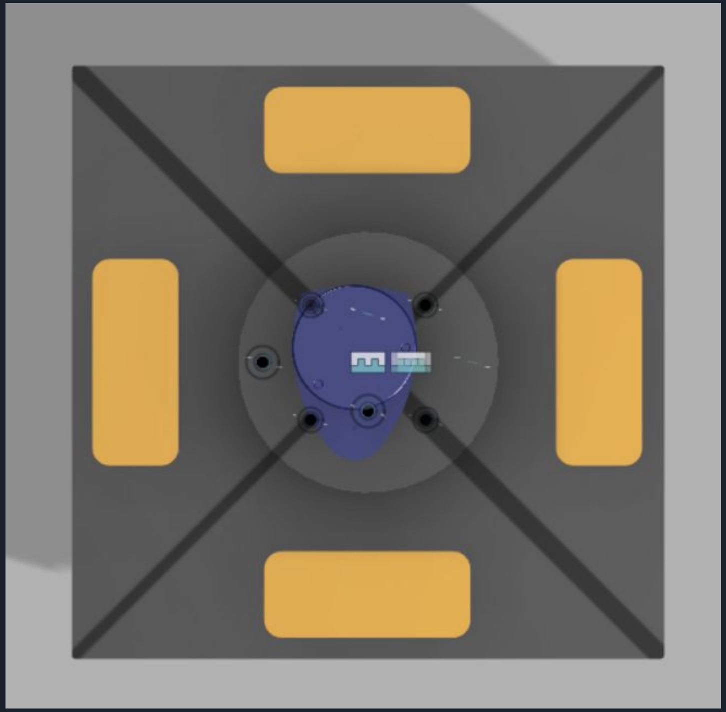
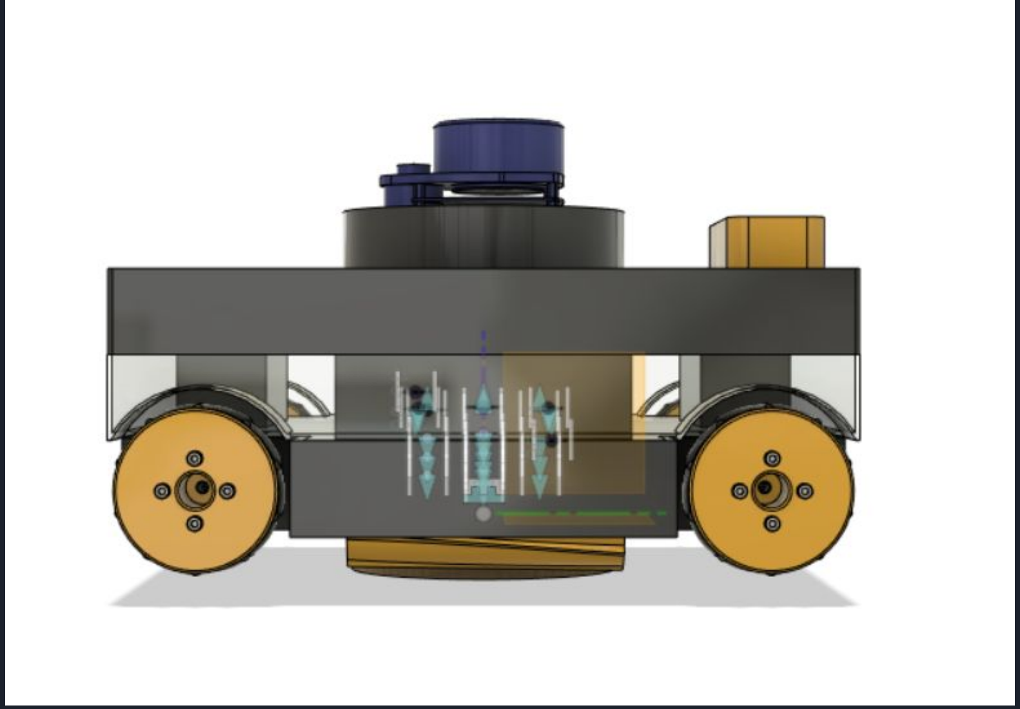
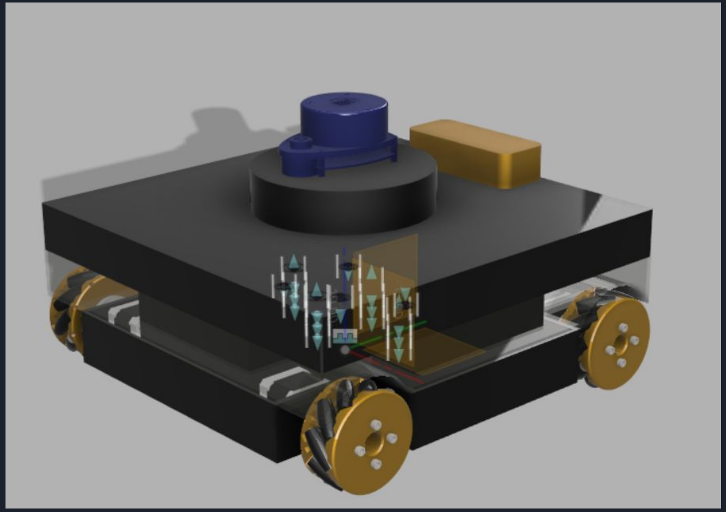
Area coverage by Swarm Robotics

By:- WALL-E

Robot Design

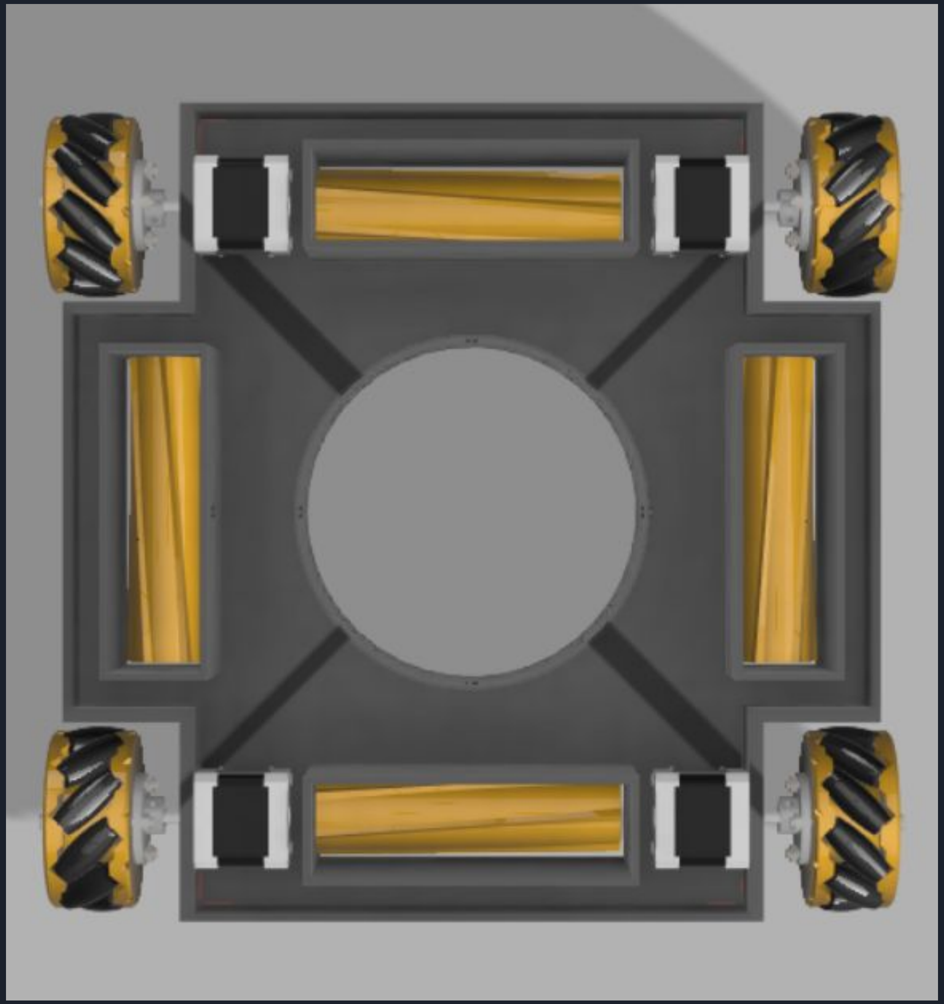
The robot is a 4 wheel planar drive robot with mecanum wheels. The main body of the robot is divided into three compartments. The lower most compartment houses the electronic components. The middle compartment is filled with the cleaning liquid and the upper compartment consists of the dry cleaning system. A Lidar is placed on the top to aid in obstacle avoidance and locomotion.


