Multi Robot Systems Final Project: Debris Detection using Swarm Robots

I. Abstract

This project addresses the challenge of effectively navigating and mapping debris from a plane crash in a bounded 2D space (considering an ocean in the real world), using a decentralized swarm of robots. The goal is to develop efficient search and mapping strategies for dynamic disaster environments, with a particular focus on long-range reconnaissance missions and accurate debris detection.

A simulation framework in MATLAB has been developed for the behavioral modeling of swarms of autonomous surface vessels. In order to efficiently cover the area, this approach incorporates a hybrid methodology that combines the use of potential field methods and probabilistic techniques. Robots use information on the results from satellite reconnaissance in narrowing the search area down and finding stationary debris piles.

Simulation Assumptions

- Initial Debris Location: Satellite reconnaissance provides an approximate location of the debris before significant scattering occurs.
- Debris Stability: Debris piles are assumed to remain stationary over time.
- Accurate Detection: Robots are assumed to be 100% accurate in detecting debris during mapping.
- Precise Localization: Data localization is accurate through satellite imagery monitoring.
- Decentralized Operation: Robots rely on localized sensing and limited communication, eliminating single points of failure.

Performance Evaluation

The swarm performance is evaluated in terms of key performance indicators such as search time, accuracy, and energy consumption against the centralized approaches. The scalability and flexibility of the framework are demonstrated, which makes it suitable for real-world applications.

In the future, dynamic debris tracking will be integrated, and physical robotic platforms will be deployed to apply the developed distributed swarm technologies in realistic disaster response scenarios.

II. Mathematical Model

The goal is to simulate a multi-robot system in a 2D space, where robots navigate using potential fields derived from expected debris locations. Robots alternate between attraction to potential maxima and exploration, aiming to effectively map the environment while handling uncertainties in debris locations.

A. Environment

A 2D square grid is taken with the size of the side being 50 units. Then debris and robots are initialized positions.

$$\mathbf{r}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}, \quad i \in \{1, 2, \dots, N\},$$

Where N is the number of robots.

$$\mathbf{d}_j = \begin{bmatrix} x_j \\ y_j \end{bmatrix}, \quad j \in \{1, 2, \dots, D\}.$$

and D is the debris points.

Each robot has a sensing range and a communication range which are defined as:

R_s=10; and R_c=15;

Within Rs the robot senses the debris by sensors which could be a combination of Sonar and high-resolution RGBD cameras capable of detecting objects within reasonable distances and regardless of degree of occlusion. Also, within R_c the robot communicates and shares information of its individual map where it has gathered updated debris location information. Because of the project's focus on convergence of maps, we have abstracted the sensor working and assumed it to be ideal and noise free. Also for mapping purposes, we assume that robots are able to accurately localize using GPS.

B. Potential Field Formulation

Each debris point generates a Gaussian potential Function

$$P_j(x, y) = \exp\left(-\left[\frac{(x - x_j)^2}{2\sigma_x^2} + \frac{(y - y_j)^2}{2\sigma_y^2}\right]\right)$$

- $\mathbf{d}_j = (x_j, y_j)$: Position of the j-th debris,
- σ_x, σ_y: Spread parameters of the Gaussian.

The total potential field P(x,y) is the sum of contributions from all debris points:

$$P(x, y) = \sum_{j=1}^{D} P_j(x, y).$$

C. Robot Dynamics

The potential gradient guides robot movement. For each robot *i*, the gradient at its position $r_i=(x_i,y_i)$ is**:**

$$\nabla P(\mathbf{r}_i) = \begin{bmatrix} \frac{\partial P}{\partial x} \\ \frac{\partial P}{\partial y} \end{bmatrix},$$

where:

$$\frac{\partial P}{\partial x} = \sum_{j=1}^{D} \left(-\frac{(x_i - x_j)}{\sigma_x^2} \right) P_j(x_i, y_i),$$

$$\frac{\partial P}{\partial y} = \sum_{j=1}^{D} \left(-\frac{(y_i - y_j)}{\sigma_y^2} \right) P_j(x_i, y_i).$$

The robot's velocity v_i is derived from the gradient but normalized to maintain a constant speed:

$$\mathbf{v}_i = \frac{\nabla P(\mathbf{r}_i)}{\|\nabla P(\mathbf{r}_i)\|} v_{\text{norm}}$$

where v_{norm}=1.0 is the desired speed.

The robots motion is updated iteratively using:

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \mathbf{v}_i \Delta t,$$

where $\Delta t = 0.01$ is the time step.

D. Phase Switching

Robots alternate between attraction and exploration phases:

- Attraction Phase:
  1. Robots move up the potential gradient.
  2. If the robot remains near a maximum for $T_{attraction}$=500 steps, it switches to exploration.

- Exploration Phase:

  Robots reverse their gradient and add noise:

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) - \mathbf{v}_i \Delta t + \boldsymbol{\eta},$$

where $\boldsymbol{\eta} \sim \mathcal{U}(-\eta_{\max}, \eta_{\max})$, with $\eta_{\max} = 0.25$.

