3-Dimensional Path Planning for Autonomous Underwater Vehicle

Yiheng Wang[1] and Yahong Rosa Zheng[2]

[1] Dept. of ECE, Missouri University of Science & Technology, Rolla, MO 65409
[2] Dept. of ECE, Lehigh University, Bethlehem, PA 18015

***Abstract*—This paper proposes a new algorithm for Autonomous Underwater Vehcle (AUV) path planning in 3D space to visit multiple targets using Dubins curves. For a given target-sequence, the 3D path planning is usually solved by two steps: Step 1 projects 3D targets to the X-Y plane and designs a 2D path on this plane; Step 2 maps the 2D path into 3D via interpolation. The proposed new algorithm defines a local coordinate system (LCS) for each pair of targets and designs the 2D Dubins curve in the LCS, then uses the Euler's rotation transformation to convert the 2D dubins curve into the 3D global coordinate system (GCS). Applying the proposed rotation method to a given target sequence and given incoming-outgoing angles yields 3D Dubins path with guaranteed G2 continuity at the joints of two Dubins curves. The proposed method is compared with the interpolation method and Bezier curve method. Computer simulations demonstrate that the proposed algorithm provides better G^2 continuity in 3D space or shorter path lengths than the existing linear or spline interpolation methods and Bezier curves.**

***Index Terms*—Autonomous Underwater Vehicles, AUV, Path Planning, 3D Dubins Curve, Bezier Curve, Geometric Continuity, Back-Propagation Algorithm.**

I. Introduction

Autonomous Underwater Vehicles (AUVs) have been widely applied in oil and gas industry, ocean exploration, environmental monitoring, underwater infrastructure monitoring and underwater data collection [1]–[3]. In underwater wireless sensor network (UWSN), multiple AUVs are employed to collect data from predetermined targets via acoustic communication [4]. However, due to the limited range and bandwidth of acoustic communication and high energy cost of sensor node, the AUV-Aided Underwater Routing Protocol in [4] still has a lot of limitations in data collecting. Recently, the Magneto-Inductive (MI) communication has the advantages of low-cost and easily-deployable [5]. Therefore, the AUV can be utilized to complete the data collection by visiting multiple sensor nodes via MI communication. Hence, path planning problems in two and three dimensional space have attracted many research attention [6], [7].

With the given target-sequence, path planning algorithm design a smooth path for the AUV to visit all the targets. Some existing works focus on 2D space and design smooth path in 2D only [8]. 2D point-to-point smooth paths in X-Y are often designed via Dubins curves [8], [9], Bezier curves or other curves to accommodate the dynamic constraints of AUVs. Smooth paths in 3D are often designed by mapping 2D curves into 3D via interpolation [7], [10]. However, the linear interpolation method [7] fails to meet with G^1 continuity at multiple targets. The spline interpolation method [10] may result in much longer total distances between targets in 3D space.

In this paper we proposes a new algorithm to solve 3D path planning without interpolation methods. This algorithm consists of two functions: Function 1 designs a 3D Dubins path for a given target-sequence and incoming-outgoing angles by a rotation method. Function 2 utilizes the back-propagation method to choose the shortest path from all possible incoming-outing angles. With the target-sequences, we apply function 1 and Function 2 to design 3D Dubins path for the AUV. In this paper, we design 3D Dubins path without interpolation methods. The rotation based 3D Dubins path design method achieves better continuity than the linear or spline interpolation method, and the new method has shorter total distance than spline interpolation method.

In practice, different types of AUVs have different motion constraints. For example, the Autonomous Benthic Explorer (ABE) equipped with five thrusters can move in any direction, and can hover and reverse [11]. In contrast, the survey-type AUVs have many constraints such as finite navigation distance, stringent non-holonomic motion constraints, and no direction reversing. In particular, the non-holonomic motion constraint requires that the vehicle moves along a smooth path with bounded curvatures and geometric continuity to support their kinematic constraints [12]. Besides motion constraints of AUV, the ocean environmental conditions will effect the movement of AUVs, such as the strong ocean current [13]. In this thesis, we focus on the strong motion constraints in our multi-AUV path planning problem where geometric continuity is required without considering the ocean environmental conditions.

