Centralized Motion Planner for Autonomous Vehicles in a City Grid

Rushikesh Deshmukh
Robotics Engineering
Worcester Polytechnic Institute
Worcester, MA, USA
rdeshmukh@wpi.edu

Piyush Malpure
Robotics Engineering
Worcester Polytechnic Institute
Worcester, MA, USA
ppmalpure@wpi.edu

Harin Vashi
Robotics Engineering
Worcester Polytechnic Institute
Worcester, MA, USA
hvashi@wpi.edu

***Abstract*—Localized motion planning on-board an Autonomous Vehicle (AV) using SLAM (Simultaneous Localization and Mapping) has been implemented. This individual motion planning of the route by AVs does not account for other AVs' routes, urgency, other such parameters and can lead to deadlock situations. A centralized computation unit that provides the registered AVs with a optimal motion plan is proposed by this project. A central planner will find a optimal motion plan for all the AVs in its control area. This central planner will consider the fuel constraint, priorities, speed constraint, etc while planning. This project proposes the use of Particle Swarm Optimization algorithm for meta-heuristic centralized decoupled motion planning. A classic decoupled centralized motion planning algorithm will be implemented for baseline comparison. Advanced algorithm like Grey Wolf Optimization will be studied and explored in the project.**

***Index Terms*—Centralized Motion Planner, multi-robot motion planning, City Commute planning, Path Planning, Autonomous vehicles, Mobile Robots, Collision avoidance, Traffic Planning, Multi-robot systems**

I. Introduction

Traffic Management and seamless traversing of Autonomous Vehicles (AVs) is going to be a crucial part of our system once we get to a point where all the vehicles on-road are AVs. Coordinated control of the traffic of AVs will help build efficient transportation systems with better collision avoidance and traffic control. These types of scenarios are also visible in industrial warehouse multi-robot systems. This project proposes to design a centralized motion planning unit. This central unit will be responsible to providing the AVs with global motion plan while letting the AVs take care of obstacle avoidance and local planning. The goal of the central unit is to find optimal motion plans for all the AVs in its neighbourhood. This including prioritizing high priority vehicles like ambulance, VIP vehicles, etc. Central Unit will try all cost to avoid total deadlock situation and find trajectories for AVs with minimum time-to-destination. While providing an optimal path for the AVs in an autonomous environment the problem arrives when deadlocks start increasing. These deadlocks are highly minimised by adding traffic rules like making one ways, and speed restrictions, as shown by Bolu et. al. [2]. They have worked with multi-robot systems in smart warehouse for planning efficient paths without deadlocks and collision using a central Warehouse Management System which communicates with the multiple robots and designs path plans for them, with each robot has its on system which is responsible for collision and deadlock avoidance. Baber et. al. [1] have worked on coordinated multi-robot systems using central server for planning paths and avoiding collisions at turnings.

Particle Swarm Optimization (PSO), is a very well known optimisation tool to find an optimal solution within various constraints in an unknown, dynamic environment [9]. We plan to adopt this optimization policy to find solutions if collision occurs. In a city traffic motion planning, the environment is well known to the central motion planner, only dynamic elements in the graph will be the AVs and their motion plans. This project will explore Particle Swarm Optimization (PSO) motion planning algorithm for centralized motion planning of AVs. Other Algorithms like Probabilistic Road Map, Grey Wolf Algorithms will also be looked into.

The code for the paper can be found here https://github.com/wokeengineer/Centralized_Motion_Planner_Project