After $T_{exploration}$=1000 steps, the robot switches back to attraction.

Robots maintain occupancy grids to track detected debris:

1. Each robot updates its occupancy grid when debris is sensed:

   $G_i(x,y)=0$ if debris detected at (x,y).

2. The grid is visualized as a binary occupancy map after the simulation.

E. Map Sharing

The robots share their individual occupancy grid maps when they move within communication range R_c of each other. A key thing to note is that the communication network formed by the robots is time-varying in nature with robots coming in and out of the range of each other at any timestep of the process.

For Robot i, the update rule for updating the value of jth grid cell in a neighborhood of N_i robots is:

$$P_i^j(k+1) = P_i^j(k) \cdot \prod_{n \in N_i(k)} \left(P_n^j(k)\right)^{a_{in}(k)},$$

where,

$P_i^j(k)$: Occupancy value of cell j in the map of robot Ri at time step k. The value ϵ {0,1}.

$N_i(k)$: Set of neighbors of robot Ri at time k.

$a_{in}(k)$: Entry in the adjacency matrix A(k), representing the weight of the influence of neighbor Rn on robot Ri.

Adjacency matrix here is of order N x N for a network of N robots which can be represented by a graph in which the nodes are robots and edges are the communication link between robots. The robots are considered neighbors if they have an edge between each other in the graph, i.e. they are within communication range of each other directly.

III. Theoretical Analysis

The primary focus of this project is to prove convergence of maps from individual robots from a swarm. The primary property of the mathematical model we intend to prove is the consensus between multiple agents with respect to their maps.

As per our mathematical model, for all Robots R_i, we have a map M_i, each of which has | D | grid cells. As time $t \rightarrow \infty$, the difference between values of each j^{th} grid cell in each map M_i should $\rightarrow 0$ for all i=1,2..,N. In other words, all maps at $t = \infty$ are identical and maps from any robot can be chosen at random should accurately reflect the state of the environment.

To do so, we use notation similar to [3] to describe consensus over time-varying graphs. Similar to approach used in [1], we define G(k) = (V,E(k)) as an undirected time-varying graph with n vertices, V = {1,...,n}, and a set of undirected edges E(k) at time step k. In our scenario, G(k) defines the communication network of the robots at time step k, in which the vertices represent the robots, $V = \{1,...,N_r\}$, and each edge $(i,\hat{n}) \in E(k)$ indicates that robots Ri and R $\hat{n}$ are within broadcast range of each other at time step k and can therefore communicate.

Let $A(k) = [aij(k)] \in R\ n\times n$ be the adjacency matrix associated with graph $G(k)$ at time step k, where $aij(k)$ denotes the element in the i th row and j th column of $A(k)$. In this matrix, $aij(k) \neq 0$ if and only if an edge exists between vertices i and j at time step k, and $ai\ j(k) = 0$ otherwise. The set of neighbors of vertex i at time step k, defined as $N_k^i = \{\hat{n} \mid (i,\hat{n}) \in E(k), i \neq \hat{n}\}$, contains the vertices j for which $ai\ j(k) \neq 0$. Suppose that at time step k, each vertex i is associated with a real scalar variable $xi(k)$ (in context of our implementation, this variable is the jth grid cell in any map). At every time step, the vertex updates its value of $xi(k)$ to a weighted linear combination of its neighbors' values and $xi(k)$, where the weights are the corresponding values of $ai\ j(k)$.

Then the vector $x(k) = [x1(k) \ldots xn(k)]T$ evolves according to the discrete-time dynamics $x(k+1) = A(k)x(k)$. If $\lim k\to\infty\ xj(k) = 1\ n \sum n\ i=1\ xi(0)$ for all $j \in V$, then the vertices are said to have achieved average consensus. It is proved in [4, Theorem 1] that the dynamics of $x(k)$ converge asymptotically to average consensus if $A(k)$ is a doubly stochastic matrix, meaning that each of its rows and columns sums to 1, and if there exists a time interval for which the union of graphs over this interval is connected. This result is at the core of the map sharing rule and validation of it in the simulations.

Assumptions:

1. Since the primary objective is to demonstrate convergence, we have not taken into account any sensor noise or errors in classification of a particular grid cell.
2. Similar to [1], we assume that there exists a time interval over which the union of communication graphs $G(k)$ is connected. It is reasonable to assume that this holds true in case of a large enough communication radius with respect to domain size and high density of robots. Although this is difficult to prove for arbitrary domains.
3. In [1], they have defined the adjacency matrix based on the number of neighbors at timestep k. If Robot i has N^i_k neighbors at timestep k, then adjacency matrix is defined as $aii(k') = 0.5$, $ai\hat{n}(k') = 0.5$ where $\hat{n}=1,2\ldots N_k^i$ and k' is subsequent timestep k+1, k+2…, $k+N_k^i$. We also adopt this assumption, because this ensures that the adjacency matrix remains doubly stochastic regardless of the topology of the robot communication graph which is a criteria for average consensus.
4. Further in [1], they have also assumed that updates from 1st neighbor will happen at k+1, 2nd neighbor at k+2 and so on. In our case, we update maps from all neighbors at the same timestep k+1. This is done to ensure that during random motion, robots may step in and out of communication range before an update even occurs. So performing updates in the same timestep ensures that convergence occurs faster.
5. There are no inaccessible grid cells in the specified domain. That is there are no obstacles which the robot needs to avoid which includes the debris as well. The only constraint on the robot motion is the boundary of the domain.

