Fermat Curve Path Planning Method for Ship Trajectory Tracking

Daoke Li
School of Navigation, Fujian Chuanzheng Communications College, Fuzhou 350007, China
E-mail: 13003886586@163.com

Keywords: ship, trajectory tracking, path-planning, guidance, accuracy

Received: February 20, 2024

Traditional ship motion systems are no longer sufficient to meet the navigation demands of vessels. Therefore, a path-planning method based on Fermat curves is proposed in this study. Within the ship operation system, Fermat curves are employed to enhance straight-line paths, achieving smooth motion between waypoints. The Fermat curve path-planning method is integrated into ship guidance algorithms to guide vessel headings. The performance of the proposed guidance algorithm was evaluated, and experimental results indicated that the algorithm can maintain a high similarity between actual and desired trajectories. In comparison to traditional Line-of-sight methods, the proposed algorithm demonstrated shorter path-planning lengths. Additionally, the algorithm's heading angles can compensate for sideslip angles in perturbed environments. Comparative experiments with other guidance algorithms revealed that the proposed algorithm had the shortest runtime, saving more energy compared with other's performance. Furthermore, it achieved accuracy and precision rates of 87.1% and 91.3%, respectively, surpassing other algorithms and providing high-precision guidance for vessels. In terms of error comparison, the proposed algorithm kept guidance errors around 2, showcasing superior performance and feasibility in ship path guidance.

Povzetek: Članek obravnava metodo načrtovanja poti za sledenje ladijskih trajektorij, ki temelji na Fermatovih krivuljah. Eksperimentalni rezultati kažejo, da predlagani algoritem vodenja dosega visoko podobnost med dejanskimi in želenimi trajektorijami, krajše dolžine načrtovanja poti ter učinkovito kompenzira odklone v motenih okoljih.

1 Introduction

Ship trajectory tracking is a widely applied technology in the maritime domain, allowing for the recording and monitoring of vessel movements at sea [1-2]. This technology encompasses key areas such as automated ship navigation, collision avoidance, and path-planning, playing a crucial role in maritime safety, ship management, and marine environmental protection. Traditional ship trajectory planning methods often rely on approaches based on the shortest path or optimal control theory. However, these methods often neglect specific constraints in navigation, such as vessel speed limitations, channel restrictions, and absolute safety requirements [3-4]. To address these issues, this paper introduces the Fermat curve path-planning method. The Fermat curve path-planning method, developed based on the theory of ray tracing, has found widespread application in telecommunications, aviation, transportation, and other fields. The primary idea behind Fermat curve path-planning is to select points along the path to minimize the total travel time from the starting point to the destination. This aligns with the objective in ship trajectory tracking, which aims to find an optimal path ensuring vessels reach their destination safely and efficiently. This research applies the Fermat curve path-planning method to ship trajectory tracking, designing a guidance algorithm based on this method. The algorithm ensures precise navigation of vessels along the defined path while meeting specific constraints.

The research is structured into four main parts. The first part summarizes the researches on tracking technology and path-planning, analyzing their achievements. The second part constructs the path-planning method based on Fermat curves. The third part validates the effectiveness and feasibility of the proposed method through experiments. The final part analyzes the results, identifies shortcomings in the research, and suggests future directions for study.

2 Related works

In the research on ship path planning, numerous scholars combined artificial intelligence for design and exploration [5-6]. Meng et al proposed an approach based on potential field functions and reflexive rewards. They integrated the concepts of multi-agent and multi-task learning, forming a new training framework for the path-planning of mobile robots. Within pathfinding scenarios, this method demonstrated higher rates of successful exploration and stability, effectively charting the shortest routes while reducing energy consumption [7]. Rath and associates navigated humanoid robots within complex environments using a combination of genetic algorithms and neural networks for control.

Initially, a genetic algorithm controller determined the robot's initial turning angle, which was later combined with a neural network controller to generate the final turning angle. Both simulation and experimental outcomes indicated that there is satisfactory consistency in navigation parameters, with minimal error constraints, validating the functionality of the proposed hybrid controller [8]. Garcia and co-researchers devised efficient collision-free multi-robot path-planning solutions for controlled environments, extending previous research. This solution merged the optimization capabilities of the A* algorithm with the search capabilities of cooperative evolution algorithms. The output constituted a set of collision-free paths derived from either the A* algorithm or the collaborative evolution process. These routes were dynamically generated and feasible for implementation on edge computing devices [9]. Li and team introduced an angle-guided ant colony algorithm to address issues prevalent in using ant colony algorithms for mobile robot path-planning, such as susceptibility to local optima and slow convergence. Their algorithm also differentially updated pheromones for paths of varying quality and introduced a pheromone chaotic disturbance update mechanism. Simulations showcased the algorithm's robust global search capability, breaking out of local optima and swiftly converging towards global optimal solutions [10].

Researchers such as Liu and Sang investigated trajectory tracking control issues in the aerospace field, proposing and designing a trajectory controller. This controller was based on the nonlinear adaptive integral backstepping method, featuring a two-level control structure that took into account various underactuated scenarios. Simulation results demonstrated that the proposed controller performed well in tracking tasks, even under conditions of parameter uncertainty [11]. In the field of path tracking, Telen and other scholars introduced a polynomial numerical path tracking algorithm. This algorithm was based on adaptive step-size prediction, prevents path jumps. Comparative experiments with other algorithms revealed that this algorithm exhibited optimal path tracking capabilities [12]. In satellite ship detection and tracking, Bai and colleagues presented an anti-similar shape interference method. They initially used a cross-convolutional network to compute similarity, followed by hierarchical dual-layer regression. The final design included a discriminative model to enhance filtering capabilities. Results indicated that, compared to other methods, this approach effectively reduced interference from similar ship details, achieving a tracking accuracy rate of 83.4% [13]. Addressing the low accuracy of tracking algorithms in remote sensing images, Wu's team proposed a multi-object tracking algorithm. The study utilized a multi-granularity network (MGN) to extract target appearance information, enhancing image generalization. Comparative experiments with other algorithms demonstrated that the proposed algorithm had the fastest average running speed, improved tracking accuracy, and met the requirements of tracking tasks [14].

Table 1: Summary of related work

Researchers	Year	Technical Methods	Key Findings	Limitations
Meng and Zhang	2022	Potential field function and reflective reward-based approach	High successful exploration rate and stability with minimized energy consumption	Trajectory planning in complex ocean conditions may not be adequately considered
Rath et al.	2021	Hybrid controller with genetic algorithm and neural network	Consistency and minimized error in navigation in complex environments	Controller design may not be suitable for dynamic changes in the marine environment
Garcia et al.	2023	Integration of A* algorithm and cooperative evolutionary algorithm	Efficient collision-free multi-robot path-planning	May face unique challenges when applied at sea
Li et al.	2022	Angle-guided ant colony algorithm	Improved global search capability and breakthrough to local optimization	Application of ACO algorithms to marine trajectory planning is not yet well defined
Liu and Sang	2018	Nonlinear adaptive integral backstepping	Good trajectory tracking performance under parameter uncertainties	Mainly for aviation and may not be directly applicable to marine navigation
Telen et al.	2020	Polynomial numerical path tracking algorithm	Optimized path tracking capability	Lack of consideration of the dynamic marine environment
Bai et al.	2022	Ship tracking method against similar shape interference	Improve tracking accuracy by reducing interference from similar ship details	Focused on satellite video, which may not be applicable to real-time trajectory planning

Wu et al.	2021	Deep learning multi-target tracking algorithm	Improve tracking accuracy of remote sensing images	Mainly focused on image processing, far from direct path-planning

The above related works are summarized in Table 1, which clearly and directly gives the main techniques, main results and limitations of each method. As can be seen from Table 1, important research results have been achieved in the field of tracking techniques and path-planning, but most of the path-planning and tracking techniques have focused on the study of ordinary roads, and there are still few researches on these two techniques in the field of ship trajectory path-planning. Therefore, the research combines the Fermat curve to realize the smooth transition of ship trajectory from curve to straight line, and completes the ship trajectory tracking problem using the optimized Fermat curve.

