Multi-Trailer Truck Parking Problem

* RBE 550 Course Final Project

Gampa, Varun
Masters in Robotics Engineering
vgampa@wpi.edu

Siddhartha, Rohin
Masters in Robotics Engineering
rvenkateswaran@wpi.edu

Kalathiya, Raj
Masters in Robotics Engineering
rmkalathiya@wpi.edu

***Abstract*—One of the most challenging tasks for truck drivers is maneuvering the truck-trailer system in different parking scenarios. In this project, we develop and test path planning algorithms for this application and verify via simulations of their performance. We believe this project has the potential to be applied in areas of autonomous driving or driving assist features.**

I. INTRODUCTION

Trailer parking is a crucial aspect of the transportation industry. The traditional methods of trailer parking are time-consuming and require human intervention. Autonomous vehicles have been a hot topic in recent years, with significant advancements in technology making them a more feasible option for everyday use. One of the challenges facing autonomous vehicles is parking, especially for large vehicles such as trailers. Trailer parking is a complex task that requires precise and accurate parking skills. With the increasing demand for autonomous vehicles, the development of autonomous trailer parking is becoming increasingly important. In this project, we propose to develop a path-planning system for autonomous trailer parking. This system will be capable of navigating a trailer through a yard or parking area and successfully parking it in a designated spot.

II. BACKGROUND

One of the most challenging tasks for truck drivers is maneuvering the truck-trailer system in different parking scenarios. The problem of parking comes under the domain of planning with kinematic constraints or non-holonomic constraints. A non-holonomic constraint is a non-integrable constraint. That is, for this problem, they are of velocity constraint type. For the trailer truck model, the kinematic model is already prevalent in the literature as noted here [4]. The reason why this problem is hard to solve is because of the limited configurations the obstacles allow for the trailer truck to access, as a result, solutions should be of the form wherein the truck should go back and forth in the same place, with just tiny adjustments in its heading to create the possibility of going into the parking space. The problem with existing methodologies is that trailer trucks being higher dimensional causes the traditional search algorithms to take significant time, especially when there are more obstacles. There have been some research papers, which describe novel algorithms to solve this problem such as [5]. Though there aren't many reports of stress testing these algorithms against other algorithms as well as extending this problem to the multi-trailer case.

III. METHODOLOGY

In this project, we have built on top of a mathematical model of a 1-trailer system where there is one pulling truck and 1 trailer being pulled by it. The trailer is hinged at the center of the rear axle of the pulling truck. The Kinematic model for the 1-trailer system defined by [1] is used in this project (shown in figure 1). The model simplifies 4 wheels to 1 turning wheel at the center, as an extension of the bicycle model. As the speed of the trailer when parking will be low, this approximate assumption is very accurate. This model is first extended to the 1-trailer system and then multi-trailer systems.

For the path planning of the truck-trailer system, we will be using a number of path-planning algorithms such as hybrid A*, RRT, etc. to compare and provide an analysis of the effectiveness of different algorithms. The major challenge, which determines the efficacy of these algorithms is to take into account the non-holonomic constraints of the model along with the constraints of the map.

For the path planning of the truck-trailer system, we used hybrid A* and RRT to understand the capabilities of the straightforward implementations of these algorithms for higher degrees of freedom and under-actuated systems. Based on the challenges and limitations of these algorithms, a modified hybrid RRT - local A* solution is developed.

We simulated our model in a 2d grid in a known environment. The algorithm will know all the relevant information such as the location of other vehicles, parking spots, and obstacles. This information will be processed to generate a feasible path for the trailer. All the algorithms were set to maneuver the truck in the environment with non-holonomic constraints in place.

A. Standard Algorithms

1) A*: The A* algorithm is a pathfinding algorithm that searches a graph to find the shortest path from a starting node to an end node. The algorithm uses both the cost of the path from the starting node to a given node

$$
\begin{aligned}
\dot{x} &= s\cos\theta_0 \\
\dot{y} &= s\sin\theta_0 \\
\dot{\theta}_0 &= \frac{s}{L}\tan\phi \\
\dot{\theta}_1 &= \frac{s}{d_1}\sin(\theta_0-\theta_1) \\
&\vdots \\
\dot{\theta}_i &= \frac{s}{d_i}\left(\prod_{j=1}^{i-1}\cos(\theta_{j-1}-\theta_j)\right)\sin(\theta_{i-1}-\theta_i) \\
&\vdots \\
\dot{\theta}_k &= \frac{s}{d_k}\left(\prod_{i=1}^{k-1}\cos(\theta_{j-1}-\theta_j)\right)\sin(\theta_{k-1}-\theta_k).
\end{aligned}
$$

Fig. 1. Kinematic model of trailer truck

and an estimated heuristic cost from that node to the end node to determine the order in which nodes are explored. This makes A* more efficient than other uninformed search algorithms such as breadth-first search or depth-first search.

