

Development of a Mission Planning and Obstacle Avoidance Algorithm for Unmanned Aerial Vehicle (UAV) using LIDAR and Potential Field Method within ROS-Gazebo

Mary Antonette C. Montalvo, Khynan Kyle Batoy and Cherry Mae G. Villame, *Member, IAENG*

Abstract—This study focuses on the development of a mission planning and obstacle avoidance algorithm for Unmanned Aerial Vehicle (UAV) in Search and Rescue (SAR) operation replacing the traditional search and rescue methods that has limited rescuers' ability to reach remote and inaccessible areas during times of disaster. The study has utilized an autonomous search and rescue UAV using Lidar sensors for the development of the obstacle avoidance algorithm, potential field method for mission planning, and YOLOv3 for human recognition system, all of which are implemented in the Robot Operating System (ROS) and Gazebo environments. Ardupilot was used to navigate through unknown and dynamic environments and detect the presence of humans within its vicinity. The virtual potential field method is a technique that uses Lidar sensor data to model the environment as a potential field. The authors used a ROS-Gazebo simulator to test and verify the performance of the UAV in a virtual environment. The experimental results of the study showed that the UAV located 86.67% of the human models and had 83.33% success in completing the mission.

Index Terms—Lidar sensors, mission planning, obstacle avoidance, potential field method, ros-gazebo, unmanned aerial vehicle.

I. INTRODUCTION

Search and rescue (SAR) refers to organized efforts to locate and assist people who are in distress or imminent danger. SAR operations are often carried out in response to natural disasters, such as earthquakes, hurricanes, and floods, as well as man-made disasters, such as plane crashes and terrorist attacks. In the Philippines, SAR is an important part of the country's disaster response efforts, as the country is prone to a variety of natural disasters due to its location in the Pacific Ring of Fire.

Manuscript received July 10, 2024; revised January 18, 2025.

Mary Antonette C. Montalvo is a postgraduate student in the Department of Computer Engineering and Mechatronics (DCEM) of the Mindanao State University-Iligan Institute of Technology (MSUIIT), Iligan City, Philippines. (e-mail: maryantonette.montalvo@g.msuiit.edu.ph).

Khynan Kyle Batoy is a postgraduate student in the Department of Computer Engineering and Mechatronics (DCEM) of the Mindanao State University-Iligan Institute of Technology (MSUIIT), Iligan City, Philippines. (e-mail: khynankyle.batoy@g.msuiit.edu.ph).

Cherry Mae G. Villame is a professor of the BS in Computer Engineering Program in the Department of Computer Engineering and Mechatronics (DCEM) of the Mindanao State University-Iligan Institute of Technology (MSUIIT), Iligan City, Philippines. (e-mail: cherrymae.galangque@g.msuiit.edu.ph).

Search and rescue operations can be risky for a variety of reasons. One reason is that SAR operations are often carried out in hazardous environments, such as in the aftermath of natural disasters, where there may be debris, collapsed structures, and other hazards present. SAR personnel may also be at risk of injury or death due to the nature of the rescue operation itself, such as when they are working at great heights or in unstable or flooded areas. Another is that SAR often involves working in time-sensitive situations where lives are at stake. This can create pressure to locate and assist people in distress quickly, increasing the risk of accidents or injuries.

UAVs, which were previously only used for military purposes, are now available for civilian use. Their application has expanded rapidly, and they are now being used in industry [1], logistics [2], crime scene surveillance [3], precision agriculture [4], photography and video filming as a hobby [5], and medicine [6]. Recently, UAVs have also been used for disaster relief and humanitarian efforts such as search and rescue operations, catastrophe prevention, and disaster management. With the advancement of technology over the past few years, the use of unmanned aerial vehicles (UAVs) has become evident in a wide variety of fields. Emergency services across the globe have already begun implementing UAVs for search and rescue operations in their attempt to reduce the time spent surveying a certain area during missions.

UAVs can be used to aid in SAR operations by providing a quick and cost-effective way to survey large areas and locate people in distress. Despite the UAV technological advancements, in terms of exploration and reliability, those systems are most often controlled by a trained operator to avoid obstructions thus there is a need to deploy UAVs equipped with cameras and other sensors to quickly scan an area and identify potential hazards, as well as locate people who may be trapped or in need of assistance. Existing works in autonomous UAV exploration missions focus on a target-oriented approach to recognize objects of interest in an unknown environment and reach them efficiently [7], [8].

Thus, the paper presents the development of a mission-planning and obstacle avoidance algorithm utilizing the Virtual Potential Field method and the power of LiDAR sensor.

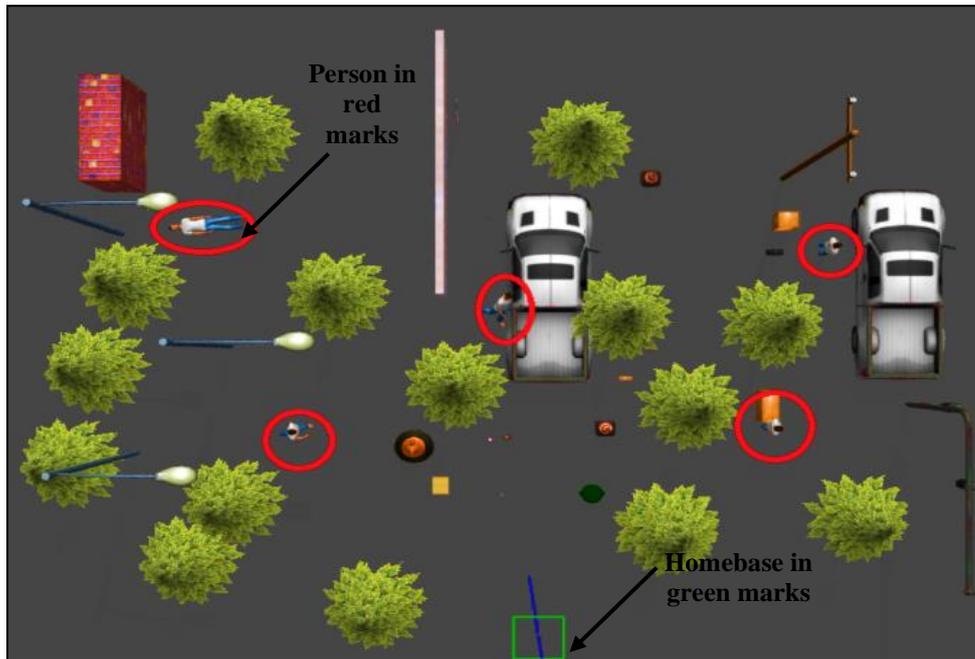


Fig. 1. Designed ROS-Gazebo World

Object detection system to recognize humans for rescue was integrated and was deployed in a simulated environment in ROS-Gazebo.