3 Ship path-planning method based on fermat curves

Ship guidance is divided into two parts. The first part involves path-planning based on Fermat curves, addressing the navigation problem between straight lines. The second part focuses on ship guidance algorithms.

3.1 Fermat Curve Path-planning

In maritime navigation, to enable autonomous navigation of ships in unmanned conditions, a comprehensive ship motion control system needs to be designed [15-17]. The autonomous ship running system consists of four components: control system, guidance system, navigation system, and path-planning. The schematic diagram of the autonomous ship running system is shown in Figure 1.

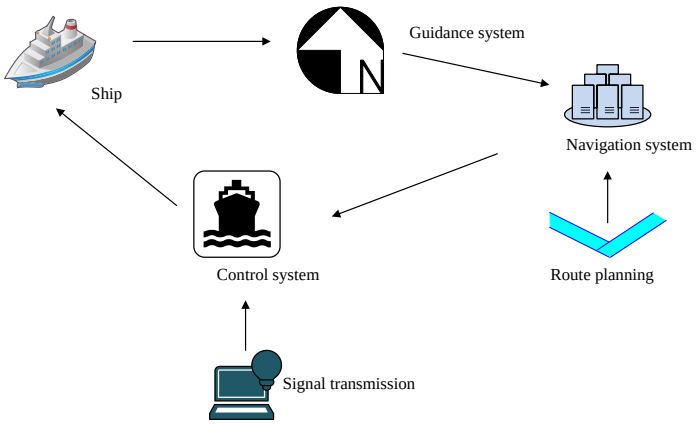


Figure 1: Diagram of ship autonomous operation system

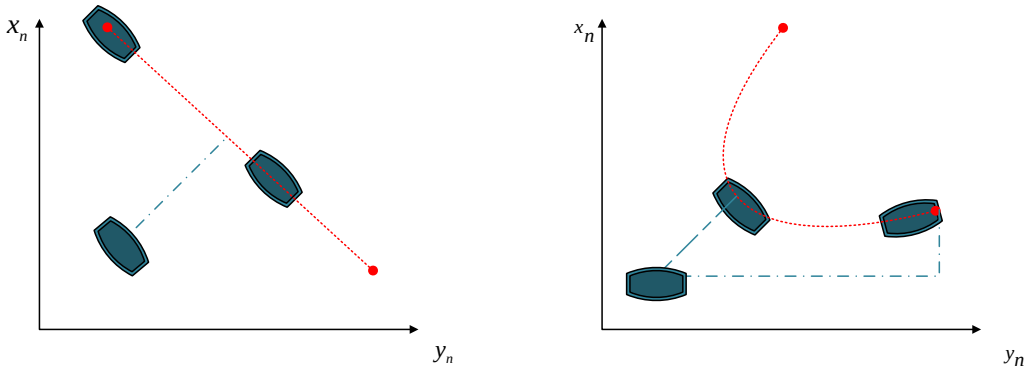
Figure 1 depicts the schematic diagram of the autonomous ship running system. The specific task of path-planning is to design the optimal path within the constraints of the maritime environment and ship conditions. Ship path-planning is characterized by complexity, randomness, multiple constraints, and multiple objectives. It considers the functionality of the ship, and takes into account navigation constraints. Path-planning involves three stages, namely, environmental modeling, path smoothing, and path searching. The flat path can be represented as different one-dimensional points, specifically expressed in Formula (1).

$$P := \left\{ P \in \Re^2 \middle| P = P_p(\overline{w}) \forall \overline{w} \in \Re \right\} \quad (1)$$

In Formula (1), $\overline{w}$ represents the parameter of the path, where the path parameter is defined as the path length, and $\overline{w}$ ranges from 0 to 1, defining $\overline{w}$ as the unit domain. $P_p(\overline{w})$ represents the specific location of waypoints in the path. Curved paths may also exist in the flat path, which can be expressed as Formula (2).

$$P_i := \left\{ P_i \in \Re^2 \middle| P_i = P_{i,p}(\overline{w}) \forall \overline{w} \in I_i = \left[\overline{w_{i,0}}, \overline{w_{i,1}} \right] \subset \Re \right\} \quad (2)$$

In Formula (2), P_i represents the curved segment on the flat path. In ship path-planning, fixed waypoints are used, and lines are drawn to connect them. The connected straight line represents the desired ship trajectory. The connection of waypoints can be either in a straight line or in a curved path, as shown in Figure 2.



(a)Linear path tracking (b)Curve path tracking

Figure 2: Waypoint connectivity

Figure 2 illustrates the waypoint connection methods. Figure 2(a) is an illustration of straight-line path tracking, while Figure 2(b) is an illustration of curved path tracking. Straight-line path tracking involves ships passing through waypoints in a straight line, minimizing distance and saving materials and time. The parametric equation for a straight-line trajectory can be expressed in Formula (3).

$$P_{line}(\bar{w}) = \begin{bmatrix} N_0 + L\bar{w}\cos(x_i) \\ E_0 + L\bar{w}\sin(x_i) \end{bmatrix} \tag{3}$$

In Formula (3), $p_0 = [N_0, E_0]^T$ represents the starting point's position, L denotes the path length, and x_i represents the angular deviation of the straight path. The curvature of the straight path is continuous, but the heading deviation is discontinuous. Due to the discontinuity in heading deviation on a straight path, continuous navigation is not possible between consecutive straight paths. The vessel needs to adjust its direction before continuing the journey. This mode of navigation is disadvantageous as stopping the vessel for adjustments can increase safety risks, accelerate machine wear, and extend travel time. Consequently, straight paths present issues in practical navigation, prompting modifications with curved paths. Common curved paths include circular arc curves and turning curves. In the combination of straight and curved paths, curved paths are often employed to connect at the turning points of straight paths. In circular arc curves, the shortest possible trajectory between the starting and ending points exhibits the maximum curvature, with a maximum of three segments forming the path. Each segment is either a straight line or a circular arc. The parametric equation for circular arc curves is represented in Formula (4).

$$P_{cir}(\bar{w}) = \begin{bmatrix} C_N + R\cos(\alpha_0 + \bar{w}(\alpha_1 - \alpha_0)) \\ C_E + R\sin(\alpha_0 + \bar{w}(\alpha_1 - \alpha_0)) \end{bmatrix} \tag{4}$$

In Formula (4), $C = [C_N, C_E]$ denotes the midpoint of the circular arc segment, α_1 represents the angle when the vessel turns from the arc to the straight line, α_0 represents the angle when the vessel turns from the straight line to the arc, and R denotes the radius of the circular arc. Combining the parametric equation of the straight path, the expression for a path that includes both circular arc curves and straight lines is known. Under this condition, the curvature and heading deviation of this path can be calculated. However, the curvature at the turning points between the straight and curved segments is discontinuous, which leads to a discontinuity in the desired lateral acceleration of the route, affecting the input to the controller and the functionality of the vessel. The parameters for calculating turning curves are given in Formula (5).

