Search and Exploration using Multi-robot Systems in Hazardous Environments

Vamsee Krishna Kella(1220077079)

Shivani Bandirala(1220073751)

Aswayuja Koduri(1223390480)

I. Abstract

In recent years, the multi-robot systems have seen an increasing penetration across multiple fields and disciplines which are not just confined to software engineering and artificial intelligence communities. This is due to the usage of multi-robot systems in solving complicated issues in different fields that are troublesome or incomprehensible for single robots or systems [1].

Here, one of such applications is considered which can potentially be solved by a multi-robot system rather than a single robot system. It is a discovery in which the focus is on surveillance, search, and rescue operations in hazardous environments.

The chosen multi-robot behavior is Combining distributed data. It is meant to solve a problem more quickly and efficiently than traditional approaches by compromising optimality, accuracy, precision, or completeness in exchange for speed. MATLAB will be used as a simulation platform to compare the mentioned uses of exploration techniques.

The type of model chosen to describe the behavior is the LIDAR model. LIDAR (Light Detection and Ranging) is a technology used to make high-resolution models of ground elevation.

Two properties of behavior are briefly discussed in the paper. One being the equilibrium/equilibria or invariant set of a system, computed from the equations that describe its kinematics or dynamics. A proposition and a theorem are cited to prove the property. The other one is the property proving the asymptotic stability of the system. A brief explanation of reasoning that is based on physical properties of the system is stated as a proof.

The analysis technique considered is object detection. Object detection is a technique related to computer vision and image processing which deals with detecting instances of semantic objects of a certain class in digital images and videos.

The algorithms are chosen for path planning and object detection. RTT algorithm is used for path-planning whereas Viola-Jones algorithm is used for object detection. MATLAB Simulink is used as a simulation software for verifying the performance of each algorithm. The simulation results show that object detection is successfully established based on the color of the object in the specified environment.

II. Mathematical Model

In this report, the primary behavior under study is the ‘Combining distributed data: ex) Collective perception, collective localization, mapping, object detection and classification.’

A. Scenario:

- Based on the behavior characteristics, a hazardous or inaccessible environment is considered. The properties of the environment are: the environment is bounded; Different objects are present in the vicinity described.
- A LIDAR model is created with three multi-robot systems. It gathers information about the environment by travelling to the boundary between known and unknown areas.
- The robots are connected to a network and share local data with the team as communication knowledge is limited in hazardous, and unknown environments. The explorer robots investigate the area and transfer data to a centralized system at a certain period.

B. Assumptions and Constraints:

<u>Assumption 1:</u> (Centrosymmetric) The robots’ attachment points are centrosymmetric around the center of mass of the object, meaning that for any robot i, there exists another robot j (not equal to i) such that $r_i = -r_j$.

Under this assumption, the centrifugal term will not have any influence on the consensus because it points towards the center of mass and will be canceled out by the paired symmetric robots.

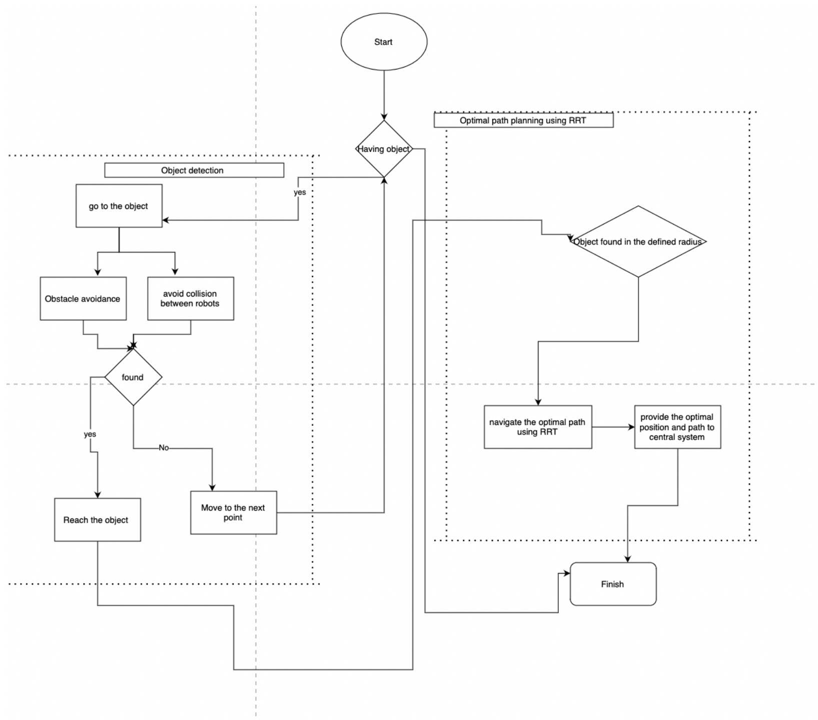
<u>Assumption 2:</u> The task is completed when the object detection is done. The assigned robots do come to their initial state when one of them detects the object.

