Updated Wavefront Planning Algorithm for Ackerman Steering System and its Comparision to Sampling-based planners.

Algorithmic Motion Planning Final Project

Kedar C. More  
*Aerospace Engineering*   
*University of Colorado, Boulder*Boulder, U.S.A  
kemo6996@colorado.edu

*Abstract*—This paper introduces an updated version of the existing Wavefront algorithm with path-smoothing for Ackerman Steering System. Most of the autonomous vehicular systems have Ackerman Steering Mechanism. So, most of the research is done on this mechanism. The proposed algorithm is complete as opposed to the sampling algorithms which are probabilistic complete.

Keywords—Wavefront algorithm, RRT (Rapidly exploring Random Trees), Ackerman Steering System, Informed Trees.

# Introduction

## Ackerman Steering Mechanism

A picture containing text

Description automatically generated

Figure. 1: Ackerman Steering

Ackerman mechanism [1] is used in most of the cars as it come the closest to the ideal steering model. Ideal steering model ensures that the wheels do not experience any lateral friction at any steering angle.

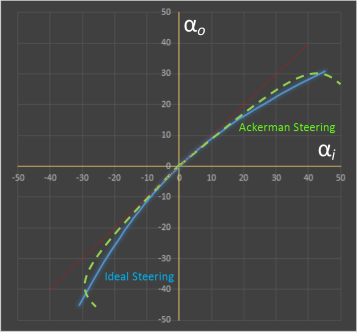


Figure. 2: Ideal Steering vs Ackerman Steering

Here and are the actual angles of the wheels. After 30 degrees of turn, the Ackerman system start deviating from the ideal steering. But in practical situations this angle is more than enough for the car.

Geometry: It is a four-bar link mechanism with on link as ground and two of the links connected to the ground have equal lengths. These links have the wheels turning with them at the same angle. The remaining link is used as the control input from the steering wheel and is connected to the rack and pinion.

## Sampling-based Algorithms

Most common Sampling-based Algorithm is RRT (Rapidly Exploring Random Tree) [2] and its variants like RRT\*, RRT-connect. After RRT many algorithms were developed owing to the ease of their usage and low computation required to run them. Informed Trees are the latest addition to these advancements and are used widely.

Text, letter

Description automatically generated

Figure. 3: RRT Algorithm

First, we initialize the tree with the start point as a node. Then a valid random point is selected in the workspace as xrand. The node that is nearest to the point is selected as xnear. A control input is selected which will take the robot closest to xrand from xnear. The state at which the robot reaches is termed as xnew. A new vertex xnew is added to and also an edge (xnear,xnew) with the control input . This process is run K times so that the tree grows out from the start node.

## Kinematic Planner

The difference between normal path planning and kinematic path planning is that the actual path depends on the control inputs available to the robot. Choosing a state does not imply that the state is achievable. This heavily affects the actual path length and the predicted one.

# Wavefront Planner

For this paper we will consider tree workspaces for comparison and benchmarking.

These are the configuration spaces of some Workspaces in 2D.

Chart, scatter chart

Description automatically generatedChart

Description automatically generated

Figure 3: Workspace 1 Figure 4: Workspace 2

Chart, histogram

Description automatically generated

Figure 5: Workspace 3

Here the obstacles are represented by the black space and the white space is the free area where the robot is allowed to move. The red dot is the starting point and the green one is the goal.

The Wavefront Planner is a complete algorithm and is optimal with respect to the grid. It uses a grid-based workspace to calculate a path from start to goal.

First the workspace is divided into grid of a certain size. The values of each grid is updated as follows:

1. Cells with obstacles is assigned the value 1.
2. Assign start cell value to 2.
3. Assign goal cell value to 3.

Update the cell values starting from the goal. Update all the neighboring cells to the value one more than the current cell. Continue this till one of the neighboring cells reaches the start cell with value 2. After that create a path by choosing the neighbors which has a value one less than the current cell. This will automatically reach the goal cell with value 3.The neighbors are described by the cell who share a facet with the other cell.

e.g.

