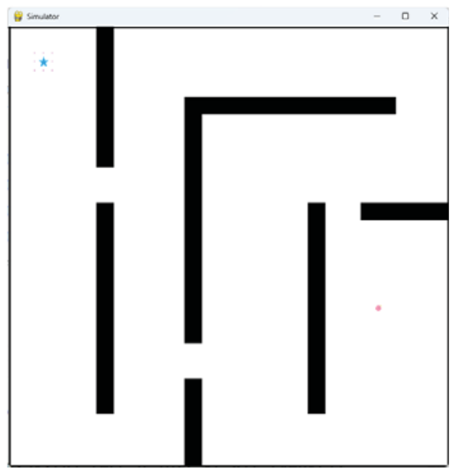

Fig. 3. Initial state of the multi agent system in the complex map

5. Results of the development of a movement coordination method for autonomous multiagent systems of intelligent mobile robots

5. 1. Design of leader-follower technique with a virtual leader

The traditional method of working in a group involves the use of the "leader-follower" technique, in which each member is a physical object. When managing a group of automated systems, it is more convenient to select a virtual leader based on the spatial coordinates of all elements. Using a virtual, rather than physical, leader ensures the system's resilience in the event of any element failure. Additionally, it allows for more accurate coordination of a multi-agent system of robots when moving towards a goal, maintaining structure and avoiding obstacles.

In this study, a mathematical technique is used to calculate the position of the virtual leader. Specifically, the method of calculating the mean values from the field of mathematical statistics is applied. Absolute values, i.e. only positive numbers, are used to calculate the virtual leader. This allows for simplification and acceleration of the calculation using formulas for the arithmetic mean.

This study proposes a method that coordinates a multiagent system of robots to move toward a target location while maintaining a desired formation and avoiding obstacles. The method employs the leader-follower technique with a virtual leader. The system comprises multiple robots that periodically update their velocity vectors, consisting of the linear velocity of motion (v) in meters per second and the rotation angle (w).

At the beginning of the motion, the virtual leader's position is assigned at the center of the desired formation. During the motion, the system updates the position of the virtual leader with a certain frequency. Depending on the virtual leader's position, the robot reviewers are reassigned, and the velocity vector is recalculated to avoid obstacles and maintain the desired formation.

The position of the virtual leader is calculated using the following equation:

$$x_L = \frac{\sum_{i=1}^{N}(x_i - x_{ides})}{N}, \quad (2)$$

$$y_L = \frac{\sum_{i=1}^{N}(y_i - y_{ides})}{N}. \quad (3)$$

Here, N represents the total number of agents present in the multi-agent system. The variables (x_i, y_i) denote the current coordinates of the i-th mobile robot, while (x_{ides}, y_{ides}) represent the desired coordinates of the i-th agent with respect to the position of the virtual leader.

5. 2. Development of a method for selecting observer robots for lidar activation and calculating turn direction depending on lidar data

The observer agents with their lidars turned on are determined based on the direction of motion. To conserve power, all other robots switch off their lidars. Only three observers with lidars are required: the leading robot in the center acts as the middle observer, and two robots positioned on the left and right sides of the multi agent system, respectively act as left and right observers. To avoid collisions during direction changes, the side lidars are slightly directed to either side. The selection of these observers is dependent on the direction of the multi agent system.

By shifting the coordinate center to the virtual leader's position and rotating the X-axis in the direction of the multi agent system (as shown in Fig. 4), it is readily possible to locate the observers. In this new coordinate system, the position of any robot can be easily calculated using the following equation:

$$x_i' = (x_i - x_L)\cos(w) + (y_i - y_L)\sin((w), \quad (4)$$

$$y_i' = (y_i - y_L)\cos(w) - (x_i - x_L)\sin(w). \quad (5)$$

Here (x_i, y_i) is the position of i-th agent, (x_L, y_L) is the position of virtual leader, (x_i', y_i') is the coordinates of the i-th agent in a new coordinate system.

In Fig. 5, an example of how to define observers is illustrated. The mobile robot that is ahead of all in the direction of motion is chosen as the central observer. Specifically, the mobile robot with the maximum x' value in the new coordinate system is selected as the average observer. In case there are multiple mobile robots with the maximum x' value, the one closest to the middle (x-axis) is chosen based on the absolute value of y. As shown in Fig. 5, although the x values of the third and fourth agents are equal, the third agent is selected as the middle agent since its absolute value of x' is smaller.

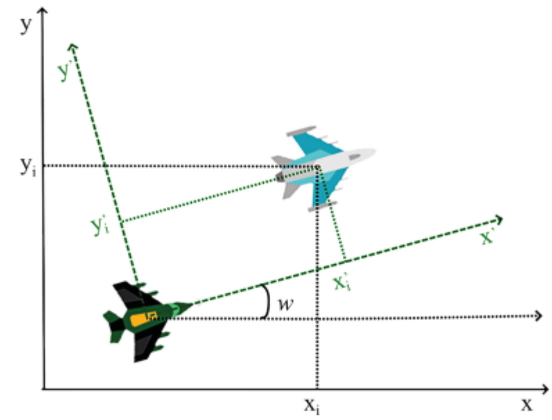

Fig. 4. Transformation to a coordinate system centered on the virtual leader's location and aligned with the multi agent system's motion

The left and right observers are defined similarly in that they choose mobile robots located farthest from the middle (i. e., x-axis). The left observer is chosen based on the mobile robot with the highest value of y, while the right observer is chosen as the mobile robot with the lowest value of y. If multiple mobile robots meet the criteria, the one with the highest value of x' is selected. In Fig. 5, the second mobile robot was chosen over the first because it had a greater x value, despite both having equal and maximum y values. When comparing coordinates, differences less than half the width of the robot are ignored, and two values are considered equal if the difference is less than half the width of the robot. For example, in Fig. 5, the fifth robot was chosen instead of the sixth, even though the sixth had the maximum value of y, because the difference in the value of y was small, and the x' value of the fifth robot was larger.

