Path Planning for Unmanned Surface Vehicle (USV) in obstacle-filled environments

1st Thanh-Toan Nguyen
Ho Chi Minh City University of Technology, VNUHCM
Vietnam
toan.nguyen99@hcmut.edu.vn

2nd Tran-Minh-Duc Ho
Ho Chi Minh City University of Technology, VNUHCM
Vietnam
duc.ho279@hcmut.edu.vn

3rd Ngoc-Huy Tran(✉)
Ho Chi Minh City University of Technology, VNUHCM
Vietnam
tnhuy@hcmut.edu.vn

4th Hung Nguyen
HUTECH Institute of Engineering, Ho Chi Minh City University of Technology (HUTECH)
Vietnam
n.hung@hutech.edu.vn

Abstract— **For robots with poor mobility such as high-speed unmanned surface vehicles (USVs) or Unmanned aerial vehicle (UAV), planning a reasonable path for Multi-waypoint missions is very essential. This paper will propose a method to create a smooth curve through given waypoints (WPs) under constraints of the vehicle's turning radius, and in complex obstacle-filled environments. This algorithm is combined with the Line-of-Sight (LOS) guidance and PID controller to become a complete GNC system. We also experimented on an USV model to verify the feasibility of this algorithm.**

Keywords— Unmanned Surface Vehicle (USV), Path planning, Obstacle Avoidance, Dubins, Line-of Sight, PID

I. INTRODUCTION

Unmanned Surface Vehicle (USV) is a self-moving robot that can perform specific tasks on water surface without human's involvement. In recent years, with enormous improvements in Artificial Intelligence and Machine Learning, numerous breakthrough researches have been promoted about USV specifically and Unmanned Vehicles (UGV, AUV, UAV, ...) as a whole. Nowadays, USVs have been implemented for several civil and military uses such as mapping, finding and rescuing, water environment monitoring, water surface supervising, or even shipway [1]….

USVs' motions controlling is more sophisticated than UAV or UGV, considering impacts from wind, water current, the vehicle's inertia. In addition, resistance and response time on water surfaces is greater than vehicles which operate on land or aerially [2]. Consequently, it is necessary to plan a path with enough smoothness for USVs' movements. If straight lines are traditionally used to connect WPs, USV will surpass the path at WPs with relatively high error, especially when moving at high velocity. This leads to great energy consumption. In addition, the robot's discontinuous motion can cause negative mechanical effects to the robot such as slipping, in-stability and wearing of actuators. Since then, path planning is an important part of autonomous system and has important role to improve the performance of the entire system [3][4].

The main purpose of this paper is to plan a path that can pass through given WPs to perform specific tasks, simultaneously have the ability to avoid obstacles with known coordinates. The most vital requirement is that the robot has to approach WPs coordinates accurately; on the other hand, the error between the planned path and vehicle's real trajectory has to be mitigated.

There have been several researches about Multiple Waypoint Path Planning using Cubic Spline [5][6], Bézier [7][8], B-spline [9][10]… With Bezier or Cubic spline, the curve cannot be altered locally. A small change in control points' position can affect directly to entire curve. Additionally, Bezier curve when performing approximate calculation at high order would be complicated; otherwise, connecting multiple low-ordered (3rd order) Bezier or Hermite segments is impossible for local controlling as required [11]. B-splines path with Genetic Algorithm (GA) application, as in previous researches ([9], [10]), has overcome the above disadvantages. Nevertheless, this algorithm is quite complex, and multiple parameters fine tuning as well as long computing time, unsuitable for real-time tasks.

In his classical result [12] (1957), Dubins had pointed out that the shortest smooth path between two points in the 2D plane with its own orientation angle, consists of exactly three path segments. They can be three arc (CCC) or two arc and one line segment (CSC). The main advantage of Dubins method is the shorter path's length in comparison to other methods. However, it's disadvantage is the discontinuous curve curvature, stepping from 0 to $\frac{1}{R}$ at joining points between arcs and lines segment [11].

