# Airo2 - Assignment II

## Content

```
AiroII_2/
        |
        visit_domain/   , task planning files
            |
            dom1.pddl       , pddl domain description
            dom1_FD.pddl    , pddl edited domain description
            landmark.txt    , beacons poses
            prob1.pddl      , pddl problem description
            prob1_FD.pddl   , pddl edited problem description
            region_poses    , association region-waypoints
            waypoint.txt    , waypoint poses
        |
        visit_module/   , motion planning files
            |
            src /        , motion planning scripts
                |
                CMakeLists.txt      , the cmake file
                ExternalSolver.cpp  , definition of the interface used for the
semantic
                                    attachments
                ExternalSolver.h    , declaration of the interface used for the
semantic
                                    attachments
                VisitSolver.cpp     , definition of the class to solve the
motion planning
                                    step based on EKF
                VisitSolver.h       , declaration of the class to solve the
motion planning
                                    step based on EKF
                buildInstruction.txt, script used to build the library
                main.cpp            , main using the solver
            |
            build /
                  |
                  Built files       , built files

        |
        .gitignore  , ignored files list
        README.md   , this file
        README.pdf   , this file, but more fancy
```

---

## Problem description

This repository contains a possible solution for the Second Assignment of the course in Artificial Intelligence for Robotics II held at the University of Genoa. The goal of the assignment is to implement a  PDDL 2.1 domain for the motion of a mobile holonomic robot that makes use of **semantic attachments** to evaluate the robot motion in a constrained 2D environment.  The

robot is considered as holonomic and has to reach specified waypoints minimizing a certain user-defined cost function. The robot motion between the waypoints was described using simplified **Extended Kalman Filter (EKF)** model and the robot was supposed equipped with a sensor capable of detecting landmarks that lie within a certain distance from the robot itself. The position of both the waypoints and the landmarks was assigned. The cost function used to solve the problem was defined as the arithmetic sum of a simplified travelled distance between the two closest waypoints and the weighted trace of the noise covariance matrix used in the EKF algorithm.

# Task Planning

The planning problem proposed in this assignment includes the definition of the optimal plan for a moving robot supposed to reach four fixed waypoints. The planner adopted in this task is the [popf-tif][2], that was selected for its capability to deal with temporal planning task and semantic attachments. In this specific case, the dedicated infrastructure needed to update the semantic attachments was provided by the instructors and further details can be found in [Thomas et al.][1]. In brief, every time that the a PDDL 2.1 keyword *increase* is encountered within an action, the cpp external solver is called to parse the keyword and, if needed, it can interpret the step information to update the value of the variable to increase.

# Motion Planning

The computation of the motion part as well as of the cost function is performed by an External Solver, through the library `libVisits.so`. Within this library, the core of the computation is nested in the `VisitSolver` class and more in detail in its `localize` method. This method, in fact, is dedicated to the implementation of the EKF for the definition of the robot motion within the predefined environment every time that a *localize* durative action is invoked by the planner. More in detail, every time that a *localize* action in between the points `(from, to)` is detected, the algorithm performs a fictitious movement and localization step with an EKF along the straight path connecting the origin and the end of the movement.
To reproduce the holonomic robot movement in the 2D space, the path is split into a number segments proportional to the acquisition rate of an imaginary sensor embedded on the robot capable of detecting the landmarks that lies within a specified distance from the robot itself.
For each step, the robot state `X_k` and its covariance matrix `P_k` are predicted as per the EKF algorithm taking into account the robot odometry and its noise. Furthermore, if the robot is close enough to any of the landmarks scattered across the mark the `X_k` and `P_k` predictions are updated as per the EKF algorithm.

## Sensitivity study and model parameters

The EKF algorithm here implemented relies on a series of parameters typically related to the robot and its sensors specification . In this specific case, since no details were given on the possible sensors or type of robot used, some values have been specified in a reasonable way while other have been changed to perform a sensitivity study.
**Fixed parameter:**

- **robot_vel = 0.2 :** The robot linear velocity.
- **odometry_freq = 10** The sampling rate of the imaginary robot sensor for detecting the landmarks.
- **odom_noise= 0.01** The $\sigma_{odom}$ of the odometry noise added each step to the robot state.
- **cov_matr_noise= 0.2** The starting noise for each localization step.

- **trace_weight = 50** The weight given to the $trace(P_k)$ in the cost function. This values was detected to obtain an equilibrium in between the two components of the function, hence defined as:

$$cost = d(start, goal) + 50 * trace(P_{(start,goal)})$$

where the notation used for *P* indicates the covariance matrix obtained by running the EKF from the starting to the goal configuration while *d* is an estimation of the effective distance travelled by the robot to reach the goal from the starting point.

**Sensitivity parameters:**

- **landmarks_th :** The distance at which the robot detects a landmarks. It is noteworthy that, if the robot detects more than one landmark the update state is performed on the predicted state using each landmark independently.
- **meas_noise :** The $\sigma_{meas}$: uncertainty on the "measured" landmark distance.

# Prerequisites

In order to correctly run scripts present in this repository the planning engine popf-tif, need to be installed together with this package. Moreover, this package uses Armadillo tools to optimize the matrix computations, and it has to be installed and all its dependencies correctly configured.

**Note:** The whole assignment has been developed using the docker image here attached: hypothe/ai4ro2_2:priv. For an easy execution of this package the use of this container is suggested.