II. System Model and Motion Constraints

We consider one AUV to visit multiple underwater targets in a 3D underwater environment, where a set of N targets $\mathcal{T} = \{T_1, T_2, \cdots, T_N\}$ are pre-assigned to the AUV and will be visited sequentially by the AUV. The AUV sets off from the origin target and returns back to the same target after visiting all targets in the assignment. Each target will be visited by the AUV exactly once.

The famous REMUS AUV model created by Prestero [14] describes the six degrees of freedom (DOF) as *surge, sway, heave, pitch, roll, yaw.* In this paper, we simplify the dynamics model by only considering the position and heading of the AUV, as shown in Fig. 1, where the AUV in the Local Coordinate System (X, Y, Z) has a heading direction $\Phi(\theta, \varphi)$, with θ and φ being the azimuth and elevation angles, respectively.

978-1-5386-4814-8/18/$31.00 ©2018 IEEE

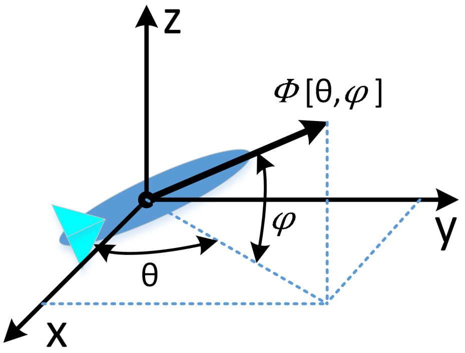


Fig. 1: AUV heading direction defined in the Local Coordinate System (LCS)

The survey-type AUV has nonholonomic constraints which require the path of AUV to have bounded curvature. The AUV is prohibited to take sharp turns or make sudden stops. Therefore, the derivative of AUV heading direction has to satisfy:

$$\dot{\Phi} = \psi, \quad \psi \in [-\psi_a, \psi_a] \tag{1}$$

where the dot operator is the first derivative with respect to θ and φ, and ψ_a is the curvature bound.

In addition, the nonholonomic constraints require that the AUV path satisfy geometric continuity. For example, the G^0, G^1 and G^2 continuities are defined as follows [12]: let $P(u) = [x_1(u), y_1(u), z_1(u)]$ and $Q(v) = [x_2(v), y_2(v), z_2(v)]$ be two parametric curves in the 3D space, where $u \in [a, b]$ and $v \in [c, d]$.

G^0 *Continuity*: If $P(b) = Q(c) = J$, then the two curves meet at the joint point with G^0 continuity.

G^1 *Continuity*: If G^0 continuous and $\dot{P}(u)|_{u=b} = \dot{Q}(v)|_{v=c}$, then the two curves meet at the joint point with G^1 continuity.

G^2 *Continuity*: If G^1 continuous and $\ddot{P}(u)|_{u=b} = \ddot{Q}(v)|_{v=c}$, then the two curves meet at the joint point with G^2 continuity.

The goal of the path planning algorithm is to design a smooth curve for the AUV to visit all targets with a shortest total length while satisfying the continuity constraints.

III. The Proposed Algorithm for 3D Dubins Path Design

This section proposes the new design algorithm for 3D Dubins path based on Euler's transformation. Consider two targets T_a and T_b defined in a Global Coordinate System (GCS) u-v-w, as shown in Fig. 2. The new algorithm first defines a local coordinate system $LCS1$ (x-y-z) that contains the two targets and the incoming heading of T_a, then designs the 2D point-to-point smooth Dubins path from T_a to T_b in the X-Y plane of $LCS1$. The 2D Dubins curve is then transformed to the 3D path in the GCS using Euler's transformation. Once the 3D Dubins segment between T_a and T_b is designed, the $LCS2$ centered at T_b is used in a similar manner to design the next segment of the 3D Dubins path.

