International Journal of Future Generation Communication and Networking
Vol. 13, No. 3, (2020), pp. 1987–1995

Path Trajectories in 2-Dimensional Motion Planning

Nilesh Hegde[1], Sumukha Nadig[1], Jyothi S Nayak[2]

[1]Department Computer Science and Engineering, BMS College of Engineering, Visvesvaraya Technological University.
*[2]Prof. Department Computer Science and Engineering, BMS College of Engineering,
Visvesvaraya Technological University.*

Abstract

In the presence of an obstacle, either static or dynamic, an autonomous guided vehicle needs to recalculate its path in order to avoid hitting such an obstacle. Motion Planning is first in order of importance in developing autonomous vehicles. Be it a drone, smart car or an underwater drone, forming and reconstructing path trajectories is a very important step. This paper discusses two of such curves used in motion planning, Dubins curve and B-Spline curves.

Keywords: *Motion Planning, Dubins Curve, B-Spline Curve, Path Trajectories.*

1. Introduction

Initially developed for military reconnaissance using drones, autonomous navigation is now booming commercially. The development in defense field has also been steady but commercial wise the growth has been exponential. In the automobile industry, many companies are investing millions and billions of dollars in developing technology to improve autonomous navigation. With the boom in drone industry many countries are revising their drone laws to accommodate the use them in public.

Pathfinding, also known as pathing, refers to finding path between two points many a times the shortest path. The two very well-known pathfinding algorithms are Dijkstra's algorithm and A* search algorithm. These two algorithms have been extensively used in research such as IP Routing, GPS systems and many more. These algorithms are widely used in the gaming industry as well. The extension of Dijkstra's algorithm is the A* algorithm. Apart from these two algorithms, Hierarchical Pathfinding A*(HPA*) and Lifelong Pathfinding A*(LPA*) have also been developed [1].

Motion Planning refers to the functionality of finding the path that needs to be taken by an autonomous object to advance from the initial position to the final position of travel without any human intervention. The path needs to be recalculated and trajectory changes in the presence of an external obstacle along its original trajectory. For example, for a drone, the innumerable number of objects such as building, lamp posts and so on cause it to recalculate its path in order to prevent itself from crashing into it. The same goes for an Autonomous Ground Vehicle such as smart car where the obstacle may be another car or the median strip. In case of Underwater Autonomous Vehicles, the oceanic crust and various fauna and flora become obstacles. Recalculating path using Euclidean distance is not fruitful. Thus, various mathematical curves are used to model these trajectories. B-Spline curves and Dubins curves have been extensively used in motion planning.

The Dubins curve has been made use of in many applications. It has been used in developing UAVs [2], robots [3] and more. B-Spline curve also has been used in development such as drones [4], collision avoidance in robots [5] and many more.

ISSN: 2233-7857 IJFGCN

International Journal of Future Generation Communication and Networking
Vol. 13, No. 3, (2020), pp. 1987–1995

2. Literature Review

There is enormous work which has been done related to motion planning using different techniques in the field of robotics and unmanned vehicles be it drones, underwater drones or autonomous cars. It focuses on techniques which can be used in planning path trajectories for motion of the object.

[1] Imants Zarembo, Sergejs Kodors, "Pathfinding Algorithm Efficiency Analysis in 2D Grid"

There are many path finding algorithms in existence. Some of the major ones are:

- A* algorithm
- BFS or Breadth First Search algorithm
- Dijkstra's algorithm
- HPA* or Hierarchical Pathfinding A*
- LPA* or Lifelong Planning A*

This paper compares the above algorithms on various criteria such as execution time, memory usage and more in a 2-dimensional grid environment.

[2] Israel Lugo-C´ardenas, Gerardo Flores, Sergio Salazar, Rogelio Lozano, "Dubins Path Generation for a Fixed Wing UAV"

A fixed wing UAV is one where the drone maintains the same altitude, has a constant airspeed and it is constrained with respect to the turning rate.
This paper presents the Dubins path generation in combination with Lypaunov-based path-following control. An application where a missing person is being located using an UAV following the given path generation technique is also discussed.

[3] Dongwon Jung, Panagiotis Tsiotras, "On-line Path Generation for Small Unmanned Aerial Vehicles Using B-Spline Path Templates"

The paper talks about smooth planar reference path generation given a family of optimal paths that are discreet. These paths are then incorporated in motion planning in vehicles.
The algorithm proposed in the paper in addition to D* lite algorithm provides an obstacle free path generation that is suitable for real time implementation.