<u>Assumption 3:</u> The robots’ sensing capability is limited to object detection based on the color of the object.

<u>Assumption 4:</u> The environment is bounded. Within these bounds it is reasonable to assume that the robots maintain full communication with the ground station throughout the mission.

C. Model:

A LIDAR model is created with three multi-robot systems. The multi-robot LIDAR sensors are used to accomplish the objective which are further connected to robot visualizers.



Let a typical observation equation for a LiDAR sensor be stated as $z_i = h(x_i) + \delta_i$, where x represents both the pose of the sensor and the landmarks, z is the acquired measurement, h(x) is some function relating x with z and δ is a zero-mean Gaussian noise vector.

An arbitrary second order algebraic surface in R^3 is given by the general equation of the second degree in x, y and z. The equation of such a surface is

$$
\begin{aligned}
&F(x,y,z) = \\
&\quad Ax^2 + By^2 + Cz^2 + Fxy + Gyz + \\
&\qquad\qquad\qquad Hxz + Jx + Ky + Lz + M = 0.
\end{aligned}
$$

$$
\text{Let } \mathbf{Q} = \begin{pmatrix} A & 1/2F & 1/2H \\ 1/2F & B & 1/2G \\ 1/2H & 1/2G & C \end{pmatrix}, \mathbf{J} = \begin{pmatrix} J \\ K \\ L \end{pmatrix} \text{ and } \mathbf{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}
$$

Then the matrix form of Equation will be $x^T Qx + J^T x + M = 0$. If the coefficients J = K = L = M = 0, then we get the general second-degree homogeneous equation (all terms with power 2) in the variables x, y and z. This simplifies into, $x^T Qx = 0$

III. Theoretical Analysis

Properties:

1) **Model's equilibrium state:** One can show that the network as a whole will reach a static equilibrium (i.e., a situation in which all nodes are stationary) by considering the total energy of the system. Each node has both potential and kinetic energy: the former arises from the node's interaction with the potential field, the latter from the node's motion.

Here, a theorem is adapted to prove the equilibrium of the model.

Theorem: Under the centrosymmetric assumption, the system will reach a consensus on all forces. The consensus value is the average of all the initial forces.

Proof: We can decompose the force as,

$F = s_x 1_x + s_y 1_y + \delta$, $s_x, s_y \in R$,

where $(s_x 1_x + s_y 1_y)$ is the state of consensus and δ is the group disagreement vector [2]. Then we have

$$\begin{aligned}\dot{F} = \dot{\delta} &= (-L_a - \frac{M}{J} R_a(t))(s_x \mathbf{1}_x + s_y \mathbf{1}_y + \delta) \\ &= (-L_a - \frac{M}{J} R_a(t))\delta.\end{aligned}$$

Knowing the dynamics of the disagreement vector, we can choose the Lyapunov function to be $V = 1/2\, \delta^T \delta$.

Then according to Proposition 2 we have

$$\begin{aligned}\dot{V} &= \delta^T \dot{\delta} = \delta^T (-L_a - \frac{M}{J} R_a(t))\delta \\ &\leqslant \lambda_{2N}(-L_a - \frac{M}{J} R_a(t))\|\delta\|^2 \leqslant 0,\end{aligned}$$

$$\ddot{V} = 2\delta^T \left(-L_a - \frac{M}{J} R_a(t)\right)^2 \delta + \delta^T \left(-\frac{M}{J} \dot{R}_a(t)\right) \delta.$$

Since δ is bounded, we have F is bounded, then ω is bounded and finally $\dot{R}a(t)$ is also bounded. According to Barbalat's, V >= 0, $\dot{V}$ <= 0 plus $\ddot{V}$ bounded will ensure that $\dot{V} \to 0$, i.e., δ will converge asymptotically to the invariant set Ω, where $\Omega = \{\delta \mid \delta = p_x 1_x + p_y 1_y, p_x, p_y \in R\}$.

