Intelligent Robotic Systems
-
Fourth Lab Activity

Simone Ceredi

12/04/2024

Objective

The task for the robot(s) in the fourth lab activity is to find a light source and move towards it, while avoiding collisions with walls, boxes, and other robots. The robot should reach the target as quickly as possible and, once reached it, it should stay close to it (either standing or moving).

Constraints

- The wheel velocity must not exceed 15 (i.e., 15^{-2} m/s) due to physical constraints.
- The robot (a footbot) is equipped with only light, proximity, differential steering and positioning sensors.
- The positioning sensor must only be used for evaluation purposes.
- The control scheme must follow the motor schemas architecture, with an emphasis on the perceptual schemas.

Additional constraints

- Values equal to 1 recorded by the footbot proximity sensors are considered collisions.

Solution implemented

Perceptual schemas

Two perceptual schemas are used in the implementation. The first schema utilizes all 24 proximity sensors, considering the values and angles returned by the sensors as vectors and then summing them together. The second schema utilizes all 24 light sensors, summing them as vectors in the same way as the proximity sensor schema.

Vector fields

Four vector fields were created to utilize the previously described perceptual schemas: a *uniform field*, an *attractive field*, a *repulsive field*, and a *tangential field*.

- The *attractive field* is used in combination with the light sensors perceptual schema in order to guide the robot toward the light.
- The *repulsive field* is used in combination with the proximity sensors perceptual schema to allow the robot to avoid the various obstacles it encounters on its path.
- The *uniform field* is used to move the robot forward when the robot is too far away from the light. Its strength depends on the strength calculated by the lights perceptual field. When far away from the light it equals 0.5, it then decreases linearly, up to a minimum of 0, as the light strength increases.
- The *tangential field* is used to allow the robot to circumnavigate the obstacles it encounters. It can be useful as usually the obstacles tend to hide away the light, so by simply avoiding them the robot could steer away from the objective.

These fields are then used to calculate the vector that moves the robot. For this task the vectors calculated as shown before are summed, the resulting vector is then translated into the values to assign at the robot wheels.

Evaluation

After running the simulation 1000 times with different arenas, the results (euclidean distance from the light) were the following:

Count	**Mean**	**Std**	**Min**	**25%**	**50%**	**75%**	**Max**
1000.0	0.36471	1.04403	0.006363	0.007296	0.007519	0.007792	5.235137

Table 1: Summary Statistics

The box plot of the results can be seen in fig. 1. The results look quite good as shown by both the 75 percentile having a very low value and the box plot. Showing that the robot most of the times arrives at it's destination without issues. While in the subsumption architecture implementation it was evident that the robot struggled with funnel-like obstacles, this implementation seems to have resolved that issue. For this reason, it is difficult to determine what happens in cases where the robot does not reach the objective. A comparison between this implementation and the subsumption architecture shows that using motor schemas reduces the likelihood of the robot getting stuck, as demonstrated by both the box plot fig. 1 and the histogram fig. 2.

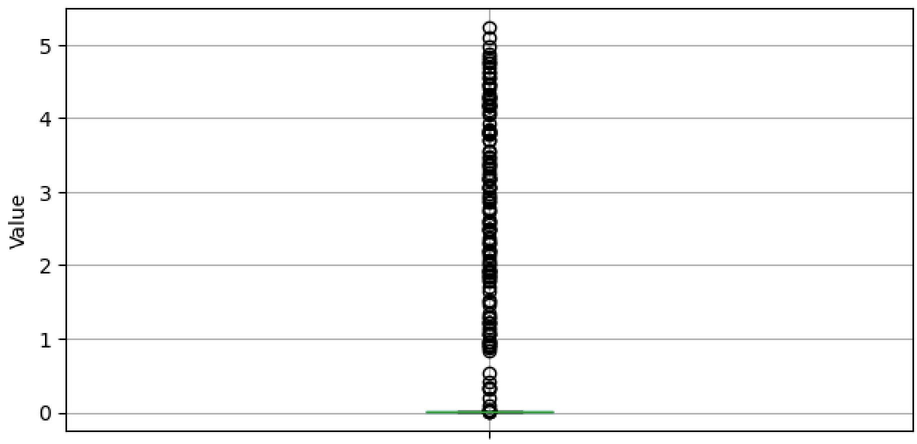

Figure 1: Box plot of the results

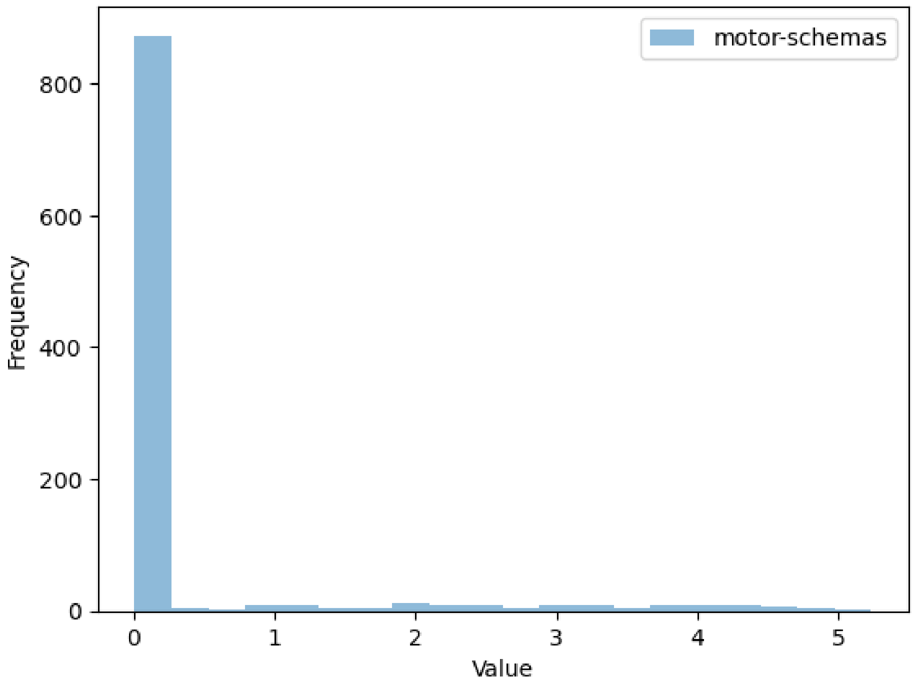

Figure 2: Histogram of the results

Problems faced

As shown in the evaluation section, the motor schemas implementation suffers less from the problems that affect both the finite state automaton and subsumption architecture. However, the biggest issue with the motor schemas implementation is debugging. In the previous implementation, it was fairly straightforward to use the onboard LEDs or logging to understand the robot's actions at any given time. However, in motor schemas, every field and every perceptual schema is active simultaneously, making debugging very challenging.

While the task was trivial to implement, it is easy to see how this kind of implementation is hard to scale, as even simple tasks such as stopping the robot are not trivial to implement. For this reason, I believe that for more complex behaviours, a *probabilistic finite state machine* could be beneficial. In such a system, states that can benefit from motor schemas architecture would be implemented using that approach, while other states could use behaviour trees, subsumption architecture, or any other control schema suited to the task. This approach would facilitate debugging and allow the use of different architectures for tasks that are difficult to implement with motor schemas.