Multi-Robot Search and Rescue

Anaam Mostafiz, Harshil Shah, Siddharth Jain and Tabsheer Askari

Abstract— **This project deals with the use of a Potential-Field Based Control algorithm in conjunction with a Frontier Exploration Algorithm for decentralized mapping of unknown areas using quadcopter swarms. By mapping with a front-facing RGB depth camera, the drones can be used for search and rescue applications where identifying survivors and types of obstacles while exploring is critical.**

I. INTRODUCTION

A. Motivation

In recent years, advances in robot architecture, onboard sensing capabilities, and processing power have allowed the deployment of unmanned vehicles to various environments that are inhospitable for humans. This inhospitability may be due to the danger these environments pose to humans, or their remote nature. Post-earthquake urban environments pose both of these dangers. Therefore, robotic swarms are ideal for search and rescue operations in such environments. The robot swarms used in our project consist of quad-copters since they are readily available and have high maneuverability in unstructured urban environments.

B. Related Works

The Potential-Field based control strategy was first introduced by Khatib [1] and has since been used to provide strategies for an array of multi-agent deployment challenges, such as obstacle avoidance, goal/frontier seeking, robot-robot collision avoidance, and diffusive behavior of the collective (swarm). These strategies, when combined, allow for the fulfillment of more complex collective goals, such as mobile sensor deployment [2], multi-robot exploration & mapping, and swarm pattern generation [3].

C. Our Proposed Solution

Perhaps one of the biggest advantages of the Potential-Field based control strategy is its decentralized nature, which allows for robustness to individual agent failure (which is to be accounted for in unsafe workspaces), and scalability of the deployment. However, without careful design, Potential-Field algorithms have a tendency to trap agents in local minima in non-convex environments, which means complete mapping is not guaranteed.

Renzaglia et al. [4] proposed a leader-trooper model that guarantees convergence using frontier exploration, where the leader robot runs a classic path-planning algorithm instead of potential-field so that it can repel the trooper robots out of local minima. Taking inspiration from their approach, in this project we implement the Potential-Field control architecture for efficient and safe multi-agent motion planning and mapping in an unknown environment. We will use the Frontier Algorithm to intermittently assign our agents new (unexplored) goals to be attracted to.

The key feature of our approach is that each drone's 2D laser will be filtered to the same front field-of-view as an RGB depth camera, so that the drone only maps what the camera sees. Meanwhile, the unfiltered 2D laser range will be used for obstacle detection and avoidance. Together, these strategies will encourage the swarm to capture 3D color information of the environment from a variety of angles. As a result, rescuers will have a 2D map, a 3D map, and camera images to assist their efforts.

II. MATHEMATICAL MODEL

Our solution uses a swarm of homogeneous quadcopters. Quadcopters were chosen for their holonomic drive mechanism (as opposed to differential drive robots that cannot move omnidirectionally), as well as their ability to avoid ground obstacles altogether by flying.

A. Problem Formulation

We consider an unknown 3-dimensional environment. We suppose each of our robots has a limited radius, omnidirectional laser sensor for mapping & obstacle avoidance, and a front-facing RGB depth camera to capture the footage of the mapped area. The latter will make it easier to detect/identify victims after mapping is complete.

Like Renzaglia et al. [4], our goal is to map the whole environment. A point in the workspace is considered explored if a robot comes closer to the point than the robot's sensing range.

B. Leader Robot

One of the robots in our swarm will be a leader. This robot will be unaffected by the potential field the other robots will experience. The leader will use a path-planning algorithm such as *Dijkstra's* or *A-star* to go the the nearest frontier. This will bypass the problem of robots getting stuck in local minima caused by non-convex obstacles (See Section III-B for further details).

C. Trooper Robots Model

The agents (other than the leader) in our system are characterized by a series of attractive and repulsive forces on each of the agents, which dictate their motion. These forces arise from an artificial potential field that consists of:

- Repulsion from the other agents;
- Repulsion from closest obstacle;
- Attraction to goals (Frontiers of Exploration).

