Movement Coordination of Swarm Robotic Systems

Abzal E. Kyzyrkanov[1,2], Sabyrzhan K. Atanov[2], Shadi A. Aljawarneh[3], and Nazira A. Tursynova[2]

[1] *Astana IT University, Astana, 010000, Kazakhstan*
[2] *L.N. Gumilyov Eurasian National University, Astana, 010000, Kazakhstan*
[3] *Jordan University of Science and Technology, Irbid, 22110, Jordan*

Abstract
This paper considers the motion of a swarm of self-organizing autonomous robots. An algorithm for controlling the motion of such systems, with a behavior-based approach, is proposed. To coordinate the movement of the robots in the swarm, a leader-follower approach with a virtual leader has been used.

Keywords
Robotic swarm, behavior-based approach, leader-follower

1. Introduction

Robotics is an extremely evolving area of modern industrial science. With the help of robots people solve various problems in different areas of life.

A great deal of work has been conducted in the history of robotics to improve robot performance and solve various specific problems of the robotics industry, such as communication approaches [1, 2], building an indoor navigation system [3], development of simplified encryption algorithms to deal with resource limitations on IoT devices [4], automatic control of a vehicle [6], etc.

One such solution is using a group of the multiple simple robots to solve a single complex problem. This approach is commonly referred to as group robotics. In recent years, research in the group robotics systems domain has attracted special attention of researchers. One of the main tasks of group robotics is to coordinate and control robots in a group so that they can interact effectively and without interfering with each other to achieve a common goal. Group robotics can have a huge impact on robotics by letting simple machines solve problems that they would not be able to solve alone and thus easing the entire development process. The idea of using a few simple cheap robots which are simple to develop has been considered by researchers for a long time. The first group robotics applications were already presented three decades ago [6]. Nowadays, group robotics is widely used in the development of mobile robots[7, 8, 9], UAV [10, 11, 12], satellites [13, 14], sensor networks, power grids, etc.

Recently, many researchers have been increasingly focusing on the swarm of robots. A robot swarm refers to a group of robots of the same type that are designed to perform a single complex common task. And, if we have to combine a large number of robots to solve a problem, the problem of organizing the interaction of these robots is reduced to the problem of coordinating the same type of robots, since the development of a large number of varieties of robots for only one task makes no sense.

One of the most important problems of Swarm Robotic Systems is the coordination of movement. Here it is necessary to solve several problems, such as preventing collisions with each other or with an obstacle, etc. There has been a lot of work in this area [15, 16, 17, 18].

2. Problem statement

This paper considers a swarm robot system with decentralized control. Unlike centralized systems, in such systems the agents are not controlled by a center unit of control. Here, each robot can use signals received from each other or from external sources, but mostly makes decisions independently based on its observations (i.e., sensor data) and state. To maintain motion a behavior-based algorithm and a leader-follower method with a virtual leader has been used.

2.1. Behavior-based algorithm

The core idea of a behavior-based approach is that robots move according to several required behaviors. Each behavior corresponds to a particular problem the robot is trying to solve. In this case, the required behaviors of the robots may be as follows:
- Target retrieval and attainment;
- Collision avoidance with an obstacle or with each other.

The main benefits of this algorithm is that it allows more than one problem to be solved simultaneously, simply by adding a behavior for each task.

The classical behavior-based algorithm can be described by the equation [19, 20]:

$$\vec{V}_{total} = \sum_{b \in B} w_b * \vec{V}_b, \tag{1}$$

Here B is the set of desired behaviors, $\vec{V}_b$ is the speed of maintenance of behavior b, which are calculated from sensor data, w_b is the weight of the behavior corresponding to the relative importance of that behavior (Figure 1).
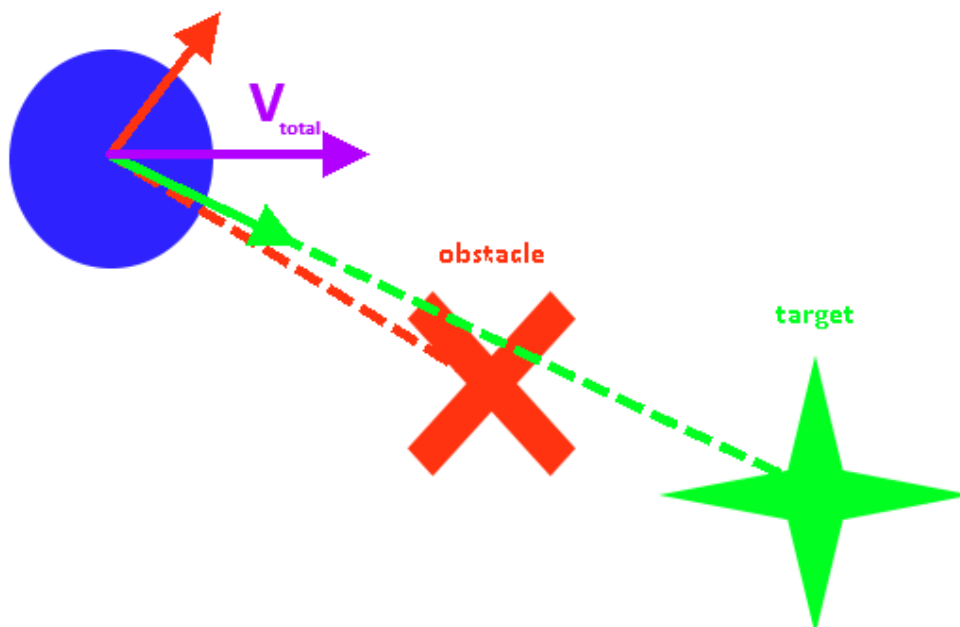