$$P_{cl} = \begin{bmatrix} x_0 + \int_0^{\bar{w}} \cos(\frac{1}{2}c\xi^2 + k_0\xi\_x_0)d\xi \\ y_0 + \int_0^{\bar{w}} \sin(\frac{1}{2}c\xi^2 + k_0\xi\_x_0)d\xi \end{bmatrix} \tag{5}$$

In Formula (5), $P_0 = [x_0, y_0]^T$ represents the initial position of the curve, k_0 represents the curvature at the initial position P_0, c describes the sharpness coefficient for curvature increase, ξ is an assumed integration variable, and x_0 represents the heading angle at the initial position P_0. Since the heading deviation is continuous in paths containing turning curves and straight lines, the curvature at their turning points is also continuous. The path curve formed by turning curves and straight lines increases linearly and exhibits continuity. Turning curves, represented by Fermat curves, can be used as transition curves to replace circular arc curves. When replacing circular arc curves with turning curves, two segments of turning curves are typically used. The first segment has an increasing curvature from the straight line's endpoint to the curve's midpoint, and the second segment has a decreasing curvature from the curve's midpoint to the starting point of the next straight line. In this process, both segments of turning curves are identical, requiring only one calculation. Vessels cannot make continuous motion transitions between two straight

lines; hence, Fermat turning curves are designed for this transition. The equation for the Archimedean curve is given in Formula (6).

$$r = k\theta^{1/n} \tag{6}$$

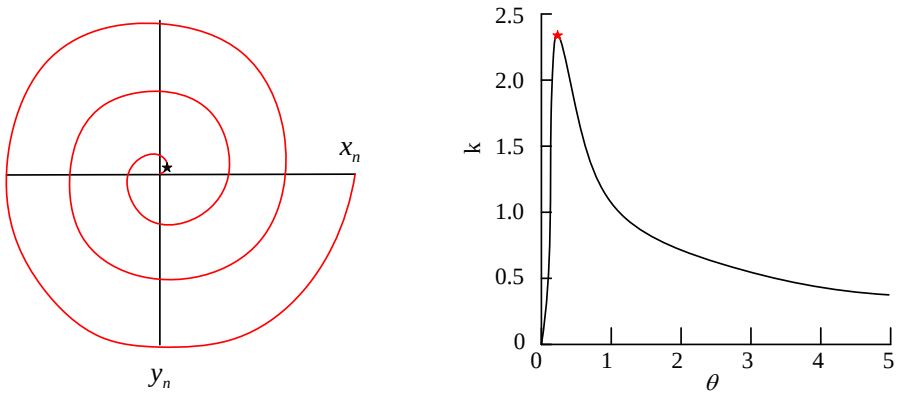
In Formula (6), k is the constant of measurement, θ represents the polar angle, r is the polar radius, and n is a natural number. When $n=2$, it represents the Fermat curve equation, expressed as Formula (7).

$$r = k\sqrt{\theta} \tag{7}$$

The radius of the transition curve can be calculated using Formula (7), and the curve radius increases with the increase of the pole angle. The calculation method for Fermat curve curvature is expressed as Formula (8).

$$k(\theta) = \frac{1}{k}\frac{2\sqrt{\theta}(3+4\theta^2)}{(1+4\theta^2)^{\frac{3}{2}}} \tag{8}$$

The curvature of the transition curve can be calculated using Formula (8). Curvature represents the curvature degree of a curve, and the curvature calculation formula for Fermat curves can calculate the local maximum or minimum points. The Fermat curve equation can be used to obtain the Fermat curve graph, and the Fermat curve curvature calculation formula can be used to obtain the Fermat curve curvature graph, as shown in Figure 3.



(a)Fermat curve (b)Fermat curvature of curve

Figure 3: Fermat curve and curve curvature plot

Figure 3 shows the Fermat curve and the curvature plot, with the labeled positions indicating the points of maximum curvature. The study employs a two-dimensional parameterization for trajectory points, representing the Fermat curve equation using Cartesian parameterization, as expressed in Formula (9).

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} r\cos(\theta) \\ r\sin(\theta) \end{bmatrix} = \begin{pmatrix} k\sqrt{\theta}\cos(\theta) \\ k\sqrt{\theta}\sin(\theta) \end{pmatrix} \tag{9}$$

Formula (9) allows the representation of polar coordinates using the coordinates of two-dimensional points. The transitional curve studied involves two segments of helical curves. The equation for the first segment, where curvature increases, can be expressed as Formula (10).

$$P_{FS}(\theta) = \begin{bmatrix} x_0 + k\cos(\rho\theta + x_0)\sqrt{\theta} \\ y_0 + k\sin(\rho\theta + x_0)\sqrt{\theta} \end{bmatrix} \tag{10}$$

In Formula (10), x_0 represents the endpoint of the straight line. When the turn is counterclockwise, $\rho = 1$. When the turn is clockwise, $\rho = -1$. The parameter θ takes values between 0 and θ_{end}, where θ_{end} corresponds to the midpoint of the helical curve on the coordinates. The equation for the second segment, where curvature decreases, can be expressed as Formula (11).

$$\overline{P_{FS}}(\theta) = \begin{bmatrix} x_{end} + k\sqrt{\theta_{end} - \theta}\cos(\rho(\theta - \theta_{end}) + x_{end}) \\ y_{end} + k\sqrt{\theta_{end} - \theta}\sin(\rho(\theta - \theta_{end}) + x_{end}) \end{bmatrix} \tag{11}$$

In Formula (11), x_{end} represents the starting point of the lower straight line. The first and second segments of the helical curve are identical, but the DNA steering angle differs. For the second segment, θ takes values between θ_{end} and 0. Therefore, the two segments of the curve can be considered as helical curves with the same trajectory but different directions.

3.2 Implementation of the ILOS guidance algorithm

Fermat curves are mainly used to optimize the ship's trajectory in the path-planning problem through its unique curve shape, to achieve a smooth transition of the path and improve the efficiency of path-planning. By utilizing the curvature continuity of the Fermat curve, this study

designs a route that meets the dynamic characteristics of the ship as well as accurate navigation. Specifically, the Fermat curve reduces the abrupt change of heading angle, which enables the ship to maintain a more stable heading during the steering process, thus reducing the extra traveling distance and time caused by heading adjustment. In addition, the method demonstrates significant advantages when dealing with complex routes, especially in navigation in narrow or congested waters, as it allows for more precise control of the ship's position and heading, avoiding unnecessary adjustments and maneuvers. The Fermat curve path-planning is integrated into the guidance system to ensure that the vessel maintains motion according to the reference path. A guidance algorithm is implemented in the system to provide guidance when the vessel deviates from the desired path. Commonly used guidance algorithms include linear trajectory guidance algorithms and curved trajectory guidance algorithms [18-20]. The specific operation of the linear trajectory guidance algorithm involves obtaining the position deviation and heading deviation of the vessel from the planned heading based on the trajectory and then providing guidance based on the signals. The study utilizes the Line-of-sight (LOS) guidance algorithm, as illustrated in Figure 4.

