Final Project
AI and Robotics (AIR) - 236609

February 19, 2023

This final project is comprised of 2 tasks: the first is a practical assignment, and the second is a theoretical assignment. We provide a complete specification of the requirements for the first task, while you will decide which extension of the basic setting you want to focus on for the second task.

1 Part 1: Practical Assignment

In the previous assignment, you had to plan the motions of a robot that needed to examine objects in a room using a Lidar. Here, we extend this setting to account for a robot with a complex task. As before, you have a robot (turtlebot3) that is placed in a fully mapped room (you will only receive the map at execution time). In addition to walls, there are workstations that are distributed in the environment in known locations. The shape of each workstation is a block with fixed dimensions and each station is associated with specific actions that can be performed from specific configurations around it. Thus, each workstation is associated with a set of activities that can be performed at that station and an *affordance function* that specifies the configurations from which a given activity can be performed for each workstation (we are abstracting the actual execution of the activities such that an activity can be executed if the robot is in a valid configuration). Note that the affordance function only considers the workstation, and not the robot or any other obstacles around it.

In addition to the description of the environment, you will receive an unordered set of activity sequences that need to be performed. To complete an activity, it is necessary to go to a relevant workstation and be in a configuration from which the activity can be performed. Each motion and activity are associated with a cost that corresponds to its execution time and its risk, which corresponds to the agent's proximity to walls and objects (excluding machines) when performing an activity (as defined below). An activity sequence needs to be performed in the specified order. Each activity sequence is associated with a reward that is achieved when it is completed.

For example, a possible input will include a map of the environment and the following lists. The first list specifies the workstations and the list of activities that can be performed in each workstation.

Work Station	Available Activities
WS1	ACT1, ACT2, PU-A, PL-A, PU-B, PL-B
WS2	ACT3, ACT2, PU-C, PL-C, PU-B, PL-B
WS3	ACT1, ACT4, PU-A, PL-A

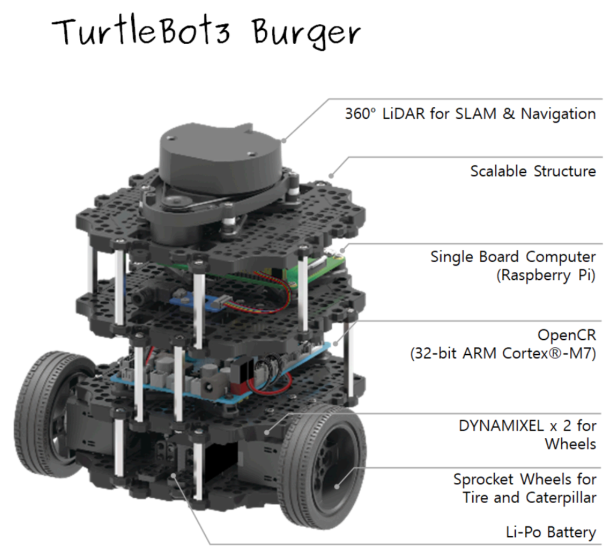

Figure 1

The second list specifies the task: the set of activity sequences that can be performed and their associated rewards.

Task List	Reward
ACT1->ACT2	20
ACT2->ACT3	30
ACT1->ACT3->ACT2	50
ACT1->ACT4	10

As an example, in Figure 2 we see 3 workstations colored blue, green, and red, and the configurations from which activity 'A' can be applied in each workstation. As seen in the given list, each task i has a reward r_i. The total reward R is the sum of rewards r_i received after completing different tasks.

$$R = \sum_i r_i$$

The cost of a path C is determined by the time T it takes to follow the path and the cost map M. M is constant and is determined by the proximity to the walls and obstacles. If at time t the location of the agent is known to be x_t on the map, then a cost $M(x_t)$ can be inferred according to the map. The cost of a path is the accumulative cost over the path calculated each second $t = 0, 1, 2, ...T$. The cost of the path can be computed as

$$C = T + \sum_t^T M(x_t)$$

Your assignment is to find a plan that will maximize reward (i.e., $R - C$) within the allotted time-bound (which will be known in execution time).