Suitability of the model:

In the update rule to share maps between robots in a network, we use the result from [4, Theorem 1], which ensures that the maps and by extension each grid cell of the maps achieve average consensus. This approach is suitable for our application because it ensures that all robots' maps asymptotically converge to a shared representation of the environment.

Considering properties of union of graphs being connected and adjacency matrix being doubly stochastic at all timesteps (assumptions 2 and 3), These properties ensure information propagates reliably among robots, averaging their maps iteratively to achieve agreement on grid.

Although analysis shows asymptotic convergence of the robots' maps to the actual map, our simulations results in Section IV show that the maps indeed converge in finite time.

IV. Validation in Simulation and Results:

Link to Matlab Simulation video:
https://drive.google.com/file/d/15hf4kMueFnh191sKs0oCrCHNZhWtqvvl/view?usp=sharing

Link to Matlab code:
https://drive.google.com/file/d/1QrW8i1q8r1ZOvz375svaSCGIAY_8Hx6Z/view?usp=sharing

The validation of the proposed multi-robot behavior was carried out using MATLAB-based simulations within a 2D grid environment.

In this simulated scenario, a swarm of robots was assigned the task of locating and mapping debris from a hypothetical plane crash. The grid environment consists of a 50×50 bounded space, initialized with a designated number of debris locations and robots starting from random positions.

The robots operated under a decentralized control strategy, utilizing Gaussian potential fields generated by the debris, along with an exploration component to navigate potential field local minima. The simulations effectively demonstrated the robots' behavior in both cooperative debris localization and environmental coverage.

Grid Specifications:

1. Dimensions:

- 50×50 units.
- Environment: Bounded, with no dynamic obstacles.

2. Robot and Debris Properties:

- Number of robots N=10
- Number of debris points D= 10.
- Sensing range (R_s)= 10 units (accurate detection within this radius).
- Communication range (R_c): 15 units (data sharing among robots).

3. Behavioral Control Parameters:

- Gaussian potential field influence parameters: $A_k = 1$ and $\sigma = 3.0$.
- Velocity constraints: Maximum speed normalized to ensure smooth robot motion.

4. Simulation Duration:

- Number of Time step: 10000

Result

1. Debris Localization and Convergence of maps

Robots successfully identified all debris points. The trajectory plots show the robots converging on estimated debris locations over time. The dynamic path planning of the robots was based on the concept of Gaussian potential fields.

Maps of most of the robots successfully converged on the true debris locations, although due to the decentralized nature of the controller, some robots (2, 6, and 9, in the below iteration) could not completely populate their maps with all possible locations. This limitation can be addressed by increasing the noise strength in the robot motion model, which can encourage exploration and potentially lead to improved global convergence.

2. Scalability

The system underwent testing with different densities of robots and debris. Simulations involving 10 robots and 10 debris points showed reliable performance, validating the scalability of the proposed model.

