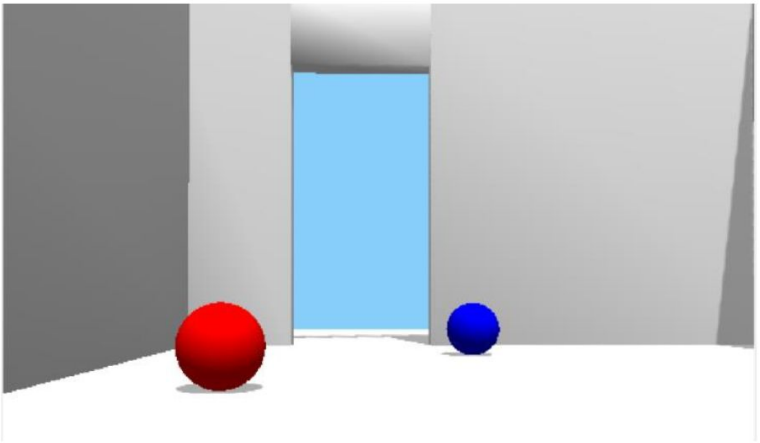
Final Project – Subsumption Architecture

"The Candy-Crazed Turtlebot"

Teacher: Kristinn R. Thórisson Teaching assistant: Arash Sheikhlar

Introduction. Remaining lab sessions will focus on this final project. You can do it either alone or in teams of up to 3. The main goal is to use the knowledge you gained in two previous sessions and combine it with a subsumption architecture of your own design for controlling the TurtleBot.



What a robot sees in the room may look like this image, where two candies with red and blue colors and a door can be seen.

Problem Description. The robot's main task is to move around in a room, find the candies and eat them one by one in a limited time. After the exploration of each room, the robot goes to another room and does the same job. The main goal of the robot is to eat the maximum number of candies by the end of its mission, after around 4 minutes of running. Use the subsumption architecture as the key method to control the robot. Here is a suggested outline:

- Level-0: Roam and explore: E.g. random movement in the room - move, turn, move, turn.
- Level-1: Find and eat candy: If there is candy in the camera image, center on it, move towards it and hit it ("eat" it).
- Level-2: Move to another room: Find a door and go through it.
- Level-3: Avoid bumping into things.

Hints for implementation.

Level-0. The simplest solution here might be some hand-selected set of angle values (e.g. 20, 30, 50, 90, -20, -30, -50, -90) and seconds (e.g. 2 seconds, 3 secs and 5 secs) that get selected at random and applied to the movements of the robot in an alternating pattern (move, rotate, move, rotate … etc.); you might get e.g. move forward for 2 secs, rotate 30, move forward for 5 seconds, rotate -90, move for 2 secs, rotate -50, move for 3 seconds … etc. Note that within each Level there can be more than one method for selecting such parametric values, and even computing from scratch - based in some way on sensor values.

Level-1. The main input for this Level comes from the camera. When one or more candies are in the frame, the simplest solution is to select a particular candy, then rotate Turtlebot until that candy is in the center of the camera, then move towards it. (If there are many candies on the floor, the robot **could** use memory to keep track of them, and navigate towards each of them in turn, but that is not mandatory.)

Level-2. To decide to go to the next room, you may consider a "perceptor" that sets a timer as soon as there are no candies in the camera view. The timer has a threshold (e.g. 10 seconds), that, when reached, sends a TRUE signal to Level-2. When this true signal is received in Level-2 this level subsumes Levels 0 and 1 (it will still avoid bumping into things because Level-3 is above it). [*Please note: A more detailed description of the doorway problem+solution will be provided by the next lab session*] To find a doorway and go through it, consider that the doors have a rectangular shape. So, you can compute a binary black and white image through thresholding, in which a doorway is a different value than the wall(s). By taking the center horizontal line of such an image - all the pixels across the middle - and looking for wall (e.g. 1) and opening (i.e. 0), rotating the robot to put the opening at the center of the image, then go forward, may allow the robot to go through the opening if the robot is reasonably orthogonally position relative to the doorway.

Level-3. Consider using the laser rangefinder (distance straight ahead) as the main input. Simplest response is to do an arbitrary rotation in either direction (e.g. between 40 and 60 degrees). A bit more sophisticated would be to compute an angle to rotate based on more detail from the rangefinder.

Hand-in. You have to hand in a PDF report along with the m.file codes and functions you use. In the report you should:

a. Provide a detailed diagram of the architecture, with a short (but sufficient) explanation.
b. Give a short summary of your solution in your overall own words (500 words max).
c. Answer the following questions (either through full or partial implementation, or by analysis):

i. Could Layer 1 and Layer 2 be swapped? What would that mean for the implicit goal of the robot?
ii. How could the robot perceive that it has passed through the door and it is in a new room, if it does not immediately see new candies?
iii. Can you think of a simple solution to prevent the robot from going back through the same doorway it just came through, in the case where there are no candies left on the floor on either side of the doorway?
iv. Assuming that the red candies have negative points (-10 for each) and blue candies have positive points (+10 for each), how could your architecture be modified to make the robot avoid red candy and approach only the blue?