European Journal of Technique

journal homepage: https://dergipark.org.tr/en/pub/ejt

Vol.13, No.2, 2023

Research Article

Leader-Follower Based Formation Control of Heterogeneous UAV-UGV Multi-Agent System

Kadir Aram[1*], Abdulmelik Bekmez[2]

[1*] Fatih Sultan Mehmet Vakıf University, Computer Engineering Department, 34445, Istanbul, Turkey. (e-mail: karam@fsm.edu.tr).
[2] Fatih Sultan Mehmet Vakıf University, Computer Engineering Department, 34445, Istanbul, Turkey. (e-mail: abdulmelik.bekmez@gmail.com).

ARTICLE INFO

Received: Aug., 26. 2023
Revised: Sep., 01. 2023
Accepted: Sep, 05. 2023

Keywords:
Formation Control
Leader-Follower
Robotics
Heterogeneous Swarm
Mobile Robot

Corresponding author: *Kadir Aram*

ISSN: *2536-5010* | e-ISSN: *2536-5134*

DOI: https://doi.org/10.36222/ejt.1350641

ABSTRACT

This paper deals with a leader-follower formation control of a heterogeneous robot swarm comprising unmanned ground vehicles (UGV) and unmanned aerial vehicles (UAV). The UGV is the leader robot, and the UAV's are the followers. A centralized system receives information about the robots, and the system makes decisions concerning the robots. Two different approaches were used to implement the leader-follower formation strategy. In the first approach, the formation points are calculated according to the position of the leader. In the second one, the position averages of the follower robots are also taken into account. The robots form a V-shape formation and are assigned to the formation points using the Hungarian algorithm. The robots move to the formation points with a proportional controller. The system was developed within the ROS2 framework and employed Turtlebot3 and Crazyflie robots for the robot swarm. The study was conducted in the Webots simulation environment, encompassing a variety of tests, with the subsequent observation and examination of the obtained results.

1. INTRODUCTION

An unmanned system encompasses a mechanical or apparatus outfitted with essential data processing components, sensors, automated control mechanisms, and communication systems, enabling it to independently carry out missions without human interference. Such systems encompass a variety of unmanned aerial vehicles (UAV), unmanned ground vehicles (UGV), underwater exploratory devices [1].

An unmanned aerial vehicle (UAV) is an aircraft capable of flight without the presence of a human pilot on board. In contemporary times, an increasing number of UAVs are being employed in civilian contexts due to their exceptional mobility and adaptability [2]. This attribute has enabled them to accomplish their objectives across numerous applications effectively. Nonetheless, incorporating multiple vehicles offers heightened versatility and efficacy in task execution. Moreover, utilizing multiple vehicles enhances resilience against failures compared to solitary units [3]. Studies on using multiple UAVs have been conducted in many areas, such as search and rescue [4],perimeter,surveillance [5] and loads carrying [6].

Agents exhibiting distinct dynamic attributes can surmount individual limitations, effectively accomplishing intricate and multifaceted missions. This elevation in capability expands the scope for tackling more challenging applications. Notably, the synergistic integration of UAVs and UGVs amalgamates their strengths, encompassing proficient payload capacity, versatile task configuration, and robust localization capabilities, culminating in heightened overall performance [7].

In previous studies, various strategies have been applied for the formation control of multiple robots. Some key ones include leader-follower, behavior-based and virtual formation structure strategies [8].

In the leader-follower approach, one or more robots are considered as leaders while the other robots are considered as followers. The leader robot moves towards a specific goal. The follower robots move by maintaining a set distance and orientation from the leader. While providing this movement, it receives the position and orientation information of the leader robot [9,10]. In the virtual structure approach, robots move by creating a rigid structure. Geometrical shapes can be applied here. This formation has a center determined by shape, speed and orientation. The positions of the robots are defined relative to this reference point. Since the reference point is given relative to a trajectory, the position of each robot must be recalculated as time passes [11].The behavioral control strategy is about each robot using certain behaviors. These behaviors can be trajectory following, obstacle avoidance and formation maintenance. After the relative weighting of these behaviors, the final control is performed [12]. Zhou et al. investigated the time-varying formation tracking problem for a heterogeneous

UAV-UGV swarm system. First, a collaborative control model is constructed with algebraic graph theory, and then a distributed observer-based formation tracking control protocol is designed [13]. In another study, an UAV lands on an UGV after delivering a package. A virtual structure approach is used here. Within this structure, there is a controller for the UGV to avoid obstacles as it moves forward [14]. Li and Zhu presented a UAV-UGV cooperative control mechanism. A leader-follower strategy is used for cooperative trajectory tracking. They used a fuzzy robust controller to control the UAV. The controller of the UGV uses a tracking algorithm and a PID controller[15]. Harik et al. designed a system for object transportation in unsafe locations. A UAV acts as a guide for obstacles. A group of UGVs performs the task using information from the UAV. The leader receives the information from the UAV and navigates while the follower robots follow the leader at a given distance using a vision-based target-tracking controller [16]. In their study, Akın and Şahin examined how UAVs and UGVs can efficiently collect data from IoT devices. They conducted these experiments in an obstacle environment. They used reinforcement learning principles to solve these problems [17].

