Fall 2018

3D Dubins curves for multi-vehicle path planning

Yiheng Wang

Department:

3D DUBINS CURVES FOR MULTI-VEHICLE PATH PLANNING

by

YIHENG WANG

A THESIS

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

in

ELECTRICAL ENGINEERING

2018

Approved by

Y. Rosa Zheng, Advisor
David Pommerenke
Levent Acar

ABSTRACT

This thesis proposes a unified algorithm for target assignment and path planning in 3D space for multiple Autonomous Underwater Vehicles (AUVs) to visit multiple targets. The multi-target assignment and path planning problem is modeled as a multiple Traveling Salesmen Problem (mTSP) and is usually solved by two separate algorithms: the multiple task assignment problem is first solved by the Genetic Algorithm (GA) using Euclidean distances between the targets; then the 3D path planning problem is solved for each assignment by selecting Dubins curves or other continuity curves. In contrast, this paper embeds the 3D Dubins curve selection into the target assignment step and uses the true path lengths rather than Euclidean distances as the fitness value of the GA. The unified algorithm is implemented by three functions: Function 1 designs a 3D Dubins path for a given target assignment sequence and given incoming-outgoing angles by an innovative rotation method extended from the well-known 2D Dubins curves; Function 2 uses the back-propagation algorithm to choose the shortest path among all possible incoming-outgoing angle combinations for a given target assignment sequence; Function 3 uses the true lengths of the 3D Dubins curves in the Genetic Algorithm (GA) to assign target sequence to multiple AUVs. Computer simulations demonstrate that the proposed algorithm provides better G2 continuity in 3D space than the existing linear or spline interpolation methods. The unified algorithm solves the NP-hard integer programming problem with an affordable computational complexity.

ACKNOWLEDGMENTS

I would first like to thank my academic advisor Prof. Y. Rosa Zheng, for her kind support on my master study and research. Her patient advice and suggestions helped me over the two years on my research and my life. She can always steer me to find the right direction to solve different problems.

I wish to thank Dr. Wenyu Cai and Ms. Meiyang Zhang for collaboration on a previous work of this path planning working. The work is supported in part by National Science Foundation under grant #CPS-1646548 and in part by the Wilkens Missouri Telecommunication Chair endowment.

I would also like to express my sincere gratitude to the other committee member, Dr. David Pommerenke and Dr. Levent Acar. Thanks for their time on my thesis.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. 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, 2, 3, 4]. In underwater wireless sensor network (UWSN), multiple AUVs are employed to collect data from predetermined targets via acoustic communication [5]. However, due to the limited range and bandwidth of acoustic communications and high energy cost of sensor node, the AUV-Aided Underwater Routing Protocol in [5] still has a lot of limitations in data collecting. Recently, the Magneto-Inductive (MI) communication has the advantages of low-cost and easily-deployable [6]. Therefore, multiple AUVs can be utilized to complete the data collection by visiting multiple sensor nodes via MI communication. Hence, multi-target assignment and path planning problems in two and three dimensional space have attracted many research attention [7, 8, 9, 10].

The multi-target assignment problem can be modeled as the multiple Traveling Salesmen Problem (mTSP). In previous work, Chow applied the K-means clustering method and Garau used the heuristic search algorithm to solve mTSP. In addition, the Genetic Algorithm (GA) is an efficient method to assign multi-target to multi-AUV [11, 12]. Due to the high computational complexity, the existing works for AUV multi-target assignment utilize the Euclidean distances between targets as the fitness function in the mTSP model. However, the actual paths of AUVs are often cured instead of a straight line between two targets, resulting in much larger distances which are ignored by the GA algorithm.

With the assigned tour sequences, path planning algorithm design a smooth path for each AUV to visit all the assigned targets. Some existing works focus on 2D space and design smooth path in 2D only [13]. 2D point-to-point smooth paths in X-Y are often designed via Dubins curves [13, 14], 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 [10, 12]. However, the linear interpolation method [10] fails to meet with G^1 continuity at multiple targets. The spline interpolation method [12] may result in much longer total distances between targets in 3D space.

In this thesis we proposes a unified algorithm to solve 3D multi-target assignment and path planning together. This algorithm consists of three 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. Function 3 uses the true 3D Dubins curves length as fitness value in mTSP to assign target sequence to multiple AUVs. With the assigned target sequences, we apply Function 1 and Function 2 to design 3D Dubins path for each AUV. In this thesis, we utilize the length of 3D Dubins paths rather than Euclidean distance in GA which leads to more accurate solutions of mTSP. 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 [15]. 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 [16]. Besides motion constraints of AUV, the ocean environmental conditions will effect the movement of AUVs, such as the strong ocean current [7]. 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.

2. SYSTEM MODEL, MOTION CONSTRAINTS AND 2D DUBINS CURVE

Constraints and 2D Dubins Curve Conventionally, the multi-target assignment is modeled as mTSP and is solved by the Genetic Algorithm by incorporating the 3D Euclidean distance between targets as the fitness function. Once the target sequences are assigned to each AUV, the path planning is designed by the 2D Dubins curves calculated for each AUV by projecting the coordinates of the assigned targets into 2D plane, then interpolating the 2D Dubins curves into 3D space. This section establishes the mTSP model and describes the existing 2D Durbins curve design method.

2.1. SYSTEM MODEL

We consider a group of survey-type AUVs $\mathcal{A} = \{A_1, A_2, \cdots, A_K\}$ as a collaborative team to complete the mission of visiting multiple underwater targets in a 3D underwater environment, as shown in Fig. 2.1. In this 3D space, we assume a set of targets $\mathcal{T} = \{T_1, T_2, \cdots, T_N\}$, where the parameter K and N represent the number of AUVs and the number of targets, respectively. These targets are randomly distributed in the Global Coordinate System (GCS). We assume that all of the AUVs set off from the origin and return back to the origin after visiting all assignments. Each target will be visited by one AUV exactly once.

The multi-target assignment problem is modeled as the multiple traveling salesmen problem and its integer programming formulation is given as [11]:

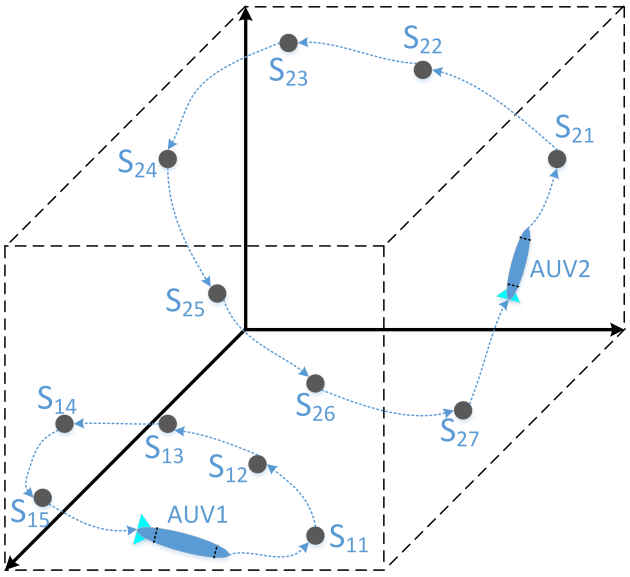


Figure 2.1. Multiple AUVs $A_1, \cdots, A_K$ visit multiple targets $T_1, \cdots, T_N$ and a target sequence $\mathcal{S}_k = \{S_{k1}, \cdots, S_{ki}, \cdots, S_{k,I}\}$ is allocated to each AUV via assignment algorithm.

$$\text{minimize} \quad L(c) = \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{K} c_{ijk} \cdot d_{ijk} \tag{2.1}$$

$$\text{subject to} \quad \sum_{j=1}^{N}\sum_{k=1}^{K} c_{1jk} = K \tag{2.2}$$

$$\sum_{i=1}^{N}\sum_{k=1}^{K} c_{i1k} = K \tag{2.3}$$

$$\sum_{i=1}^{N}\sum_{k=1}^{K} c_{ijk} = 1, \quad j = 2, 3, \cdots, N \tag{2.4}$$

$$\sum_{j=1}^{N}\sum_{k=1}^{K} c_{ijk} = 1, \quad i = 2, 3, \cdots, N \tag{2.5}$$

where (2.1) is the objective function representing the total distance of the AUVs visiting all targets. The constraints (2.2) and (2.3) ensure that all AUVs start from and return to the origin T_o. The constraints (2.4) and (2.5) guarantee that all of the targets are visited by one AUV exactly once. The list of notations is shown in Table. 2.1.