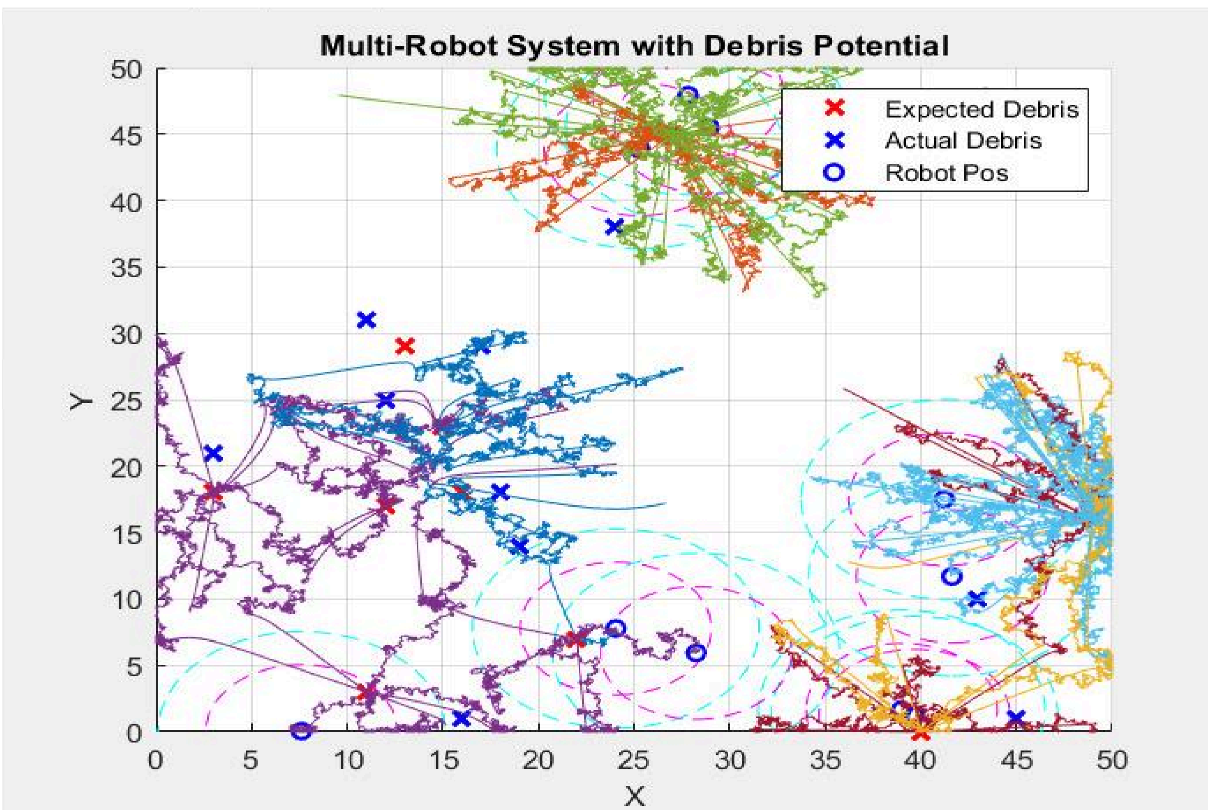

Fig 1: Showing trajectories of all Robots.

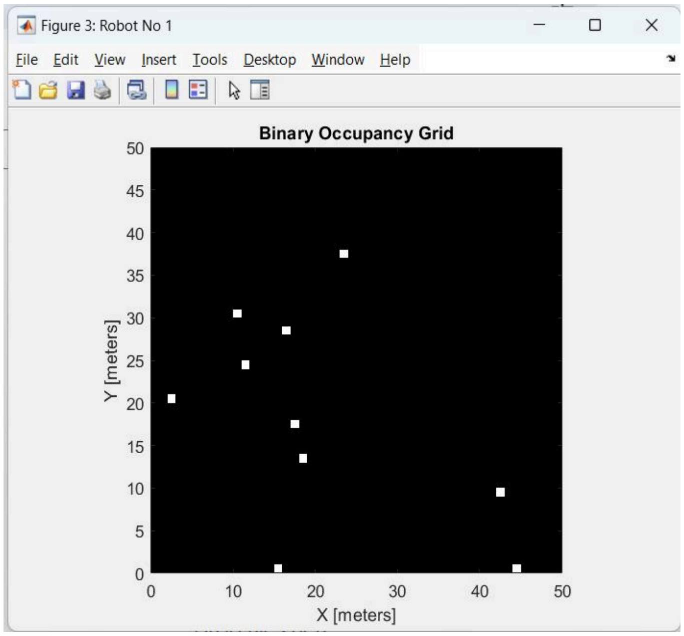
Figure 3: Robot No 1
File Edit View Insert Tools Desktop Window Help
Binary Occupancy Grid
Y [meters]
X [meters]

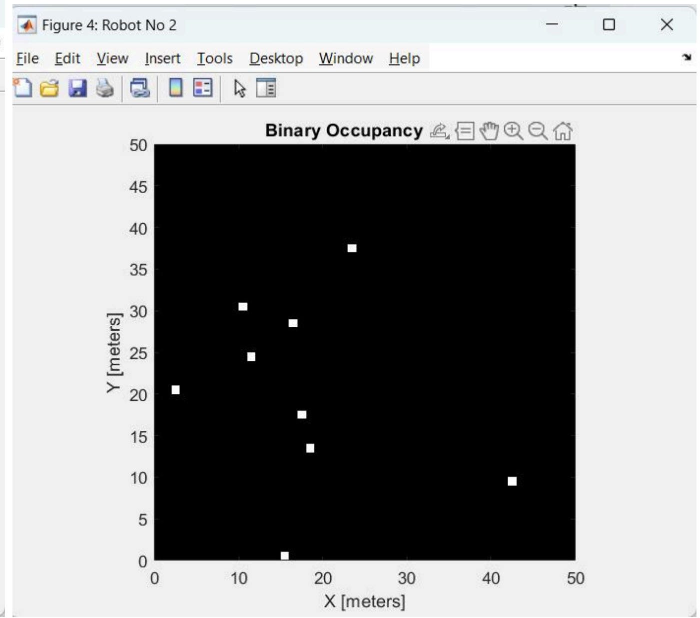
Figure 4: Robot No 2
File Edit View Insert Tools Desktop Window Help
Binary Occupancy
Y [meters]
X [meters]