Lower Compartment

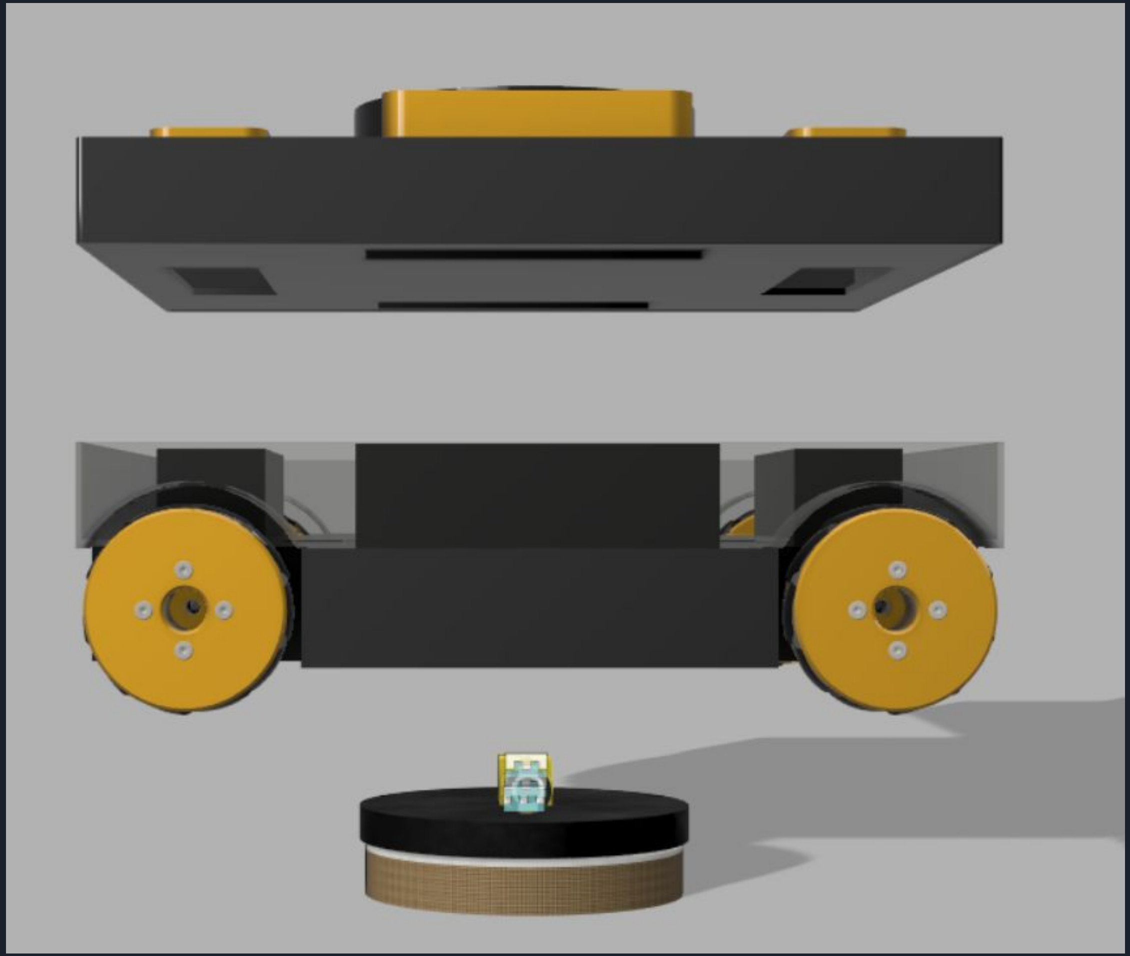
The lower component contains most of the electric components:

1) Rechargeable battery
2) A4988 stepper motor driver
3) Stepper Motor Driver Controller Board for Arduino
4) Arduino UNO as control unit
5) 4 ultrasonic sensors are attached to the bottom to detect the type of surface the robot is cleaning.



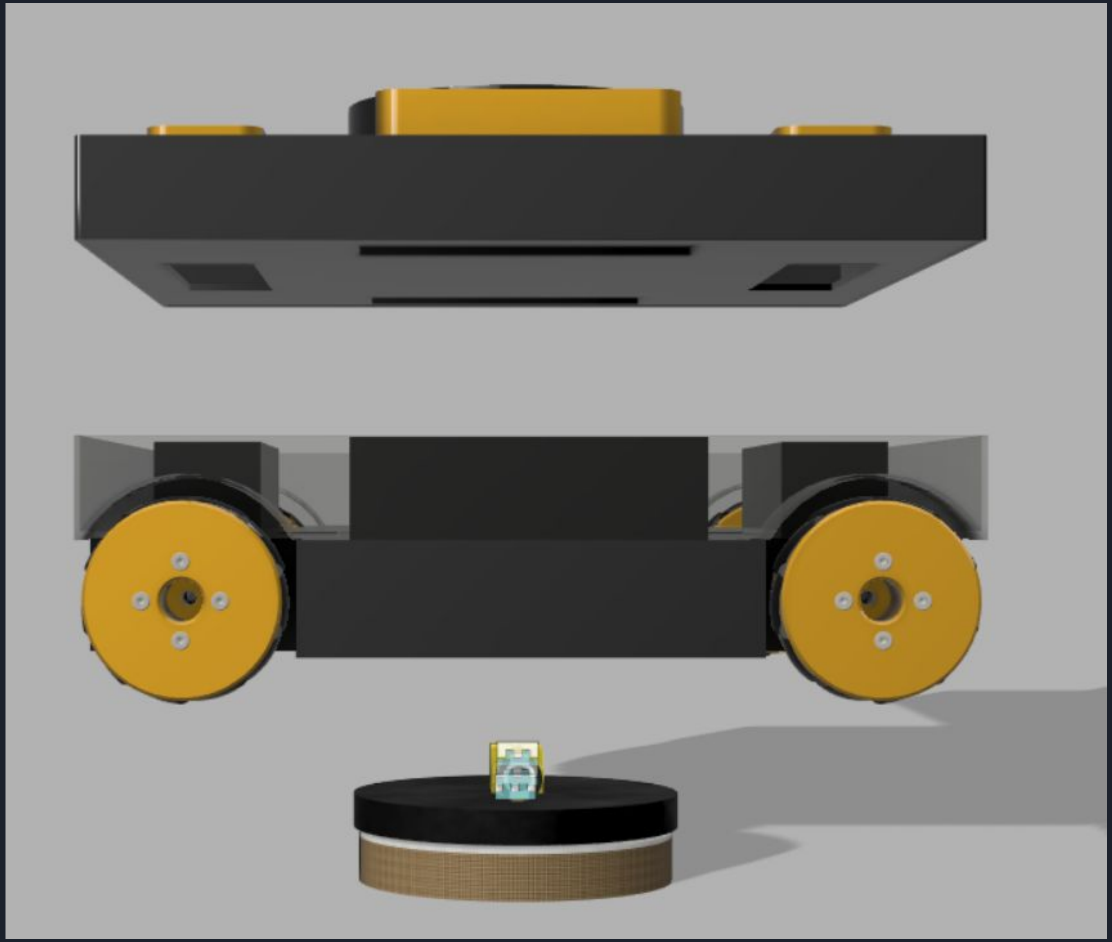
Middle Compartment

1. Middle Component
2. The middle component consists of the
3. Pneumatic system: This system controls the movement of the rotating sponge. The working is analogous to the working of a syringe. Some surfaces may not be suitable for wet cleaning (Eg carpets). The ultrasonic sensor on detecting the surface sends an input to the pneumatic system via the Arduino which pulls up the sponge.
4. Dispenser system: At regular intervals of time the dispenser dispenses the cleaning liquid
5. Cleaning Liquid



Upper Compartment

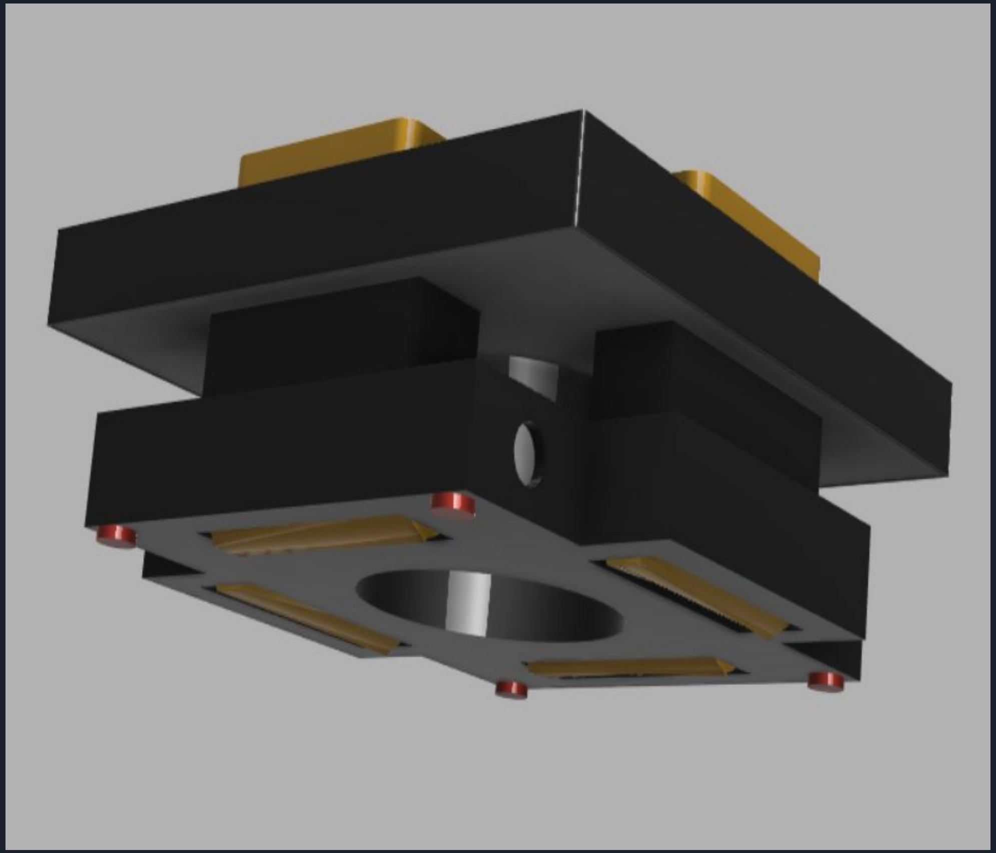
The upper component houses the vacuum cleaning motor.The dry waste is collected in this component. The rectangular slits have one way valves attached as well to prevent the dry waste going back. The upper component also contains the air compressor for the pneumatic system.