All potential-field equations and system model equations have been adopted from Renzaglia et al. [4].

1) Repulsive Potential: The repulsive potential is defined as:

$$U_{rep}(\mathbf{q},\mathbf{q}_i) = \begin{cases} \frac{1}{2}k_{rep}(\frac{1}{\rho(\mathbf{q})} - \frac{1}{\rho_0})^2 & ,\rho(\mathbf{q}) \leq \rho_0 \\ 0 & ,\rho(\mathbf{q}) > \rho_0 \end{cases} \tag{1}$$

where $\mathbf{q}_i$ is the position of the robot/obstacle, $\rho(\mathbf{q}) = \|\mathbf{q} - \mathbf{q}_i\|$, and ρ_0 is the range in which the potential field is defined to be non-zero. The force generated by this potential field is $\mathbf{F}_{rep}(\mathbf{q}) = -\nabla U(\mathbf{q})$:

$$\mathbf{F}_{rep}(\mathbf{q},\mathbf{q}_i) = \begin{cases} k_{rep}(\frac{1}{\rho(\mathbf{q})} - \frac{1}{\rho_0})\frac{\mathbf{q}-\mathbf{q}_i}{\rho^3(\mathbf{q})} & ,\rho(\mathbf{q}) \leq \rho_0 \\ 0 & ,\rho(\mathbf{q}) > \rho_0 \end{cases} \tag{2}$$

Therefore, each robot of the N total robots in the swarm experiences a total repulsive force of:

$$\mathbf{F}_{rep}(\mathbf{q}) = \sum_{i=1}^{N} \mathbf{F}_{rep}(\mathbf{q},\mathbf{q}_i) \tag{3}$$

where we sum over the other $N-1$ agents and the nearest obstacle. We consider $\rho_0^{obstacle} \ll \rho_0^{robot}$ since the robot-obstacle threshold only needs to be large enough to avoid collision, but the robot-robot distance needs to be large enough to make the agents disperse away from one another for better coverage.

2) Attractive Potential: The attractive potential we use is defined as:

$$U_{att}(\mathbf{q}) = \frac{1}{4}k_{att}\ \rho_{goal}^4 \tag{4}$$

and the resultant attractive force as:

$$\mathbf{F}_{att}(\mathbf{q}) = k_{att}(\mathbf{q}_{goal} - \mathbf{q})\rho_{goal}^2 \tag{5}$$

where $\rho_{goal} = \|\mathbf{q} - \mathbf{q}_{goal}\|$ and $\mathbf{q}_{goal}$ is the midpoint of the frontier. Furthermore, we add a viscous term to the system, $v\,\dot{\mathbf{q}}$, that causes energy dissipation and allows the system to reach equilibrium (see Section III-A for proof). Therefore, the equation of motion for each agent is:

$$\mathbf{F}_{tot} = \mathbf{F}_{rep} + \mathbf{F}_{att} = m\ddot{\mathbf{q}} - v\dot{\mathbf{q}} \tag{6}$$

where m is the virtual mass of the robot and is taken to be unitary.

D. Assumptions and Constraints

The assumptions and constraints of our implementation are as follows:

- Sensing - 2D laser with 360 degree range and 9 meter distance, and an RGB depth camera with a 60 degree field of view and 9 meter distance.
- Global Communication: Communication was considered global, for the sake of simplicity. In future work, limited communication range and its effect on the mapping speed can be explored.
- Decentralized Control: to make the system scalable and robust to failures.
- Bounded environment: in a real-world disaster mapping scenario, you need to define search limits. We are physically putting up boundary walls for our domain to mimic this.
- only working at a particular height (2D) due to limited onboard computational capabilities. 3D navigation was out of the scope of this project - may be pursued in future work.
- Our Environment contains obstacles, as any post-disaster environment would.
- Mathematical model of collective behavior: potential fields [4][5], frontier algorithm which sets frontiers (boundaries between explored and unexplored regions of the domain), that the robots *attract* to.

