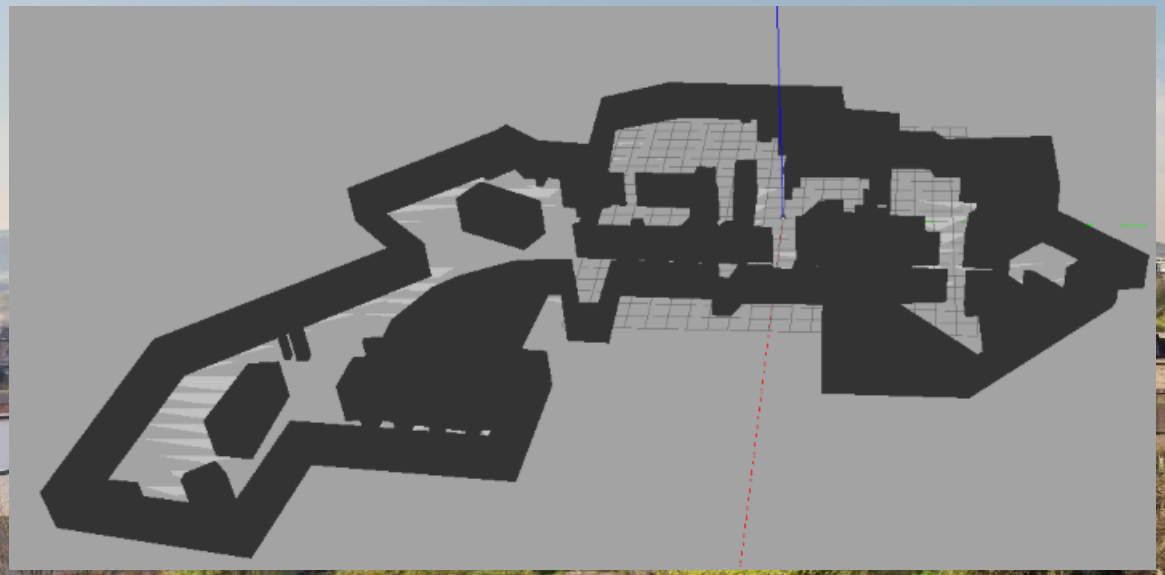
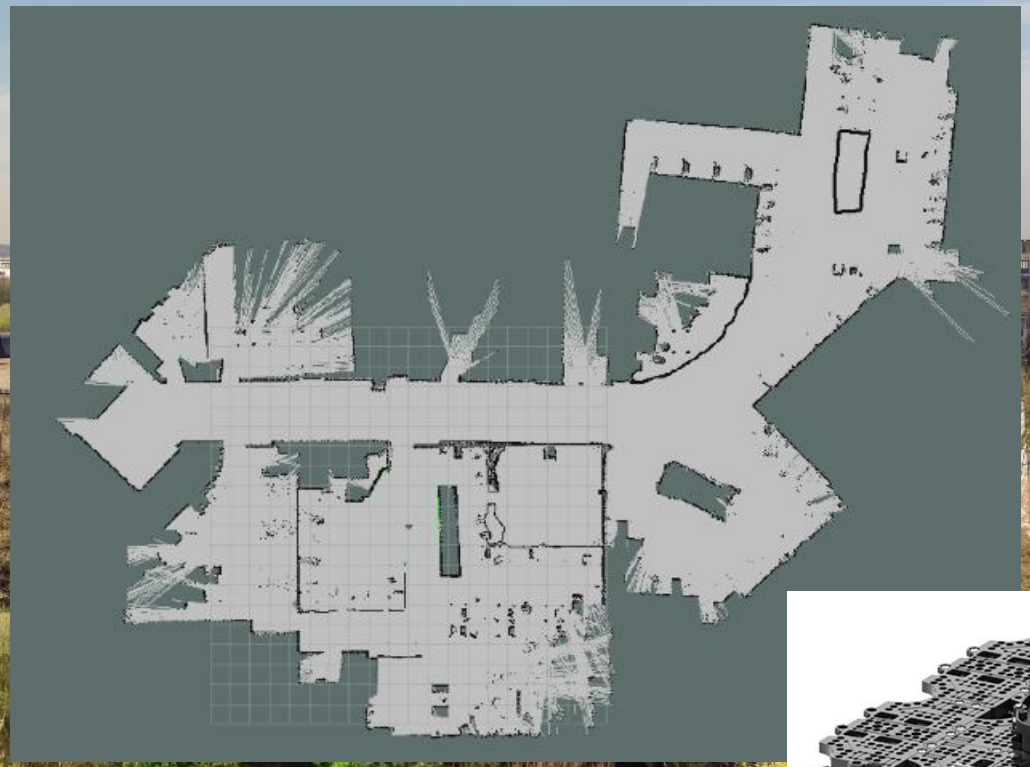
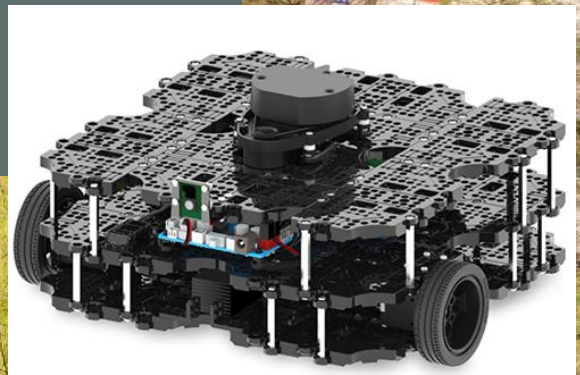
dortmund university

Mobile Robots Mandatory Assignment (SoSe 2024)