3. Dubins Curve

The optimal path calculated by Dubins curve consists of no more than three basic steps:

SEGMENT	ACRONYM
Straight	S
Left	L
Right	R

TABLE 1. Acronym Definition

Given the initial and final position in a plane, and the direction of traversal, the curved trajectory can be calculated using Dubins curve. Dubins paths are of six types which are the combinations of the above three segments. The optimal path can be any of these six types. The six types of paths are as follows:

ISSN: 2233-7857 IJFGCN

ABBREVIATION	PATH
LRL	Left, Right, Left
RLR	Right, Left, Right
LSR	Left, Straight, Right
LSL	Left, Straight, Left
RSL	Right, Straight, Left
RSR	Right, Straight, Right

TABLE 2. Abbreviation Definition

The main condition for Dubins path is that the object can move in only one direction. For example, if an autonomous car is considered then the car can only move forward with its reverse functionality being reversed [6].

Let us consider an Autonomous car with the following variables:

- Absolute position of the car is (x, y)
- Moving with a constant speed 'v'
- Direction of movement with an inclination of 'θ' with the x-axis of the x-y plane
- Limited turning radius with circle of radius 'r'

The prime notation x' shows how the x coordinate changes over time. Similarly, y' and θ' show how y coordinate and direction changes with time [7]. The variables of motion are related as follows:

- $x' = v \cos(\theta)$
- $y' = v \sin(\theta)$
- $\theta' = v / \mathrm{r}$

Once the new values of these variables are calculated, they are updated as follows:

- $x_{new} = x_{prev} + x'$
- $y_{new} = y_{prev} + y'$
- $\theta_{new} = \theta_{prev} + \theta'$

If the object is allowed to move in two directions, then the algorithm would follow Reeds-Shepp Curve. The six paths can be visualized as follows:

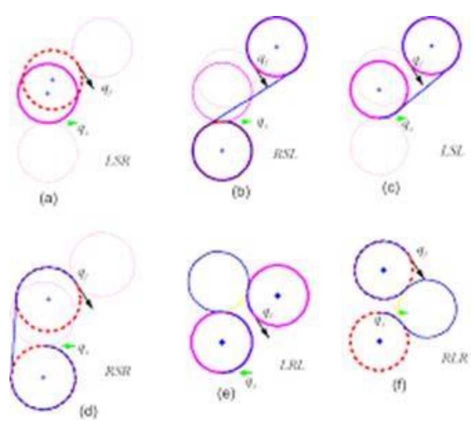

FIGURE 1: Visualization of various paths

ISSN: 2233-7857 IJFGCN

Let us consider the following examples:

- The initial position of the car is (2000,1000) and the initial direction is 0° while the final position of the car is (8000,10000) and final direction is 260°. The following path will be taken:

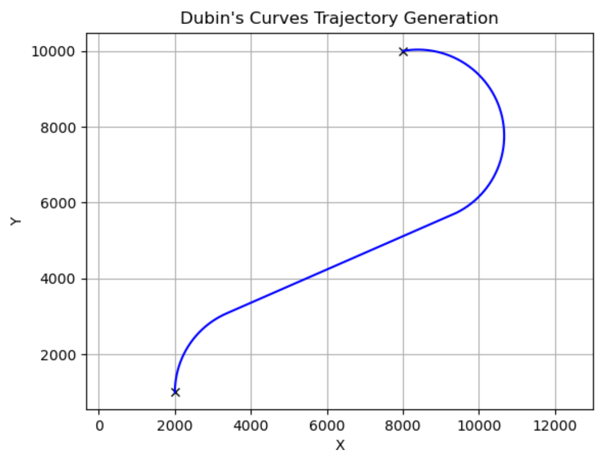

FIGURE 2: Dubins Curve Example 1 (RSL)

It follows the RSL trajectory as seen above.

- The initial position of the car is (1000,15000) and the initial direction is 180° while the final position of the car is (12000,2500) and final direction is 270°. The following path will be taken:

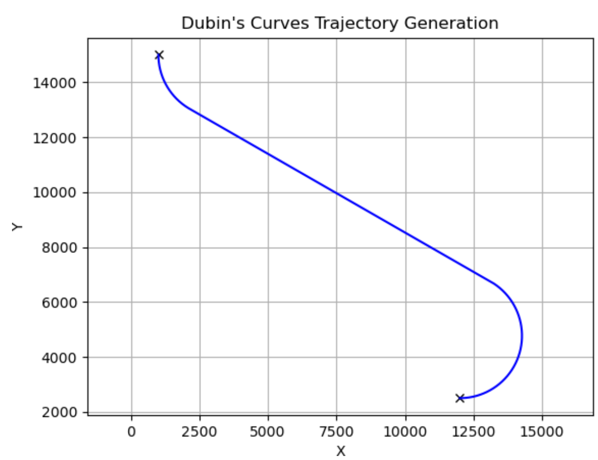

FIGURE 3: Dubins Curve Example 2 (LSR)

It follows the LSR trajectory as seen above.