Figure 1: The working principle of the behavior-based approach

Some papers refer to the same approaches as an artificial potential field [26, 27 and 28] or virtual forces [29]. The classical versions of these algorithm are the identical: the only difference is that instead

of velocities of behavior, they use potential fields or virtual forces that are attracted to desired positions or repulsed from undesirable positions.

2.2. Leader-Follower approach

The main idea of the leader-follower algorithm is that the followers try to maintain the required distance during the movement [16]. By representing the Leader and Followers as points in the surface and having the pose of each point, it is easy to calculate the distance between them. Supposing that the follower pose is (x_F, y_F) and the leader pose is (x_L, y_L), the distance is calculated as:

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

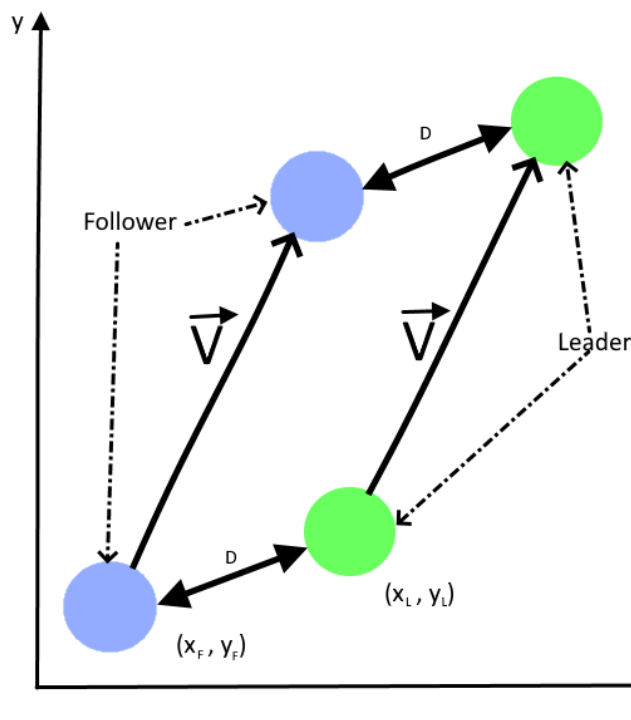This distance can be used as a range that the follower tries to keep (Figure 2).

Figure 2: Representation of the robot leader and the robot follower as material points

If some robot from the swarm is assigned as the leader, the robust of the swarm is at serious risk, if the leader crashes the swarm fully becomes inoperable. To deal with such a challenge, we can assign a virtual leader and maintain a distance with respect to him [17, 18]. The location of the virtual leader could be simply marked in the center of the swarm. During the motion, the leader's position can be calculated as the mean of the positions of the robots in swarm. .

3. Algorithm

The robots of the swarm start motion, updating their velocity vector with time intervals. At each time interval swarm will perform following steps
- the location of the Leader is estimated;
- the velocity vector for each desired behavior is calculated;
- using calculated in previous step velocities (equation (1)) the total velocity for each robot in the swarm is calculated.

The location of the virtual leader is estimated using the following equation:

$$x_L = \frac{1}{N} \sum_{i=1}^{N} x_i$$

$$y_L = \frac{1}{N} \sum_{i=1}^{N} y_i$$

(3)

Here, N is the count of the robots in swarm, (x_i, y_i) is the current location of the i-th robot, (x_L, y_L) is the location of the virtual leader.

To calculate the total velocity, a behaviors-based approach is used. In our case, the robots observe two types of behavior: moving to the target and avoiding collision with each other or with an obstacle.

