Random Walk

The problem

Random walk and obstacle avoidance are two key behaviors in swarm robotics. Obstacle avoidance behaviors aim to provide safety to the robot by staying clear of detected obstacles and avoiding collisions between robots. They can follow many different schemes. For example, the robots might be repelled from detected obstacles or they might react (deterministically or stochastically) to sensor input in the direction of motion. Random walk can be defined as the behavior in which a robot moves without aiming a particular trajectory. In most random walk methods, changes in the direction of movement of the robots occur when they interact with other robots or objects in the environment. The selection of a direction of movement can follow deterministic or stochastic rules. In many cases, robots also embed obstacle avoidance behaviors to safely navigate in the environment.

Experiment layout

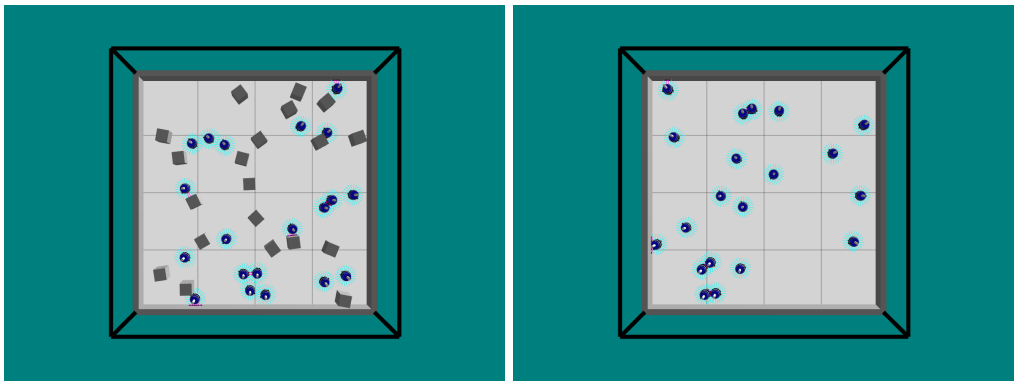

Figure 1: The obstacle avoidance experiment (left) and the random walk experiment (right).

In both experiments, twenty footbots are distributed randomly in the experimental arena that is bounded by walls. The arena for the obstacle avoidance experiment additionally contains twenty boxes randomly distributed in it.

Robot layout

In this experiment, the robot has access to the following sensors and actuators:

Available sensors	Available actuators
`robot.id`	`robot.wheels`
`robot.random`	`robot.leds`
`robot.proximity`	

Objective

This exercise is composed of two parts. In the first part, you will write control software for the obstacle avoidance experiment (`obstacle_avoidance.argos`). Program the robots in the swarm to move through the environment, without colliding with each other or the boxes in the environment.

For the second part, you will work now in the random walk experiment (`random_walk.argos`). The goal of this experiment is for the swarm to explore and cover the arena uniformly. As a first step, test your control software from the obstacle avoidance mission now also in this mission. What do you observe? Do the robots cover the arena uniformly? Or do they tend to stay around the walls? If your swarm is not covering the arena well, try to modify your control software until obstacle avoidance and random walk work together.

General remarks

The control software of the robots is executed in the form of time steps–that is, the script is executed in the simulator once for each time step. In this experiment, the time step has a length of 100ms. In other words, each of the actions defined in the Lua script will be executed 10 times per second.
Many basic robotic behaviors follow the *Sense-Think-Act* scheme, where the following three steps are executed iteratively: first, gather information through sensors; then, compute the action to be executed; finally, execute the action.
If you are stuck, try to think about how you behave when moving on the street. How are you avoiding other people while walking?
Remember that the individual actions of each robot are the ones that lead to the robot swarm achieving the desired behavior.