Cleaning Mechanism

Dry Cleaning Mechanism

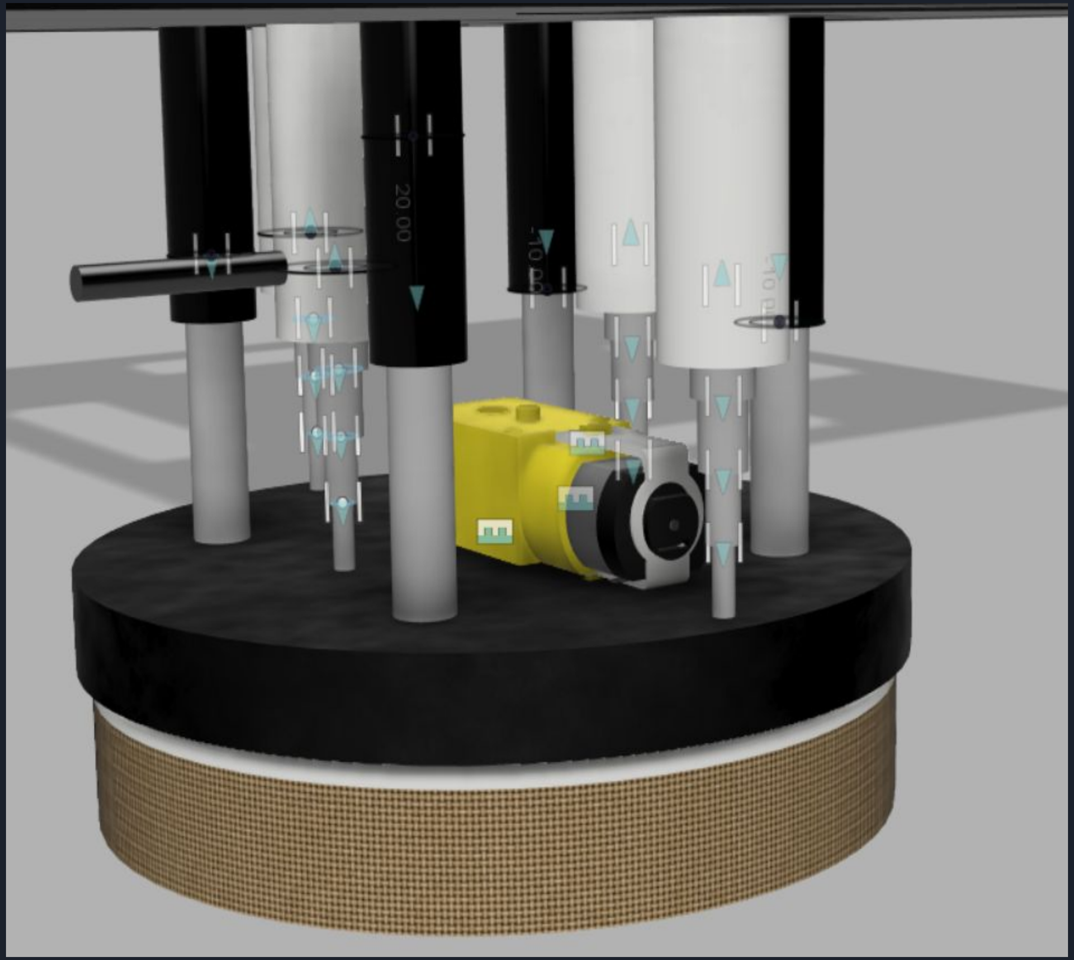
The dry cleaning is done by 4 vacuum dry-cleaning slits near each edge of the bot. The 4 rubber rollers pick up the dry waste and the vacuum cleaner motor provides the necessary pressure difference and power to pick up the dry waste. The placement of the slits also allows simultaneous drying of the surface after the wet-cleaning.



Cleaning Mechanism

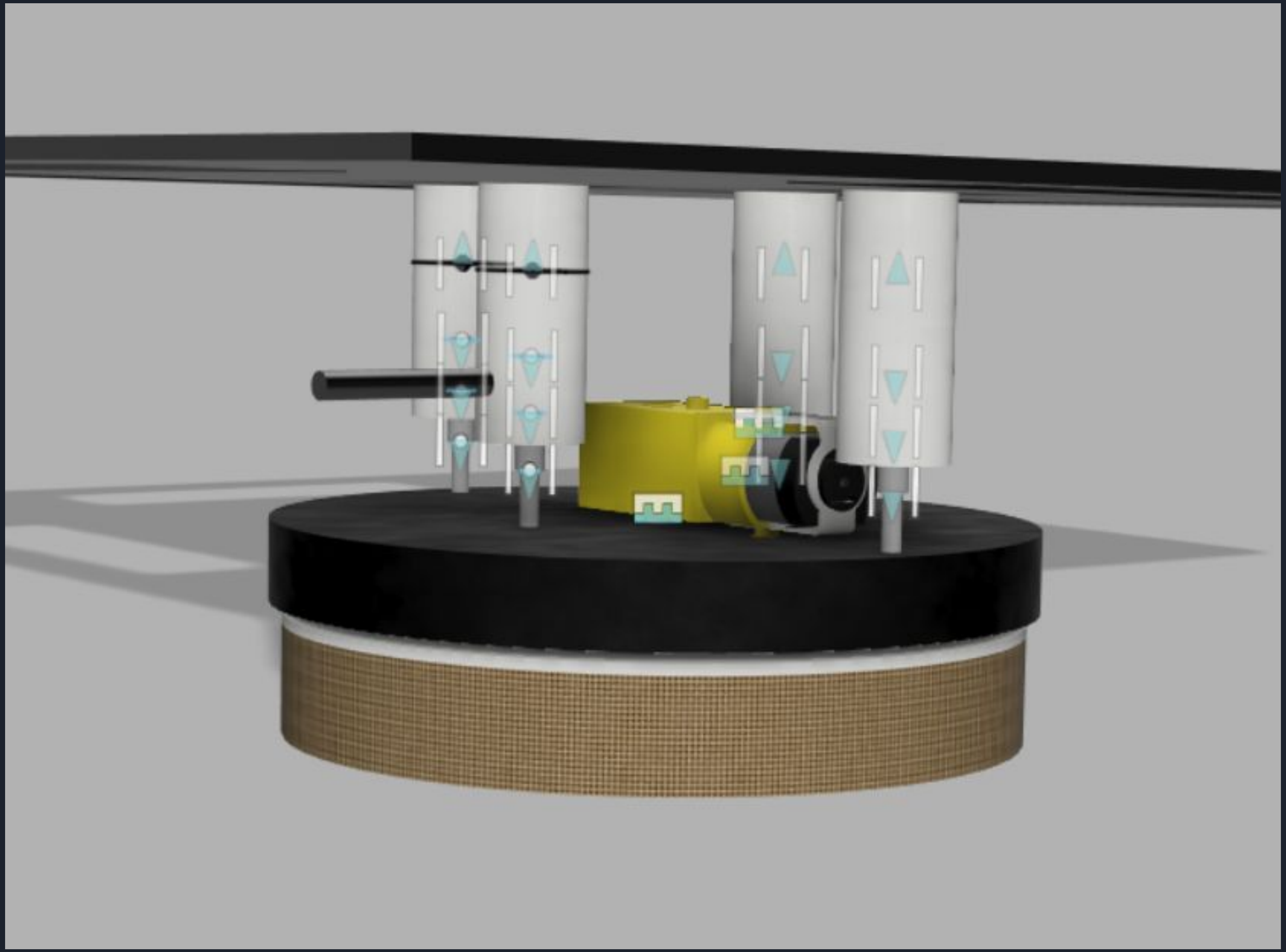
Wet Cleaning Mechanism

The wet cleaning mechanism consists of a rotating sponge attached to a plastic body which is connected to a motor. This motor sitting on the rubber body rotates the sponge. The pneumatic design allows the rubber to move or down depending on if the surface requires wet cleaning. Another advantage of the pneumatic design is it allows cleaning on uneven surfaces as the pneumatic pipes in the chamber allows it to act analogous to a spring. Pipes are attached from the liquid compartment to the sponge. Liquid is dispensed at regular intervals.