II. METHODOLOGY

In this study, a simulated environment was created using ROS-Gazebo to mimic a possible disaster scene. The designed Gazebo world for search and rescue UAV operations would simulate a realistic environment in which a UAV can navigate and perform search and rescue tasks. As shown in Fig. 1, the designed world has five (5) people in red markers (person to be search and rescue), one(1) home based in green mark and various obstacles, such as cars, trees, posts, tables, and other objects.

Fig. 2 shows the overall system architecture of the study. After loading the initial waypoint for the UAV to navigate, the flight controller automatically perform the necessary commands to control the drone's movement and orientation, those commands are passed to the mission planning and obstacle avoidance algorithms. These algorithms take into account the mission plan and any potential obstacles in the environment and adjust the commands accordingly. The output of the mission planning and obstacle avoidance algorithms is then sent to the GNC which is responsible for controlling the drone's movement and ensuring that it stays on the correct course. The code will then be passed to the simulated actuators, such as motors and servos, which move the drone in the virtual environment.

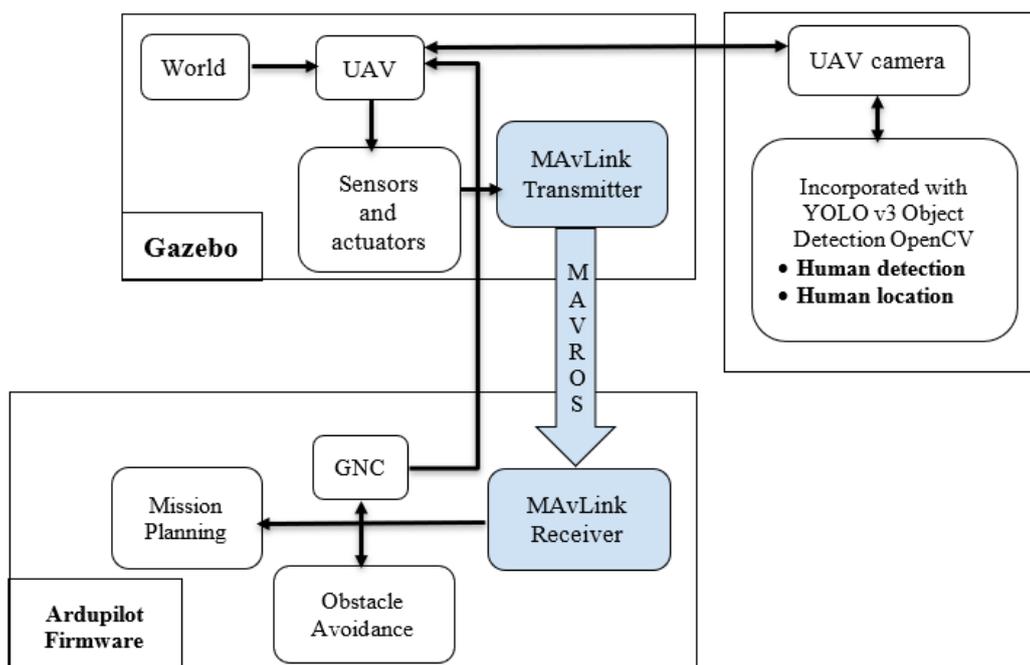


Fig. 2. Overall System Architecture



Fig. 3. PX4-SITL multirotor UAV

A camera is attached to the UAV and was integrated with YOLOv3 Object Detection OpenCV to detect humans in the virtual environment. The human's location is then determined based on the drone's location. The output of the simulation, including the sensor data, the commands sent to the actuators, and the drone's performance, can be logged and analyzed to further evaluate the performance of the algorithms and identify any issues or areas for improvement.

The PX4-SITL multirotor UAV "Iris" MODEL was used in simulation experiments as shown in Fig. 3. This multirotor UAV has a GPS to determine its position, one 2D LiDAR in obstacle avoidance, and a micro air vehicle link (MAVLink) to communicate with other entities such as other UAVs, base stations, and ground vehicles. The maximum flight time for Iris UAV is roughly 20 minutes. The PX4 autopilot and an inertial measurement unit (IMU) assist the UAV's flight controller in providing velocity and heading. Through the MAVLink, the onboard autopilot accepts yaw angle, velocity, and position waypoints. MAVROS, an application programming interface (API), runs on the companion computer and delivers the generated path planning waypoints to the UAV's autopilot at a frequency of 2 Hz. Table I shows the Iris drone model specifications.

Mission Planning Algorithm

The mission planning module was implemented using the GPS data coordinates to determine the location of the UAV and used virtual potential field methods to plan a path for the UAV to follow.

The virtual potential field is a mathematical model that simulates the forces acting on the UAV or vehicle, such as attractive forces toward a goal location and repulsive forces away from obstacles.

 TABLE I
IRIS DRONE MODEL SPECIFICATION

Dimensions	Motor to motor dimension: 550mm
Height	100mm
Weight	Weight (with battery): 1282 grams
Performance	Average flight time: 10-15 minutes
Payload capacity	400 g (.8 lb)
Battery	3-cell 11.1 V 3.5 Ah lithium polymer with XT-60 type connector
Weight	262 grams
Propellers	(2) 10 x 4.7 normal-rotation, (2) 10 x 4.7 reverse-rotation
Motors	AC 2830, 850 kV
Telemetry/Control	Radios available in 915mHz or 433mHz
Autopilot	Next generation 32-bit Pixhawk autopilot system with Cortex M4 processor
GPS	uBlox GPS with an integrated magnetometer
Dimensions	Motor to motor dimension: 550mm

By simulating these forces, the UAV or vehicle can be guided toward its goal while avoiding obstacles along the way. The distance d between the current position and the target destination is calculated using (1), wherein $X_{current}$ and $Y_{current}$ are X and Y coordinates of the UAV's current position and X_{target} and Y_{target} are X and Y coordinates of the UAV's next position.

$$d = \sqrt{(X_{current} - X_{target})^2 + (Y_{current} - Y_{target})^2} \quad (1)$$

Attraction Vector A_v was determined using (2). If the distance d between the UAV and the target is less than 0.65 with altitude greater than 3 then it will navigate towards next destination wherein tr is the target position, Cu is the current position of the UAV, Z_r is the zone of attraction and distance based on (1).

$$A_v = \begin{cases} \frac{tr-c}{d} & , d < Z_r \\ 0 & , d \geq Z_r \end{cases} \quad (2)$$

```

1: Counter ← 0
2: distance = sqrt(pow(current position x - target x[counter], 2) + pow(current position y -
3: target y[counter], 2))
4: if distance < 0.65 and altitude > 3 then
5: counter ← counter + 1
6: end if
7: if counter = 10 then
8: Land
9: end if
10: if distance < zone radius of attraction then
11: attraction vector x = (target x - current x)/distance
12: attraction vector y = (target y - current y)/distance
13: end if
14: total attraction x ← attraction vector x + current x
15: total attraction y ← attraction vector y + current y
16: Set Destination (total attraction x, total attraction y, altitude, headings)
    
```

