Research Article

Three Dimensional Formation Control of Unmanned Aerial Vehicles in Obstacle Environments

Abdulmelik Bekmez, Kadir Aram, Burhanettin Can

***Abstract*—Today, the use of small unmanned aerial vehicles (UAVs) has increased due to technological advances. There has been an interest in using multiple UAVs instead of a single UAV to accomplish a given mission. This is because there are many scenarios where the capabilities of a single UAV are inadequate due to certain constraints (battery capacity, time in the air). For this reason, swarm UAV studies have increased. A swarm UAV consists of a large number of UAVs cooperating to accomplish a specific mission.**

This study shows three-dimensional formation control of a swarm UAV system in an obstacle environment. A centralized control architecture is used in this process. All task assignments are made from a centralized system. The Artificial potential fields method creates the formation at the target point by avoiding obstacles. The study was carried out using the Robot Operating System (ROS). The methods were tested in Webots simulation environment. Crazyflie robots were used in the experiments.

In the simulation environment, square, star and v formations were first tested in two dimensions. Then, as an example of three-dimensional formation, the cubic and pyramid formations were created and observed.

***Index Terms*—Robotics, Swarm, UAV, Formation.**

I. Introduction

ROBOTS are used for many different purposes in daily life. Some of these needs are in the terrestrial environment, but some are also required in aerial operations. Ground or aerial robots can be used depending on the need for use. While a single robot can be used while performing these tasks, it may be more advantageous to use multi-robot systems in some cases.

Swarm robotics constitutes an area of scholarly investigation that examines the utilization of multiple autonomous agents (robots) within a system to achieve collective tasks. These tasks are typically beyond the capability of individual robots in isolation, or they are executed with greater efficiency when performed collaboratively as a cohesive group [1].

Swarm structures are inspired by observations of social insects, which show that large numbers of individuals can interact to form intelligent systems. Swarm structures in nature do not have a centralized coordination system, but their functioning at the system level is robust, flexible and scalable. These characteristics have also been a motivation for swarm robots[2].

Unmanned aerial vehicles (UAV) are used in both civilian and military fields. These areas include fire detection and response applications [3][4] , search and rescue operations [5] and road construction works [6]. In the contemporary era, a plethora of missions demands UAVs to navigate over expansive regions or accomplish tasks within tight time constraints. However, relying on a single UAV might not suffice to meet the performance criteria due to its constrained dimensions and limited battery capacity [7].

The use of multiple UAVs has many advantages. These advantages are time efficiency, cost, simultaneous actions, complementarity, fault tolerance and flexibility [8] . Therefore, it is used in many fields, such as Video surveillance [9] , traffic monitoring [10], search and rescue [11], Simultaneous Localization and Mapping [12]. UAVs frequently engage in formation flight, where the relative distances between each pair of UAVs remain constant, leading to the cohesive movement of the entire formation. To preserve the shape of the formation, it suffices to maintain the distances between a specific number of agent pairs, effectively ensuring a constant distance between all pairs in the formation [13].

Brandao and Filho used a centralized approach for formation control in their study. These layers are responsible for creating the desired path for the formation, ensuring the robot's desired posture and generating the control signal for each robot to reach its desired position [14]. Kim et al., designed a framework for swarm systems that uses a decentralized control approach. Each agent uses artificial potential functions to form swarms and control their formation [15]. Li et al. created a game model for UAVs to ensure flight safety by avoiding obstacles in a given formation [16]. Yavuz et al., developed a method for task assignment to multiple UAV systems. First, target locations are clustered according to the number of UAVs. Then, appropriate target assignment was made with the Hungarian algorithm. Finally, the optimal path for the mission location was calculated with the ant colony algorithm [17].

This study provides formation control of a system consisting of swarm UAVs. A centralized system calculates formation points, and proportional control is used for position control. Hungarian algorithm was used to assign the formation points to the robots in the swarm. The experiments tested star, crescent, cube, and v-formation shapes. The artificial potential field method is used for obstacle avoidance during the formation

Abdulmelik Bekmez is with the Department of Computer Engineering, Engineerig Faculty,Fatih Sultan Mehmet Vakıf University University, İstanbul, 34445 TURKEY e-mail: abdulmelik.bekmez@stu.fsm.edu.tr
https://orcid.org/0009-0008-4211-941X

Kadir Aram is with the Department of Computer Engineering, Engineerig Faculty,Fatih Sultan Mehmet Vakıf University University, İstanbul, 34445 TURKEY e-mail: karam@fsm.edu.tr
https://orcid.org/0000-0002-5780-6334

Burhanettin Can is with the Department of Computer Engineering, Engineerig Faculty,Fatih Sultan Mehmet Vakıf University University, İstanbul, 34445 TURKEY e-mail: bcan@fsm.edu.tr
https://orcid.org/https://orcid.org/0000-0002-0998-4371

Manuscript received Aug 18, 2023; accepted Sep 18, 2023.
DOI: 10.17694/bajece.1345915

process. In the study's second part,materials and methods are shown; in the third part,the system's structure explained, in fourth part the results of the tests are shown. In the last section, the results obtained are interpreted.

