Autonomous Driving for Highway Roads

Name: .....

Abstract—This paper presents an overview of this history of driving autonomous vehicles for highway roads. It presents path planning algorithms for two different types of surrounding environments. For path planning of static environment, it presents A-Star and Dijkstra path planning algorithms. For path planning of dynamic environments, it presents D-Star and Rapidly-exploring Random Trees(RRTs) path planning algorithms. Finally it presents examples of motion control systems.

Index Terms—Autonomous Driving, Path Planning, A*Algorithm, Dijkstra Algorithm, RRTs Algorithm, D* Algorithm, Lane Change, Car Follow, Dynamic Environment, Static Environment.

1 Introduction

The 1930's has witnessed the first trials of introducing an autonomous electric car, manufactured by General Motors [1]. Developing Autonomous vehicles systems, has been the interest of engineers and scientists for a long time. The main focus stemmed from the ambition of landing on the moon and explore life outside. The importance of autonomous vehicles is not only to be used on the moon, but also the usage of such vehicles to save humans in life threatening situations. The research focus has been on the development of driving aspect of the autonomous system since it is a driver-less vehicle, it has to function as if it was operated by humans [2].

Autonomous vehicles are expected to operate and navigate in an unknown environment. For the vehicle to actively function in a drive-less aspect, it must contain two modules, a path planning and navigation controller which work in a sequentially. In the path planning controller, there are three elements which aid the navigation controller as the following; real-time surrounding environment perception, decision making based on the controller inputs (Vision, sensors) [3]. As soon as the path planning module makes the decision based on all the real-time inputs, the navigation module takes action. The design of the path planning module is different based on the type of environment which will inhabit the autonomous vehicle.

Path planning for autonomous vehicles for highway roads is a challenge. The autonomous vehicle is in a constantly changing environment and it has to be fully alert to all of the changes in the surrounding dynamic environment. In this paper, we will explore path planning algorithms of two contrasting types of environments and examples of the navigation modules which puts the path planning algorithm results into action. The rest of the paper is organised as the following: An overview of the path planning algorithms for highway roads driving is given in Section II. A description of the motion control systems like obstacle avoidance control systems and lane change and speed control systems is presented in Section III. Discussion is in Section IV. Finally, conclusions are in Section V.

2 Path Planning Algorithms for Highway Roads Driving

The concept of path planning is based on finding the best navigating path starting from the origin point of our autonomous agent towards the goal point. The planning and navigation differs based on the chosen environment for our autonomous agent.

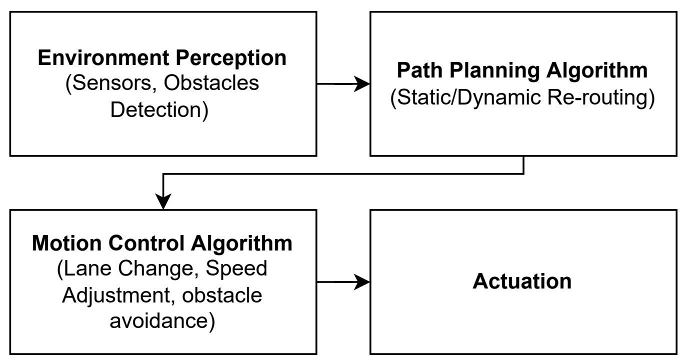

Fig. 1. The figure illustrates the abstract flow of the driving algorithm of an autonomous agent.

Path Planning for highway roads is considered to be planning for dynamic environment, since the autonomous agent has little to none information regarding the surrounding environment prior to navigation. We have two different types of path planning which are presented in this paper. The first path planning type is for static environment of known obstacles and constraints. Our agent solves the planning problem entirely based on the knowledge acquired before the problem-solving starts. Contrastingly, path planning for dynamic environments requires perpetual adjustment of the resulting path as the surrounding obstacles and constraints are constantly changing.

For each type of environment, two path planning algorithms are discussed.

2.1 Path Planning for Static Environments

Static environments are fixed environments with respect to time. No new elements are changed or added throughout the execution of the path planning algorithm. The planning

algorithm does not take into consideration any additions to the environment while solving the path [4].
These environments are typically indoor environments with defined elements, such as obstacles, cars or pedestrians.
We will introduce two of the commonly used path planning algorithms for static environments. The first algorithm is Dijkstra algorithm and the second algorithm is A-Star algorithm.

2.1.1 Dijkstra Algorithm

Dijkstra Algorithm is an uninformed search algorithm which explores all the possible nodes existing in a graph, without knowing the target node prior to the search. The algorithm main goal is to find the shortest-path tree between the starting and goal points. The shortest-path tree can be defined as having the least number of node jumps between two points [6]. Equation (1) represents the heuristic function of the Dijkstra Search algorithm

$$f(i) = g(i), \tag{1}$$

where $f(i)$ is the estimated total cost of a path from the origin node, passing node i till the goal node. $g(i)$ is the real cost distance leading to our goal node.
The algorithm starts navigation from the origin node towards all the nodes in the graph. It later backtracks all the shortest-path trees between the origin node with all neighbouring nodes and chooses the shortest-path tree from the origin node to any node in the graph [5]. The Dijkstra search algorithm has other variations, which all dependent on the choice of the heuristic search function. These variations lead to finding the shortest-path tree for any environment graph, depending on the application.

