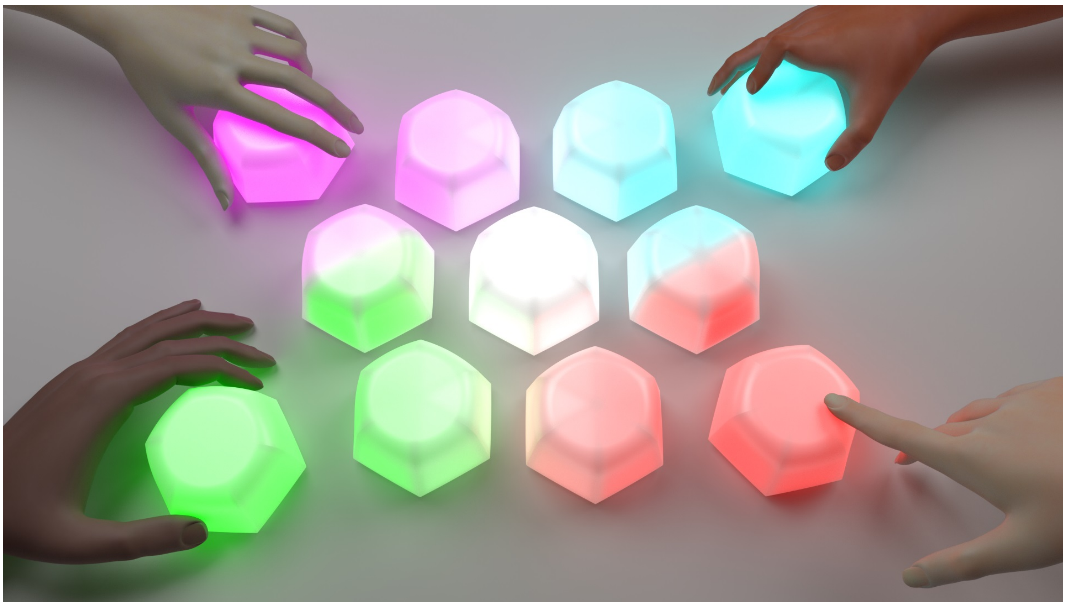
CHILI Robots

Tangible Swarm Interaction with the Cellulo Robots

To Read before the first session



Part I

Theoretical background

Swarm robotics is a subfield of Robotics that make use of a distributed system to control several robots. Swarm robots are usually simple and relatively cheap, with limited functionalities, but together (with capacity of aggregation or formation) can form a complex and robust system. Swarm robotics aims to study the design and the behavior of these small robots. Simple rules can lead to the emergence of complex behaviours at the swarm level. A key element of the swarm is the communication between its members, aiming at cooperation among them. Such systems are expected to hold at least the following 3 properties:

- Robustness: ability for the swarm to function even with some failures of some individuals or changes in the environment.
- Flexibility: ability to propose adapted solution to the given tasks
- Availability: ability of the swam to perform whatever its size (number of individuals)

In a swarm every robot is autonomous. Robots are usually able to localize themselves relatively to their closest neighbors. Swarm robots can interact with each other and with their environment. Their detection and communication abilities are usually local and limited. The robots are usually not connected with a central unit; and they don't have a global knowledge of the system in which they stand.

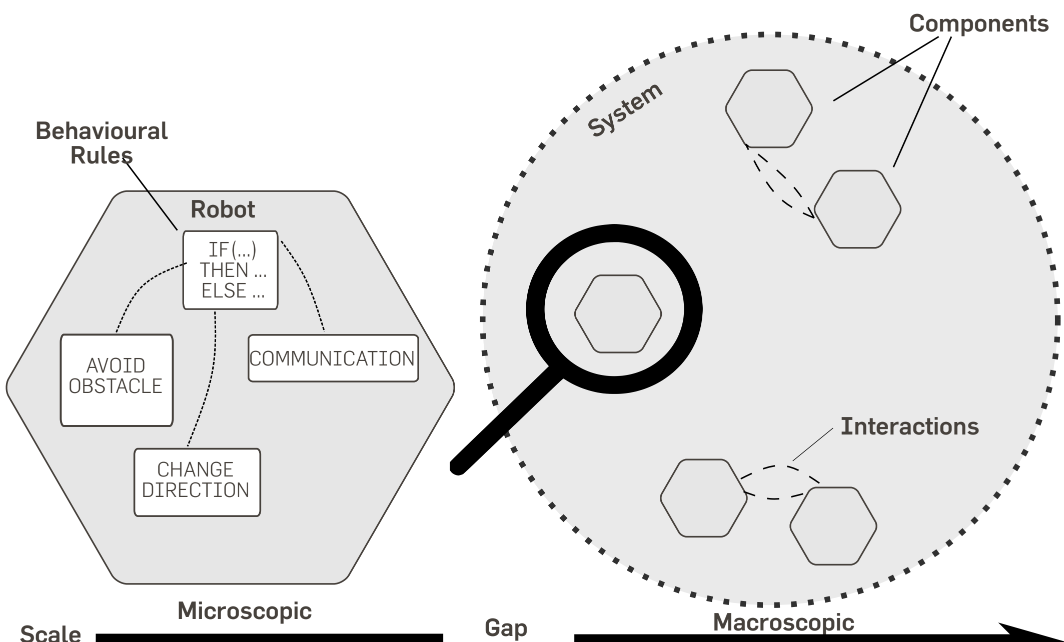


Figure 1: The gap between two levels of description of the same self-organizing system. On the left, the microscopic view: the focus is set on a single component of the system and its behavioral rules. On the right, the macroscopic view represents the full system made of numerous interacting components. The environment in which the system is embodied is not represented for the sake of clarity (**campo2011design**)