II. Background

With advent in Autonomous Vehicles (AV) technology, soon on-road vehicles will all be autonomous and will be able to travel to desired location without human interaction or interference. These AVs will be need to plan their trajectories through the city. There are motion planning algorithms implemented for individual AVs that can plan these trajectories, but they are not sufficient or sophisticated enough to coordinate their motion with neighbouring AVs and can lead to sub-optimal trajectories, mismanagement of emergency vehicles, or deadlock problems. [6] At intersections, flow of vehicles need to be managed to maintain a smooth traffic flow. Typically, intersections utilize red lights to ensure safety, but these are not ideal, as vehicles are not moving for a certain period of time. Work has been done for improving the efficiency of vehicles passing through a single intersection without halting in [3]. However, vehicles were considered to move in a straight path and not make turns. Some of the concepts discussed can be modified and expanded upon, to create a grid with a centralized motion planner. [4] uses a hierarchical approach to solving this problem, using one layer each for defining the topography, planning the path of an individual AV, and for managing conflicts that may arise in these planned paths. Multi-Robot Motion Planning can be broadly divided into two type: Centralized Motion Planning and Decentralized Motion Planning. In Decentralized Motion Planning, AVs will calculate their own motion plans and coordinate amongst themselves in a decentralized manner. In a Centralized Motion Planner, a central system will calculate the motion plans for all the AVs in a given network. Centralized Motion Planning is computationally heavy but will provide closer to optimal solution for all the AVs present.

Centralized and Decentralized motion planning algorithms are further comprised of coupled and decoupled motion planning algorithms. Decoupled Multi-Robot Motion Planning algorithms were explored in [7]. Here, first the path for each vehicle will be calculated individually and the most efficient paths are given to each vehicle. The vehicle then individually handles the collision conditions if and when they arise using different techniques like velocity reduction, stopping, etc. In coupled motion planning algorithms, all the AVs are considered together for generation of motion plans. This is computationally heavy as the configuration space size increases with the number of AVs.

However, these coupled motion planning algorithms will generally have better success rates as these algorithms anticipate deadlocks and collisions and gives more optimal solutions than decoupled algorithms. Another efficient way to tackle the deadlock situations is using Probabilistic Road Maps (PRM), Clark [7]. Using the Dynamic Robot Networks (DRN) in combination with PRM has shown to provide efficient solution against the problem of deadlocks by including the Centralized planner due to faster on-the-fly motion planning in dynamic obstacle constraints.

III. Methods

We designed a system for seamless movement of AVs through a city, without stopping at any intersections, while avoiding all collisions. To implement this, we made a 2 dimensional city grid with multiple vehicles spawning as point robots. Each vehicle has its own position and a destination to get to. A centralized controller provides a path to reach the destination, handles all deadlocks, and helps avoid collisions. This project makes the following assumptions.

Assumptions:

- All roads are two-lane and two-way.
- Vehicles have 5 discreet speed levels.
- Vehicles have finite fuel that is spent based on the length of the path.
- Emergency vehicles can appear which need priority to reach their destination the fastest

A. Registration of AVs with Central Planner

AVs send a registration packet to the central planner with necessary information including start location, goal location, fuel limit, and priority level. Based on this information, the central planner calculates the optimal motion plan for all the AVs and sends it to them. The global path plan will factor in prioritization for emergency vehicles such as ambulances or firetrucks, and the fuel limit will ensure vehicles with a low fuel level reach their destination earlier. For the purpose of this project, this communication is be done via ROS using client/service model, where each vehicle is be a node initiated at a random point in time.

B. Motion Planning Algorithms

This project works a 2D grid world that can be easily converted into a graph. Efficient Graph searching algorithms are used for motion planning [5]. Central Planner uses the following algorithms to find optimal trajectories for the AVs, and the Motion Planning Algorithms that were explored are mentioned in the table below:

- A* and variants based decoupled centralized motion planning algorithms
- PSO based centralized motion planning algorithms
- Progressive Motion Planning Algorithms

C. Constraints

Since an optimal motion planner has been designed for the project, we also explored some more scenarios which would add constraints to the motion planner.

Multiple levels of priority can be given to vehicles like police cars, fire fighters and school buses. According to the priority of the vehicle different speed and path will be allotted to the vehicles in the system.

During particular cases like new year events, road repairs, etc. certain roads might be intermittently closed, solving these constraints can be added. When a vehicle is at low fuel levels it would reroute to a charging station and priority will be given according to its charging levels.

D. Cost Function

As seen in the figure 1, even though the path from the start to the goal location is optimal in the classical sense, it would not be optimal practically. This is because the path takes multiple turns, for which a real-world vehicle would need to slow down. It is possible to create a path with fewer turns and in order to promote that path in the algorithm, we add cost for each turn taken. We also use the cost function to make the roads one-way and prevent AVs from backtracking or taking u-turns.