Table 2.1. List of Notations

Notation	Definition
$\mathcal{A}$	the set of survey-type AUVs
$\mathcal{T}$	the set of randomly targets
A_k	the k-th survey-type AUV
T_n	the n-th target
$t_n = (u_n, v_n, w_n)$	coordinates of the n-th target
K	the total number of survey-type AUVs
N	the total number of targets
T_o	the origin where all AUVs start and return
L	total distance of all AUVs visiting all targets
$\mathcal{L}_k$	length set of Dubins paths for the k-th AUV
L_{ki}	length of Dubins segment i of AUV k's sub-tour
$\mathcal{S}_k$	the target sequence for the k-th AUV
S_{ki}	the i-th target in the k-th target sequence
n_k	the number of targets in k-th target sequence
ϕ	azimuth heading angle
c_{ijk}	= 1 if AUV k is assigned to travel from target i to targets j
	= 0 otherwise
d_{ijk}	cost of AUV k traveling from target i to target j

2.2. AUV MOTION CONSTRAINTS

The famous REMUS AUV model created by Prestero [17] describes the six degrees of freedom (DOF) as *surge, sway, heave, pitch, roll, yaw.* In this paper, we simply the dynamic model by only considering the position and heading of the AUV, as shown in Fig. 2.2, where the AUV in the Local Coordinate System (X, Y, Z) has a heading direction $\Phi(\theta, \varphi)$, with φ being the angle between heading direction and the X-Y plane, and θ being the angle between the projected heading direction on the X-Y plane and the X-axis.

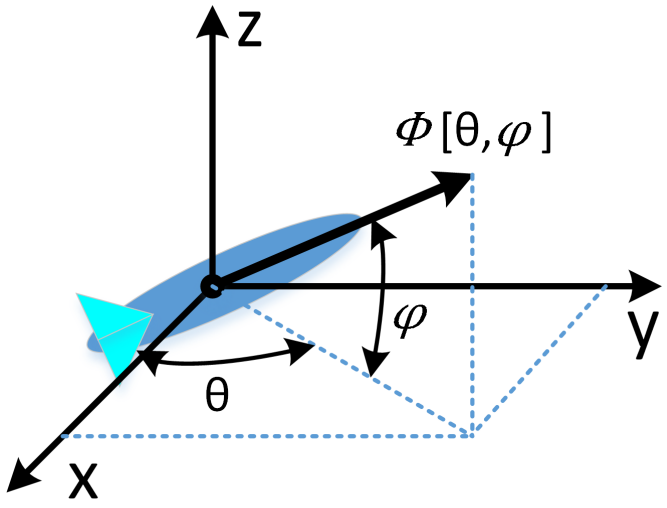


Figure 2.2. Local Coordinate System (LCS) and AUV heading direction

The survey-type AUV has strong nonholonomic constraints which require the path of AUV to have bounded curvature. The AUV is not allowed to take sharp turns. Therefore, the derivative of AUV heading direction has to satisfy:

$$\dot{\Phi} = \psi_k, \quad \psi_k \in [-\psi_a, \psi_a] \tag{2.6}$$

where the dot operator is the first derivative with respect to θ and φ, ψ_k is k-th AUV heading direction's first derivative and ψ_a is the curvature bound.

In addition, the nonholonomic constraints require that the AUV path satisfies geometric continuity. For example, the G^0, G^1 and G^2 continuities are defined as follows [16]: 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 $P(b) = Q(c)$, 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.

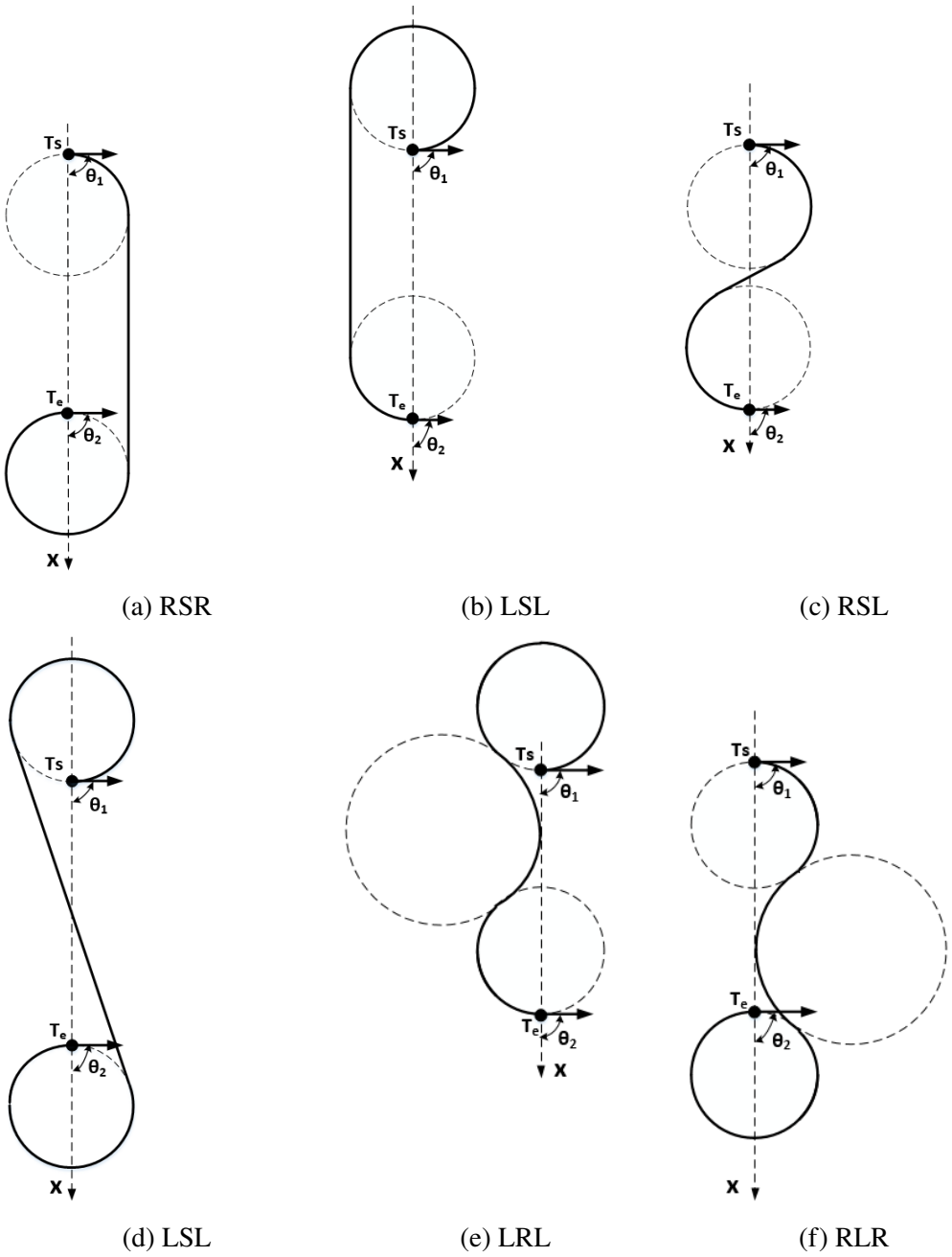
G^2 *Continuity*: If G^1 continuous and $\ddot{P}(u)|_{u=b} = \ddot{Q}(v)|_{v=c}$, then the two curves meet at the joint point with G^2 continuity.

2.3. 2D DUBINS CURVE

The Dubins curves satisfy the motion constraints by a combination of maximum curvature arcs (C) and/or a straight line segment (S). Consider 2D Durbins curves on the X-Y plane. For a given incoming and outgoing angle pair, the Dubins curves includes four CSC curves and two CCC curves: RSR, LSL, RSL, LSR and LRL, RLR, as the example shown in Fig. 2.3, where Dubins curves starts from T_s and arrive at T_e with both incoming and outgoing angles equal to $\pi/2$.

When the targets are separated far apart in comparison to their turning radius, the shortest Dubins path is among the four CSC type of curves, since it has been shown [18] that the CCC curves are always longer than the CSC curves when the distance d between T_s and T_e satisfies:

$$d > \sqrt{4r^2 - (|r\cos\theta_1| + |r\cos\theta_2|)^2} + |r\sin\theta_1| + |r\sin\theta_2| \tag{2.7}$$



(a) RSR (b) LSL (c) RSL

(d) LSL (e) LRL (f) RLR

Figure 2.3. Dubins curves : CSC family and CCC family

where r is the turning radius of the AUV, θ_1 and θ_2 are the azimuth headings of the AUV at T_s and T_e. In this paper, we assume that the distances between targets always satisfy (2.7). Hence, in our unified algorithm, we only consider the CSC Dubins family when designing 3D Dubins paths.

Points on the CSC Dubins curves can be calculated by three operations L_ι (for left turn), R_ι (for right turn) and S_ι (for straight line). The transform from one starting point $[(x, y), \theta]$ to the desired point are:

$$L_\iota(x, y, \theta) = (x + r\sin(\theta + \iota/r) - r\sin(\theta), y - r\cos(\theta + \iota/r) + r\cos(\theta), \theta + \iota) \tag{2.8}$$

$$R_\iota(x, y, \theta) = (x - r\sin(\theta - \iota/r) + r\sin(\theta), y + r\cos(\theta - \iota/r) - r\cos(\theta), \theta - \iota) \tag{2.9}$$

$$S_\iota = (x + \iota\cos(\theta), y + \iota\sin(\theta), \theta) \tag{2.10}$$

where ι is the path length from the starting point $[(x, y), \theta]$ to the desired point.