A* algorithm is designed to work on known graphs, and as the dimensionality of the search space increases, the algorithm can become less efficient and less effective. This is because the number of possible states that need to be explored grows exponentially with each added dimension. In the context of trailer routing, each trailer added to the problem adds another dimension to the search space, and as the number of trailers increases, the complexity of the problem grows exponentially. This can lead to the A* algorithm becoming inefficient and not being able to find the optimal solution in a reasonable amount of time. Additionally, the A* algorithm may not be able to handle the presence of obstacles or other constraints in higher dimensional search spaces, which can further limit its effectiveness.

2) RRT: The RRT algorithm is a probabilistic algorithm used for motion planning in robotics. The algorithm incrementally builds a tree of feasible configurations (or states) of a robot in the configuration space (C-space). The tree is constructed by randomly sampling the C-space and adding nodes to the tree that are connected to the nearest existing node, until a path from the start configuration to the goal configuration is found.

   RRT is a powerful algorithm for finding collision-free paths in high-dimensional spaces. However, it has limitations when it comes to solving problems with complex kinematic constraints, such as multi-truck trailer systems. As the number of trailers increases, the complexity of the system grows, making it difficult for RRT to generate feasible trajectories. The kinematics of the trailer-truck system poses a significant challenge, as trajectories generated by RRT may be difficult or impossible for the system to execute. Therefore, while RRT is a useful tool for path planning in many contexts, it may not be sufficient for more complex systems with many trailers.

B. Proposed Algorithm: Hybrid RRT - local A*

3) The hybrid RRT-local Astar algorithm combines the strengths of both RRT and A* to overcome their limitations. In this algorithm, the RRT is used to generate a tree towards the goal, and local Astar is used to traverse the nodes of the tree.

   To improve the efficiency of the RRT, the end goal is sampled with a probability of 0.1, so the algorithm can start exploring the goal area earlier. This is known as goal bias. In order to further speed up the algorithm, along with sampling just from the goal node, with a higher probability points are sampled from Gaussian distributions centered around the goal and the obstacles.

   To avoid generating trajectories that are difficult for the trailer-truck system to execute, the steer methodology is used to generate multiple nodes from the nearest node to the sampled node. In this method, given a sampled node point and the nearest neighbor, assuming no obstacles a local path from the nearest node to the sampled node is constructed. All points in the local path which are collision-free in reality are added as nodes to the tree. This helps to ensure that the generated paths are collision-free and feasible for the trailer-truck system to execute and also increases the rate of expansion of the biased random tree.

   The Reeds Shepp distance metric is used to find the nearest node in the tree to the sampled node, which helps to generate more feasible paths. Reeds Shepp distance metric is developed for kinematic models of cars. This is extended to multi-trailer systems by increasing the turning radius. Further in the heuristic of local A-star, ReedS Shepp distance metric is used, along with additional penalties on the angle between trailers in the truck. This ensures the trailers don't have large angles between them.

   By combining the global exploration capability of RRT and the local optimization capability of Astar, the hybrid RRT-local Astar algorithm is able to efficiently generate collision-free paths toward the goal while ensuring that the generated paths are feasible for the trailer-truck system to execute.

IV. Experimental Setup

For our truck trailer system, we used the Ackerman steering model for the truck. The Ackerman model is a simple and widely used steering model that describes the kinematics of a vehicle with four wheels. It assumes that the wheels of the vehicle follow circular arcs around their centers of rotation. The model calculates the turning radius of the front wheels and the rear wheels separately, allowing the vehicle to turn around a curve without slipping or skidding.

In our implementation, we used the Ackerman model to calculate the steering angle of the front wheels of the truck based on the desired turning radius and the current speed of the vehicle. We then used this steering angle to calculate the position and orientation of the truck in the simulation environment.

To simulate the kinematics of the trailers we used velocity model, this helps reduce the complexity of u. We followed the Ackerman steering model for the truck and calculated the turning radius of each trailer based on the steering angle and position of the truck. We then used this turning radius to calculate the position and orientation of each trailer in the simulation environment.