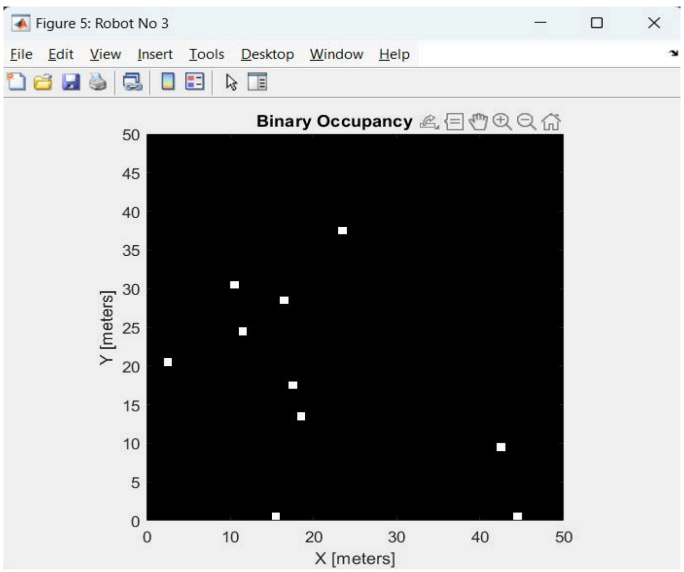
Figure 5: Robot No 3
File Edit View Insert Tools Desktop Window Help
Binary Occupancy
Y [meters]
X [meters]

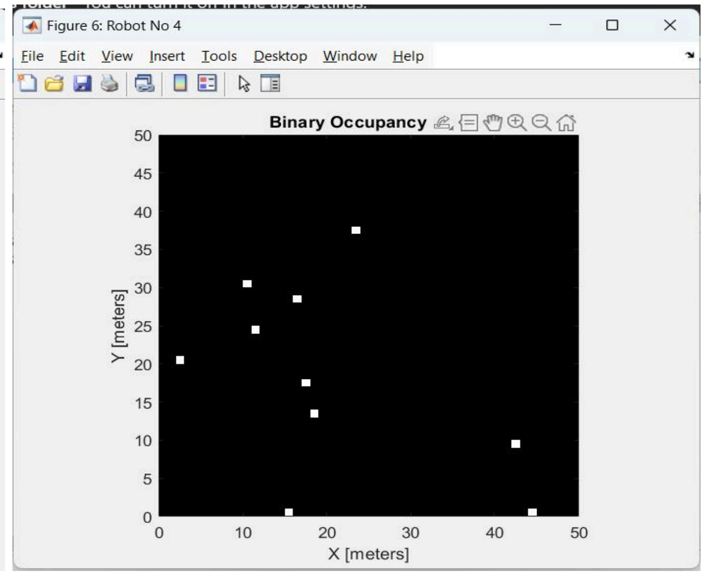
Figure 6: Robot No 4
File Edit View Insert Tools Desktop Window Help
Binary Occupancy
Y [meters]
X [meters]

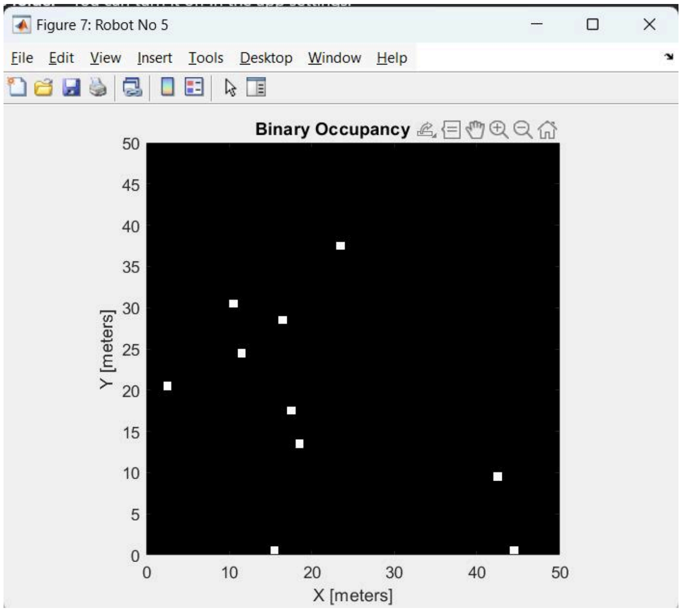
Figure 7: Robot No 5
File Edit View Insert Tools Desktop Window Help
Binary Occupancy
Y [meters]
X [meters]
0 5 10 15 20 25 30 35 40 45 50
0 10 20 30 40 50