The value of the collision avoidance speed is inversely proportional to the distance to the closest obstacle. Any object that in any way may interfere with the robot's movement, including all other robots in the swarm, is considered an obstacle. The collision avoidance velocity angle is Perpendicular to the closest obstacle, which way it goes differs depending on which side has more obstacles, because in case of opposite movement to the obstacle the robots may end up in a deadlock state.

The velocity of movement to the goal is oriented from the virtual leader's position to the target's position and is the same for all robots in the swarm.

After all, using these velocities (equation (1)), the total velocity can be calculated as:

$$\vec{V}_{total} = w_1 * \vec{V}_{avoid\ collision} + w_2 * \vec{V}_{move\ to\ goal}$$

(4)

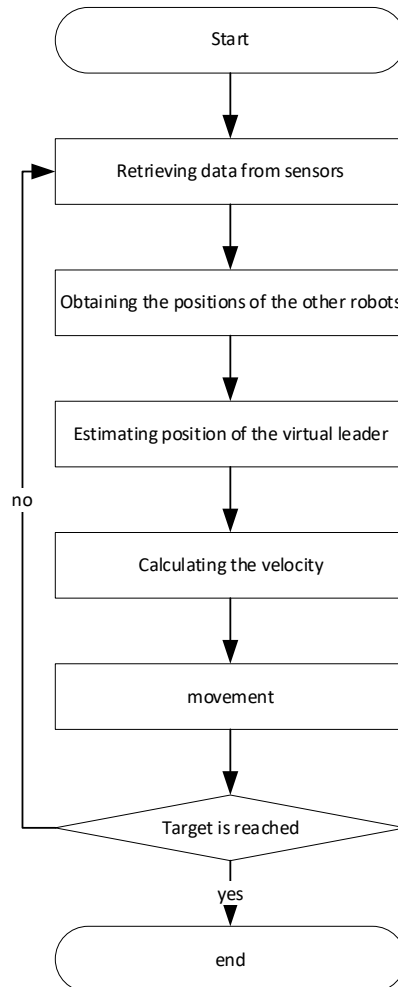The flowchart of the complete algorithm is shown in Figure 3.

Figure 3: The flowchart of the complete algorithm

4. Simulation

To examine the work of the algorithm proposed in this paper, we performed experiments with a swarm robotic system of nine robots. The swarm motion was simulated using a software application developed in the Python programming language. The Pygame library was used for visualization (Figure 4).
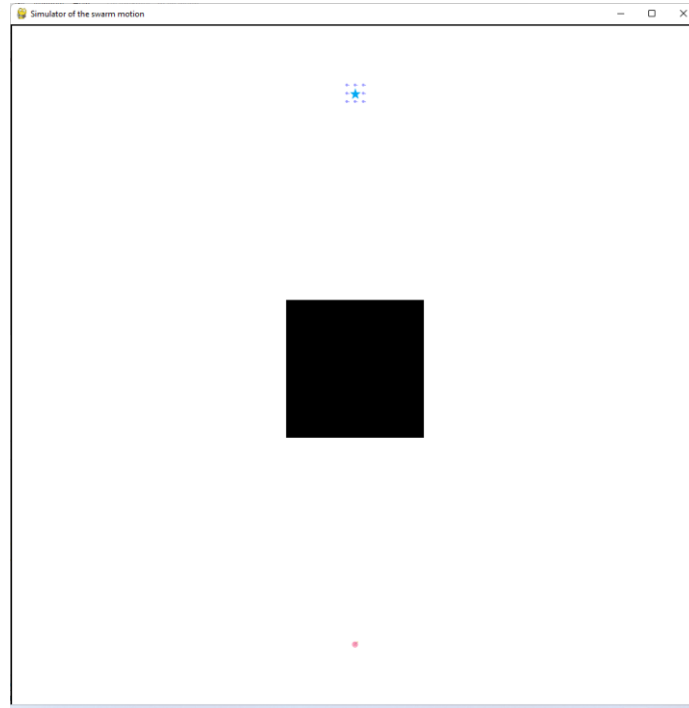

