# Initial Report

*Bruno Borges Da Costa, Camilla Carta, Alexis Dewaele, Yoann Lapijover*

## Description of the Project:

The purpose of this project is to implement a multi-robot navigation strategy in two kinds of environments. First of all, we have a cluttered environment i.e. a group of robots starts from one side of the "world" and has to navigate through a maze with obstacles to reach the other side of the "world". The group of robots should be able to navigate around said obstacles and regroup. In a second scenario, we have two groups of robots and each start at opposite ends of the "world." They both have to cross the board thus avoiding each other.

The navigation strategy has to be fully distributed and will have to be implemented in simulation using Webots and in real experiments with real E-pucks robots. We will use several performance metrics to assess the efficiency of our strategy.

## Preliminary Results:

At this point, we only worked in simulation.

For both scenarios, our implementation is based on the code given for Lab 4. In particular, we modified the Reynold's Rule to obtain smoother displacements (Figure 1). We also implemented the performance metrics given in the assignment.

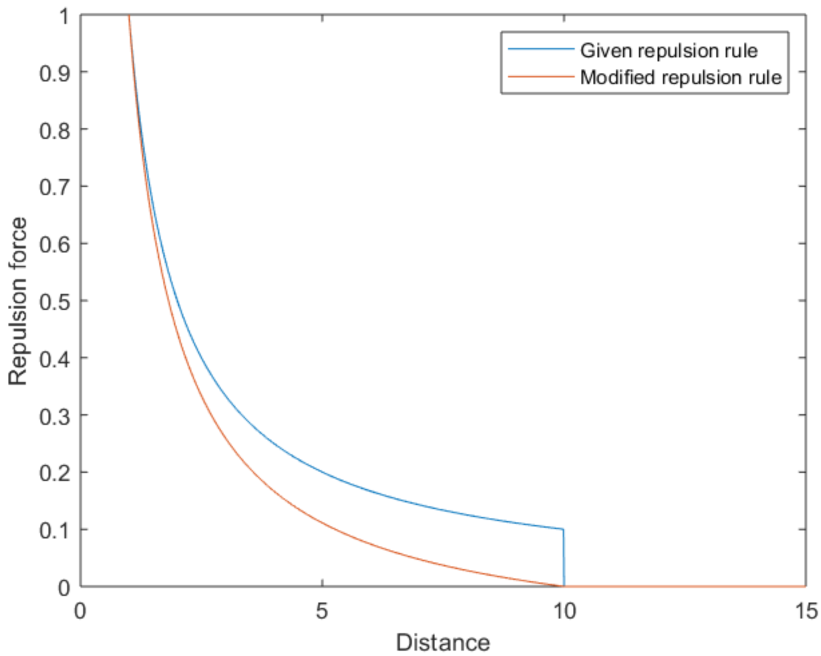


Figure 1: Modified Reynold's Rule

Scenario 1:

Our navigation strategy in the first scenario works rather well: testing for a group of 5 robots, we used the "obstacle.wbt" world given in Lab 4 while modifying the obstacle position to obtain a cluttered environment.

Our group of robots stayed clustered and managed to navigate through various configurations of obstacle placement to reach the other side of the board. The performance metrics showed that the results for orientation and cohesion were greater than 0.9 for the experiments we run. The performance metric for the velocity was always around 0.5, which was expected as the robots velocity is inferior to the maximal speed they can achieve. Thus, the overall performance was between 0.45 and 0.5.

We found that, while running the simulation, there is a slight deviation of the cluster's direction, which could be due to two main causes:

- the imprecisions coming from the wheels' positioning;

- the drift of the robot caused by possible obstacle collisions, which could induced by a conflict between the results given by the obstacle avoidance algorithm and the flock controller.

Scenario 2:

For this scenario, we used the "crossing.wbt" world which was given in the assignment. We tested the standard configuration where 10 robots are divided between 2 groups of 5.

We had to modify the "flock_controller.c" code to separate the two groups and we added a parameter to know which scenario we are working in.

We tested different values for the gains, but the results are quite chaotic for now. This was tested before we implemented the performance metrics, thus the next step would be to analyse the results more methodically.

## Future Work Plan:

The first priority is to correct the behaviour for the $2^{nd}$ scenario, then we will adapt the code to the real ePuck robots and test in the real world arena. An important step will be the optimisation of the various parameters.

The second priority will be to improve the robots performance by prioritizing certain events over others (i.e. obstacle avoidance over migration urge).

The last point will be to test for an higher number of robots (up to 20) and to analyse the corresponding performances.