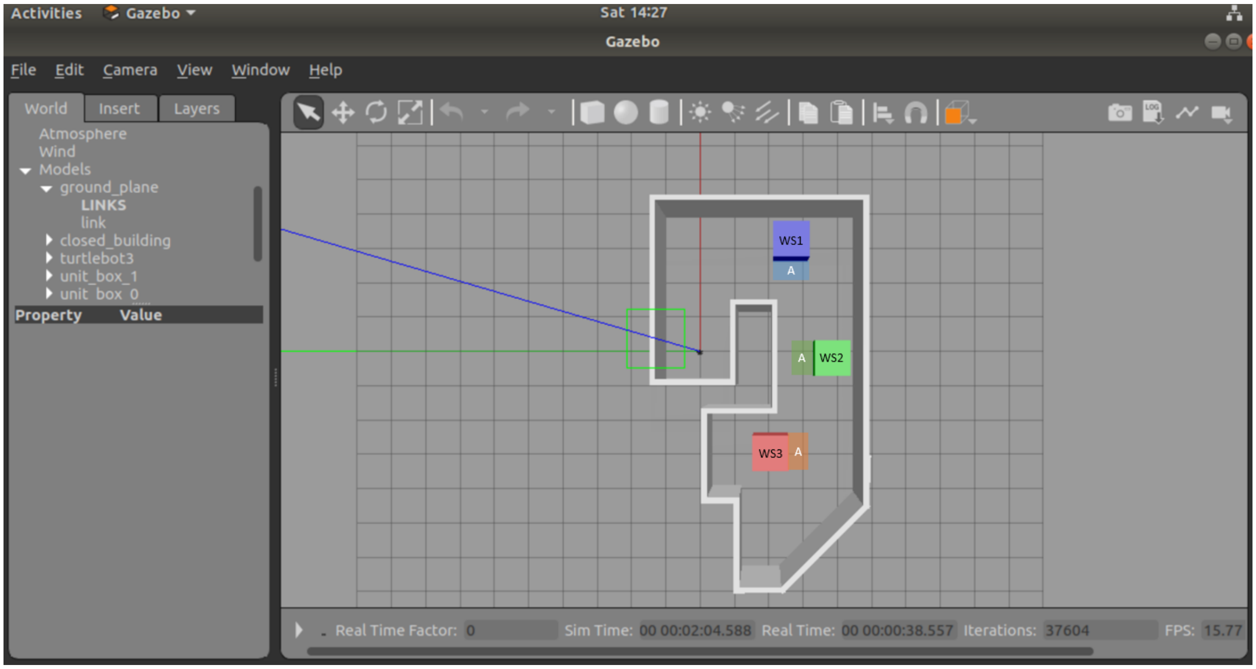

Figure 2

1.1 Extra considerations:

- The duration of all activities is fixed and uniform.
- After the robot performs a PU-X (pickup an object of type X) action and before it performs a PL-X (place an object of type X) action its speed is limited to $0.1[m/s]$ (this needs to be considered when planning its movement).
- The robot can only carry one item at a time.
- Two consecutive tasks cannot be performed on the same workstation.
- Each Task can only be performed once.

1.2 Technical details

- We will be using Ubuntu18 with ROS melodic.
- The repo for the exercise can be downloaded from `https://github.com/sarah-keren/MRS_236609`
- You need to use the Turtlebot3.
- Our code should be installed under the src folder of the ROS project you created.
- You can use all the packages we have seen so far, including move_base.
- You will need to complete assignment3.py.

- You will be allowed to localize your robot, using the relevant ROS command.
- To complete the practical task of the assignment you must run your solution both in simulation and on real robots.

1.3 Deliverables

For this part, you are required to submit 2 documents: A single python module containing your code named assighnment3.py, and a PDF file generated using the latex template ”Lab Report.tex” provided at `https://www.overleaf.com/read/mkphtcyjbjpc`.

1.4 Evaluation

Your performance in this part will be evaluated by the accumulated reward achieved by running your plan.

1.5 Pay attention

- **Do not !** include ANY package that is not already installed by default. If you want a package that is not installed - you need our approval.
- You need to make sure your code is python2 compatible.
- The lab report must not exceed 5 pages.
- The maximal speed of the turtlebot is 0.22 m/s and 2.84 rad/sec.
- This may not be the final description of the assignment. As you start working on it, you will have questions, and we may need to slightly update the description accordingly.

2 Part 2: Integrating Task and Motion Planning and Accounting for Uncertainty

In the first part of the project, the robot had a complex task but was able to decouple its motion planning and task planning phases. In the second part of the project, you are required to consider an extension of this setting by considering some form of uncertainty or motion and task planning integration we discussed in class. Example extensions include:

1. Localization uncertainty - the agent does not know its exact location
2. Map uncertainty - the map is not fully given a-priori
3. Affordance uncertainty - the affordance function is not given and needs to be learned.
4. Multi-modal planning
5. Hierarchical ITMP

6. etc ...

You are required to present a problem statement (that includes a formal account of the uncertainty or additional aspect you chose to model) and a full description of a solution approach. Your project must fulfill the following criteria:

- You need to provide an in-depth investigation of one of the key ideas discussed in class.
- The solution to the problem you pose should be non-trivial.
- The project must have an appropriate workload for the number of members in your group.

We do not require implementing your idea, but you will need to provide a full account of how your suggestion maps to the real robotic setting of part 1: how it would be modeled, how your solution would be implemented, what has been done in the past to solve the problem you aim to tackle, and how would you evaluate your suggested solution against existing approaches.

NOTE! You need to get approval for your chosen task from the course staff before starting to work on it.

2.1 Deliverables

For this part, you are required to submit 1 document: A PDF file generated using the latex template "Final Project.tex" provided at `https://www.overleaf.com/read/mkphtcyjbjpc`. Note that there is a limit of 5 pages.

Good luck and don't forget to enjoy the process!