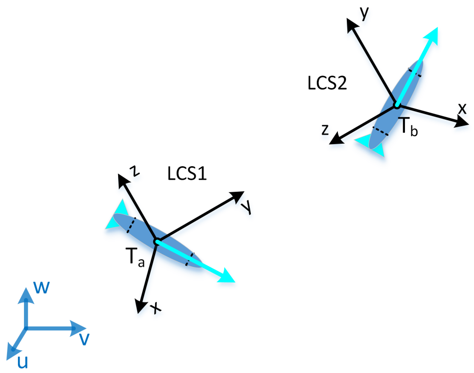


Fig. 2: Global coordinate system (GCS) and two local coordinate systems: $LCS1$ and $LCS2$

A. *Coordinate System Rotation Algorithm*

To facilitate the design of the 3D Dubins curve between two targets T_a and T_b, we first shift the origin of the GCS to the location of target T_a, and denote the shifted GCS as GCS' (u'-v'-w'). The Local Coordinate System $LCS1$ (x-y-z) of T_a is defined by the heading vector O_a and the line linking T_a with T_b, as shown in Fig. 3. The y axis lays on the vector connecting T_a and T_b, the x-y plane contains the vector O_a, and the z axis is perpendicular to the x-y plane with its direction following the right-hand rule.

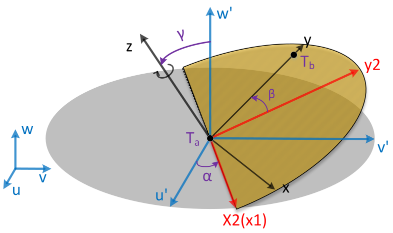


Fig. 3: GCS and LCS and rotations and three Euler transformation angles α, β, and γ

The rotation between $LCS1$ and GCS' follows the Euler's transform and is described in three detailed steps:

- Rotate axis w' with angle α by matrix $\mathbf{D}$, where α is the angle between axis u' and vector O_a. The u' axis becomes the $x1$ axis as shown in Fig. 3.
- Rotate axis $x1$ with angle γ by matrix $\mathbf{C}$. The $z1$ axis is rotated to $z2$. The $y2$ axis becomes the $y2$ axis as shown in Fig. 3.
- Rotate axis $y2$ to axis y with angle β about $z2$ axis by matrix $\mathbf{B}$. The direction of y is the same as vector V_{ab}, as shown in Fig. 3.

With the three steps, we obtain the three Euler rotation angles α, β, γ. The three rotation matrices $\mathbf{D}, \mathbf{C}, \mathbf{B}$ are defined as:

$$\mathbf{D} = \left\{ \begin{matrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{matrix} \right\} \tag{2}$$

$$\mathbf{C} = \left\{ \begin{matrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{matrix} \right\} \tag{3}$$

$$\mathbf{B} = \left\{ \begin{matrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{matrix} \right\} \tag{4}$$

Then the vector $\mathbf{G} = (u', v', w')$ in the GCS' will rotate to the vector $\mathbf{L} = (x, y, z)$ by:

$$\mathbf{L} = (\mathbf{B} \times \mathbf{C} \times \mathbf{D}) \times \mathbf{G} \tag{5}$$

where × denotes the matrix multiplication.

B. The Proposed 3D Dubins Path Design

This subsection describes a rotation based method to design a 3D Dubins path with a given target sequence and a pair of heading angles. The basic steps to design the 3D path is shown in Algorithm 1, where the inputs are the target coordinates in the GCS $T_1, T_2, \cdots, T_n$ with the given target sequence, and the outputs are the 3D Dubins curve coordinates in the GCS.

According to the proposed algorithm for designing the 3D Dubins path, we now show the continuity at the joint target point between two Dubins curves. The curve $C_{LCS'}$ is in the $x' - y'$ plane for the T_a-T_b segment, and the curve $C_{LCS''}$ is in the $x'' - y''$ plane for the T_b-T_c segment. The two curves touch at point T_b and have the same tangent. Hence, the two curves satisfy G^0 and G^1 continuity based on the definition of geometric continuity. In addition, the two curves are designed with the same turning radius. Therefore, the radius of curvature of the two curves at point T_b is the same, thus the 3D Dubins path satisfies the G^2 continuity.

C. Back Propagation Algorithm via Trellis

The algorithm 1 designs 3D Dubins path for a given target sequence and given heading angles directly without interpolation. However, every target can have multiple heading angles which lead to B^n combinations, where B is the number of angles in the discrete set, and n is the number of targets in the assigned sequence. How to select the shortest path from start to end and reduce the computational complexity becomes an important issue. In previous work, Wang applied Genetic Algorithm [10] and Cai used exhaustive search [7] to choose the shortest path. In this paper, we use a back propagation method to select the optimal heading angles for a given target sequence.