II. MATERIALS AND METHODS

Utilizing multiple robots enhances the system's robustness compared to a single robot. A multi-robot system exhibits unparalleled superiority, particularly in scenarios involving environmental detection, survivor search, and other intricate tasks. The critical issue is ensuring that multiple robots can be organized effectively. This requires communication and control according to the desired situation [18].

A. Formation Control Methods

Some of the main elements that need to be considered when controlling the formation are sharing information within the swarm agents and assigning tasks to these agents. Here the decision maker has to take into account issues such as fault tolerance and careful use of energy. There are two main control schemes in formation control: centralized and distributed control methods. In a centralized control framework, a collective of cooperative robots receives control directives from a potent central unit, a robot within the formation equipped with robust computational capabilities or a control station. The core unit establishes communication with all team members, assimilates information from each individual, optimizes vehicle coordination, monitors mission progress, and handles any individual faults that may arise[19]. The distributed formation control approach operates independently of a central control center, granting equal status to all UAVs within the formation. Within the distributed control structure, the global control challenge is decomposed into several sub problems, each independently tackled by the controller within individual UAVs. Through distributed cooperative control, the formation of the UAV swarm can be successfully achieved[20].

B. Formation Control Strategies

Three main problems must be considered when performing formation control of multiple UAVs. The first one is the task of directing the agents to form the desired formation structure. The second one is the preservation of the created formation. For this, a control strategy should be created. The third is the reconstruction of the formation. When performing operations in the environment, the formation must be preserved in case the formation encounters obstacles or other disturbing factors. Different control strategies have been proposed for such problems. [7].

1) Virtual structure: Within the virtual structure approach, the entirety of the formation is regarded as a unified entity capable of coordinated movement akin to a rigid body in a specified direction and orientation. Moreover, this approach ensures the preservation of the geometric arrangement among multiple vehicles based on a reference point within the virtual structure.[21].

2) Leader-Follower: The leader-follower technique involves designating a specific robot as the leader, with the remaining robots acting as followers. The leader communicates with all other UAVs within the group and determines the path for the entire UAV formation. The follower UAVs track the leader's reference state and carefully maintain a prescribed distance and angle to avert collisions and disarray, achieving overall consistency in the formation. [22].

3) Behavior-based: The behavior-based UAV formation control strategy represents a distributed approach that governs UAVs based on fundamental behavior rules. A robot's overall behavior results from a fusion of various sub behaviors, such as encompassing going to the goal, obstacle avoidance, speed matching,wall-following and formation maintenance. In practical scenarios, each robot's behavior is expressed as a vector comprising magnitude and direction, with the vector's weight modifiable through parameter adjustments. The robot selects appropriate behaviors to generate a movement command by assimilating information from the surrounding environment. The robot's collective behavior vector emerges as the summation of all individual sub behavior vectors [23].

4) Artificial Potential Field: Artificial Potential Field (APF) is a method that emulates repulsion and attraction forces akin to those in a gravitational field. The fundamental concept revolves around utilizing attractive and repulsive forces to generate control actions. These forces correspond to the negative gradients of the attractive and repulsive potential fields, respectively. The attractive force facilitates formation convergence, while the repulsive force ensures collision avoidance within the formation[7].

C. Robot Operating System (ROS)

ROS (Robot Operating System) is an open-source and freely available robotics software framework suitable for deployment in diverse research and commercial applications [24]. It is a software platform that enables the development of robot applications with the libraries and packages it contains [25]. ROS provides many conveniences to users. Some of them are the support of multiple programming languages, the ability to communicate between processes and the use of the developed program code in other projects. It also includes visualization tools[26]. Communication in ROS is done through messages. These messages are transmitted through topics. The nodes in ROS communicate over these topics. The ROS Master, launched at the start of the ROS server, manages the communication between all nodes. The node sends all its information to the ROS master, including the type of data it sends or receives. Nodes that send data are called publisher nodes, and nodes that receive data are called subscriber nodes. The ROS Master has all the publisher and subscriber information running on computers [24].

D. Simulation

Webots is an open-source and multi-platform desktop application for simulating robots. It provides a development environment for modeling, programming and simulating robots. Webots was chosen because of its rich documentation, ROS2 integration is effortless to set up and configurable, and it has a ready-made crazyflie model.