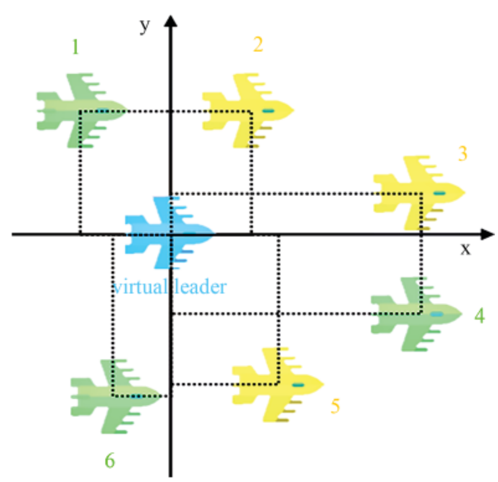

Fig. 5. Choosing observers in a coordinate system with the virtual leader's position as the center (highlighted in yellow)

When one agent meets the criteria for both the middle and left observers, it is chosen as the left (or right) observer, and the next one that meets the criteria is chosen as the middle observer. This situation is shown in Fig. 6, where the fifth agent meets the criteria for both the middle and right observers and is chosen as the right observer. The fourth agent, which is the next one to meet the criteria, is chosen as the middle observer.

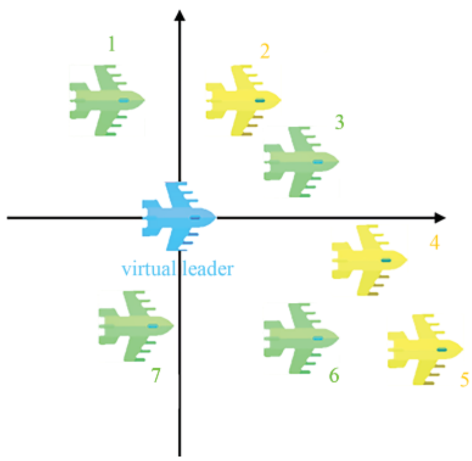

Fig. 6. Selecting an observer in the scenario where one agent meets both the middle and left observer criteria

If there are no other agents remaining in the multi agent system, the last remaining agent will act as the middle observer. If there are two agents remaining, they will act as the left and right observers.

Overall, the method for selecting observer robots for lidar activation and calculating turn direction depending on lidar data involves the following steps:

– determine the observer agents with their lidars turned on based on the direction of motion. To conserve power, all other robots switch off their lidars. Only three observers with lidars are required: the leading robot in the center acts as the middle observer, and two robots positioned on the left and right sides of the multi agent system, respectively act as left and right observers;

– shift the coordinate center to the virtual leader's position and rotate the x-axis in the direction of the multi agent system to locate the observers. In this new coordinate system, the position of any robot can be easily calculated;

– choose the central observer as the mobile robot with the maximum x' value in the new coordinate system. In case there are multiple mobile robots with the maximum x' value, choose the one closest to the middle (x-axis) based on the absolute value of y;

– define the left and right observers similarly. Choose the left observer based on the mobile robot with the highest value of y, and choose the right observer based on the mobile robot with the lowest value of y. If multiple mobile robots meet the criteria, select the one with the highest value of x';

– when one agent meets the criteria for both the middle and left observers, choose it as the left (or right) observer, and choose the next one that meets the criteria as the middle observer;

– if there are no other agents remaining in the multi agent system, the last remaining agent will act as the middle observer. If there are two agents remaining, they will act as the left and right observers.

By following these steps, the observer robots can be efficiently selected, and the turn direction can be accurately calculated based on lidar data, leading to safe and effective multi-robot systems.

The chosen observers activate their lidar and based on their detection, they determine the direction of movement – straight, left, or right. The angular velocity is then calculated based on the selected direction. The calculation process will be explained in subsequent sections, while this section focuses on the direction selection process.

The direction of movement is determined based on the lidar data collected by the selected observers. If obstacles are detected solely by the left lidar or by the left and middle lidars, the direction of movement is to the right. Conversely, if obstacles are detected by the right lidar or by the middle and right lidars, the direction of movement is to the left.

If all lidars detect an obstacle or only the middle lidar detects it, several cases need to be considered:

– if the direction chosen in the previous step was either right or left, the same direction will be chosen in this step;

– if the previous direction was forward, the direction towards the target will be selected;

– if the direction of movement matches the direction towards the target, the multi agent system selects a direction randomly, either right or left, to avoid colliding with obstacles head-on.

If there are no obstacles detected by any of the lidars, the direction towards the target is selected.

The proposed method aims to reduce the energy cost of swarms of robots by selectively turning off lidars. To assess the effectiveness of this approach, the percentage of energy savings resulting from turning off lidars in swarms of different sizes was calculated. As shown in Fig. 7, the impact of turning off lidars to the energy savings in different sizes of robots is demonstrated. Here the x-axis represents the total number of robots in the swarm, while the y-axis represents the percentage of energy savings by turning off lidars in the swarm.

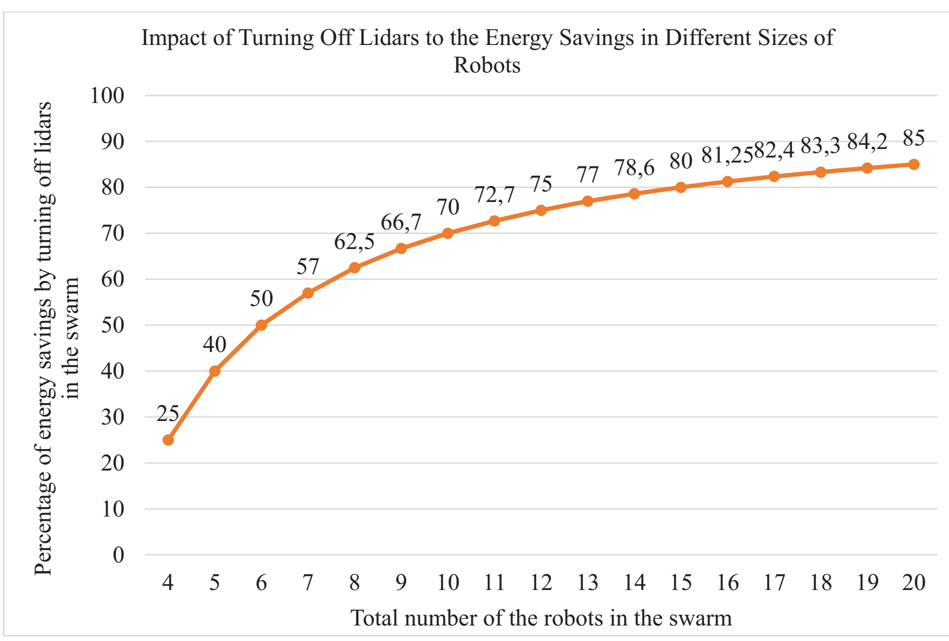

Fig. 7. Impact of turning off lidars to the energy savings in different sizes of robots"

