ALMA MATER STUDIORUM – UNIVERSITY OF BOLOGNA
CAMPUS OF CESENA

Engineering and Architecture Faculty
Master's Degree in Engineering and IT

ROBOT EXPLORATION

Subject
INTELLIGENT ROBOTIC SYSTEMS

Presented by
GABRIELE GUERRINI

Accademic Year 2019 – 2020

Table of Contents

Chapter 1

Project goal

Chapter 2

State of the art

An overall of four different well-known tasks can be detected in the problem, namely:

- **Arena exploration**: the robots must explore the arena to find the landmarks.
- **Clustering**: the robots must create cluster near any landmark so it can be considered explored.
- **Phototaxis**: used to allow the robots to easily come back to the base.
- **Obstacle avoidance**: general behaviour to be executed to prevent any collision (between robots as well as between robots and obstacles/walls).

It follows a brief description of the state of the art for each type of task.

2.1 Arena exploration

The exploration of the area is a well-known task in the literature, and lots of algorithms and methodologies have been studied. However, when talking about swarm robotics, the main adopted method is the *random walk* approach. This is mainly due to the usage of off-the-shelf robots that own limited individual abilities (low processing power, local sensing etc.). Nevertheless, the usage of a swarm of robots for exploration tasks is still widely applied thanks to the advantages it entails respect to the usage of a single robot (flexibility, robustness, scalability).

There are few different common sense implementations of a random walk task [1], namely:

- Brownian motion
- Lèvy Flight
- Lèvy Taxis
- Correlated radom walk
- Ballistic motion

The main idea is to randomly choose a direction and go straight till a new one is selected.
All the aforementioned models share the same underlying mathematical model, and the different tuning of the parameters provides different behaviours and exploration capabilities. Only one exception can be identified, that is the ballistic motion, where the two parameters *step length* (μ) and *turning angle* (ρ) are not provided and/or limited.

Further from the above basic approaches, new solution have been probed to overcome the limits they impose. Indeed, there are two main problem that emerge, that is:

- Execution of repeated exploration of the same point.
- Exploration does not scale with arena widening. Ineed, just zona near the starting position will be quite well explored.

New approaches have been studied to overcome the aboe issues. The example reported in [2] improves the random walk by considering relative distribution of robots among the arena so that each zone can be equally explored by a even quantity of robots.

Once the robots are spread over the arena, different algorithm can be applied to map it (e.g., *GMapping* that produces a two-dimensional occupancy grid of the environment).

2.2 Clustering

2.3 Obstacle avoidance and phototaxis

Obstacle avoidance and phototaxis tasks are two trivial tasks that represent a very basic idea and so, considered their simplicity, there are not so many ad hoc techniques and algorithms that can be applied. The implementation

strongly depends on the adopted architecture (e.g., subsumption against motor schema).
For example, when exploiting the motor schema patterns, phototaxis and obstacle avoidance may be modeled using attractive and tangential potential fields respectively.
In the end, as well as the classical approaches, novel solutions have been explored by combining genetic algorithms and neural networks (as reported in [3]).

Chapter 3

Solution development

3.1 Total behaviour

The behavior of each robot is modeled using a non-deterministic finite state automata (*NFA*)(see figure 3.1). Such an architecture has been chosen for the robot since its behavior is not trivial and more tasks must be executed at different times. By using a FSA, it is easy to explicitly express activities, conditions and non-deterministic transitions. Moreover, each state can be developed separately from the others, and the most suitable solution can be implemented case by case.
The robot starts in the “resting” state. A non-deterministic transition makes the robot leave the base and start the exploration with a probability $P1$.
When exploring, the robot looks for any close landmark and, if it is the case, it joins the cluster around such landmark with a probability $P2$ and enters in “waiting for cluster to complete” state. Moreover, the robot may quit the exploration task at any moment and return to the base with a probability $P3$. In the latter casuistry, the robot enters in the “returning to base” state.
When a cluster is completed by reaching the required number of robots N, the robot enters in the “reconnaissance” state. Such a state conceptually correspond to the phase during which the robots in the cluster should explore the area near the landmark. Such a task is not executed in this project for simplicity, but few different algorithms have been created to let a set of robots to explore and map a given area.
The robot ends the reconnaissance task after t_1 seconds, and it returns to the base.
When returning to the base, the robot executes a phototaxis task. The robot enters in the “resting” state once it is detected that it successfully returned to the base.
In the end, the robot may leave the cluster before it is completed with a

probability $P4$. In this case, it enters in the “biased exploration” state, that is a state where the robot executes the regular exploration tasks but with inhibited stimuli from landmarks. After t_2 seconds it returns in the normal exploration behaviour. Such a intermediate state is required so that the robot may step away from the landmark (otherwise, even though it is a probabilistic transition, it would join the cluster again in most of the cases). TODO: required?