To reduce the computational complexity, we confine the azimuth headings of an AUV to a finite number of directions, as shown in Fig. 2.4, where only a set of headings θ are available. For example, the left figure has four azimuth headings for each of the target in the assignment sequence. For a target in the middle of a multi-target assignment sequence, the outgoing angle has to be the same as the incoming angle. For a given target sequence, there will be B^I possible incoming and out-going combinations, where B is the total number of angles allowed in the direction set, and I is the total number of targets in the assignment. Function 2 is to find the angles for all targets that yield the shortest path length.

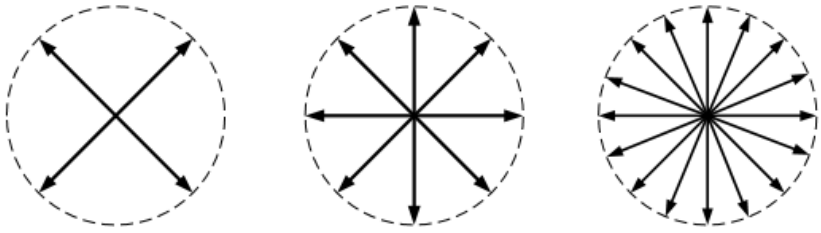


Figure 2.4. Sets of discretized headings. Left: $\theta = \{\frac{b\pi}{4}\}$ with $b = 1, 3, 5, 7$. Middle: $\theta = \{\frac{b\pi}{4}\}$ with $b = 0, 1, 2, \cdots, 7$. Right: $\theta = \{\frac{b\pi}{8}\}$ with $b = 0, 1, 2, \cdots, 15$.

3. THE ROTATION BASED 3D DUBINS PATH DESIGN ALGORITHM

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 the Global Coordinate System (GCS) u-v-w, as shown in Fig. 3.1. 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 Durbins curve is transformed to the 3D path in the GCS using Euler's transformation. Once the 3D Durbins segment between T_a and T_b is designed, the $LCS2$ centered at T_b is used in a similar manor to design the next segment of the 3D Durbins path.

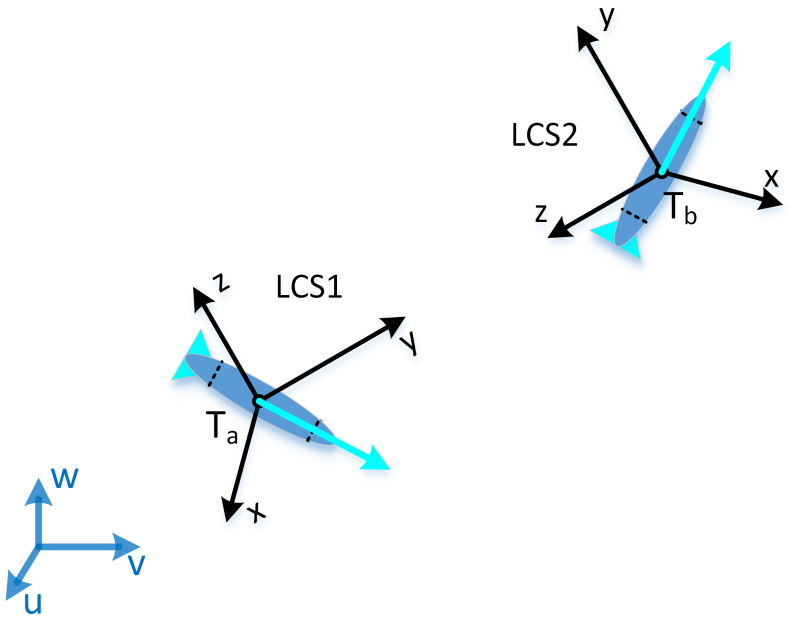


Figure 3.1. Global coordinate system (GCS) and two local coordinate systems: $LCS1$ and $LCS2$

3.1. 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 3.2. 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.

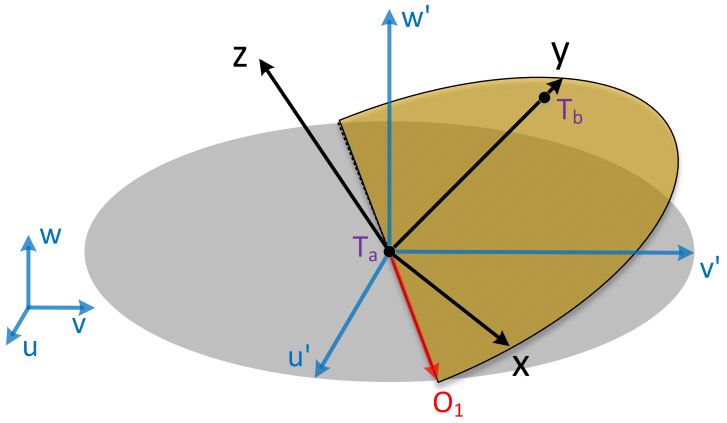


Figure 3.2. Shift GCS (u-v-w) to GCS' (u'-v'-w') and rotate w' axis with angle α.

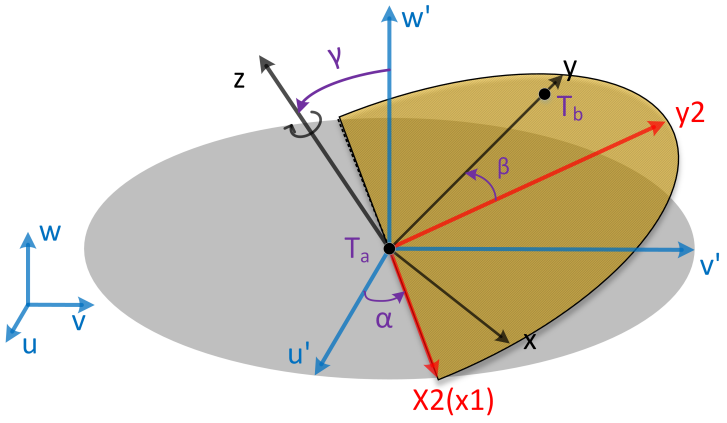


Figure 3.3. Rotate the u' axis with angle γ and rotate the $y2$ axis with angle β.

The rotation between $LCS1$ and GCS' follows the Euler's transform. The following are the steps of coordinate system rotation.

Step 1 Rotate axis w' with angle α by matrix **D**, where α is the angle between axis u' and vector O_a. The u' axis becomes the $x1$ axis as shown in Fig. 3.2.

Step 2 Rotate axis $x1$ with angle γ by matrix **C**. The $z1$ axis is rotated to $z2$. The $y2$ axis becomes the $y2$ axis as shown in Fig. 3.3.

Step 3 Rotate axis $y2$ to axis y with angle β about $z2$ axis by matrix **B**. The direction of y is same as vector V_{ab} as shown in Fig. 3.3.

With the three steps, we obtain the three Euler rotation angles α, β, γ. The three rotation matrices **D**, **C**, **B** are defined as:

$$\mathbf{D} = \begin{Bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{Bmatrix} \tag{3.1}$$

$$\mathbf{C} = \begin{Bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{Bmatrix} \tag{3.2}$$

$$\mathbf{B} = \begin{Bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{Bmatrix} \tag{3.3}$$

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

$$\mathbf{L} = (\mathbf{B} \times \mathbf{C} \times \mathbf{D}) \times \mathbf{G} \tag{3.4}$$

where $\times$ denotes the matrix multiplication.

3.2. FUNCTION 1: 3D DUBINS PATH DESIGN

Function 1 is 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 9, where the inputs are the target coordinates in the GCS $T_1, T_2, \cdots, T_n$ in the given target sequence, and the outputs are the 3D Durbins curve coordinates in the GCS.

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.3.3. 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 Φ_1 and Φ_2, the headings for T_1 and T_2, respectively;

7: Repeat Step 4-6 to design Durbins curves for other heading angles of T_a and T_b, if required;

8: To design Durbins 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.

According to the proposed design algorithm of the 3D Dubins path, we now show the continuity at the joint target point between two Durbins curves. As shown in Fig. 3.4, 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.