Prepared by: Ding Ken Chuah

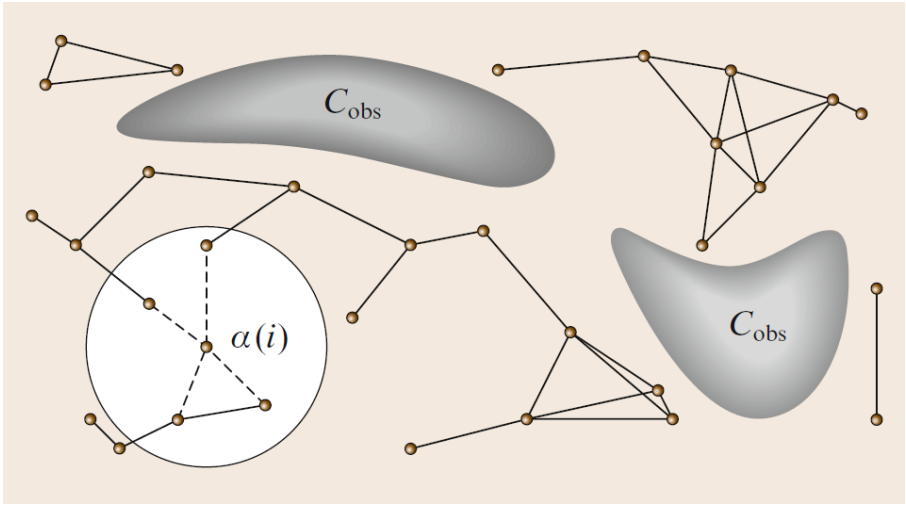
Objectives

- To navigate the robot from a starting point to the desired goal pose through a collision-free trajectory while adhering to the constraints.
- To investigate different parameter tunings in the resultant trajectories in both PRM and RRT global path planning methods.

Global Path Planning

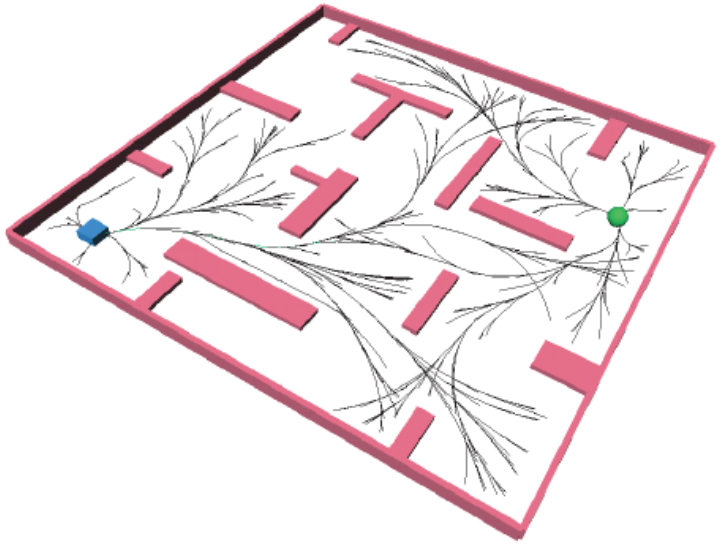
- **Probabilistic Roadmap Planner (PRM)**

➢Multi-Query Sampling Based Planner



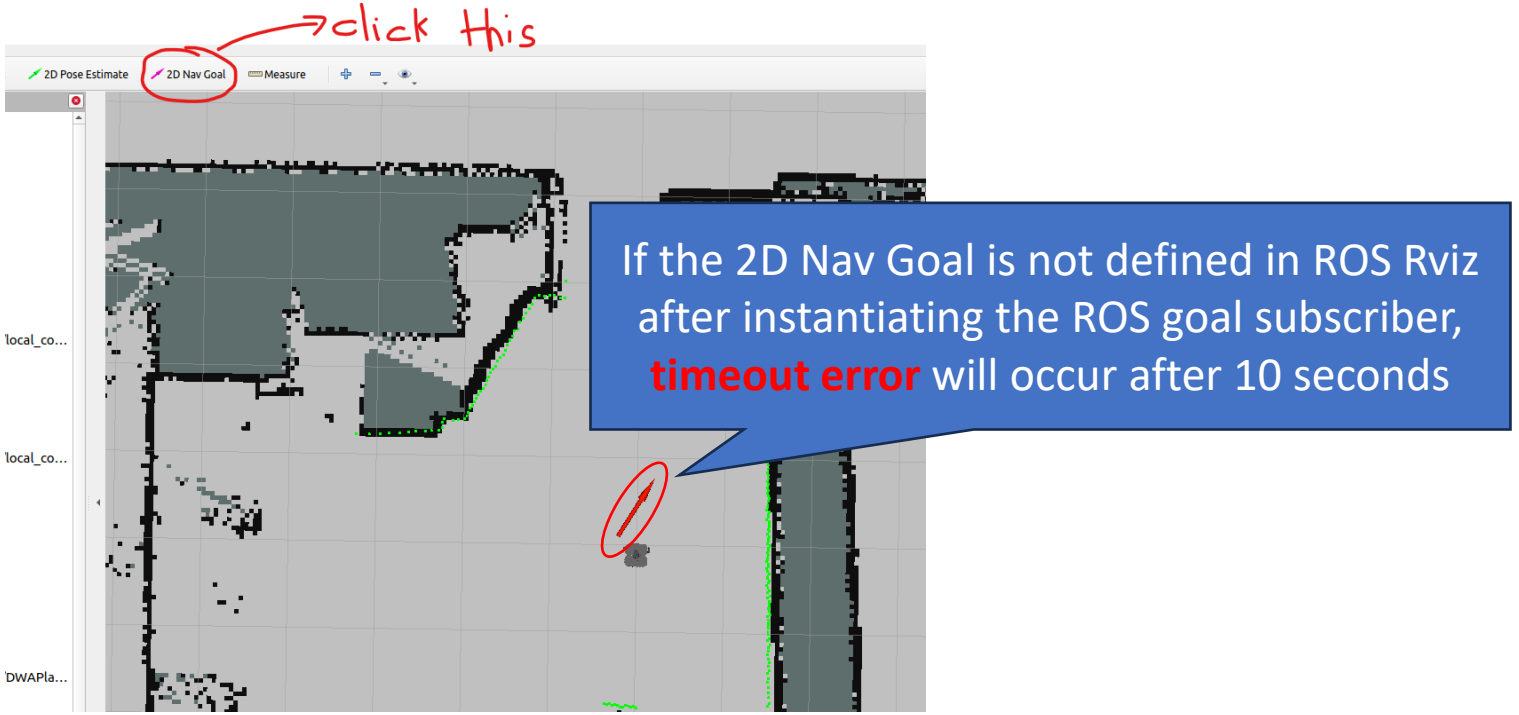
- **Rapidly Exploring Random Tree (RRT)**