E. Simulation Environment

In this project, a python class called 'Blocks' was designed to represent different type of road blocks like a cross-section, straight roads, turns,etc. Each Block is a road element, shown in figure 2. This project integrated Pygame with this class in order to stitch together a map of a simulated city grid, one block at a time. The grid is a 72 x 36 array of Blocks, and each block is a 20 x 20 block of pixels.

Autonomous Vehicles spawn at random points on the city grid, with a predefined destination, and they must navigate the streets to get to it.

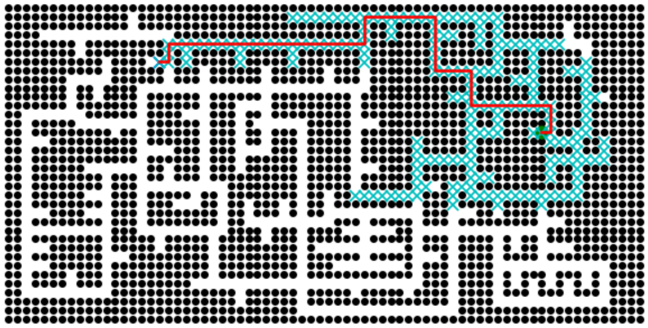

Fig. 1. A* being performed from Start to Goal for One Car

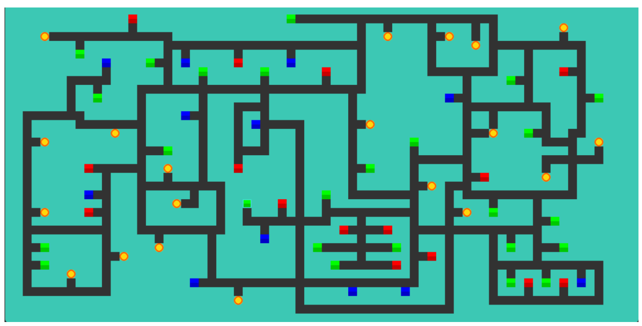

Fig. 2. Static environment in Pygame

F. Centralized Motion Planner

The centralized motion planner works in the following manner:

- Find an optimal path for the AVs using A*, it will do this for each AVs.
- Check if any AVs are colliding in the vector of (position, time).
- If the AVs are colliding try to avert the collision by varying the speed of vehicles
- If the previous method does not work, use A* again to find another, possibly sub-optimal solution
- If alternate optimal paths cannot be formed, use PSO to find sub- optimal solutions for the AVs that can't have optimal motion plans without braking or collisions. PSO will try to next best solution for the given AVs. [8]

G. Bringing everything together with ROS

The ROS structure is as shown in the figure 4

There are 4 main components to the ROS structure

- Planner Server
- Car Client
- Car Publisher
- Simulator Subscriber

1) Planer Server: The planner server gets the start and goal destination from the car node, it is then passed on to the centralised path planner which keeps track of the cars and their positions, as described above. It further generates the