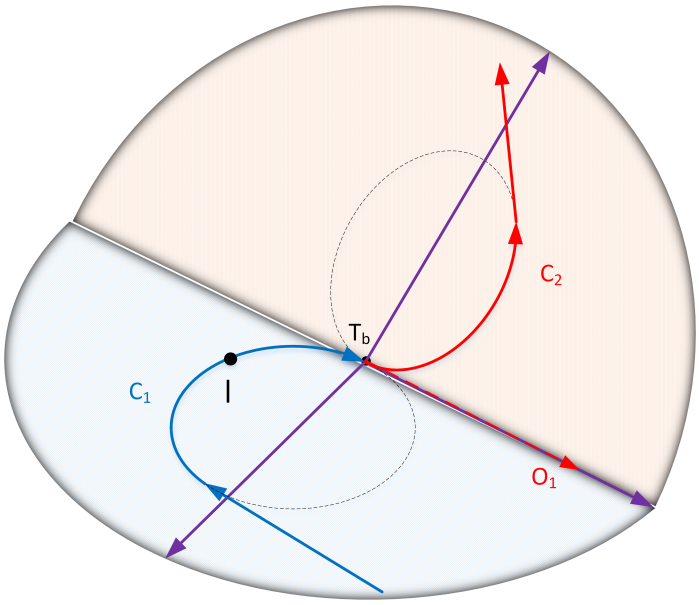


Figure 3.4. The continuity at the joint point of two Durbins segments

3.3. SPLINE INTERPOLATION METHOD

In contrast to the rotation algorithm, previous works designed 2D Dubins paths by projecting the 3D coordinates on to a 2D plane and then used spline interpolation method to map 2D Dubins path to 3D. In [10], Cai applied linear interpolation method mapping 2D path to 3D. The segment length of 2D Dubins path and Z coordinate of each target are combined together to achieve the path mapping. For example in Fig. 3.5, there are

three targets in an AUV sub-tour. Therefore, it has two Dubins paths in this sub-tour which are denoted as $L_k = \{L_{k1}, L_{k2}\}$. As shown in Fig. 3.5, we use length of Dubins path as horizontal coordinate and the Z coordinates of targets as vertical coordinates. From T_i to T_{i+1}, we use a cubic polynomial (3.5) to interpolate the curve.

$$z_i(l) = a_{i3}(l - l_i)^3 + a_{i2}(l - l_i)^2 + a_{i1}(l - l_i) + a_{i0} \tag{3.5}$$

for $l \in [l_i, l_{i+1}]$, where $[a_{i3}, a_{i2}, a_{i1}, a_{i0}]$ are coefficients of the polynomial and l_i are the horizontal coordinate of T_i in Fig. 3.5.

The spline interpolation constructs a polynomial such that

$$z_i(l_{i+1}) = z_{i+1}(l_{i+1}) \tag{3.6}$$

$$\dot{z}_i(l_{i+1}) = \dot{z}_{i+1}(l_{i+1}) \tag{3.7}$$

where the operator $\dot{z}$ is the derivative of z. The restrictions of (3.6) and (3.7) guarantee the G^1 continuity of cubic spline interpolation. Therefore the interpolated 3D Dubins path is continuous at each target and meets the requirement of survey-type AUV motion constraints. On contrast, the existing linear interpolation fails to maintain the continuity in the Z axis, as also shown in Fig. 3.6.

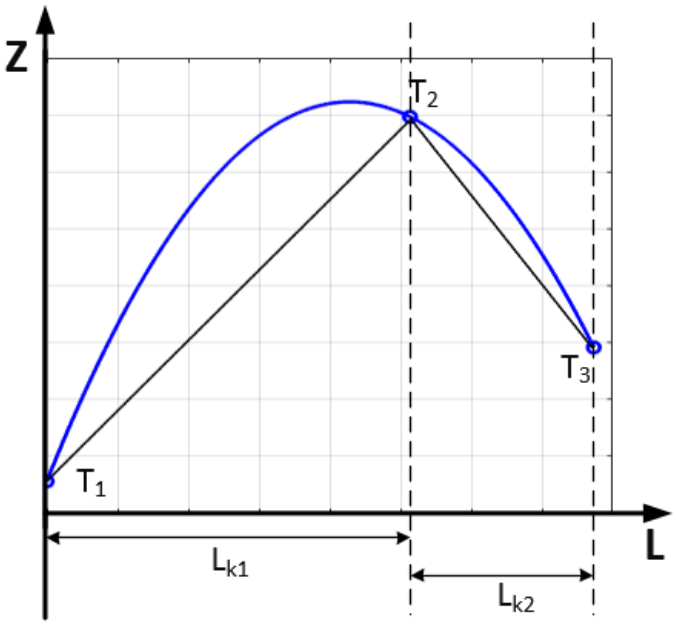


Figure 3.5. Interpolation procedure

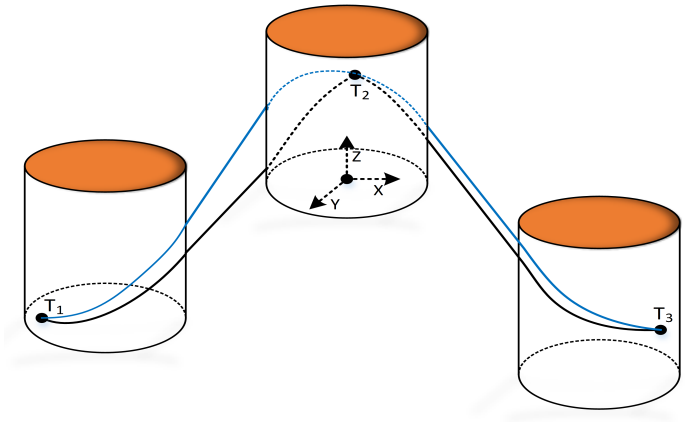


Figure 3.6. 3D continuity

4. BACK PROPAGATION ALGORITHM AND GENETIC ALGORITHM FOR MTSP

4.1. FUNCTION 2: BACK PROPAGATION ALGORITHM VIA TRELLIS

Function 1 designs 3D Durbins 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 [12] and Cai used exhaustive search [10] to choose the shortest path. In this thesis, we use a back propagation method to select the optimal heading angles for a given target sequence.

Assume the k-th target sequence has a total of n_k targets. Because the AUV sets off and returns to the first target, there will be $n_k + 1$ stages for the k-th target sequence. In Fig. 2.4, we have defined 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 $n_k + 1$, which is implemented by the back propagation method in Function 2.

The back propagation method is illustrated by a trellis diagram, as shown in Fig. 4.1, 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.

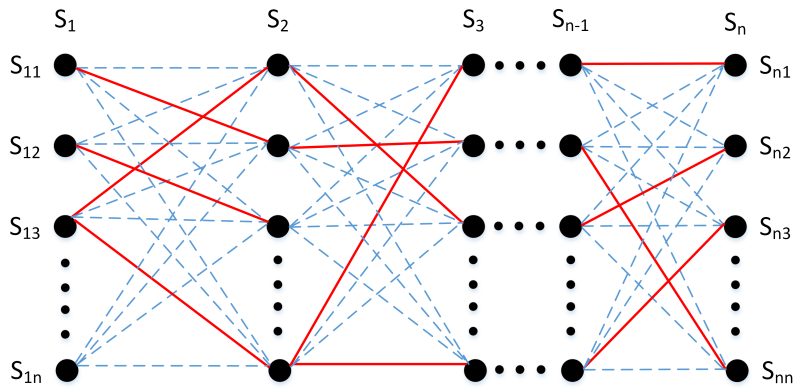


Figure 4.1. Trellis diagram for the back-propagation algorithm in Function 2.

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 Function 1 and choose a shortest path for each state in Stage 2, illustrated as the red lines in Fig. 4.1. 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 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 $(n_k - 2)$ times, until Stage $(n_k + 1)$.

4. Choose the shortest path among the B surviving path, and denote it as P_k. 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.

4.2. FUNCTION 3: GENETIC ALGORITHM FOR MTSP

Function 3 applies the Genetic Algorithm [19] to solve the NP hard problem of multi-target assignment (2.1). In previous works, researchers utilized the total Euclidean distance as the fitness value to choose the optimal solution in the mTSP model. In this thesis, we use the true lengths of 3D Dubins paths as the fitness value instead of Euclidean distance. Function 3 combines the GA with Function 2 and Function 1 to solve multi-target assignment problem in a unified approach. In the multi-target assignment problem, a feasible solution or a chromosome is a set of selections $C = \{c_{ijk}\}$ that satisfies the constraints and maps the N targets into K ordered target sequences $\mathcal{S}_k$. The crossover operator is the exchange of the targets among the different AUVs and/or at different order. The end condition is that the iteration number has exceeded the pre-set number or the newest parent has no significant improvement of fitness value.

The proposed algorithm is shown in Algorithm 2.

Algorithm 2 : Unified Target assignment and path planning algorithm

1: Generate random population Q (feasible solutions for the problem).
2: Apply Function 1 and Function 2 to evaluate the fitness valute of each chromosome in population Q (the 3D Dubins path length of each solutions).
3: Create a new generation
4: a) Select the best parent chromosome which has the smallest fitness value in Q.
5: b) Use crossover and mutation operators to generate the offspring from the chosen parent.
6: c) Put the offspring and the chosen parent in the new generation to replace the population Q.
7: Use the new population Q for the next iteration.
8: If the newest parent satisfy the end condition, stop and return the newest parent as the optimal solution. If not, go to step 2 until the end condition is satisfied.