Part II

Organization of the project

In this practical, you will learn and implement basic swarm behaviours using the Cellulo robots. A computer with Linux (Ubuntu 18.04) will be put at your disposition with everything already installed (ROS, Cellulo API ...). You will be using small tabletop robots developed and produced at the CHILI Lab of EPFL, the Cellulo robots (Özgür et al. 2017a).

Cellulo Robots

Cellulo is a handheld, small-sized, mobile, haptic robot that operates on printed paper sheets (Özgür et al. 2017b). These graspable robots can be used as an interface for interacting with many virtual objects that reside on a plane.

The robots are designed to be simple to operate; all robots are connected wirelessly to a PC device that runs the activity and game logic. The current robots includes: self-localization on the activity sheet covered with a dot pattern; holonomic motion robust against human manipulation; six capacitive touch buttons (independently back-illuminated in full RGB) and wireless Bluetooth communication. The platform provides fast (>90Hz) and accurate localization (sub-mm) (Hostettler et al. 2016) of many robots which can be logged to record all the interactions during the game, such as user motion.

For the practical, you will be given 5 robots. Please make sure you charge them and turn them off after each session.

Schedule of the practical

In this practical, you will get familiar and will implement some classically known swarm robotic tasks. The additional document will guide you through the process. Your overall work on the two sessions will consist in 4 parts:
Part 1 Familiarisation with Cellulo and ROS enviroment
Part 2 Leader-Follower
Part 3 Aggregation
Part 4 Coverage
Some parts of the practical can be prepared in advance.

Report

In this practical, you will be mainly:

1. code the various swarm behaviors
2. discuss the effect of varying some parameters on the performance.

Therefore, for the report, we require you to submit (1) a zip file of your project (2) a report including a discussion of the results you observe and the answers to some questions. Make it concise. The suggested structure is just to answer each deliverable.

In particular the question you will have to answer in your report are marked in boxes:

> ➠ **Deliverable 1**

Yellow boxes indicate that the question can be prepared before coming to the lab session.

> ➠ **Deliverable 2**

Part III

Background and Installation

ROS

We will be using ROS (Robot Operating System) in this practical. Although it is not a mandatory prerequisite for the project, it is important to have an idea of what is ROS, and basic ROS concepts. To do that, please check `https://wiki.ros.org/ROS/Introduction` and `https://wiki.ros.org/ROS/Concepts`.

Installation

To reduce as much as we can the installation problems, we have created a VM(virtual machine) that has all what you need to have. Here are the steps to follow for the installation for the robotics practical:

1. Download a virtual machine program. We recommend using virtualbox (`https://www.virtualbox.org/`) but if you have vmware, it should work too.
2. Download the VM for this practical from `https://drive.google.com/file/d/1o2uft4OXtAoyHQ6y_Z8aJ-MEVqTCPhbA/view?usp=sharing`
3. Import the VM from the virtualbox and start it.
4. You should see a standard log in with user name tp9. The password is cellulotp.
5. To test it: in a terminal, you do:

   ```
   cd catkin_ws
   source devel/setup.bash
   roslaunch ros_cellulo_practical basic.launch
   ```

   Rviz should launch and you will see 1 robot. In a new terminal, you do:

   ```
   rostopic pub /cellulo_node_00_06_66_74_40_D1/setGoalVelocity geometry_msgs/Vector3 "x: 50.0 y: 0.0
       z: 0.0"
   ```

   You should see the robot moving.

Part IV

Software framework

As you can see from Figure 2, the Cellulo robots are actually all centrally connected. Each of the robots actually features a precise localization system. However, for the sake of this practical, we emulate a basic swarm robot that features a proximity sensor.