Fig. 4. Mission Planning Pseudocode

```

1:  for j = 1 → size of current Lidar scan do
2:      d0 ← 0
3:      k ← 0.2
4:      if 0.35 < current Lidar scan > d0 then
5:          avoid = true
6:          x = cos(current_2D_scan.angle_increment*j)
7:          y = sin(current_2D_scan.angle_increment*j)
8:          U = -0.5 * k * pow((1/current Lidar scan) - 1/d0), 2)
9:          avoidance vector x += x * U
10:         avoidance vector y += y * U
11:     end if
12: end for
13:
14: if avoid is true then
15:     total vector ← sqrt(pow(avoidance vector x ,2) + pow(avoidance vector y,2))
16:     if total vector > 3 then
17:         avoidance vector x = 3 * (avoidance vector x / total vector)
18:         avoidance vector y = 3 * (avoidance vector y / total vector)
19:     end if
20: end if
21: f_total_x = attraction vector x + avoidance vector x + current position x
22: f_total_y = attraction vector y + avoidance vector y + current position y
23: Set destination ( f_total_x, f_total_y, altitude, headings)
    
```

Fig. 5. Obstacle Avoidance Pseudocode

Once the mission is completed, the UAV lands using the land() function. If the distance is less than a predefined zone radius, an attraction vector is computed as a unit vector pointing towards the target. The attraction vector is then transformed to the UAV's frame of reference using a rotation matrix. Fig. 4 shows the pseudocode of the mission planning function deployed in the UAV. The mission planning algorithm updates the UAV's destination based on the attraction vector. It also has a condition loop to correct the increment of values and not let the UAV stray off course.

Obstacle Avoidance Algorithm

Obstacle avoidance module used a LiDAR sensor to scan the environment and detect obstacles, and then using a repulsive potential field to plan a path for the UAV to follow to avoid the obstacles.

The module starts by iterating through each range measurement in the LiDAR sensor data and checks if the range is between a threshold distance d_t which is set to 1.5 and a minimum distance (0.35 m) from the UAV. If the range measurement is within this range, the algorithm considers it an obstacle and sets the flag "avoid" to 1. Then the algorithm calculates the magnitude of the repulsive force (U) using a positive scaling factor k which is set to 0.2, and the distance between the UAV and the obstacle. The magnitude of the repulsive force is calculated using (3), wherein U is the magnitude of the repulsive force, k is a positive scaling factor and d_{obs} is the distance between the UAV and obstacle and d_t is threshold distance.

$$U = -0.5k \left(\frac{1}{d_{obs}} - \frac{1}{d_t} \right)^2 \quad (3)$$

The potential field method combines both attractive and repulsive forces to guide a UAV toward its goal while avoiding obstacles.

The attraction force is modeled as a vector that points towards the goal, and its magnitude is proportional to the distance between the UAV and the goal.

The closer the UAV is to the goal, the stronger the attraction force will be. The overall movement of the UAV is determined by the sum of these two forces. The attraction force will pull the UAV towards the goal, while the repulsion force will push the UAV away from obstacles. By adjusting the relative strengths of these two forces, the UAV can be made to move toward the goal while avoiding obstacles. The overall potential field was shown in (4) where in U_q is the overall potential field, U_{att} is the force of attraction, and U_p is the force of repulsion.

$$U_q = U_{att} + U_p \quad (4)$$

In the potential field method, the waypoint is set as the goal or target location and is assigned a high attractive potential. The attraction force is then calculated based on the distance between the UAV and the waypoint, with the force increasing as the UAV gets closer to the waypoint and decreasing as it moves farther away. Fig. 5 shows the pseudocode of the Obstacle Avoidance function deployed in the UAV.

Virtual Force Method for Local Minimum

A UAV may get stuck in the local minimum, unable to reach the global minimum, which is the optimal path. This situation is handled by the free force method using (5) wherein θ_f free space orientation, force constant F_{cf} and free force F in (6).

$$F_f = F_{cf} \left[(\cos \theta_f) e_x + (\sin \theta_f) e_y \right] \quad (5)$$

$$F = F_{att} + F_{rep} + F_f \quad (6)$$

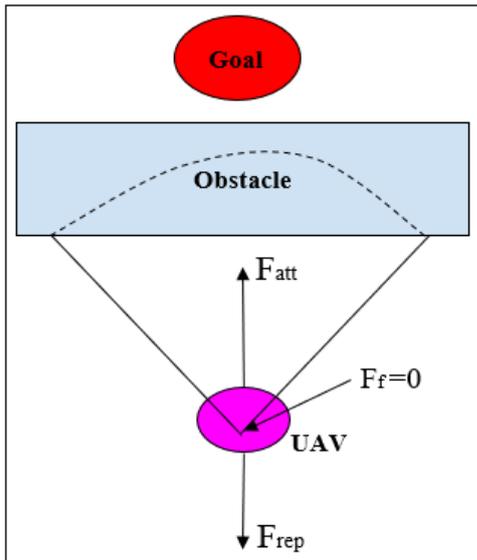


Fig. 6. Local Minimum Trap Null Resultant Force Case

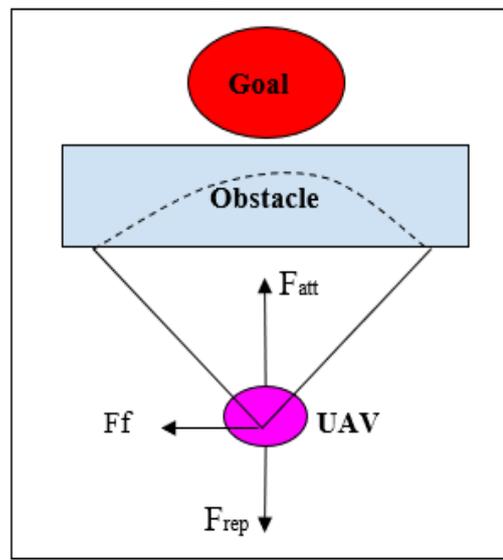


Fig. 7. Free Space Vector F_f added in the Obstacle Detection

Fig. 6 shows that in path planning, a local minimum is a point on a path where the UAV's movement is restricted, and it cannot move to a lower-cost path without first increasing the cost. This can happen when the UAV is in an area with a concave shape, where the cost function has a lower value than its surroundings. In the potential field method, the UAV is guided towards the goal or target location by the F_{att} attraction force generated by the waypoint, while avoiding obstacles by the F_{rep} repulsion force generated by the obstacles. However, this can lead to the UAV getting stuck in a local minimum, where it cannot move to a lower-cost path without first increasing the cost. To avoid this problem, a free space force is added as shown in Fig. 7 to the potential field method. The F_f free space force guides the UAV towards areas of the environment that are free of obstacles and have a lower potential, thus helping the UAV escape from local minima and reach the global minimum, which is the optimal path.

