Università di Genova

Artificial Intelligence for Robotics II

Second Assignment Report

Students

Simone Contorno S4638263

Mattia Piras S4524072

Gabriele Russo S5180813

June 26, 2022

University of Genoa

Department of Informatics, Bioengineering, Robotics and Systems Engineering

Task Motion Planning

The content of this repository is an implementation of the system required by the Second Assignment of the **Artificial Intelligence for Robotics II** at University of Genoa, MSc in Robotics Engineering. The system here presented is a variation of the base implementation produced by [Antony Thomas et al.][1], which goal is to implement a [PDDL 2.1][2] domain dealing with the movement of a mobile robot in a constrained environment. The robot is able to move between predefined waypoints, grouped into regions, and, in this scenario, It must visit a subset of them in pseudo-optimal way. Each path between two regions is associated with a cost, a function of the Euclidean distance between their two closest waypoints.
This last distance is computed by the addition of the private function *distance_euc to the class VisitSolver* which compute the beforementioned Euclidian distance represented by the private variable *cost_dist* added to the same class.
The path planning is carried out by the PDDL 2.1 planner, whilst an external solver (a *semantic attachment*) computes the weight associated to each edge connecting regions. More info on each step in the respective section later on.

Content

```
popf-tif/
  |
  visit_domain/, task planning files
    |
    dom1.pddl      , domain description
    landmark.txt   , beacons poses
    prob1.pddl     , problem description
    region_poses   , association region-waypoints
    waypoint.txt   , waypoint poses
  |
  visit_module/  , motion planning files
    |
    src/         , motion planning scripts
      |
      CMakeLists.txt, the cmake used, linking directories to the library
      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 implementing the interface,
                     computing the EKF
      VisitSolver.h      , declaration of the class implementing the interface,
                     computing the EKF
      buildInstruction.txt, script used to build the library
      main.cpp          , main using the solver, can be used for a standalone
                     test executable (unused here)
```

Task Planning

The planner for which the system is defined is [popf-tif][3], chosen thanks to its ability to deal with numerical fluents, temporal planning and *semantic attachments*. The original domain and problem files have been only lightly modified, for instance, the previous action visit_region, that moves the robot between two regions and it is responsible for calling the *semantic attachment*. Such call is carried out thanks to a syntactic trick: every time the PDDL 2.1 keyword *increase* is encountered a call to the external solver is made, passing to it the map of all the parameters interested by such increase, and receiving a map of all the values modified in return.

Motion Planning

The computation-side of the planning is being taken care of by an External Solver, expressed here by the library libVisits.so. The majority of its implementation is done in the VisitSolver class. The robot here modelled is a simple holonomic one, so as not to have kinematic constraints in the path planning, allowing for simple straight paths between start and goal poses; the choice has been made to speed up developing and testing.
Each time a request for a localization is detected, the system finds the two closest waypoints in the regions to move (from, to), and then performs a fictious movement.
NOTE: here only one waypoint is expressed for each region, but the system can seamlessly be extended to a more general scenario.

Building and running

As mentioned, the planning used is popf-tif, which needs to be installed together with this package. This package uses Armadillo tools to optimize the matrix computations, please install it as well and pay attention to all the dependencies it needs. Lastly, the *semantic attachment* library can be built with