Figure 4: Initial state of the robots in swarm

Each of the robots in the swarm has 12 lidars. The detection radius of all lidars is same. The nearby lidars are relative to each other at an angle of 30 degrees, so they cover all sides of the robot (Figure 5).
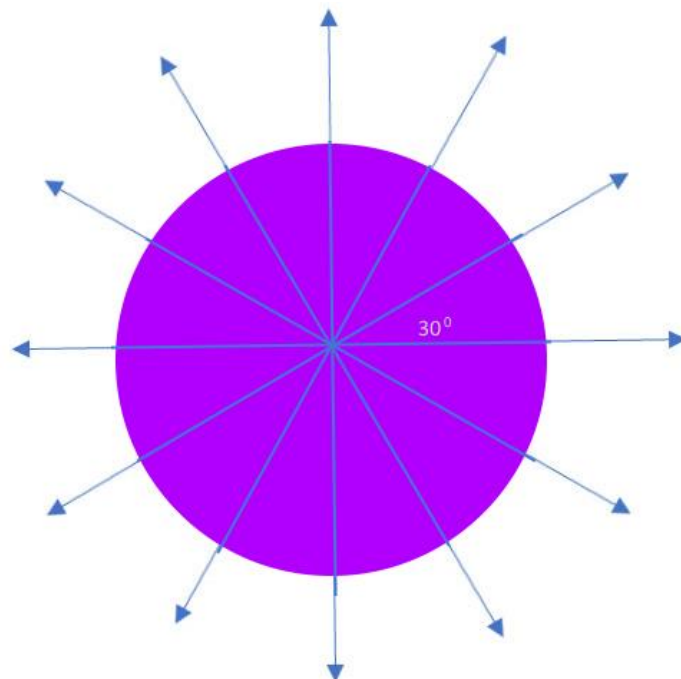

Figure 5: Model of the robot

It is also assumed that each robot in the swarm is aware of the positions of the other robots: in fact, they can broadcast information about their own location to each other.

At the beginning of the movement the robots in the swarm form a square (Figure 4). Robots move in the direction of the target (marked in red) avoiding the obstacle.

5. Experimental results and discussion

Experimental research was carried out with weights of each behavior ranging from 0.1 to 10. Two values were observed: the total number of steps to reach the goal and the count of robots reaching the target.

For a small value of collision avoidance speed weight ($w1$), the count of agents which reach the target is very small, for a value less than 0.3 the robots do not reach the target at all. Experimental results for values less $w1 <=1$ in which at least one robot reaches the target are presented in Table 1.

Table 1
Experimental results for values less $w1 <=1$ in which at least one robot reaches the target

The weight of the speed of avoiding collision (w_1)	The weight of the speed of reaching the target (w_2)	The count of the robots reaching the target	The total number of steps to reach the goal
0.3	0.1	4	7049
0.4	0.1	3	7108
0.5	0.1	5	6051
0.6	0.1	7	4785
0.7	0.1	5	4657
0.8	0.1	7	4520
0.9	0.1	6	4863
1	0.1	3	5071
0.6	0.2	6	2307
0.7	0.2	4	2450
0.8	0.2	5	2982
0.9	0.2	5	2679
1	0.2	5	2594
0.8	0.3	2	1525
0.9	0.3	3	1895
1	0.3	4	1714

Also, robots cannot reach target on the large values of the weight of the speed of reaching the target. For a value $w_2>2.7$ the robots do not reach the target at all. Experimental results for values less $w_2>2$ in which at least one robot reaches the target are presented in Table 2.

Table 2
Experimental results for values less $w2 >2$ in which at least one robot reaches the target

The weight of the speed of avoiding collision (w1)	The weight of the speed of reaching the target (w2)	The count of the robots reaching the target	The total number of steps to reach the goal
7.8	2	1	225
9.1	2	1	220
9.2	2	1	221
9.5	2	2	222
9.8	2	1	220
9.9	2	2	216

10	2	1	217
9	2.1	1	209
9.1	2.1	2	208
9.4	2.1	2	209
9.8	2.1	1	205
9.9	2.1	1	205
10	2.1	2	206
9.3	2.2	1	200
9.5	2.2	2	197
9.9	2.2	2	197
8.8	2.3	1	199
9.3	2.4	1	186
8.6	2.5	1	181
9.5	2.5	1	180
9.6	2.5	1	182
9.9	2.6	1	175
10	2.6	1	174