E. Quadrotor Robot

The Crazyflie 2.0 nano quadrotor helicopter, functioning as an open-source experimental platform, holds significance for research and educational endeavors in robotics. Weighing a mere 27g and fitting comfortably within one's hand, this quadcopter boasts a diminutive stature. This compact size renders it amenable to indoor flights conducted in closely packed formations. Additionally, owing to its limited inertia, the Crazyflie can endure high-velocity impacts, minimizing potential hazards to humans [27] [28].

III. IMPLEMENTATION

The first step is to select a suitable software architecture for the implementation of the application. In order to create the desired formation for the swarm, the locations where the robots should go were determined. Then the most suitable robot is selected for the determined formation points. A user interface was designed to observe all these processes.

A. Software Architecture

The application was developed on a docker image with Ubuntu 22.04 operating system.ROS2 Humble version was used. This version provides tier 1 support on Ubuntu 22.04. All the requirements for the project to be installed and run are created as a Docker image. In this way, users can control both the physical robots and the simulation environment with a container created using the relevant docker image. There is a position controller node for each agent in the swarm. These position controller nodes were developed using the Rust programming language. A user interface has been developed for the control of the swarm system. While developing the interface, the egui library in the Rust language was used. The egui library uses immediate mode architecture while developing the interface. The whole interface is redrawn in each iteration. This allows the system to be developed more dynamically. The user interface is shown in Figure 1.

The following commands can be given to swarm agents through the interface.

- Departure
- Landing
- Navigation
- Formation

B. Calculation of Formation Points

Formation points are calculated using a circle formation algorithm. All formations are formed by circles, and these circles are calculated and combined one by one. Most regular polygon geometries can be created using circles. When creating any circle formation, a generic algorithm is generated by taking parameters such as circle center, radius and distance between points.

=

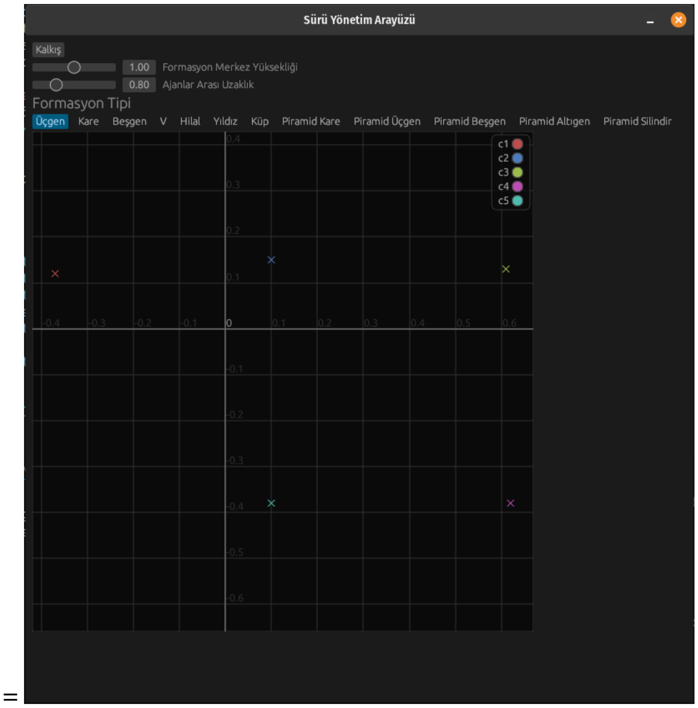

Fig. 1. User Interface

Three-dimensional formations are a combination of two-dimensional ones. Separate two-dimensional formations are created for the desired three-dimensional formation shape. These formations may vary according to the desired shape. For example, two squares are created for a cube formation. For a square pyramid, one is sufficient.

Before calculating the formation points, the radius of the circle is calculated. This is because the radius of the circle will change depending on the distance between the formation points. The radius of the circle is calculated according to Equation 1.

$$r = \frac{d/2}{sin(\theta/2)} \tag{1}$$

Here : d=distance between formation positions, θ = angle between formation positions.

A formation shape using a circle is shown in Figure 2.

Theta value in Figure 2 indicates the angle between the formation points. This value is calculated in Equation 2.

$$\theta = 2\frac{\pi}{n} \tag{2}$$

n_i indicates the number of position in the formation. θ indicates the angle difference between formation points.
Using these equation, the angles of all points are calculated.

$$(0\ \theta\ 2\theta\ ...\ 2\pi - \theta\ )$$

Once all angles are calculated, the required positions are calculated using these angles.