Human Detection using YOLOv3

YOLO (You Only Look Once) v3 designed by [8] was used for human detection. It is a real-time object detection algorithm that can be used to detect and classify objects, including people, in images and videos as it can detect and track multiple objects in the UAV's field of view.

It is a small model that can easily be deployed on the UAV's onboard computer. The YOLOv3 model was implemented as a convolutional neural network and was evaluated on a PASCAL VOC detection dataset [8]. The initial convolutional layers of the network extract feature from the image while the fully connected layers predict the output probabilities and coordinates. The network architecture was inspired by the GoogLeNet model for image classification [9]. The network has 24 convolutional layers followed by 2 fully connected layers. Instead of the inception modules used by GoogLeNet, it simply use 1×1 reduction layers followed by 3×3 convolutional layers, similar to Lin et al [10]. The full network is shown in Fig. 8.

A python script function was created that would determine the location of the human when detected based on the drone's current location. The darknet_ros in Github repository was used in the implementation. Darknet is an open-source neural network framework that runs on CPU and GPU. Currently, darknet_ros does not support higher YOLO versions other than YOLOv3. Thus, YOLOv3 model was considered for human detection module.

The YOLOv3 is robust to changes in perspective, which makes it well-suited to be used on UAVs that move around in 3D space and can change viewing angles quickly.

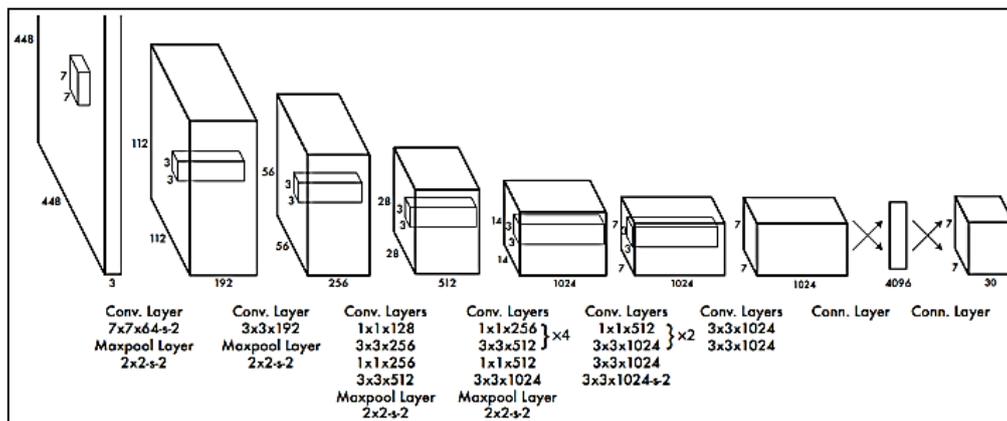


Fig. 8. The YOLOv3 Network Architecture

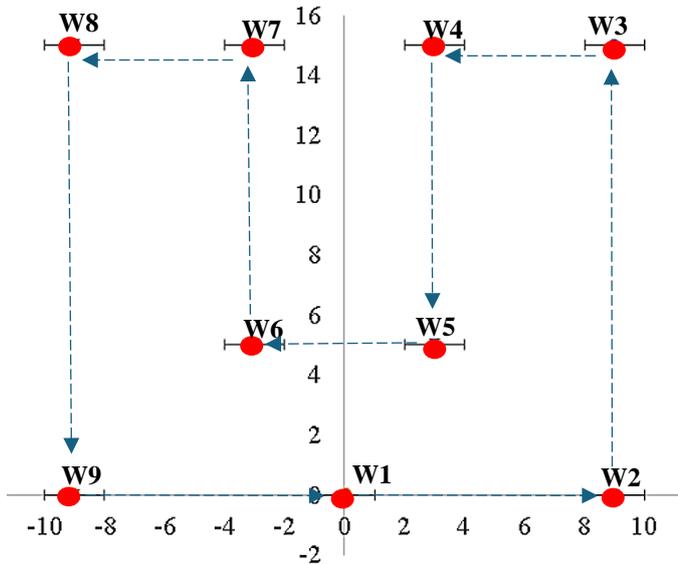


Fig. 9. The 9 waypoints for U-path

III. RESULTS AND DISCUSSIONS

The developed mission-planning and obstacle-avoidance algorithm based on the Virtual Potential Field method is analyzed based on three (3) different test runs. All these experiments are run at a maximum speed of 0.5m/s. The algorithms are tested individually before integrating it to the YOLOv3 human detection system and deployment to the UAV model in ROS-Gazebo.

Mission-planning Algorithm results

Fig. 9-11 shows the actual path navigated by the UAV. UAV was first placed at home based (0, 0) in x and y coordinates, respectively.

The mission-planning algorithm was tested on three mission paths with and without environmental obstacles, with each path tested five (5) times. The first path follows a U-shaped path to cover the square 15 x 18 meters area with 9 waypoints (see Fig. 9), star-shaped path with 5 waypoints (see Fig. 10) and a set of four-connected line paths with 5 waypoints (see Fig. 11) respectively.

There are 5 trials designed to test mission planning algorithms without obstacles.

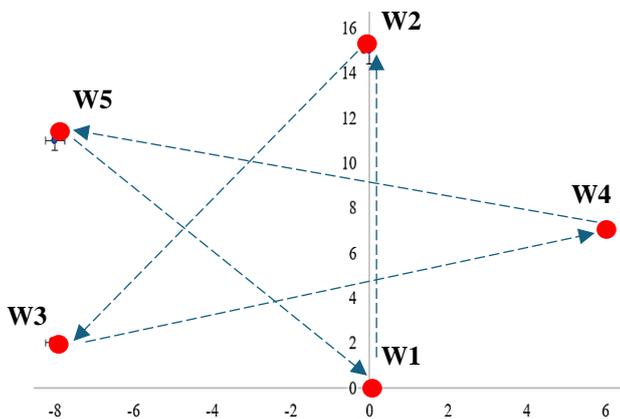


Fig. 10. The 5 waypoints for star-shaped path

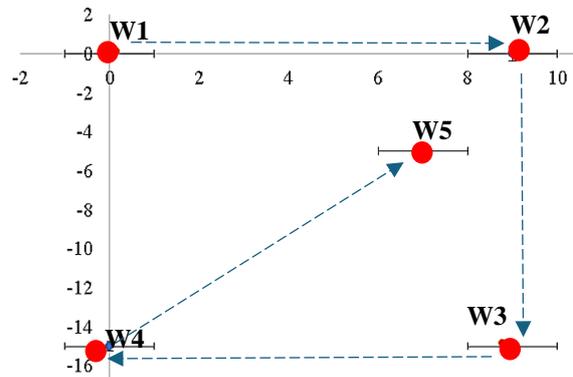


Fig. 11. The 5 waypoints for four-connected line path

The errors were calculated by calculating the difference between the initial waypoints and the actual waypoints traveled by the UAV. For testing the mission planner for u-path without obstacles, the maximum error (error difference in the initial waypoint and the actual waypoint readings traveled by the UAV) is 5.7% for x coordinates and maximum error (error difference in the initial waypoint and the actual waypoint readings traveled by the UAV) is 14% for y coordinates as shown in Fig. 12.a.