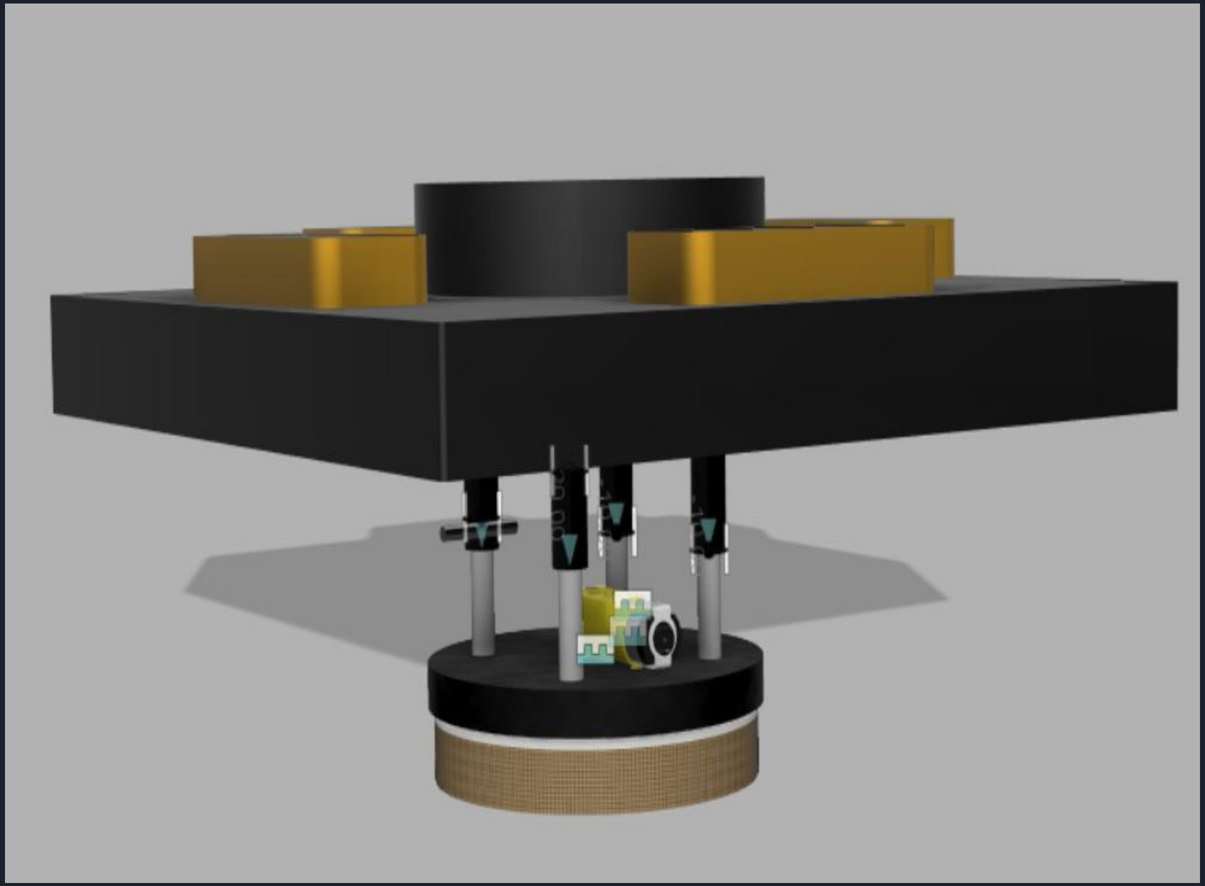
Dripping Mechanism

The cleaning liquid from the middle chamber enters the drippers from pipes at regular intervals using a dispenser system.



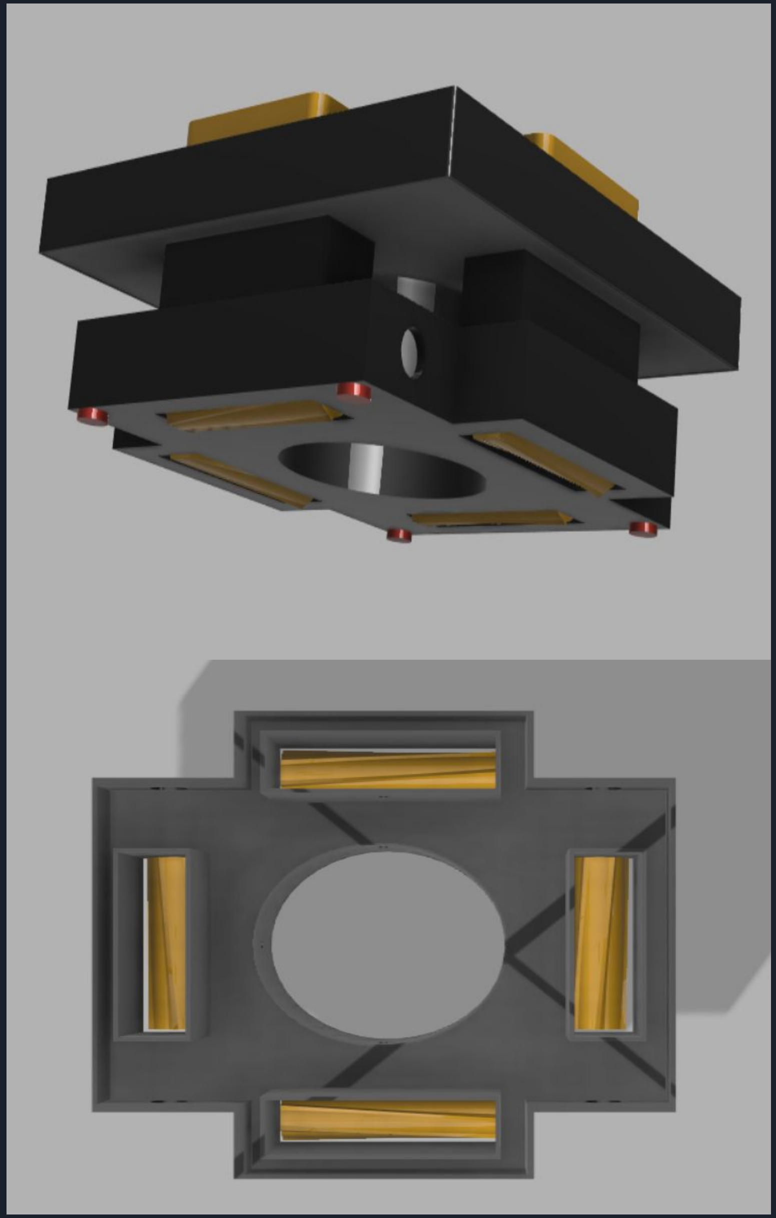
Pneumatic System

The pneumatic system consists of an air compressor present at the top of the bot along with the pneumatic cylinders. This system allows us to move the sponge up and down and also allows the sponge to effectively clean uneven surfaces.



Vacuum System

The vacuum system consists of a vacuum motor pump in the upper compartment along with the 4 vacuum dry-cleaning slits fitted with rubber rollers to collect the dry waste.



Additional Features

Docking System

This is a unique feature provided along with our cleaning bot. The docking system allows the robot to autonomously locate the docking system and dock itself when the resources are depleted.

1) Here the batteries are recharged.
2) The cleaning liquid is reloaded.
3) The dry waste is vacuumed out and stored into bags.
4) The sponge will also be thoroughly cleaned.

Additional Features

User Interface & Scheduling

The cleaning bot will also come with an application to see the cleaning process in real time. The application will also provide the feature to schedule the cleaning process and the frequency of cleaning.

Materials & Manufacturing

Materials

1. The Body frame is made out of Aluminium 6061.
2. The liquid compartment will be made out of ABS Plastic which will allow the cleaning liquid level to be visible at all times.
3. The drippers will be made out of stainless steel.
4. The wet cleaning will be done using melamine sponge.

Manufacturing

For manufacturing the bot , we will be using processes such as hammering, sheet metal bending and welding.

SERVICEABILITY

1. Regular software updates will be provided to constantly improve autonomous navigation algorithm.
2. All the electronic components of our bots are easily available online (links provided in cost mapping). In case of any kind of malfunctioning of the robot, respective electronic component can be replaced.
3. Additional components will be provided along with the bot for easy replacement in case of any defects or damages.

Additional components:

- Rubber Rollers
- Beads of the mecanum wheels
- Sponges

COST ESTIMATE ~ ₹18,300.00

S.No	TITLE	QUANTITY	COST PER PIECE (in Rs.)	NET AMOUNT (in Rs.)	PURCHASE LINK
1	Lidar	x1	₹ 6,499.00	₹6,499.00	Lidar
2	Air compressor	x1	₹ 319.00	₹ 319.00	Air compressor
3	Vacuum motor	x4	₹ 400.00	₹ 1,600.00	Vacuum motor
4	Pneumatic cylinder	x4	₹ 286.00	₹ 1,144.00	pneumatics
5.	Arduino uno	x1	₹ 399.00	₹ 399.00	Arduino
6.	Stepper motor	x4	₹ 575.00	₹ 2,300.00	Stepper
7.	Motor driver	x4	₹ 74.00	₹ 296.00	Motor driver
8.	Ultrasonic sensors	x4	₹ 62.00	₹ 248.00	Ultrasonic sensors
9.	Bo motor	x1	₹ 55.00	₹ 55.00	Bo motor
10.	Light motor	x2	₹ 99.00	₹ 198.00	Light motor
11.	Mecanum wheels	x4	₹ 2,933.00	₹ 2,933.00	Mecanum wheels
12.	Rechargeable Battery	x1	₹ 1,689.00	₹ 1,689.00	Rechargeable Battery
13.	Aluminium + Miscellaneous	X1 sheet (1.5kg)	₹ 300.00	₹ 300.00	Aluminium