➢Single-Query Sampling Based Planner

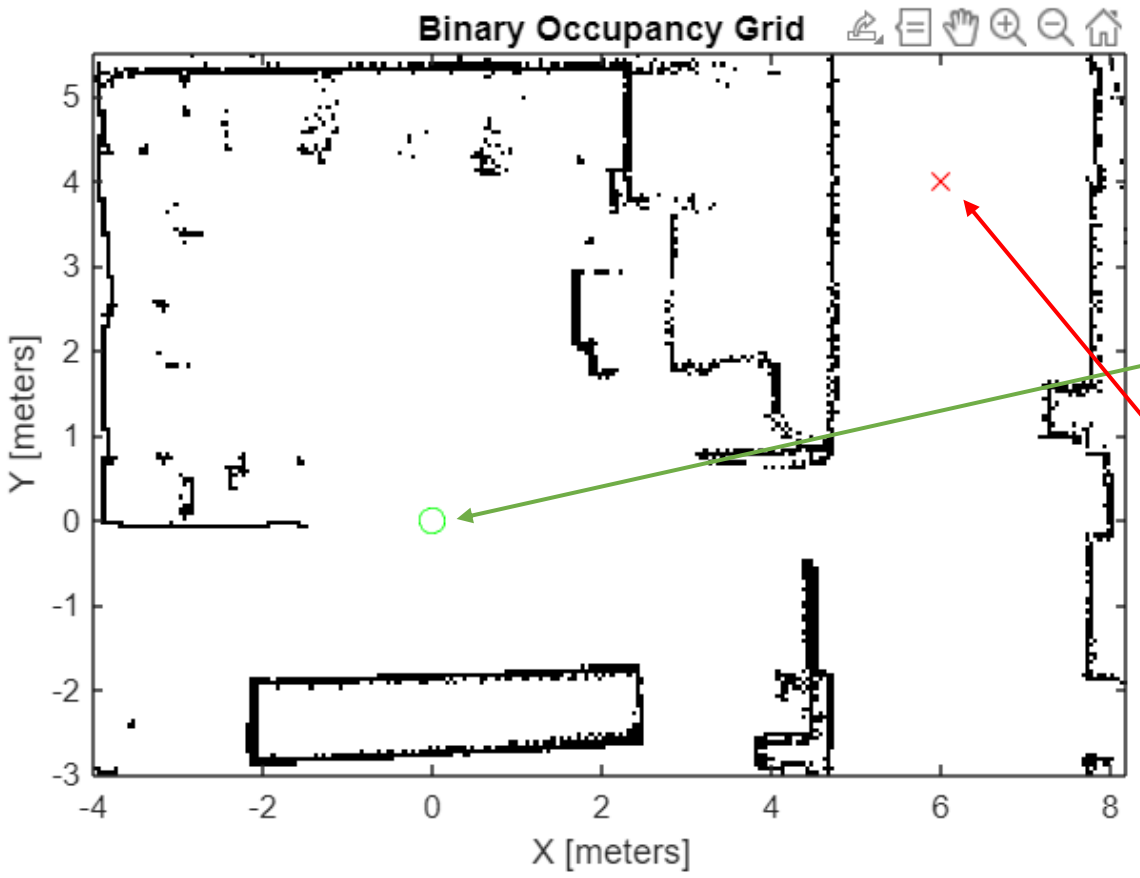


Instantiating ROS publisher and subscriber

- ROS publisher → **cmd_vel** (velocity publisher)
- ROS subscriber → **odom** (current pose)
- ROS subscriber → **move_base_simple/goal** (goal subscriber)



Load the Map

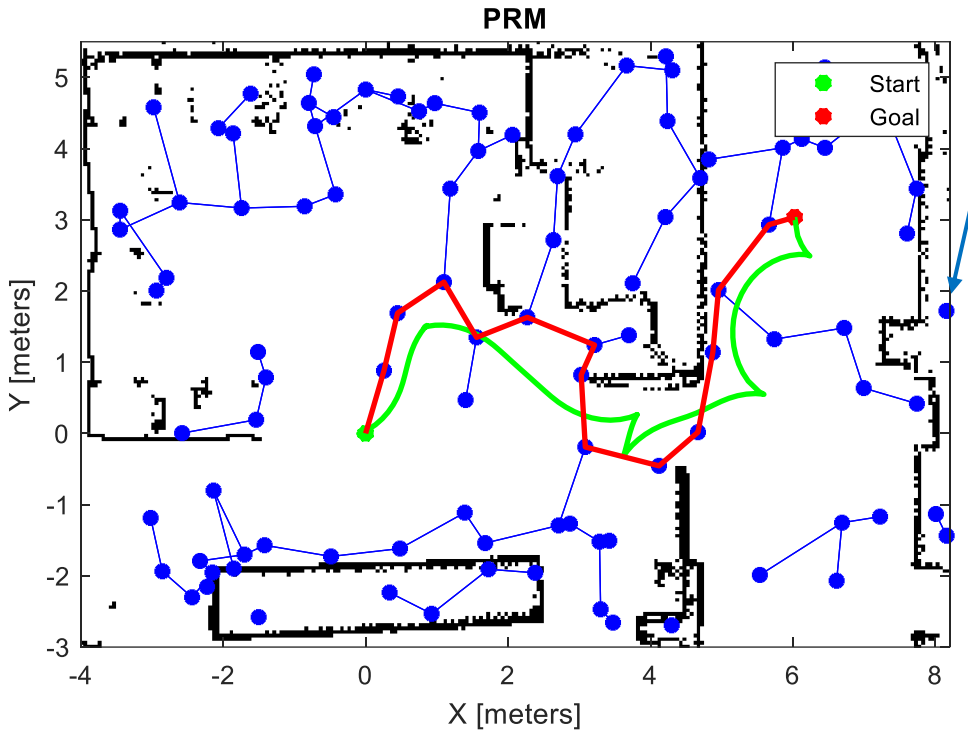



```
% plot robot pose
plot(robot_pose(1),robot_pose(2),'go',MarkerSize=8);
```

```
% plot goal pose
plot(goalHandle.x,goalHandle.y,'rx',MarkerSize=8);
```