Assume the target sequence has a total of n targets. Because the AUV sets off and returns to the first target, there will be $n + 1$ stages for the target-sequence. We define the discrete azimuth heading sets to reduce the computational complexity. We assume that there are B different heading angles for the AUV to choose at each target, which corresponds to B states in each stage. Now, we need to find a shortest state path from Stage 1 to State $n + 1$, which is implemented by the back propagation method in Function 2.

Algorithm 1 : 3D Dubins Path Design

```
1) Assign the coordinates of targets $T_1$ and $T_2$ in the $GCS$ as $T_a(u,v,w) := T_1(u,v,w)$ and $T_b(u,v,w) := T_2(u,v,w)$.
2) Define a new coordinate system $GCS'$ by shifting the origin of $GCS$ to $T_a$. Find the vector $V_{ab}$ connecting $T_a$ and $T_b$. Define $LCS$ for $T_a$ as shown in Fig.3. The coordinate of $T_b$ in $LCS$ is then $(0, L, 0)$, where $L$ is the length of vector $V_{ab}$;
3) Since both $T_a$ and $T_b$ are on the $X$-$Y$ plane of $LCS$, we now select a pair of out-going and in-coming vectors in the available set for $T_a$ and $T_b$, respectively. Denote the vectors in $LCS$ as $O_a$ and $I_b$, respectively.
4) Apply the 2D Dubins curve method to find the shortest path from $T_a$ to $T_b$ in $LCS$ and denote the path as $C_{LCS}$.
5) Apply the Euler's transform algorithm to find the rotation matrices $\mathbf{D}, \mathbf{C}, \mathbf{B}$ for transferring vectors in $LCS$ to $GCS'$. Hence, the transformed coordinates and heading vectors are $C_{GCS'} = (\mathbf{B} \times \mathbf{C} \times \mathbf{D})^{-1} \times C_{LCS}$, $O_{GCS'} = (\mathbf{B} \times \mathbf{C} \times \mathbf{D})^{-1} \times O_a$, and $I_{GCS'} = (\mathbf{B} \times \mathbf{C} \times \mathbf{D})^{-1} \times I_b$, respectively;
6) Shift $C_{GCS'}$ in $GCS'$ to coordinates in $GCS$ and output it as $C_{GCS}$ which is the 3D path from $T_a$ to $T_b$; Output the corresponding $O_{GCS'}$ and $I_{GCS'}$ as $\Phi_1$ and $\Phi_2$, the headings for $T_1$ and $T_2$, respectively;
7) Repeat Step 4-6 to design Dubins curves for other heading angles of $T_a$ and $T_b$, if required;
8) To design Dubins curve for the next segment of the target sequence, substitute $T_a$, $T_b$, and $O_a$ by $T_2$, $T_3$, and $I_2$, respectively. Repeat Step 2-7 to design the paths from $T_2$ to $T_3$.
9) Repeat Step 8 until all remaining targets in the given target sequence are visited and the AUV returns to $T_1$.
```

The back propagation method is illustrated by a trellis diagram, as shown in Fig. 4, where the back propagation method keeps only one surviving path entering into each state at each stage. This method achieves the optimal solution and reduces the computational complexity from exhaustive search. The back propagation method also has lower computational complexity than the GA algorithm which may not achieve the optimal solution.

The steps to choose the shortest path for a target sequence is listed as follows.

1) From Stage 1 to Stage 2, there are B^2 different paths. Compute the lengths of these paths via Algorithm 1 and choose a shortest path for each state in Stage 2, illustrated as the red lines in Fig. 4. Discard other paths and only keep the B shortest surviving paths from Stage 1 to each state in Stage 2 and record the path lengths as the path metric $M_{1b}, b = 1, \cdots, B$.
2) From Stage 2 to Stage 3, repeat Function 1 to calculate the lengths of the possible B^2 paths, add the path metric

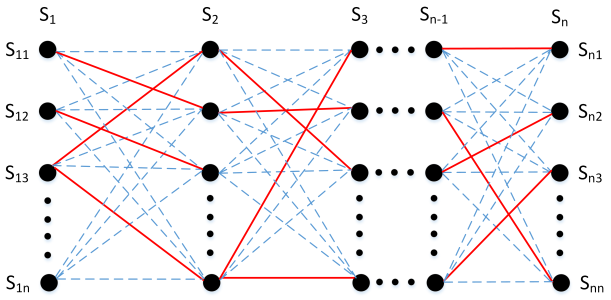


Fig. 4: Trellis diagram for the back-propagation algorithm for shortest path selection.