Histogram

Description automatically generated

Figure 6: Applying Wavefront Planner on Workspace 2

Here the darker squares (purple) represent the grids with lesser value and in turn the lighter squares (yellow) represent the grids with grids with higher value.

# Updating the Wavefront Planner

The wavefront planner is already an optimal planner, but it can be improved. In this paper two methods are applied on the existing planner to make it better. The metric for betterment of the planner is considered to be length of the path and the computation time.

The proposed methods are:

1. Changing the definition of Neighbor cells:

Instead of using the neighbor cells as the ones sharing a facet, the cells sharing a facet as well as point are considered as neighbors. This will enable the robot to move to the diagonal cell which can reduce the path length.

A picture containing diagram

Description automatically generated

(a) (b)

Figure 7: Comparison of path on different kind of neighbors (a) sharing a facet and (b) sharing a facet and point

The grid in this example has the sides of length a. Here the red cell and green cell represents the start and goal respectively. In (a) the total distance of the path is 2\*a and in (b) the total distance of the path is \*a. In a best case scenario the final path will be reduced by a factor of .

1. Choosing the best Neighbor.

In the original planner the neighbors for updating the path are chosen randomly. This might lead to a sub-optimal path for the robot. This method proposes that there can be a cost assigned to each cell and the cell with least cost can chosen for updating the path.

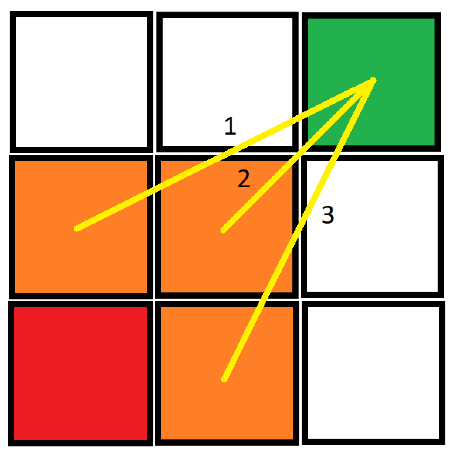


Figure 8: Cost of the cells with respect to the goal

Here the three yellow lines represent the cost of the respective cells. The cost chosen is the Euclidian distance to the goal. Here the Euclidian Distance is defined as the distance between the centers of the cells. We have to choose the cell with the least cost. In this case the path 2 is selected from the start node.

After these updates are applied on the planner, we get the following results for the workspaces.

Chart

Description automatically generated

Figure 9: Updated path for Workspace 1

Chart, histogram

Description automatically generated

Figure 10: Updated path for Workspace 2

A picture containing histogram

Description automatically generated

Figure 11: Updated path for Workspace 3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Workspace 1 | | Workspace 2 | | Workspace 3 | |
|  | Original | Updated | Original | Updated | Original | Updated |
| Path Length |  |  |  |  |  |  |
| Computation Time |  |  |  |  |  |  |

Table 1: Comparison between the original and updated Wavefront planner

# Path Smoothning

The path we get from the updated Wavefront planner is with respect to the grid, so it has sharp 45 degree turns which might be very difficult to maneuver with the Ackerman steering mechanism. For this a path smoothing technique is applied on the initial discrete path to make it easier for the Ackerman controller to reach the goal.

There are two part for this technique:

1. Eliminate the redundant points on the path.

Text

Description automatically generated

Here the main concept used is that we can remove extra nodes if we can reach the next node directly from the previous one. This will reduce the path length by the triangle sides rule. It states that the sum of the two sides of a triangle is always greater than the third side.

1. Introduce curves to the segments of the path.

A picture containing text

Description automatically generated

Text, email

Description automatically generated

This will give the list of points on which the Ackerman mechanism can be easily maneuvered. This algorithm makes sure that the turns are continuous and there are no sudden jerks while turning.