2.1.2 A-Star Algorithm

A-Star Algorithm is an informed algorithm which works towards finding the shortest-path between the origin node and the target node [7]. The search algorithm branches out to Dijkstra search algorithm. Since the algorithm is informed with the target node, it investigates the best candidate nodes rather than investigating all the nodes in the graph. The branched search of the A-Star is based on different variations of the Dijkstra search algorithm.
Equation (2) represents the heuristic function of the A-star Search algorithm

$$f'(i) = g(i) + h'(i), \tag{2}$$

where $f(i)$ is the estimated total cost of a path from the origin node, passing node i till the goal node. $f'(i)$ has to estimate the real distance leading to our goal node. $g(i)$ represents the cost distance between the starting node and node i. $h'(i)$ expresses the approximate distance from the i to our goal node.
The used heuristic function in the A-Star algorithm constraints the choice of the next node to be investigated. The use heuristic search function gives priority to the best fitted nodes compared to the goal node, be explored first, rather than going through all the nodes.

2.2 Path Planning for Dynamic Environments

Path planning for constantly changing environments creates more challenge. The environment contains more variables to be considered while designing the path planning search algorithm. Dynamic environments resembles the outdoor environment of changing obstacles.
There are two commonly used algorithms for dynamic path planning. The D-star algorithm which is a dynamic version of the A-Star algorithm and the second algorithm is Rapidly-exploring Random Trees (RRTs) which is based on the A-Star algorithm with enhancements [8].

2.2.1 D-Star Algorithm

For Dynamic environments, D-Star Algorithm is used for dynamic path finding and navigation [9].
D-Star algorithm is the dynamic search algorithm version of the A-Star algorithm. It perpetually computes alternative re-routes as dynamic obstacles appear and the environment changes [10]. It divides the entire surrounding environment into sections, at which every section has an ending point which acts as the starting point for the next section of the environment. The algorithm puts preliminary assumptions about the unknown existing space, for example the highway road does not have any stop signs, and finds the shortest path and then updates its map with the information until it reaches its goal node.
Equation (2.2.1) represents the heuristic function of the A-star Search algorithm

$$\begin{aligned} k &= min(h_{i+1}, h_i), \\ key &= k + g \end{aligned} \tag{3}$$

where k outputs the minimum cost distance between the current node and previous node. g is the accumulating cost estimate from the starting node. key is the priority node queue total cost.
The algorithm further tries to build up its knowledge map by dividing it to small unknown spaces, updates it and sometimes replaces the existing path plan if the assumptions are found to be false. It finally tends to learn about the environment as it progresses towards its intended goal node.

2.2.2 RRTs Algorithm

The Rapidly-exploring Random Trees(RRTs) algorithm works through creating random branching tress rapidly. The algorithm starts from a random point in the graph and generates path tress across the space until one of the branched tree coincides to be the goal node. After the creation of all the branching trees, It consistently searches for the goal node and backtracks to the origin node through all the possible paths and selects the shortest-path from the goal node to the origin node [11].
The RRT divides the whole space into equal squares in distance, it further starts from the middle point of each square and randomly generates trees in a directions within the current square, to explore the four corners of that square. It tends to connect all the square together through the nearest vertex up to the goal node [12]. Whenever an obstacle has been met by the algorithm in the robot space, the path tree at which this obstacle was found is immediately deleted as

a potential shortest path tree, and a new path within the current robot space is explored [13].

3 Motion Control Systems for Highway Roads Driving

The implementation of The path planning algorithm

and navigation for autonomous driving for highway roads, it is important to design motion control system to aid the path planning algorithm to achieve its main goal of steering the autonomous car from the starting point towards the goal point.

As the autonomous driving is set to take place on highway roads, the surrounding environment is constantly changing and requires our agent to identify and avoid obstacles through its obstacle avoidance system. Moreover, it should change lanes and control the speed accordingly, if needed.

3.1 Obstacle Detection and Avoidance Control System

Obstacle Avoidance Control Systems is an essential system to exist in any autonomous system. It acts like the driver in terms of detecting obstacles, calculating the appropriate distance left before collision, switch lanes and alter the vehicle speed accordingly.

An Avoidance System has been introduced through hardware sensors and the most effective is through LiDAR sensors with its high accuracy and range to give enough time for detection and avoidance of any obstacle in range [14].

3.2 Lane Change and Speed Control System

The two mentioned motion control systems in this paper are interconnected through the path planning algorithm. The obstacle detection and avoidance system acts as the input signal for the path planning algorithm, the changes are reflected as input signals to the lane change and speed control system.