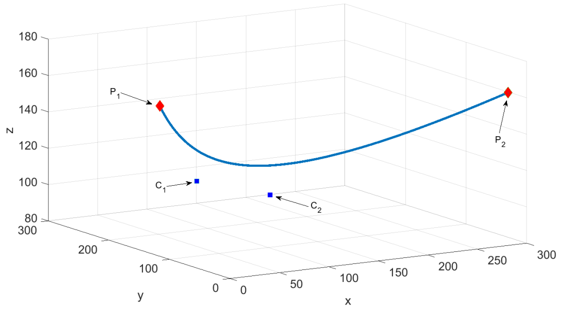


Fig. 5: Cubic Bezier curve with four control points.

M_1 to the corresponding paths; choose the shortest B paths that originate from Stage 1 and arrive at each state in Stage 3; Record the total path lengths as path metric $M_{2b}, b = 1, \cdots, B$.

3) Repeat Step 2 for $(n-2)$ times, until Stage $(n+1)$.
4) Choose the shortest path among the B surviving path, and denote it as P. The corresponding states along the trellis are the optimal heading angle sequence to achieve the shortest path for a given target-sequence.

For a given sequence, Function 2 is used to choose the shortest 3D Dubins path from all possible heading angles with affordable computational complexity.

IV. 3D Bezier Curve

In this section, we apply the Bezier curve to design the 3D path for AUV [15], [16] and compare 3D Bezier curves to 3D Dubins curves. Bezier curves have advantage in continuity, hence, are often used to design smooth paths. For AUV motion-control, the path of AUV has to satisfy G^2 continuity at each point of the path. With G^2 continuity, the AUV can change its acceleration gently. Therefore, we choose the cubic Bezier curve to design 3D path for AUVs, as shown in Fig. 5, where four control points are used to design one Bezier curve. Two of the four points are the starting and ending target locations, and the other two points may be arbitrarily chosen to control the curvature of the path.

The cubic Bezier curve is defined by the following equation:

$$P(t) = (1-t)^3 P_1 + 3(1-t)^2 t C_1 + 3(1-t)t^2 C_2 + t^3 P_2 \tag{6}$$

where P_1, P_2, C_1, and C_2 are four control points of the cubic Bezier curve and $t \in [0, 1]$ with $t = 0$ corresponding to the starting target and $t = 1$ corresponding to the ending target. In AUV path planning, we denote the control points P_1 and P_2 as the targets. The control points C_1 and C_2 are two points out of the path to control the curve's shape and length.

Each cubic Bezier curve is differentiable and continuous to degree 3 at any point on the curve. Therefore, the continuity of the overall path of the sequence is to ensure that the joint point of two Bezier curves are continuous. We now consider two cubic Bezier curves $P(t)$ with its four control points at (P_1, P_2, C_1, C_2) and $Q(t)$ with its four control points at (Q_1, Q_2, C_3, C_4). Based on the definition of continuity, G^0 continuity can be achieved by $P(1) = Q(0)$, which yields

$$P_2 = Q_1. \tag{7}$$

For G^1 continuity, we have $\dot{P}(1) = \dot{Q}(0)$, which results in

$$P_2 - C_2 = C_3 - Q_1. \tag{8}$$

Combining (7) and (8), we get:

$$C_3 = 2P_2 - C_2. \tag{9}$$

Similarly for G^2 continuity, we have $\ddot{P}(1) = \ddot{P}(0)$, yielding:

$$C_4 - 2C_1 + Q_1 = P_2 - 2C_2 + C_1. \tag{10}$$

Combining (10), (8), and (7), we get:

$$Q_2 = 4P_2 - 4C_2 + C_1. \tag{11}$$

In path planning of the overall sequence, we assume P_1, P_2, Q_1, and Q_2 are the targets. The other control points will be chosen to minimize the total length of the 3D Bezier curves. For a given target sequence, we choose C_1 and C_2 for first two targets T_1, T_2 and design the first segment of cubic Bezier curve between these two targets. The control points of the following segments of Bezier curves are calculated by (9) and (11) to satisfy the G^2 continuity at the joint target locations. Therefore, how to choose the two control points for the first segment T_1 to T_2 determines the overall Bezier curve path for the whole sequence. A cubic grid in the 3D space near T_1 and T_2 may be defined and all points on the grid may be selected as the two control points for the Bezier curve design. If the total number of grid points is N, then the two control points have N^2 possible combinations. Then how to choose the grid and how to choose the two control points on the grid becomes a difficult problem.

V. Simulation

A. Results of the Proposed 3D Dubins Curve Algorithm

In this paper, computer simulation was set up with 8 randomly distributed underwater targets in a cube of 600 × 600 × 600 m^3 space which would be visited by one survey-type AUVs. The turning radius of survey-type AUV is usually larger than 10 meters. For example, the MBARI Dorado class torpedo-style AUV has a minimum 10 meters turning radius. Therefore, we set the turning radius $r = 12$ m for Dubins path. The simulated space is much smaller than the real AUV navigation space. Because of the small turning radius, we intend to zoom in to our simulation results to show the Dubins curves clearly. Therefore, we choose a small underwater cube space. For the sake of simplicity, we chose the typical set of the azimuth headings for AUV movement as $\phi = \{\frac{b\pi}{4}\}$ with $b = 1, 3, 5, 7$.

Following the method proposed in sections III, the proposed algorithm designed the 3D Dubins path for eight targets, as shown in Fig. 6, where different segments are shown in different colors.

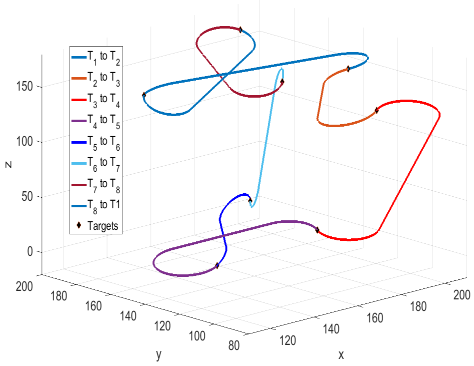


Fig. 6: 3D Dubins paths for one AUV visiting 8 targets, designed by the new algorithm

The total distances traveled by the AUV are compared among the different design methods, as shown in Fig. 7. The proposed 3D algorithm reduced the total length of 3D path over the spline interpolation, and has the similar distances as the linear interpolation method. It is interesting to note that the total Euclidean distance is the shortest path among all designed paths. The spline interpolation method has the longest total distance, because the cubic spline interpolation has to increase the distance to satisfy the continuity requirement. The proposed algorithm satisfied the continuity requirement without increasing the path length and achieved a similar total distance as the linear interpolation method; while the linear interpolation method failed to satisfy the G^1 continuity at the joint target locations.

To demonstrate the difference between the proposed algorithm and the interpolation methods, we applied these methods to design 3D path with the same target sequence, and the results are shown in Fig. 8(a) and Fig. 8(b), respectively. Compared with linear interpolation, the proposed algorithm and spline interpolation achieve G^2 continuity at joint target locations. In addition, the proposed algorithm has shorter total distance than spline interpolation.