This study presents a formation control of a heterogeneous multi-robot system consisting of one Turtlebot 3 as UGV and multiple Crazyflie robots as UAVs. The proposed work is a centralized system that utilizes a leader-follower strategy to implement the desired formation. PID controller is used for the control of the robots. The study was tested in a simulation environment. In Section 2, the methodology and materials used are described in detail. Section 3 presents the experiments done in the test phase and the simulation results. Finally, in Section 4, the results of the experiments are evaluated, and a conclusion to the study is made.

2. MATERIALS AND METHODS

The study consists of three stages. The first is for the leader robot to go to the target point, the second is to ensure that the follower robots follow the leader, and the third is to maintain the formation of the multi-robot group. V formation shape was chosen for multi-robots. The application was realized in the Webots Simulation environment using ROS.

2.1. Robot Operating System (ROS)

ROS is an open-source framework that enables the development of robotic applications with the help of libraries and packages. It is used in both commercial and research activities [18]. ROS allows application development using different programming languages. Also, an implemented program part can be used in other applications [19]. Processes in ROS communicate with each other, and these processes, called nodes, communicate using a publisher/subscriber structure. These nodes send data to each other using messages. The nodes that send data are called publisher nodes, and they send messages through topics to the receiver nodes, which are called subscribers. The ROS master is responsible for the nodes to locate each other, and it is initialized at startup to provide communication.[20]. In ROS, different packages are offered to users to perform certain operations like mapping and navigation. For example, the move base package handles the operations to move a robot to a given destination point [21]. ROS includes a practical tool called Rviz (Ros Visualizer), which is a tool for visualizing robots, sensors, and algorithms in three dimensions. It can be used for all types of robots. Rviz can plot data streaming on a ROS system, and its panel can be configured for various applications [22].

2.2. Webots

Webots is a simulation environment used both academic and industrial settings working on robots. Three-dimensional environments can be modeled using Webots, and robots defined in it can be used in these environments. It also has libraries containing sensors, actuators and other materials. Robot designs can be made using these libraries [23,24]. Figure 1 shows an example simulation environment.

Figure 1. Simulation environment

Figure 2 shows a Rviz visualization of the environment.

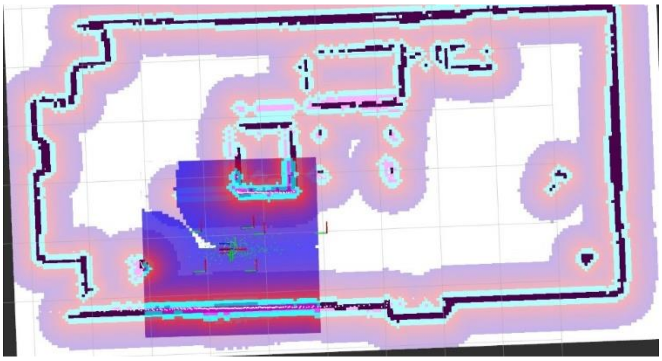

Figure 2. Rviz Visualization

2.3. Robots

In the study, simulation was performed with two different robots. One mobile robot and a multi-UAV group were used in the study.

The Turtlebot3 burger model was used as a mobile robot in the study. Turtlebot is a ROS based mobile robot used in both research and education. Many packages are provided with ROS to run simulations and control the robot[25]. Turtlebot is used as a ground robot and acts as the leader robot in the application.

In this study, the Crazyflie quadrotor robot was preferred as the follower robot group. Crazyflie is a robot platform used for educational and research purposes in robotics. With its small size and low weight, the robot is preferred in swarm robot applications[26,27]. It was preferred because it is compatible with ROS and because it has a ready-made model in the simulation.

2.4. Leader-Follower Strategy

The strategy aims for the leader robot to move to a given target location while other follower robots follow it. There must be continuous communication between the leader and the follower. The leader continuously broadcasts its position and orientation information with the data it receives from its sensors. The follower robots receive this information from the leader robot. The follower robots use the leader's position to follow the leader and navigate a given formation. The target of the follower robots is the current position of the leader. Its orientation is the leader's orientation. The distance between the leader robot and the followers is continuously maintained during the tracking process [28–31]. Figure 3 shows the general structure of the leader-follower strategy.

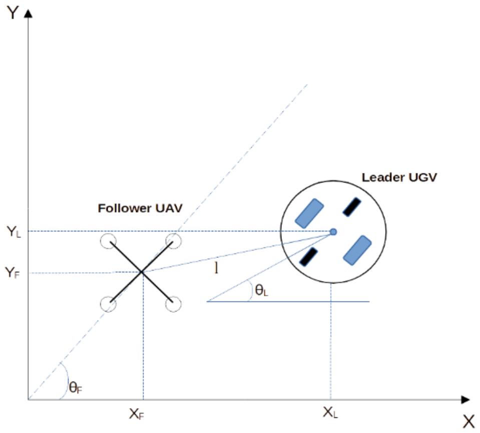

Figure 3. Leader-Follower Formation Scheme

In figure X_L, Y_L is the leader's position, θ_L is the leader's orientation, X_F, Y_F is the follower robot's position, and θ_F is the follower robot's orientation. l is the distance between the leader and the follower. The leader-follower strategy aims to maintain the desired distance and relative bearing between the robots.