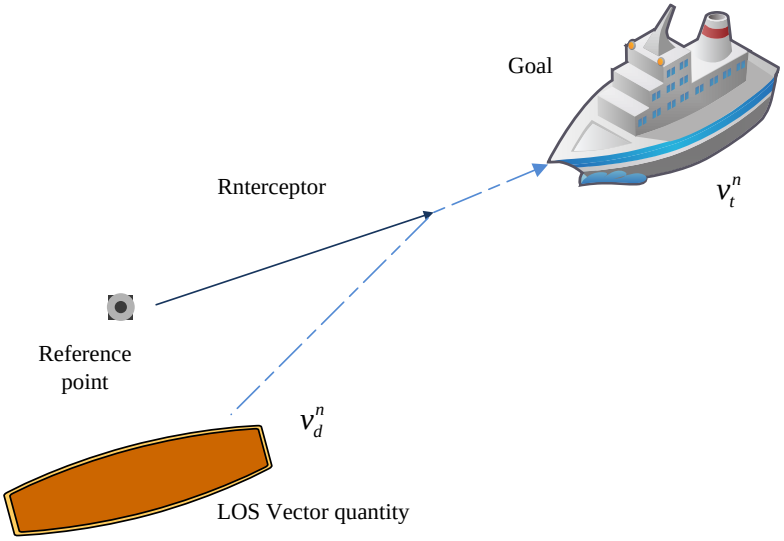


Figure 4: Schematic diagram of the LOS algorithm

Diagram 4 illustrates the principle of the LOS algorithm. From the diagram, it can be observed that the LOS algorithm involves three guiding points, which are the fixed reference position, the ship itself, and the target point. In this context, v_a^n ' represents the interceptor's speed, and v_d^n signifies the desired speed. The LOS algorithm defines the ship's velocity in two dimensions, as expressed in Formula (12).

$$U(t) = \|v(t)\| = \sqrt{\dot{x}(t) + \dot{y}(t)} \geq 0 \qquad (12)$$

In Formula (12), $U(t)$ denotes the ship's speed, while $\dot{x}(t)$ ' and $\dot{y}(t$ respectively represent the average velocities on the horizontal and vertical coordinates. Based on the fixed coordinate system along the path, the ship's coordinates are described by Formula (13).

$$\varepsilon(t) = R_P(\psi_k)^T (p_n(t) - p_k^n) \qquad (13)$$

In Formula (13), p_k^n represents the coordinate system's origin, and ψ_k represents the positive angle by which the origin has been rotated. According to the LOS guidance rules, the ship's trajectory angle is divided into two parts, as represented by Formula (14).

$$\psi_{LOS}(e) = \psi_p + \psi_r(e) \qquad (14)$$

In Formula (14), $\psi_p = \psi_k$ represents the heading cut angle of the ship's trajectory. A moving point is set as the point where the ship arrives at each moment, hence the heading angle needs to align with the LOS vector direction. As for the trajectory tangential angle, it is illustrated in Figure 5.

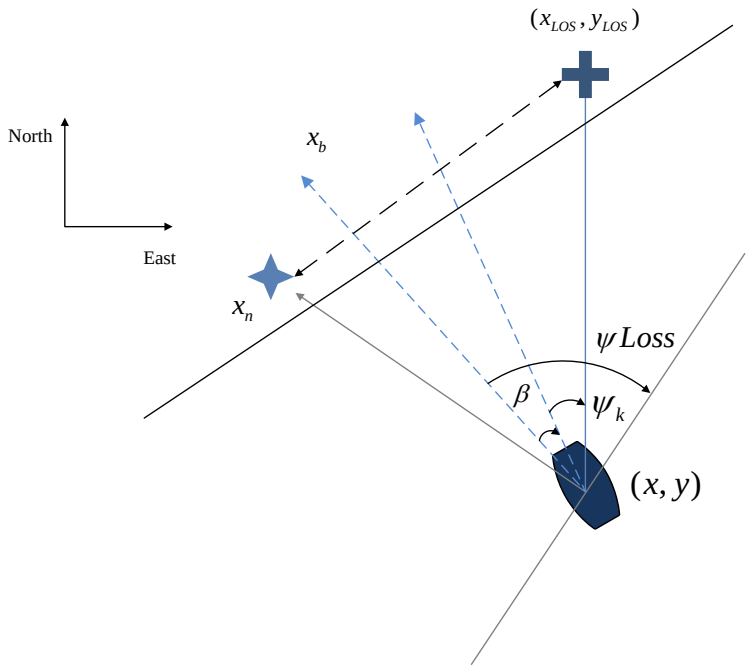


Figure 5: Schematic diagram of LOS guidance law parameters

Figure 5 depicts a schematic of LOS guidance rule parameters. As shown, the LOS guidance algorithm involves relatively low computational complexity. At each moment, the LOS vector starts from the ship's position coordinates and ends at the desired coordinates. When the two heading angles are the same, and the ship's overall velocity is not aligned with the x-axis direction of the fixed coordinate system, a nonzero longitudinal tracking error occurs. To align the overall velocity with the heading angle, a compensatory sideslip angle is required. There are various methods to calculate the compensatory sideslip angle, but it is challenging to estimate in situations where the ship encounters slow disturbances such as ocean currents. Therefore, the research focuses on designing an integral guidance system for ship trajectory tracking in straight-line motion. In the presence of longitudinal oscillation error, integrating to eliminate the slow disturbance changes becomes crucial for the ship control system. The variable integral Line-of-Sight (ILOS) algorithm's guidance rule is expressed in Formula (15).

$$\begin{cases} \psi_{mlLOS} \cong -\arctan(K_p d + e^{-\rho|d|} y_d) \\ \dot{y}_d = \dfrac{e^{-\rho|d|} y_d}{\sqrt{1+(k_p d + e^{-\rho|d|} y_d)}} \\ \psi_d = \psi_{ILOS} - \psi_k \end{cases} \tag{15}$$

In Formula (15), ρ ' represents a design parameter. When the ship is at a considerable distance from the desired trajectory, the integral gain is low. Consequently, the integral effect serves as an auxiliary rather than a dominant factor, allowing the ship to align with the desired trajectory at a faster pace. Simultaneously, the integral gain does not overly saturate.

4 Performance study of guiding algorithm based on fermat curves

By integrating Fermat curves into the ILOS guidance system, this research made key modifications to the algorithm, mainly including optimizations to the ship control logic and improvements to the trajectory generation algorithm. These modifications enabled the guidance system to be more flexible in responding to complex sea conditions, improving the reliability of the route and the operational efficiency of the ship. For example, by dynamically adjusting the parameters of the Fermat curve, the guidance system could optimize the heading angle in real time, so that the ship could maintain an optimal course even under unfavorable conditions such as strong winds and tides. In addition, the introduction of the Fermat curve enhanced the control accuracy of the guidance system for the ship's speed and position, especially in operations such as emergency avoidance and precise docking, showing better performance. The proposed algorithm in this study was analyzed by first comparing it with the traditional LOS algorithm to validate its effectiveness. Subsequently, comparisons were made with other algorithms to verify its advancements.

4.1 Implementation process of guiding algorithm based on fermat curves

Utilizing the ship controller and disturbance adaptive rules, the Fermat Curve-based Improved ILOS guiding algorithm demonstrated its functionality in the ship

motion system. The experiment employed a model ship, designed autonomously in a marine laboratory, with a total mass of 224.5 kg and a length of 2.8 m. The laboratory setup details are provided in Table 2.

Table 2: Laboratory environment setup

Hardware and software configuration	Version model
CPU	Intel(R)Core i7-7700@3.6GHz
GPU	GTX 1060
RAM	32G
Display memory	6G

Subsequently, the desired values, initial position, simulation time, and other parameters for the ship model were configured. Simulation experiments were conducted to track the trajectory of the ship model in various directions, and the results are illustrated in Figure 6.