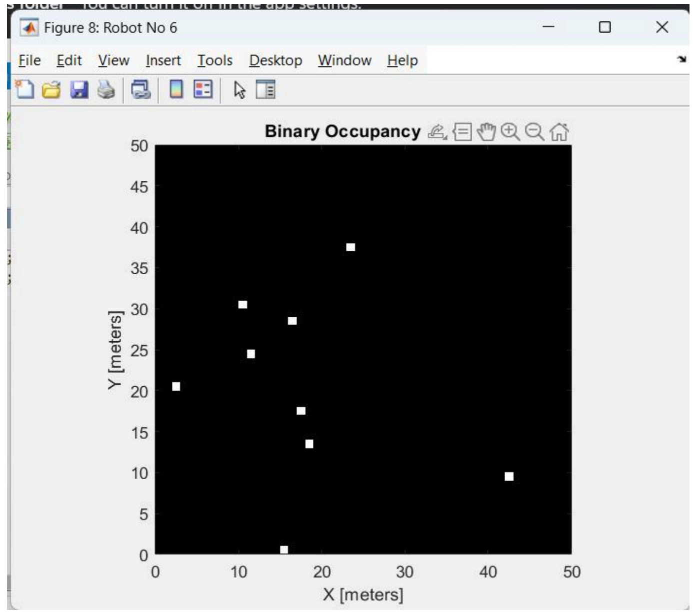
Figure 8: Robot No 6
File Edit View Insert Tools Desktop Window Help
Binary Occupancy
Y [meters]
X [meters]
0 5 10 15 20 25 30 35 40 45 50
0 10 20 30 40 50

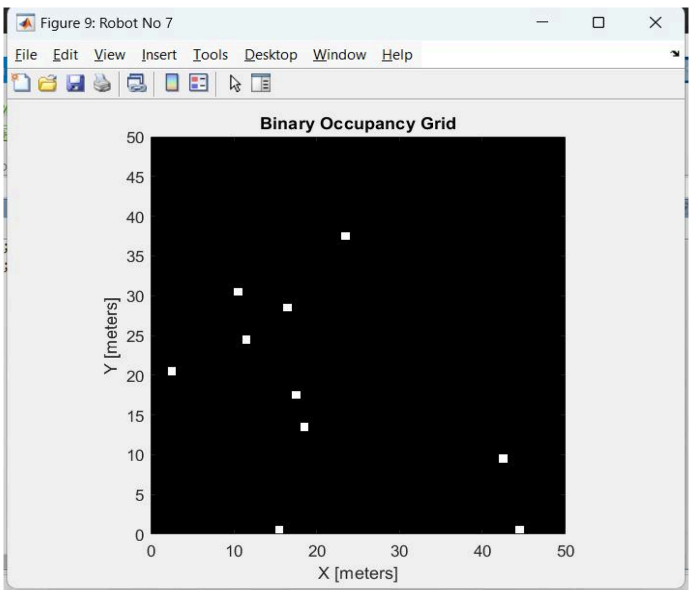
Figure 9: Robot No 7
File Edit View Insert Tools Desktop Window Help
Binary Occupancy Grid
Y [meters]
X [meters]
0 5 10 15 20 25 30 35 40 45 50
0 10 20 30 40 50

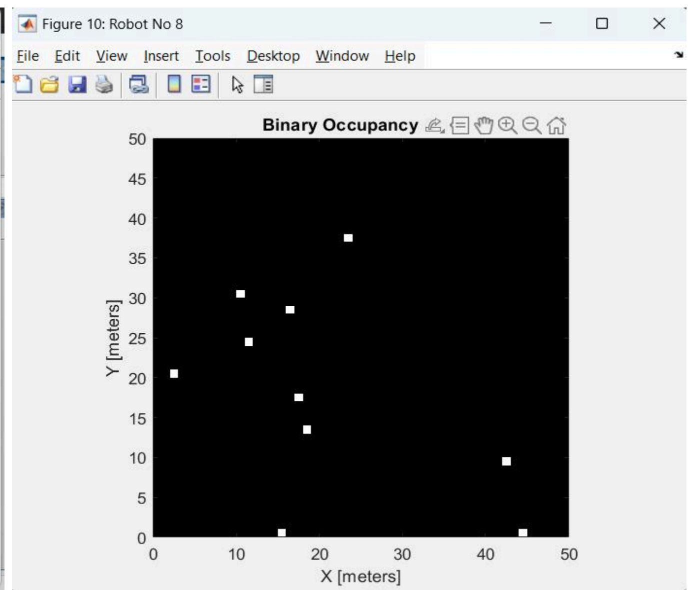
Figure 10: Robot No 8
File Edit View Insert Tools Desktop Window Help
Binary Occupancy
Y [meters]
X [meters]
0 5 10 15 20 25 30 35 40 45 50
0 10 20 30 40 50