According to Fig. 7 as the number of robots in the swarm increases, the percentage of energy savings also increases. For example, a swarm of 4 robots can save up to 25 % of its energy by selectively turning off lidars, while a swarm of 10 robots can save up to 70 % of its energy. These results confirm that the proposed approach can effectively reduce the energy cost of swarms of robots.

5. 3. Preventing deadlocks in multi-agent systems using the epsilon-greedy method

In this section, let's apply the epsilon-greedy algorithm to prevent deadlocks in multi-agent systems. The task is to avoid agents moving unconsciously in the same trajectory that begins and ends with the same position, as a result of following the algorithm outlined in this article, and to ensure successful completion of tasks in multi-agent systems.

The previous chapter explains how a multi-agent system selects the direction of movement using lidar data collected by selected observers. The direction is determined based on the number and location of obstacles detected, and if no obstacles are detected, the system selects the direction towards the target.

But experimental studies have revealed that a multi-agent system can experience a deadlock, which occurs when agents move unconsciously in the same trajectory that begins and ends with the same position, as a result of following the algorithm outlined in this article. For example, as shown in Fig. 8, the multi-agent system can become deadlocked, and unable to get off.

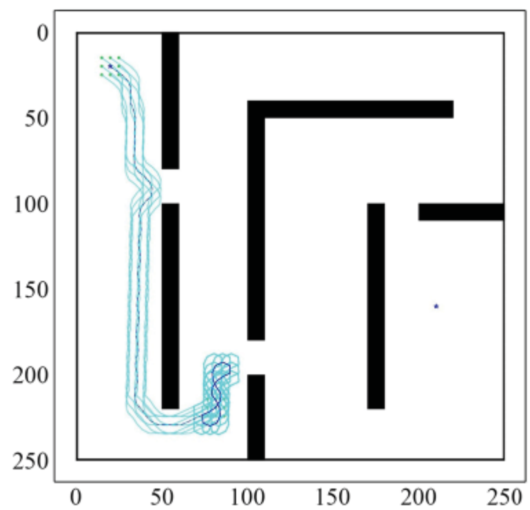

Fig. 8. Multi-agent system path without epsilon-greedy approach and deadlocked

To address this issue, the epsilon-greedy method has been applied, which entails selecting an "irrational" solution when making decisions with a small probability equivalent to epsilon. Typically, in machine learning algorithms, epsilon ranges from 5 % to 10 %.

To avoid the deadlock in the algorithm considered in this article, agents with an epsilon probability choose the opposite direction (that is, if they were moving left, they select right or vice versa), thereby introducing a minor variance in the trajectory. At first glance, the multi-agent system may seem to collide with an obstacle or go in the "wrong direction," but in reality, the trajectory of the system changes insignificantly and is quickly corrected in the subsequent steps. Even if agents collide with obstacles, only one agent will be affected, and that too with a probability significantly smaller than epsilon. A small value of epsilon guarantees that the multi-agent system ultimately emerges from all loops and finally reaches the intended destination.

The magnitude of epsilon is chosen in such a way that it affects the trajectory of the multi-agent system, thereby increasing the probability of a "fatal error." Nevertheless, the larger the value of epsilon, the faster the multi-agent system emerges from the loop. When an obstacle is detected, let's use an epsilon value of 0.1 %, and when there is no obstacle, we used 1 %. The epsilon is lower when obstacles are present since the risk of collision with them is greater. As shown in Fig. 9, the multi-agent system successfully reaches the target, having the same configuration as the system in Fig. 8, but it gets off the deadlock without losing any robots.

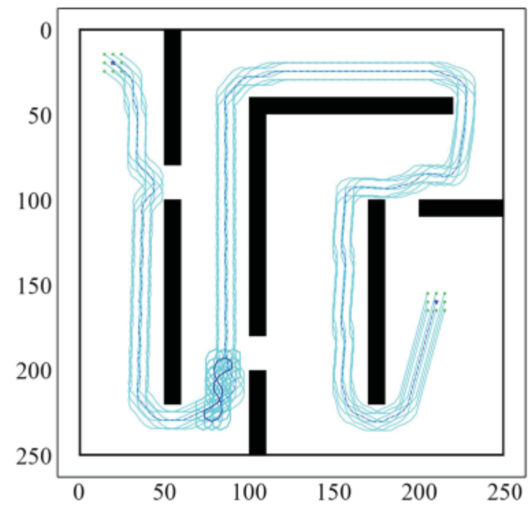

Fig. 9. Illustration of the multi-agent system successfully reaching the target using the epsilon-greedy approach to avoid deadlock

The article discusses the issue of deadlocks in multi-agent systems, where agents move in the same trajectory and become stuck. To address this issue, the epsilon-greedy method is employed, which involves selecting an irrational solution with a small probability. By introducing a small probability of selecting an opposite direction, the multi-agent system successfully avoids getting stuck in a loop and reaches the target without losing any robots. The effectiveness of the epsilon-greedy method in preventing deadlocks is demonstrated through experimental studies, which shows significant improvement in the performance of multi-agent systems. Therefore, the epsilon-greedy method can ensure the successful completion of tasks in multi-agent systems.

In conclusion, the epsilon-greedy method can be applied to prevent deadlocks in the multi-agent system by randomly selecting a non-optimal action with a probability of epsilon, encouraging exploration, and selecting the optimal action with a probability of epsilon, ensuring convergence towards the optimal solution.