About obstacle avoidance problem, when initial Dubins path passes through danger area (determined as a circle whose center coincided with that of the obstacle, and radius equaled to the obstacle's radius added with a safety distance), it must be changed to avoid this obstacle. Paper [13] suggests creating a subsequent circular arc coincided with obstacle's circumcircle and then "drive" the Dubins path to travel this arc. However the results only stop at three WPs and one obstacle.

From the above reality, this paper proposes a method for multiple waypoint path planning using Dubins interpolation in two cases: free-space environment and obstacles-filled environment. We as well perform simulations on Matlab/Simulink in the combination of the planned path with

LOS guidance system and PID controllers for speed and heading angle to confirm the feasibility of this method. Lastly, we conduct experiments on an available USV system.

II. Approach

A. Problem statement

The main problem is how to plan a reasonable path for autonomous vehicles with steering system so that it passes multi-waypoints with high accuracy in the 2D obstacle-filled environment. During the mission time, the vehicle speed should be kept unchanged. The distance error between actual path to WPs, actual path length, calculation time of algorithm and cross track error between actual and planned path should be as small as possible.

B. Turning radius

The turning radius, or turning path, of a vehicle is the smallest circular turn that it can make. It can be calculated by (1).

$$R = \frac{u}{\max(r)} \qquad (1)$$

With *u, r* is the vehicle's linear and angular velocity. This paper will examine the turning radius of two models including a model in [14] for simulation and an USV2000 model (Fig. 1) for experiment. The parameters of USV2000 are shown in TABLE I.

Fig. 1. USV2000 Model

TABLE I. Parameter of Cyber Ship II and USV2000

Parameter	Cyber Ship II	USV 2000
L x W (m x m)	1.26 x 0.29	1.8 x 0.8
Weight (kg)	23.8 kg	80 kg
Speed (m/s)	2.0	0.8
Maximum angular velocity (rad/s)	0.5	0.22
Turning radius (m)	4	3.64

C. Path planning using Dubins interpolation

With the task is to find the shortest smooth path that connect 2 points, with the directions of motion. M. Shkel và Lumelsky, in [15], showed six admissible paths: {LSL, LSR, RSL, RSR, RLR, LRL} (L for turn left, R for turn right and S is straight). They calculate the lengths of these 6 lines and choose the shortest one.

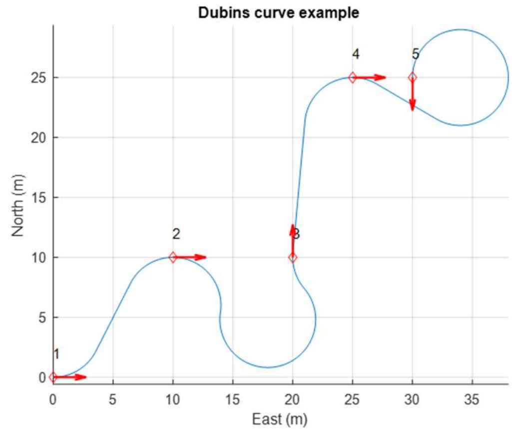

Fig. 2. Multi-waypoints Dubins path with desired directions of motion at each point

Without regard to heading angle conditions, we propose a simpler method

1) *First, create circles whose centers belong to the bisector of the angel between three adjacent WPs and the radius is USV 's turning radius.*
2) *Calculate the rotation directions ρ_i from WP_{i-1} -> WP_i -> WP_{i+1} (i=2:number of WPs-1).*
3) *Calculate the rotation directions from WP_{i-1} -> WP_i -> WP_{i+1} (i=2:number of WPs-1).*
4) *After that, choose 1 of 4 admissible paths according to*

$\rho_i\rho_{i+1}=1$ \ $\rho_i=1$	**TRUE**	**FALSE**
TRUE	RSR	RSL
FALSE	LSR	LSL

5) *Connect these circles of common tangent line to generate a completed Dubins path.*