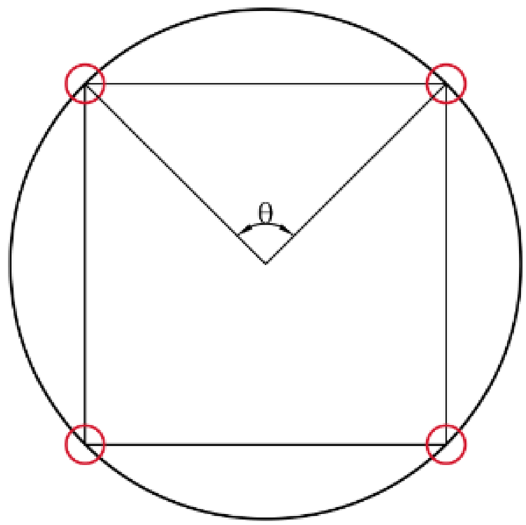

Fig. 2. Formation Circle Points

$$\begin{pmatrix} \mathrm{rcos}(0) + \mathrm{x}_0 & \mathrm{rsin}(0) + \mathrm{y}_0 & \mathrm{z}_0 \\ \mathrm{rcos}(\theta) + x_0 & \mathrm{rsin}(\theta) + y_0 & \mathrm{z}_0 \\ \mathrm{rcos}(2\theta) + x_0 & \mathrm{rsin}(2\theta) + y_0 & \mathrm{z}_0 \\ \dots & \dots & \dots \\ \mathrm{rcos}(2\pi - \theta) + x_0 & \mathrm{rsin}(2\pi - \theta) + y_0 & \mathrm{z}_0 \end{pmatrix}$$

1) Star Formation: Two five-element circles are used to calculate the star formation points. The initial angles and radius of these circles are determined to obtain the star formation.It is sufficient that the difference between the starting angles of the circles is 36°. This value is equal to the length of one side of a regular polygon which has 10 points. To calculate their radius, the ratio between the radius of the inner circle and the radius of the outer circle must be known.The circles used for the star formation are shown in Figure 3.

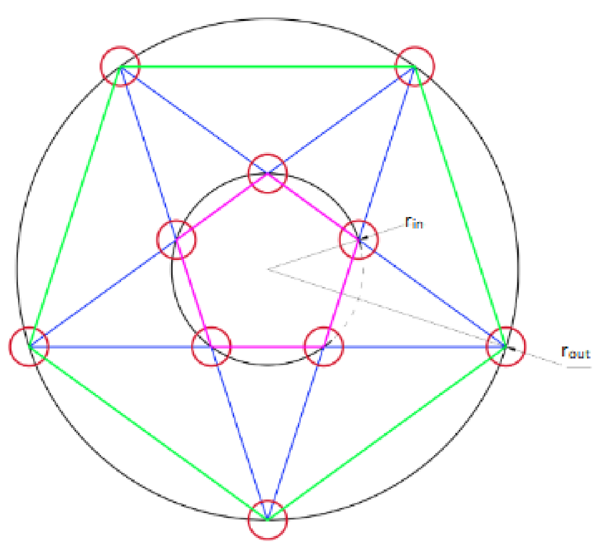

Fig. 3. Circles in Star Formation

In order to find the ratio between r_{out} and r_{in}, a corner of the star in Figure 4 is considered. Using the angles and lengths at this corner, the ratio between r_{out} and r_{in} is calculated.

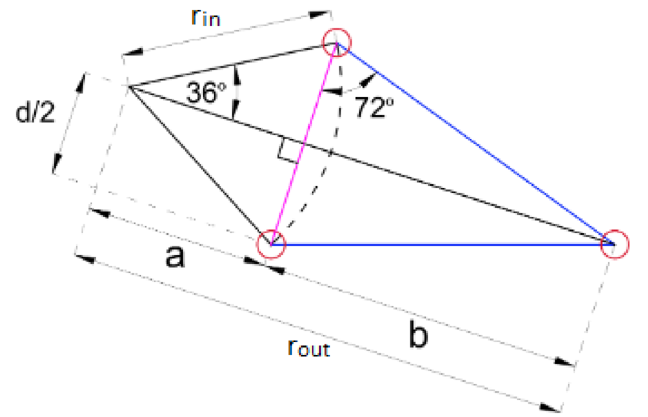

Fig. 4. Star Formation Points

In Figure 4, the value a indicates the height of the triangle formed inside the inner circle. b is the height of the triangle formed by the corner of the star. d is the distance between the points in the inner circle. These values are calculated according to equations 3-8.

$$\alpha = \frac{360}{5 * 2} = 36 \tag{3}$$

$$\beta = \frac{180 - 36}{2} = 72 \tag{4}$$

$$a = \frac{d}{2} cot(36) \tag{5}$$

$$b = \frac{d}{2} cot(72) \tag{6}$$

$$r_{out} = a + b \tag{7}$$

$$r_{in} = \frac{d}{2sin(36)} \tag{8}$$

After calculating the values of r_{in} and r_{out}, the ratio between the radius is obtained according to equation 9.

$$ratio = \frac{r_{out}}{r_{in}} = 2.618 \tag{9}$$

After the ratio between the radiuses is calculated, two circle formations are created. The radius of the circle to be formed on the outer side is calculated as 2.618 times bigger, and the starting angle is calculated as 36° bigger. After the circle points are calculated, the star formation points are calculated by combining them.