The distance between the leader robot and the follower is calculated as in Equation 1.

$$distance\ (l) = \sqrt{(X_L - X_F)^2 + (Y_L - Y_F)^2} \quad (1)$$

In the ROS environment, the algorithm works as follows. The leader robot continuously broadcasts its instantaneous position as a publisher on the way to the specified target. The follower robots receive the leader's position as subscribers. For the follower robots, the target position is the leader's current position, and the target orientation is the current orientation of the leader. The distance between the follower and the leader is calculated by Equation 1. This process is repeated until the desired target point is reached.

2.5. *Formation Process*

The formation of the robots takes place in three steps. First, formation points should be calculated using the desired shape and the number of robots in the swarm. Then, which robot in the swarm should go to these points should be determined. In the last step, the robots should reach this point and maintain the formation. In this study, a V-formation shape was used.

The leader robot is located at the end of the V shape in the formation structure. Follower robots are lined up to the right and left of the leader robot. While forming formation points, the distance between the followers and the angle value they should be placed to the right and left are determined. The structure of the V formation is shown in Figure 4.

We used two different approaches to calculate the formation points. In the first approach, the rotation vector of the leader robot was taken and rotated 180 degrees. In order to find the direction vectors indicating the wings of the V formation to be created, the leader's vector is expanded to the right and left by the theta angle. The desired distance value between these direction vectors and formation points is multiplied by the positions where the robots will be placed.

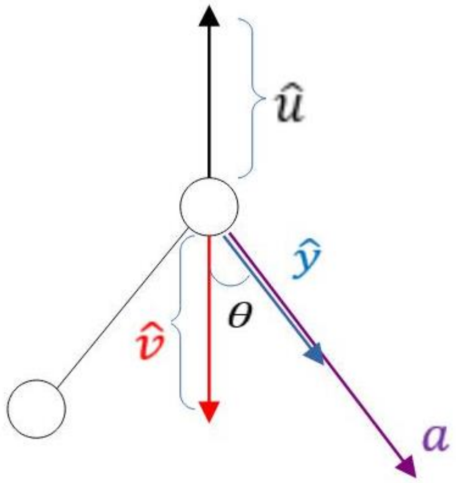

Figure 4. V- Formation Points

In the figure, θ is the angle of the triangle forming the V formation shape, $\hat{u}$ is the leader's rotation vector, $\hat{v}$ is the leader's rotation vector rotated 180 degrees, $\hat{y}$ is the vector value extended by the angle θ, and a is the formation point. The formation point a is calculated according to Equation 2.

$$a = \hat{v} \times d \quad (2)$$

Where d is the distance between formation points.

In the second approach, the leader's rotation vector is not used. Instead, the average point of the positions of the follower robots was first determined. Then, the rotation between this and the leader's position is taken. Thus, the v value in equation 2 is calculated according to equations 3 and 4.

$$p_{avg} = \frac{p_1 + p_2 + \cdots + p_n}{n} \quad (3)$$

$$\hat{v}_u = \frac{p_l - p_{avg}}{|p_l - p_{avg}|} \quad (4)$$

Where;

p_{avg}: Average position value of follower robots.
$p_1 \ldots p_n$: Position of follower robots.
p_l: Position of leader robot.
$\hat{v}_u$: unit vector of the orientation of the formation

Once the formation positions are determined, the appropriate robot for each position needs to be identified. Hungarian algorithm was used for the assignment of positions. The first objective is to ensure that the robots take the shortest path to the positions. Another objective is to minimize collisions. Before starting the algorithm, the distance each robot needs to take for each position is calculated. The matrix containing these calculations is called the Cost matrix. The matrix is filled according to Equations 5-6.

$$distance = \sqrt{\left(X_{Ri} - X_{pj}\right)^2 + \left(Y_{Ri} - Y_{pj}\right)^2} \quad (5)$$

$$Cost_{ij} = distance^2 \quad (6)$$

Where (X_{Ri}, Y_{Ri}) is the position of the robot whose distance to the formation point will be calculated, and $(X_{pj}$, Y_pj) is the position of that formation point. After the cost matrix is filled, the algorithm process starts. These values are taken as input to the algorithm. As a result, the algorithm returns the row and column indices with the lowest cost. With these values, each formation point is assigned to the relevant robot in the robot group.

2.6. Controller Design

After determining the positions in the desired formation, which robots will go to these positions is determined. For the robot to go to this position, it must turn there and drive forward. For this process, a proportional controller was used due to the advantages of its applicability. The pseudo-code of the controller is given in Algorithm 1.

Algorithm 1 Controller Pseudo Code

```
Determine the reference point
Determine the $K_p$ coefficient
while reference do
    calculate distance to reference
    calculate the angle difference from the reference point.
    determine the angular and linear velocity by multiplying the error values by the coefficients
end while
```

The controller takes the position of the target point as the reference value. The distance to the reference position is calculated according to Equation 3, and the angle difference is calculated according to Equations 7 and 8.

$$\theta_{refi} = \arctan\left(\frac{y_T - y_R}{x_T - x_R}\right) \tag{7}$$

$$\theta_e = \theta_{refi} - \theta_R \tag{8}$$

θ_{refi}: The angle between the robot and the target point
x_R, y_R: Position of the robot
x_T, y_T: Target position
θ_R: Orientation of the robot
θ_e: Orientation error.