5. 4. Design and evaluation of the movement coordination method

The article outlines a method for controlling the movement of a swarm of robots towards a desired destination. Each robot updates its velocity at regular intervals, with the velocity vector of the motion represented by a pair (v, w), where v is the linear velocity in meters per second, and w is the angle of rotation. The direction of motion changes while the linear velocity remains constant. To conserve power, lidars are turned off for all robots except for the observer agents. The position of the virtual leader is computed, and three observer agents with their lidars activated are determined based on the direction of motion. These agents detect the direction of movement and calculate the angular velocity. Agents with epsilon probability choose the opposite direction to avoid live lock and achieve a slight diversity of trajectory.

The velocity vector for motion consists of two components: linear velocity (v) and angular velocity (w), with the linear velocity remaining constant while the angular velocity is calculated based on the selected direction of motion. The angular velocity is adjusted depending on the direction of the previous step, with a specified value manually set. To evaluate the effectiveness of the method, a Python simulator was created and tested on two types of maps. The study aimed to observe the impact of the values of linear velocity (v) and the angle (w) that changes the angular velocity on the number of steps required to reach the target and the number of collisions that occurred during the motion. A flowchart of the method is illustrated in the Fig. 10.

The results of the experiment are displayed in Fig. 11, 12. Fig. 11 shows the variation of the number of steps taken by the swarm to reach the target point. Instances where the swarm was unable to reach the target point are depicted as gaps in the graph.

Fig. 12 illustrates the count of agents that collided during the movement. Whenever a robot collides with an obstacle, it is considered to be hit and stops moving.

Additionally, Fig. 13, 14 depict the trajectory of the multi-agent system in simple and complex maps, respectively, while varying the angular velocity by 1 degree (angle w) and the linear velocity by different values.

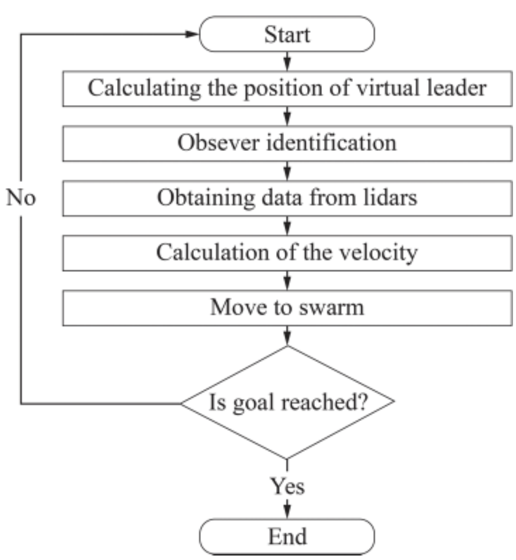

Fig. 10. Flowchart of the method

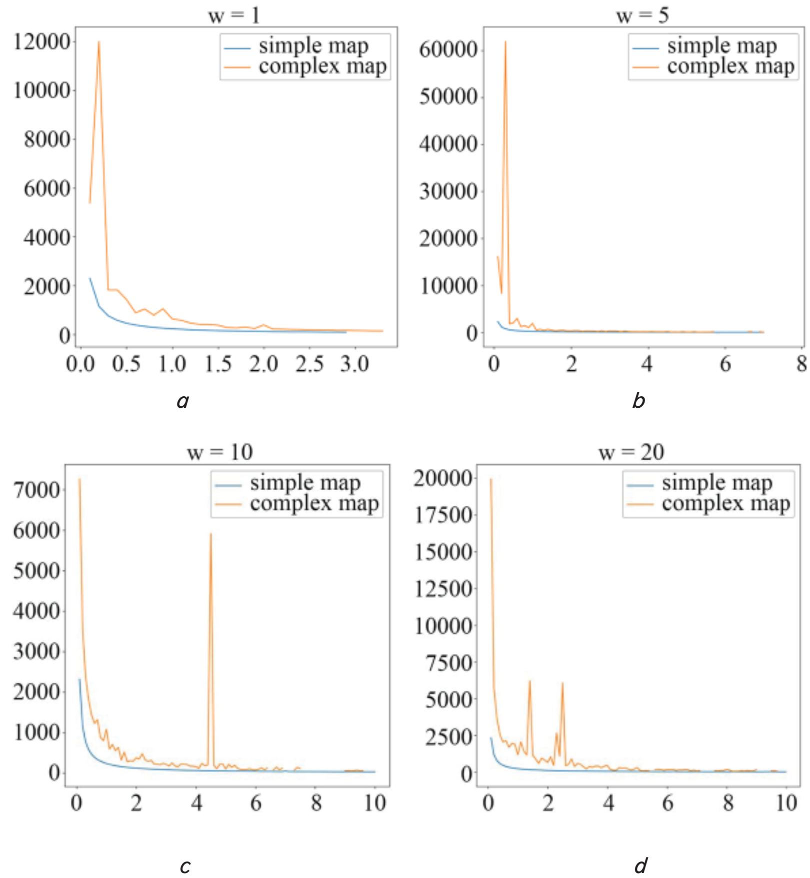

Fig. 11. Change in the number of steps to reach the goal for different angles of angular velocity change: *a* – w=1 degree; *b* – w=5 degrees; *c* – w=10 degrees; *d* – w=20 degrees

In conclusion, the proposed method effectively controls the movement of a swarm of robots towards a desired destination while avoiding collisions. The experimental results demonstrate the impact of the values of linear velocity and the angle that changes the angular velocity on the performance of the swarm. The movement coordination method can be summarized by the following steps:

– each robot updates its velocity at regular intervals, with the velocity vector of the motion represented by a pair (v, w), where v is the linear velocity in meters per second, and w is the angle of rotation;

– lidars are turned off for all robots except for the observer agents to conserve power;

– the position of the virtual leader is computed, and three observer agents with their lidars activated are determined based on the direction of motion;

– the observer agents detect the direction of movement and calculate the angular velocity;

– agents with epsilon probability choose the opposite direction to avoid live lock and achieve a slight diversity of trajectory;

– the velocity vector for motion consists of two components: linear velocity (v) and angular velocity (w), with the linear velocity remaining constant while the angular velocity is calculated based on the selected direction of motion;

– the angular velocity is adjusted depending on the direction of the previous step, with a specified value manually set.