FAILURE CASES HANDLING

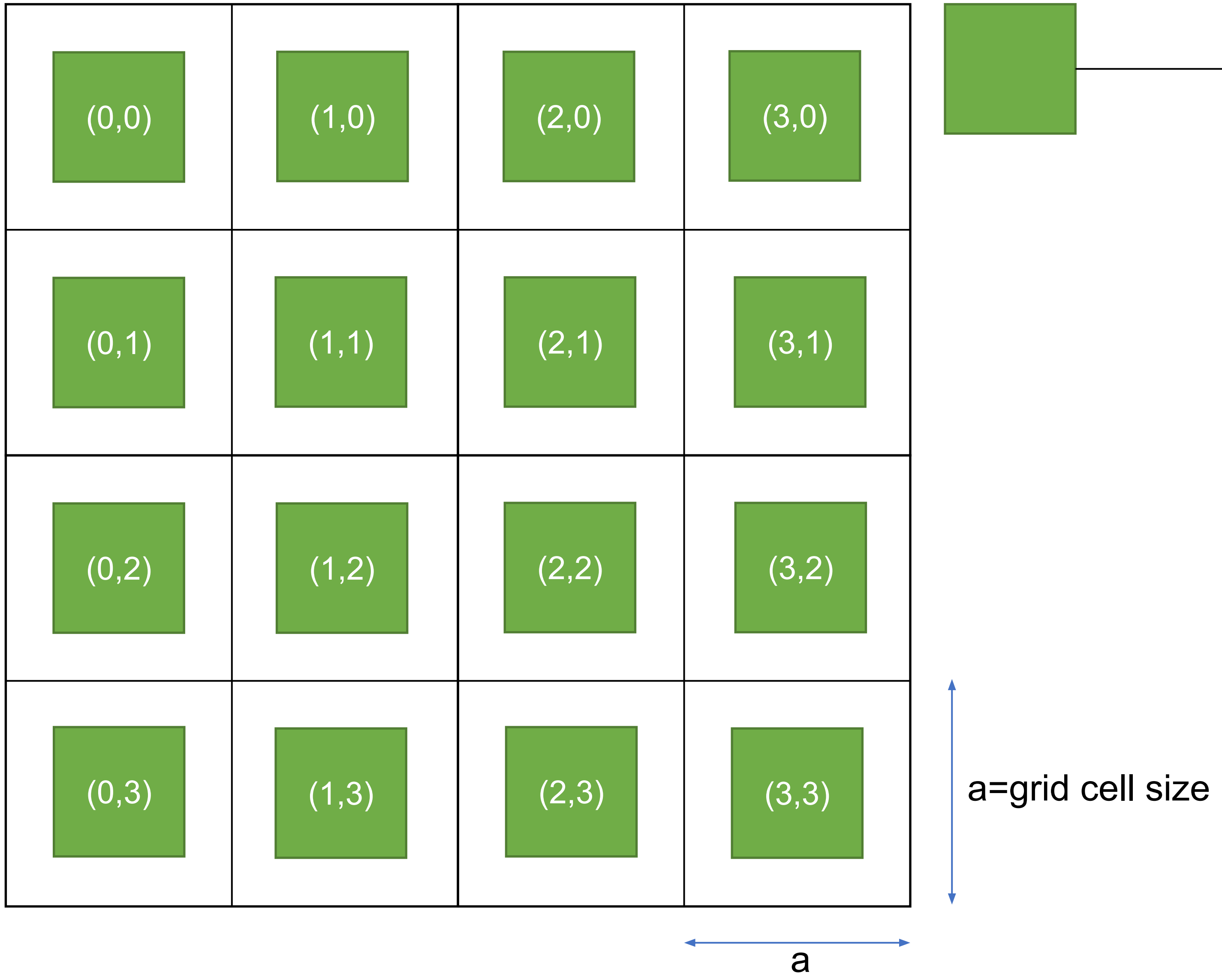
1. All the four rubber rollers are removable. They are cylindrical in shape which avoid any hair strands getting entangled with the roller.
   Rollers are easy to replace and can be done without the help of any professionals
2. The mecanum wheels are constructed from many small beads which can be replaced after they are worn out.
3. All the electrical components are safely protected inside strong and insulated casings.
4. Our robot is provided with ultrasonic sensors at the bottom to prevent falls.
5. In case of any issue with rotation of rollers, the bot has in built instructions to go back to the docking station where the rollers are replaced automatically.

Advantages over other traditional cleaning bots

This robot has certain advantages over other traditional cleaning robots.

1) The 4 mecanum wheels allow the bot to move independent of its orientation.
2) The 4 vacuum dry-cleaning slits adds an extra layer of symmetricity to the bot and also allows faster drying.
3) The application provides a real time interface to check the progress and to schedule the cleaning.
4) The Docking system allows the bot to be truly autonomous by allowing hands free recharging, reloading and storage of dry waste.

Area Division of the Map For decision making

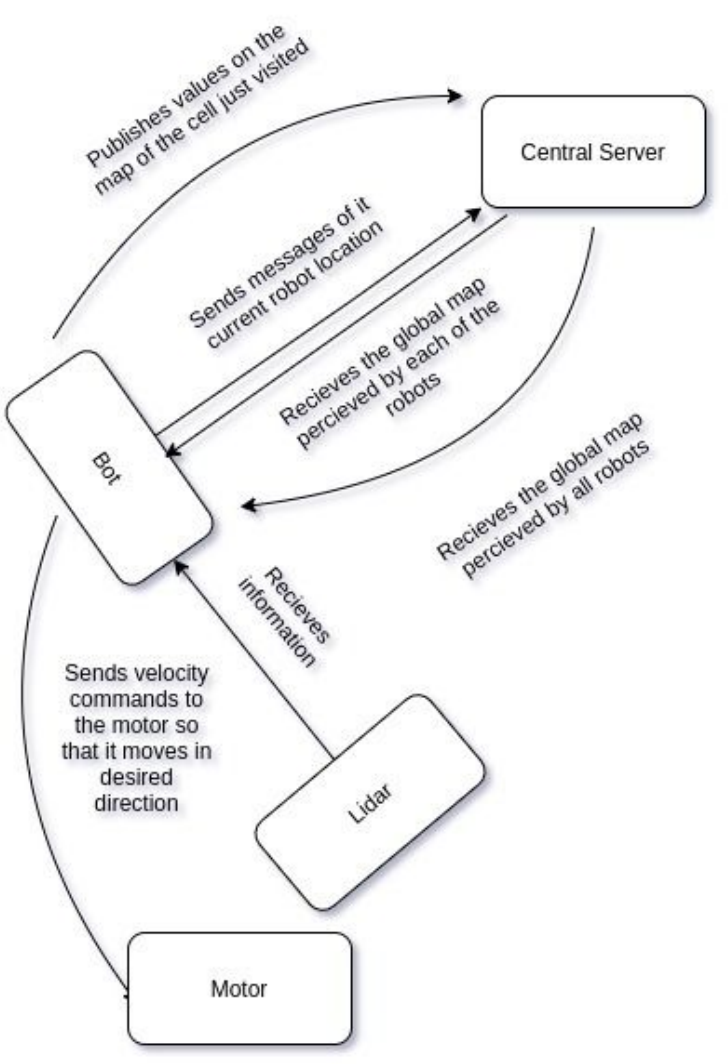


- We divide the map into 1*1 cells that when the bot visits it will clean.
- A threshold region is defined so that whenever
  it comes in that it is allowed to take a decision.
- The map is stored as a 2D array whose cells have the attribute of pheromone.
- The cell size is dependent on the robot size and the lidar range taken and can be changed accordingly.

Communication Regimes between the agents and the server.

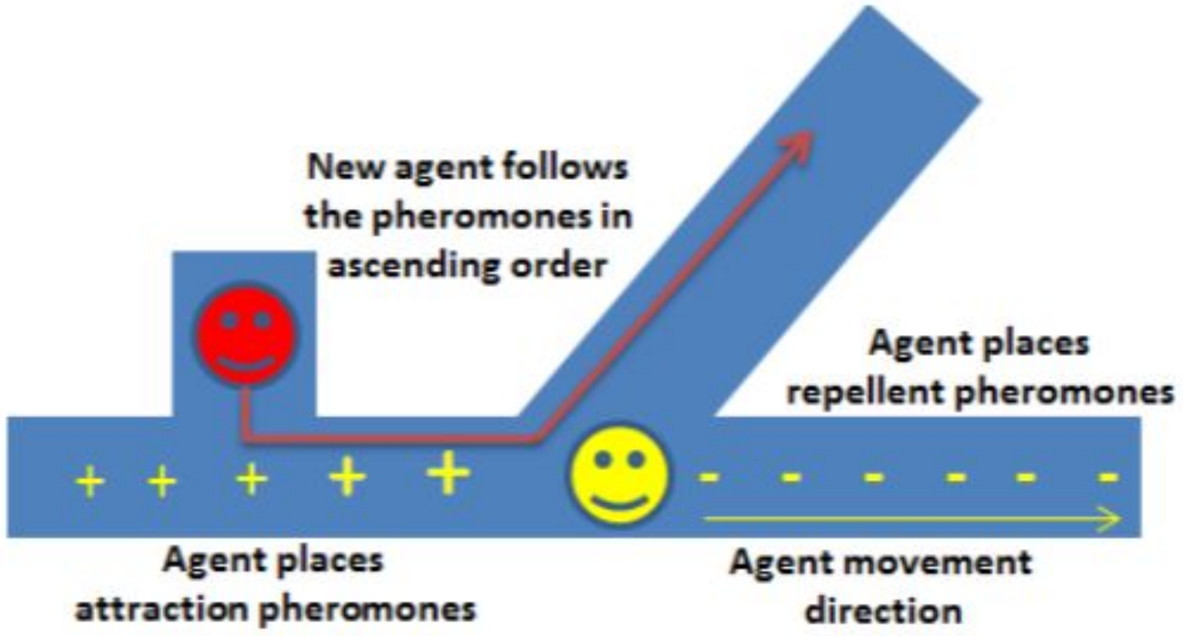
- There are two ways for linking the communication between the robots are local and global depending on the communicating range between the robots.
- In the central server method, every iteration of the algorithm the robots communicates its visited cells and its local map with the central server and the robots also receive information from the central server on what are the locations of other agents.
- The local method involves using a Map merging algorithm every-time it is in communicating range of remaining agents. Here the agent transmits velocity and location.(Here the rejection vector will be calculated by extrapolating the last seen location + last seen velocity*time).
- Other than this the sensors present on the bot such as LiDAR, Localization GPS module help in gaining information from environment.