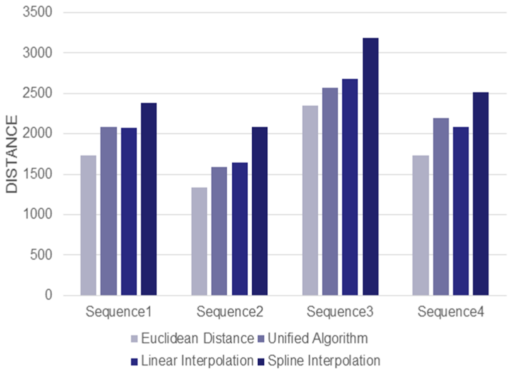
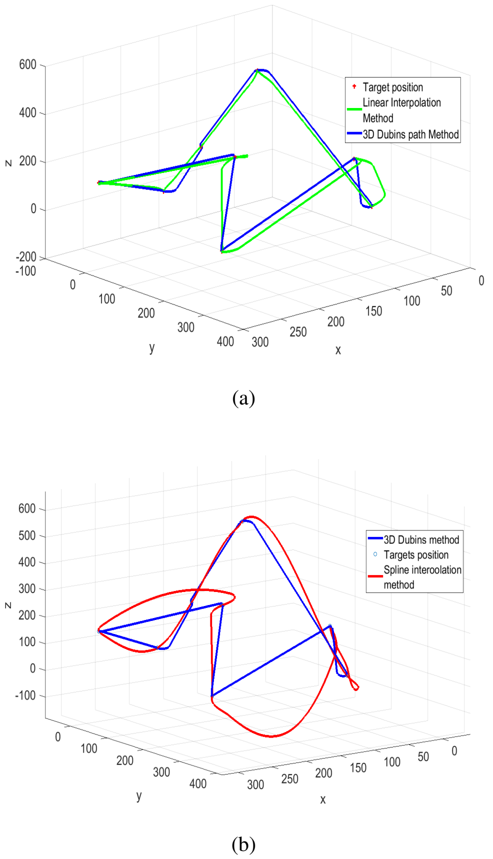


Fig. 7: Total distance comparison for four target-sequence.



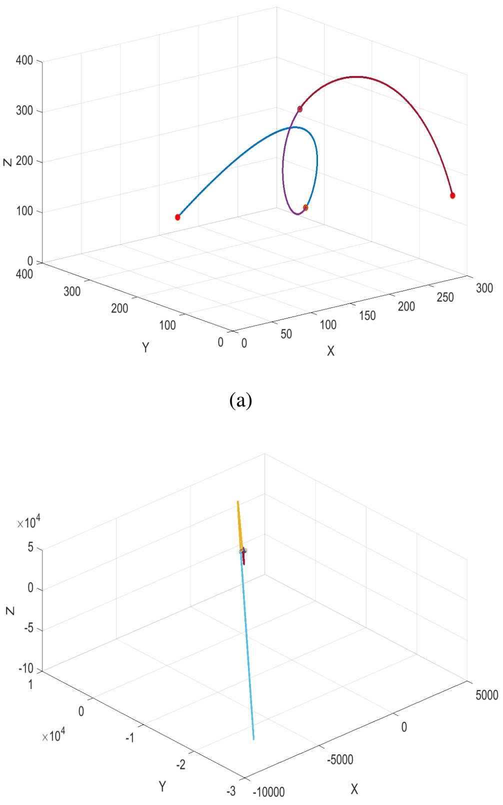
(a)

(b)

Fig. 8: Comparison of 3D paths designed by the proposed algorithm and (a) linear interpolation method; (b) the Spline interpolation method

B. Results of 3D Bezier curves

In section IV, we introduce the method of designing 3D Bezier curves. The lengths of Bezier curves are determined by the choice of the first two control points. In our simulation, we defined a cubic grid around the first two targets and chose the first two control points C_1 and C_2 on this grid randomly. We used the trial and error method to select the control points randomly for one million trials and designed the 3D Bezier curve for each trial. The shortest path among the one million trials was selected as the relatively shortest path for the given eight target sequence. The best result is shown in Fig. 9. In Fig. 9(a), the bezier curves from T_1 to T_4 perform well and each segment is smooth and short. However, when the number of targets increases, the control points for the last two targets in the sequence became far away from the targets and the designed cubic Bezier curves became very long, as shown in Fig. 9(b), where the scale of the figure is on 10^4. Although the 3d Bezier curves satisfied the G^2 continuity, the tight coupling of control points in the target sequence made the design of the 3D Bezier curve much more difficult than the 3D Dubins curves.



(a)

(b)

Fig. 9: 3D Bezier curves for an eight-target sequence.(a) Bezier curve from T_1 to T_4; (b) Bezier curve from T_4 to T_8.