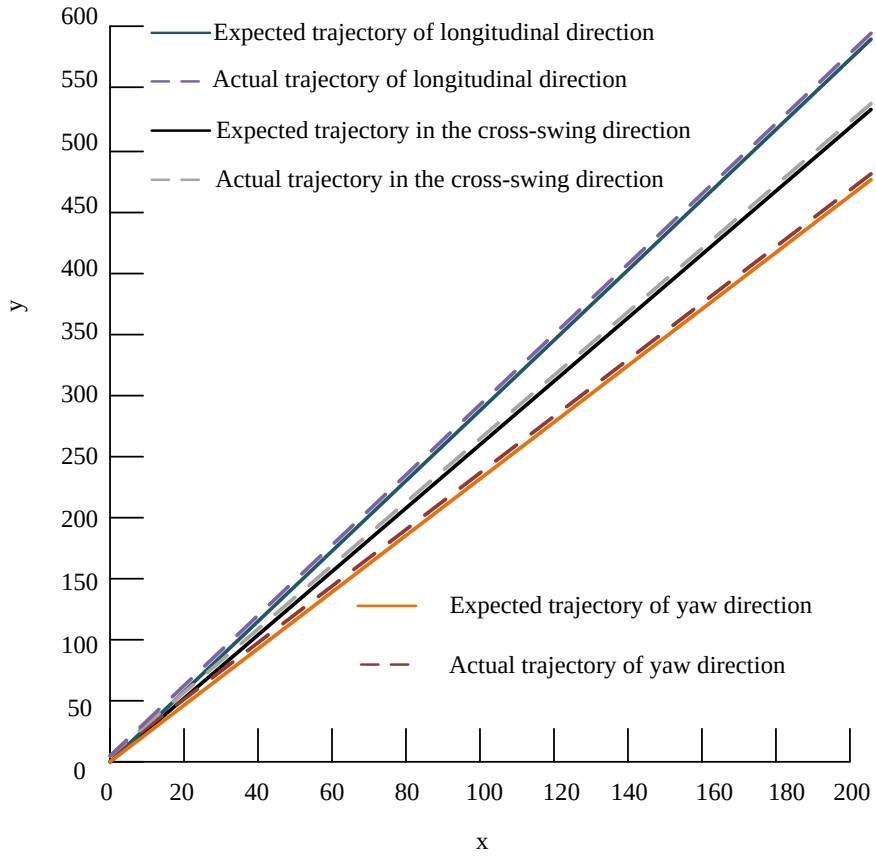


Figure 6: Desired versus actual trajectories of ship models in different directions

In Figure 6, dashed lines represented the expected trajectory, while solid lines represented the actual trajectory of the ship. As the distance covered by the ship increased, variations in the trajectory were observed in different directions, with longer tracking distances in the heave direction. In heave, sway, and yaw directions, the expected and actual trajectories exhibited remarkable consistency. These directions were the primary control degrees of freedom for dynamically positioned ships, indicating that the ILOS guiding algorithm achieved high consistency between expected and actual trajectories, demonstrating the ship's guidance and control system's ability to track the trajectory comprehensively. Under the same controller and adaptive rules, different algorithms were applied to track the ship's trajectory in the same laboratory conditions, as depicted in Figure 7.

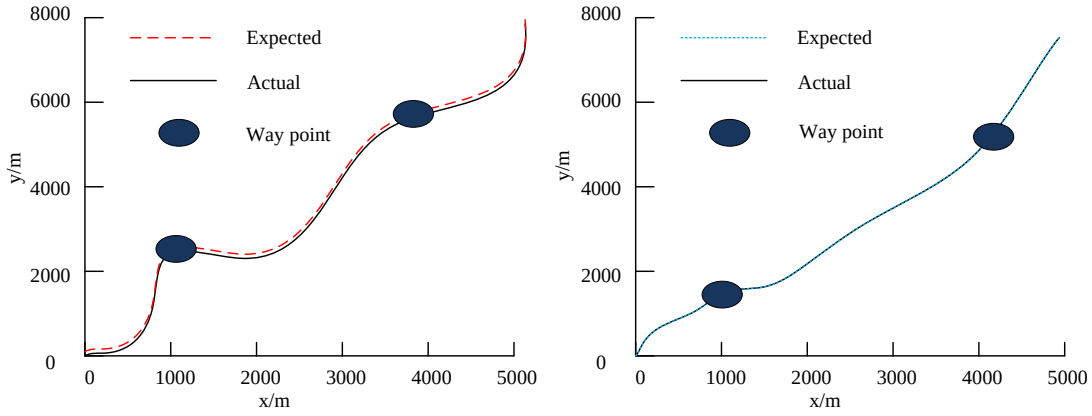


Figure 7: Tracking trajectories produced by different algorithms

From Figure 7, it was evident that under the traditional LOS algorithm, the actual trajectory of the ship closely approximated the expected trajectory but with some distance. In contrast, using the ILOS algorithm, the actual and expected trajectories overlapped, with an increasing degree of overlap as the distance increased. Moreover, in the same path, the ship guided by the ILOS algorithm had a clearer heading, saving more time and distance compared to the LOS algorithm-guided ship. Therefore, compared to the traditional LOS algorithm, the improved ILOS algorithm resulted in a ship's actual trajectory closer to the expected trajectory, indicating its effectiveness in ship trajectory tracking tasks. In the navigation of the ship, the heading angles generated by the LOS algorithm and ILOS algorithm are illustrated in Figure 8.

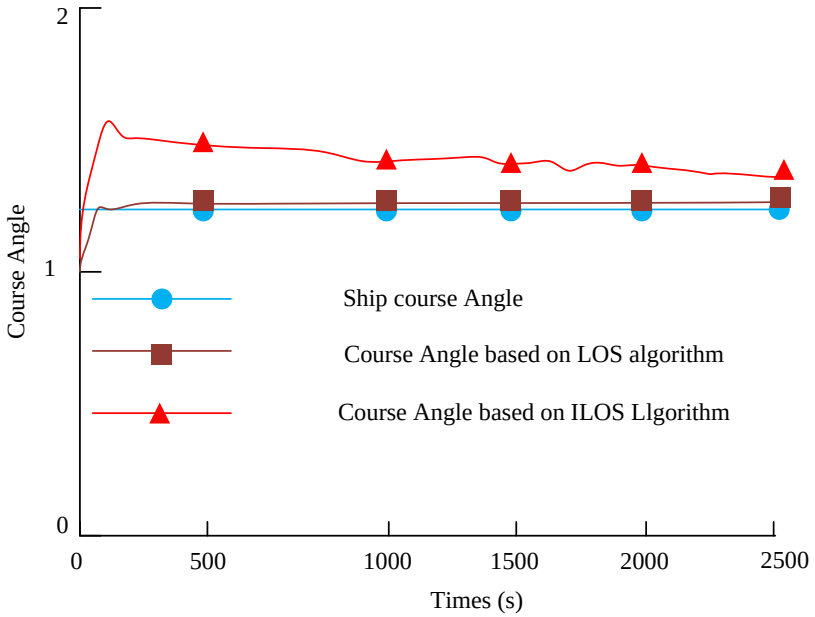


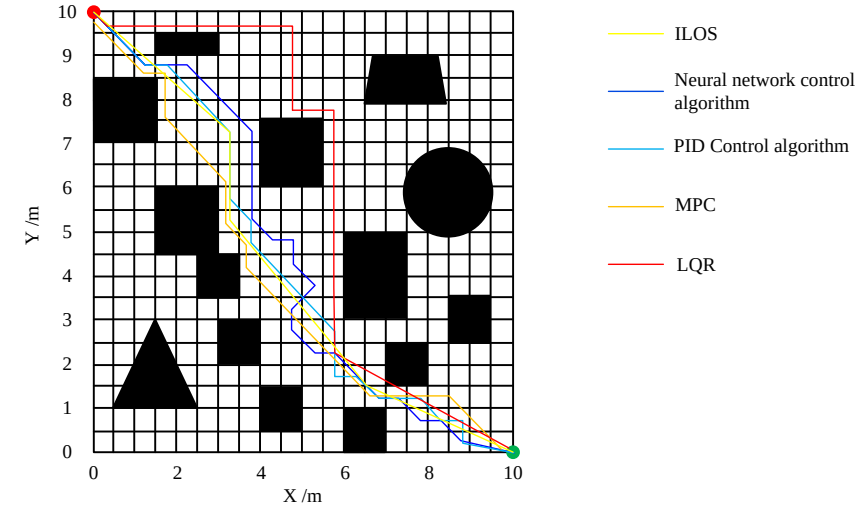
Figure 8: Different algorithms for the presence of heading angles in ship navigation