As the weight increases, the number of robots reaching the goal gradually decreases (Figure 6). But the robots reach the goal faster, the number of steps to reach the goal decreases (Figure 7).
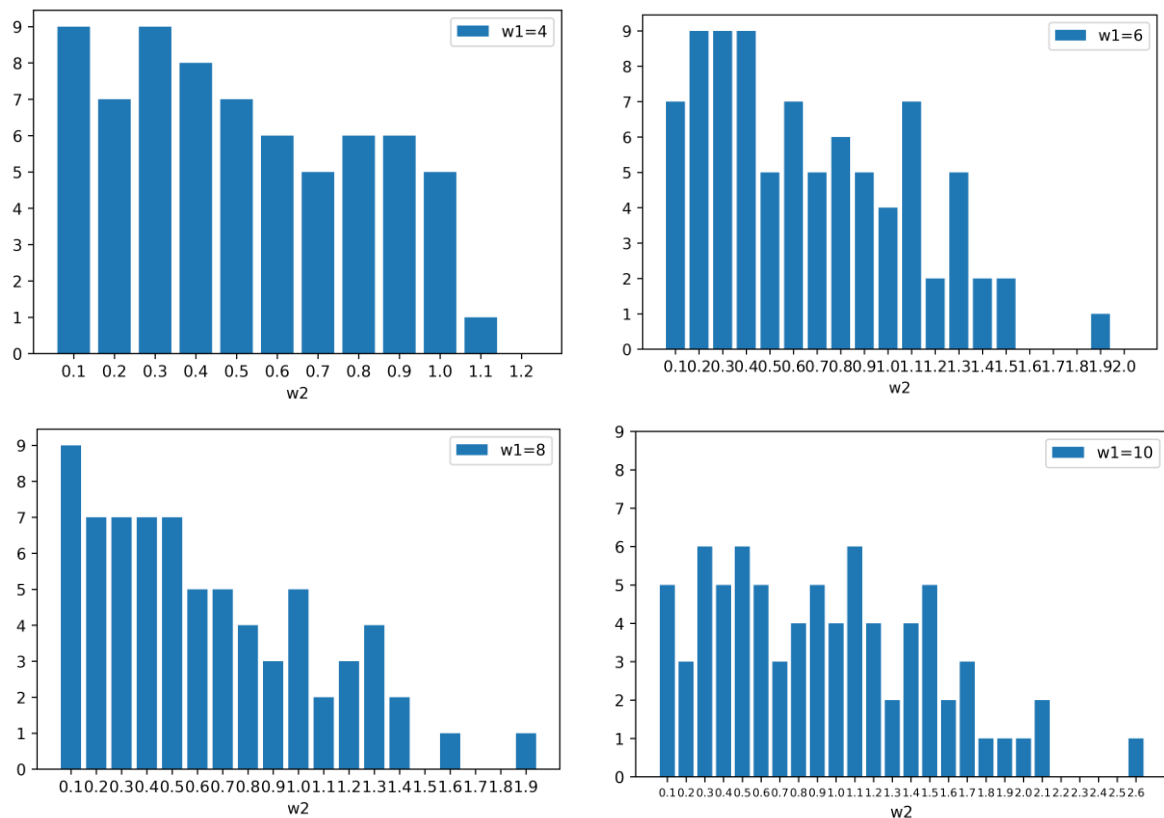

Figure 6: The graph of dependence of the count of robots reaching the goal on the goal reaching speed weight (w2) at the value of the collision avoidance speed weight: w1=4, 6, 8, 10
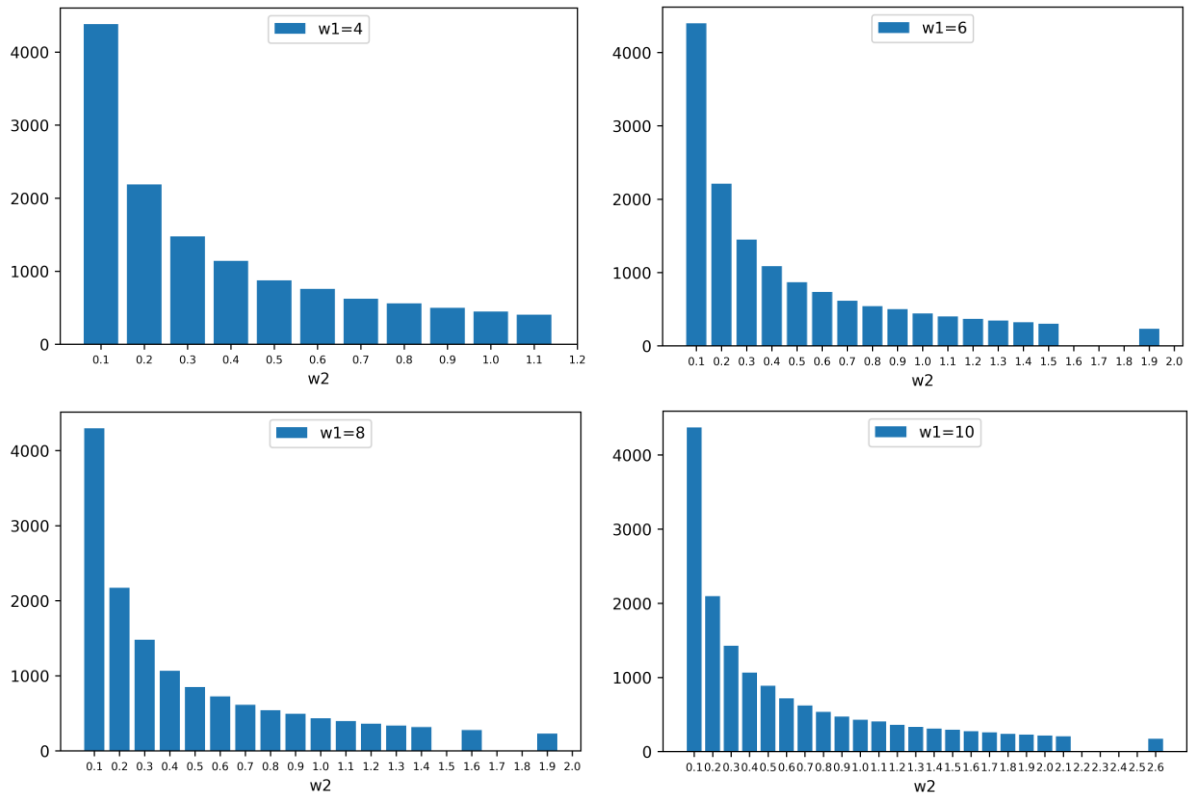
Figure 7: The graph of the dependence of the number of steps of reaching the goal on the goal reaching speed weight (w_2) at the value of the collision avoidance speed weight: w1=4, 6, 8, 10