5. SIMULATION RESULTS

5.1. UNIFIED ALGORITHM RESULT

In this paper, computer simulation was set up with $N = 32$ randomly distributed underwater targets in a cube of $600 \times 600 \times 600$ m^3 space which would be visited by $K = 4$ 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 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 and IV, Function 2 and Function 3 generate four sub-tour sequences for AUVs and choose the optimal heading-angle sequence. Function 1 design the 3D Dubins path for each AUV. The results of the four sub-sequence 3D Dubins path are shown in Fig. 5.1, where the 32 targets are clearly divided into four separated sections. Each AUV has been assigned to its own working space which will avoid collision of multiple AUVs.

The 3D Dubins paths of the AUVs are shown in Fig. 5.2-5.5 individually. In these figures, the 3D point-to-point Dubins paths are shown in different colors. As we can see in each segment of 3D Dubins path, there are not sharp turning in the 3D paths, especially at joint points of targets.

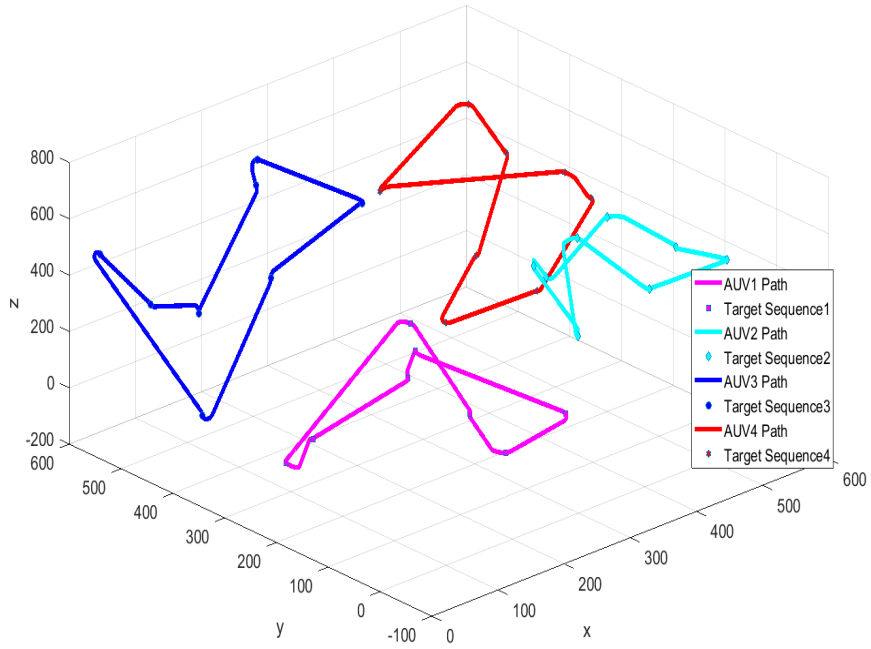


Figure 5.1. 3D paths for four AUVs visiting 32 targets, designed by the new unified algorithm