Figure 8 shows the heading angles of different algorithms during ship navigation. According to the figure, the improved ILOS algorithm exhibited a more pronounced heading angle. The heading angle played a role in controlling the ship's direction, adjusting the guidance trajectory, and tracking the target direction. Under the same adaptive rules and the influence of nonlinear controllers, the ILOS algorithm, due to its variable integral gain effect, could guide the ship's direction before achieving stability, steering it towards the desired trajectory. In contrast, the LOS algorithm maintained the ship's original heading and lacked the ability to guide the ship's heading during the response phase. The heading angle generated by the ILOS algorithm, which

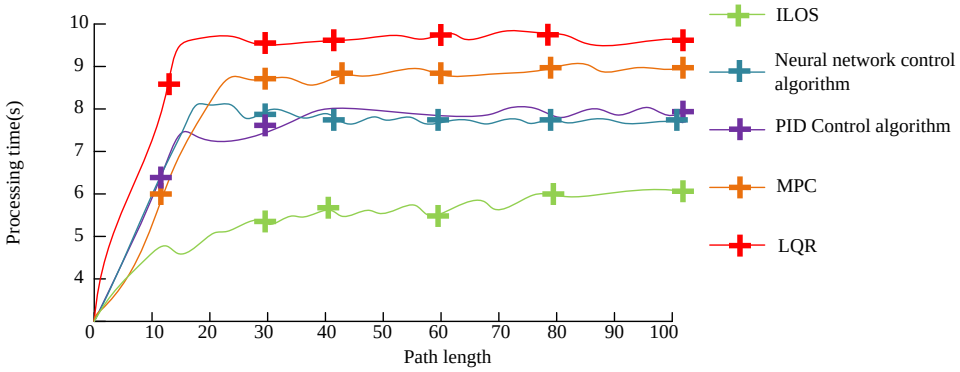
maintained a certain angle, can compensate for sideslip angles in perturbed environments. The angle generated between the desired and actual trajectories during ship navigation can produce a certain lateral thrust, which can eliminate the effects of disturbances. Therefore, the proposed ILOS guidance algorithm not only compensated for the effects of disturbances but also compensated for drift angles, making it more suitable for complex real-world environments.

4.2 Simulation analysis of guidance algorithm based on fermat curve

To validate the algorithm's performance in complex paths, comparative experiments were conducted simultaneously using other linear trajectory guidance algorithms. The results are shown in Figure 9.



(a)The performance of different algorithms in obstacle paths

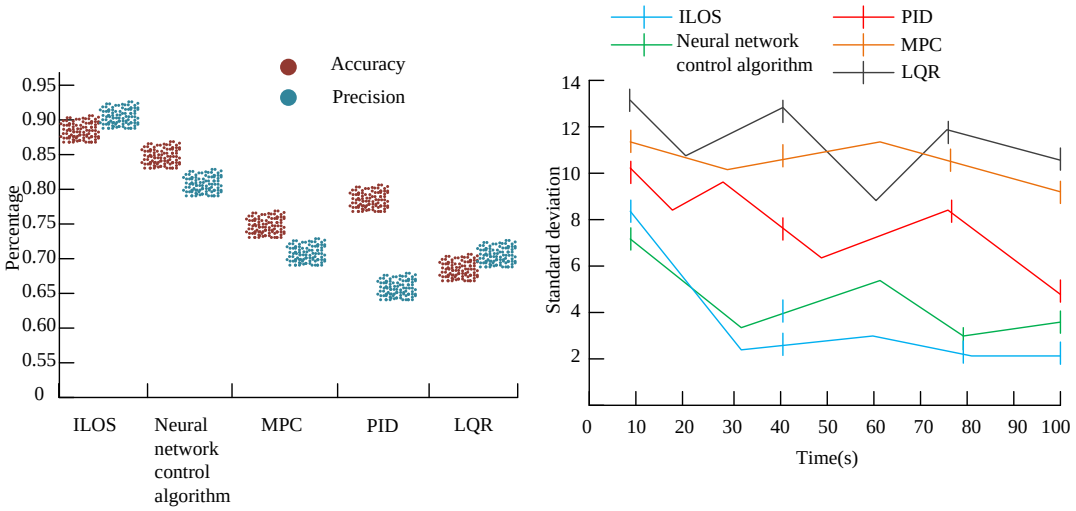


(b)The processing time of different algorithms

Figure 9: Path-Planning and runtime results for different algorithms

Figure 9 presents the results of path-planning and execution time for different algorithms. The Proportional-Integral-Derivative Control Algorithm (PID) used three controllers-proportional, integral, and derivative-to adjust the system's output to achieve the desired state response. The Linear Quadratic Regulator (LQR) algorithm linearized the trajectory using a state-space model and designed an optimal regulator to minimize the system's performance index. The Model Predictive Control (MPC) algorithm predicted the trajectory for a future time period using a predictive model and adjusted the system's control actions by minimizing the performance index. The Neural Network Control Algorithm utilized neural networks to learn and approximate unknown trajectory models, enabling control and guidance of the trajectory. Figure 9(a) illustrates the path-planning results for different algorithms. It was evident from the figure that the proposed ILOS algorithm had the shortest path, while the LQR algorithm had the longest path. Figure 9(b) shows the execution time results for different algorithms, indicating an increase in execution time with the length of the path. Overall, the ILOS algorithm had the shortest execution time, while the LQR algorithm had the longest. When the path length

was 100, the ILOS algorithm's execution time was around 5.9 seconds, less than the LQR algorithm's execution time of approximately 4 seconds. Shorter execution times ensured that the system could guide straight trajectories quickly and accurately, reducing delays and enhancing real-time performance. The detection results of various algorithms under the same conditions are shown in Figure 10.



(a)Accuracy and precision results of various algorithms

(b)Standard deviation of multiple algorithmic trajectories

Figure 10: Accuracy, precision, and trajectory standard deviation results for multiple algorithms

Figure 10 presents the accuracy, precision, and trajectory standard deviation results for various algorithms. From Figure 9(a), it could be observed that the ILOS algorithm exhibited higher accuracy and precision, reaching 87.1% and 91.3%, respectively, while the accuracy and precision of other algorithms were lower than those of the ILOS algorithm. The LQR algorithm, when facing environmental changes, experienced compromised control effectiveness, resulting in lower accuracy and precision at 68.4% and 70.8%, respectively. In Figure 9(b), it is noticeable that the trajectory standard deviation varied for different algorithms as runtime increased. Both LQR and MPC algorithms showed higher trajectory standard deviation, indicating larger trajectory deviations during ship navigation. On the other hand, the neural network control algorithm and ILOS algorithm displayed lower trajectory standard deviation, signifying smaller trajectory deviations during ship operation. The ILOS algorithm demonstrated increased stability, as the trajectory standard deviation decreased from around 8 to stabilize near 2 between 10s and 100s. The guidance error results for various algorithms under the same conditions are depicted in Figure 11.