2) V Formation: In order to calculate the points required for the V formation, first, a circle is created that encompasses the V shape to be formed. Then the intervals of these three points are filled with points according to the desired distance parameter between the agents.V formation points are shown in Figure 5.

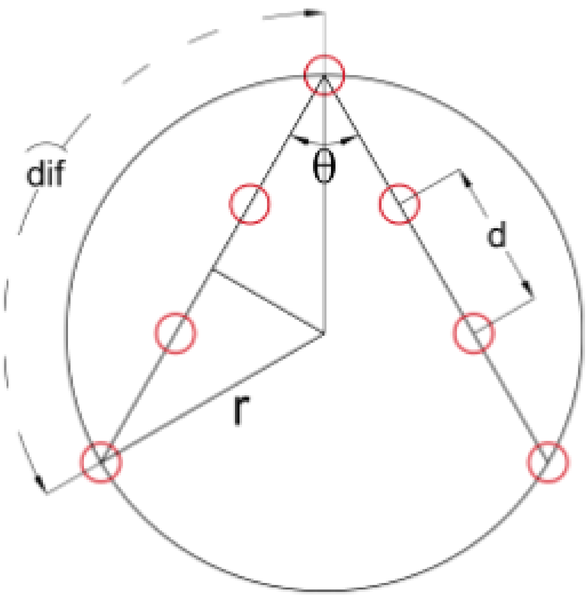

Fig. 5. V Formation Points

θ denotes the angle of the triangle forming the V-shape, d is the distance between the formation points denotes r is the radius of the circle enclosing the triangle formed as a result of the formation.

$$r = d\frac{intervalCount}{2cos(\frac{\theta}{2})} \tag{10}$$

To calculate the points in a circle, it is enough to calculate three angles.

$$diff = 180 - \theta \tag{11}$$

And the angles are:

(90 90+diff 90-diff)

The positions are formed as follows after the angle and difference values are found.

$$\begin{pmatrix} \mathrm{rcos(90)+x_0} & \mathrm{rsin(90)+y_0} & \mathrm{z_0} \\ \mathrm{rcos(90+diff)+x_0} & \mathrm{rsin(90+diff)+y_0} & \mathrm{z_0} \\ \mathrm{rcos(90-diff)+x_0} & \mathrm{rsin(90-diff)+y_0} & \mathrm{z_0} \end{pmatrix}$$

C. Assigning Formation Points

After the formation points are calculated, they need to be assigned to the robots in the swarm. Hungarian Algorithm, a simultaneous assignment algorithm, was used for this assignment process. In order to minimize the path taken by the robots in the swarm and reduce the probability of collision, the formation points should be assigned to the most suitable robot. For this reason, the cost for each robot to go to each formation point is calculated, and these values are collected in a matrix. The values in this cost matrix are calculated according to equations 12-13.

$$dist = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} \tag{12}$$

$$Cost_{ij} = dist^2 \tag{13}$$

After the cost matrix is calculated, this matrix is given as input to the Hungarian algorithm. The algorithm returns the required row and column indexes for the scenario with the least cost. By using these indexes, formation points are assigned to the relevant swarm robots.

D. Avoiding Obstacles

An artificial potential field strategy is used for robots to avoid obstacles while reaching the desired goal. According to the basic principle of this strategy, the workspace is under the influence of artificial potential forces. The positions of the obstacles and the positions of the agents have a repulsive effect, pushing the robot away from the obstacles, while the destination point, where the robot needs to go, has an attractive effect, pulling the robot towards the destination. Robot control is achieved by finding the sum of the repulsive and attractive forces (net force) at any point [29].

The total force is calculated as in Equation 14-16 [29].

$$F_{net} = F_{rep}(X_R) + F_{att}(X_R) \tag{14}$$

where F_{net} is net force,F_{rep} is repulsive force,F_{att} is attractive and X_R is robots' position. The repulsive force can be defined as,

$$\mathrm{F}_{rep}(X_R) = \begin{cases} d \leq d_m, & K_r \times (\frac{1}{(d(X_R - X_o) - d_0)^2} - \frac{1}{(d_m - d_0)^2}), \\ d > d_m, & 0 \end{cases} \tag{15}$$

where $d(X_R, X_o)$ is distance between robot and obtacle,d_m is distance treshold,d_0 is minimum safety distance from obstacles, K_r is repulsive potential field constant. The attractive force can be defined as,

$$F_{att}(X_R) = K_a \times d(X_R - X_G) \tag{16}$$

where $d(X_R - X_G)$ is distance from robot to goal point, K_a is attractive potential field constant. Figure 6 shows the robots moving towards the obstacle. Figure 7 shows the robots avoiding the obstacle.