III. THEORETICAL ANALYSIS

A. System Convergence

Given a system of N robots, the total Potential Energy of the system, U, is given by:

$$U = \sum_{i=1}^{N}[U_{att,i} + U_{rep,i}] \tag{7}$$

The total Kinetic Energy of the system is given by:

$$T = \sum_{i=1}^{N}\frac{1}{2}m_i\dot{\mathbf{q}}_i{}^2 \tag{8}$$

The sum of these gives the total system energy:

$$E = U + T \tag{9}$$

In order to find the time evolution of the system, we take the time derivative of the total energy $\frac{dE}{dt}$. This is naturally the sum of the rates of change of the kinetic and potential energies:

$$\frac{dE}{dt} = \frac{dU}{dt} + \frac{dT}{dt} \tag{10}$$

Since the Equation of Motion for the system contains a viscous force term, which acts to remove energy from the system, we consider the system to be *dissipative* in nature [5].

Considering a state of the system in a finite domain with a fixed set of frontiers (attractors), the system has a finite amount of potential energy. The change in potential energy for this setup is given by [5]:

$$\frac{dU}{dt} = -\sum_{i=1}^{N}\mathbf{F}_{tot,i}\,\dot{\mathbf{q}}_i \tag{11}$$

The change in kinetic energy for the system is equal and opposite to the change in potential energy minus the viscous term [5]:

$$\frac{dT}{dt} = \sum_{i=1}^{N}(\mathbf{F}_{tot,i}\,\dot{\mathbf{q}}_i - v\dot{\mathbf{q}}_i{}^2) \tag{12}$$

Computing the resultant $\frac{dE}{dt}$, we get:

$$\frac{dE}{dt} = \frac{dU}{dt} + \frac{dT}{dt}$$
$$\Rightarrow \frac{dE}{dt} = -\sum_{i=1}^{N} \mathbf{F}_{tot,i}\,\dot{\mathbf{q}}_i + \sum_{i=1}^{N} (\mathbf{F}_{tot,i}\,\dot{\mathbf{q}}_i - v\dot{\mathbf{q}}_i{}^2)$$
$$\Rightarrow \frac{dE}{dt} = -v\sum_{i=1}^{N} \dot{\mathbf{q}}_i{}^2$$

Therefore, we can see that the energy of the system decreases over time.

It is important to note that once our agents explore the current set of frontiers, this will give rise to a new set of frontiers, adding fresh potential energy to the system. The system will have a new equilibrium at this point and will try to attain this new equilibrium. This will keep repeating, however, since our domain is bounded, there is only a finite number of new frontiers that can be given rise to before all the points in the domain have been explored. In other words, there is only a *bounded*, *finite* amount of potential energy that can be added to the system via exploration. Once no more frontiers remain, the system Potential Field will not change further, and according to the results given above, the agents will converge to rest.

B. Robots Trapped in Local Minima

However, without a leader drone (that would be unaffected by local minima), there exists a possibility that one of our agents gets stuck in a local minima. This could happen for various reasons, such as a case where the attractive force from the frontier is equal and opposite to the repulsion from the nearest obstacle and other drones. Figure 1 illustrates such a case. In this situation, the forces on the drone are such that:

$$\mathbf{F}_{tot} = \mathbf{F}_{rep} + \mathbf{F}_{att} = \begin{bmatrix} F_x \\ 0 \end{bmatrix}$$