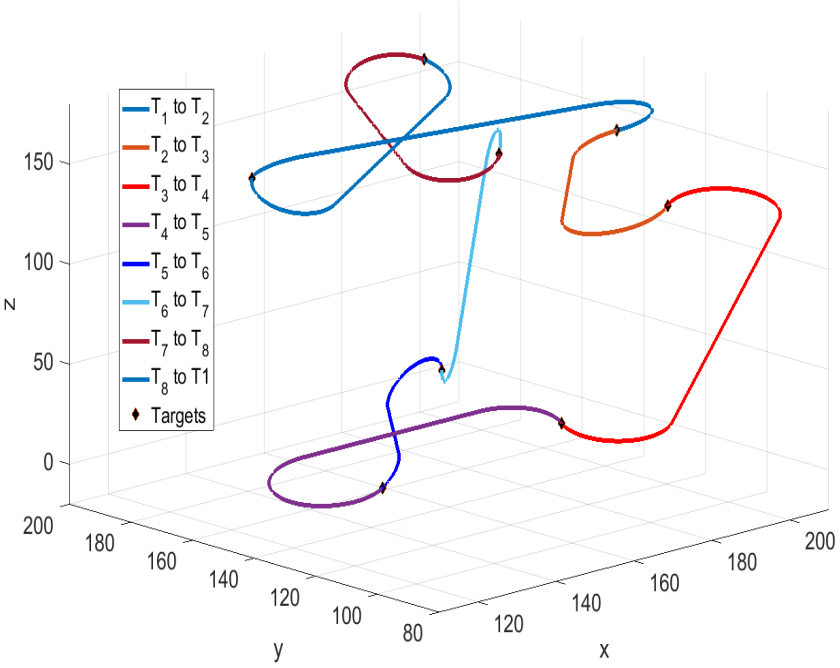


Figure 5.2. 3D Dubins path of AUV1

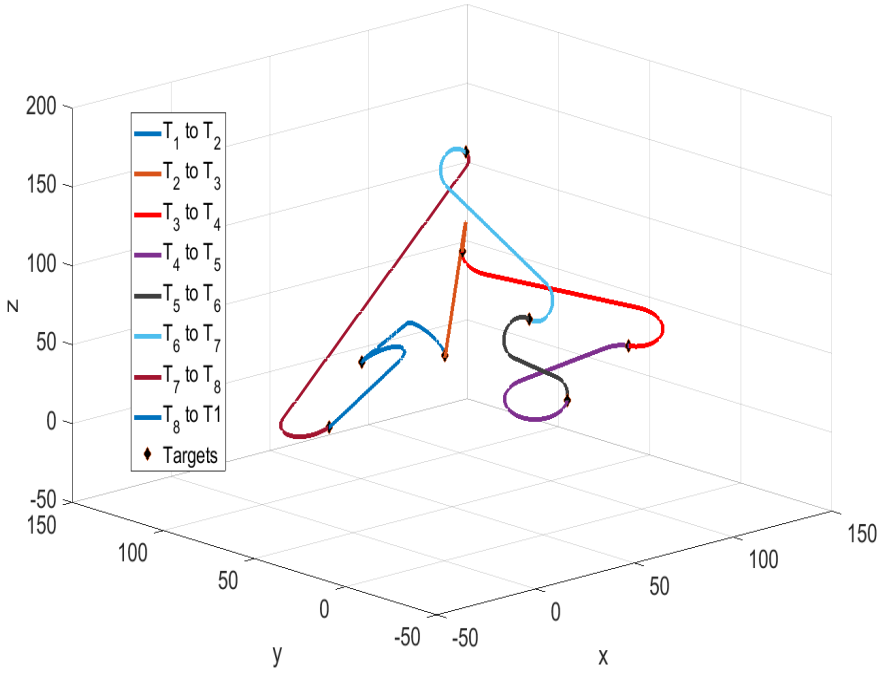


Figure 5.3. 3D Dubins path of AUV2

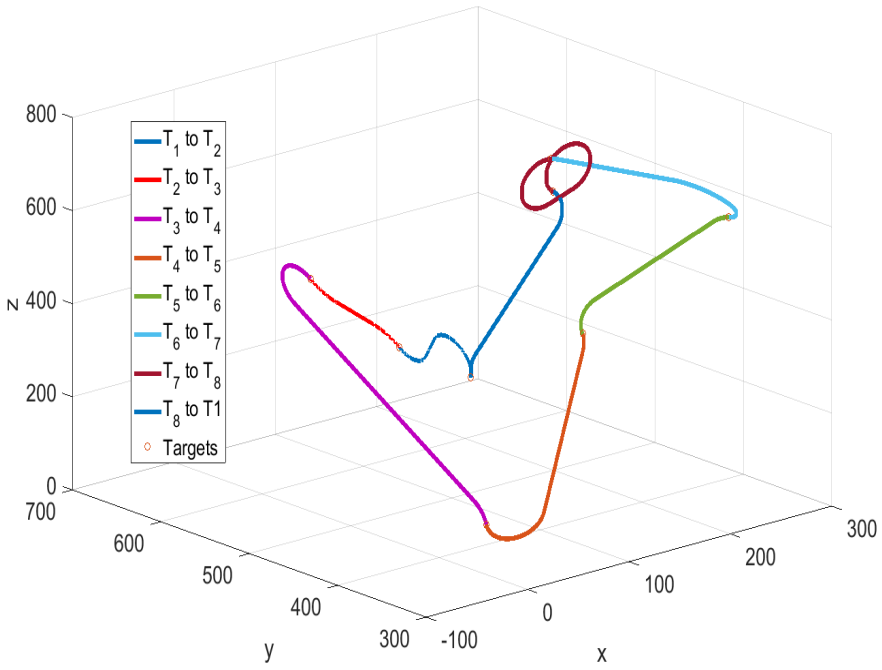


Figure 5.4. 3D Dubins path of AUV3

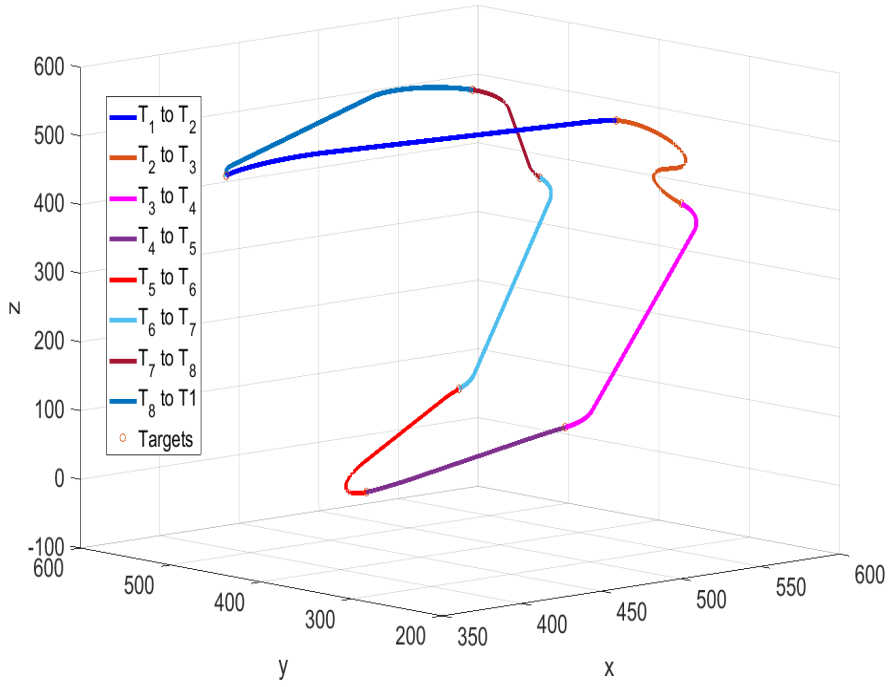


Figure 5.5. 3D Dubins path of AUV4

In Fig. 5.6-5.11, we choose the 3D Dubins of AUV1 to show the continuity of the unified algorithm. The 3D Dubins paths of T_1 to T_3, T_2 to T_4, T_3 to T_5, T_4 to T_6, T_5 to T_7 and T_6 to T_8 are shown in these figures. These figures focus on the continuity at the joint target. As we can see, the unified algorithm generate smooth curves to satisfy the motion constraints.

5.2. COMPARISON OF TOTAL DISTANCES

The total distances traveled by the AUVs are compared among the different design methods, as shown in Fig. 5.12. The unified 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