On the other hand, testing the mission planner for u-path with obstacles, the maximum error (error difference in the initial waypoint and the actual waypoint readings traveled by the UAV) is 9.1% for x coordinates and maximum error (error difference in the initial waypoint and the actual waypoint readings traveled by the UAV) is 12.7% for y coordinates as shown in Fig. 12.b.

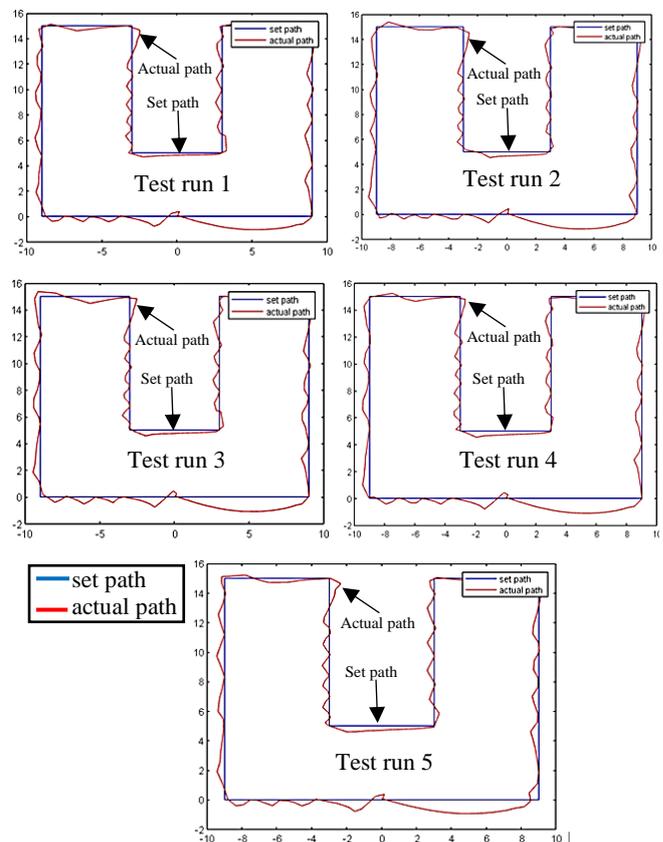


Fig. 12.a Five (5) testing for U-path without obstacles

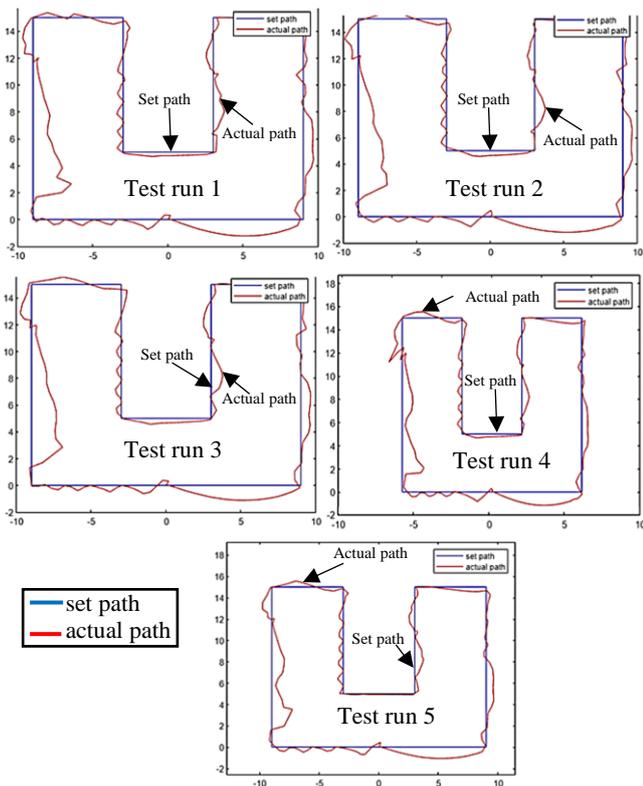


Fig. 12.b Five (5) testing for U-path with obstacles

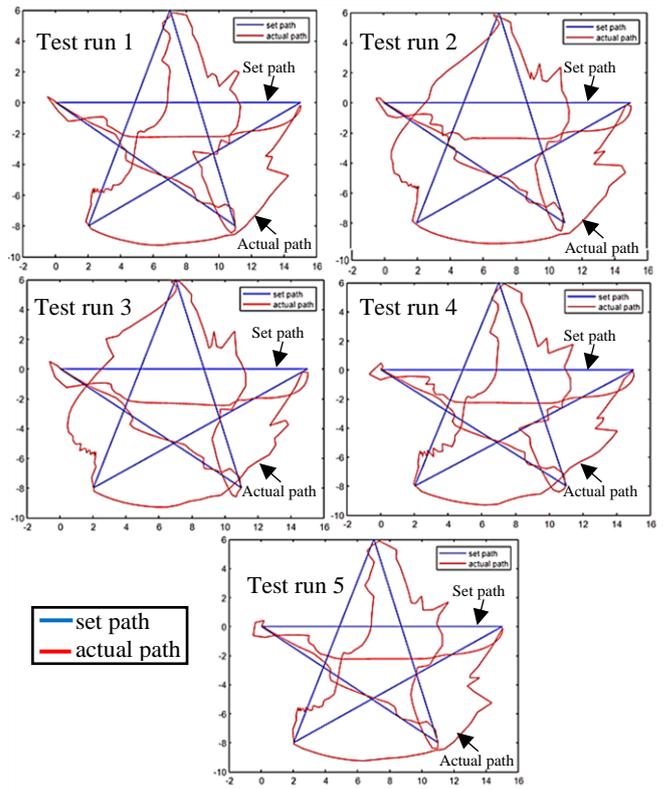


Fig. 13.b Five (5) testing for Star-path with obstacles

Fig. 13.a shows the testing result of the mission planner for star-path without obstacles. The maximum error is 5.4% for x coordinates and 4.9% for y coordinates. On the other hand, Fig. 13.b for star-path with obstacles, the maximum error is 10% for x coordinates and 10% for y coordinates.