Overall, this method can be a useful tool for controlling the movement of swarm robots in various applications.

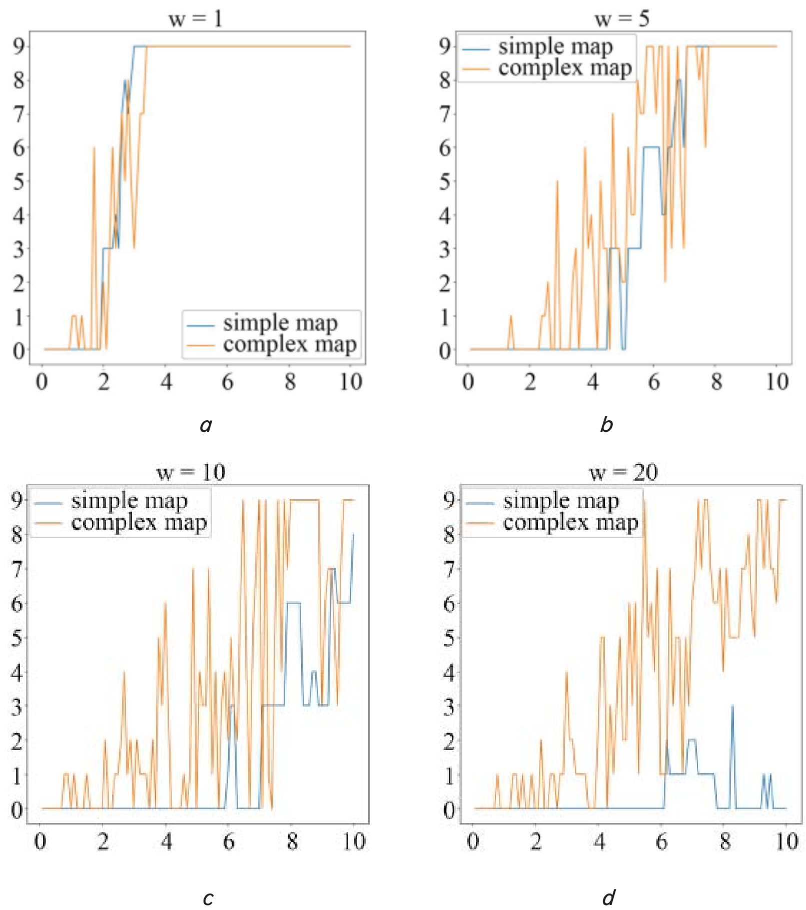

Fig. 12. Change in the number of colliding agents with varying angles of angular velocity: *a* – *w*=1 degree; *b* – *w*=5 degrees; *c* – *w*=10 degrees; *d* – *w*=20 degrees

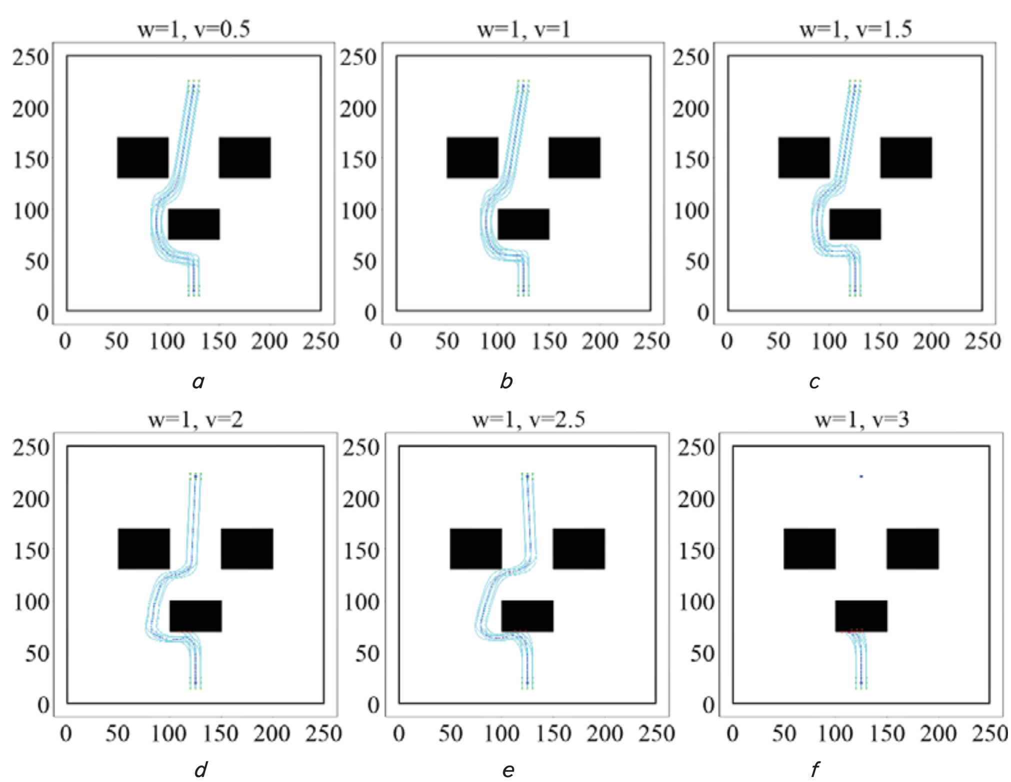

Fig. 13. Path followed by the multi-agent system in simple map with changing angular and linear velocities: *a* – *v*=0.5 m/s, *b* – *v*=1 m/s, *c* – *v*=1.5 m/s, *d* – *v*=2 m/s, *e* – *v*=2.5 m/s, *f* – *v*=3 m/s

