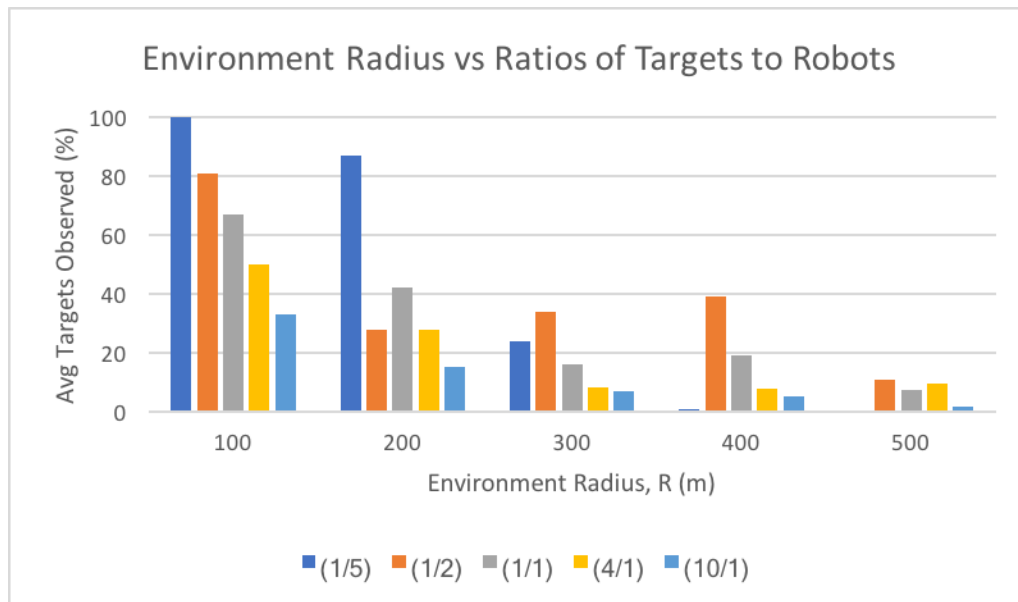


Assignment 1 Write-up

Combined results chart:



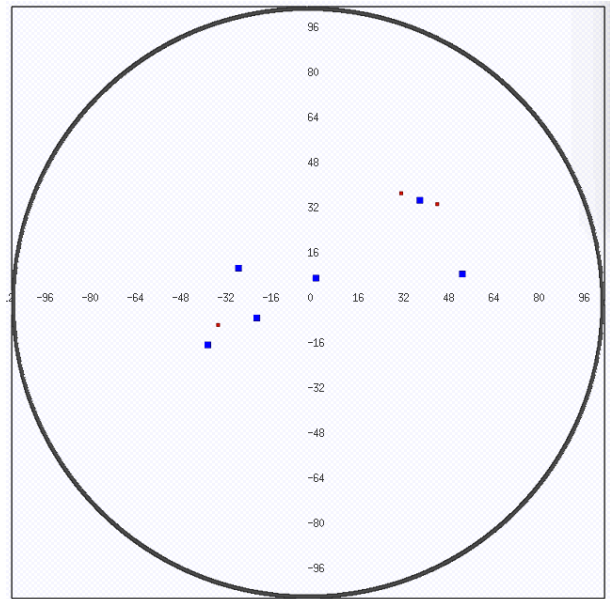
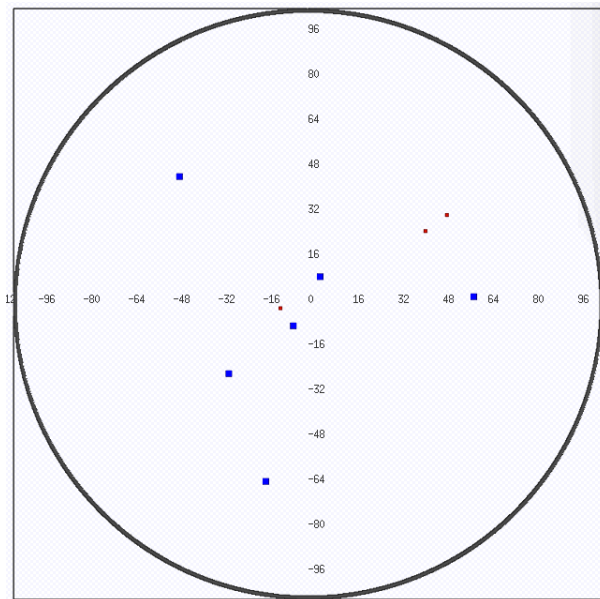
As expected, the data collected gives best results in a smaller environment, where the robots can cover the area in relatively good speed. As the environment increased, it became increasingly difficult for the robots to find the more sparse targets. The robots have max speed of 2m/s, which is also too slow for a circle of radius 500m.

