Class Project

Report

The goal of the Project:

The goal of the project was to control 6 robots in a simulator (PyBullet) and navigate them from starting position to the destination. While doing this they had to change the formation like a square at the starting position and triangle at the end position. One more important thing was to keep them at least 1m apart from each other in any formation.

To run the simulation:

Open the folder pybullet_swarm\python and run the python file with name run_simulation.py.

I have not changed anything in any file other then robot.py . And I have done everything in sublime text.

Sometimes, you have to wait for one formation to another, and I have also noticed that sometimes I need to click and see the cmd while running the simulation, so that it doesn't hang.

Description of the controller:

I used the following controlled:

$$F_{formation} = \begin{bmatrix} K_f\left(Dz_{x,desired} - Lx\right)_i + D_f(D\dot{z}_x - L\dot{x})_i \\ K_f\left(Dz_{y,desired} - Ly\right)_i + D_f\left(D\dot{z}_y - L\dot{y}\right)_i \end{bmatrix}$$

Where, K_f and D_f are the positive gain which I gave valua e equal to 10 and $2\sqrt{K}_f$ respectively. L is the Laplacian matrix and D is the incidence matrix . $z_{desired}$is the desired direcrtion vectors associated to each edge. This was given in question 2 of Assignment 3.

But I didn't use the 2nd term of the above equation as I am assuming that acceleration is 0, so velocity will be constant and every robot has a GPS so its easy for them to go to a specific location.

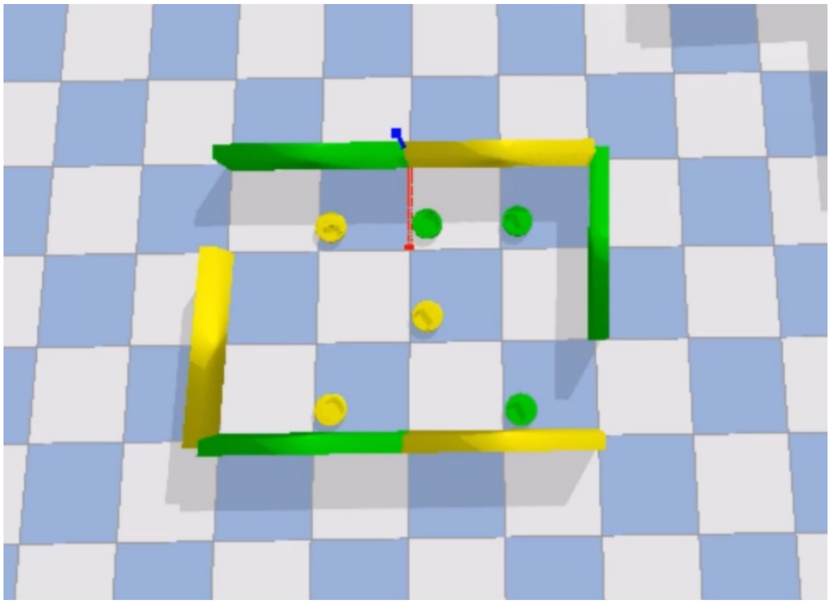
Results:

The result can be seen in the video, but I have pasted the screen-shot of the main formations.

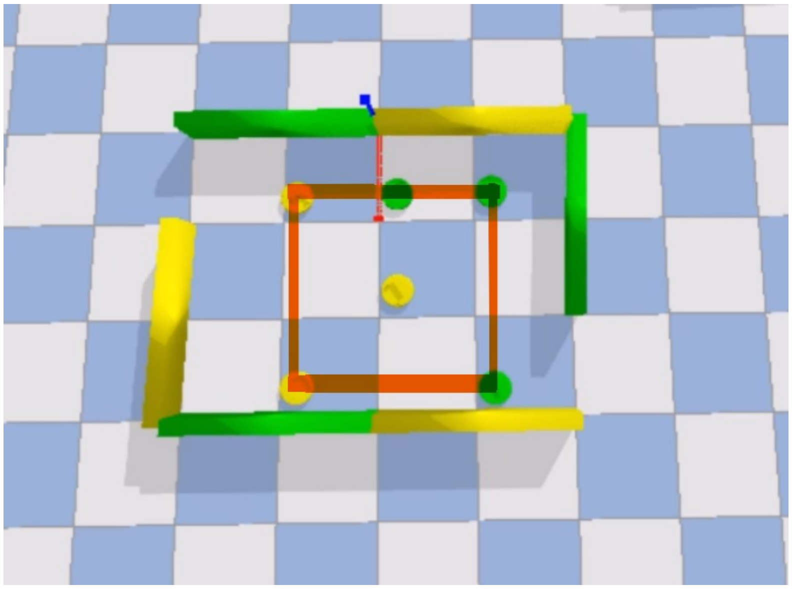
In the simulator, after the initial position, the robots first form the square formation. I did send one of the robots in the middle so that all the robots keep a connection to the middle one otherwise at 2m the corner one would have lost the connection. Then I needed to get them out of the first room, but if I would have given them the direct command to become straight line they could have jumbled up. Therefore, given a different configuration so that they can easily become line and go out of the first room.