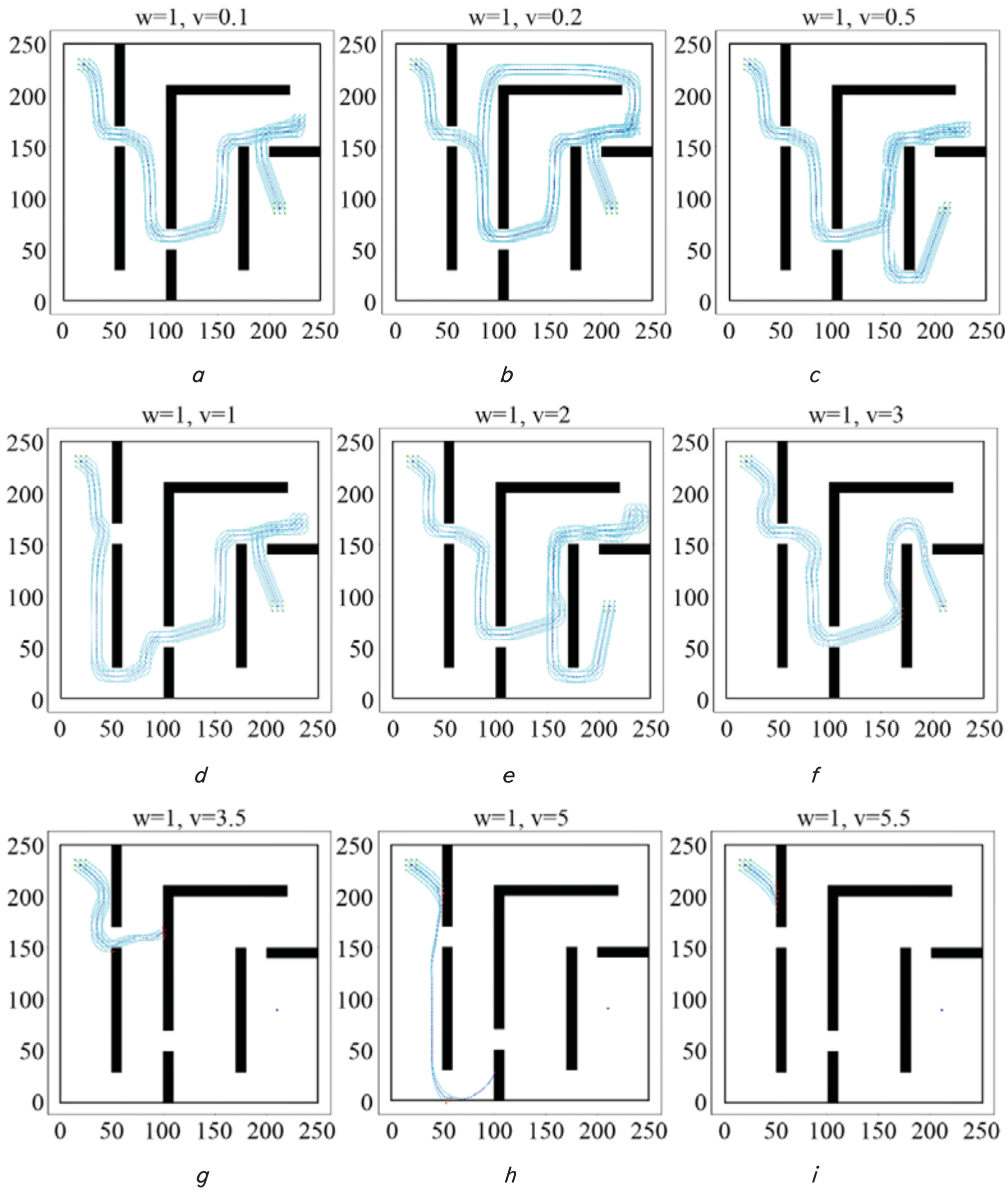

Fig. 14. Path followed by the multi-agent system in complex map with changing angular and linear velocities: *a* – *v*=0.1 m/s, *b* – *v*=0.2 m/s, *c* – *v*=0.5 m/s, *d* – *v*=1 m/s, *e* – *v*=2 m/s, *f* – *v*=3 m/s, *g* – *v*=3.5 m/s, *h* – *v*=5 m/s, *i* – *v*=5.5 m/s

6. Discussion of experimental results

In this work, a method for controlling the movement of a swarm of robots towards a desired destination is presented. The method updates the velocity of each robot at regular intervals, with the velocity vector of the motion represented by a pair (v, w), where v is the linear velocity and w is the angle of rotation. Lidars are turned off for all robots except for the observer agents to conserve power. The position of the virtual leader is computed, and three observer agents with their lidars activated are determined based on the direction of motion. The observer agents detect the direction of movement and calculate the angular velocity. The velocity vector for motion consists of two components: linear velocity (v) and angular velocity (w), with the linear velocity remaining constant while the angular velocity is calculated based on the selected direction of motion. The angular velocity is adjusted depending on the direction of the previous step, with a specified value manually set.

The effectiveness of the method is evaluated using a Python simulator on two types of maps. The experimental results are depicted in Fig. 11, 12. Fig. 11 shows the variation of the number of steps taken by the swarm to reach the target point for different angles of angular velocity change. The gaps in the graph represent instances where the swarm was unable to reach the target point. Fig. 12 illustrates the count of agents that collided during the movement. The count of agents that collided with obstacles is plotted against varying angles of angular velocity.

The relationship between linear velocity and the number of steps to reach the goal is clear: as the linear velocity increases, the number of steps to reach the goal decreases, resulting in a faster arrival at the target position.

The trajectory of the multi agent system also has an impact on the number of steps required to reach the target, and even at a lower speed, the optimal path can lead to a faster arrival at the target position. The variations observed in the graphs of the Fig. 10 indicate that the multi agent system follows different trajectories at different values of linear velocity v and angle w.

The maps have been designed in a way that the multi agent systems can reach the target through multiple routes. However, the optimal paths in both maps are narrow and have a higher chance of encountering obstacles.

According to the Fig. 12, *a* at low values of the angle w, specifically w=1 degree, the robots in the multi agent system tend to collide with obstacles frequently. They are able to reach the target position successfully only when the linear velocity is set to a low value.

According to the Fig. 12, *a* for the light map, the multi agent system can reach the target completely at values of linear velocity v less than 2 m/s. In the range of linear velocity values from 2 to less than 3 m/s, the number of colliding agents is unstable, while at values greater than or equal to 3 m/s, all agents in the multi agent system collide. According to the Fig. 11, *a* the number of steps required to reach the target decreases gradually as the linear velocity increases. The robots move along a similar path for all values of linear velocity. However, at values of linear velocity greater than or equal to 3 m/s, the robots collide with the first obstacle before being able to turn away. Fig. 13 illustrates the trajectory of the multi-agent system in a simple map while the angular velocity changes by 1 degree (angle w) and the linear velocity takes different values.