Chart

Description automatically generated

Figure 12: Final Smooth path for Workspace 1

Graphical user interface, chart

Description automatically generated

Figure 13: Final Smooth path for Workspace 2

A picture containing chart

Description automatically generated

Figure 14: Final Smooth path for Workspace 3

# Kinematics for Ackerman Steering

Radius of Curvature:

Here y is a function with respect to x. R is the list of radii of the curve at every x. This radius enables us to determine the turning radii of the wheels. And as the functions are continuous the turning of the car will be smooth.

Diagram

Description automatically generated

Figure 15: Radius of curvature of the path and the corresponding steering angle.

Here R represents the radius of curvature of the path and is the angle turned by the wheel in the bicycle model. For the Ackerman steering system, the two front wheels turn at different angles. The car dimensions are considered to be length = and width = . The following calculations derive the value of the two angles.

Similarly, for the left wheel:

And for the right wheel:

The inputs given to the car will be the turning radii and the velocity. To calculate the maximum possible velocity achievable by the car without slipping is by equating the maximum possible friction to the centrifugal force acting on the car. This is due to the directions of these forces being opposite to each other. The friction resists the car to move lateral where the centrifugal force to pushing the car. The centrifugal force on the car depend on the velocity of the car and the radius of curvature of the path. Maximum friction can be calculated by the product of the coefficient of friction between the road and the wheels and the weight of the car where is the mass of the car and is the acceleration due to gravity.

The car can be accelerated upto this linear velocity given the coefficient of friction of the ground and the radius of curvature of the road.

# Sampling based Methods for Ackerman Steering

The recently developed sampling-based method like AIT\* and ABIT\* are used on the NASA/JPL-Caltech’s Axel Rover System which is a differential drive system. The C-space of the kinematic control system is 2D (input for two wheels). But for Ackerman system using the two value boundary problems additional dimensions are introduced. This cannot be handled by these algorithms efficiently. In other case simple RRT can be used in more than 2D spaces with little or no difficulty.

In the paper [3] RRT is used to plan the Ackerman steering model. It presents Gaussian sampling-based strategy, path formulation meeting system kinematics, and trajectories

creation over the sample points.

Running RRT on the following workspaces give the following results.

A picture containing chart

Description automatically generated

Figure 16: RRT on Workspace1

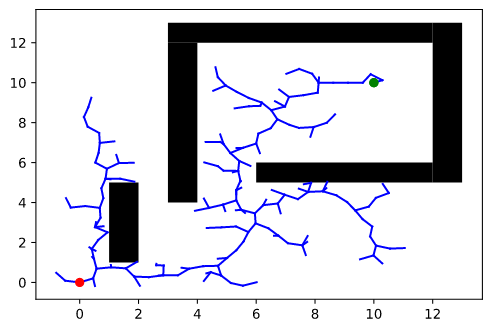


Figure 17: RRT on Workspace2

Chart, histogram

Description automatically generated

Figure 18: RRT on Workspace3

Kinematic Planning:

Here the Ackerman model is approximated by a unicycle maodel. When we steer to xnew, theoretically it gives a straight line form the xnearest to xnew. But when we give control inputs with the two value boundary problem, the paths is approximated.

Diagram, histogram

Description automatically generated

Figure 19: Two Value Boundary Problem (Shooting Method)

Shooting Method is used in this algorithm to solve the two value problem by converting it to single value problem. It is an iterative method using the first order derivative and the end point will reach as close to the next point on the path as possible. This will lead to a smooth path which can be traversed by an Ackerman system. In this paper the back wheel drive system is used where the equations of motion are:

Where, is the orientation of the vehicle and is the input steering angle.

# Benchmarking

# References

1. Kamner, Haim J. "Combined articulated and ackerman steering system for vehicles." U.S. Patent No. 3,515,235. 2 Jun. 1970.
2. LaValle, Steven M. "Rapidly-exploring random trees: A new tool for path planning." (199)
3. Acharya, Rajaneesh, and Debashisha Jena. "Sampling based motion planning of Ackermann steering system using transformation." *2018 IEEMA Engineer Infinite Conference (eTechNxT)*. IEEE, 2018.