At the controller output, the linear and angular velocity of the robot is calculated, and it is ensured to go to the desired position.

An artificial potential field strategy was used to prevent the robots from bumping into each other as they move to their positions. According to this strategy, the robots are the repulsive force, and the target point is the attractive force [32].

The steps to create the formation are shown in the flow chart in Figure 5. First, the number of robots that will form the shape of the formation is determined, and the positions are calculated according to this number. Then, a cost matrix containing the distances of the robots to these points is created. According to this matrix, each robot is assigned a position to move. Finally, the robots are driven to these positions.

3. EXPERIMENTS

The study was conducted on Ubuntu 22.04 using the ROS 2 Humble version. In order to test the techniques used in the study, an environment with simulation was established. Webots was used as the simulation environment, and robots were moved in this environment. Webots allows each robot to develop separate plugins that appear as a different process when running. Crazyswarm2, which was developed based on Crazyswarm[33], was used while performing the ROS 2 integration of the application. Crazyswarm2 was also used in the communication layer. Both ground and aerial robots were used in the application. Turtlebot 3 was used as a ground robot, and Crazyflie robots were used as aerial robots in the formation.

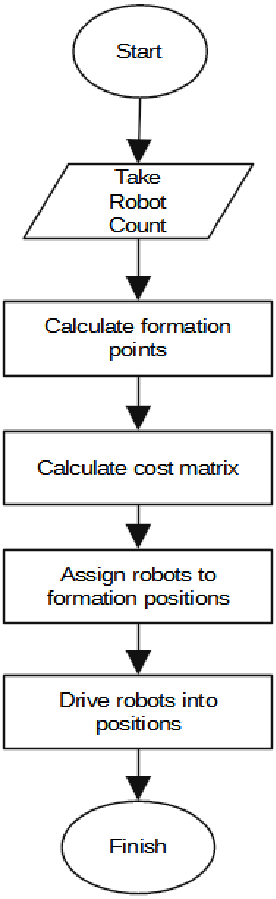

Figure 5. Formation Flow Chart

Two different scenarios were used in the developed environment. The formation process for each scenario was carried out using both approaches. At the beginning of the scenarios, the robots form a V-shape with the leader robot at the top of the formation. Here, the leader robot is tasked to reach a desired point. As the robot moves towards the target, the follower aerial robots follow it in the desired formation. These processes were confirmed by examining different graphs. In the first scenario, the starting position of the leader robot after the formation was (6.3, -4.3).

The leader robot moved to (8.7, -3.5), and the other robots followed it. The appearance of the robots in their initial positions is shown in Figure 6.

As seen in Figure 6, the robots wait at the starting point by forming the desired formation. The path graph of the robots is shown in Figure 7.

Figure 7 shows that the robots move from the starting point to the endpoint by maintaining the formation. The graph of the position errors of the follower robots as they move along the path is shown in Figure 8.