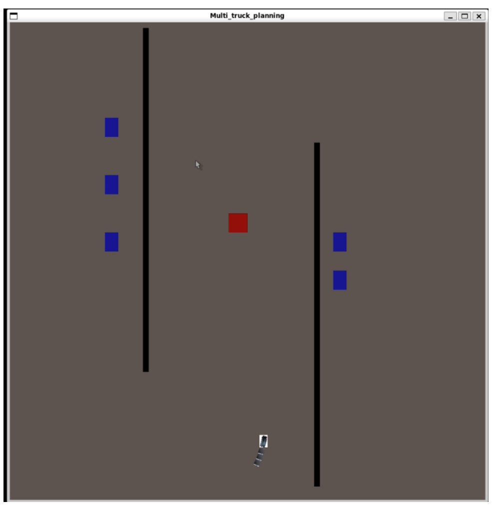

Fig. 2. Path planning in action using Hybrid RRT Local A-Star

Challenges overcomed: One of the main challenges faced during the development of the simulation environment was the higher dimensionality of the truck-trailer system and limited computational resources. To address this challenge, memory usage was optimized by dynamically generating graphs.

Another challenge was the use of Euclidean distance as a metric to compute the heuristic for path planning. This approach did not generate feasible paths due to the non-holonomic constraints of the trailer truck system. To solve this problem, the Reeds Shepp distance metric was used, which considers the turning radius of the truck and generates smoother trajectories. Additionally, the metric was updated to include a penalty on the trailer's angle not being different from the angle of the truck.

The truck sometimes used to get stuck at locations close to obstacles and could not find a path quickly. To address this problem, the heuristic used in the local A-star algorithm was updated to include a penalty on the distance from obstacles. The penalty function used was (1/(1+ exp(d))), where d is the distance of the truck from the center of the obstacle. In the RRT portion of the algorithm, nodes were randomly sampled from a Gaussian distribution with a mean of the center of a randomly chosen obstacle, with a probability that improved the chances of finding a path quickly while still maintaining completeness in probability. A penalty on the angle between the trailers is also added to generate smooth curves

Another challenge faced was the time taken by the RRT algorithm to reach the goal. To solve this problem, the goal was sampled as a node with a small probability (0.01), and the steering technique was used to generate multiple nodes in the direction of the sampled node from the nearest node. The direction was determined by A-star, which assumes that there are obstacles, and the nearest node was determined using the Reeds Shepp distance metric.

In summary, the challenges faced during the development of the simulation environment were overcome by optimizing memory usage, using the Reeds Shepp distance metric, updating the heuristic to include a penalty on the distance from obstacles, sampling nodes from a Gaussian distribution with a mean of the center of a randomly chosen obstacle, and sampling the goal as a node with a small probability and generating multiple nodes in its direction using the steering technique.

Fig. 3. Final Path Obtained using Hybrid RRT Local A-Star

V. Expected Results

The proposed path planning algorithm for autonomous trailer parking will provide a highly effective solution for the challenge of parking trailers in complex environments. The system will be designed to handle multiple trailers in compact spaces with difficult kinematics while considering non-holonomic constraints and obstacle avoidance. A graph of paths generated by the algorithm will be shown, along with the motion of trailers parking in a given goal location. A performance evaluation of different path planning algorithms implemented will be provided to understand the effectiveness of each algorithm.

VI. Obtained Results

Time taken to find path to goal using our proposed algorithm, Hybrid-RRT Local A-Star: 800 seconds. We have concluded that both RRT and A-star, can't find solution under 1500 seconds for three trailer problem.

References

[1] LaValle, Steven M. "Planning Algorithms". In: Cambridge University Press (2006).

[2] H. Esteban Cabezos, 'Optimization of the Parking Manoeuvre for a 1-Trailer Truck', Dissertation, 2020.

[3] Bill Triggs. Motion Planning for Nonholonomic Vehicles: An Introduction. 1993. ⟨inria-00548415⟩.

[4] "Truck and Trailer Automatic Parking Using Multistage Nonlinear MPC - MATLAB and Simulink".

[5] A. C. Manav and I. Lazoglu, "A Novel Cascade Path Planning Algorithm for Autonomous Truck-Trailer Parking," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 7, pp. 6821-6835, July 2022, doi: 10.1109/TITS.2021.3062701.