As the collision avoidance speed weight increases, the number of robots reaching the target increases to a maximum and in many cases (especially with a small target speed weight) keeps this position (Figure 8) But as the weight of the collision velocity grows, the robots cannot always reach the target fully, they begin to run into each other because of the small sensitivity of the lidar.
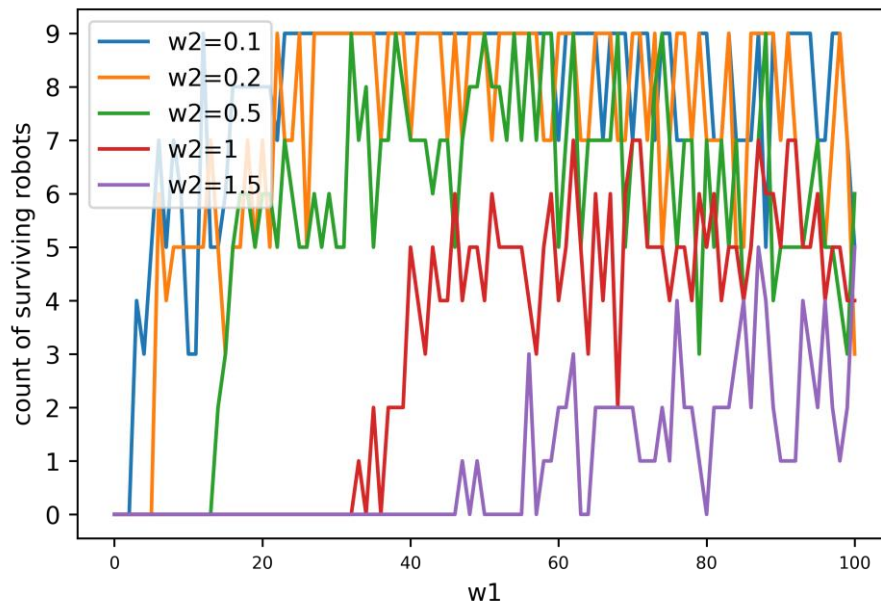

Figure 8: The graph of dependence of the count of robots reaching the goal on the collision avoidance speed weight(w1) at the value of the goal reaching speed weight: w_2=0.1, 0.2, 1, 1.5

