Final Project Report
Warehouse and Task Planning

Shreyas Gokhale, Uros..; Group 5

***Abstract*—You should summarize your work here. A good intro with one sentence: e.g. This technology is needed due to . Then how you approached to the problem. What is your contribution/the most valuable part of the work emphasis that. What methodologies you used. How you demonstrated your results. How was the system performance. It should not be too short or too long**

***Keywords*—IEEEtran, journal, LaTeX, paper, template.**

I. Introduction

A. Motivation

A smart factory environment is an important aspect of ”Industry 4.0” where products can be ordered, produced, sorted, stored in a warehouse and finally delivered to the recipient. The different parts of the scenario are modular and hence the information exchange between the modules is limited.

The goal of our project is to implement a scalable system for cooperative multi-agent task and path planning. The project is used in a warehouse scenario, where robot agents have to store packages into shelves as per the tasks issued. These packages theoretically arrive without any pre-planning and hence the system should be able to accommodate for the unpredictable demand.

To increase efficiency the transportation, multiple robots have to work in parallel in a limited driving space of a warehouse and with co-ordination with the other robots to avoid collisions.

Congestion is another conundrum that this system has to consider. The robots will lose valuable time if they have to wait due to congestion, reducing lower the overall effectiveness of the warehouse system.

This project strives to provide a solution for the problem of assigning packages to the robots, planning the driving paths of the robots and coordinating them. The problem can be split into two separate parts: the task planning and the path planning. Both parts have to incorporate the coordination aspect.

The task planning is concerned with assigning incoming packages to the robots considering the time needed to store the package, energy consumption of the robots and overall optimality as far as possible given that the packages arrive randomly. The technical challenge here is to find an algorithm can assign a package to the best robot under the previously mentioned considerations.

The path planning is concerned with finding and traversing an optimal path for a robot between two points in the known area of indoor warehouse, with consideration of the other moving robots, non-holonomity criteria for optimality reasons and a continuous, non-linear environment. The technical problem is to find an algorithm which plans an optimal path for a robot under the previously mentioned considerations.

B. Problem Defination

A successful implementation of the project will go through following steps:

- The project starts with receiving a package that is to be stored in the warehouse.
- The pick-up point on which the package arrives and the drop-off point, where the package is to be stored, together constitute a task that has to be assigned to a robot. An example of such a task would be to place the package from the input tray to the storage tray.
- The system should accordingly decide a storage tray and assign the task to a suitable robot agent.
- The agent should be able to successfully navigate through the environment using the path and task planner and deliver the package. Thus, finishing the task
- The system must be stable enough to be able to perform multiple tasks concurrently and accommodate for inter-robot interactions.

C. Related Literature

how literature approaches to it (very briefly as we have a related works section)

D. Approach

To approach this problem, we first considered the state of the initial framework which was given. We received a morse-ros project which already included: 1. A functioning morse environment for indoor warehouse with a demo map 2. ROS environment with essential components, excluding path and task planning. 3. A bridge between ROS and Morse environments.

The first step was to classify problem into different tasks. The main tasks were Path planning and Task Planning and then we divided them further into subtasks. After literature survey, we decided on top-level system that we will try to implement. For example: using decentralized architecture for path planning with coordination within the robot agents for path reservations. Accordingly we set following objectives to be met:

- To use existing system as a base architecture

- Implementing and testing different path planning algorithms such as RRT* and Theta*-heurisitc in ROS environment
- Decentralized Multi- Robot collaboration
- Hybrid (decentralized election but centralized assignment) task planner

In the second step of planning, we started by analysing our top level goal: To successfully implement task and path planning with multi robot collaboration and then setting assigning it for milestone 2. For path planning the research and demo implementation of the algorithm was first step. We compared and demonstrated different path planning approaches such as Theta* and RRT* on a C++ based environment and decided our milestones for implementations on ROS. Another important aspect of our system was Multi-Robot collaboration, which we decided to implement after the second milestone. A detail working of the system will be explained in III.