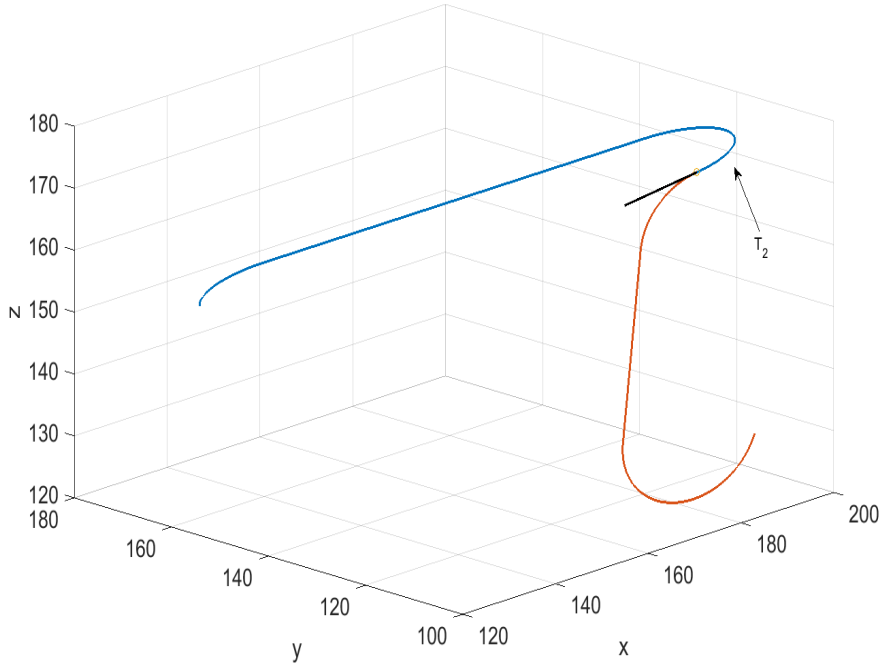


Figure 5.6. 3D Dubins path of AUV1 from T_1 to T_3

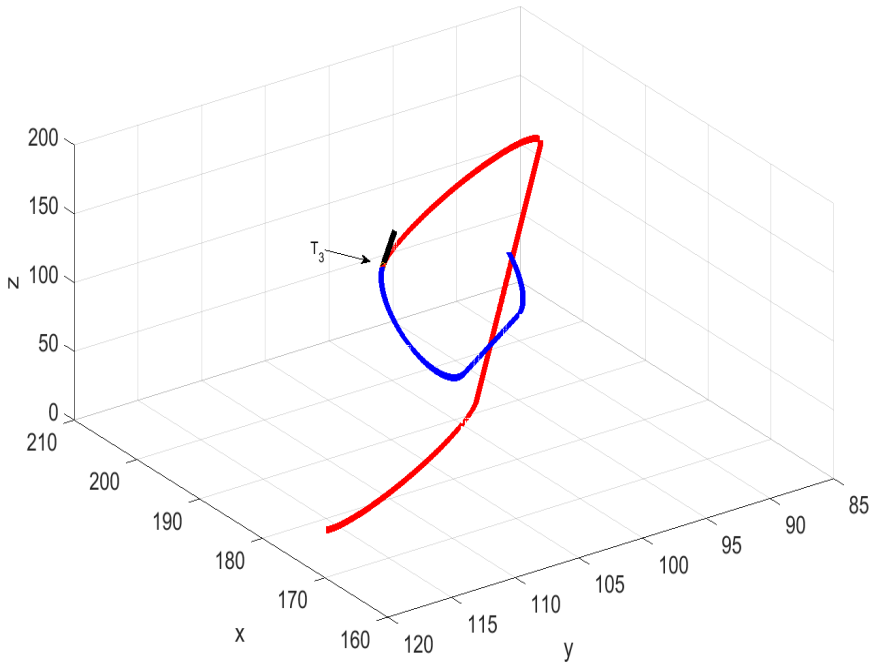


Figure 5.7. 3D Dubins path of AUV1 from T_2 to T_4

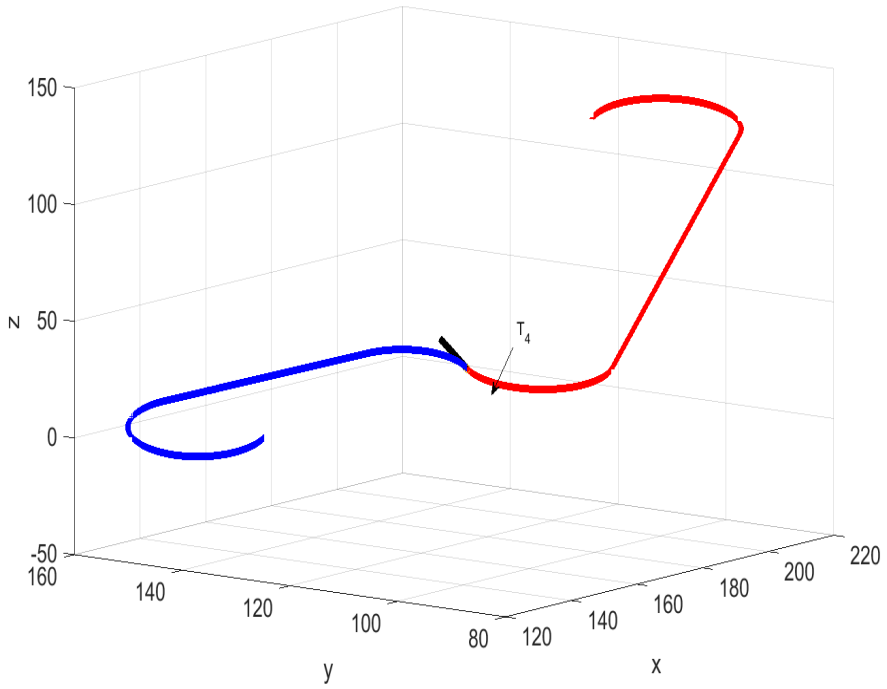


Figure 5.8. 3D Dubins path of AUV1 from T_3 to T_5

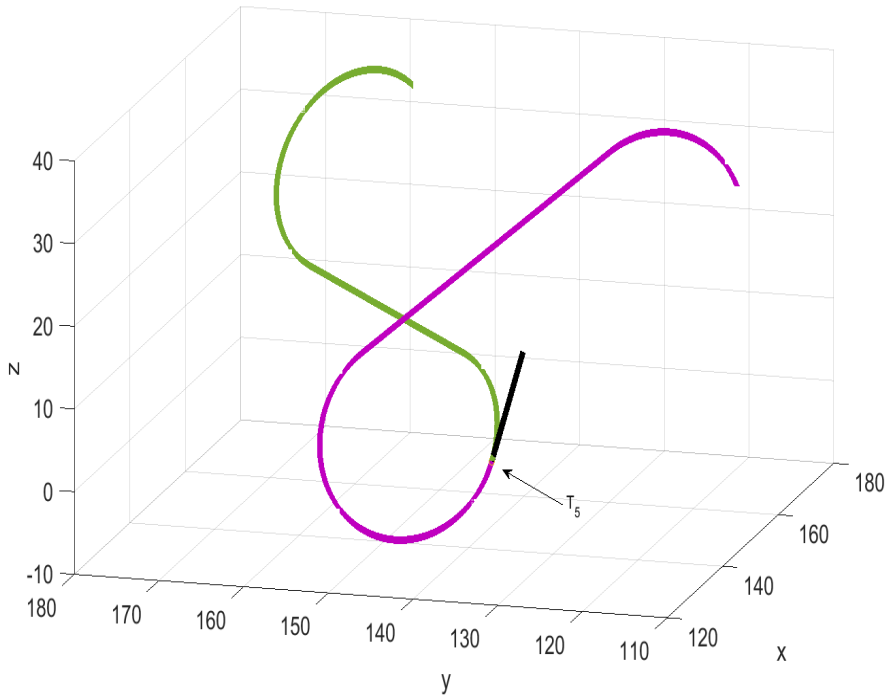


Figure 5.9. 3D Dubins path of AUV1 from T_4 to T_6

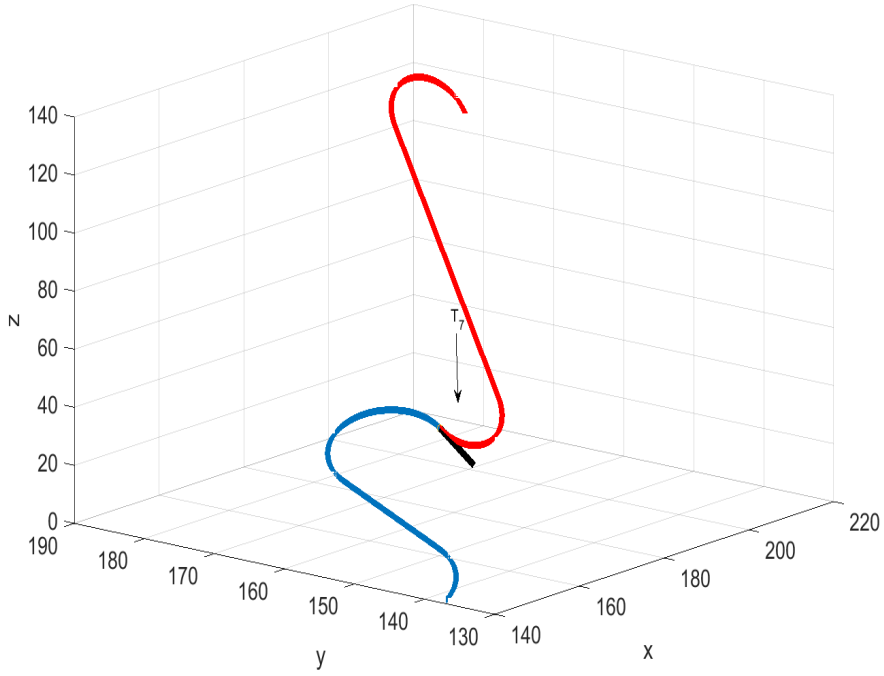


Figure 5.10. 3D Dubins path of AUV1 from T_5 to T_7

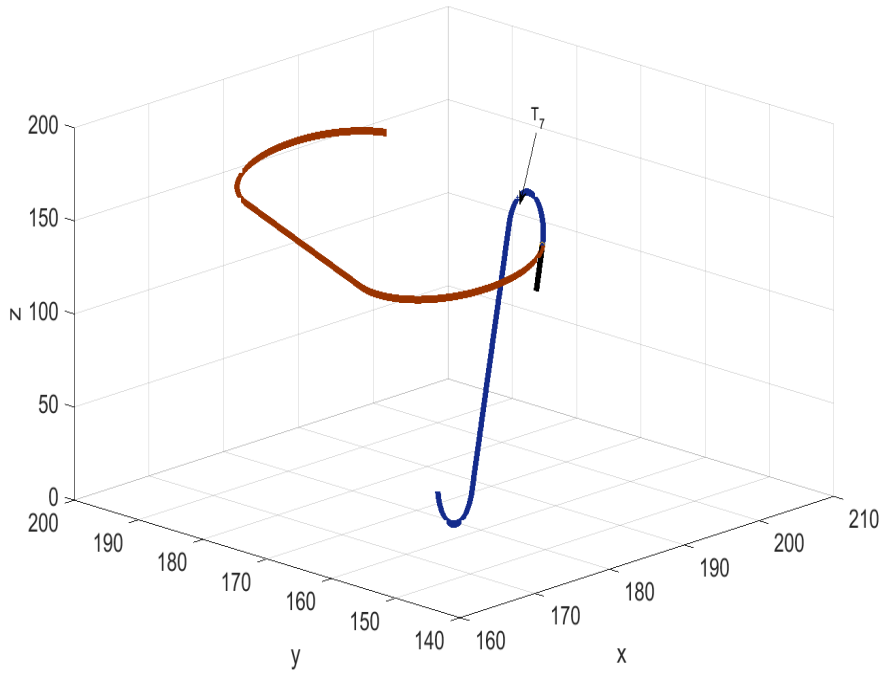


Figure 5.11. 3D Dubins path of AUV1 from T_6 to T_8

the cubic spline line will increase the distance to satisfy the continuity requirement. The unified algorithm satisfies the continuity requirement without increasing the path length and achieves 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.

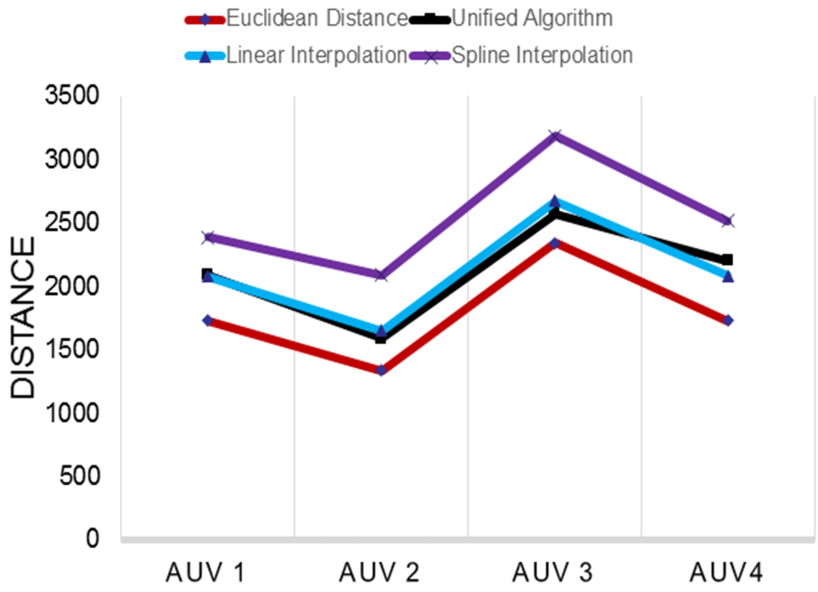


Figure 5.12. Total distance comparison

To represent the difference of the unified algorithm and interpolation method, we apply these method to design 3D path with the same sub-tour sequence. Fig. 5.13 and Fig. 5.14 are shown the difference of unified algorithm and interpolation method. Besides total distance, the unified algorithm have advantage on continuity. Compared with linear interpolation, the unified algorithm and spline interpolation achieve G^2 continuity at joint target. In addition, the unified algorithm has shorter total distance than spline interpolation.