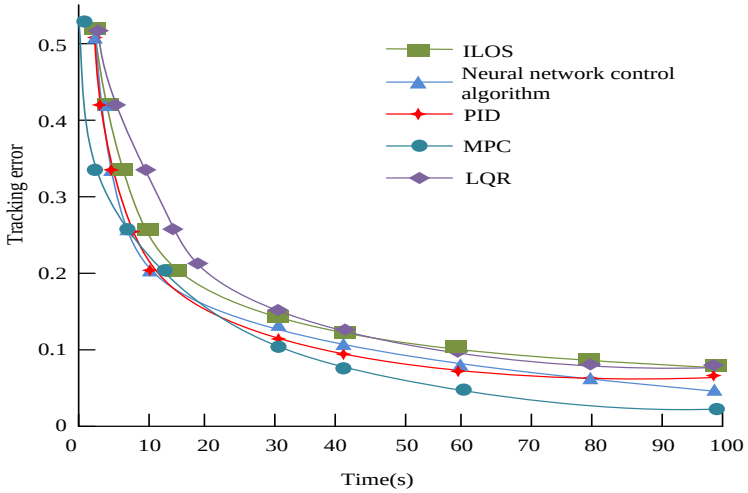


Figure 11: Bootstrap error results for multiple algorithms

It can be observed from the figure that the guidance error decreased with increasing runtime for multiple algorithms. At 100s, the ILOS algorithm reduced its guidance error to around 0.03, while the guidance error for the other four algorithms remained around 0.08. This indicated that the ILOS algorithm exhibited more accurate guidance performance, with a smaller deviation between the fitted actual trajectory and the expected trajectory. Algorithms with larger guidance errors resulted in more significant deviations between the fitted actual trajectory and the expected trajectory, consequently impacting algorithm precision. Furthermore, larger guidance errors increased the sensitivity of the algorithm to noise, potentially leading to oscillations or instability in the fitted trajectory.

In order to further prove that the ILOS guidance algorithms based on Fermat curves have better adaptability, three sea conditions were selected for this study to expand the analysis, and the three sea conditions were noted as calm, ripples, and storms, respectively, and the performance of the various algorithms under different sea conditions and different traffic conditions was obtained as shown in Table 3.

Table 3: Path-planning and navigation performance of the three algorithms in different sea conditions

Environmental conditions	Algorithm type	Path-planning length	Time-consuming	Navigation accuracy
Calm	Fermat-ILOS	5.8 kilometers	45s	96.8%
	MPC	6.1 kilometers	52s	92.3%
	LOS	6.5 kilometers	58s	90.1%
Ripple	Fermat-ILOS	5.9 kilometers	48s	96.2%
	MPC	6.3 kilometers	56s	91.5%
	LOS	7.1 kilometers	65s	89.7%
Storm	Fermat-ILOS	6.7 kilometers	51s	93.8%
	MPC	9.5 kilometers	63s	86.5%
	LOS	12.9 kilometers	78s	82.0%

As can be seen from Table 3, when the sea state condition is calm, the path-planning length, time consumption and navigation accuracy of the Fermat-ILOS algorithm were 5.8 kilometers, 45s and 96.8%, respectively. Under this condition, the navigation path of the Fermat-ILOS algorithm was the shortest, with the least amount of time consumption. The navigation accuracy was also the highest. When the sea state was rippled, the performance of Fermat-ILOS algorithm was still the best, and its path-planning length, time consumed, and navigation accuracy were 5.9 kilometers, 48s, and 96.2%, respectively. Compared with the calm sea state, the performance of Fermat-ILOS algorithm was weakened in the rippled sea state, but it still outperformed the other two comparative algorithms in general. When the sea state condition was stormy, the navigation paths and elapsed time of the three algorithms, Fermat-ILOS, MPC, and LOS, had a substantial increase, and the navigation accuracy was reduced. The path-planning length, elapsed time, and navigation accuracy of Fermat-ILOS in stormy sea states were 6.7 kilometers, 51s, and 93.8%, respectively.

5 Discussion

In order to improve the ship trajectory tracking accuracy and enhance the navigation efficiency of the ship motion system, this research proposed a path-planning method based on Fermat curves, combined the Fermat curve path-planning with the ship ILOS guidance system, and finally designed the Fermat-ILOS method. Compared with existing path navigation methods, Fermat-ILOS demonstrated significant advantages. For example, compared to the potential field function and reflection reward-based approach proposed by Meng et al. (2022), although both aimed to optimize the path for efficient exploration and energy minimization, Fermat-ILOS achieved smoother adjustment of the heading angle and optimization of path continuity by exploiting the geometrical properties of the Fermat curve. This not only reduced the path length and execution time but also improved the adaptability and robustness of the ship in complex navigational environments. In addition, compared to the study of Rath et al. (2021), although Rath et al. designed a hybrid controller using genetic algorithm and neural network and showed good navigation consistency in specific cases, the flexibility and accuracy of the method still needed to be improved when facing dynamic changes in the marine environment. The Fermat-ILOS method, on the other hand, optimized the Fermat curve parameters more efficiently and effectively by dynamically optimizing the Fermat curve parameters. It was more effective in dealing with this challenge. Through experimental validation, the Fermat-ILOS algorithm demonstrated excellent performance in different sea conditions, and the navigation accuracy of the algorithm was as high as 96.8%, 96.2%, and 93.8% in calm, rippled, and stormy sea conditions, respectively. The main reasons why Fermat-ILOS achieved better path-planning and navigation performance were the following two points.

Firstly, the Fermat curve effectively balanced the heading angle adjustment and path length, optimized the selection of steering points, and made the route more in line with the actual navigation needs. Secondly, the design of the Fermat-ILOS algorithm took into account the dynamic characteristics of the ship and the real-time changes in the marine environment, which improved the adaptability and flexibility of the path-planning. Overall, the introduction of the Fermat-ILOS algorithm in the field of ship trajectory planning represented a brand-new idea and method. Compared with traditional algorithms, it was not only innovative in theory but also provided effective solutions to the complex problems encountered in actual navigation. The novelty of the Fermat-ILOS algorithm was reflected in its ability to provide ships with smoother, more economical, and safer routes. Especially in the context of the frequent global maritime transportation and the increasing complexity of the use of the sea area at that time, this optimized route planning was essential to improve navigation efficiency, reduce energy consumption, and enhance maritime safety. In the future, with the development of autonomous ship technology, the application of the Fermat-ILOS algorithm in automatic navigation systems would further expand its practical influence, contributing to the progress of autonomous ship navigation technology and the development of marine engineering.

6 Conclusion

A path-planning method based on Fermat curves is proposed to address the issues arising from constraints in ship route planning. The method integrates ship trajectory tracking and utilizes Fermat curves to achieve navigation between straight-line ship trajectories. Subsequently, the Fermat curve path-planning method is incorporated into the ship ILOS (Integrated Line of Sight) guidance system to facilitate ship navigation. The study analyzes the guidance algorithm that combines Fermat curve path-planning. Experimental results indicated that the proposed algorithm achieved a high similarity between the expected and actual trajectories. Comparisons between the traditional LOS algorithm and the proposed algorithm revealed that the ship's heading was more precise with the proposed algorithm, saving both time and distance. The ILOS algorithm, with its variable integral gain effect, could guide the ship's direction even before stable ship navigation was achieved, steering it towards the expected trajectory. Furthermore, the proposed algorithm was compared with other guidance algorithms under similar path complexity conditions. The results demonstrated that the proposed algorithm yielded the shortest path and the fastest runtime of 5.9 seconds, outperforming the LQR algorithm with a runtime of 4 seconds. This suggested that the proposed algorithm can guide ship trajectories quickly and accurately. Additionally, the proposed algorithm achieved high accuracy and precision rates of 87.1% and 91.3%, respectively, surpassing other algorithms and enabling high-precision path-planning. In trajectory standard deviation comparison experiments, the proposed algorithm demonstrated minimal trajectory deviation during ship operation, maintaining a trajectory standard deviation of around 2. In guidance error comparison, the proposed algorithm converged quickly with lower error values than other algorithms, indicating accurate guidance performance and minimal trajectory deviation.