According to the Fig. 12, *a* for the complex map, the results show that for linear velocity values less than 1 m/s, all robots in the multi agent system can reach the target position. However, as the linear velocity increases between 1 m/s and 3.4 m/s, the number of robots reaching the target gradually decreases with fluctuation. For values greater than 3.3 m/s, the robots cannot reach the target, and at a linear velocity of 0.2 m/s, the multi agent system enters a loop and after making several loops finally reaches the goal but with a large number of steps (Fig. 11, *a*). The trajectory of the multi agent system varies at different values of linear speed, and at values greater than 3.3 m/s, all robots in the multi agent system collide with the obstacle. Moreover, for values greater than 5.2 m/s, robots collide with the first obstacle without having enough time to turn away. Fig. 14 shows the path followed by the multi-agent system in a complex map while the angular velocity changes by 1 degree (angle w) and linear velocity (v) takes different values.

Increasing the value of w to 5 degrees doesn't change the overall pattern much according to the Fig. 12, *b*, but the multi agent system becomes more robust to collisions and faster at lower linear velocities in reaching the target which is clearly illustrated in Fig. 11, *b*. This is because increasing the value of w allows the multi agent system to turn away from obstacles more quickly and adjust its direction more rapidly when traversing gaps and bypassing obstacles.

Increasing the value of w (up to 10, 20 degrees) in the light map improves the stability of the multi agent system as it becomes less prone to losses. At w=10 degrees, only at high values of linear speed, the number of knocked-down robots exceeds 3 (Fig. 12, *c*), while at 20 degrees, it doesn't exceed 3 at all (Fig. 12, *d*). This suggests that the multi agent system becomes more flexible while moving at these values of w. However, this increase also has its disadvantages, which are demonstrated in the complex map where the number of knocked-down robots becomes variable (Fig. 12, *c*, *d*). This is because the robots turn more sharply and collide with obstacles on the left or right side without having enough time to detect them.

Despite the promising results obtained in this study, there are several limitations and potential areas for improvement. One limitation is that the study did not take into account the effect of communication delays between the robots. In real-world scenarios, communication delays can occur due to factors such as network congestion or interference, which could affect the performance of the multi agent system. Therefore, future studies should consider the impact of communication delays on the proposed method and explore strategies to mitigate their effects.

In addition, the proposed method relies on predefined maps and obstacle positions. In practice, the environment may change over time, which could affect the performance of the multi agent system. To address this issue, future work could investigate methods for adaptively updating the maps and obstacle positions based on sensor measurements and other environmental cues.

Finally, the proposed method assumes perfect sensor measurements and control actions. In reality, sensors can be noisy or faulty, and control actions may not be executed exactly as intended due to hardware limitations or other factors. Future work could explore strategies to make the multi agent system more robust to sensor and actuator uncertainties, such as using probabilistic models or incorporating feedback control.

In conclusion, the proposed method uses a velocity vector for motion consisting of linear velocity and angular velocity components. To conserve power, lidars are turned off for all robots except observer agents. The virtual leader's position is computed, and three observer agents with their lidars activated are determined based on the direction of motion. Agents with epsilon probability choose the opposite direction to avoid live lock and achieve a slight diversity of trajectory. The method was tested on two types of maps, and the results demonstrate its effectiveness in controlling a swarm of robots towards a desired destination while avoiding collisions. The impact of the values of linear velocity and the angle that changes the angular velocity on the performance of the swarm was observed. The experimental results suggest that the proposed method outperforms existing methods.

In conclusion, the experiment results demonstrate that the proposed method can effectively control the motion of a swarm of robots towards a target position. However, the design of the environment and the values of the linear and angular velocity should be carefully considered to ensure the swarm can reach the target position without colliding with obstacles. Further studies can be conducted to test the method in more complex environments and with different types of obstacles. The results of such studies can help refine the proposed method and enhance its practical applicability.

7. Conclusions

1. In this study, a leader-follower technique with a virtual leader has been designed to coordinate a multi-agent system of robots to move towards a target location while maintaining a desired formation and avoiding obstacles. The proposed method updates the position of the virtual leader with a certain frequency and recalculates the velocity vector to maintain the desired formation. By using this tech-

nique, the system can adapt to the changes in the formation caused by robot failures or crashes. The effectiveness of the leader-follower technique has been demonstrated through simulation experiments, which show that the system can maintain a specific geometric formation and avoid obstacles while following a designated path.

2. In this paper, we have proposed a method for selecting observer robots for lidar activation and calculating turn direction depending on lidar data. The selection of observer robots is dependent on the direction of the multi-agent system, with the leading robot acting as the middle observer and two robots positioned on the left and right sides acting as the left and right observers, respectively. By shifting the coordinate center to the virtual leader's position and rotating the x-axis in the direction of the multi-agent system, we were able to easily locate the observers. We have also described in detail the process for choosing observers and the direction of movement based on the lidar data collected.

The proposed method is highly effective in conserving power by switching off lidars of all robots other than the observers, thereby increasing the overall efficiency of the multi-agent system. Additionally, the method helps avoid collisions during direction changes, making the system safer and more reliable.

Overall, our proposed method provides an efficient and effective solution for selecting observer robots and calculating turn direction based on lidar data. We believe that this method can be applied to a wide range of multi-agent systems, making them more efficient and safer.

3. The epsilon-greedy algorithm has been successfully applied to avoid infinite loops during motion in the multi-agent system. This algorithm adds randomness to the robots' movements and helps them avoid following the same way over and over again. By choosing an «illogical» solution with a certain very small probability equal to epsilon, the system can prevent deadlocks in multi-agent systems. The value of epsilon is chosen in such a way that it affects the trajectory of the multi-agent system, thereby increasing the probability of a «fatal error.» However, the experiments have shown that the epsilon-greedy method will not affect the count of the robots that reach the target. This approach can be a useful tool for future developments in multi-agent systems.