Therefore, F will converge to its equilibrium, $(s_x + p_x)1_x + (s_y + 1_y)1_y$, s_x, s_y, p_x, $p_y \in R$, which is the state of consensus since all the forces will be equal. Finally, from (7) we know $\sum N_i{=}1\ \dot{F}_i = 0$, meaning that the total force of the group is conserved during the entire process. Hence F will converge to the average of all the initial forces.

2) Stability characteristics (asymptotically stable):

An equilibrium is reached when $v_i = 0 \Rightarrow \dot{q}_i = 0$ and $u_i = 0 \Rightarrow \nabla f^2(q_i) = 0 \Rightarrow f(q_i)\nabla f(q_i) = 0$. The goal is to minimize error associated with convergence to the function $f(q_i)$ given as

$$E(q) = \frac{1}{2}\sum_i f^2(q_i)$$

this is minimized when:

$$\sum_i f(q_i)\nabla f(q_i)\dot{q}_i = 0$$

This is true because when f(qi) lies on the targeted curve or surface, $f(q_i) = 0$ so the entire term goes to zero. Then we can consider the Lyapunov function $V(q, \dot{q})$ given by

$$V(q,\dot{q}) = k\phi(q) + \frac{1}{2}\sum_i \dot{q}_i{}^T \dot{q}_i$$

. This function has an elegant interpretation as the sum of potential energy due to the field and the sum of the kinetic energy of the system. φ can be defined as

$$\phi(q) = \sum_i f^2(q_i)$$

$$\dot{V}(q,\dot{q}) = k\sum_i 2f(q_i)\nabla f(q_i)\dot{q}_i + \sum_i \dot{q}_i(-k\nabla f^2(q_i, p_{obj}, R) - C\dot{q}_i) = \ldots$$

Then we get,

$$k\sum_i 2f(q_i)\nabla f(q_i)\dot{q}_i - \sum_i \dot{q}_i(k\nabla f^2(q_i) + C\dot{q}) = -\sum_i C\dot{q}$$

Then clearly $\dot{V}(q, \dot{q})$ is monotonically decreasing and therefore the system is asymptotically stable. Due to the damping term and design of $f(q_i)$, any given initial condition with no input will approach an equilibrium onto the desired curve or surface.

IV. Validation in Simulations

Simulations are performed in MATLAB Simulink in order to validate the model properties described in section III.

A. MATLAB Simulation

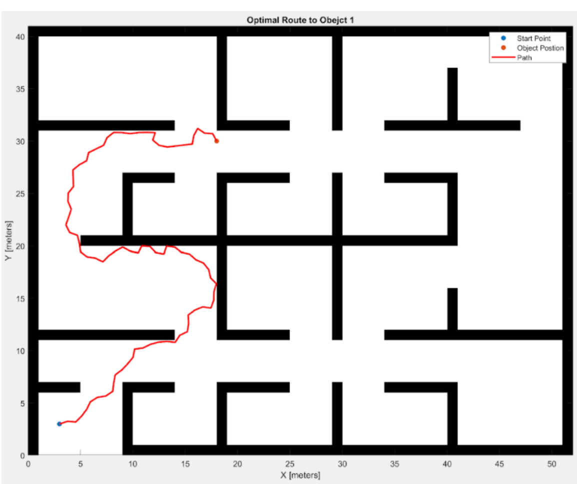
i. Algorithms:

- **RRT algorithm for Path Planning:**

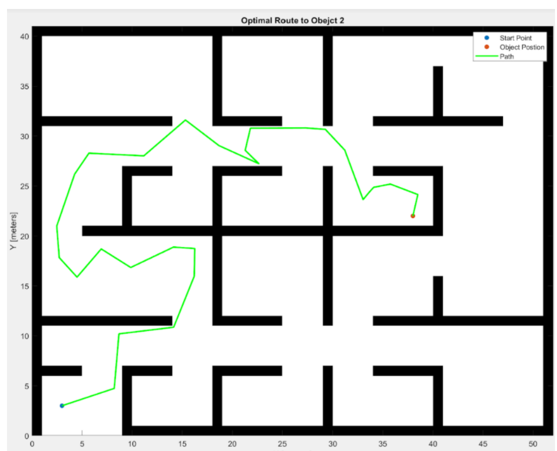
  A rapidly exploring random tree (RRT) is an algorithm designed to efficiently search high-dimensional spaces by randomly building a space-filling tree. The tree is constructed incrementally from samples drawn randomly from the search space and is inherently biased to grow towards large unsearched areas of the problem.