7 Future work

Although the Fermat curve-based path-planning method proposed in this research has been experimentally demonstrated to have the potential to improve the accuracy and efficiency of ship tracking, a series of shortcomings still exist. With the development of artificial intelligence and machine learning technologies, autonomous ships will be able to achieve more complex navigational tasks, and efficient and accurate path-planning algorithms are key to achieving this goal. Future research will further explore how to combine the Fermat curve path-planning method with advanced decision support systems to further improve the navigation capabilities of autonomous ships in complex marine environments. In addition, future research will also consider the scalability of this method for different types of vessels, such as cargo ships, fishing vessels, yachts, etc., and different navigation scenarios, such as narrow waterways, high-traffic-density areas, and complex marine environments. Different types of vessels and navigation scenarios may face different challenges, such as different hull characteristics, speed limitations, and the need for accurate navigation. In future work, it is planned to evaluate the applicability and effectiveness of the present method by analyzing actual track tracking data in different scenarios, and by combining other intelligent algorithms to accommodate these diverse needs. Ultimately, although the algorithm is able to achieve the path-planning task better, the actual environment may be more complex, so subsequent research can also optimize the navigation parameters of the ship model in a more detailed way to better apply to the ship navigation problem.

References

[1] L. Huo, “The global path planning for vehicular communication using ant colony algorithm in emerging wireless cloud computing,” Wireless Networks, vol. 29, no. 2, pp. 833-842, 2023. https://doi.org/10.1007/s11276-022-03152-0

[2] W. Yu, H. Sun, T. Feng, Y. Lv, X. Guo, and G. Xin, “A novel reliable path planning approach for multimodal networks based on a two-factor bound convergence algorithm,” Modern Physics Letters B, vol. 36, no. 23, pp. 2250007-2250042, 2022. https://doi.org/10.1007/s11227-022-04998-z

[3] A. Zou, L. Wang, W. Li, J. Cai, H. Wang, and T. Tan,

"Mobile robot path planning using improved mayfly optimization algorithm and dynamic window approach," The Journal of Supercomputing, vol. 79, no. 8, pp. 8340-8367, 2023. https://doi.org/10.1007/s11227-022-04998-z

[4] X. Wang, Z. Lyu, Z. Wei, L. Wang, Y. Lu, and L. Shi, "Multi-objective path planning algorithm for mobile charger in wireless rechargeable sensor networks," Wireless Networks, vol. 29, no. 1, pp. 267-283, 2023. https://doi.org/10.1007/s11276-022-03126-2

[5] A. Williams, "Human-centric functional computing as an approach to human-like computation," Artificial Intelligence and Applications, vol. 1, no. 2, pp. 118-137, 2023. https://doi.org/10.47852/bonviewAIA2202331

[6] Y. Li, A. H. Aghvami, and D. Dong, "Path planning for cellular-connected UAV: A DRL solution with quantum-inspired experience replay," IEEE Transactions on Wireless Communications, vol. 21, no. 10, pp. 7897-7912, 2022. https://doi.org/10.1109/TWC.2022.3162749

[7] H. Meng, and H. Zhang, "Mobile robot path planning method based on deep reinforcement learning algorithm," Journal of Circuits, Systems and Computers, vol. 31, no. 15, pp. 85-103, 2022. https://doi.org/10.1142/S0218126622502589

[8] A. K. Rath, D. R. Parhi, H. C. Das, P. B. Kumar, and M. K. Mahto, "Design of a hybrid controller using genetic algorithm and neural network for path planning of a humanoid robot," International Journal of Intelligent Unmanned Systems, 2021, vol. 9, no. 3, pp. 169-177, 2021. https://doi.org/10.1108/IJIUS-10-2019-0059

[9] E. García, J. R. Villar, Q. Tan, J. Sedano, C. Chira, "An efficient multi-robot path planning solution using A* and coevolutionary algorithms," Integrated Computer-Aided Engineering, 2023, vol. 30, no. 1, pp. 41-52, 2023. https://doi.org/10.3233/ICA-220695

[10] Y. Li, Y. Huang, L. Ge, and X. Li, "Mobile robot path planning based on angle-guided ant colony algorithm," International Journal of Swarm Intelligence Research (IJSIR), vol. 13, no. 1, pp. 1-19, 2022. https://doi.org/10.4018/IJSIR.302603

[11] S. Liu, and Y. Sang, "Underactuated stratospheric airship trajectory control using an adaptive integral backstepping approach," Journal of Aircraft, vol. 55, no. 6, pp. 1-15, 2018. https://doi.org/10.2514/1.C034923

[12] S. Telen, M. V. Barel, and J. Verschelde, "A robust numerical path tracking algorithm for polynomial homotopy continuation," SIAM Journal on Scientific Computing, vol. 6, no. 42, pp. 3610-3637, 2020. https://doi.org/10.1137/19M1288036

[13] Y. Bai, J. Lv, C. Wang, and Y. Geng, "Ship tracking method for resisting similar shape information under satellite videos," Journal of Applied Remote Sensing, vol. 16, no. 2, pp. 26517-026534, 2022. https://doi.org/10.1117/1.JRS.16.026517

[14] J. Wu, C. Cao, Y. Zhou, X. Zeng, Z. Feng, Q. Wu, and Z. Huang, "Multiple ship tracking in remote sensing images using deep learning," Remote Sensing, vol. 13, no. 18, pp. 3601-3601, 2021. https://doi.org/10.3390/rs13183601

[15] M. P. Strub, J. D. Gammell, "Adaptively informed trees (AIT*) and effort informed trees (EIT*): Asymmetric bidirectional sampling-based path planning," The International Journal of Robotics Research, vol. 41, no. 4, pp. 390-417, 2022. https://doi.org/10.1177/02783649211069572

[16] X. Wang, X. Song, "Optimal path planning of logistics distribution of urban and rural agricultural products from the perspective of supply chain," Informatica, vol. 47, no. 5, 2023. https://doi.org/10.31449/inf.v47i5.4557

[17] K. Parmar, and D. Guzzetti, "Interactive imitation learning for spacecraft path-planning in binary asteroid systems," Advances in Space Research, vol. 68, no. 4, pp. 1928-1951, 2021. https://doi.org/10.1016/j.asr.2021.04.023

[18] A. Liu, and J. Jiang, "Solving path planning problem based on logistic beetle algorithm search–pigeon-inspired optimisation algorithm," Electronics Letters, vol. 56, no. 21, pp. 1105-1108, 2020. https://doi.org/10.1049/el.2020.1895

[19] X. Meng, "Research on the Development of Modern Design Through Data Mining Technology," Informatica, vol. 48, no. 6, 2024. https://doi.org/10.31449/inf.v48i6.5241

[20] B. D. Wang, J. Song, R. Li, R. Han G. Zheng, and H. Yang, "A novel particle tracking velocimetry method for complex granular flow field," Chinese Physics B, vol. 29, no. 1, pp. 14207-14207, 2020. https://doi.org/10.1088/1674-1056/ab5936