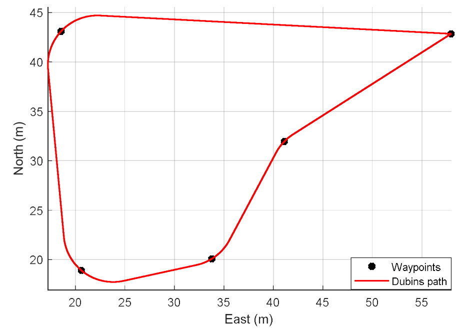

Fig. 3. Multi-waypoints Dubins path without regard to heading angle conditions

D. Obstacle Avoidance Problem

In this section, we refer to the method in [13] to propose a method to create Dubins path with the constraint of known static obstacles on the map. Assume that the obstacles are circles whose center $WP_i WP_{i+1}$ and radius r_b. A safe distance $r_b + d_{safe}$ is also defined because of the possible error between the achieved and planned path. Collision can occur when distance from O_b to the path is less than $r_b + d_{safe}$. In that cases, we will replace the original trajectory with 3 parts consisting of 2 tangent segments from 2 adjacent WPs to the circle $(O_b, r_b + d_{safe})$ and the arc connecting the circle connecting these two tangents.

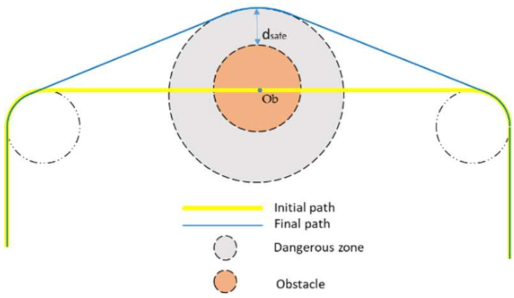

Fig. 4. An example of a Dubins path not go through the round obstacle

This method can be applied similarly to any shape of obstacles if we assume the danger circle is the circumcircle of the obstacle. The main improvement of proposed method compared to [13] is that many obstacles in the orbit of multiple waypoints can be avoided.

Our algorithm for generating completed Dubins path is as follow:

1) *Initialize [m x n] avoid matrix which m is number of WPs – 1, n is number of obstacles.*
2) *Create circles whose centers belong to the bisector of the angel between three adjacent WPs and radius is USV 's turning radius.*
3) *Connect these circles by common tangent lines to generate Dubins path.*
4) *Calculate the cross-track distance dy and along-track distance dx of the centers of obstructions rounded to each segment of the path.*

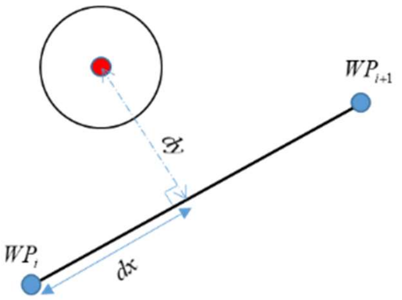

Fig. 5. Cross track distance *dy* and along-track distance *dx*

5) *If* $0 < dx(i,j) < WP_iWP_{i+1}$ *and* $dy(i,j) < r + d_{safe}$ *then predict collision occur -> avoid(i,j) = 1.*
6) *Check each element of avoid matrix. If* $avoid(i,j) = 1$ *then add the safety circle of the* j^{th} *obstacle into Dubins' circle set. The center of this circle is the obstacles 's center or circumcircle of obstacles 's center and the radius is* $r_{safe} = \max(r + d_{safe}, R)$.
7) *If* $avoid(i,j) = 0\ \forall i,j$ *finish generating path. Otherwise, return to step 3.*

E. Guidance and Controller

Guidance, "is the action or the system that continuously computes the reference position, velocity and acceleration of a marine craft to be used by the motion control system" [11]. Guidance systems are divided into many types, but within this paper, we only consider "Guidance for path following" for 2D robots.

Line-of Sight (LOS) is a simple Guidance method, but many experiments and practical applications have proven its effectiveness in the traction system, especially for ships operating on water surface. The LOS guidance system is used to calculate the often combined with an autopilot system to calculate desired heading angle and send it to controller, that "drive" the vehicle to the desired path.