  The path is planned by building a tree starting from the position of the robot. When a point in the space is randomly sampled, it is checked if that point collides with an obstacle in the space. If the sampled point has no collisions, it is then checked if the straight-line path between the sampled point and the nearest existing point in the tree has any collisions. If this straight-line path has no collisions, the sampled point is added to the tree with the nearest point as its parent node. If there is a collision, this point is thrown out.

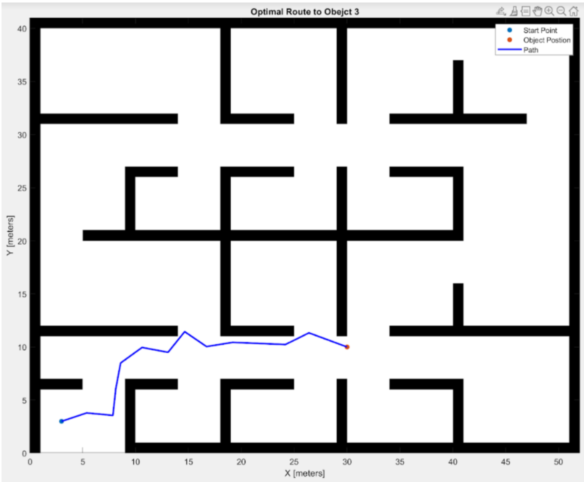
  Each time after a node is added to the tree and the node is less than some threshold distance from the goal position, it is checked if the goal can be reached in a straight-line path from the added node. If the goal position is reachable, the goal position is added to the tree with the recently added node as its parent. At this point, the path planning is complete. If the goal position is still unreachable, additional points are sampled.



RRT Image of Object 1 path



RRT Image of Object 2 path

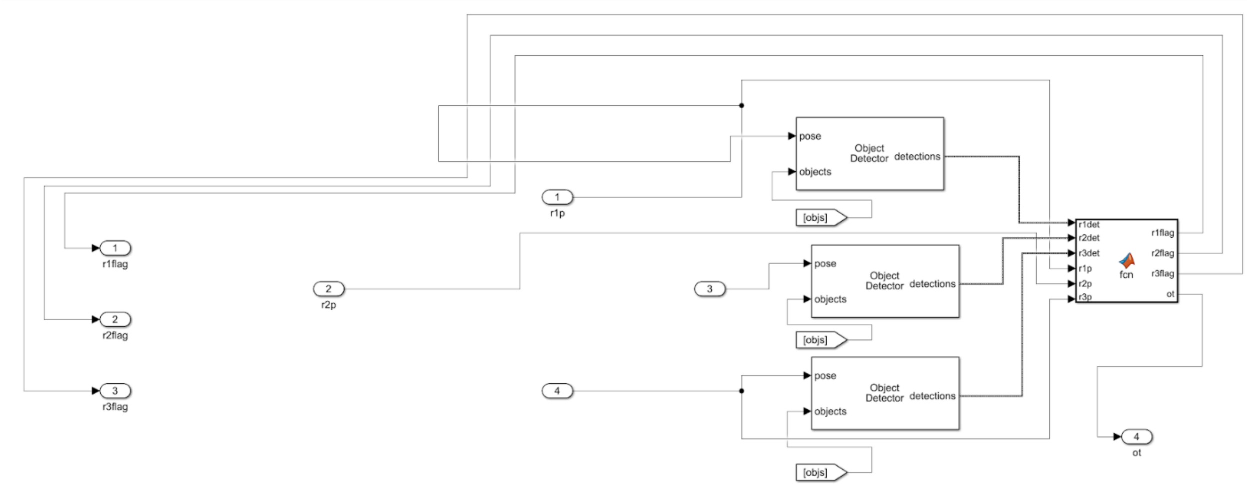


RRT Image of Object 3 path

- **OpenCV – Viola-Jones algorithm for Object Detection:**

  The algorithm is based on machine learning. The first step involves training a cascade function with a large amount of negative and positive labeled images.