- The initial position of the car is (2000,8000) and the initial direction is 120° while the final position of the car is (3000,4000) and final direction is 45°. The following path will be taken:

ISSN: 2233-7857 IJFGCN

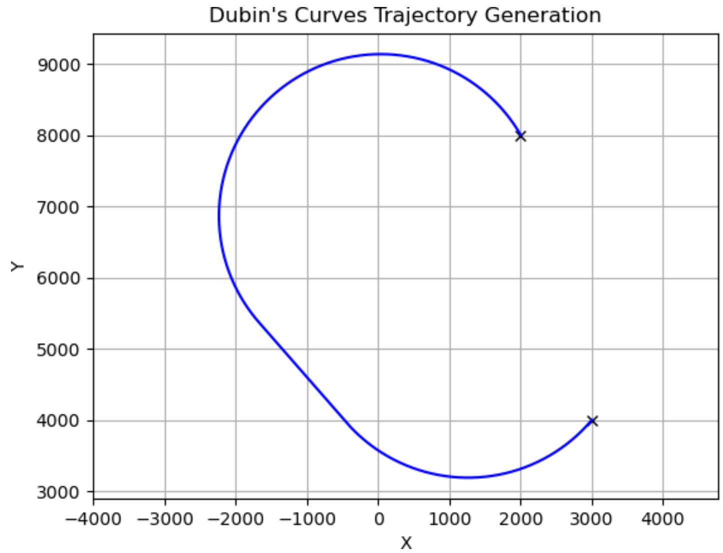

FIGURE 4: Dubins Curve Example 3 (LSL)

It follows the LSL trajectory as seen above.

- The initial position of the car is (4000,4000) and the initial direction is 45° while the final position of the car is (9000,4000) and final direction is 130°. The following path will be taken:

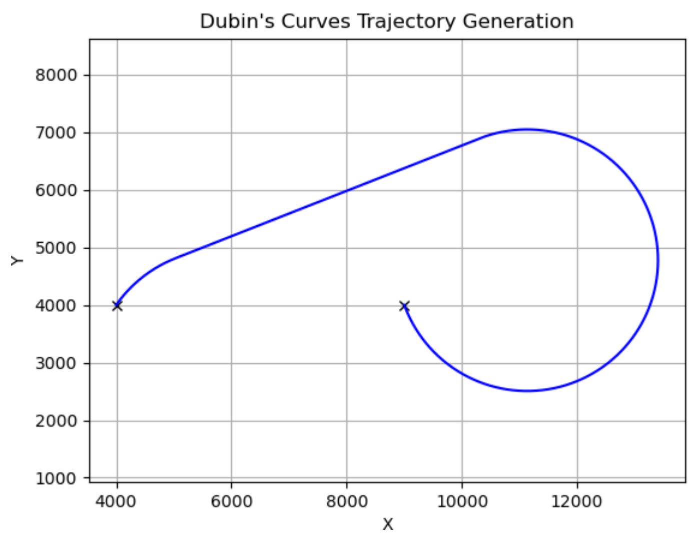

FIGURE 5: Dubins Curve Example 4 (RSR)

It follows the RSR trajectory as seen above.

- The initial position of the car is (8000,9000) and the initial direction is 35° while the final position of the car is (8000,800) and final direction is 315°. The following path will be taken:

ISSN: 2233-7857 IJFGCN

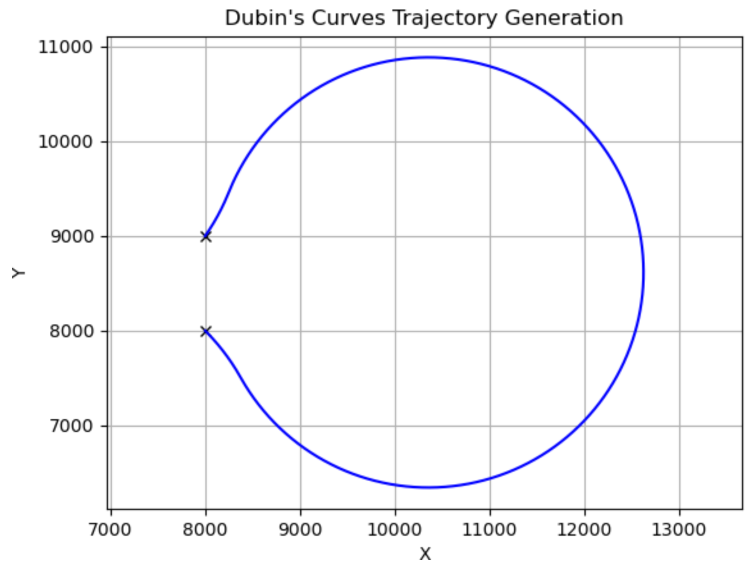

FIGURE 6: Dubins Curve Example 5 (LRL)

It follows the LRL trajectory as seen above.

