Multirobot formation for target enclosing

1. GOAL

The goal is to study the problem of multirobot formation in the application of enclosing a target in this practical exercise. The task is to program the algorithm of the enclosing formation control and the implementation of the simulation of the team of robots performing the formation control.

2. METHODOLOGY AND SETUP

Before starting the exercise, read carefully the complete instructions of the practice. The practical session consists of programming from scratch the formation control task. Python is recommended, but any other programming language can be used.

3. ENCLOSING FORMATION CONTROL

Given a set of mobile robots $\mathbf{q_i}$ and the desired formation $\mathbf{c_{ij}}$ around a target $\mathbf{q_N}$, enclosing it, define a simulation environment. Consider first a 2D space in which the robots move on the floor plane.

Implement the control law for every robot in order to perform the enclosing task.

Consider as enclosing formation different geometric patterns such as square, triangle circle, etc.

4. OPTIONAL TASKS

Task F1: Alternatively to the enclosing task, implement the formation task where the robots are required to reach a particular shape formation. In this case, no target is considered.

Task F2: Program the orbiting formation task where the enclosing formation is required to gyrate around the enclosed target maintaining the desired formation.

Task F3: Instead of a static target, now impose an arbitrary motion to the target. Study the performance of the system with different parameters and target speeds.

Task F4: Extend the 2D implementation to the 3D case.

Task F5: In the control law, each robot perceives the location of the other robots. This means global information is required. Modify the program to perform a distributed control where the information of the other robots is obtained through a network defined with a static graph of communications.

Task F6: Simulate the results of the enclosing task in ROS + Gazebo using a robotic platform (for example: turtlesim, turtlebot, etc.).

5. SESSION REPORT

As a result of this practical session, the final developed code will be submitted through the ADD (https://moodle.unizar.es/add/).