  Once the classifier is trained, identifying features, namely "HAAR Features," are extracted from these training images. HAAR features are essentially rectangular features with regions of bright and dark pixels.



Position Detector

ii. ***Performance Evaluation:***

The performance of RRT and Viola-Jones algorithms are evaluated by using Simulated Annealing as the benchmark. Several simulations with different

sets of task locations are used in this comparison. It is observed that Simulated Annealing provides the optimal path.
Few challenges were faced when simulations were done with ROS.

V. Hardware Developments

Components use to build the model

· YDLIDAR X4 360° Laser Scanner

Specifications:

Field of View	360 degrees
Sample rate	3000Hz
Scanning Frequency	4-8 Hz
Scanning rate	0.05m to 10m
Communication type	UART 3.3v to USB 5v
Operation Voltage	4.8v - 5.2v

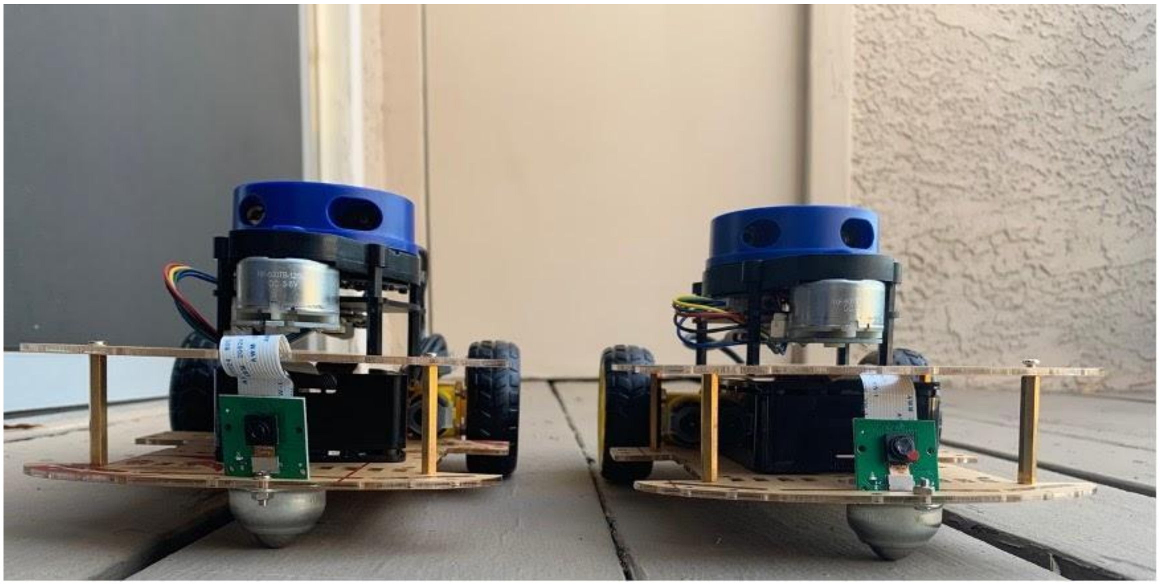
· Raspberry Pi Camera Module V2

Specifications:

Resolution	5 Megapixel
Sensor type and resolution	OmniVision 5647; 2592x1944 pixels
Video format	1080p @30fps, 720p@60fps
Dynamics range	67db @8xgain
fixed focus	1m - Infinity
focal length	3.60mm+/- 0.01
Horizontal view	53.50+/-0.13 deg
vertical view	4.41+/-0.11 deg

· Motor drivers l298

- DC Motor Wheel, 4PCS TT Motor 3-6V Dual Shaft Gear Motor



Hardware model of LIDAR based multi robots

VI. Conclusion

In this report, we discussed the application of search and exploration in hazardous environments using multi-robot systems. A LiDAR-based model is chosen for the same. Two properties of equilibrium state and asymptotic stability are discussed for the model.

We then developed simulations in MATLAB to test our models, showing the effectiveness of each algorithm. The simulations are done in two phases: Path planning and Object detection. The algorithms used are RRT and Viola-Jones for path-planning and object detection respectively. The simulation results established the efficiency of the robots successfully detecting the objects based on color.

A few challenges were encountered during the simulations using ROS which further hindered the testing of the hardware model of the project.

