Report on Dubins Path Algorithm

by

Pranav Natu

A report presented in partial fulfillment of the course
MEEN 612- Mechanics of Robot Manipulators

Dr. Swaroop Darbha
Department of Mechanical Engineering
Texas A and M University
6 December 2019

Contents

1 Introduction 2

2 Dubins Path with Known Heading 3
2.1 CCC Path . . . 3
2.2 CSC Path . . . 4

3 Dubins Path with Unknown Heading 5
3.1 Obtaining the optimal sequence . . . 5
3.2 Obtaining heading angles . . . 5
3.3 Obtaining the path and costs . . . 6

4 Improvisations in the Algorithm 7
4.1 Increasing Discretizations . . . 7
4.2 Checking the reverse path . . . 7

5 Conclusion 8

6 References 9

1 Introduction

Dubins path is the shortest path connecting two points with constraints on the radius of curvature and that the vehicle can only travel in the forward direction. Thus, the path is a combination of curves and straight lines such that the path length is minimized.

In the first part of the problem, we have selected an array of ten random points in the xy plane, with the first and the last points being the source and the destination. The heading angles of all the points are known. The objective is to find the sequence of the remaining points to minimize the cost, that is, the euclidian distance between two points, and then to construct an optimal Dubins path. There are two variations in the path constructed- a path with only curves(CCC), and a path with curves and straight lines(CSC).

The second part of the problem is similar to the first part with the exception that the heading angles are unknown. They have to be computed using numerical simulations such that the Dubins path is optimized. Also the sequence of the points to be visited has to be obtained such that the path is optimized

2 Dubins Path with Known Heading

The 10 points and the heading angles chosen for the path are given by the folowing matrix. The columns are the x,y and θ(heading angle). The first row is the start and last row is the destination.

$$\text{waypoints=}\begin{bmatrix} 0 & 0 & 0 \\ 2 & 3 & pi/2 \\ 5 & 8 & pi/4 \\ 8 & 10 & pi/8 \\ 10 & 13 & -pi/2 \\ 20 & 15 & -pi/6 \\ 22 & 18 & -pi/4 \\ 26 & 22 & 0 \\ 27 & 28 & pi/4 \\ 30 & 30 & pi \end{bmatrix}$$

The points are visited in the order in which they are expressed in the matrix.

2.1 CCC Path

In this path the Dubins vehicle is not allowed to follow straight lines and only travels along curved paths.
The following figure is the path obtained-