4. This article presented a method for controlling the movement and coordination of a swarm of mobile robots in multiagent systems, aimed at maintaining a specific geometric formation while moving towards a target and avoiding collisions with obstacles. The proposed method employs a leader-follower technique with a virtual leader, a strategy for selecting observer robots that activate lidars during movement and calculate the turn direction, and the epsilon greedy algorithm to avoid infinite loops during motion. The study evaluated the proposed method by simulating the movements of a swarm of nine unmanned mobile robots in both complex and light maps.

The results showed that the proposed method effectively controls the motion of a swarm of robots towards a target position while avoiding collisions. The linear velocity and angle of rotation significantly affect the performance of the swarm, with increasing linear velocity reducing the number of steps required to reach the target. The optimal path taken by the swarm also has a significant impact on the number of steps required to reach the target. Moreover, the trajectory of the swarm varies at different values of the angular velocity, with low angular velocities increasing the frequency of collisions with obstacles and high angular velocities increasing the likelihood of collisions before turning away.

Overall, the proposed method provides a more efficient solution for controlling the movement of a swarm of robots towards a desired destination while avoiding collisions, and it has certain advantages over known results. These advantages include potential benefits for system scalability, reliability, energy consumption, and avoidance of infinite loops. The study's quantitative and comparative evaluations demonstrate the effectiveness of the proposed method in controlling the motion of a swarm of robots towards a target position in a simulated environment.

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 research is funded by the Science Committee of the Ministry of Science and Higher Edu Data availability cation of the Republic of Kazakhstan (Program No. BR1090140221).

Data availability

Manuscript has no associated data.

References

1. Kereyev, A. K., Atanov, S. K., Aman, K. P., Kulmagambetova, Z. K., Kulzhagarova, B. T. (2020). Navigation system based on bluetooth beacons: Implementation and experimental estimation. Journal of Theoretical and Applied Information Technology, 98 (8), 1187–1200. URL: http://www.jatit.org/volumes/Vol98No8/6Vol98No8.pdf
2. Ioannidi, K., Christakis, Ch., Sautbekov, S., Frangos, P., Atanov, S. K. (2014). The Radiation Problem from a Vertical Hertzian Dipole Antenna above Flat and Lossy Ground: Novel Formulation in the Spectral Domain with Closed-Form Analytical Solution in the High Frequency Regime. International Journal of Antennas and Propagation, 2014, 1–9. doi: https://doi.org/10.1155/2014/989348
3. Ouiazzane, S., Barramou, F., Addou, M. (2020). Towards a Multi-Agent based Network Intrusion Detection System for a Fleet of Drones. International Journal of Advanced Computer Science and Applications, 11 (10). doi: https://doi.org/10.14569/ijacsa.2020.0111044
4. Sakurama, K., Azuma, S.-I., Sugie, T. (2019). Multiagent Coordination Via Distributed Pattern Matching. IEEE Transactions on Automatic Control, 64 (8), 3210–3225. doi: https://doi.org/10.1109/tac.2018.2885491

5. Babazadeh, R., Selmic, R. (2020). Distance-Based Multiagent Formation Control With Energy Constraints Using SDRE. IEEE Transactions on Aerospace and Electronic Systems, 56 (1), 41–56. doi: https://doi.org/10.1109/taes.2019.2910361
6. Ermekbayuly Kyzyrkanov, A., Kubeisinovich Atanov, S., Abdel Rahman Aljawarneh, S. (2021). Formation control and coordination of swarm robotic systems. The 7th International Conference on Engineering & MIS 2021. doi: https://doi.org/10.1145/3492547.3492704
7. Li, Y., Zhang, J., Tong, S. (2022). Fuzzy Adaptive Optimized Leader-Following Formation Control for Second-Order Stochastic Multiagent Systems. IEEE Transactions on Industrial Informatics, 18 (9), 6026–6037. doi: https://doi.org/10.1109/tii.2021.3133927
8. He, S., Wang, M., Dai, S.-L., Luo, F. (2019). Leader–Follower Formation Control of USVs With Prescribed Performance and Collision Avoidance. IEEE Transactions on Industrial Informatics, 15 (1), 572–581. doi: https://doi.org/10.1109/tii.2018.2839739
9. Yan, J., Guan, X., Luo, X., Chen, C. (2017). Formation Control and Obstacle Avoidance for Multi-Agent Systems Based on Virtual Leader-Follower Strategy. International Journal of Information Technology & Decision Making, 16 (03), 865–880. doi: https://doi.org/10.1142/s0219622014500151
10. Darintsev, O. V., Migranov, A. B. (2019). The Use of Genetic Algorithms for Distribution of Tasks in Groups of Mobile Robots with Minimization of Energy Consumption. 2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon). doi: https://doi.org/10.1109/fareastcon.2019.8934927
11. Simon, D., Kapellos, K., Espiau, B. (1998). Control laws, tasks and procedures with ORCCAD: application to the control of an underwater arm. International Journal of Systems Science, 29 (10), 1081–1098. doi: https://doi.org/10.1080/00207729808929599
12. Nagy, I., Bencsik, A. L. (2007). A Simulation System for Behaviour based Potential Field Building in Multi-Agent Mobile Robot System. CI '07: Proceedings of the Third IASTED International Conference on Computational Intelligence, 7–12. URL: https://dl.acm.org/doi/abs/10.5555/1672041.1672044
13. Chang, Y., Yamamoto, Y. (2008). On-line path planning strategy integrated with collision and dead-lock avoidance schemes for wheeled mobile robot in indoor environments. Industrial Robot: An International Journal, 35 (5), 421–434. doi: https://doi.org/10.1108/01439910810893590
14. Bulut, V. (2022). Optimal path planning method based on epsilon-greedy Q-learning algorithm. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 44 (3). doi: https://doi.org/10.1007/s40430-022-03399-w
15. Liu, X., Zhang, P., Fang, H., Zhou, Y. (2021). Multi-Objective Reactive Power Optimization Based on Improved Particle Swarm Optimization With ε-Greedy Strategy and Pareto Archive Algorithm. IEEE Access, 9, 65650–65659. doi: https://doi.org/10.1109/access.2021.3075777