VII. Division of Labor:

Section II (Mathematical Model):

LIDAR model - Koduri, Aswayuja

Section III (Theoretical Analysis)

Model's equilibrium state - Koduri, Aswayuja;

Stability characteristics - Kella, Vamsee Krishna; Bandirala, Shivani

Section IV (Validation in Simulations or Experiments)

RRT algorithm for Path Planning - Kella, Vamsee Krishna;

OpenCV – Viola-Jones algorithm for Object Detection - Bandirala, Shivani; Koduri, Aswayuja

Performance evaluation - Kella, Vamsee Krishna; Bandirala, Shivani; Koduri, Aswayuja

Section V (Hardware Developments) - Kella, Vamsee Krishna; Bandirala, Shivani; Koduri, Aswayuja

VIII. References

[1] F. Ingrand and M. Ghallab, "Deliberation for autonomous robots: A survey," Artificial Intelligence, vol. 247, pp. 10 – 44, 2017, special Issue on AI and Robotics.

[2] Z. Wang and M. Schwager, "Multi-robot manipulation with no communication using only local measurements," 2015 54th IEEE Conference on Decision and Control (CDC).

[3] L.S. Tessema, R. Jaeger, U. Stilla, "A Mathematical Sensor Model for Indoor use of a Multi-beam Rotating 3d LiDAR," Photogrammetric Image Analysis & Munich Remote Sensing Symposium, 18–20 September 2019, Munich, Germany.

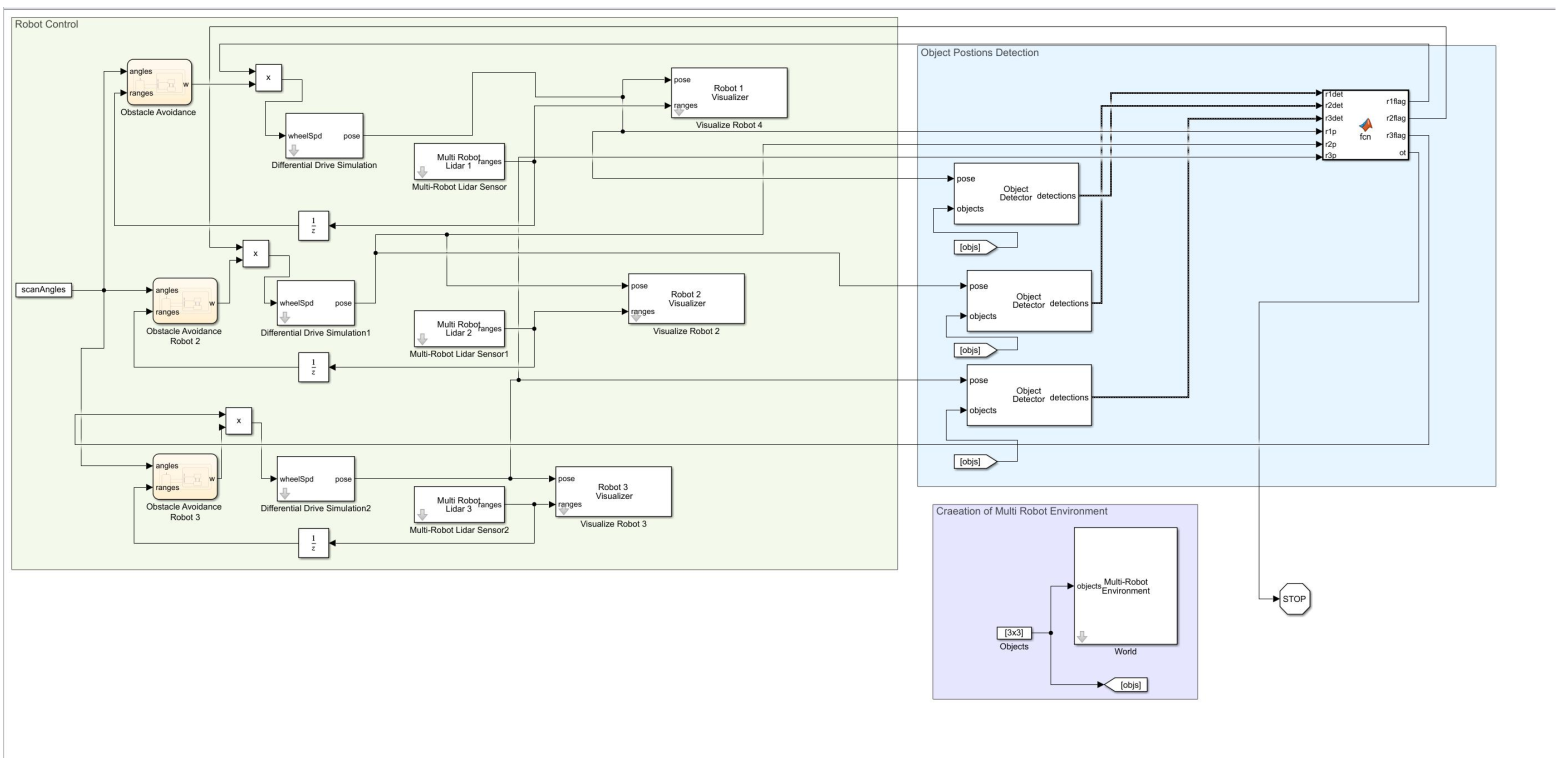
[4] Luiz Chaimowicz, Nathan Michael, and Vijay Kumar, "Controlling swarms of robots using interpolated implicit functions," Proceedings - IEEE International Conference on Robotics and Automation, 2005(April):2487–2492, 2005.