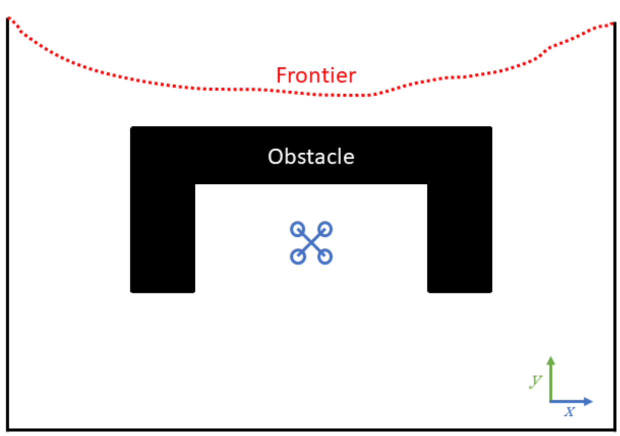


Fig. 1: Drone stuck at minima.

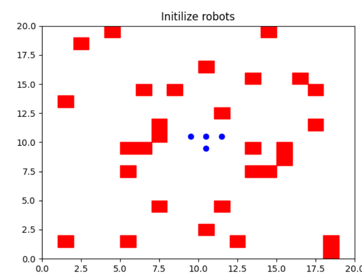
Any movement in the positive y direction will cause the net force on the drone to act in the negative y direction, pushing it down. Any movement in the negative y direction will cause the net force on the drone to act in the positive y direction, pushing it up. The side walls of the obstacle prevent any escape through motion in the x direction. In such a scenario, the robot is completely dependent on other robots in the swarm to explore the frontier, updating the Potential-Field and even then, there is no guarantee that the new potential field would not also have this point as a local minimum.

In order to prevent such situations in non-convex environments, it is necessary to have a leader drone that is unaffected by such local minima.

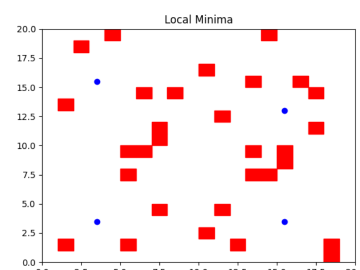
IV. VALIDATION IN SIMULATIONS

A. Python

First, we wrote Python graphical simulations to test the potential field algorithms. Robots are represented by colored dots, and the map is a cellular grid with randomized obstacles. After the robots spread out to local minima with repulsion only, they attract to the frontiers of the grid by moving one cell at a time while avoiding the coordinates of other robots.



(a)



(b)

Fig. 2: (a) Initial positions of robots in Python, (b) Robots spread out to local minima

B. Gazebo

Then, we developed a more realistic simulation in Gazebo using ROS. The key package being used is Real-Time Appearance-Based Mapping (RTAB-Map), which simultaneously collects 2D maps, 3D maps, and images [6]. These maps were not merged in real-time by the swarm in order to encourage detailed and exhaustive mapping of the 3D space. Furthermore, RTAB-Map can merge stored maps from

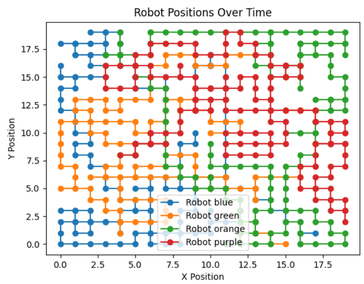


Fig. 3: Robot exploration trajectories in a random map in Python

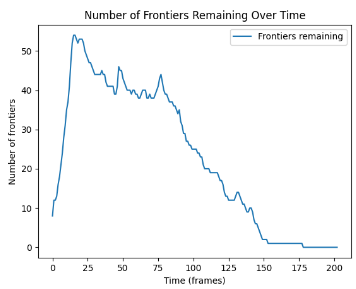


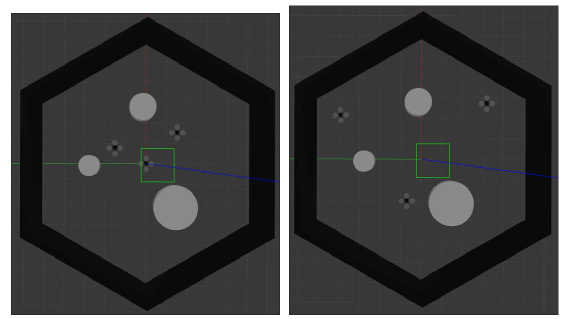
Fig. 4: Frontiers remaining over time in Python