Finally, we decided to analyse the results based on a number of criterion which we thought might be useful for a warehousing scenario. For example: average time taken for a successful completion of the task and average time spent in different activities by each robot agent. These parameters are indicative of some interesting matrics such as what is the impact of increasing number of robots on the system and how often a robot agent needs charging. More evaluation methods and results will be explained in IV

II. Related Works

Different algorithms have been developed for path planning. One of the approaches is that we present the space as a potential field, where obstacles have high potential and the goal point has a low potential. Although the computation difficulty is low, the artificial potential fields method did not have much success in more complex environments.

One of the first proposed algorithms was A* [1] which is based on a grid-based search where the robot is allowed to move just on the line between adjacent grid points. Based on the A* algorithm several variants for optimal multi-agent pathfinding were submitted. However, the feebleness of the algorithm is the constraint of moving just between the lines connecting the grid points. This weakness was eliminated with Theta* algorithm [2], where the path is not constrained by edges of adjacent grids.

Probabilistic roadmap (PRM) [3] is a sampling-based approach where the random samples are taken and tested if they fulfil the constraints of the space. Connecting points that are near each other a roadmap is built. The path is estimated by a graph searching algorithm. Comparing to other approaches PRM is slower and takes more computational effort.

In 1998 the rapidly-exploring random trees (RRT) was proposed [4]. Later it was significantly improved with RRT* [5], which is a variant of RRT that converges towards an optimal solution. In recent years it is one of the most used algorithms and there are many variants of RRT* that try to optimize the search of the path. One of them is Theta*-RRT [6] that is two-phase path planner. In the first phase it uses Theta* to estimate the path, which is later smoothed based on nonholonomic constraints of the robot with RRT* algorithm.

For multi-agent path planning decentralization is necessary to reduce the complexity and computational effort. Moreover, decentralization enables the use of single-agent path planners, which can simplify the estimation, but some coordination between the decision makers should be provided. Due to the nonlinearity of system where delays or other unexpected events can happen, the method should include also the possibility of updating the plans according to current situation.

One of the possible forms of coordination is the round-robin approach, where one agent in each iteration replans the path [7]. Due to the fixed planning order the method is inefficient in a situation with many agents. The method was improved with the merit-based token passing coordination strategy [8], where in each iteration the agent with the highest potential path improvement (PPI) updates its plan and passes the token to the agent with the next highest PPI. The method was implemented with the RRT algorithm and is named as DMA-RRT. Due to the fact that each agent minimizes just its own cost while selecting the path DMA-RRT was improved with cooperative DMA-RRT [8] by adding the emergency stops, where an agent can safely stop if it is requested by another agent. The decision to stop is made by cost comparison of both path plans.

Based on the properties of described approaches we decided to use the cooperative DMA with Theta*-RRT single agent path planner. This approach optimizes the single-agent path plan and by adding the cooperation strategy it minimizes also the global cost of the estimated path plan. The method is also appropriate for the nonlinear systems where the task is not completely deterministic.

FROM CAN: Please inspire from your SDS, but this time make it more elaborated. It should be clear that how the problems you stated was approached in general, and how you approached with a brief paragraph. Which studies were really inspiring to you and in what sense. You should mention those. Do not forget to cite the works, both in introduction and here [1].

III. Methodology

A. Overview

FROM CAN:

You should start with describing your project, how you approach the problems state under introduction (now in detail), overall architecture and briefly describing each part of the architecture. Your architecture drawing (with its last version) will be placed here. An example figure is shown in Fig. 1.

B. Main Components of the system

FROM CAN:

Now that you described the overall architecture, it is time to elaborate more the subparts you set out above. Answer the question of how you implemented each part. What this module involves and the clear methodology followed (e.g. POMDP model is used here inspired from [?] as the robots planner).

1) Path Planner: Short summary of I/O

Gets x from taskplanner

Generates a, b,c

In depth explanations of the component

2) Task Handler: Same as path planner

C. Motion Planner

Eg what was improved from the default one

D. Warehouse and Robot Configuration

Eg map configuration, properties of individual robot, added colour to paths in RViz etc

IV. Results

Analysing the results is a very important validation for any project. But before that, one must define exact constraints for evaluation of these results.