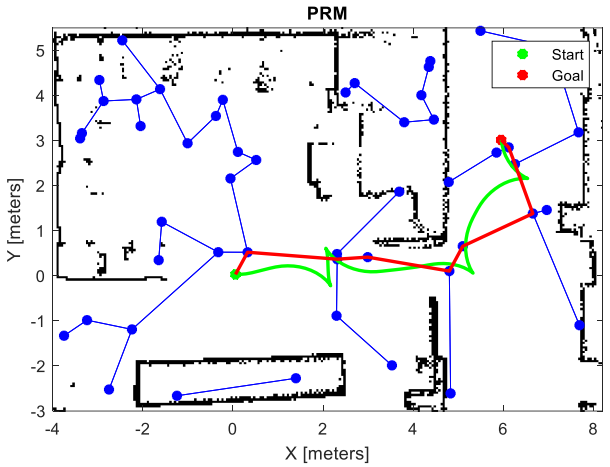
Probabilistic Roadmap Planner (PRM)

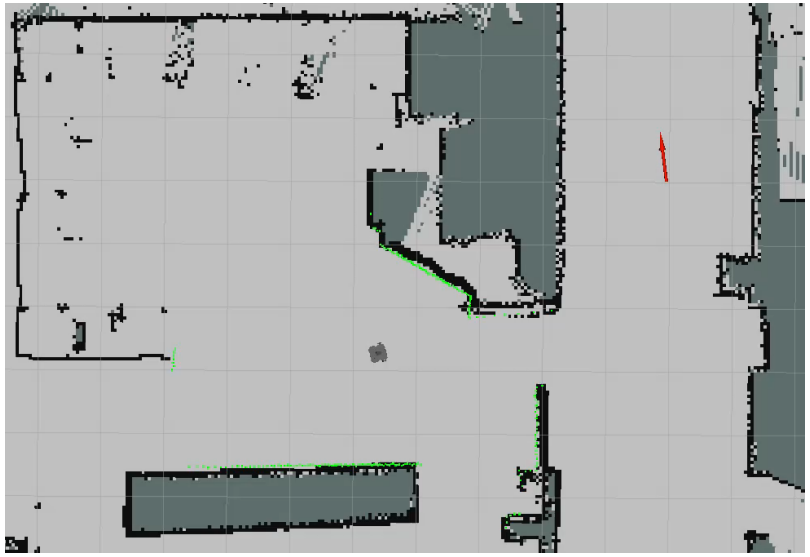
- Parameters tuned: **Number of sampled nodes (50, 100, 200, 500) and Maximum Connection Distance**



Effect of Number of Nodes and Maximum Connection Distance in **PRM**

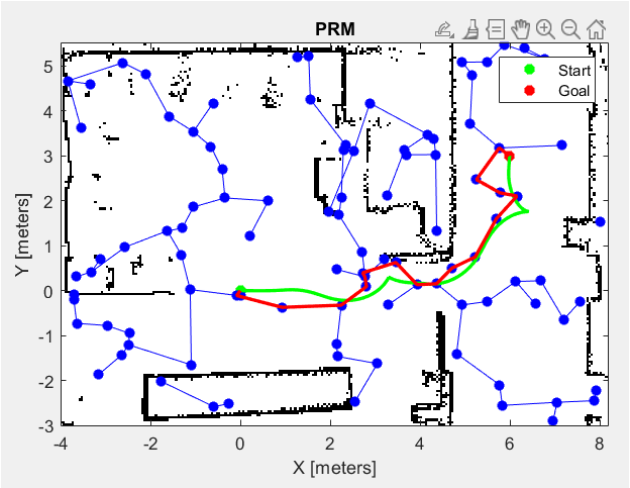
of Nodes = 50, Maximum Connection Distance = 3.5





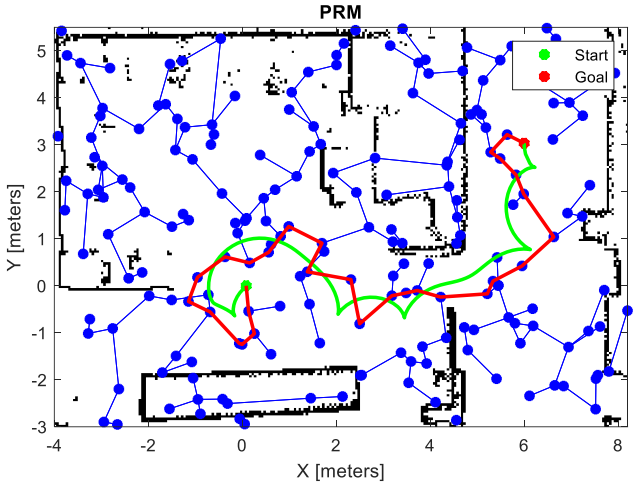
Due to repeated errors when connecting the nodes, the maximum connection distance is increased to 3.5 instead