In the code, I have given all the different formation coordinates in a matrix and it goes to the next formation till a counter T reaches 12000. So for every formation, all the robots wait till T reaches 12000 and then changes to the new formation.

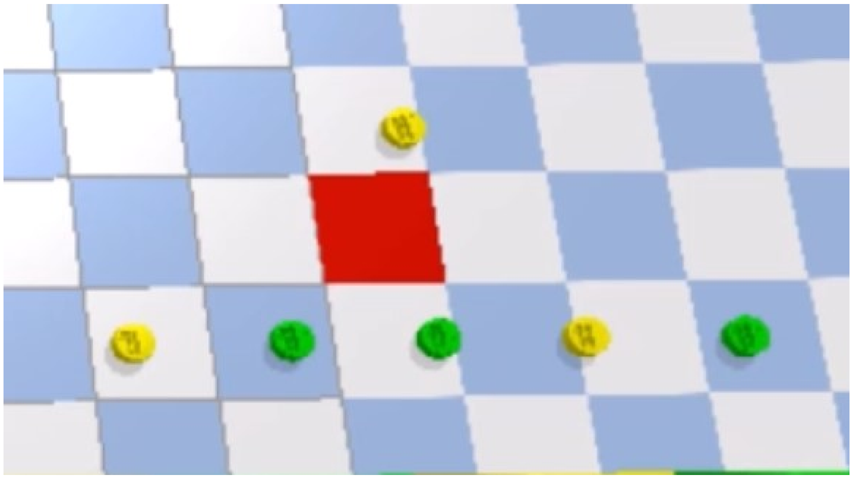
From start to end, the robots didn't bang themselves into each other or to the walls. It seems from some angle that they are touching the walls but I zoomed in and checked.



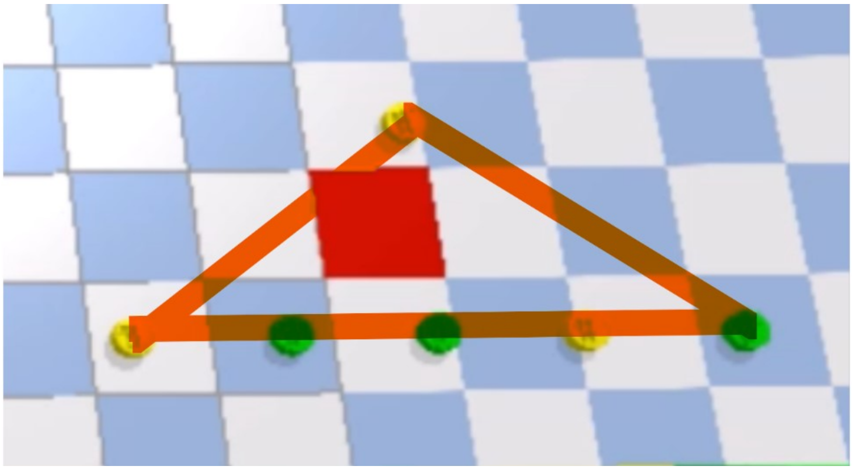
(a)



(b)



(c)



(d)

Figure (a) and (b) shows the square formation at the starting point. Figure (c) and (d) shows the triangle formation at the destination.

Pros of the approach:

The advantages to this approach are it is very easy to implement and we can control the way the robots move from one point to another, having different formations.

Cons of the approach:

The disadvantage of this approach is that it's not dynamic. If we add any obstacle in-between then we have to make changes in the path of the robots.

Conclusion:

I was able to do the above-defined task. Although the simulation was slow but it performed everything with perfection. This was a great experience of learning how a swarm of robots can work and do different formation. Watching them on simulation, doing different formations was fun. I learned that there could be many possible ways of performing this task.

I am aware that if we perform the same task in the real world, it would be a bit difficult and we have to handle many more obstructions.