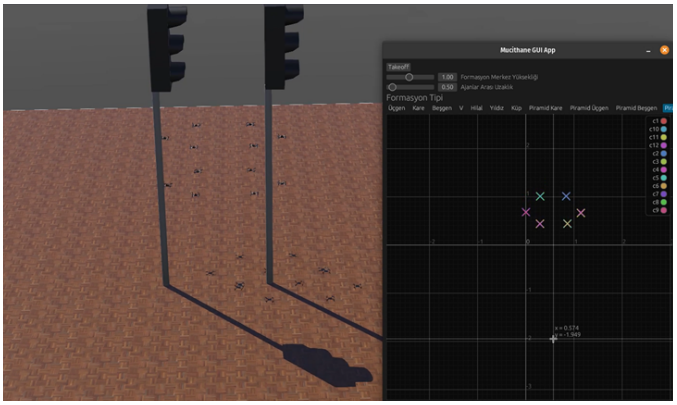

Fig. 6. Robots with Obstacles

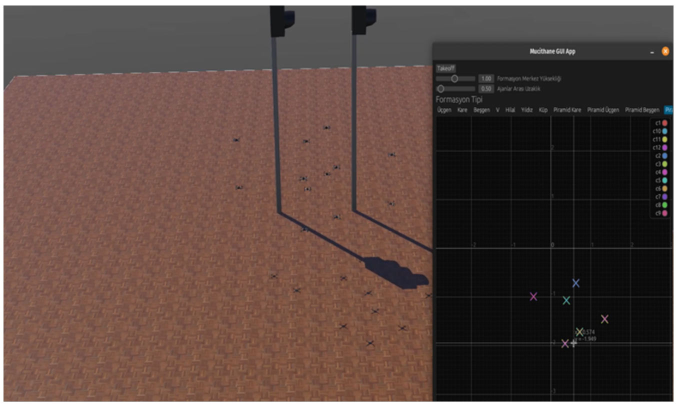

Fig. 7. Robots Avoiding Obstacles

IV. SIMULATION AND TEST

Webots simulation environment was used to test the swarm system. Crazyflie quadcopters were selected for swarm robots. The crazyflie firmware was used in the simulation to ensure that the crazyflie robots produce the most realistic output. Webots allows the development of separate plugins to use any robot with ROS2. The developed plugins work as a separate process by running each robot. The firmware for Crazyflie was developed using C programming language. However, the developers have provided the relevant links to use these codes in Python programming language. These links were used in the Webots plugin to generate the motor control outputs required for simulation.

Various test environments have been created in order to test the robots in the swarm according to different algorithms during the simulation process. Since the formation types are dependent on the number of robots in the swarm, environments with different numbers of robots were created. For testing the obstacle avoidance of the swarm, some static obstacles with known positions were added to some environments.

A. 2D Simulation Results

Three different formations were tested for the two-dimensional simulation test.

1) Square Formation: The simulation result is in Figure 8.

Fig. 8. Square Formation

2) Star Formation: The simulation result is in Figure 9.

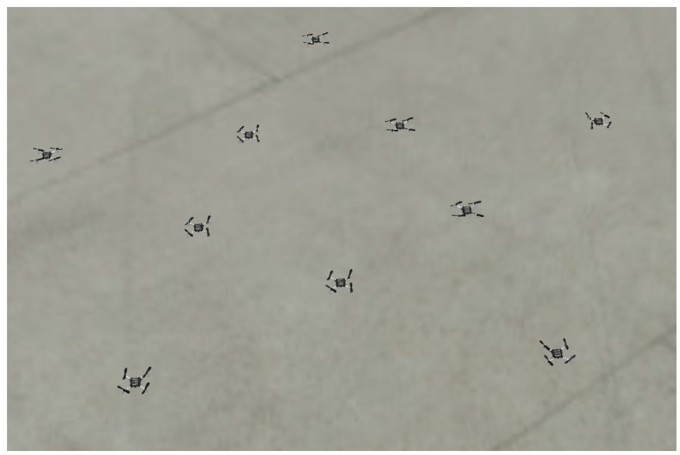

Fig. 9. Star Formation

3) V Formation: The simulation result is in Figure 10.