# of Nodes = 100, Maximum Connection Distance = 2.5

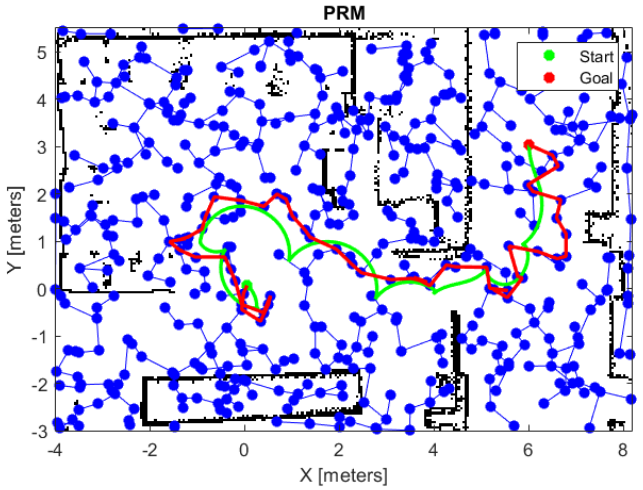


Due to repeated errors when connecting the nodes, the maximum connection distance is increased to 2.5 instead

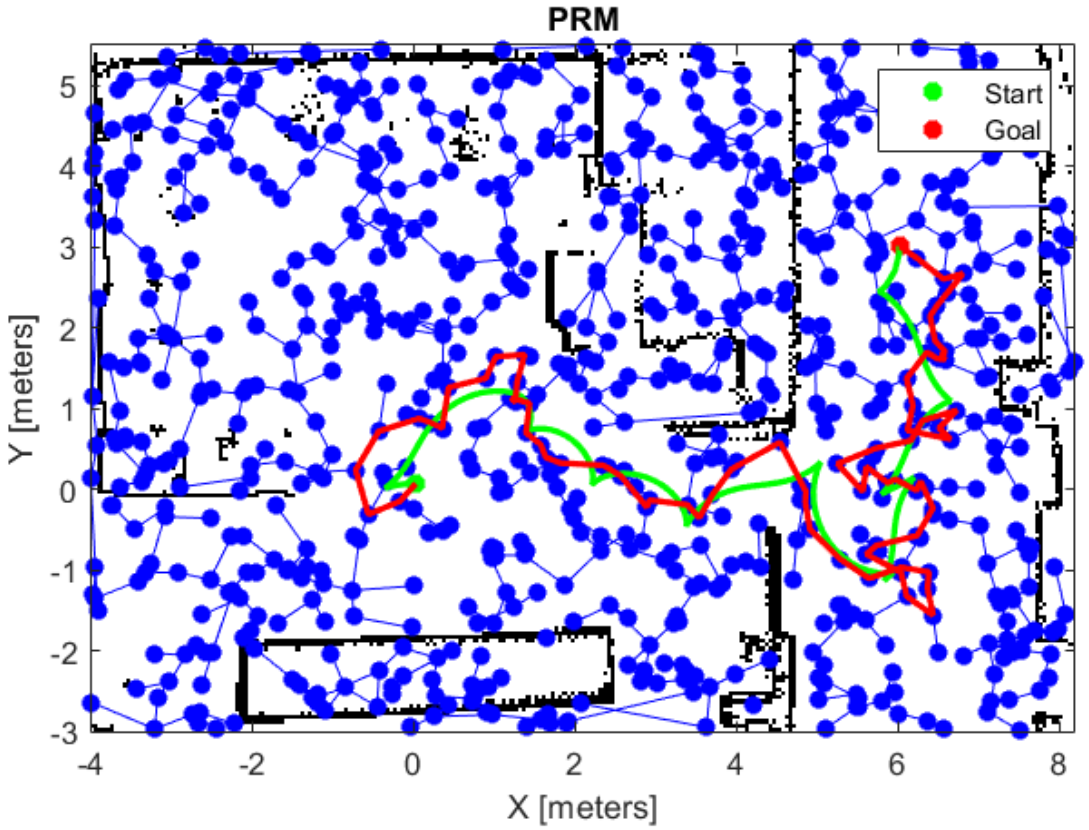
# of Nodes = 200, Maximum Connection Distance = 1.5