The number of steps to reach the goal does not depend as much on the collision avoidance speed weight, only in some cases (especially when the goal speed weight is small) the robots can go around the obstacle for a long time (Figure 9).
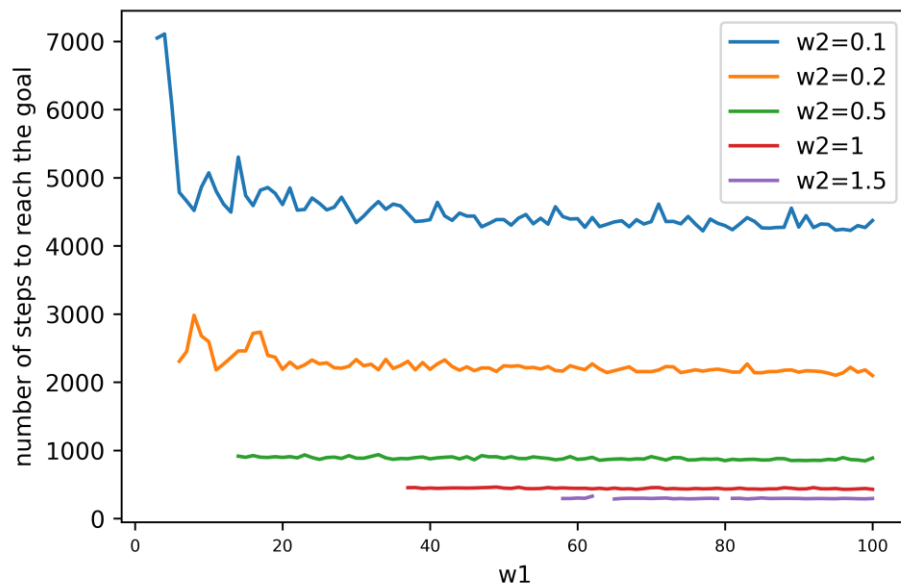

Figure 9: The graph of the dependence of the number of steps of reaching the goal on the collision avoidance speed weight(w1) at the value of the goal reaching speed weight: w_2=0.1, 0.2, 1, 1.5

6. Conclusion

In this paper, we proposed an algorithm to control the motion of a swarm of robots. A behavior-based approach has been used to control the motion of the robots. This method allows to establish a movement without a management center, which increases the fault tolerance. In addition, the leader - follower method was proposed for coordination, which is one of the simplest algorithms to develop.

The presented algorithm was used to simulate the movement of a swarm of 9 robots in size. The changes in the number of steps to reach the goal and the number of robots reaching the goal were examined. During the experiments it was found out.

- The number of steps of reaching a goal depends very strongly on the weight of the speed of reaching the goal. By increasing the value of this weight the robots reach the goal faster.
- The number of robots reaching the goal decreases as the weight of the speed of reaching the goal increases. For at least one robot to reach the goal, the collision avoidance speed weight must be at least two and a half times the goal achievement speed weight.
- By increasing the weight of the collision avoidance speed the number of robots reaching the goal reaches a maximum.
- But as both weights increase, the robots start colliding with each other due to the low sensitivity of the lidar.

7. References

[1] M. B. Yassein, S. Aljawarneh and A. Al-Sadi, Challenges and features of IoT communications in 5G networks, in: Proceedings of the 2017 International Conference on Electrical and Computing Technologies and Applications, ICECTA, 2017, pp. 1-5, doi: 10.1109/ICECTA.2017.8251989.
[2] W. Mardini, S. Aljawarneh, and A. Al-Abdi, Using Multiple RPL Instances to Enhance the Performance of New 6G and Internet of Everything (6G/IoE)-Based Healthcare Monitoring Systems, Mobile Netw Appl 26 (2021) 952–968.

[3] A. K. Kereyev, S. K. Atanov, K. P. Aman, Z. K. Kulmagambetova and B. T. Kulzhagarova, Journal of Theoretical and Applied Information Technology 98.88 (2020) 1187–1200.

[4] M. A. Al-Husainy, B. Al-Shargabi, S. Aljawarneh, Lightweight cryptography system for IoT devices using DNA, Computers & Electrical Engineering 95 (2021) 107418.

[5] Q. Chen, and J. Y. S. Luh, Coordination and control of a group of small mobile robots, in: Proceedings of the 1994 IEEE International Conference on Robotics and Automation, IEEE, 1994.

[6] S. L. Zenkevich, A. V. Nazarova, H. Zhu, Logical control a group of mobile robots, in: Gorodetskiy, A., Tarasova, I. (Eds.) Smart Electromechanical Systems. Studies in Systems, Decision and Control, vol 174, Springer, Cham. URL: https://doi.org/10.1007/978-3-319-99759-9_4.

[7] J. Huang, S. M. Farritor, A. Qadi, S. Goddard, Localization and follow-the-leader control of a heterogeneous group of mobile robots, IEEE/ASME transactions on Mechatronics11.2 (2006) 205-215.

[8] E. Kuiper, S. Nadjm-Tehrani, Mobility models for UAV group reconnaissance applications, in: Proceedings of the 2006 International Conference on Wireless and Mobile Communications, ICWMC'06, IEEE, 2006 Jul 29, pp. 33-33.