Fig. 10. V Formation

B. 3D Simulation Results

Three different formations were tested for the three-dimensional simulation test.

1) Cube Formation: The simulation result is in Figure 11.

Fig. 11. Cube Formation

2) Pyramid Formation: The simulation result is in Figure 12.

Fig. 12. Pyramid Formation

3) Hexagonal Prism Formation: The simulation result is in Figure 13.

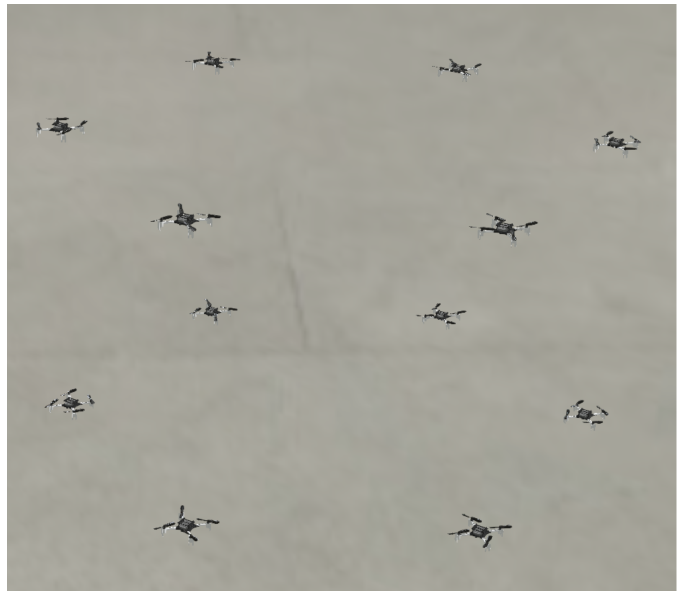

Fig. 13. Hexagonal Prism Formation

V. CONCLUSION

In this study, formation control of UAV swarm was realized. Formations are both two-dimensional and three-dimensional. All control commands of the robots come from a single location. In this respect, it has a centralized control method. The study was implemented in the ROS environment. Webots simulation environment was used to test the methods. Crazyflie robots were selected to create a swarm robot. There are static obstacles in the simulation environment where the experiments are performed. The artificial potential field method was used for obstacle avoidance. Two-dimensional formation shapes such as square, star and v formation were tested. Cube and pyramid formations were created as three-dimensional formations. When the simulation results were analyzed, it was observed that success was achieved. As an extension of this study, it is considered to manage robots not from a single center but with a distributed control method.

REFERENCES

[1] L. Bayındır, “A review of swarm robotics tasks,” *Neurocomputing*, vol. 172, pp. 292–321, 2016.

[2] E. Şahin, “Swarm robotics: From sources of inspiration to domains of application,” in *International workshop on swarm robotics*. Springer, 2004, pp. 10–20.

[3] Z. Jiao, Y. Zhang, J. Xin, L. Mu, Y. Yi, H. Liu, and D. Liu, “A deep learning based forest fire detection approach using uav and yolov3,” in *2019 1st International conference on industrial artificial intelligence (IAI)*. IEEE, 2019, pp. 1–5.

[4] N. Jayapandian, “Cloud enabled smart firefighting drone using internet of things,” in *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*. IEEE, 2019, pp. 1079–1083.

[5] M. Silvagni, A. Tonoli, E. Zenerino, and M. Chiaberge, “Multipurpose uav for search and rescue operations in mountain avalanche events,” *Geomatics, Natural Hazards and Risk*, vol. 8, no. 1, pp. 18–33, 2017.

[6] F. Tükenmez, “Harita mühendsiliğinde iha ile karayolu projelendirme,” *Türkiye Fotogrametri Dergisi*, vol. 3, no. 2, pp. 53–61, 2021.

[7] H. T. Do, H. T. Hua, M. T. Nguyen, C. V. Nguyen, H. T. Nguyen, H. T. Nguyen, and N. T. Nguyen, “Formation control algorithms for multiple-uavs: a comprehensive survey,” *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, vol. 8, no. 27, pp. e3–e3, 2021.

[8] G. Skorobogatov, C. Barrado, and E. Salami, “Multiple uav systems: A survey,” *Unmanned Systems*, vol. 8, no. 02, pp. 149–169, 2020.

[9] A. Saeed, A. Abdelkader, M. Khan, A. Neishaboori, K. A. Harras, and A. Mohamed, “On realistic target coverage by autonomous drones,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 15, no. 3, pp. 1–33, 2019.