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

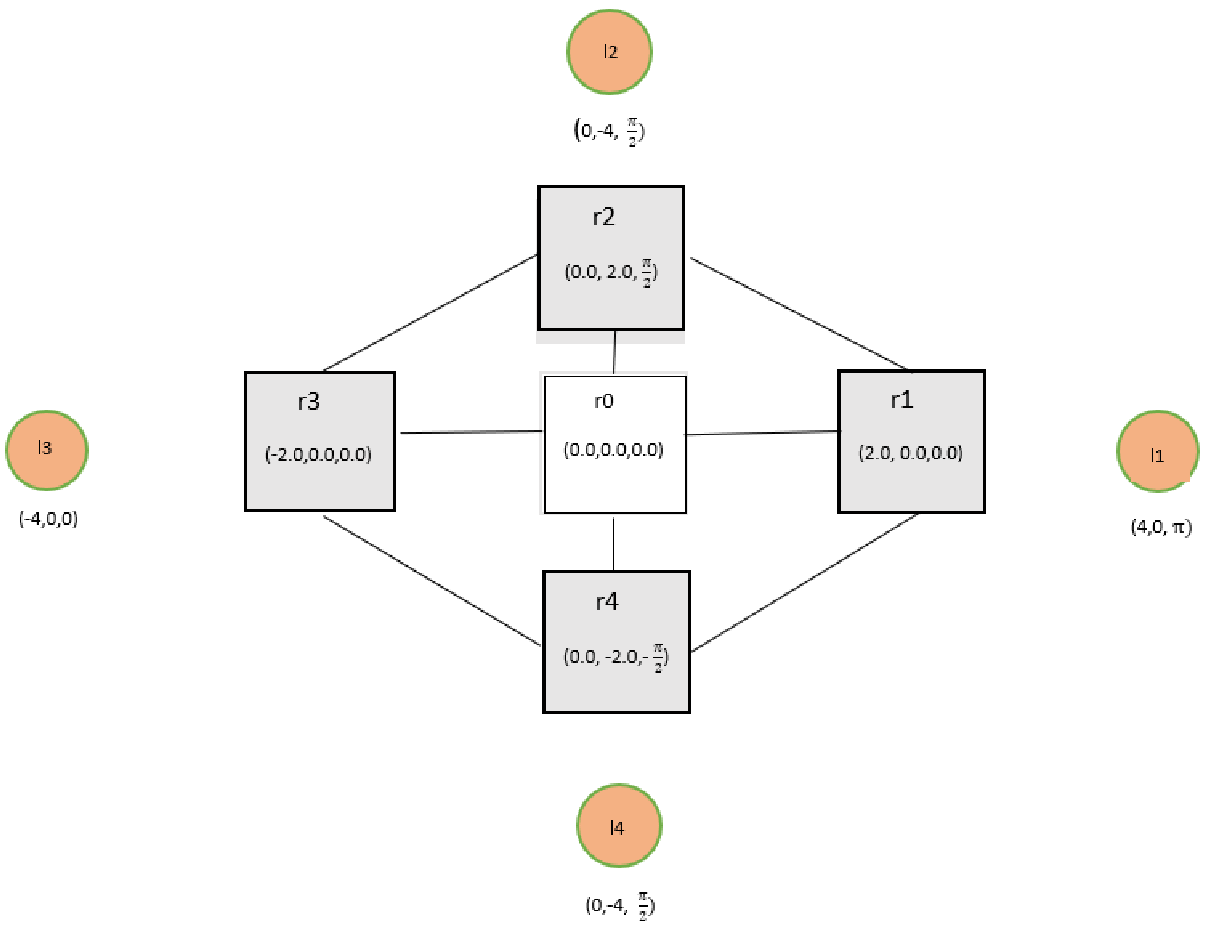
Once the building process completes, the planner can be executed with

```
visits_domain$ path/to/popf3-clp -x dom1.pddl prob1.pddl
visits_module/build/libVisits.so visit_domain/region_poses
```

The -x flag is used to inform the planner that we are going to pass it a library with the external solver description

Testing

The region_poses, waypoint.txt and landmark.txt files describe a planning scenario developed to highlight the impact of the covariance matrix on the choice of a path. These files can be modified in order to depict different scenarios, even complex ones, thanks to their minimalistic syntax.



This is the map described by the files currently present in this repository, where the box represents waypoints, circles are beacons, and the lines the edges of the region graph. The robot starts in the center region ` (0,0)` and has to reach the 4 cardinal regions (1:E, 2:N, 3:W, 4:S, here in light gray.

References

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

[2]: Fox, Maria, and Derek Long. "PDDL2. 1: An extension to PDDL for expressing temporal planning domains." Journal of artificial intelligence research 20 (2003): 61-124.

[3]: 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.