•

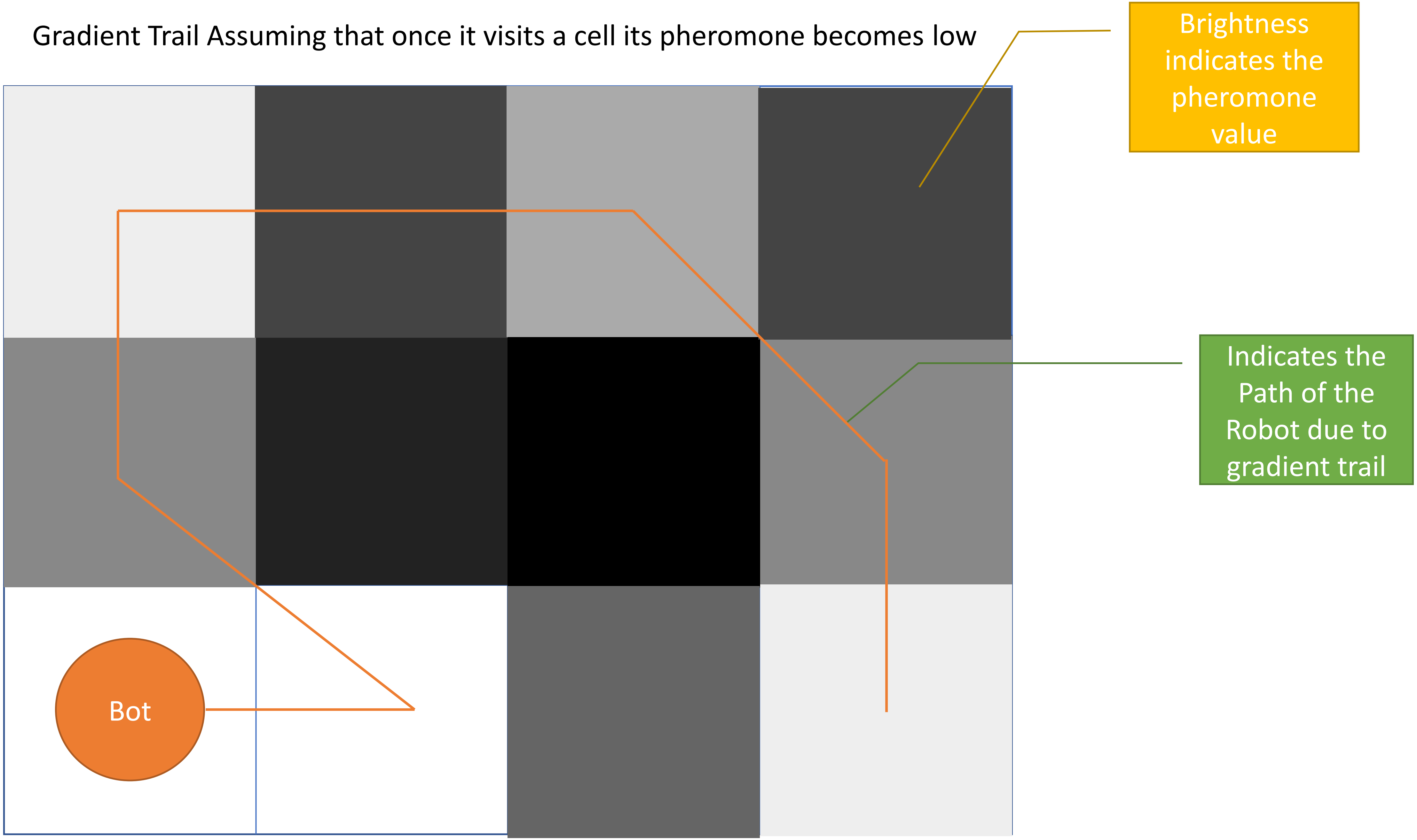


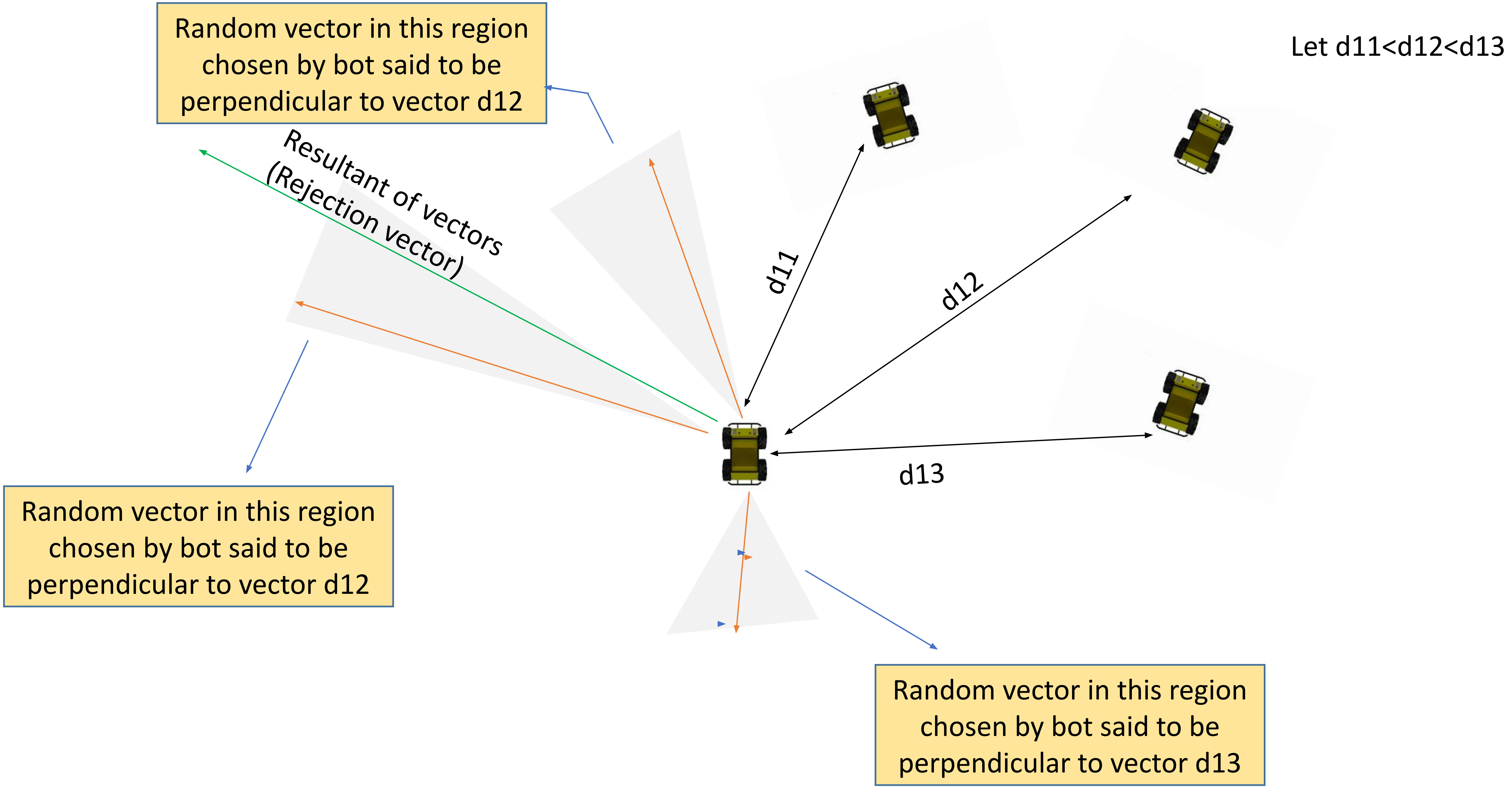
Use of pheromones for Navigation

- Pheromone(in the context of this algorithm) are values given to the grid cells when a agent visits a cell or when it detects an obstacle.
- Pheromone values are given based on the number of exit cell locations available to the bot and whether it has visited the cell previously or not.
- We use repelling pheromones when the agent has no other option to visit except the exit location which it has already visited.
- For example when there are more than one exit location possible we give positive pheromone value(which is equal to number of valid unexplored exit locations).