A. Evaluation Methodology

As mentioned in I-A,the goal of our system was to design a *scalable* and *collaborative* autonomous warehouse system. The first requirement was to able to successfully demonstrate the workability of our system. This can be separated in following sub requirements:

- A task should be correctly assigned to the nearest robot
- The robot agent should complete the task by correctly following the path created by path planner.
- The robot should be able to handle obstacles and should be able to place packages on the shelves.
- The agent should be able to coordinate with other agents for reserving path space.

For successful simulation: A number of robots (>1) must be able to handle multiple tasks.

B. Evaluation Process

Considering all the above points, we decided to go for the following evaluation:
Time required per package averaged over 40 tasks.

- Constant Parameters:
  - Map: Standard
  - Starting battery: Randomized between 95 - 100
  - Number of averaged transportation tasks: 40
  - Robot type: Default
  - Path planner: Smoothed-Theta*
  - Number of repetitions per simulation: 3
- Variable Parameter: Number of robots
- Evaluated Metrics:
  - Average time required to perform task from assignment to completion
  - Average time required to perform task from execution start to completion
  - Individual time spent in:
    - Waiting for execution
    - Driving to pick up the package
    - Picking up the package
    - Driving to drop off the package
    - Dropping off the package

For each instance, the ROS, MORSE and RViz environment was initialized with said parameters. After 40 transportation tasks are successfully completed, a .csv file is generated with following parameters:

- Type of the task : If it is charging or transportation
- ID of the robot agent
- ID or sequence number of the task
- Time since start of simulation
- Time durations for various activities like: assignment, driving to pickup, picking up, drive to dropoff and dropoff

Then a MATLAB script converts this data into various intuitive graphs.

C. Evaluation results and interpretation

Graphs and explanations
FROM CAN:
Think about how you can show your projects performance. This is the most important section that will reflect your entire project, therefore it comes with the most weight on the grading of the Final Report. Therefore, analyse well what your project promises and how you can NUMERICALLY show this. Then, you discuss on the findings. Sub-sections can be created freely. Usually, the common metrics to be used are:

- overall accuracy in estimation (of whatever you are estimating: e.g. human belief, human activity, object type + shape, object pose, comparing the estimated time to store a pkg of a robot and the real time of realization etc). Of course for such analysis you need multiple runs of the system Dont forget to separate the training set from the test set (80
- overall success / precision (of the overall decisions your system makes: e.g. robotA is the most suitable for loading and storing pkg, comparison of the average time a task is realized with or without this component shows the impact of the component.)

If you can show your results on graphics it is a BONUS. If you can show on such graphs that how changing of some parameters of your mehtods/env. conditions/the context in the scenario etc. would affect your systems performance, then you will receive +10 points for your overall document.
As the first step, you start discussing on what sort of results you will show and we will definitely discuss them in the upcoming two weeks. Feel free to suggest and ask anything on that phase (be brave !).
Finally, you will conclude with the discussion of these results you showed/described. Basically answer this question, what these results tell you? Even though they are bad /not significant then you need to comment on why it is how it is. There HAS TO BE A DISCUSSION even though it is short.
You can also criticize your approach and offer better ones if it would lead to better performances. You can also briefly discuss the points to be improved/extended in your system. You will get the GRADES FROM THE DISCUSSION more than the significance or the performance of your results. In the past some groups got the highest grades although their system didnt

work well. Your goal is not to build an end product, but to be aware of what you build, a strong code base and discussion points for us to improve on your code. Positive results are of course a plus.

V. Conclusion

FROM CAN:
A brief summary of your system, what was the problem, how you achieved, your objectives, and your methodologies you used very very briefly. Then again very briefly share your finding (the discussion you made). Finally, you will conclude with the possible extensions / changes to improve the performance, basically the future works. You can find more about the templates of IEEE on the internet.

Appendix
Class Diagram

Here I would like to see your final class diagram. It doesnt require any description (unless it is complicated). It will be sufficient and necessary to mention at least once inside the methodology part that your class diagram is attached as an appendix. Also give figure caption to it.

Acknowledgment

The authors would like to thank...

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.