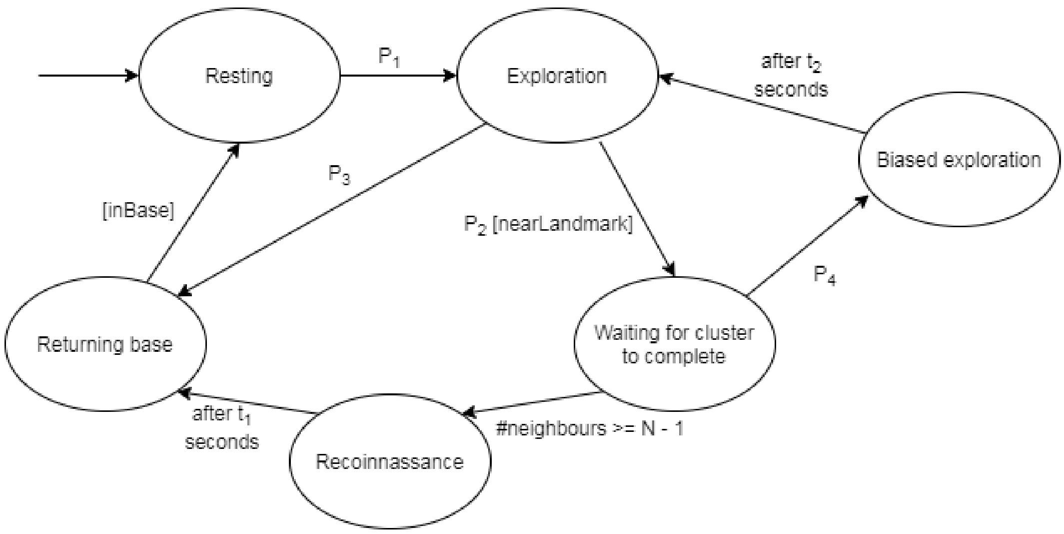

Figure 3.1: *NFA representing the behaviour of the robot. P_i represent a non-deterministic transition that happens with such probability.*

The implementation of the FSA is executed by using a “states“ table to store the code be executed in each state. The variable “current_state” represents the current active state, while transition are executed by checking simple“if-then-else” conditions.
In the end, a variable “t” stores the time spent by the robot in the current state.

3.2 Resting

The robot in this state just checks if a transition to the exploration one should occur. The following function has been used to model the probability:

$$p = tanh((t - Shift)/Patience) + 1$$

The idea is to create a probability that is directly proportional to the time spent in the current state using a non linear relation. Moreover, the probability should grow very slowly at the beginning so that the robot will remain in the resting state for a while.
The function $tanh(x)$ is the basic function used to represent such relation[1] (figure 3.2). The $Shift$ value is constant (500) so we can take the slice of the function having up concavity. The $Patience$ value is a parameter used to tone all values down. In the end, the +1 factor let the function have positive values. Table 3.1 reports few reference values of the function.

t	p
0	0.013
10	0.015
50	0.022
100	0.036
500	1

Table 3.1: *Few example values of the function used to model dependency with elapsed time using a non linear relation.*

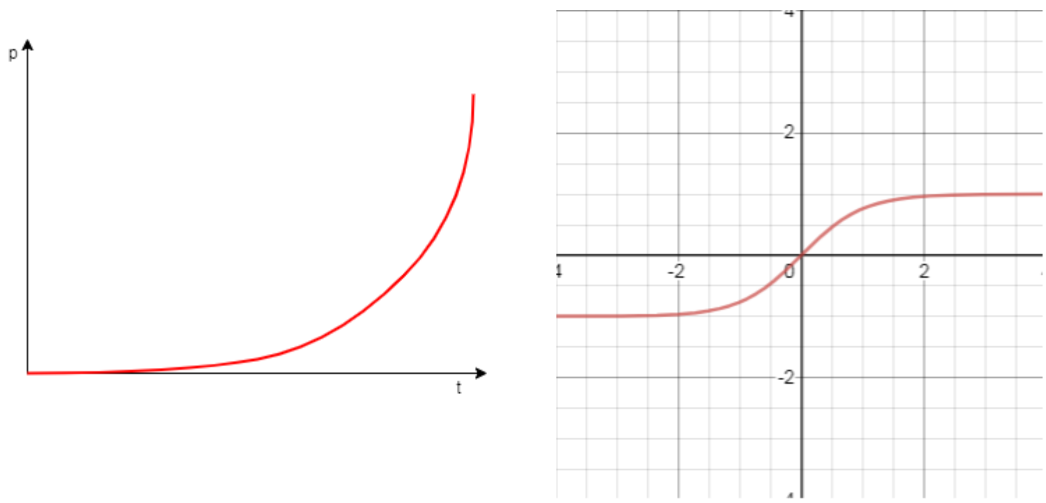

Figure 3.2: *A template of the expected function (left) and the regular tanh function (right).*

[1] A reasonable alternative would be the exponential function, but it requires the tuning few parameters so that the "tail" of the function take reasonable values

3.3 Exploration

3.4 Clustering

3.5 Returning to base

Chapter 4

Performance evaluation

Chapter 5

Conclusions

Bibliography

[1] https://link.springer.com/chapter/10.1007/978-3-030-25332-5_19

[2] https://www.hindawi.com/journals/jr/2019/6914212/

[3] http://anderslyhnechristensen.com/pubs/alchristensen_alifex_holeavoidanceandphototaxis_2006.pdf