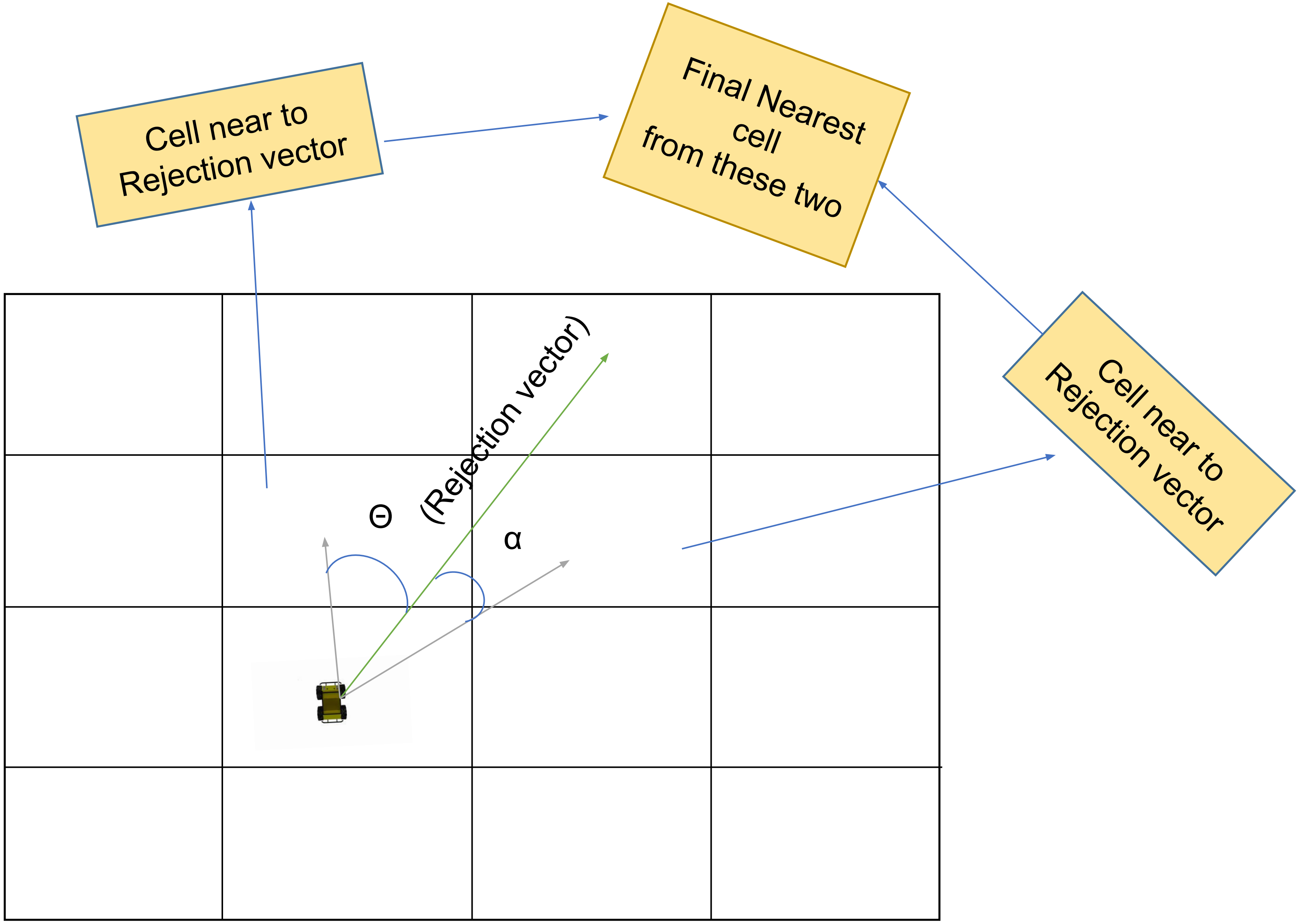
Gradient Trail Assuming that once it visits a cell its pheromone becomes low