[5] Dr. Fang Tang, "Multi-robot path planning", Cal Poly Pamona University.

[6] Tim Chin, "Robotic Path Planning: RRT and RRT* Exploring the optimized version of a orthodox path planning algorithm," https://theclassytim.medium.com/robotic-path-planning-rrt-and-rrt-212319121378

[7] Jorge Pena Queralta, Jussi Taipalmaa, Bilge Can Pullinen, Victor Kathan Sarker, Tuan Nguyen Gia, Hannu Tenhunen, Moncef Gabbouj, Jenni Raitoharju, Tomi Westerlund, "Collaborative Multi-Robot Systems for Search and Rescue: Coordination and Perception," 2020.

[8] A. Marjovi, J. G. Nunes, L. Marques and A. de Almeida, "Multi-robot exploration and fire searching," *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1929-1934, doi: 10.1109/IROS.2009.5354598.


Robot Control
Object Postions Detection
Craeation of Multi Robot Environment
angles
ranges
w
Obstacle Avoidance
x
wheelSpd
pose
Differential Drive Simulation
Multi Robot Lidar 1
ranges
Multi-Robot Lidar Sensor
pose
ranges
Robot 1 Visualizer
Visualize Robot 4
scanAngles
Obstacle Avoidance Robot 2
Differential Drive Simulation1
Multi Robot Lidar 2
Multi-Robot Lidar Sensor1
Robot 2 Visualizer
Visualize Robot 2
Obstacle Avoidance Robot 3
Differential Drive Simulation2
Multi Robot Lidar 3
Multi-Robot Lidar Sensor2
Robot 3 Visualizer
Visualize Robot 3
objects
Object Detector
detections
[objs]
r1det
r2det
r3det
r1p
r2p
r3p
fcn
r1flag
r2flag
r3flag
ot
STOP
[3x3]
Objects
Multi-Robot Environment
World

Editor - C:\Users\vamse\OneDrive\Desktop\598 Project\RRT.m

RRT.m

```matlab
%Code to Find sthe Optimal Path Using RRT
ss = stateSpaceSE2;
sv = validatorOccupancyMap(ss);
sv.Map = map1;
sv.ValidationDistance = 0.5;
ss.StateBounds = [map1.XWorldLimits; map1.YWorldLimits; [-pi pi]];
planner = plannerRRTStar(ss,sv);
planner.ContinueAfterGoalReached = true;
planner.MaxIterations = 3000;
planner.MaxConnectionDistance = 2.5;
start = [3,3, 0];
for i = 1:3
goal = [poseMat(i,1),poseMat(i,2), 0]; %Getting coordinates from robot
rng(100, 'twister')
[pthObj, solnInfo] = plan(planner,start,goal);
map1.show;
hold on;
scatter(start(1),start(2),'filled')
scatter(goal(1),goal(2),'filled')
plot(pthObj.States(:,1),pthObj.States(:,2),'b-','LineWidth',2); % draw path
title('Optimal Route to Obejct'+ i)
legend('Start Point','Object Postion','Path')
end
```