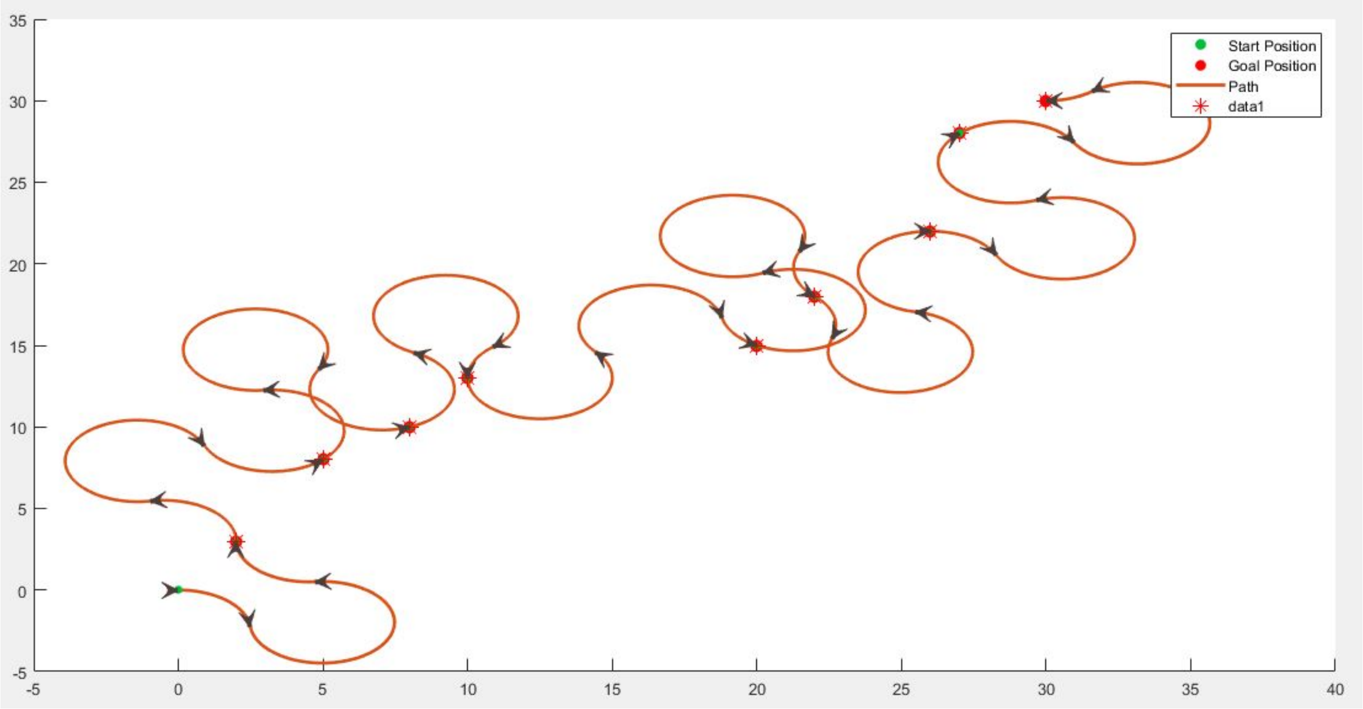

Figure 1: CCC Path

The cost obtained in this path is 196.54

2.2 CSC Path

In this path the Dubins vehicle is allowed to follow both straight lines and curved paths. The following figure is the path obtained-