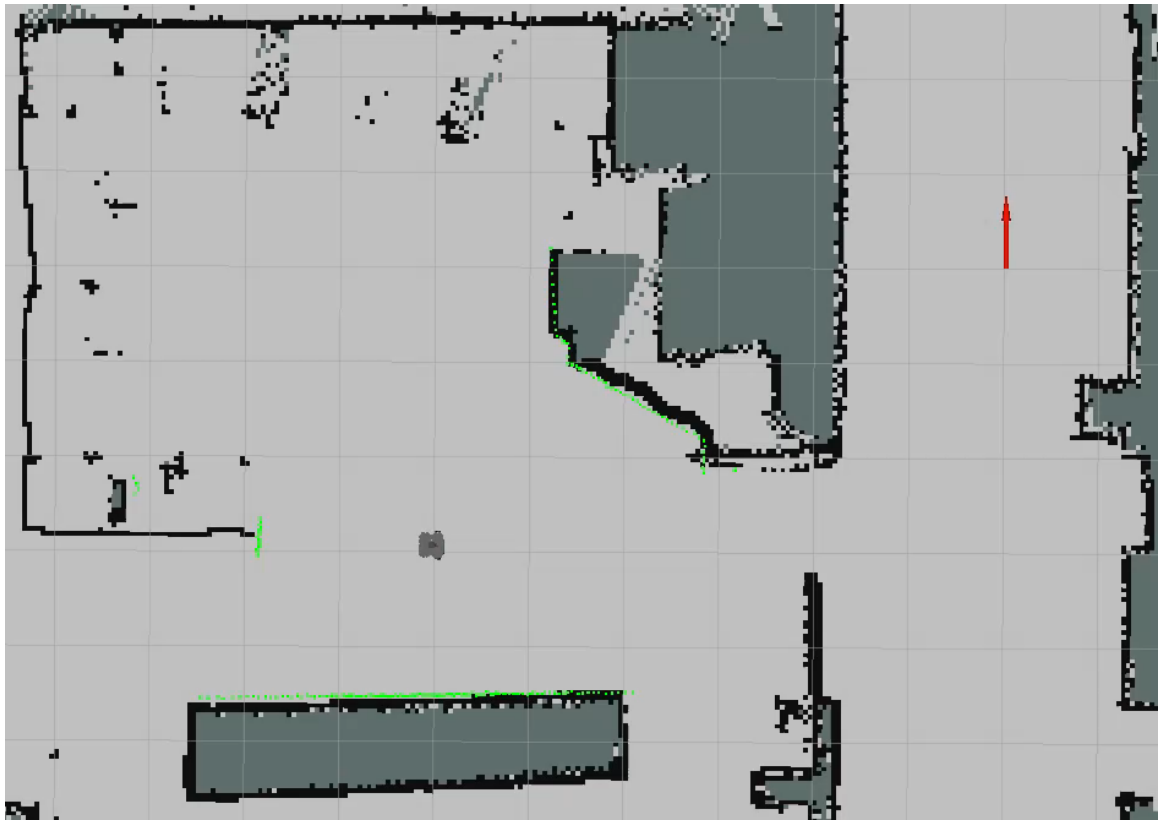


# of Nodes = 500, Maximum Connection Distance = 1.5



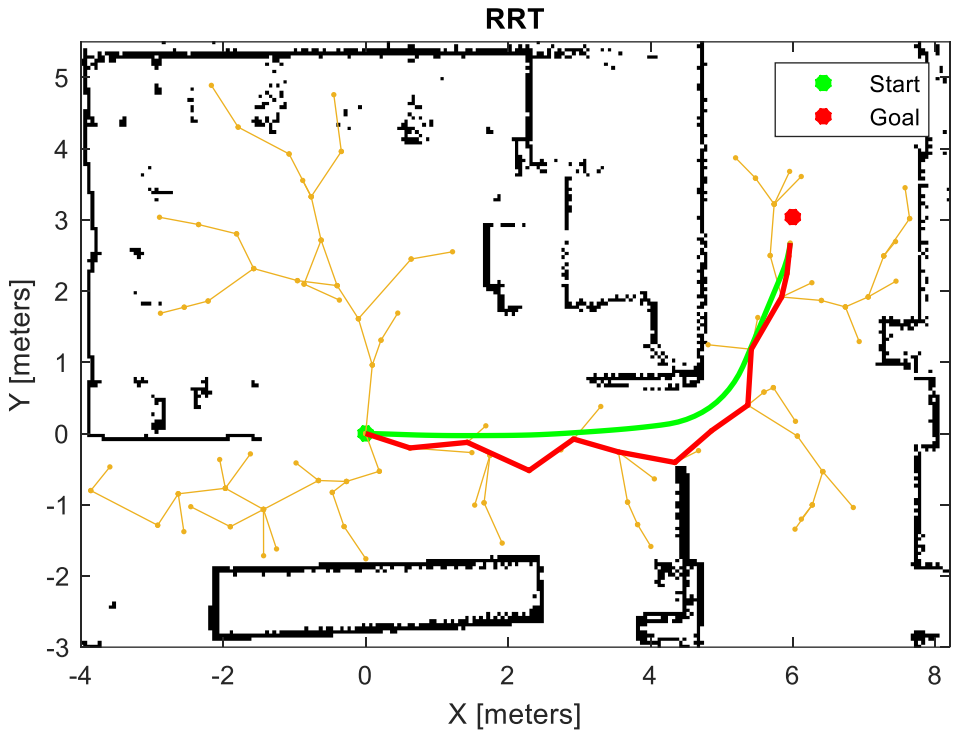
Bonus: When Number of Nodes = 750, and Maximum Connection Distance is Increased to 3