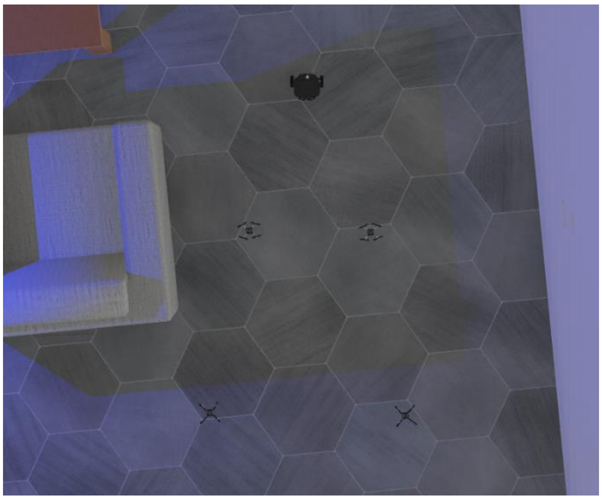

Figure 6. Initial Positions of Robots

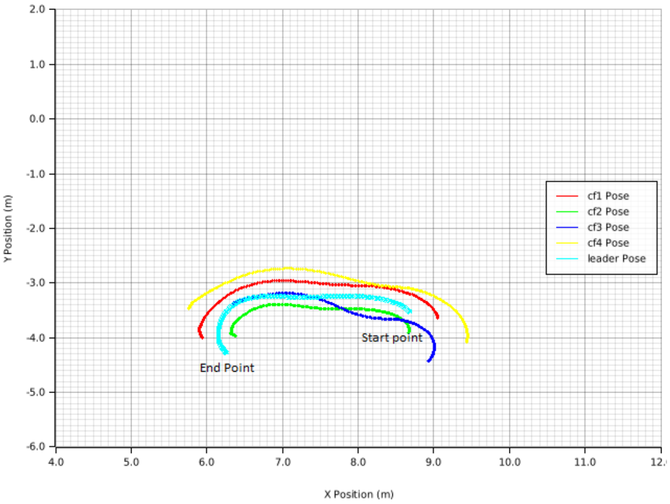

Figure 7. Scenario1 the path Robots Follow for Approach 1

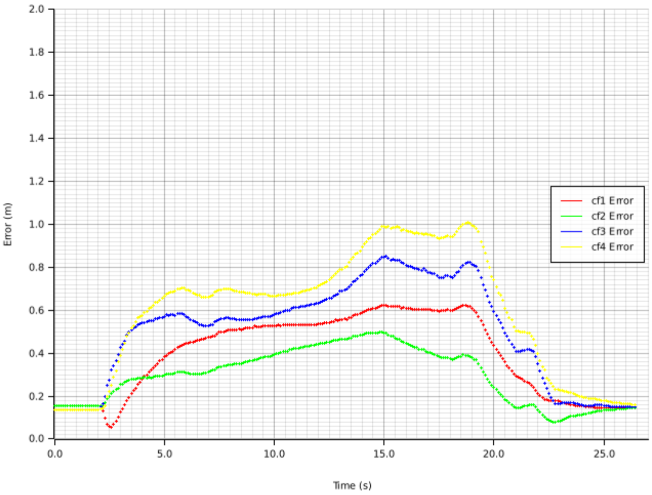

Figure 8. Scenario1 Position Error of Followers for Approach 1

The error of about 0.17, consistently observed before and after the navigation starts and ends, is due to the robots' use of the artificial potential field algorithm. The error rate increases steadily in the navigation process because the Turtlebot 3 is more agile and faster. During the rotation, there are significant jumps in the error due to the increase in the difference between the newly calculated formation points and the old ones. The final position of robots shown in Figure 9.

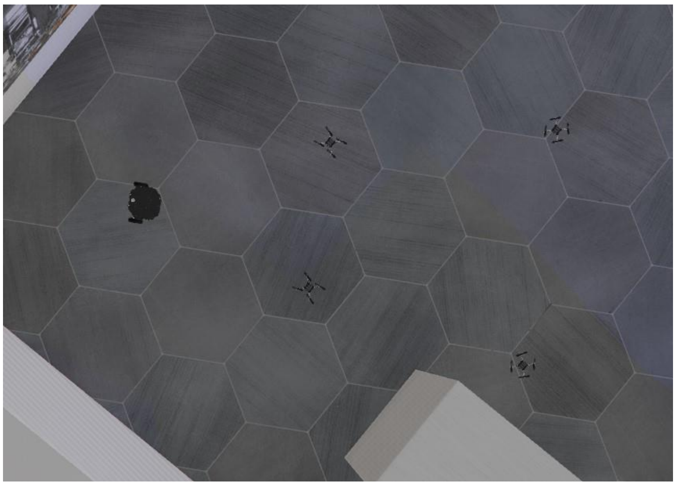

Figure 9. Final Positions of Robots

A similar experiment was performed for the second approach. The graph of the path followed by the robots is given in Figure 10.

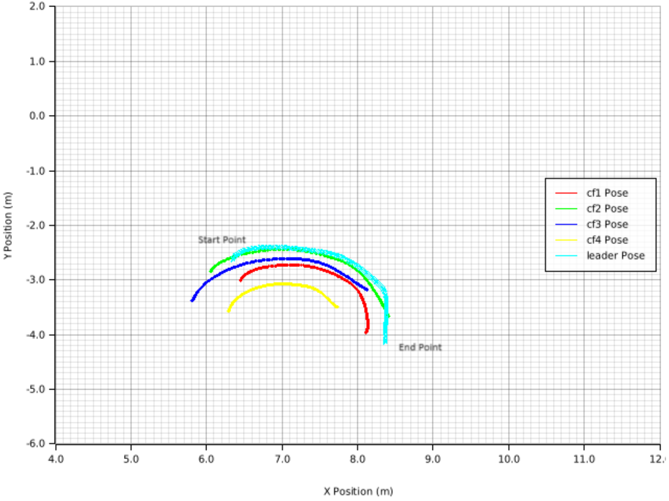

Figure 10. Scenario1 The path robots follow for Approach 2

The position errors of the follower robots are shown in Figure 11.

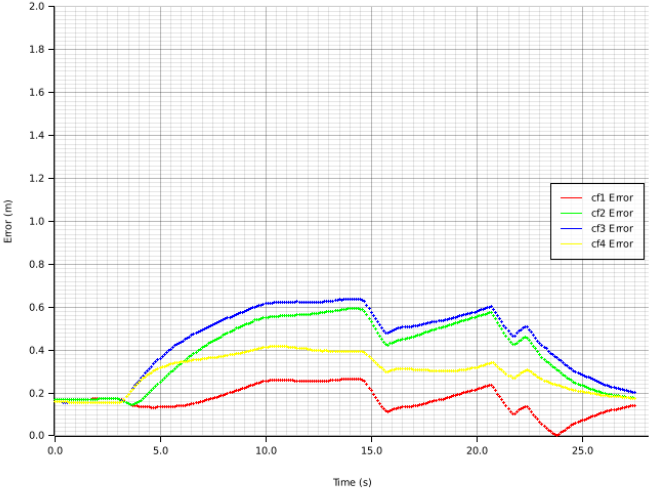

Figure 11. Scenario1 Position Error of Followers for Approach 2

Figure 11 shows that the position error decreased. In the second approach, the leader's changing orientation less affects the formation structure.

In the second scenario, robots were requested to follow different paths in the same environment. The starting and target points of the robots were changed, and their path and position errors were analyzed. Again, while moving along this path, the position errors increase, especially where the leader robot makes turns. In this scenario, unlike the first one, Rviz images of the start and end positions of the robots are also added. The path graph of the robots is shown in Figure 12.