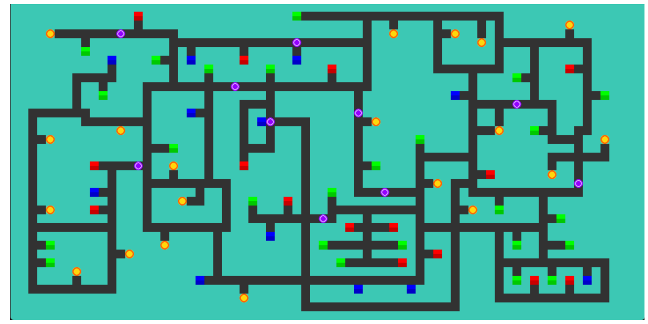

Fig. 3. A Still of 10 Vehicles Travelling through the Environment in Pygame

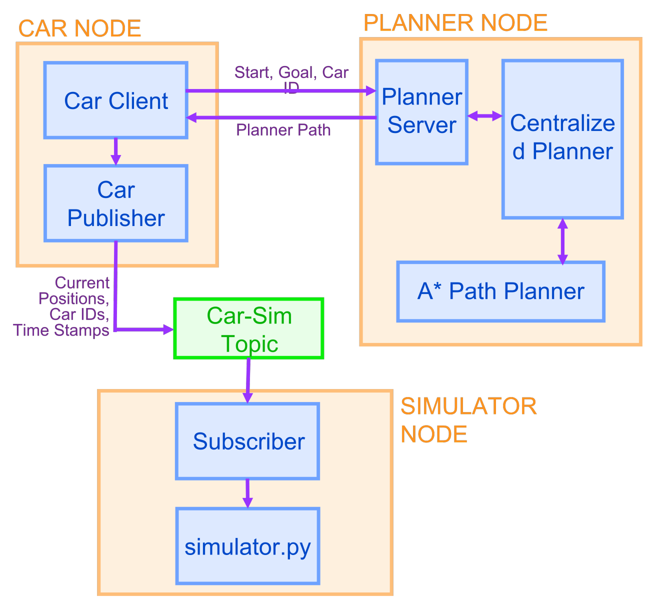

Fig. 4. ROS Structure

whole path from the A* Path planner. It returns the path to the car node which requested the path.

Every car has a node created with at a random time, the car node acts as a client to the planner server, sending it the start node and required goal node with its ID, and receives the path planned from the global planner. Using the local planner it further traverse in the environment and publishes the current positions at each time step with car ID to the topic car_simulator. Further once the car has reached its destination without collision or encounters a collision the same information is passed to the centralised planner using a car_status topic.

The Simulator node subscribes to the car_simulator topic on which every car publishes its current position and ID, using the car IDs it simulates each and every car in the simulator environment.

IV. GOALS

- Centralized Motion Planning Unit based on Probabilistic Road Map algorithm
- Priority routing to emergency vehicles.

- Effective Hand-off between two neighbouring central planners.
- Different speed limits on roads and vehicles shall be considered.
- Fuel limits of the AVs shall also be considered by the centralized planner.

V. Results

A* algorithm was implemented to navigate a vehicle from a corner of the city to another point within the city. The algorithm generated the solution shown in the figure 1.

This project designs a central motion planning unit that handles motion planning of multiple AVs entering and exiting its coverage region. It works to avoid deadlock situations and tries to give an optimal trajectory for each of the AVs. This algorithm can also take into account different fuel constraints of the AVs, their priority levels, and other constraints. This project also visualizes the working of this planner on a 2D Grid based simulation environment, and a City Grid visualization was built towards that end.

For checking the effect of our algorithms against each other we performed the following set of experiments:

- Naive Centralized planner:
  The Naive planner acts as a baseline to compare the performance of the algorithms. In this algorithm the centralized planner does not check for collisions and take any preventive measures for solving any conflicts. It uses an A* algorithm to plan paths for multiple cars.
- Centralized planner with prioritized safe-interval planning:
  This planner reduces the velocity of the obstacles when the cars are on the path to collision. It showed better performance than the Naive algorithm with lesser collisions and more cars reaching safely home.
- Combinational Centralized planner with prioritized safe-interval and path recalculation:
  This planner reduces velocity and if necessary recalculates the whole path so that there is no collision. This algorithms is the most efficient algorithm. With very less collisions even when the traffic is very high.
  The comparative analysis of the performance is as shown in the figure 5.

During the experimentation, performance of the algorithms was recorded by running the experiment for 10 minutes with cars popping up at random places and going to random destinations at random time intervals. Almost around 300 plus cars were generated within the time and the traffic was handled.

