



Task Sheet 2 - Behavior-based robotics and collective behaviors

Objective

- Get used to robot simulators
- Learn how to program a single robot
- Learn how to program swarm behaviors

1 Preparations

In this tutorial, we are using a robot simulator. Feel free to use any multi-robot simulator of your choice; a few options, for example, are:

- A. Henrik's sim based on Box2D <https://gitlab.com/InfectedBytes/SwarmRobotics.git>
- B. RoboroBo <https://github.com/nekonaute/roborobo3>
- C. ARGoS <http://www.argos-sim.info/>
- D. Gazebo <http://gazebo-sim.org/>
- E. PlayerStage <http://playerstage.sourceforge.net/>
- F. JBotEvolver <https://github.com/BioMachinesLab/jbotevolver>

Try to make a good choice in what is available for your preferred OS, and also in terms of the overhead created by setting it up.

2 Behaviors of a single robot

First, we focus on the behaviors of a single robot. Design a rectangular environment, bounded by walls, and almost empty (maybe a few obstacles). Initially, position the robot randomly somewhere in your robot arena.

- A. Program a collision avoidance behavior – a robot that moves around while avoiding crashing into walls.
- B. Program a wall follower – a robot that moves along the walls without touching them (i.e., avoid the easy solution of letting the robot slide along the wall).
- C. Program an appropriate behavior for a vacuum cleaning robot. What are strategies to ensure that (almost) everything gets clean?
- D. Just play around a little and see what kind of behaviors you can generate.

3 Behaviors of robot swarms

Next, we focus on the behaviors of robot swarms. Initialize your robot arena with 20 randomly distributed robots. All robots should be operated by instances of the same controller (i.e., copies of the same controller).

- A. Program a behavior that makes robots stop if they get close to another robot. Create an implementation to show this behavior. Possibly, you can also fake such a sensor based on distances between robots to keep the simulation simple.
- B. Extend your program to limit the time a robot stays stopped according to a defined waiting time. When a robot wakes up and moves again, it might immediately stop again, depending on your implementation. However, the idea is that the robot leaves at least small clusters before stopping again. Think of a strategy to implement this behavior and change your program accordingly.
- C. Tweak your implemented behavior and parameters such that your robot swarm eventually aggregates in one big cluster with robots leaving only from time to time but rejoining it soon again.