Fig. 14.a shows the testing result of the mission planner for four line path without obstacles. The maximum error is 5.4% for x coordinates and 4.9% for y coordinates. On the other hand, Fig. 14.b for four line path with obstacles, the maximum error is 10% for x coordinates and 10% for y coordinates.

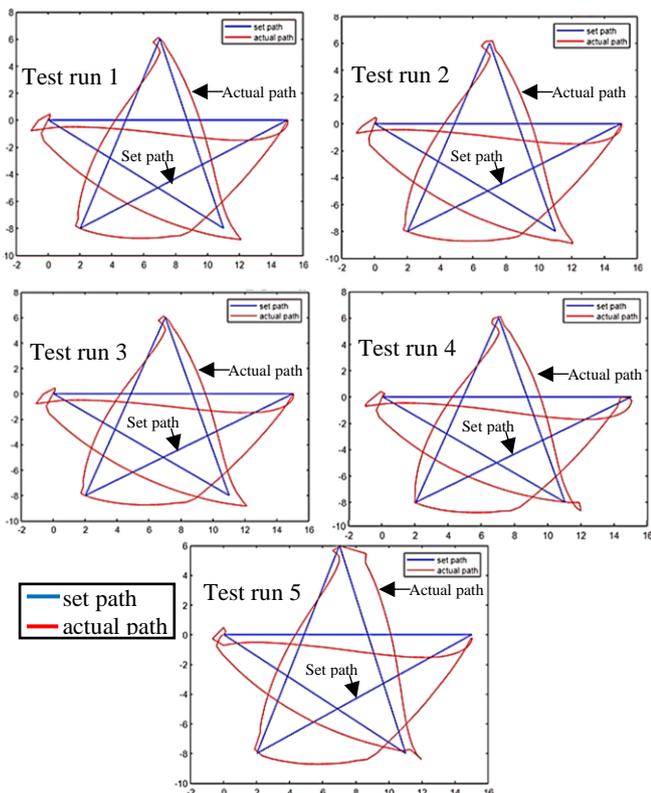


Fig. 13.a Five (5) testing for Star-path without obstacles

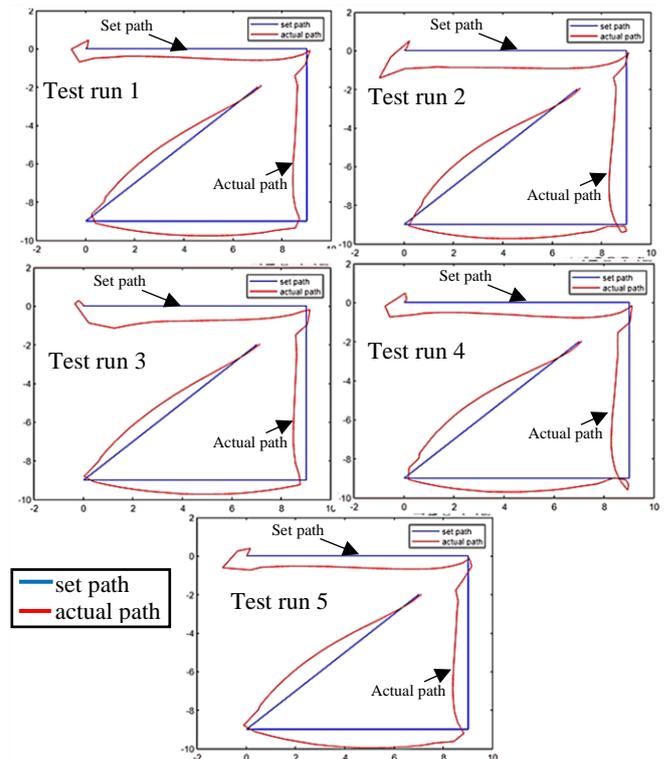


Fig. 14.a Five (5) testing for Four line path without obstacles

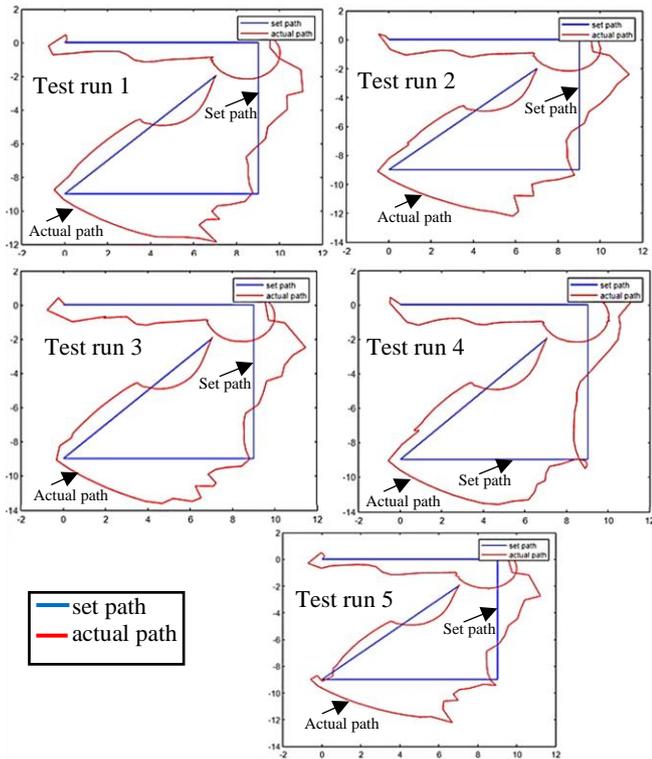


Fig. 14.b Five (5) testing for Four line path with obstacles

Obstacle Avoidance Algorithm results

There are 30 test runs to test the obstacle avoidance module of the study. All of these test runs are tested with a maximum velocity of 0.5 m/s. Fig. 10 shows some trials designed to test obstacle avoidance algorithms. It is worth noting that the paths are all smooth and follow reasonably efficient paths through the obstacle field. The x-axis shows the waypoints taken by the drone before reaching the goal. The y-axis denotes the distance of the drone to the goal.

Obstacle Avoidance Algorithm Exception

There are several scene exceptions considered in the test conducted for obstacle avoidance. First is using concave obstacles or U-trap as shown in Fig. 16. The UAV did not detect the concave trap, leading to continuous moving forward without realizing it was trapped. Concave obstacles usually result in a local minimum. When the UAV is trapped by a concave-shaped obstacle, it will get confused, resulting in oscillation or running in a closed loop.