To further demonstrate the continuity of the designed 3D paths, we express the coordinates of the 3D paths as functions $x(l), y(l)$ and $z(l)$, where l is the length from the starting point to any point (x, y, z) on the 3D Dubins path. We compute the second derivatives of x,y,z with respect to l and show them in Fig. 5.15-5.17. At each target point, the second derivative satisfy the definition of G^2 continuity. Hence, the continuity

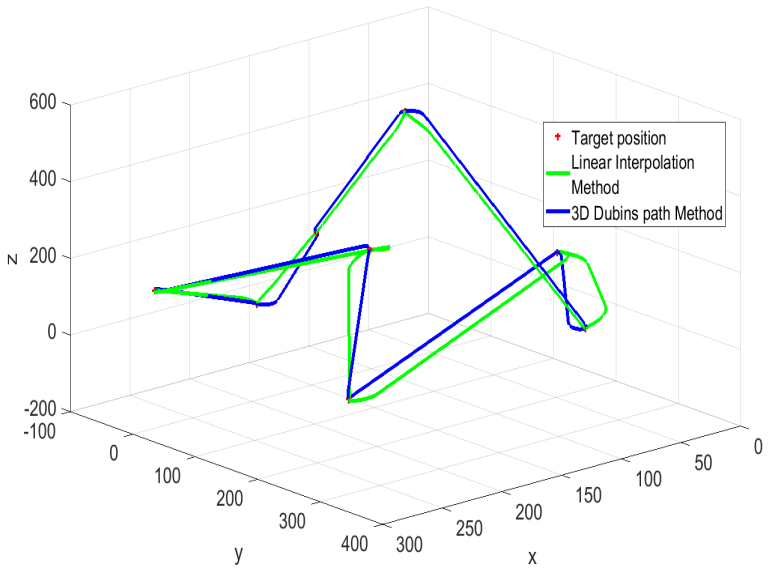


Figure 5.13. Comparison of 3D paths designed by the unified algorithm and linear interpolation

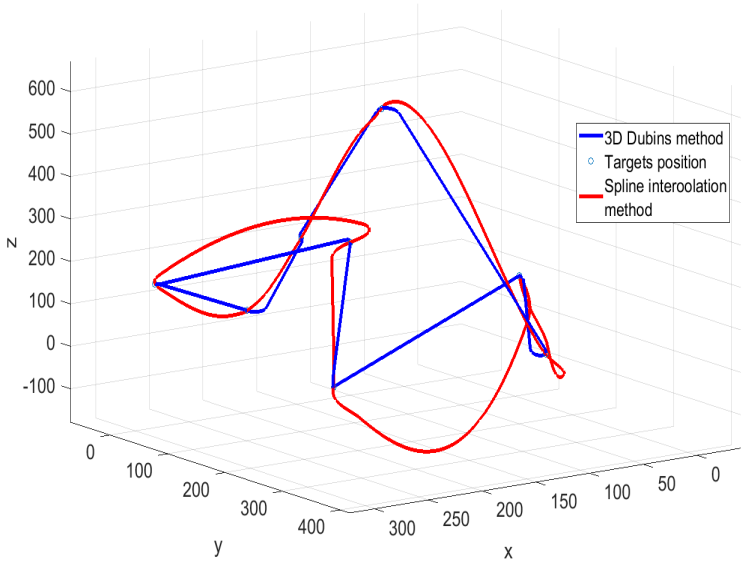


Figure 5.14. Comparison of 3D paths designed by the unified algorithm and spline interpolation

of the new unified algorithm is better than the linear interpolation. Although the spline interpolation also satisfy the G^2 continuity, it has much longer total distance than the unified algorithm.

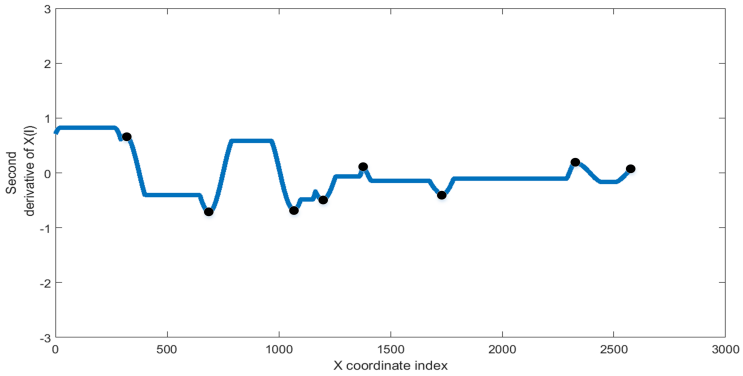


Figure 5.15. Second derivative of $x(l)$

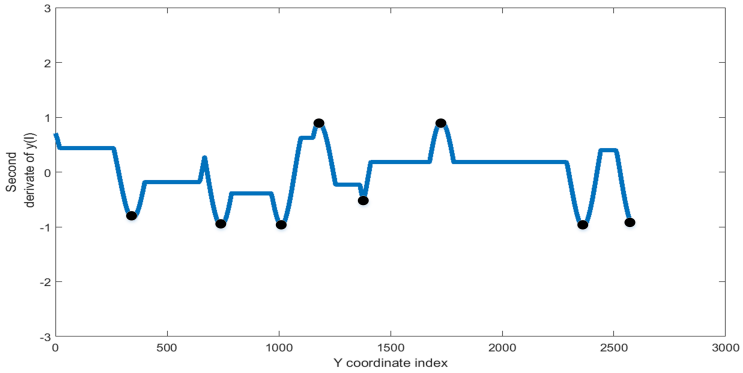


Figure 5.16. Second derivative of $y(l)$

5.3. COMPUTATIONAL COMPLEXITY

The computational complexity of the three algorithms is compared by their simulation time, as shown in Fig. 5.18. We use four AUVs to visit 32, 48, or 60 targets to see the time consumed in the simulation. With the increasing of number of AUV, the time consuming increase quickly. Hence, in our new algorithm, we use computational complexity as cost to reduce the total distance of 3D Dubins path and achieve better continuity.

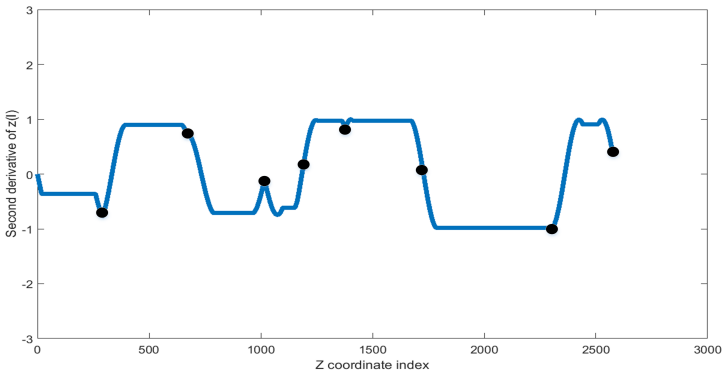


Figure 5.17. Second derivative of $z(l)$

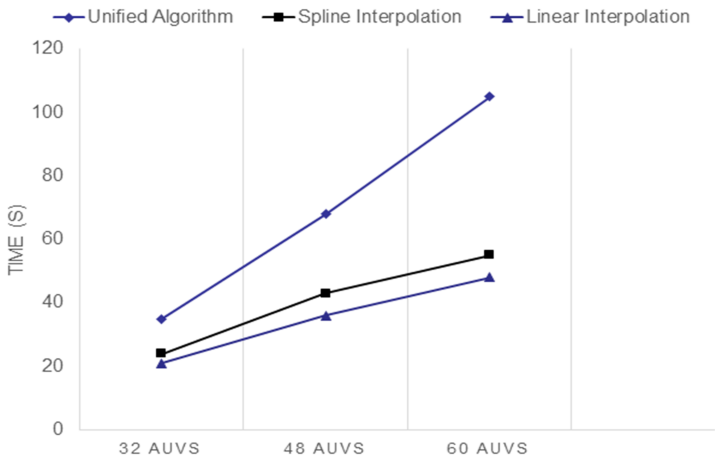


Figure 5.18. Time consumed in simulations of the unified algorithm, spline interpolation, and linear interpolation

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] J. Yuh, "Design and control of autonomous underwater robots: A survey," *Autonomous Robots*, vol. 8, no. 1, pp. 7–24, Jan. 2000.

[3] 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. IEEE, Apr. 2000, pp. 709–716.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] D. Kruger, R. Stolkin, A. Blum, and J. Briganti, "Optimal AUV path planning for extended missions in complex, fast-flowing estuarine environments," in *Proc. IEEE Int. Conf. Robotics and Automation*. IEEE, Apr. 2007, pp. 4265–4270.

[9] 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.

[10] 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.

[11] A. Király and J. Abonyi, "Optimization of multiple traveling salesmen problem by a novel representation based genetic algorithm," in *Proc. 10th Int. Symp. Hungarian Res. Comput. Intell. Informat.*, 2010, pp. 241–269.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

[18] A. M. Shkel and v. Lumelsky, "Classification of the dubins set," *Robotics and Autonomous Syst.*, vol. 34, no. 4, pp. 179–202, 2001.

[19] D. Whitley, "A genetic algorithm tutorial," *Statistics and computing*, vol. 4, no. 2, pp. 65–85, 1994.

VITA

Yiheng Wang received the B.S. degree in electrical from University of Electronic Science and Technology of China, in Chengdu, China, in 2016. He began his master study in August 2016 in the department of Electrical and Computer Engineering at Missouri University of Science and Technology, Rolla, MO, USA. His research interests are task assignment, path planning and optimization algorithm. He received his MS degree of Electrical Engineering from Missouri University of Science and Technology in December 2018.