

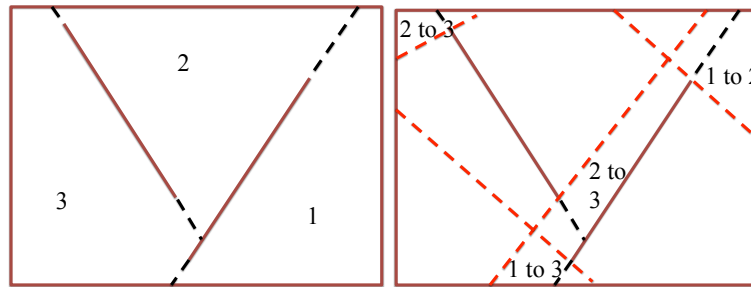
## Weekly report

### 1 *My Objectives this week*

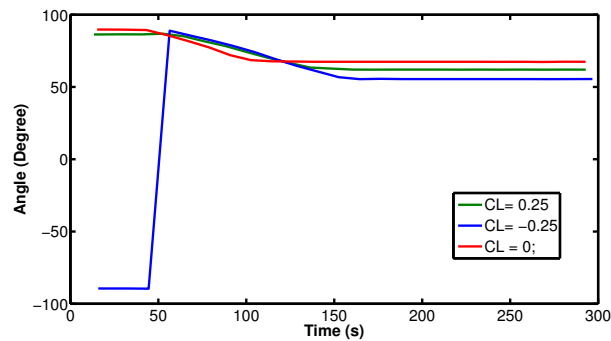
- I want to implement the partitioning part with map. I have to write an algorithm for that and load that.
- I hope to be able to finish at least 4 more full experiments and also write all the missing part of the paper.
- I would start the torque control journal paper with Mable and Lillian, I want to start with the pure torque control task this week and write the code for that.

### 2 *My Accomplishments this week*

- Wrote an automatic torquecontroller which finds the object, understands its orientation and push it.
- We discussed together about the algorithm of partitioning the map. So the algorithm is this: We learned from humans that they would ignore the robots that are not on the path of the robots. In our previous work we had this problem that when the robots were separated, and they were in variance control mode, they would get stuck and our algorithm failed. In this work we present an algorithm to make sure that the swarm will not get stuck at any point. First we make the whole map, all the obstacles are pink, and we detect them with our vision system. For simplicity assume that all the obstacles are lines. When an obstacle is detected, it has one or two end points and also its orientation is detected by processing of the snapshot of the webcam. Any of the ending points is a potential for having a big region. Without loss of generality, assume that all the obstacles are attached from one point to the walls. For the other ending point, draw a line until it intersects another obstacle. This cause the map to divide into the regions made by those lines. We call these regions our main regions. There are two states that the object could be in. It starts in the "main region" state until it hits a boundary line. If it passes a boundary line, it will go to the "transferring" region state. In this state, we have two temporary lines which each one is in one of the main regions. We consider this area as transferring region. If the object passes those temporary lines, it will go again to the "main region" state. In each region the algorithm will consider the robots inside that region only. Therefore, if some of the robots were behind, they would not be considered for mean and variance and the swarm would not get stuck.
- I wrote the pure torque control challenge in matlab. it is now COMPLETELY autonomous. It sees the pink object, understands its orientation, and you can set different CLs to determine where the goal should be.
- Mahek made a light long rectangle with 3D printer. We tested torqueControl using that and got some data, these are the very first datas, I would get the better data and share that with you. There is a small bug for the orientation determining, that at exactly 90 degrees it will change its sign. That is why it starts from -90 degrees and jumps to 90 degrees. It seems to me that because we are not controlling them exactly to the point that they should go, they look really the same.



**Figure 1:** An example of having three arbitrary straight obstacles. Left is how the main regions are detected, and right is the transferring regions.



**Figure 2:** Three different CLs to find the best. I have to redo these experiments, but I wanted to show how they look.

- I taught Mahek how to use the laser cutter.

### 3 My Plan for next week

- I want to implement the partitioning part with map. And test that idea.
- I would finish the pure torque control experiment and finalize its plots.

#### 3.1 Meeting with Dr. Becker

- I need to talk about the case reviews.
- I need to talk about our team.
- I want your feedback on the algorithm.