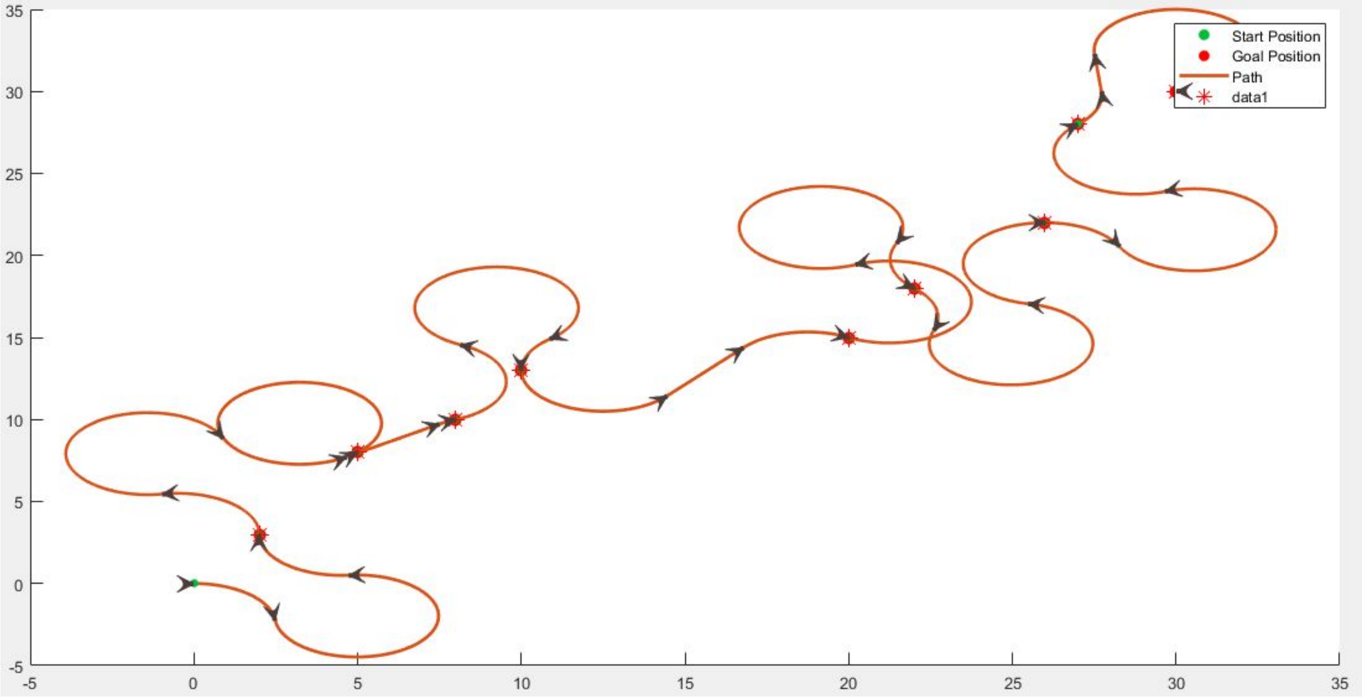

Figure 2: CSC Path

The cost obtained in this path is 180.74

3 Dubins Path with Unknown Heading

The 10 points and the heading angles chosen for the path are given by the folowing matrix. The columns are the x,y coordinates. The first row is the start and last row is the destination. The heading angles need to be found out. Also, the order in which the Dubins vehicle visits the points to minimize costs has to be found out.

$$\text{waypoints}=\begin{bmatrix} 0 & 0 \\ 2 & 3 \\ 5 & 16 \\ 2 & 4 \\ 10 & 16 \\ 20 & 13 \\ 25 & 12 \\ 15 & 24 \\ 20 & 20 \\ 30 & 30 \end{bmatrix}$$

3.1 Obtaining the optimal sequence

The order in which the Dubins vehicle visits the points has been found by a MATLAB function tspofga.m[1]. The rearranged order obtained is -

$$\text{waypoints}=\begin{bmatrix} 0 & 0 \\ 2 & 3 \\ 2 & 4 \\ 5 & 16 \\ 10 & 16 \\ 15 & 24 \\ 20 & 20 \\ 20 & 13 \\ 25 & 12 \\ 30 & 30 \end{bmatrix}$$

3.2 Obtaining heading angles

The algorithm begins with discretizing the heading angles of the first and the second point and finding the optimal angles for both the points such that the cost is minimized. The angle of the second point is fixed and the angle for the third point is found in the same way. The same procedure is repeated to get the resulting matrix with heading angles as-

$$\text{waypoints}=\begin{bmatrix} 0 & 0 & 1.00 \\ 2 & 3 & 1.00 \\ 2 & 4 & 6.22 \\ 5 & 16 & 1.57 \\ 10 & 16 & 4.02 \\ 15 & 24 & 0.38 \\ 20 & 20 & 5.15 \\ 20 & 13 & 4.65 \\ 25 & 12 & 2.14 \\ 30 & 30 & 1.26 \end{bmatrix}$$

3.3 Obtaining the path and costs

Now that we have all the information we construct the path with no constraints on CSC or CCC paths, we obtain the following path.The cost obtained is 117.44.