- The initial position of the car is (2000,2000) and the initial direction is 45° while the final position of the car is (4000,6000) and final direction is 330°. The following path will be taken:

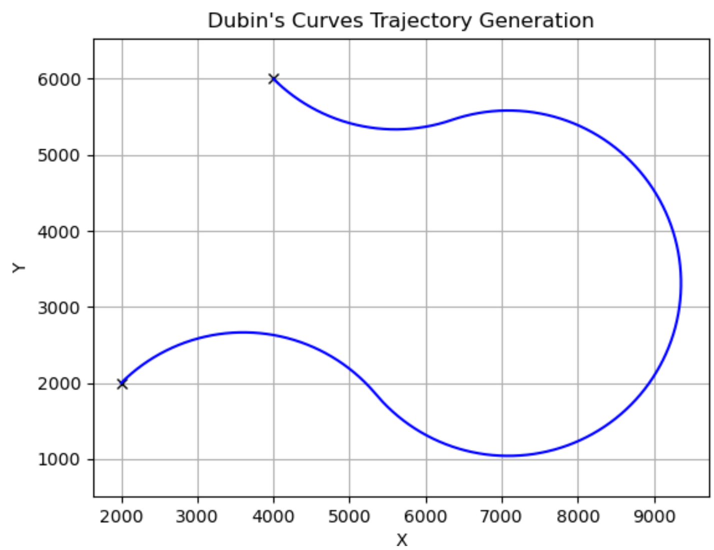

FIGURE 7: Dubins Curve Example 6 (RLR)

It follows the RLR trajectory as seen above.

ISSN: 2233-7857 IJFGCN

4. B-Spline Curve

B-Spline curves are related to Bezier curves. In a Bezier curve with ' $n+1$' control points, the curve is of degree 'n', that is, the equation can be written as:

$$B_{curve}(x) = a_n x^n + \ a_{n-1}x^{n-1} + \cdots + a_1 x + a_{n0}$$

In Bezier Curve, the degree of the polynomial representing the curve cannot be controlled and is fixed. B-Spline curve removes this drawback. These are determined using segments.

Characteristics of B-Spline curve is as follows:

- B-spline curve consists of '$n+1$' control points.
- Order of B-Spline curve is 'k'.
- The curve exhibits local control that is if a control point changes the only the segment dependent on that point changes rather than the entire curve.
- The degree of the polynomial representing the curve is dependent on the order of the curve 'k'. If order of the is 'k' then degree of the polynomial is '$k-1$'.
- The value of 'k' lies between 2 and 'n', that is:
$$2 \le k \le n$$
- The number of segments is '$n-k+2$'.

The x-coordinate of the B-Spline curve is given by [8-9]:

$$x(u) = \sum_{i=0}^{n} N_{i,k}\,(u)x_i$$

where, $0 \le u \le n-k+2$

Similarly, the y coordinate is given by:

$$y(u) = \sum_{i=0}^{n} N_{i,k}\,(u)y_i$$

Here, function 'N' is B-Spline Basis Function. It is defined in the following way:

$$N_{i,k}(u) = \frac{(u-t_i)N_{i,k-1}(u)}{t_{i+k-1}-t_i} + \frac{(t_{i+k}-u)N_{i+1,k-1}(u)}{t_{i+k}-t_{i+1}}$$

Where $t_i(0 \le i \le n+k)$ are called knot values. The values of t_i varies as follows:

$$t_i = \begin{cases} 0 & \forall\ i < k \\ i-k+1 & \forall\ k \le i \le n \\ n-k+2 & \forall\ i > n \end{cases}$$

And, $N_{i,k}(u) = \begin{cases} 1, & t_i \le u < t_{i+1} \\ 0, & otherwise \end{cases}$

For example, let the number of control points be 6 whose coordinates are (0,0), (1,3), (3,5), (5,4) and (6,1). Let order be 3 which implies degree is 2. The B-Spline curve for the above defined variables is:

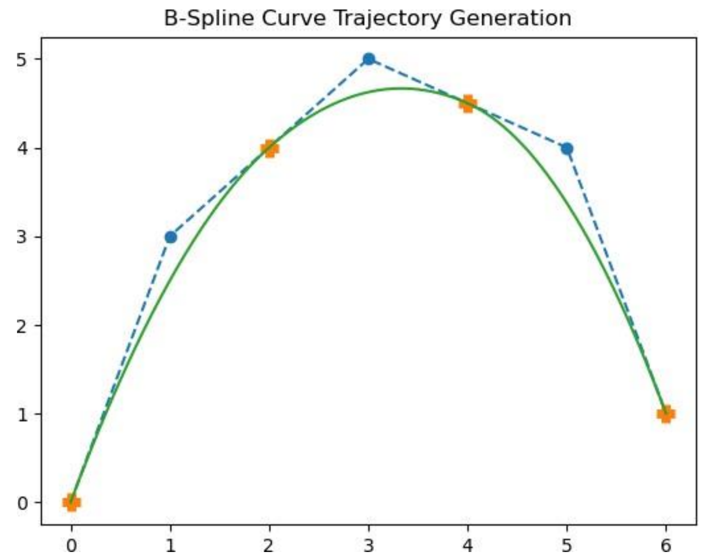

Figure 8: B-Spline Curve Example

In Figure 8, the blue points are control points, the blue dashes form control hexagon, the green line is the B-Spline curve and the orange spots demarcate the different segments.

5. Conclusion

The Dubins curve and B-Spline curves have their own advantages and disadvantages. The Dubins curve is relatively easier to understand and due to small list of possible paths, that is 6, it is easy to construct a Dubins curve in comparison to a B-Spline curve. The main disadvantage of Dubins curve is that the motion can be in only one direction, that is, the object can be moving either forward or backward. If the object shows both the functionality, then Reeds-Shepp curve has to be used.

The B-Spline curve on the other hand allows one to modify the curve as required as the degree of the curve can be changed to the requirements. The curve can quadratic, cubic, quartic and so on. The major disadvantage is the extensive and intricate calculations that go into forming the curve.

In case of collision detection, the intermediate points in B-Spline curve are considered as objects that form obstacles and in Figure 8, as seen the B-Spline curve avoids them. In Dubins curve, the detection is done in real time. As soon as an obstacle is detected, a new shortest curved path with the current position as the initial position and the actual final position is found by calculating all the six possible paths and the shortest length path is chosen.

References

[1] Zarembo, Imants & Kodors, Sergejs. (2015). Pathfinding Algorithm Efficiency Analysis in 2D Grid. Environment. Technology. Resources. Proceedings of the International Scientific and Practical Conference. 2. 46. 10.17770/etr2013vol2.868

[2] Zhu, Moning & Zhang, Xuehua & Luo, He & Wang, Guoqiang & Zhang, and. (2020). Optimization Dubins Path of Multiple UAVs for Post-Earthquake Rapid-Assessment. Applied Sciences. 10. 1388. 10.3390/app10041388.
[3] Hameed, Ibrahim. (2017). Path Planning for Line Marking Robots Using 2D Dubins' Path. 10.1007/978-3-319-48308-5_86.R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.
[4] Jung, Dongwon & Tsiotras, Panagiotis. (2013). On-Line Path Generation for Unmanned Aerial Vehicles Using B-Spline Path Templates. Journal of Guidance Control and Dynamics. 36. 10.2514/1.60780.
[5] W. Fu, H. Hadj-Abdelkader and E. Colle, "Reactive collision avoidance using B-spline representation: Application for mobile robot navigation," 2013 European Conference on Mobile Robots, Barcelona, 2013, pp. 319-324.
[6] Manyam, Satyanarayana & Casbeer, David & Von Moll, Alexander & Fuchs, Zachariah. (2019). Shortest Dubins path to a circle. 10.2514/6.2019-0919.
[7] Lugo-Cárdenas, Israel & Flores, Gerardo & Salazar, Sergio & Lozano, Rogelio. (2014). Dubins path generation for a fixed wing UAV. 339-346. 10.1109/ICUAS.2014.6842272.
[8] Rashidinia, Jalil & Sharifi, Sh. (2012). Survey of B-spline functions to approximate the solution of mathematical problems. Mathematical Sciences. 6. 10.1186/2251-7456-6-48.
[9] Park, Hyungjun & Lee, Joo-Haeng. (2007). B-spline curve fitting based on adaptive curve refinement using dominant points. Computer-Aided Design. 439-451. 10.1016/j.cad.2006.12.006.