different sessions using loop closures, or multiple image captures with a high-number of repeated features, so the mapping redundancy could be useful for post-exploration merging in the real world. However, Gazebo environments do not have anywhere as much detail as the real world, so it was difficult to test the increase in loop closures.

For algorithm testing, a small, simple "sphere" world was created and a large, detailed maze world was borrowed [7]. Additionally, the drone models were borrowed [8] and the navigation stack package [9] was used to implement path-planning and navigation with the leader drone. Due to computational limitations, only one leader drone and two trooper drones were used, although our code could be expanded and possibly more effective with a larger number of drones.

We wrote ROS Python code to implement the potential fields and frontier selection, which uses frontier detection code inspired by the rrt-exploration package [10]. Since the drones have odometry, their global positions were known and broadcast to each other as a way to differentiate between obstacles and drones in range. However, the drones do not know what frontiers the others are exploring, so to encourage separation the drones can be initially spread out using the pure repulsive potential field. For exploration, the trooper drones move omnidirectionally, only rotating to a random angle while selecting the next frontier to mix up the camera angles. A rotation matrix had to be used on the calculated forces since potential field algorithms normally do not consider the orientation of the robot. Meanwhile, the leader drone was allowed to move like a differential-drive robot so that it naturally rotates as it executes planned paths. Once any drone comes within close proximity of its chosen frontier, it will blacklist that frontier in its memory and choose the next closest one. Only the leader drone has the benefit of blacklisting frontiers that it cannot reach, as its path-planning node will give up or flag that it is stuck.

For the following simulation results (which continue onto next page), note that the simulations were not run to 100% completion due to computational limits.



(a) (b)

Fig. 5: (a) Initial position of drones in simple Gazebo world, (b) Drones spread out to local minima

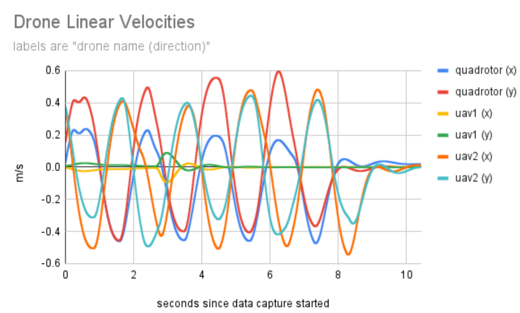
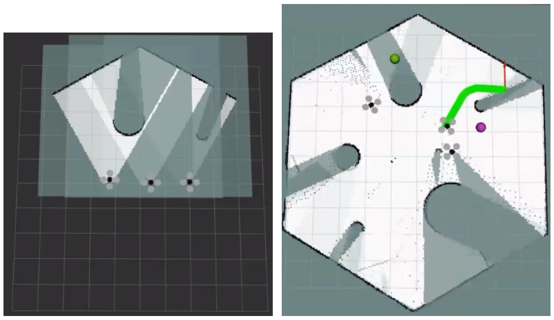
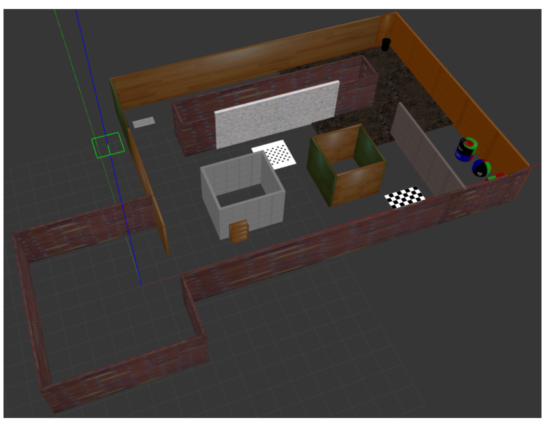


Fig. 6: The drone velocities near equilibrium. In the previous images, "quadrotor" is the middle drone, and "uav2" is the left drone. The oscillation is due to the fact that the robots are being repelled back and forth between obstacles and each other.