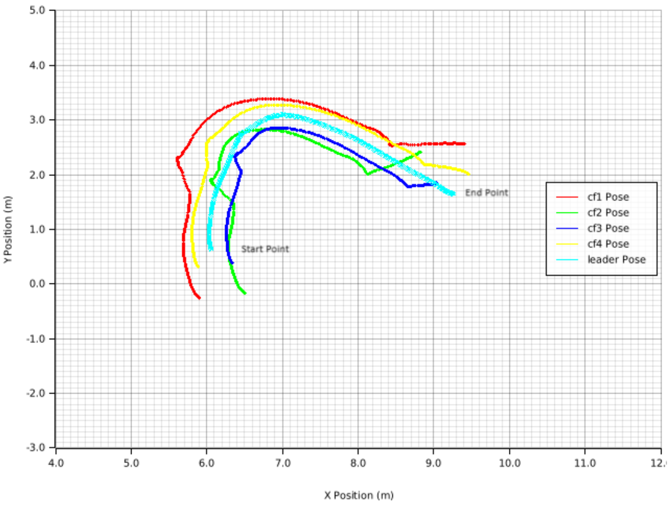

Figure 12. Scenario 2 The Path Robots Follow for Approach 1

For Scenario 2, the graph of the position errors of the follower robots as they move along the path is shown in Figure 12.

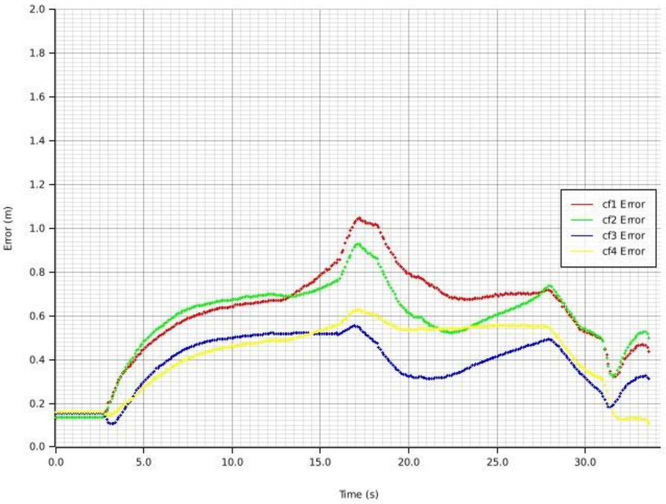

Figure 12. Scenario 2 Position Error of Followers for Approach 1

The Rviz image of the initial positions of the robots is shown in Figure 13. And the Rviz image of the final positions of the robots is shown in Figure 14.

For approach 2, the path graph of the robots is shown in Figure 15. The position errors along the path of the follower robots are shown in Figure 16. Examining the outcomes from this particular scenario, it becomes evident that the second approach yields significantly more favorable results.