The control system is responsible for switching lanes upon receiving input from the path planning algorithm regarding obstacle detection [15]. furthermore, the speed control goes into effect as switching lanes cannot be achieved on the same speed used on direct routes [16].

4 Discussion

From the discussion above in the path planning algorithms for static environments, it has shown that A-Star algorithm presents better results in terms of discovering the shortest-path tree in comparison to the Dijkstra search algorithm. A-Star search algorithm is an informed search algorithm, which provides a more optimized approach towards finding the shortest-path through prioritizing the nodes which more in favor of reaching our goal node.

The A-Star algorithm is an enhanced version of the Dijkstra search algorithm but the main difference about the A-Star algorithm, is not exploring all possible nodes in the graph but eliminating the less lucky nodes and in fact perform the Dijkstra search algorithm only on some of the nodes.

For dynamic path planning algorithms, we have built on the discussion of the validity of the A-Star search algorithm and choosing its dynamic version, which is called D-Star search algorithm. It functions in the same way as the A-Star search algorithm but in a dynamic manner. The Algorithm constantly computes the changes in the surrounding environment and updates its map as it converges towards its goal node.

Regardless of the choice of the path planning algorithm, the motion control systems completes the whole picture presented by the path planning algorithm.

5 Conclusions

In this paper we presented two elements of autonomous driving for highway roads. The first elements addressed the path planning algorithms. The selection of the path planning algorithm is based on the type of the surrounding environment. For static environments, the algorithm is fed any constraints or obstacles and they are considered to be constant during run-time. We introduced two static path planning algorithms, Dijkstra and A-Star search algorithms. For dynamic path planning, the algorithms perpetually re-routes the autonomous agent as the environment is constantly changing throughout run-time. The updated data about the new constraints and variables are used for faster future planning. We introduced two dynamic path planning algorithms, D-Star and RRTs search algorithms.

The motion control system is fed all the decisions made by the path planning algorithm to reflect the changes introduced by the algorithm to accommodate the input changes provided by the environment in real-time manner. We introduced two motion control systems in this paper. The first motion control system is obstacle detection and avoidance control system. It is responsible for signalling the path planning algorithm about obstacles existing in the environment for rerouting. The path planning algorithm uses the second motion control system, which is the lane change and speed control system to reflect the input signals from the obstacle detection and avoidance control systems.

References

[1] Bimbraw, K. (2015). Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology.

[2] Levinson, J., Askeland, J., Becker, J., Dolson, J., Held, D., Kammel, S., Kolter, J.Z., Langer, D., Pink, O., Pratt, V., Sokolsky, M., Stanek, G., Stavens, D., Teichman, A., Werling, M. and Thrun, S. (2011). Towards fully autonomous driving: Systems and algorithms.

[3] Beiker, S.A., 2012. Legal aspects of autonomous driving. Santa Clara L. Rev., 52, p.1145.

[4] Yao, Z. and Ma, L. (2010). A Static Environment-Based Path Planning Method by Using Genetic Algorithm.

[5] Felner, A. (2011). Position Paper: Dijkstra's Algorithm versus Uniform Cost Search or a Case Against Dijkstra's Algorithm.

[6] Fadzli, S.A., Abdulkadir, S.I., Makhtar, M. and Jamal, A.A. (2015). Robotic Indoor Path Planning Using Dijkstra's Algorithm with Multi-Layer Dictionaries. 2015 2nd International Conference on Information Science and Security (ICISS).

[7] Cui, S.-G. ., Wang, H. and Yang, L. (2012). A simulation study of A-star algorithm for robot path planning.

[8] Karur, K., Sharma, N., Dharmatti, C. and Siegel, J.E. (2021). A Survey of Path Planning Algorithms for Mobile Robots. Vehicles, 3(3), pp.448–468.

[9] Kadry, S., Alferov, G. and Fedorov, V. (2020). D-Star Algorithm Modification.

[10] Stentz, A. (1997). Optimal and Efficient Path Planning for Partially Known Environments. Intelligent Unmanned Ground Vehicles, pp.203–220.

[11] Tian, Y., Yan, L., Park, G.-Y., Yang, S.-H., Kim, Y.-S., Lee, S.-R. and Lee, C.-Y. (2007). Application of RRT-based local Path Planning Algorithm in Unknown Environment.

[12] LaValle, S.M., 1998. Rapidly-exploring random trees: A new tool for path planning.

[13] Otte, M. and Frazzoli, E., 2015. RRT^x Real-Time Motion Planning/Replanning for Environments with Unpredictable Obstacles. In Algorithmic Foundations of Robotics XI (pp. 461-478). Springer, Cham.

[14] Storms, J.G. and Tilbury, D.M. (2014). Blending of human and obstacle avoidance control for a high speed mobile robot.

[15] Huang, C., Naghdy, F. and Du, H. (2016). Model predictive control-based lane change control system for an autonomous vehicle.

[16] Hatipoglu, C., Ozguner, U. and Redmill, K.A. (2003). Automated lane change controller design. IEEE Transactions on Intelligent Transportation Systems, 4(1), pp.13–22.