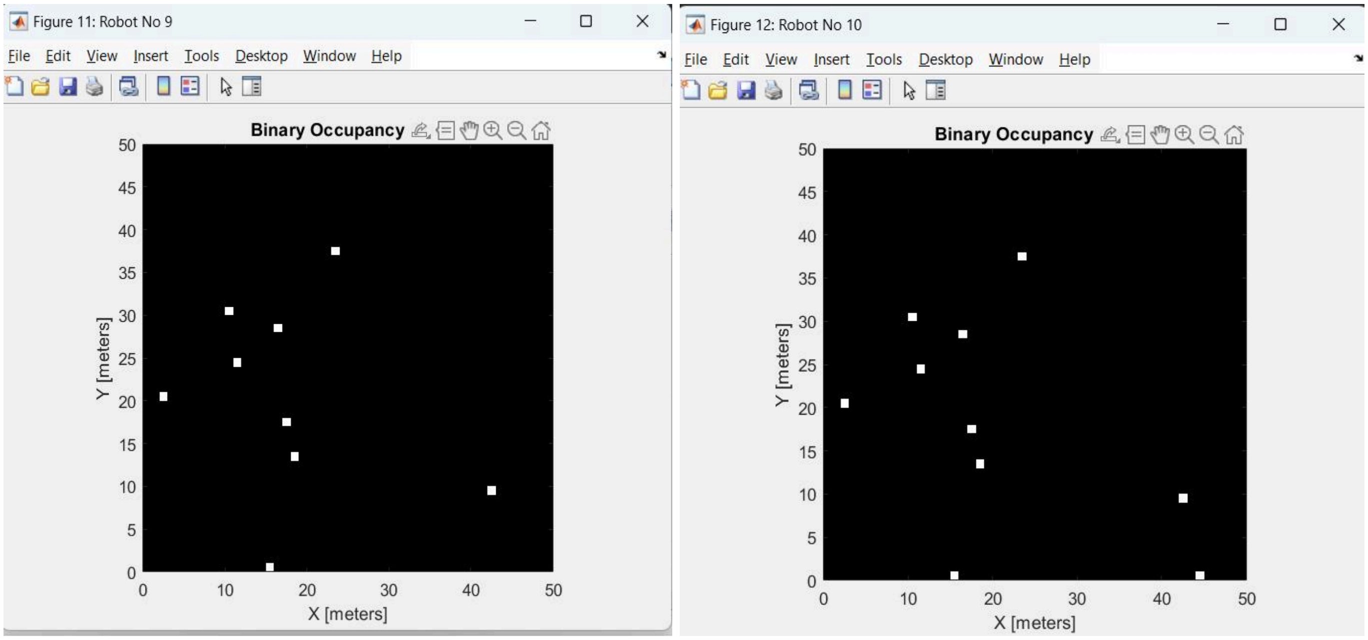

Fig. 2 Showing final binary occupancy grid maps for N=10 robot system. All robots except 2, 6 and 9 were able to converge on a map with the true position of all point debris. Robot's 2, 6 and 9 failed to populate all points due being out of network for too long causing issues with consensus.

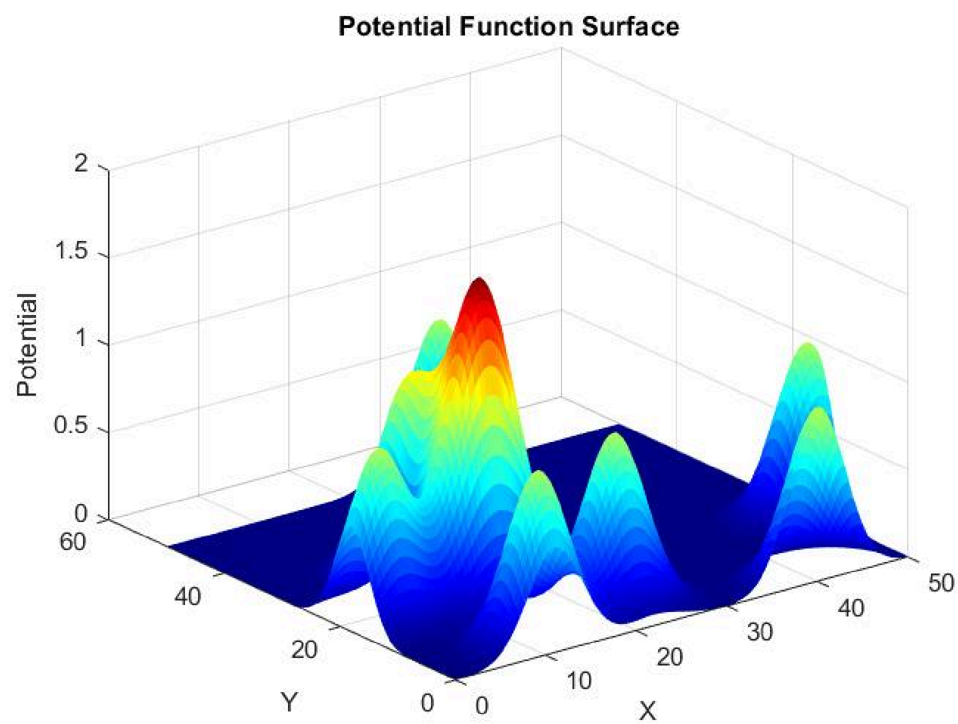

Fig. 3: Cumulative potential function from individual potential function modelled for each point debris. Acts as guidance for robot motion by setting general direction of movement for all robots

Video Demonstrations

Simulation videos were produced to showcase how the robot swarm behaves:

1. **Initialization**

   Robots and debris are randomly placed at the beginning of the simulation.