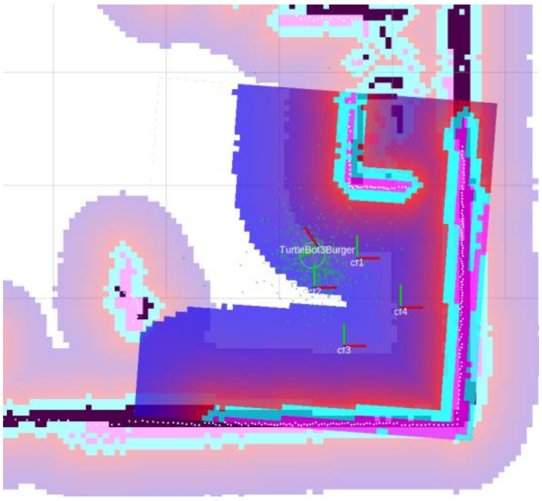

Figure 13. Rviz Image of Initial Positions of Robots

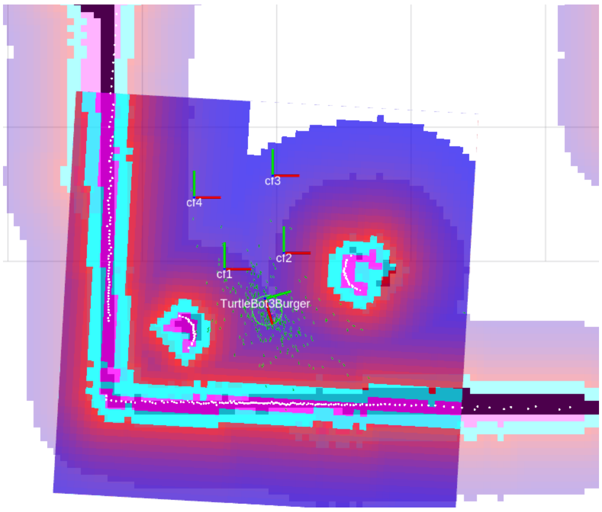

Figure 14. Rviz Image of Final Positions of Robots

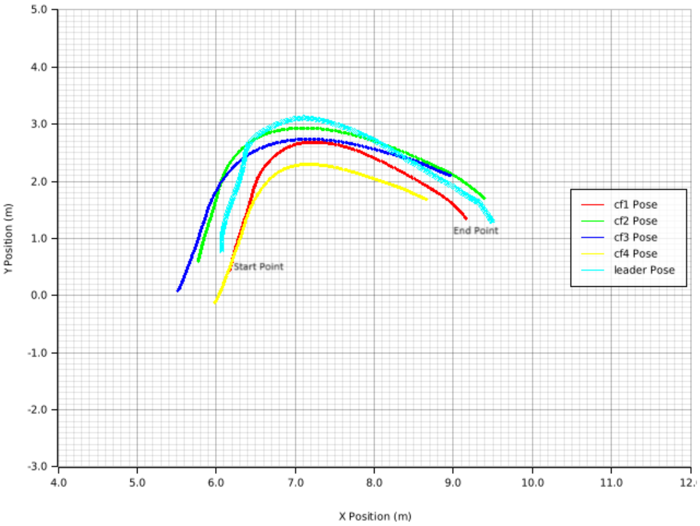

Figure 15. Scenario 2 The Path Robots Follow for Approach 2

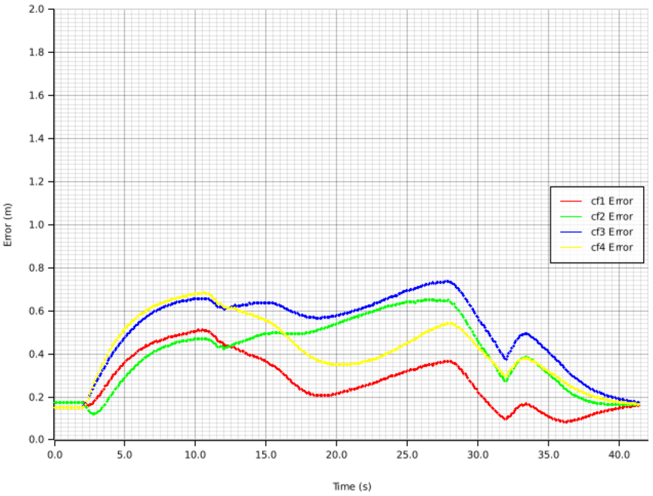

Figure 16. Scenario 2 Position Error of Followers for Approach 2

4. CONCLUSION

This study focused on implementing formation control for a heterogeneous multi-robot ensemble comprising ground and aerial robots. The strategy employed for formation was the leader-follower approach, wherein a ground robot assumed the role of the leader while the aerial robots operated as followers. A centralized control system was adopted to oversee the entirety of the study. Data received from the robot group were aggregated at a central point, where subsequent decisions were rendered. The formation configuration adopted for the robots was a V-shaped pattern. Following the determination of formation points based on the number of robots, the allocation of robots to their respective positions was achieved by applying the Hungarian algorithm. The experimental evaluation of this study was conducted within the Webots simulation environment, using Turtlebot3 and Crazyflie robots as the testbed.

During the experimental phase conducted within the test environment, an equal number of robots were deployed for each approach, with variations introduced into the traversed paths. Upon a comprehensive examination of the path trajectories and position errors associated with each approach, it becomes evident that both strategies effectively preserve the desired formation. However, distinctions emerge in their responses to specific factors. In the first approach, the formation tends to show more sensitivity when the leader robot makes a rotational movement. The second approach, characterized by a more centrally weighted reference point, displays a higher resilience in the face of such rotational deviations. Analyzing the position error data further reinforces the notion that the second approach surpasses the first in performance. Different formation strategies can be applied in future studies, and a distributed system can be preferred instead of a centralized one. Furthermore, different formation shapes can be applied in future research work beyond the V formation.

REFERENCES

[1] S. Çaşka and A. Gayretli, “A survey of UAV/UGV collaborative systems,” *CIE44 IMSS*, vol. 14, pp. 453–463, 2014.

[2] Y. Lu, Z. Xue, G.-S. Xia, and L. Zhang, “A survey on vision-based UAV navigation,” *Geo-spatial Inf. Sci.*, vol. 21, no. 1, pp. 21–32, 2018.

[3] K. A. Ghamry and Y. Zhang, “Formation control of multiple quadrotors based on leader-follower method,” in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015, pp. 1037–1042.

[4] Y. Tian *et al.*, “Search and rescue under the forest canopy using multiple UAVs,” *Int. J. Rob. Res.*, vol. 39, no. 10–11, pp. 1201–1221, 2020.

[5] D. Kingston, R. W. Beard, and R. S. Holt, “Decentralized perimeter surveillance using a team of UAVs,” *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1394–1404, 2008.

[6] R. Ritz and R. D’Andrea, “Carrying a flexible payload with multiple flying vehicles,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 3465–3471.

[7] W. Cheng, B. Jiang, K. Zhang, and S. X. Ding, “Robust finite-time cooperative formation control of UGV-UAV with model uncertainties and actuator faults,” *J. Franklin Inst.*, vol. 358, no. 17, pp. 8811–8837, 2021.

[8] Z. A. Ali, A. Israr, E. H. Alkhammash, and M. Hadjouni, “A leader-follower formation control of multi-UAVs via an adaptive hybrid controller,” *Complexity*, vol. 2021, pp. 1–16, 2021.