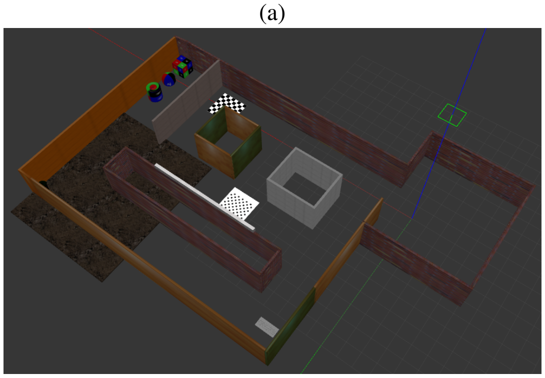


(a) (b)

Fig. 7: (a) Initial drone maps overlain. The drones are in the simple Gazebo world with two additional obstacles. (b) Intermediate maps overlain. The colored dots represent the trooper frontiers, the arrow represents the target pose of the leader's frontier, and the thick green line represents the leader's planned path. At this instance, the dots are closest to their respective trooper, and one trooper is actively avoiding the leader while the other is avoiding an obstacle.

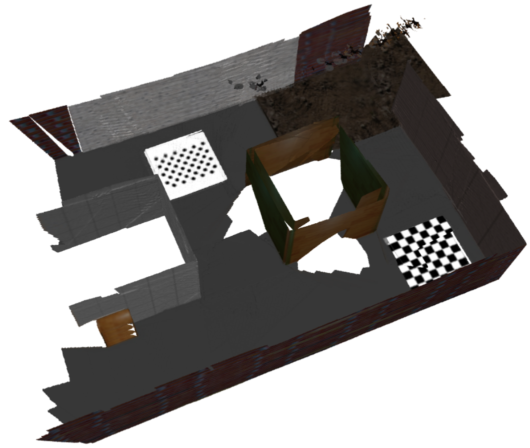


(a)

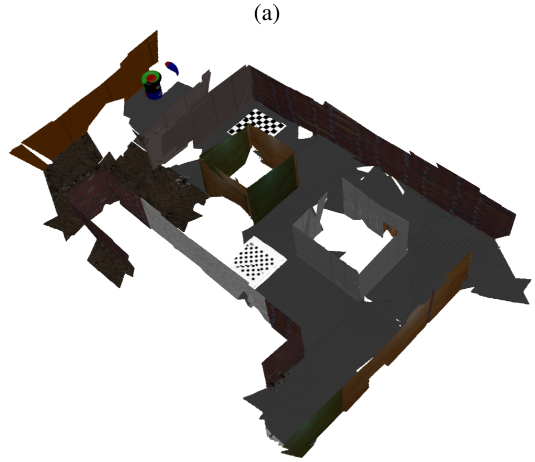


(b)

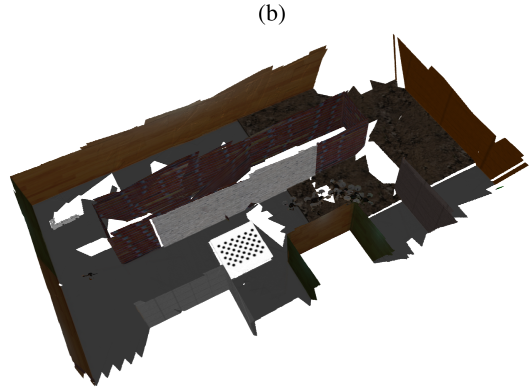
Fig. 8: (a) and (b) are views of the complex Gazebo world



