Behaviour-based control in ARGoS

Andrea Roli
andrea.roli@unibo.it
Dept. of Computer Science and Engineering (DISI)
Alma Mater Studiorum Università di Bologna

A.Y. 2018/2019

Motor schemas

Motor schemas are among the most used models for implementing behaviour-based control. In this lab session you will be asked to program a robot for a given task according to it.

Task

The robot must be able to find a light source and go towards it, while avoiding collisions with other objects, such as walls, boxes and other robots. The robot should reach the target as fast as possible and, once reached it, it should stay close to it (either standing or moving). For physical constraints, try not to apply velocities higher than 10—or at least do not exaggerate. The robot (a footbot) is equipped with light and proximity sensors.

Exercise 1: Motor schemas

Implement the robot control program in ARGoS on the basis of the motor schemas architecture. Before starting to code, define the basic behaviours and the corresponding potential fields.

Some suggestions:

- Try to think of the possible ways to implement a potential field and do not stop at the first one that comes to your mind. You will be surprise to discover that they are many, even for this simple task.

- Test your robot on different environments; for example, an arena with boxes placed randomly (vary the number of boxes), another arena with one or more walls between the robot and the light—the walls may also have thin doors. Try also to set the light at different intensities and heights—this latter parameter impacts the visibility of the target.
- Set the light sensor ray on, so as to be able to directly inspect whether the robot is seeing the light or not.

For vector operations, you can use the Lua module `vector.lua`, which you can load from your main program (the controller) via this instruction:

```
local vector = require "vector"
```

Functions can be called with dot notation.
E.g., `lenght = vector.vec2_length(v)`

Remember that light and proximity sensor readings in ARGoS–Lua return two fields, namely *value* and *angle*. Sensors are numbered from 1 to 24, counterclockwise. Angles of sensors from 1 to 12 are from 0 to (almost) π, whilst they have negative values for sensors 13 to 24 (from $-\pi$ to 0).

In usual implementations of this architecture, you just need to create and operate on vectors in polar coordinates, which are defined and assigned as follows:

```
v = {length = 0.0, angle = 0.0}
....
v.length = 2.1
v.angle = 1.2
```

Then you'll just need to sum the vectors (in polar coordinates) by using the function `vector.vec2_polar_sum(v1, v2)`.

To convert commands in the form of translational and angular velocities into differential actions (wheel velocities) you can make use of the following formulas:

- v: translational velocity, ω: angular velocity (expressed coeherently with ARGoS convention: positive angles towards the left, negative otherwise)

- v_l, v_r: linear left and right wheel velocity
- L: distance between the two wheels

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 \\ -1/L & 1/L \end{bmatrix} \begin{bmatrix} v_l \\ v_r \end{bmatrix}$$

$$\begin{bmatrix} v_l \\ v_r \end{bmatrix} = \begin{bmatrix} 1 & -L/2 \\ 1 & L/2 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

To know the distance between the two wheels in the footbot, you have to add as sensor the *differential steering* and use the following piece of code:

```
L = robot.wheels.axis_length
```

A possibility is to set a global variable `L` in the `init()` function.

Food for thought: comparison between subsumption architecture and motor schemas

- What are advantages and disadvantages of the architectures implemented? Try to state the main differences between them.
- Is there any that outperforms the other?
- What is the easiest to implement? To test?
- What is the best from the viewpoint of extendibility and composition of behaviours?
- Is there a way to design a hybrid architecture composed of both the subsumption architecture and motor schemas? Would it be a good idea to do it?