# Building and Running

In order to correctly build the *semantic attachment* library the following command has to be executed from the *src* folder of the repository:

```
    ~/../AiroII_2/visits_module/src$ ./buildInstructions.txt
```

After that the building process is completed, the planner can be executed with

```
    ~/.../AiroII_2/visits_domain$ ~/..path_to../popf3-clp -x -n -t10 dom1.pddl
prob1.pddl \
                            ~/..path_to../visits_module/build/libVisits.so
region_poses
```

The `-x` flag is used to inform the planner that semantic attachment will be updated using the external solver criteria while the `-n` flag is used instead to enable the *anytime* planning modality, which means that the planner searches for improved solutions until it has exhausted the search space or is interrupted. A time bound of 15 seconds has been then specified using the flag `-t5` to limit the research of the optimal TMP sequence.

# Testing

The `region_poses`, `waypoint.txt`, and `landmark.txt` files provided by the instructors describe the planning environment developed for this task but they can be modified to further more complex scenarios.
A map of the 2D space used in this task is show in Figure 1.

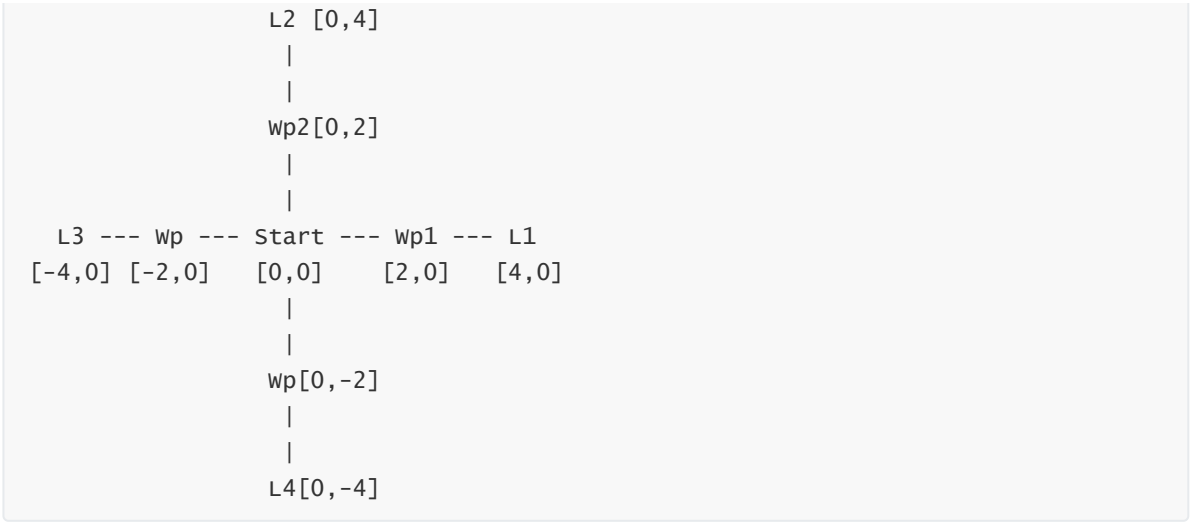


Figure 1: Simplified map where *Wp* and *L* stand for waypoint and landmark respectively.

## Results and sensitivity study

In this section the results in terms of cost function and action sequence are presented.
Two different type of sensitivity study have been performed for this task:

**Sensitivity to sensor range of action:**

- Three different values of range of actions were investigated in this study [0.25, 3.0, 10.0] m while the sensor error was kept constant to its intermediate value of 0.05 m. This choice was done to investigate the following relevant condition:
  - "blind robot" (0.25 m): The robot is capable to detect maximum one landmark when it lies very close to the robot. Due to the map geometry, the robot can not detect any landmark along is movements.
  - "awake robot" (3.0 m): The robot can detect one landmark for most of its movement.
  - "omniscient robot" (10.0 m): The robot detects all the landmarks during all its movements.

Table 1 shows the obtained results for these simulations in which is clear, as expected, that by increasing the knowledge of the surrounding world, the robot reduces through the EKF algorithm the uncertainty in its movements.

| landmarks_th | 0.5 | 3.0 | 10.0 |
|---|---|---|---|
| States evaluated | 6875 | 1145 | 2755 |
| Cost | 40.5 | 13.2 | 10.5 |
| External Solver | 2.8 | 0.6 | 2.0 |
| Time | 5.0 | 1.1 | 3.0 |

Table 1: Range of action sensitivity results for *meas_noise = 0.05 m*.

**Sensitivity to sensor error:**

- Three different values of the sensor error were investigated [0, 0.05, 0.1] m while the sensor range of action was kept constant to its intermediate value of 2.5 m. This values were selected to investigate the effect of a progressive reduction in the sensor accuracy that is expected to produce an increase in the cost function. The obtained results are shown in Table 2.

| meas_noise | 0.0| 0.05| 0.1|
| --- | --- | --- | --- |
| States evaluated | 1170| 1170| 1210|
| Cost | 12.0| 13.3| 14.1|
| External Solver | 0.6| 0.6| 0.6|
| Time | 1.0| 1.0| 1.1|

Table 2: sensor error sensitivity results for *landmarks_th = 3.0 m*.

# References

[1]: Thomas, Antony, Fulvio Mastrogiovanni, and Marco Baglietto. "Task-motion planning for navigation in belief space." arXiv preprint arXiv:1910.11683 (2019).

[2]: Bernardini, Sara, et al. "Boosting search guidance in problems with semantic attachments." Proceedings of the International Conference on Automated Planning and Scheduling. Vol. 27. No. 1. 2017.