In addition, as the ratio of targets to robots increase, there are less targets observed in average because there are many more targets than robots to find. Due to robot tracking, once a robot finds a target, it will follow it, making it harder to "free roam" and find other targets if they are sparse. However, the average number of targets observed was significantly impacted by the robots and targets initial position. Some simulations will perform significantly better because the robots started near the targets or the targets were congregated in a certain area.

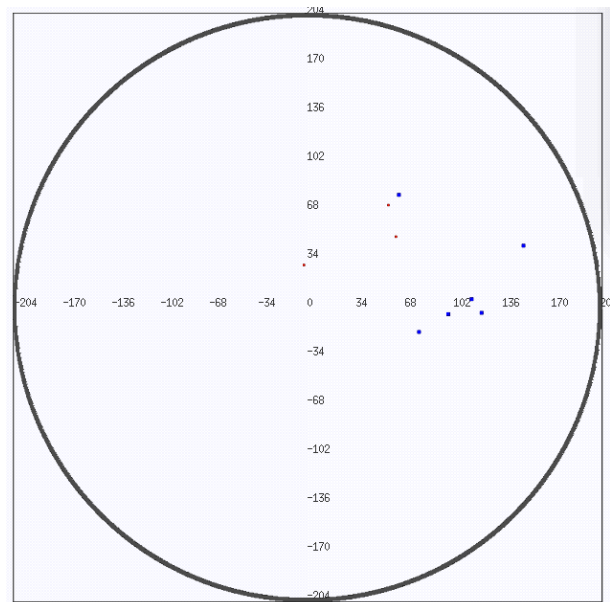
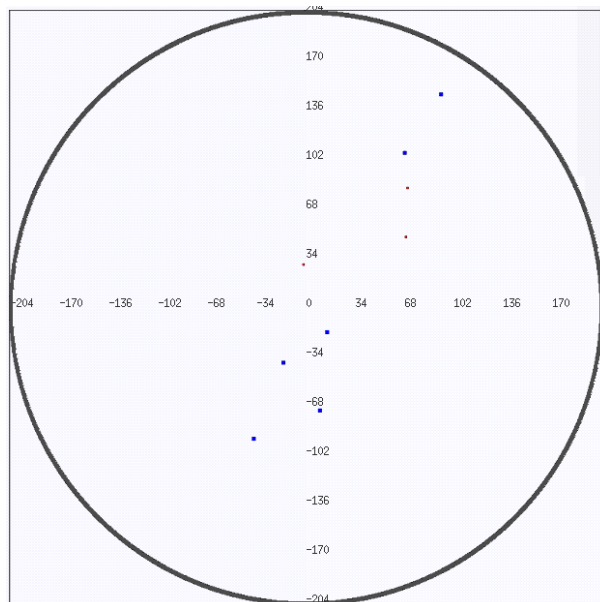
One last observation, for the $n/m = 1/5$, the outcome was almost completely random! When the target spawns near a robot, the target is tracked for the whole duration for a 100% targets observed. However, if the target was initialized very far away from robots, odds are against the robots to find it leading to a 0% targets observed. In a smaller area, the robots almost always find the target though!