[9] B. Bethke, J. P. How, J. Vian, Group health management of UAV teams with applications to persistent surveillance, in: Proceedings of the 2008 American Control Conference, IEEE, 2008 Jun 11, pp. 3145-3150.

[10] K. Gai, Y. Wu, L. Zhu, K. K. Choo, B. Xiao, Blockchain-enabled trustworthy group communications in UAV networks, IEEE Transactions on Intelligent Transportation Systems 22.7 (2020) 4118-30.

[11] I. Banik, D. O'Ryan, H. Zhao, Origin of the Local Group satellite planes, Monthly Notices of the Royal Astronomical Society 477.4 (2018) 4768-91.

[12] F. Jiang, A. Dekel, J. Freundlich, F. C. van den Bosch, S. B. Green, P.F. Hopkins, A. Benson, X. Du, SatGen: a semi-analytical satellite galaxy generator – I. The model and its application to Local-Group satellite statistics, Monthly Notices of the Royal Astronomical Society 502.1 (2021) 621-41.

[13] M. F. Baimukhamedov, K. Moldamurat, M. K. Akgul, Optimal control model of the automobile transport (2019) 1312–1316.

[14] A. K. Yemelyev, Kh. Moldamurat, and R. B. Seksenbaeva, Development and Implementation of Automated UAV Flight Algorithms for Inertial Navigation Systems, in: Proceedings of the 2021 IEEE International Conference on Smart Information Systems and Technologies, SIST, IEEE, 2021, pp. 1-5.

[15] A. E. Kyzyrkanov, S. Atanov, and S. Aljawarneh, Formation control and coordination of swarm robotic systems, in: Proceedings of the 7th International Conference on Engineering and MIS, ICEMIS, 2021.

[16] A. Kyzyrkanov, S. Atanov and S. Aljawarneh Coordination of movement of multiagent robotic systems, in: Proceedings of the 2021 16th International Conference on Electronics Computer and Computation, ICECCO, 2021, pp. 1-4. doi: 10.1109/ICECCO53203.2021.9663796.

[17] T. Balch, and R. C. Arkin, Behaviour-based formation control for multirobot teams, IEEE transactions on robotics and automation 14.6 (1998) 926-939.

[18] D. Xu, X. Zhang, Zh. Zhu, Ch. Chen, and P. Yang, Behaviour-based formation control of swarm robots, Mathematical Problems in Engineering (2014).

[19] F. A. Raheem, and M. M. Badr, Development of Modified path planning algorithm using artificial potential field (APF) based on PSO for factors optimization, American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS) 37.1 (2017) 316-328.

[20] Sh.-M. Wang, M.-Ch. Fang, and Ch.-N. Hwang, Vertical obstacle avoidance and navigation of autonomous underwater vehicles with H∞ controller and the artificial potential field method, The Journal of Navigation 72.1 (2019) 207-228.

[21] X. Yan, D. Jiang, R. Miao, and Y. Li, Formation Control and Obstacle Avoidance Algorithm of a Multi-USV System Based on Virtual Structure and Artificial Potential Field, Journal of Marine Science and Engineering 9.2 (2021) 161.

[22] X. Wang, D. Zhang, L. Shen, and J. Zhang, A virtual force approach for cooperative standoff target tracking using multiple robots, in: Proceedings of the 2016 Chinese Control and Decision Conference, CCDC, IEEE, 2016, pp. 1348-1353.

[23] E. Kuiper, S. Nadjm-Tehrani, Mobility models for UAV group reconnaissance applications. In: Proceedings of the 2006 International Conference on Wireless and Mobile Communications, ICWMC'06, IEEE, 2006 Jul 29, pp. 33-33.

[24] B. Bethke, J. P. How, J. Vian, Group health management of UAV teams with applications to persistent surveillance, in: Proceedings of the 2008 American Control Conference, IEEE, 2008 Jun 11, pp. 3145-3150.

[25] K. Gai, Y. Wu, L. Zhu, K. K. Choo, B. Xiao, Blockchain-enabled trustworthy group communications in UAV networks, IEEE Transactions on Intelligent Transportation Systems 22.7 (2020) 4118-30.

[26] I. Banik, D. O'Ryan, H. Zhao, Origin of the Local Group satellite planes, Monthly Notices of the Royal Astronomical Society 477.4 (2018) 4768-91.

[27] F. Jiang, A. Dekel, J. Freundlich, F. C. van den Bosch, S. B. Green, P. F. Hopkins, A. Benson, X. Du, SatGen: a semi-analytical satellite galaxy generator – I. The model and its application to Local-Group satellite statistics, Monthly Notices of the Royal Astronomical Society, 502.1 (2021) 621-41.