VI. Conclusion

In this paper, we have proposed a new algorithm to design 3D Dubins path with the Euler's rotation transform method rather than interpolation. The proposed algorithm guarantees the G^2 continuity at the joint target locations of multiple target sequence. In addition, we utilize the back propagation method to choose the shortest path for a given target assignment sequence, which reduces the computational complexity and guarantees the optimality of the path selection. The proposed algorithm achieves better 3D continuity and/or shorter path length than the existing interpolation methods. Compared with cubic Bezier curves, our new 3D Dubins curve can achieve much shorter path length.

References

[1] D. Bingham, T. Drake, A. Hill, and R. Lott, "The application of autonomous underwater vehicle (AUV) technology in the oil industry–vision and experience," in *FIG XXII Int. Congr , Washington DC, USA*, Apr. 2002.

[2] L. L. Whitcomb, "Underwater robotics: Out of the research laboratory and into the field," in *2000 Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, vol. 1, Apr. 2000, pp. 709–716.

[3] G. A. Hollinger, S. Choudhary, P. Qarabaqi, C. Murphy, U. Mitra, G. s. Sukhatme, M. Stojanovic, H. Singh, and F. Hover, "Underwater data collection using robotic sensor networks," *IEEE J. Select. Areas Commun.*, vol. 30, no. 5, pp. 899–911, Jun. 2012.

[4] S. Yoon, A. K. Azad, H. Oh, and S. Kim, "Aurp: An auv-aided underwater routing protocol for underwater acoustic sensor networks," *Sensors*, vol. 12, no. 2, pp. 1827–1845, Feb. 2012.

[5] B. Gulbahar and O. B. Akan, "A communication theoretical modeling and analysis of underwater magneto-inductive wireless channels," *IEEE Trans. Wireless Commun.*, vol. 11, no. 9, pp. 3326–3334, Sep. 2012.

[6] R. Cui, Y. Li, and W. Yan, "Mutual information-based multi-auv path planning for scalar field sampling using multidimensional RRT," *IEEE Trans. Syst., Man, and Cybern. Syst.*, vol. 46, no. 7, pp. 993–1004, Dec. 2016.

[7] W. Cai, M. Zhang, and Y. R. Zheng, "Task assignment and path planning for multiple autonomous underwater vehicles using 3D dubins curves," *Sensors*, vol. 17, no. 7, p. 1607, Jul. 2017.

[8] J. P. H. B. Chow, C. M. Clark, "Assigning closely-spaced task points to multiple autonomous underwater vehicles," *J. Ocean. Technol.*, vol. 6, no. 1, pp. 179–202, 2011.

[9] Y. Lin and S. Saripalli, "Path planning using 3D dubins curve for unmanned aerial vehicles," in *Proc. Int. Conf. Unmanned Aircraft Syst.* IEEE, May. 2014, pp. 296–304.

[10] Y. Wang, W. Cai, and Y. R. Zheng, "Dubins curves for 3d multi-vehicle path planning using spline interpolation," in *Proc. MTS/IEEE OCEANS 2017 Conf. Anchorage, AK, USA*. MTS-IEEE, 2017, pp. 1–5.

[11] C. R. German, D. R. Yoerger, M. Jakuba, T. M. Shank, C. H. Langmuir, and K. Nakamura, "Hydrothermal exploration with the autonomous benthic explorer," *Deep Sea Research Part I: Oceanographic Research Papers*, vol. 55, no. 2, pp. 203–219, 2008.

[12] B. A. Barsky and T. D. DeRose, "Geometric continuity of parametric curves: three equivalent characterizations," *IEEE Comput. Graph. Appl.*, no. 6, pp. 60–68, 1989.

[13] A. Alvarez, A. Caiti, and R. Onken, "Evolutionary path planning for autonomous underwater vehicles in a variable ocean," *IEEE J. Ocean. Eng.*, vol. 29, no. 2, pp. 418–429, Apr. 2004.

[14] T. J. Prestero, "Verification of a six-degree of freedom simulation model for the remus autonomous underwater vehicle," Master's thesis, MIT, 2001.

[15] M. K. Agoston and M. K. Agoston, *Computer graphics and geometric modeling*. Springer, 2005, vol. 1.

[16] K. Jolly, R. S. Kumar, and R. Vijayakumar, "A bezier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits," *Robotics and Autonomous Systems*, vol. 57, no. 1, pp. 23–33, 2009.