Below are some screen shots for comparison:

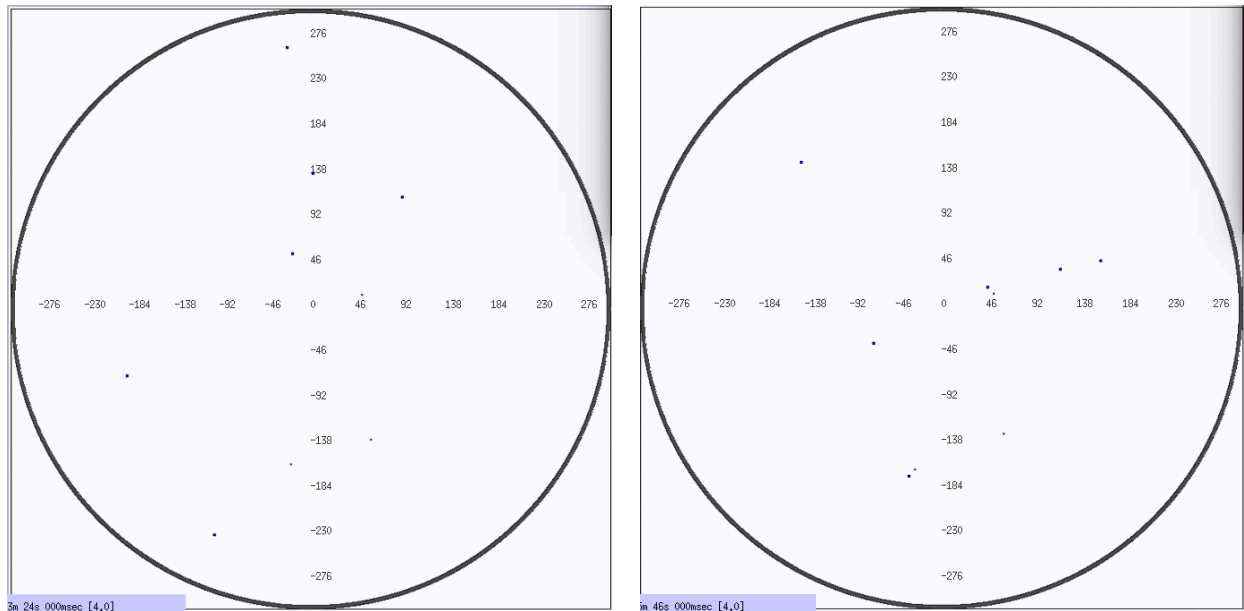
$R = 100, m = 3, n = 6$ (left: start, right: end)



$R = 200, m = 3, n = 6$ (left: start, right: end)



$R = 300$, $m = 3$, $n = 6$ (left: start, right: end)



In order to run the simulation:

- 1) open terminal
- 2) go to directory
- 3) `> player stageall.cfg`
- 4) open another terminal
- 5) go to directory
- 6) `> python sim_hol.py n m R`
 - where n is num of targets $[1, 20]$
 - where m is num of robots $[1, 10]$
 - where R is the radius of the environment in meters
- 6*) in order to change the radius, the `rink.world line size[x y 1]`
 - where x, y are the circle diameter size (plus 10 for bitmap boundary)

Notes on the simulation and control design:

In order to simplify the simulation design, I installed the sensors on the targets instead of the robots. The sensors (ranger or blobfinder) can be easily installed on the robots and the simulation results would remain the same. It was however, simpler for the simulation to put the blobfinders on the target. The simulation results are not affected; however, the design/code is simpler!

If the sensors were installed on the robots, then I would have had to run a for loop to create the $B[t]$ matrix, then another for loop on it to calculate the force summation. With the current design, I can calculate the matrix and force summation in one loop.