(a)



(b)



(c)

Fig. 9: (a), (b), and (c) are the intermediate 3D map reconstructions collected by the swarm in the complex Gazebo world

V. CONCLUSION

It was found that the trooper drones move to frontiers much quicker than the leader drone since potential fields are computationally inexpensive compared to classic path-planning algorithms, whereas the leader drone captures more camera angles by rotating more and gets stuck less than the troopers because it is unaffected by local minima. Therefore, the swarm can balance the trade-offs of each type of drone. Because control is decentralized, it is easy to see that a swarm can capture much more information about the environment than any individual drone can in the same period of time.

Overall, our approach and implementation is effective in comprehensively mapping and exploring an unknown area, which would be useful for search and rescue operations where the more information the better. In the future, we would like to optimize the potential field parameters, as the trooper drones occasionally bump into obstacles. Furthermore, we would like to test with a larger number of drones, more realistic environments, and longer run-times, given we get access to more computational power.

REFERENCES

[1] Khatib, O. (1986). The potential field approach and operational space formulation in robot control. In Adaptive and Learning Systems: Theory and Applications (pp. 367-377). Boston, MA: Springer US.

[2] Howard, A., Matarić, M. J., & Sukhatme, G. S. (2002). Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In Distributed autonomous robotic systems 5 (pp. 299-308). Springer Japan.

[3] Chaimowicz, L., Michael, N., & Kumar, V. (2005, April). Controlling swarms of robots using interpolated implicit functions. In Proceedings of the 2005 IEEE international conference on robotics and automation (pp. 2487-2492). IEEE.

[4] Renzaglia, A., & Martinelli, A. (2010, July). Potential field based approach for coordinate exploration with a multi-robot team. In 2010 IEEE Safety Security and Rescue Robotics (pp. 1-6). IEEE.

[5] Goldstein, H., Poole, C., & Safko, J. (2002). Classical mechanics.

[6] M. Labbé and F. Michaud, "RTAB-Map as an Open-Source Lidar and Visual SLAM Library for Large-Scale and Long-Term Online Operation," in Journal of Field Robotics, vol. 36, no. 2, pp. 416–446, 2019.

[7] Moroni, J (2018) mapping_3d/worlds [Source code]. https://github.com/jacobmoroni/mapping_3d/tree/master/worlds

[8] Juarez, A (2021) hector-quadrotor-noetic [Source code]. https://github.com/RAFALAMAO/hector-quadrotor-noetic

[9] Lu, David et al. (2009) navigation [Source code]. https://github.com/ros-planning/navigation?tab=readme-ov-file

[10] Umari, H & Velumani, D (2016) rrt_exploration [Source code]. https://github.com/hasauino/rrt_exploration

APPENDIX I
TEAM MEMBER CONTRIBUTIONS

1) Anaam Mostafiz: Sections I, II and IV
2) Harshil Shah: Section III
3) Siddharth Jain: Sections II and IV
4) Tabsheer Askari: Sections I, II and III

APPENDIX II
VIDEOS OF SIMULATIONS

Robots spreading out in Python:
`https://drive.google.com/file/d/1dd6xoDiMtKmY93a9LqRK7mUUfND8doxy/view?usp=sharing`

Drones spreading out in Gazebo:
`https://drive.google.com/file/d/1OwcF2zFqYdAnUZkI6iUcYlcgCGoIcM2l/view?usp=sharing`

Drones exploring in Gazebo:
`https://drive.google.com/file/d/1ryf6GDeg-YkeXwkHBgh_mtcz-Zo9TiP5/view?usp=sharing`

APPENDIX III
CODE FOR SIMULATIONS

Our Github repository containing all project code and files:
`https://github.com/tellsiddh/ros-multirobot-search-rescue`