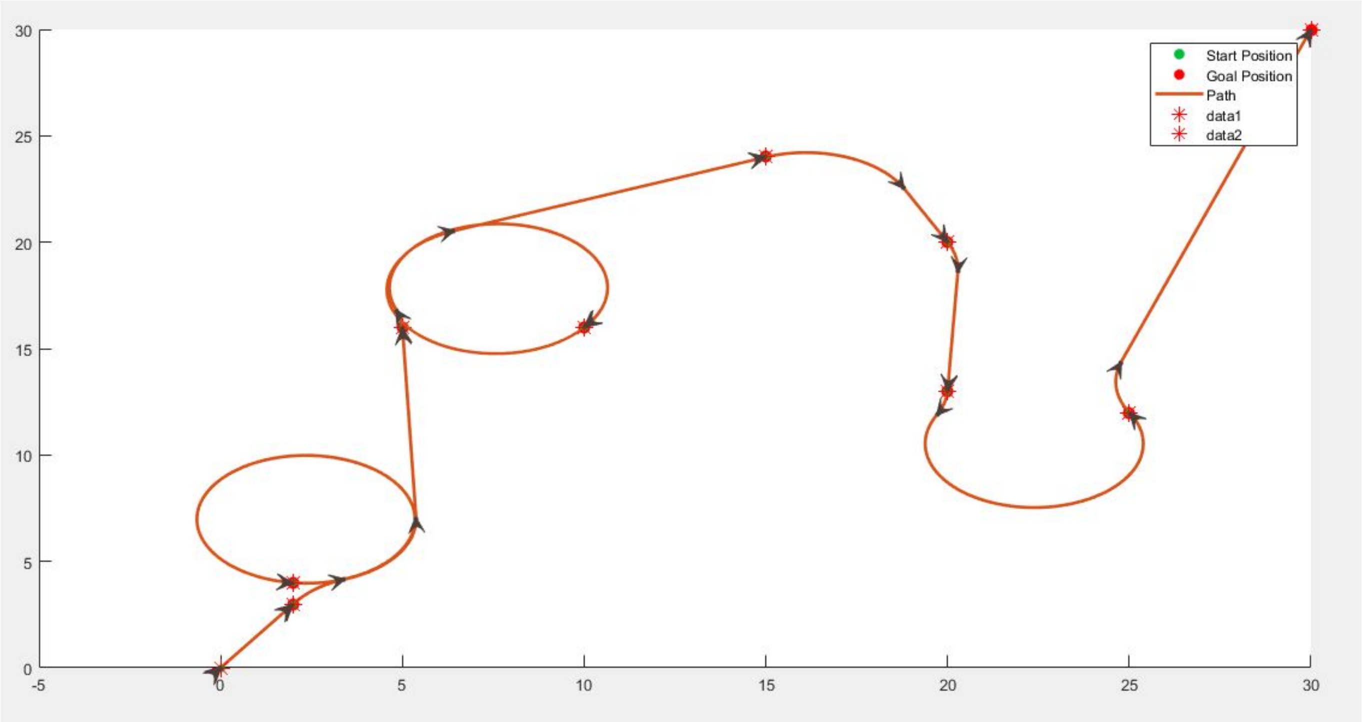

Figure 3: Dubins Path

4 Improvisations in the Algorithm

4.1 Increasing Discretizations

To improve the algorithm and get more accurate results, we can increase the number of units in which the angles are discretized provided the computer has the required processing power.

The following table gives the number of discretizations and the corresponding minimum cost.

Number of discretizations	Cost
20	118.27
50	117.76
100	117.44
200	117.34

4.2 Checking the reverse path

Since there is an asymmetry in the algorithm the costs obtained iterating from (0,0) to (30,30) will not be the same as that obtained vice versa. Thus, another way to improve the cost is to switch the entire sequence of points and follow the same procedure to calculate the costs. We can pick the heading angles of the path that gives the lesser cost among the two.

For the points that we have, the path in the reverse order, that is, from (30,30) to (0,0) gives the cost as 127.17. Therefore we pick the path from (0,0) to (30,30) and keep the heading angles the same. If the reverse path had a lesser cost, the heading angles would be reversed by 180 degrees. The important point to be noted here is that the source and destination remain the same.

5 Conclusion

It is observed that the cost in the CSC part of the problem is less than the CCC part. Also, a more optimal path can be constructed by calculating the heading angles at each point compared to the case where the heading angles are fixed.

The methods to improve the results are to increase the discretizations of heading angles and to try the reverse path to check for minimum costs. These are the possible refinements that can be done in the algorithm.

6 References

[1]https://in.mathworks.com/matlabcentral/fileexchange/21197-fixed-endpoints-open-traveling-salesman-problem-genetic-algorithm
[2]https://in.mathworks.com/help/driving/ref/driving.dubinspathsegment.html
[3]Texas A and M University MEEN 612-Lecture Notes