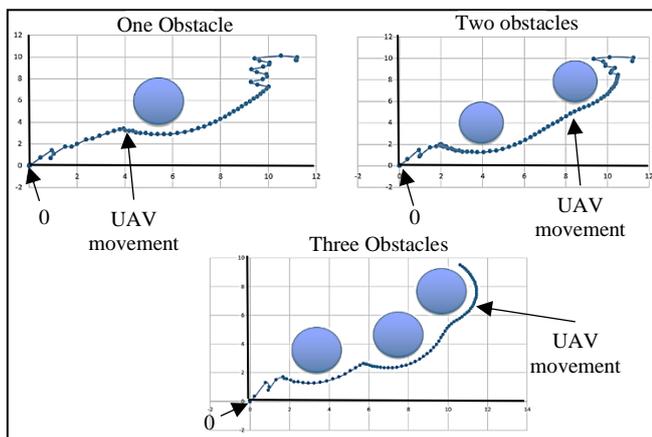


Fig. 15. Test runs for obstacle avoidance.

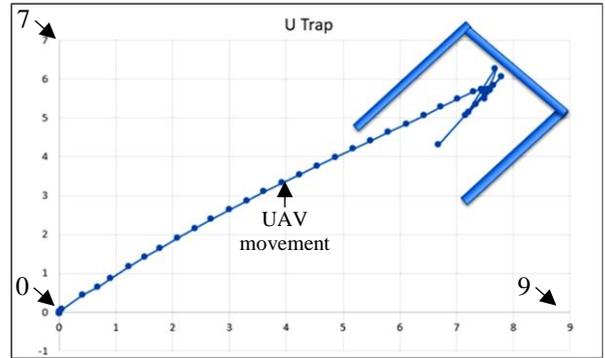


Fig. 16. UAV movements in Concave / U-trap

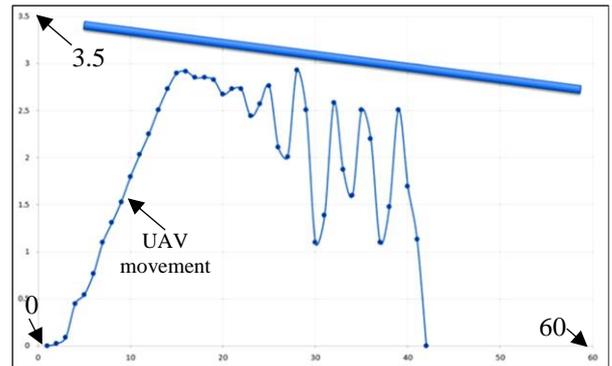


Fig. 17. UAV movements in long wall trap

Another scene exception is that the potential field method can be vulnerable to "long-wall traps," where the UAV encounters a long, straight obstacle (e.g. wall) that it cannot pass. This is because the potential field method can create a force that is too strong for the UAV to overcome, trapping it in the same place.

TABLE II
YOLOV3 HUMAN DETECTION PERFORMANCE

No. of Test	Flight Time	Expected Humans Detected	Actual Humans Detected	Mission Success Status
U-path				
1	3:17	5	5	Completed
2	2:57	5	5	Completed
3	3:14	5	5	Completed
4	2:55	5	5	Completed
5	3:11	5	5	Completed
Star path				
1	4:31	5	6	Completed
2	4:54	5	6	Completed
3	4:44	5	7	Completed
4	4:26	5	6	Completed
5	4:20	5	6	Completed
Four -connected line path				
1	2:38	5	4	Completed
2	2:47	5	4	Completed
3	2:45	5	4	Completed
4	3:16	5	4	Completed
5	3:06	5	4	Completed

The results of the search and drone system are shown in Table II. The authors counted the humans detected using the YOLO v3 camera. Flight time is the drone's flight time from take-off to landing and is shown in the SITL console. Humans detected by the system is the number of humans detected in the path given. Success Rate is used to identify if the drone has finished its flight from takeoff to landing. The percentage

of human detection varies on the path followed by the UAV. The U-path has 5 out of 5 or 100% result in human detection as the U plan path covers all areas.

A total of 6 out of 5 or 80% of people were picked up by the star path. Human detection redundancy occurs as the UAV passed again to the same spot. The UAV can only detect four humans in the environment along the star route because the last human was not covered by the path it had taken.

The four-connected line path detected 4 out of 5 or 80% of the humans. This occurred because the undetected human in the tests was not covered by the path taken by the drone.

The UAV following the U path mission, took 3 minutes and 6 seconds to complete its mission, for the star route the UAV took 4 minutes and 35 seconds and the four-connected line path took only 2 minutes and 54 seconds. Considering the segments and obstacles, the star route has the longest completion time because the UAV has to cover all the 5 segments with intersections and the obstacles in the area whereas, the four-connected line path has the shortest completion time because it will only cover 4 segments and the same obstacles.

IV. CONCLUSION

A well-planned and optimized mission path can reduce the overall time required to complete a task, while a poorly planned path can result in longer completion times due to increased travel time and inefficiencies. A well-designed obstacle avoidance system can ensure that the drone successfully navigates through the mission path without crashing into any obstacles. This, in turn, can improve the success rate of completing the mission and increase the efficiency of the UAV in SAR operations. The percentage error in mission planning with and without obstacles is less than 5% in all the test runs conducted. The YOLO v3 that was incorporated in the UAV's camera has detected humans depending on its path taken. The U-path has 100% results of human detection as it covers all the points of interest without redundancy. A total of 80% of detected humans were picked up by the star path. Inconsistencies occur because of the repeating detection that happened as it took the star path. The four-connected line path detected 80% of the humans. This happened because the undetected human in the four-connected lines is out of the UAV's path. Lastly, regardless of the obstacles present in the virtual environment, the implemented mission planning assures that the UAV could complete its mission.