2. **Robot Movements**

   The robots follow dynamic paths influenced by potential fields and exploration parameters.

3. **Debris Detection**

   The robots map debris and populate their occupancy grid maps as they explore the domain.

4. **Final State**

   At the end of the simulation, any robot's occupancy grid map can be chosen for locating point debris since all robot's maps have converged to actual debris maps.

MATLAB Code Details:

- **Environment Initialization:**

  This step involves defining the grid, setting up the positions of debris and robots, and establishing the necessary parameters.

- **Potential Field Computation**

  Here, Gaussian functions are applied to compute the total field values at each point on the grid.

- **Control Logic**

  This part utilizes gradient descent, incorporates exploration terms, and follows collision avoidance rules.

- **Visualization and Analysis**

  This phase creates dynamic plots, and outputs from the simulation.

Future Scope of the Project

This project showcases a strong method for multi-robot swarm systems aimed at detecting and mapping debris in a controlled setting. However, there are still several challenges to address when moving from simulation to real-world scenarios, especially in harsh or dynamic environments. In this future scope, we plan to test and refine the system by incorporating new elements such as environmental conditions, enhanced robot capabilities, and better coordination among the robots. The main areas for future development include:

1. Testing in Harsh Environments

Environmental Factors: The current model is based on a static, idealized environment. In reality, the locations of debris can be affected by external factors like wind, rain, or changes in ocean currents. In challenging environments, debris may scatter in unpredictable ways, adding complexity to localization efforts. Future work will focus on adapting the system to manage these dynamic factors, such as utilizing sensor fusion techniques to anticipate debris movement or developing more intricate models that consider environmental noise.

Terrain Variability: Besides aquatic surface level exploration, Robots need to navigate various types of terrain, including rugged, sandy, or uneven surfaces. We will assess the robots' ability to adjust their movements in response to terrain feedback, and investigate specialized locomotion models (e.g., wheeled versus legged robots) to enhance exploration capabilities in these environments.

2. Increased Realism and transfer to Real world

The robots in the current simulation function under the assumption of perfect sensing within a specific range. Future iterations will assume a more realistic sensor working by incorporating real-world noise and errors in classification of grid cells. Further, this project can be expanded to hardware. This will involve integrating high-definition cameras, LIDAR, and Sonar sensors to simulate mapping on robust autonomous hardware.

3. A special focus on particular parts of the aircraft

In addition to enhancing the system's overall performance in challenging environments, a key focus for future efforts will be on pinpointing specific components of the aircraft, particularly the black box. This crucial device plays a vital role in investigations and recovery efforts following a plane crash, as it holds essential flight data that can help quickly ascertain the cause of the incident. Enabling the robot swarm to prioritize the detection and localization of high-priority items like the black box will represent a significant advancement.

Conclusion

These simulations demonstrate that the suggested Gaussian potential field-based method is effective for decentralized multi-robot systems in tasks like debris localization and environment mapping. Future research could focus on incorporating dynamic obstacles and imperfect sensing to improve the model's applicability in real-world scenarios.

The future direction of this project aims to enhance the adaptability, coordination, and performance of the multi-robot swarm in challenging environments. By improving the capabilities of the robots, fine-tuning coordination methods, and conducting tests in dynamic, real-world scenarios, we can develop a more dependable and efficient system for complex tasks like debris detection and environmental mapping. These improvements will make the system more applicable for real-world uses in disaster recovery, search and rescue, and environmental monitoring, where difficult conditions and significant uncertainty are often present.

V. References

[1] Ramachandran, Ragesh K., Zahi Kakish, and Spring Berman. "Information correlated Lévy walk exploration and distributed mapping using a swarm of robots." IEEE Transactions on Robotics 36.5 (2020): 1422-1441.
[2] Wang, Jinhuan & Hu, Xiaoming. (2010). Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications (Ren, W. and Beard, R.W.; 2008) [Book Shelf]. IEEE Control Systems Magazine - IEEE CONTROL SYST MAG. 30. 85-86. 10.1109/MCS.2010.936430.
[3] C. Godsil and G. Royle, Algebraic Graph Theory, ser. Graduate Texts in Mathematics. New York: Springer-Verlag, 2001, vol. 207.
[4] D. B. Kingston and R. W. Beard, “Discrete-time average-consensus under switching network topologies,” in Proceedings of the American Control Conference, June 2006.