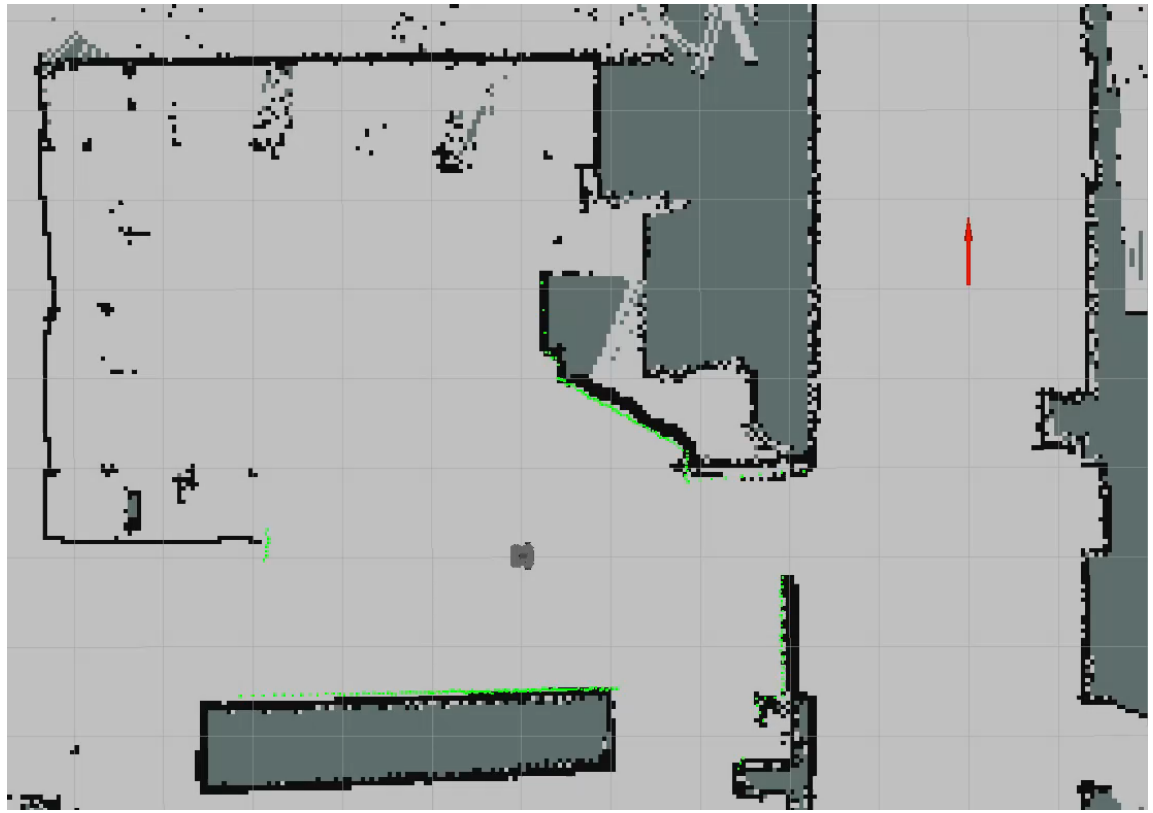




Rapidly Exploring Random Trees

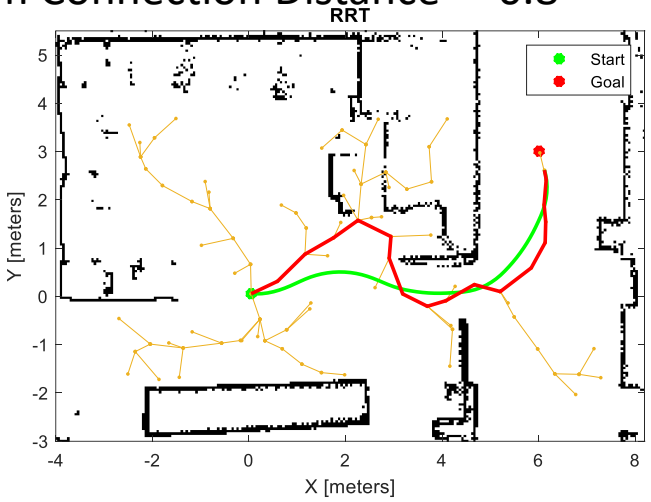
- Parameter tuned: **Maximum Connection Distance and Minimum Turning Rate (Radius)**