LOS vector is defined with initial point at the center of vehicles and terminal point (x_{LOS}, y_{LOS}) in the tangent of the path at considered point, which the distance from (x_{LOS}, y_{LOS}) to the projection of the vehicle to the path is $\Delta > 0$ can be chosen depend on the ship's dynamics.

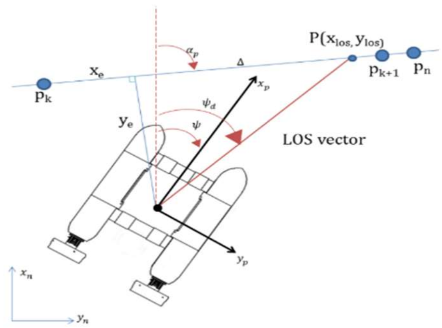

Fig. 6. Line-of Sight (LOS) 's illustration

As Fig. 6, we can calculate desired heading angle according to (2)

$$\psi_d = \alpha_p + \mathrm{atan}(\frac{-y_e}{\Delta}) \qquad (2)$$

Finally, we combine Dubins path with the LOS Guider and PID Controller for heading angle to form a complete GNC system as Fig. 7. Parameters of PID controller for simulation and experiment are shown in TABLE II. For experiments, because of no accurate sensor measuring boat velocity exactly, we only give a fixed PWM to the propulsion.

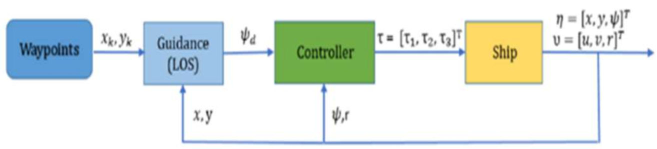

Fig. 7. Overview of a simple GNC system

TABLE II. PARAMETER PID CONTROLLER

[Kp Ki Kd]	CyberShip II	USV 2000
Heading	[10.0 0.0 0 0]	[20.0 0.0 20.0]
Speed	[15.0 1.0 0.0]	80% PWM

III. SIMULATION AND EXPERIMENTAL RESULTS

On the map with 6 waypoints and 3 static obstacles. The first obstacle has a circular shape with small radius and not situated on any line segments connecting WPs. The second one also has a circular shape situated on the line segment between WP1 and WP2. The third one is rectangular-shaped situated on the line segment between WP3 and WP4.

After that, we conduct simulations to compare our result, using Dubins interpolation, with GA – B-spline interpolation, from [10] for possibility proof of suggested Dubins method.

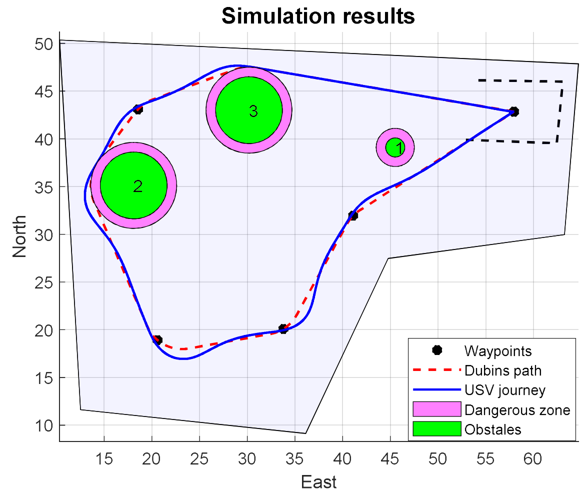

Fig. 8. Result of achieved path with GNC system using Dubins path planning

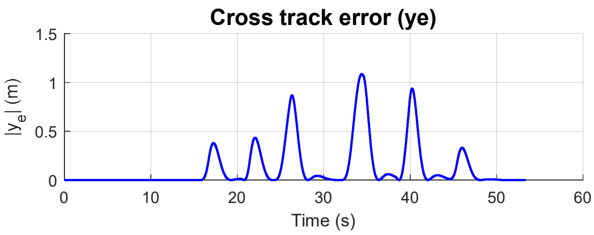

Fig. 9. Graph showing cross-track error, in simulation

TABLE III. DUBINS PATH PLANNING AND PATH FOLLOWING, SIMULATION RESULTS

Criteria	Value
Cross-track error $\frac{RMS}{MAX}$ **(m)**	0.2152 1.0808
Travelled Path Length (m)	118.5870
Path creation time (s)	1.82
Distance error to WPi i= [2 3 4 5]	0.0205 0.1622 0.1026 0.0603
Smoothness	G1

