

Robot Simulation

1 Summary

Our project is to develop algorithms and protocols for the localization of sensors and mobile robots. We focus more on the analysis and modelling using simulation tools.

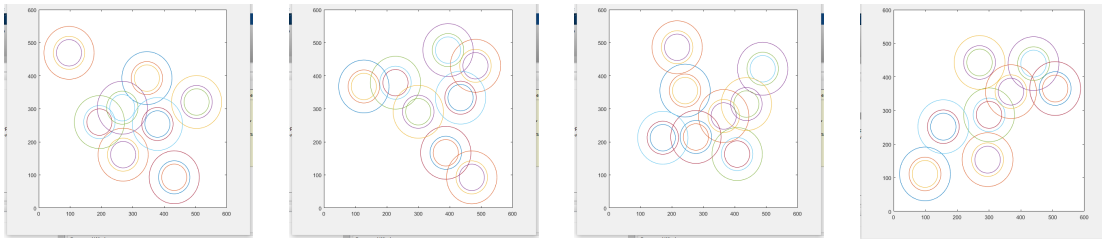
2 Simulation Program

Our first task is to simulate random location of robots in Matlab. Each robot should have three ranges, sensory range, communication range and reject range. Sensory range is the largest range that allows each of robots to sense each other within that area. Communication range is the smaller range that allows robots to communicate with each other. Reject range is the smallest range in the case when two robots are too close, they can't sense each other, which is what we want to avoid during the simulation. In order to do this, we create three files, described as following:

- "drawCircle.m": A simple function to draw circle with parameters x,y and r which stands for the locations and radius.
- "Robot.m": A class that has properties called "sensory", "communication" and "reject" which hold values for the radius of each range. Properties "x" and "y" stands for the location of robots in graph to be drawn. This class has four methods of drawing circles, "drawSensory(obj)", "drawCommunication(obj)", "drawReject(obj)" and "drawAll(obj)" which can draw three types of circle range individually and in one time.
- "simulation.m": The driver file that creates 8 robot object in random location in the graph. In order to avoid robots correlated with each other within the reject range, the algorithm is to keep looking for the position of a new robot until the condition satisfied and then create robot according to that position.

3 Simulation Result

The following graphs are 4 simulation results that randomly allocate 8 robot objects:



(a) Sample simulation 1 (b) Sample simulation 2 (c) Sample simulation 3 (d) Sample simulation 4

Figure 1: Four sample simulations