[9] R. Rafifandi, D. L. Asri, E. Ekawati, and E. M. Budi, “Leader--follower formation control of two quadrotor UAVs,” *SN Appl. Sci.*, vol. 1, pp. 1–12, 2019.

[10] E. dos Santos Cardoso and V. Bacheti, “Package Delivery Based on the Leader-Follower Control Paradigm for Multirobot Systems.”

[11] N. H. M. Li and H. H. T. Liu, “Formation UAV flight control using virtual structure and motion synchronization,” in *2008 American Control Conference*, 2008, pp. 1782–1787.

[12] Y. Zhang and H. Mehrjerdi, “A survey on multiple unmanned vehicles formation control and coordination: Normal and fault situations,” in *2013 International conference on unmanned aircraft systems (ICUAS)*, 2013, pp. 1087–1096.

[13] S. Zhou, X. Dong, Q. Li, and Z. Ren, “Time-varying formation tracking control for UAV-UGV heterogeneous swarm systems with switching directed topologies,” in *2020 IEEE 16th International Conference on Control \& Automation (ICCA)*, 2020, pp. 1068–1073.

[14] V. P. Bacheti, A. S. Brandao, and M. Sarcinelli-Filho, “A path-following controller for a uav-ugv formation performing the final step of last-mile-delivery,” *IEEE Access*, vol. 9, pp. 142218–142231, 2021.

[15] Y. Li and X. Zhu, “Design and testing of cooperative motion controller for UAV-UGV system,” *Mechatronics Intell. Transp. Syst.*, 2022.

[16] E. H. C. Harik, F. Guérin, F. Guinand, J.-F. Brethé, and H. Pelvillain, “UAV-UGV cooperation for objects transportation in an industrial area,” in *2015 IEEE International Conference on Industrial Technology (ICIT)*, 2015, pp. 547–552. doi: 10.1109/ICIT.2015.7125156.

[17] E. Akın and Y. Şahin, “Q-Learning Based Obstacle Avoidance Data Harvesting Model Using UAV and UGV,” *Eur. J. Tech.*, vol. 13, no. 1, pp. 54–60.

[18] K. Aram, G. Erdemir, and B. Can, “Türkçe Dogal Dil Isleme Ile Ozerk Gezgin Robotun Yol Planlamasi, Path Planning of an Autonomous Mobile Robot by Turkish Natural Language Processing.”

[19] M. Quigley *et al.*, “ROS: an open-source Robot Operating System,” in *ICRA workshop on open source software*, 2009, vol. 3, no. 3.2, p. 5.

[20] 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.

[21] Z. Tüfekçi and G. Erdemir, “Experimental Comparison of Global Planners for Trajectory Planning of Mobile Robots in an Unknown Environment with Dynamic Obstacles,” in *2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2023, pp. 1–6.

[22] M. Quigley, B. Gerkey, and W. D. Smart, *Programming Robots with ROS: a practical introduction to the Robot Operating System*. “ O’Reilly Media, Inc.,” 2015.

[23] O. Michel, “Cyberbotics ltd. webots$^{\text{TM}}$: professional mobile robot simulation,” *Int. J. Adv. Robot. Syst.*, vol. 1, no. 1, p. 5, 2004.

[24] A. Farley, J. Wang, and J. A. Marshall, “How to pick a mobile robot simulator: A quantitative comparison of CoppeliaSim, Gazebo, MORSE and Webots with a focus on accuracy of motion,” *Simul. Model. Pract. Theory*, vol. 120, p. 102629, 2022.

[25] C. Williams and A. Schroeder, “Utilizing ROS 1 and the Turtlebot3 in a Multi-Robot System,” 2020.

[26] W. Hönig and N. Ayanian, “Flying multiple UAVs using ROS,” *Robot Oper. Syst. Complet. Ref. (Volume 2)*, pp. 83–118, 2017.

[27] W. Giernacki, M. Skwierczyński, W. Witwicki, and P. Wroński

Paweland 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)*, 2017, pp. 37–42.

[28] Y. Liang, D. Qi, and Z. Yanjie, “Adaptive leader--follower formation control for swarms of unmanned aerial vehicles with motion constraints and unknown disturbances,” *Chinese J. Aeronaut.*, vol. 33, no. 11, pp. 2972–2988, 2020.

[29] J. Luo, C.-L. Liu, and F. Liu, “A leader-following formation control of multiple mobile robots with obstacle,” in *2015 IEEE International Conference on Information and Automation*, 2015, pp. 2153–2158.

[30] V. Walter, N. Staub, A. Franchi, and M. Saska, “Uvdar system for visual relative localization with application to leader--follower formations of multirotor uavs,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2637–2644, 2019.

[31] Y. Ziquan, Y. Zhang, B. Jiang, F. U. Jun, and J. I. N. Ying, “A review on fault-tolerant cooperative control of multiple unmanned aerial vehicles,” *Chinese J. Aeronaut.*, vol. 35, no. 1, pp. 1–18, 2022.

[32] M. J. Hong and M. R. Arshad, “A balance-artificial potential field method for autonomous surface vessel navigation in unstructured riverine environment,” *Procedia Comput. Sci.*, vol. 76, pp. 198–202, 2015.

[33] 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)*, 2017, pp. 3299–3304.

BIOGRAPHIES

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

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.