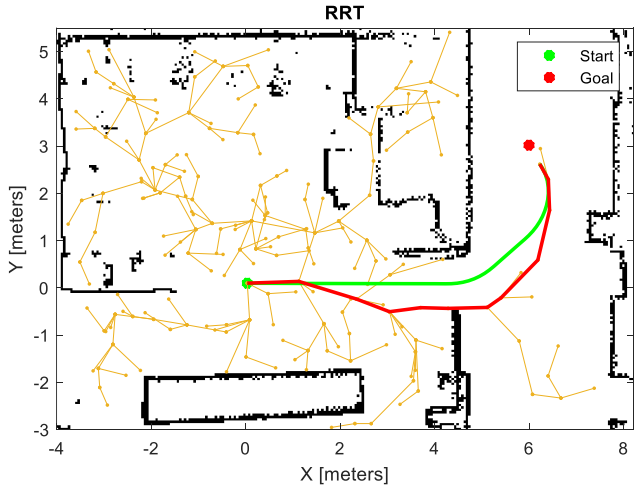


Effect of Maximum Connection Distance in **RRT**

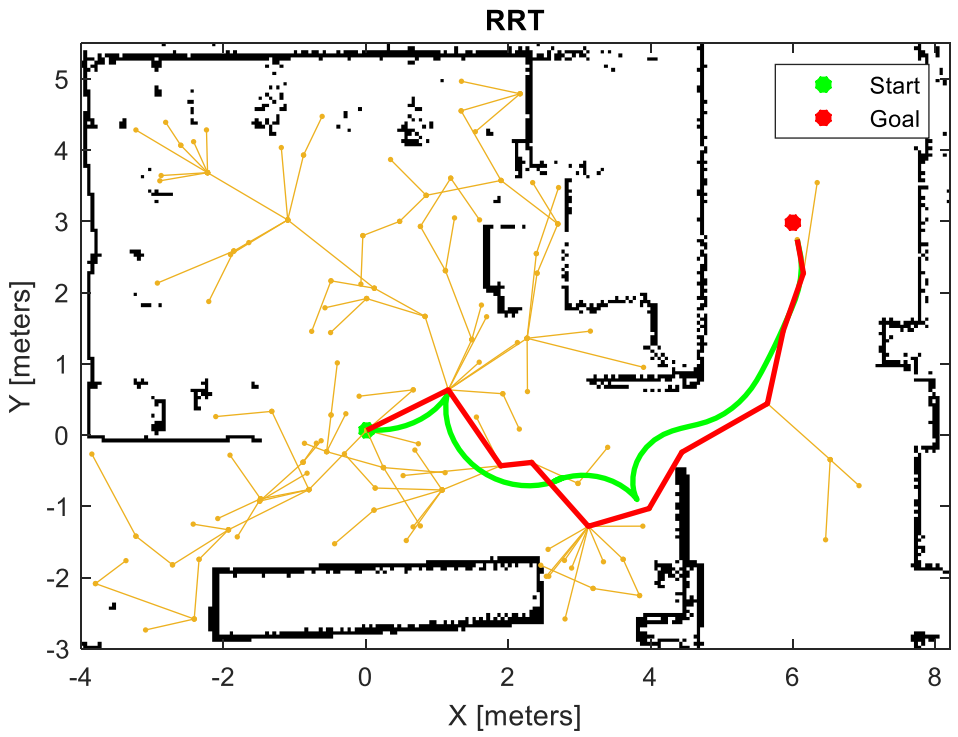
Maximum Connection Distance = 0.8



Maximum Connection Distance = 1.2



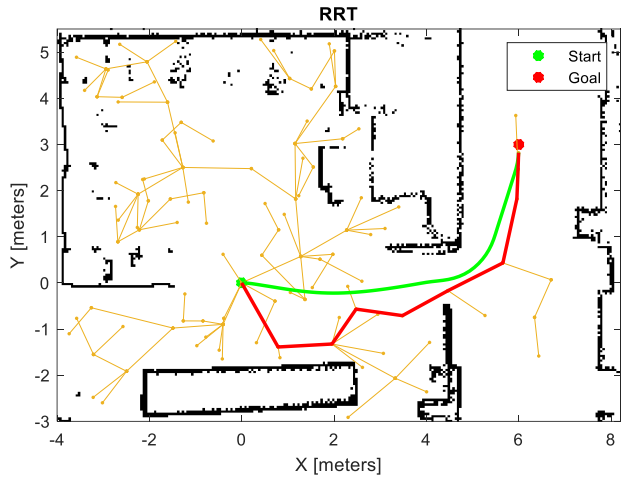
Maximum Connection Distance = 1.6



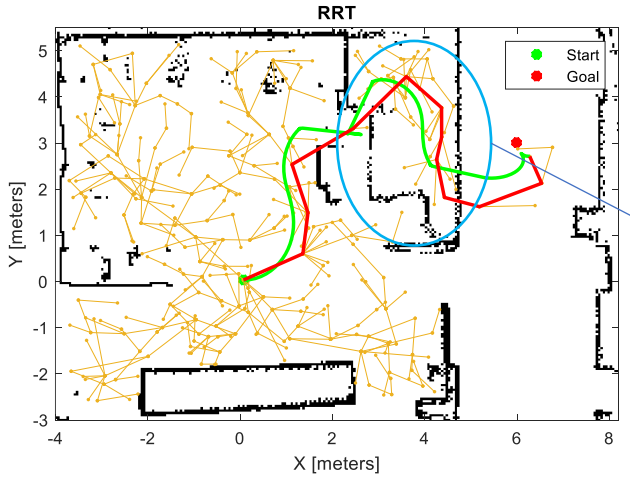
The robot couldn’t reach the goal pose as it collides with the wall (No Animation Video)

Effect of Minimum Turning Rate (Radius) in **RRT**

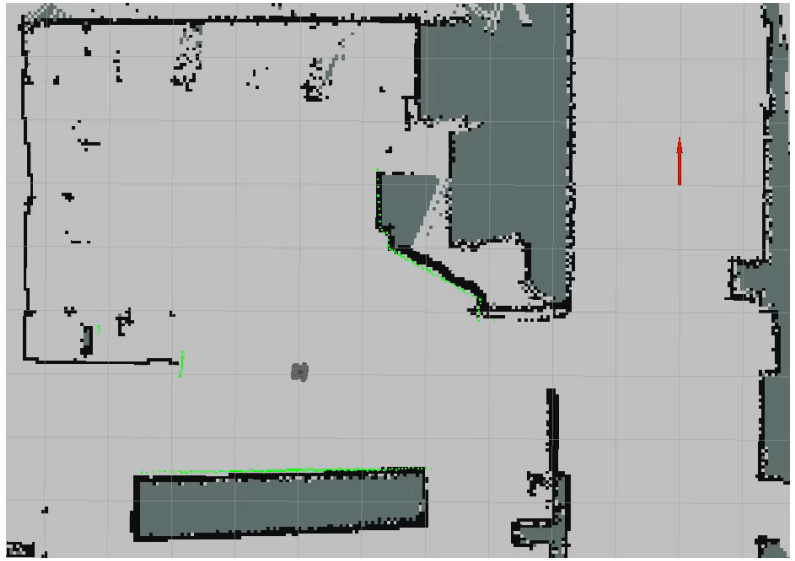
Minimum Turning Rate (Radius) = 0.03



Minimum Turning Rate (Radius) = 0.08



Possible Path
“Glitches” ?



How Velocity is Generated?

```
% Get pose
[msg,~,~] = odomSub.receive(rx_timeout);
curpose = msg2pose(msg);
```

The **msg** receives the **odomSub** subscriber , it is then converted to **curpose** using the function **msg2pose**

```
% Take last velocity
v = msg.Twist.Twist;
curvel = [v.Linear.X, v.Angular.Z];
```

The velocity is extracted from the msg.Twist.Twist. Twist message is used to command the robot movement, that consists of 'Linear' and 'Angular' field.

```
% Generate new vel commands with teb
tic;
[velcmds,tstamps,curpath,info] = step(teb,curpose,curvel);
toc;
```

```
>> velcmds

velcmds =

   -0.2256   -0.3671
   -0.3599   -0.6289
   -0.2115   -0.8267
    0.0210   -1.0124
    0.2096   -1.2104
    0.3549   -1.0819
    0.5000   -0.9109
    0.5000   -0.6393
    0.5000   -0.5327
    0.5000   -0.4097
    0.5000   -0.2810
    0.5000   -0.1506
    0.5000   -0.0210
    0.5000    0.1076
    0.5000    0.2378
    0.5000    0.3732
    0.5000    0.3732
```

```
cmd_vel.Linear.X = velcmds(1,1);
cmd_vel.Angular.Z = velcmds(1,2);
velPub.send(cmd_vel);
```

velcmds(1,1) corresponds to the first column of velcmds, indicating the linear velocity along x-direction. velcmds(1,2) corresponds to the second column of velcmds, indicating the angular velocity along z-direction.

Observations

- **For PRM**

➢ Due to random sampling of nodes → Sometimes the robot may not navigate itself properly.

➢ Number of Nodes and Maximum Connection Distance between nodes must be tuned together properly

	Large Connection Distance between Nodes	Small Connection Distance between Nodes
High Number of Nodes Sampled	More feasible and shorter path, but less smooth path	More feasible and smoother path, but high computational cost
Low Number of Nodes Sampled	Able to find paths quickly, but might not able to find feasible ones	Reduces computational load but hard to find feasible path due to limited connectivity

➢ Additionally, by using PRM, it will sometimes sample the nodes near the corner. Shortest Path will likely pass by the corners of obstacles → **Nachteil: Risk of Collision!**

Observations (Part 2)

- **For RRT**

➢ The robot is able to reach the desired goal pose faster with smoother trajectories.

➢ Minimum turning rate (radius) of robot can be tuned → Path that is compliant to the robot kinematic can be constructed.

If the Minimum Turning Radius is set LOW	If the Minimum Turning Radius is set HIGH
Shorter Paths can be generated (More direct path)	Longer paths will be generated (Usually result in indirect turns)