[10] V. Sharma, H.-C. Chen, and R. Kumar, "Driver behaviour detection and vehicle rating using multi-uav coordinated vehicular networks," *Journal of Computer and System Sciences*, vol. 86, pp. 3–32, 2017.
[11] J. Scherer, S. Yahyanejad, S. Hayat, E. Yanmaz, T. Andre, A. Khan, V. Vukadinovic, C. Bettstetter, H. Hellwagner, and B. Rinner, "An autonomous multi-uav system for search and rescue," in *Proceedings of the first workshop on micro aerial vehicle networks, systems, and applications for civilian use*, 2015, pp. 33–38.
[12] J. Dong, E. Nelson, V. Indelman, N. Michael, and F. Dellaert, "Distributed real-time cooperative localization and mapping using an uncertainty-aware expectation maximization approach," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 5807–5814.
[13] B. D. Anderson, B. Fidan, C. Yu, and D. Walle, "Uav formation control: Theory and application," in *Recent advances in learning and control*. Springer, 2008, pp. 15–33.
[14] A. S. Brandao and M. Sarcinelli-Filho, "On the guidance of multiple UAV using a centralized formation control scheme and delaunay triangulation," *Journal of Intelligent Robotic Systems*, vol. 84, no. 1-4, pp. 397–413, Nov. 2015. [Online]. Available: https://doi.org/10.1007/s10846-015-0300-5
[15] D. Kim, H. Wang, G. Ye, and S. Shin, "Decentralized control of autonomous swarm systems using artificial potential functions: analytical design guidelines," in *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*. IEEE, 2004. [Online]. Available: https://doi.org/10.1109/cdc.2004.1428623
[16] Z. Li, W. Xue, D. Li, D. Lin, and J. Huang, "Distributed differential game for formation control of multi-uav with obstacle avoidance," in *2022 China Automation Congress (CAC)*. IEEE, 2022, pp. 4877–4882.
[17] H. S. Yavuz, H. Goktas, H. Cevikalp, and H. Saribas, "Optimal task allocation for multiple uavs," in *2020 28th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2020, pp. 1–4.
[18] T. Wang, Q. Dang, and P. Pan, "A multi-robot system based on a hybrid communication approach," *Studies in Media and Communication*, vol. 1, no. 1, pp. 91–100, 2013.
[19] M. A. Kamel, X. Yu, and Y. Zhang, "Formation control and coordination of multiple unmanned ground vehicles in normal and faulty situations: A review," *Annual reviews in control*, vol. 49, pp. 128–144, 2020.
[20] Q. Ouyang, Z. Wu, Y. Cong, and Z. Wang, "Formation control of unmanned aerial vehicle swarms: A comprehensive review," *Asian Journal of Control*, vol. 25, no. 1, pp. 570–593, 2023.
[21] N. H. Li and H. H. Liu, "Formation uav flight control using virtual structure and motion synchronization," in *2008 American Control Conference*. IEEE, 2008, pp. 1782–1787.
[22] X.-p. Xu, X.-t. Yan, W.-y. Yang, K. An, W. Huang, and Y. Wang, "Algorithms and applications of intelligent swarm cooperative control: A comprehensive survey," *Progress in Aerospace Sciences*, vol. 135, p. 100869, 2022.
[23] D. Xu, X. Zhang, Z. Zhu, C. Chen, P. Yang *et al.*, "Behavior-based formation control of swarm robots," *mathematical Problems in Engineering*, vol. 2014, 2014.
[24] L. Joseph, *Robot operating system (ros) for absolute beginners*. Springer, 2018.
[25] L. Joseph and J. Cacace, *Mastering ROS for Robotics Programming: Design, build, and simulate complex robots using the Robot Operating System*. Packt Publishing Ltd, 2018.
[26] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.
[27] W. Giernacki, M. Skwierczyński, W. Witwicki, P. Wroński, and P. Kozierski, "Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering," in *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*. IEEE, 2017, pp. 37–42.
[28] J. A. Preiss, W. Honig, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3299–3304.
[29] M. J. Hong and M. Arshad, "A balance-artificial potential field method for autonomous surface vessel navigation in unstructured riverine environment," *Procedia Computer Science*, vol. 76, pp. 198–202, 2015.

Abdulmelik Bekmez is currently studying in his bachelor's degree in the Department of Computer Engineering, Engineering Faculty, at the Fatih Sultan Mehmet Vakıf University. His current research areas include robotics and formation control.

Kadir Aram is research assistant in Computer Engineering department at Fatih Sultan Mehmet Vakıf University. He completed his PH.D. in Computer Science and Engineering at Istanbul Sabahattin Zaim University in 2023. His research interest are mobile robotics and natural language processing.

Burhanettin Can is Professor in Computer Engineering department at Fatih Sultan Mehmet Vakıf University. He completed his PH.D. Nuclear Technology at Istanbul Technical University in 1984. His research interest are robotics and fuzzy logic.