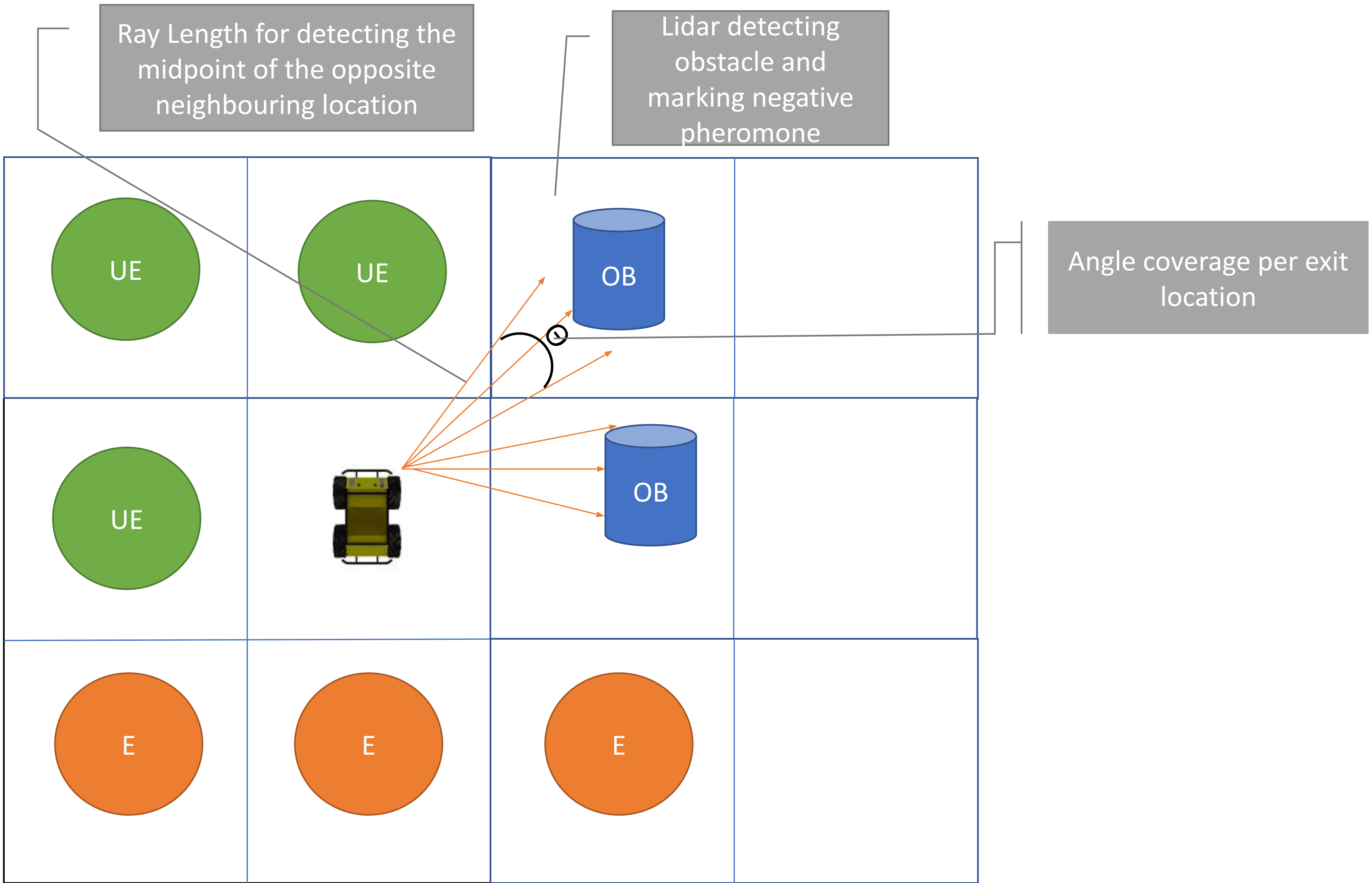


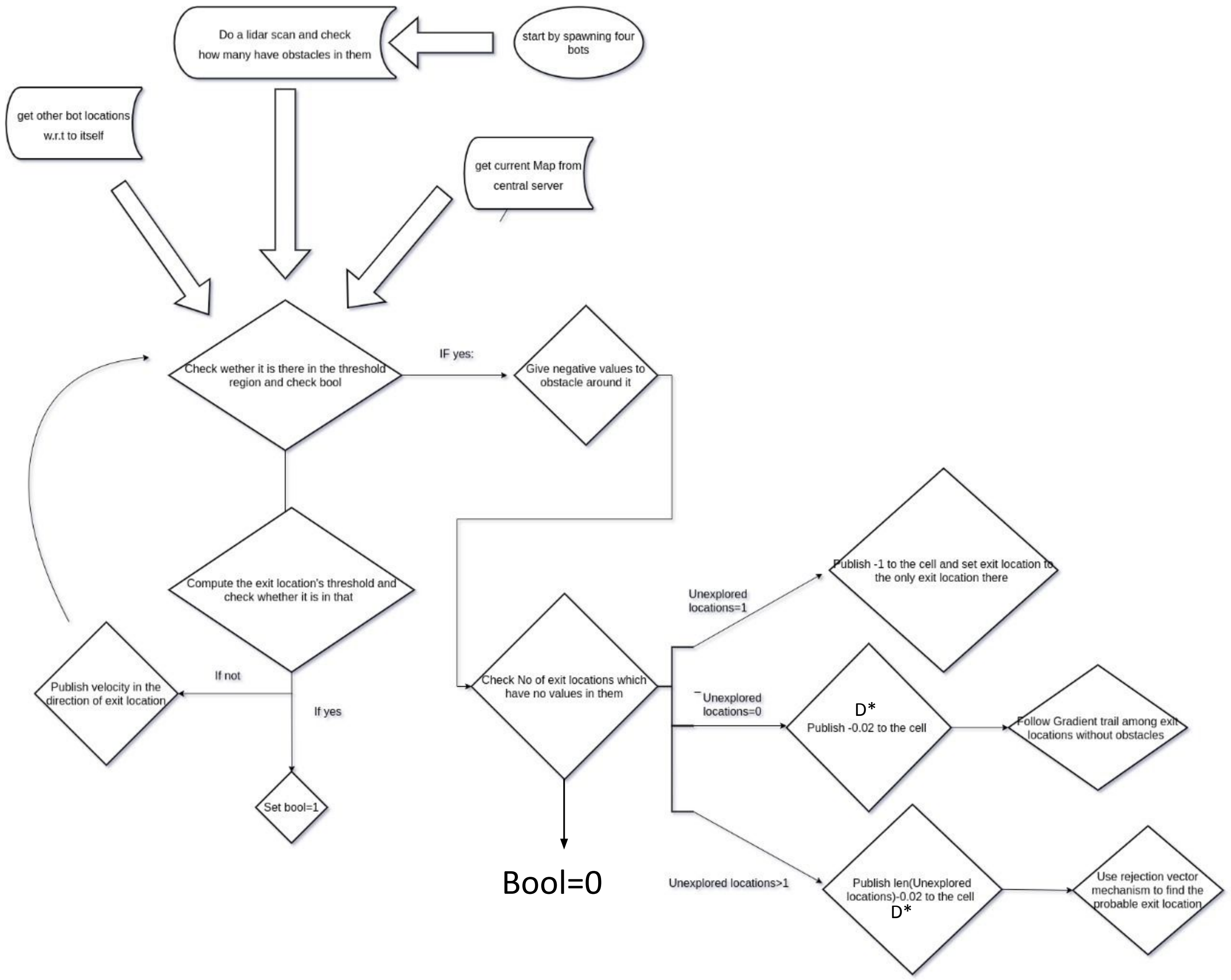
Random vector in this region chosen by bot said to be perpendicular to vector d12
Let d11<d12<d13
Resultant of vectors (Rejection vector)
d11
d12
d13
Random vector in this region chosen by bot said to be perpendicular to vector d12
Random vector in this region chosen by bot said to be perpendicular to vector d13

Finding closest exit cell based on Rejection Vector

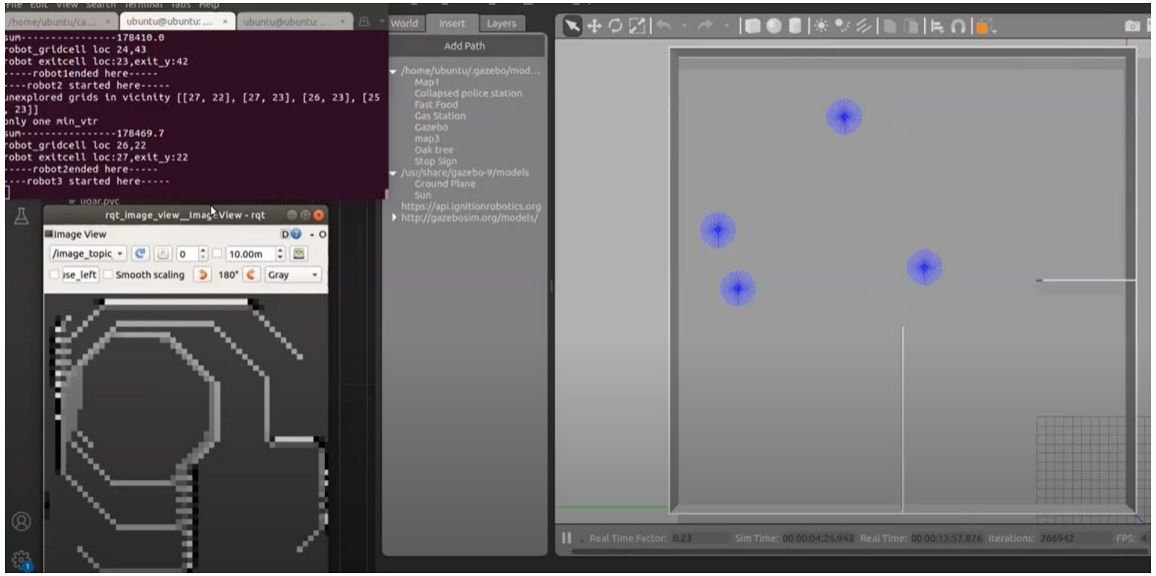


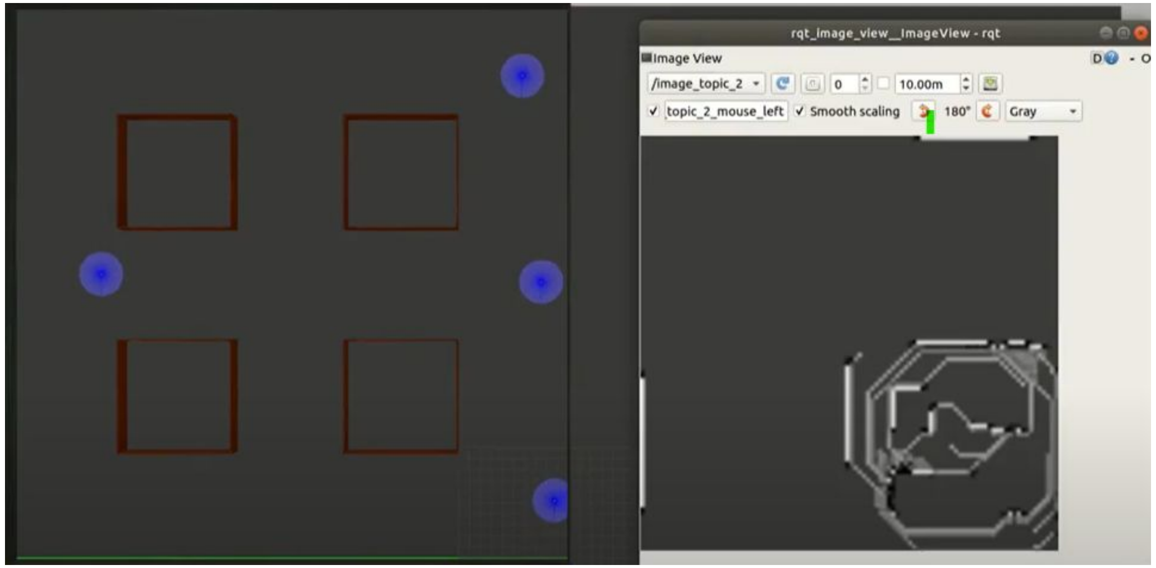
If Θ-α lies between specified threshold then the priority will be given to maximum unexplored region(checks the consecutive 10 cells for unexplored area)and thereby decides the exit location


Ray Length for detecting the midpoint of the opposite neighbouring location
Lidar detecting obstacle and marking negative pheromone
Angle coverage per exit location
UE
UE
OB
UE
OB
E
E
E


Do a lidar scan and check how many have obstacles in them
start by spawning four bots
get other bot locations w.r.t to itself
get current Map from central server
Check wether it is there in the threshold region and check bool
IF yes:
Give negative values to obstacle around it
Compute the exit location's threshold and check whether it is in that
If not
Publish velocity in the direction of exit location
If yes
Set bool=1
Check No of exit locations which have no values in them
Bool=0
Unexplored locations=1
Publish -1 to the cell and set exit location to the only exit location there
Unexplored locations=0
D*
Publish -0.02 to the cell
Follow Gradient trail among exit locations without obstacles
Unexplored locations>1
Publish len(Unexplored locations)-0.02 to the cell
D*
Use rejection vector mechanism to find the probable exit location

Simulation


World
Insert
Layers
Add Path
rqt_image_view__ImageView - rqt
Image View
/image_topic
10.00m
Smooth scaling
180°
Gray


rqt_image_view__ImageView - rqt
Image View
/image_topic_2
10.00m
topic_2_mouse_left
Smooth scaling
180°
Gray

Challenges and Debugging

Overshooting[means that the robot moved ahead of its threshold region of its exit location which caused it to oscillate] this was resolved by using threading and running 4 ROS nodes on different cores. Other parameters affecting overshooting is velocity limit(0.8ms) and grid dimensions.

Video overshooting :

https://drive.google.com/file/d/1I91Kvov-E0zWyfKwmXKIKOZBdNwDqag9/view?usp=sharing

We plan on creating a algorithm that works in limited communication range by using a map merging algorithm (this has a O(n^2) complexity, but it only occurs when it is in communication range) while this approach will occasionally make the robot visit the same cell again it reduces the computational operations done.

There can be instances where at least two robots moves towards the same grid cell ,so to avoid collision we have used an impulsive force opposite in the direction between the line joining the two robots when a certain threshold distance is reached, and the decision making is suspended temporarily.

While we were implementing our Algorithm we observed there were patches of unexplored area which were not accessible due to the surrounding explored area hence we have created a decay principle so that this error is resolved. We have chosen the decay parameters based on the velocity limit ,map size.

Initially we decided on using velocity based on Coulomb's law only then we added weightage to the direction it was previously moving in this reduced time in decision making and allowed for faster area coverage.