VI. Discussion

The centralized planner and localised planner approach of the project shows significant improvement in the handling of traffic planning for a city grid. The original goal of the project was to implement Particle Swarm Optimization to further improve the optimization of the centralized planner in the highly dynamic environment. The authors tried to implement

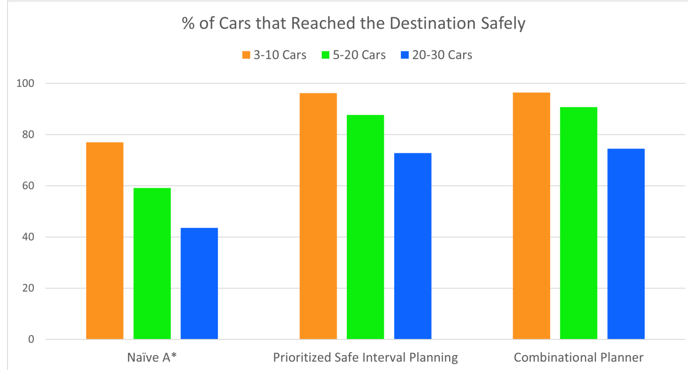

Fig. 5. Comparative analysis

and try the PSO algorithm but found out that the optimization to find an optimal goal point for this particular problem takes around one to two minutes per car, thus was discarded as a solution for this particular problem. PSO was tried to find an optimal path when a point of collision occurs, this should be done by the global planner faster enough so that collision does not occur. With the given time restrictions the authors could not explore PSO further. It seems like a good direction to explore further for multi-robot path planning in dynamic solution.

VII. Task Division

The following task division in table I.

TABLE I
Task Division

Name	Task
Rushikesh Deshmukh	City grid visualization, ROS Module design, Motion planning system design
Harin Vashi	City grid visualization, Motion planning system design, constraint design
Piyush Malpure	ROS Module design, Motion planning system design, constraint design

VIII. Schedule

The following schedule was followed.

References

[1] Baber, J., Kolodko, J., Noel, T., Parent, M., Vlacic, L. (2005). Cooperative autonomous driving: intelligent vehicles sharing city roads. IEEE Robotics & Automation Magazine, 12(1), 44-49.

[2] Bolu, A., Korçak, Ö. (2019, November). Path planning for multiple mobile robots in smart warehouse. In 2019 7th International Conference on Control, Mechatronics and Automation (ICCMA) (pp. 144-150). IEEE.

[3] Müller, E. R., Carlson, R. C., Junior, W. K. (2016). Intersection control for automated vehicles with MILP. IFAC-PapersOnLine, 49(3), 37-42.

[4] Pratissoli, F., Battilani, N., Fantuzzi, C., & Sabattini, L. (2021, May). Hierarchical and Flexible Traffic Management of Multi-AGV Systems Applied to Industrial Environments. In 2021 IEEE International Conference on Robotics and Automation (ICRA) (pp. 10009-10015). IEEE.

TABLE II
SCHEDULE

From	To	Task
-	31-Jan-2022	Project Proposal
1-Feb-2022	8-Feb-2022	Designing City Grid
9-Feb-2022	16-Feb-2022	One Point AV motion plan design
20-Feb-2022	21-Feb-2022	Project Proposal Update
22-Feb-2022	8-March-2022	ROS setup for multi-AV system
8-March-2022	21-March-2022	Central Motion planner for multi-robot system
22-March-2022	04-April-2022	Exploring motion planning algorithms for the system
05-April-2022	18-April-2022	Adding constraints and testing
19-April-2022	25-April-2022	Final testing of system
26-April-2022	29-April-2022	Project Report, Presentation

[5] Le, D., & Plaku, E. (2019). Multi-robot motion planning with dynamics via coordinated sampling-based expansion guided by multi-agent search. IEEE Robotics and Automation Letters, 4(2), 1868-1875.

[6] Cirillo, M., Uras, T., & Koenig, S. (2014, September). A lattice-based approach to multi-robot motion planning for non-holonomic vehicles. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 232-239). IEEE.

[7] Clark, C. M. (2005). Probabilistic road map sampling strategies for multi-robot motion planning. Robotics and Autonomous Systems, 53(3-4), 244-264.

[8] Banerjee, D., Majumder, C. G., Roy, S., Chakraborty, A., Konar, A., & Janarthanan, R. (2011, December). Multi-robot motion planning amidst dynamic obstacle. In 2011 International Conference on Recent Trends in Information Systems (pp. 191-196). IEEE.

[9] Masehian, E., & Sedighizadeh, D. (2013). An improved particle swarm optimization method for motion planning of multiple robots. In Distributed autonomous robotic systems (pp. 175-188). Springer, Berlin, Heidelberg.