Lab report 3

Benedetta Pacilli
ID: 0001136705
email: benedetta.pacilli@studio.unibo.it

This program implements a motor schema architecture to perform both obstacle avoidance and phototaxis behaviors.

- `step()`: It is the core of the program. It increments the step count, checks for light sources via the `check_lighting()` function, calculates the final velocity vector considering both obstacle avoidance and phototaxis, and sets the wheel velocities accordingly.
- `check_lighting()`: Checks if any of the light sensors detects a light intensity above the defined threshold. In case any light is detected it returns `true`, otherwise it returns `false`. The returned value is intended to be passed both to `obstacle_avoidance(light_flag)` and to `phototaxis(light_flag)` so that in case the robot is near the light source the final resulting vector makes the robot stop.
- `obstacle_avoidance(light_flag)`: It accepts the parameter `light_flag`, which, if true, indicates that the robot has reached a light source, it can stop and thus doesn't need to perform obstacle avoidance. Otherwise, the function collects the readings from the proximity sensors to find the maximum sensor value, its corresponding angle, and the index of the sensor with the maximum value. Then, the velocity vector is calculated based on the proximity data. The length of the vector is set to the maximum velocity multiplied by the maximum sensor value, meaning that the closer it gets to an obstacle the faster it will become) and sets the angle so that if the maximum proximity value sensed was on the left (index between 1 and 12), the robot steers right (`-90°`) and if it was on the right (index between 13 and 24), the robot steers left (`+90°`) .
- `phototaxis(light_flag)`: It guides the robot towards light sources detected by its light sensors. It accepts the parameter `light_flag`. If `light_flag` is `true`, meaning the robot is near enough the light source, the function returns the vector `{length = 0, angle = 0}` indicating that the robot doesn't need to move towards or away from the source. If `light_flag` is `false`, the function proceeds to sense light using the light sensors on the robot. It iterates through the light sensor readings to find the sensor with the highest light intensity (`max_value`) and stores its angle (`max_angle`). If light was detected, maximum light intensity (`> 0.01`), it calculates the length velocity vector `v` as the maximum velocity (`MAX_VELOCITY`) divided by the maximum light intensity. This enables the robot to move faster the farthest it is from the light source and slow down the closer it gets. Resulting velocities exceeding `15` are limited to this value, through the `limit_velocity()` function. The angle remains the same as the angle of the sensor with the highest light intensity reading. If no light was detected, it generates a

random velocity vector `v` with a random length between `5` and `MAX_VELOCITY`, and a random angle between `-π` and `π`.