REFERENCES

- [1] Shahmoradi, J., Talebi, E., Roghanchi, P., & Hassanalani, M. (2020). A comprehensive review of applications of drone technology in the mining industry. *Drones*, 4(3), 34. DOI : <https://doi.org/10.3390/drones4030034>
- [2] Moshref-Javadi, M., & Winkenbach, M. (2021). Applications and Research avenues for drone-based models in logistics: A classification and review. *Expert Systems with Applications*, 177, 114854. DOI : <https://doi.org/10.1016/j.eswa.2021.114854>
- [3] Yaacoub, J. P., Noura, H., Salman, O., & Chehab, A. (2020). Security analysis of drones systems: Attacks, limitations, and recommendations. *Internet of Things*, 11, 100218. DOI : <https://doi.org/10.1016/j.iot.2020.100218>
- [4] Daponte, P., De Vito, L., Glielmo, L., Iannelli, L., Liuzza, D., Picariello, F., & Silano, G. (2019, May). A review on the use of drones for precision agriculture. In *IOP conference series: earth and environmental science* (Vol. 275, No. 1, p. 012022). IOP Publishing. DOI : <https://doi.org/10.1088/1755-1315/275/1/012022>
- [5] Hildebrand, J. M., & Hildebrand, J. M. (2021). Situating hobby drone practices. *Aerial Play: Drone Medium, Mobility, Communication, and Culture*, 45-71. DOI : https://doi.org/10.1007/978-981-16-2195-6_3
- [6] Poljak, M., & Šterbenc, A. J. C. M. (2020). Use of drones in clinical microbiology and infectious diseases: current status, challenges and barriers. *Clinical Microbiology and Infection*, 26(4), 425-430. DOI : <https://doi.org/10.1016/j.cmi.2019.09.014>
- [7] Bjelonic, M. (2018). YOLO ROS: Real-time object detection for ROS. URL : https://github.com/leggedrobotics/darknet_ros. DOI : <https://doi.org/10.22214/ijraset.2021.39044>
- [8] Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111, 98-136. DOI : <https://doi.org/10.1007/s11263-014-0733-5>
- [9] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9). DOI : <https://doi.org/10.1109/cvpr.2015.7298594>
- [10] Lin, M., Chen, Q., & Yan, S. (2013). Network in network. *CoRR* abs/1312.4400 (2013). arXiv preprint arXiv:1312.4400. DOI : <https://doi.org/10.1063/pt.5.028530>
- [11] Guo, J., Liang, C., Wang, K., Sang, B., & Wu, Y. (2021). Three-dimensional autonomous obstacle avoidance algorithm for UAV based on circular arc trajectory. *International Journal of Aerospace Engineering*, 2021, 1-13. DOI : <https://doi.org/10.1155/2021/8819618>
- [12] Hayat, S., Yanmaz, E., Bettstetter, C. et al. Multi-objective drone path planning for search and rescue with quality-of-service requirements. *Auton Robot* 44, 1183-1198 (2020). DOI : <https://doi.org/10.1007/s10514-020-09926-9>
- [13] Hayat, S., Yanmaz, E., Brown, T. X., & Bettstetter, C. (2017). Multi-objective UAV path planning for search and rescue. 2017 IEEE International Conference on Robotics and Automation (ICRA). DOI : <https://doi.org/10.1109/icra.2017.7989656>
- [14] Kumar, G., Anwar, A., Dikshit, A., Poddar, A., Soni, U., & Song, W. K. (2022). Obstacle avoidance for a swarm of unmanned aerial vehicles operating on particle swarm optimization: A swarm intelligence approach for search and rescue missions. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 44(2). DOI : <https://doi.org/10.1007/s40430-022-03362-9>
- [15] Li, T., & Hu, H. (2021). Development of the use of Unmanned Aerial Vehicles (uavs) in emergency rescue in China. *Risk Management and Healthcare Policy*, Volume 14, 4293-4299. DOI : <https://doi.org/10.2147/rmh.p.s323727>
- [16] Lygouras, E., Santavas, N., Taitzoglou, A., Tarchanidis, K., Mitropoulos, A., & Gasteratos, A. (2019). Unsupervised human detection with an embedded vision system on a fully autonomous UAV for search and rescue operations. *Sensors*, 19(16), 3542. DOI : <https://doi.org/10.3390/s19163542>
- [17] Megalingam, R. K., Prithvi, D. V., Kumar, N. C. S., & Egumadiri, V. (2021). Drone Stability Simulation Using ROS and Gazebo. *Advanced Computing and Intelligent Technologies*, 131-143. DOI : https://doi.org/10.1007/978-981-16-2164-2_11
- [18] Mishra, B., Garg, D., Narang, P., & Mishra, V. (2020). Drone-surveillance for search and rescue in natural disaster. *Computer Communications*, 156, 1-10. DOI : <https://doi.org/10.1016/j.comcom.2020.03.012>
- [19] Moshref-Javadi, M., & Winkenbach, M. (2021). Applications and research avenues for drone-based models in Logistics: A classification and Review. *Expert Systems with Applications*, 177, 114854. DOI : <https://doi.org/10.1016/j.eswa.2021.114854>
- [20] Poljak, M., & Šterbenc, A. (2020). Use of drones in Clinical Microbiology and Infectious Diseases: Current status, challenges and barriers. *Clinical Microbiology and Infection*, 26(4), 425-430. DOI : <https://doi.org/10.1016/j.cmi.2019.09.014>
- [21] Sakhthivel, P., & Anbarasu, B. (2020). Integration of Vision and LIDAR for Navigation of Micro Aerial Vehicle. 2020 Third International Conference on Multimedia Processing, Communication & Information Technology (MPCIT). DOI : [10.1109/mpcit51588.2020.93504](https://doi.org/10.1109/mpcit51588.2020.93504)
- [22] Selvam, P. K., Raja, G., Rajagopal, V., Dev, K., & Knorr, S. (2021). Collision-free path planning for uavs using efficient artificial potential field algorithm. 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring). DOI : <https://doi.org/10.1109/vtc2021-spring51267.2021.9448937>
- [23] Shahmoradi, J., Talebi, E., Roghanchi, P., & Hassanalani, M. (2020). A comprehensive review of applications of drone technology in the mining industry. *Drones*, 4(3), 34.

- DOI: <https://doi.org/10.3390/drones4030034>
- [24] Stecz, W., & Gromada, K. (2020). UAV mission planning with SAR Application. *Sensors*, 20(4), 1080.
DOI: <https://doi.org/10.3390/s20041080>
- [25] Stecz, W., & Gromada, K. (2020). UAV mission planning with SAR Application. *Sensors*, 20(4), 1080.
DOI: <https://doi.org/10.3390/s20041080>
- [26] United Nations Office for Disaster Risk Reduction. (2019). *Disaster Risk Reduction in the Philippines*.
- [27] Victoria, L. P. (n.d.). *Community Based Disaster Management in the Philippines: Making a Difference in People's Lives*.
- [28] Woods, A. C., Lay, H. M., & Ha, Q. P. (2016). A novel extended potential field controller for use on Aerial Robots. 2016 IEEE International Conference on Automation Science and Engineering (CASE).
DOI: <https://doi.org/10.1109/coase.2016.7743420>
- [29] Woods, A. C., Lay, H. M., & Ha, Q. P. (2016). A novel extended potential field controller for use on Aerial Robots. 2016 IEEE International Conference on Automation Science and Engineering (CASE).
DOI: <https://doi.org/10.1109/coase.2016.7743420>
- [30] Y. Alborzi, B. S. Jalal and E. Najafi, "ROS-based SLAM and Navigation for a Gazebo-Simulated Autonomous Quadrotor," 2020 21st International Conference on Research and Education in Mechatronics (REM), Cracow, Poland, 2020, pp. 1-5,
DOI: [10.1109/REM49740.2020.9313875](https://doi.org/10.1109/REM49740.2020.9313875).