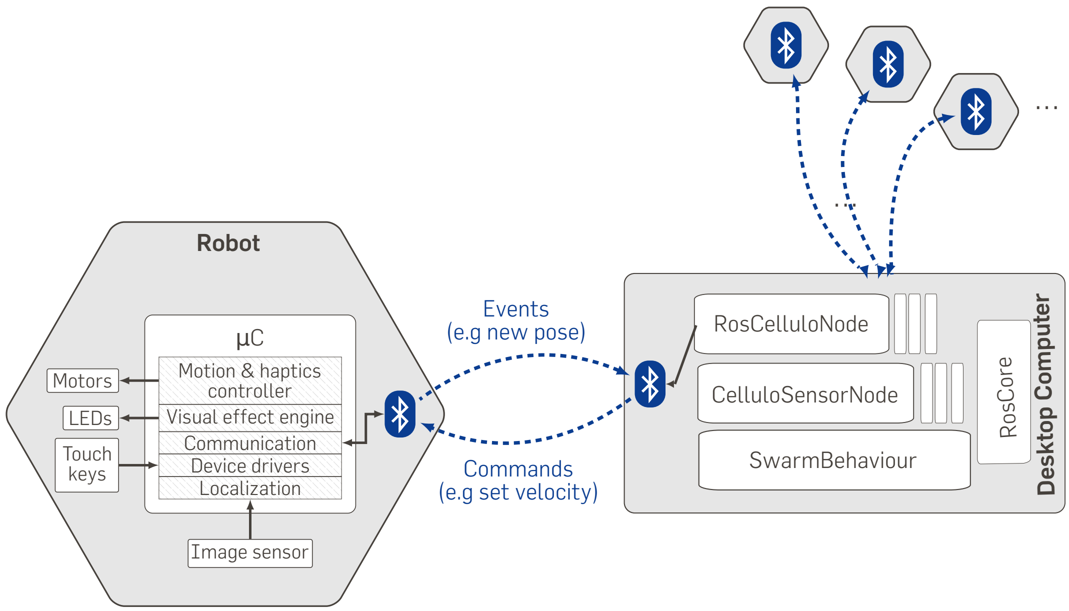


Figure 2: Cellulo ROS Software Architecture

ROS Cellulo Node

The ROS Cellulo node implements the ROS API for the Cellulo Robot. The different topics, publishers and subscribers are detailed in the `README.md` file.

Useful Subscribers:

- `cellulo_node_(mac_adr)/setGoalVelocity` sets the (x, y, θ) velocity of the robot
- `cellulo_node_(mac_adr)/clearTracking` clears the velocity buffer of the robot
- `cellulo_node_(mac_adr)/setVisualEffect` sets the LEDs color
- `cellulo_node_(mac_adr)/shutdown` turns off the robot
- `cellulo_node_(mac_adr)/reset` clears all buffers and settings of the robot

Useful Publishers:

- `cellulo_node_(mac_adr)/velocity`
- `cellulo_node_(mac_adr)/touchKey`
- `cellulo_node_(mac_adr)/longTouchKey`
- `cellulo_node_(mac_adr)/kidnapped`

ROS Cellulo Sensor Node

The sensor node has the role to emulate the presence of proximity sensors on the robot. This node has two main publishes and one subscriber:
The sensor message has 4 arguments:

- timestamp: indicating the time of the detection in ns.
- detected: indicating the number of detected objects.
- Distance: an array of 3D vectors where x is the difference in x direction, y is the difference in y direction, z is the Euclidean distance. Fig. 3 describes how the distances are calculated.

Publishers:
- `sensor_node_(mac_adr)/detectedRobots` publishes the distances to detected robots within a predefined detection threshold.
- `sensor_node_(mac_adr)/detectedObstacles` publishes the distances to detected obstables (including walls) within the predefined detection threshold.
Subscriber:
`/setThreshold` modifies the detection threshold.

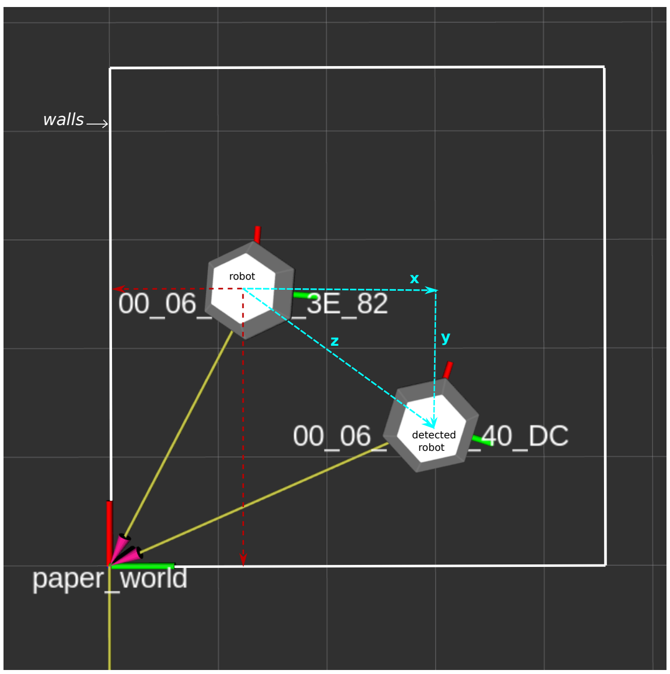


Figure 3: Sensor Description: The walls are considered as obstacles. Dotted red arrows indicated the distance from the walls (obstacles) and the dotted aqua lines indicates the distance from another detected robot.

References

Hostettler, Lukas et al. (2016). "Real-time high-accuracy 2D localization with structured patterns". In: *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, pp. 4536–4543.
Özgür, Ayberk et al. (2017a). "Cellulo: Versatile handheld robots for education". In: *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, pp. 119–127.
Özgür, Ayberk et al. (2017b). "Haptic-enabled handheld mobile robots: Design and analysis". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, pp. 2449–2461.