Based on data in Fig. 9. Graph showing cross-track error, in simulation

Table III and Fig. 8, in obstacle-filled environment, path planning gives degree of continuity (G1), and cross-track error less than 1.08 m, can be acceptable compared to the entire length of the trip.

Some other simulations with have also been carried out for this method's correctness guarantee, provided by Fig. 10.

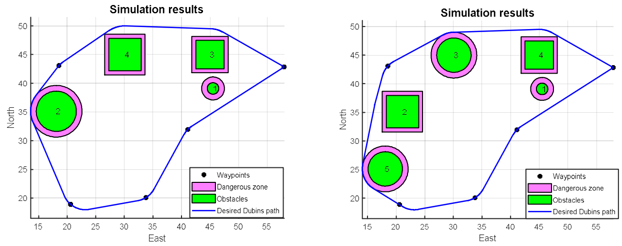

Fig. 10. Planned path using Advanced Dubins in some other case

From simulation results, we conducted experiments for Dubins method. Our location for experiment is a natural lake with low currents (~ 0.1m/s), and partly cloudy. Because of the limit of paper's length, we only present the experimental results for case in Fig. 8.

Fig. 11. Location for experiment

Fig. 12 shows the result of the planned path and achieved path, in experiment, and distance error of the planned path to WPs with the minimum distance to the obstacles can be seen in Table IV.

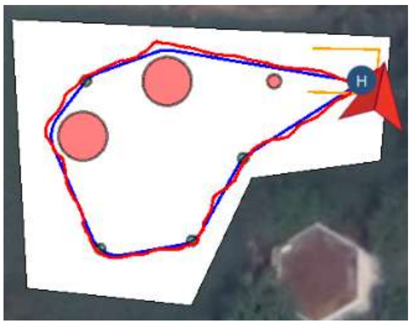

Fig. 12. Experiment result: Actual path (red) and planned path (blue)

TABLE IV. SUMMARY TABLE OF DUBINS PATH 'S CRITERIA, IN EXPERIMENT

Criteria	Value

Defined Path Length (m)	115.7812
Travelled Path Length (m)	128.9262
Cross-track error (RMS/MAX) (m)	0.2145 / 0.9113
Distance error to WPi (m) i = [1 2 3 4 5]	0.0123 0.2541 0.2253 0.2203 0.1040
Minimum distance to Obstacles j (m) j = [1 2 3]	3.2392 0.8400 1.7942

From Fig. 12 and Table IV, the cross-track error between planned path and actual path is 0.2145 m and can be acceptable compared to the entire length of the trip. The error compared with the waypoints is small, and the maximum distance from USV' to the obstacles is 0.84 (m), i.e. encroach into dangerous areas 0.16 m. This means, it is reasonable to choose 1.0 m safe zone (d_{safe}).

In general, USV has basically completed its missions and our proposed method has proven its feasible in experiment.

IV. Conclusion

This paper proposed a multi-waypoints path planning algorithm with Dubins interpolation in an obstacle-filled environment for 2D robots with minimum turning radius, in particular, USV.

The Dubins path with G1 continuity, helps the robots perform better, reduce cross-track error compared with traditional straight-line method. Our proposed method is also effective to plan a suitable, near-optimal trajectory for obstacle avoidance with less computation time. This algorithm has also been verified through simulation and operated on a real system. In the future, it can be used to plan a path with high generalization and low Computation time for 2D and 3D robots with steering system.

However, the discontinuous curvature of Dubins decreases its path following performance, more cross-track error compared with G2 smooth path such as B-spline.

Acknowledgment

This research is funded by Ho Chi Minh City University of Technology (HCMUT), VNU-HCM under grant number To-ĐĐT-2020-08. We acknowledge the support of time and facilities from HCMUT, VNU-HCM and Laboratory of Advance Design and Manufacturing Processes, HCMUT for this study.

References

[1] C. Zhou *et al.*, “The Review Unmanned Surface Vehicle Path Planning: Based on Multi-modality Constraint,” *arXiv*, 2020.

[2] T. van der Sande and H. Nijmeijer, *From cooperative to autonomous vehicles*, vol. 474. 2017.

[3] M. Elbanhawi, M. Simic, and R. Jazar, “Continuous-curvature bounded trajectory planning using parametric splines,” *Front. Artif. Intell. Appl.*, vol. 262, pp. 513–522, 2014, doi: 10.3233/978-1-61499-405-3-513.

[4] A. M. Lekkas, A. R. Dahl, M. Breivik, and T. I. Fossen, “Continuous-Curvature Path Generation Using Fermat’s Spiral,” *Model. Identif. Control A Nor. Res. Bull.*, vol. 34, no. 4, pp. 183–198, 2013, doi: 10.4173/mic.2013.4.3.

[5] B. Wu, Y. Wen, Y. Huang, and M. Zhu, “Research of unmanned surface vessel (USV) path-planning algorithm based on ArcGIS,” *ICTIS 2013 Improv. Multimodal Transp. Syst. - Information, Safety, Integr. - Proc. 2nd Int. Conf. Transp. Inf. Saf.*, no. June 2013, pp. 2127–2136, 2013, doi: 10.1061/9780784413036.285.

[6] J. Lian, W. Yu, K. Xiao, and W. Liu, “Cubic Spline Interpolation-Based Robot Path Planning Using a Chaotic Adaptive Particle Swarm Optimization Algorithm,” *Math. Probl. Eng.*, vol. 2020, 2020, doi: 10.1155/2020/1849240.

[7] F. Zhou, B. Song, and G. Tian, “Bézier curve based smooth path planning for mobile robot,” *J. Inf. Comput. Sci.*, vol. 8, no. 12, pp. 2441–2450, 2011.

[8] Y. J. Ho and J. S. Liu, “Collision-free curvature-bounded smooth path planning using composite bezier curve based on voronoi diagram,” *Proc. IEEE Int. Symp. Comput. Intell. Robot. Autom. CIRA*, pp. 463–468, 2009, doi: 10.1109/CIRA.2009.5423161.

[9] M. Behroo and A. Banazadeh, “Near-optimal trajectory generation, using a compound B-spline interpolation and minimum distance criterion with dynamical feasibility correction,” *Rob. Auton. Syst.*, vol. 74, pp. 79–87, 2015, doi: 10.1016/j.robot.2015.07.003.

[10] N. H. Tran, A. D. Nguyen, and T. N. Nguyen, “A Genetic Algorithm Application in Planning Path Using B-Spline Model for Autonomous Underwater Vehicle (AUV),” *Appl. Mech. Mater.*, vol. 902, pp. 54–64, 2020, doi: 10.4028/www.scientific.net/amm.902.54.

[11] A. M. Lekkas, *Guidance and Path-Planning Systems for Autonomous Vehicles*, no. April. 2014.

[12] Dubin, “Dubins Curves 1957 original,” *J. Math.*, vol. 79, no. 3, pp. 497–516, 2012.

[13] D. Yang, D. Li, and H. Sun, “2D dubins path in environments with obstacle,” *Math. Probl. Eng.*, vol. 2013, 2013, doi: 10.1155/2013/291372.

[14] R. Skjetne, Ø. N. Smogeli, and T. I. Fossen, “A Nonlinear Ship Manoeuvering Model: Identification and adaptive control with experiments for a model ship,” *Model. Identif. Control*, vol. 25, no. 1, pp. 3–27, 2004, doi: 10.4173/mic.2004.1.1.

[15] A. M. Shkel and V. Lumelsky, “Classification of the Dubins set,” *Rob. Auton. Syst.*, vol. 34, no. 4, pp. 179–202, 2001, doi: 10.1016/S0921-8890(00)00127-5.