



# Path Planning in Localization Uncertaining Environment Based on Dijkstra Method

Can Wang, Chensheng Cheng, Dianyu Yang, Guang Pan and Feihu Zhang\*

School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, China

Path planning obtains the trajectory from one point to another with the robot's kinematics model and environment understanding. However, as the localization uncertainty through the odometry sensors is inevitably affected, the position of the moving path will deviate further and further compared to the original path, which leads to path drift in GPS denied environments. This article proposes a novel path planning algorithm based on Dijkstra to address such issues. By combining statistical characteristics of localization error caused by dead-reckoning, the replanned path with minimum cumulative error is generated with uniforming distribution in the searching space. The simulation verifies the effectiveness of the proposed algorithm. In a real scenario with measurement noise, the results of the proposed algorithm effectively reduce cumulative error compared to the results of the conventional planning algorithm.

**Keywords:** path planning, greedy search, cumulative error estimation, global planning, Dijkstra

## 1. INTRODUCTION

To obtain the optimal trajectory from one point to another, path planning needs to combine the robot's geometric and dynamic information (Bidot et al., 2013), environment map (Peng and Green, 2019), the initial state and target state (Choset et al., 2005), etc. According to task requirements, the optimal path seeks the shortest length and the best energy (Ibraheem and Hassan Ajeil, 2018). In specific tasks, path planning is commonly performed by combining sensor type and performance, carrier kinematics and dynamics characteristics, and task requirements.

Classical path planning methods consists of heuristic searching, sampling planning, and model-dependent methods (Yilmaz et al., 2008). In particular, when localizing through IMU (Tick et al., 2013), visual odometry (He et al., 2020), or other sensors (Paull et al., 2014), none of the mentioned methods considers the localization uncertainty issue. However, the GPS may be subject to some limitations in practical applications, especially in underwater scenarios (Li et al., 2019). In applications with GPS-denied, it is not feasible to combine the robot's motion attributes with inaccurate odometry sensors, which will cause localization errors in long-term missions. It is generally accepted that positioning errors do not affect the planning task since planning is first performed and then control decisions are made. In robot tasks where errors exist, however, it is also possible to impact localization errors by changing the path planning strategy.

To address the cumulative error of navigation, many studies first perform accurate statistical analysis on it. Miller et al. (2010) proposed an error state formula for the navigation algorithm of an underwater vehicle. The kinematics model of the system is augmented with unknown parameters from the sensor model, and the difference between the estimation of the real augmented system equation is expressed as the error state system. And a Kalman filter is designed to estimate this error state by the measurement residuals of the auxiliary sensor.

## OPEN ACCESS

### Edited by:

Chen Qiao,  
Xi'an Jiaotong University, China

### Reviewed by:

He Chen,  
Hebei University of Technology, China  
Xiao Liang,  
Nankai University, China

### \*Correspondence:

Feihu Zhang  
feihu.zhang@nwpu.edu.cn

**Received:** 25 November 2021

**Accepted:** 24 January 2022

**Published:** 11 March 2022

### Citation:

Wang C, Cheng C, Yang D, Pan G  
and Zhang F (2022) Path Planning in  
Localization Uncertaining Environment  
Based on Dijkstra Method.  
*Front. Neurobot.* 16:821991.  
doi: 10.3389/fnbot.2022.821991

Yin et al. (2013) established a strap-down inertial navigation system error model based on various error sources of inertial components. By using Particle Swarm Optimization (PSO) to optimize the parameters of SVM, the positioning error prediction method of a navigation system is realized. By redesigning the system parameters and using data recalculation algorithms, Xu et al. (2014) proposed an improved alignment method for Strapdown Inertial Navigation System (SINS) based on Doppler Velocity Log (DVL). Dai et al. (2016) proposes a particle swarm algorithm to identify the error parameters of the Delta parallel robot, and the geometric parameter errors can be identified by a simple iterative process. Mansouri et al. (2020) settles positioning uncertainty by defining adaptive weights for tracking position and speed reference points, and calculating based on the uncertainty associated with measurement. Accurate error estimation methods facilitate the correction and compensation of navigation positioning. Nevertheless, few studies have effectively integrated error estimation with the navigation planning process, which is uncharted territory.

To effectively reduce the navigation error, generally intermittent global position correction methods based on GPS, SLAM, acoustic positioning, (Thomson et al., 2017; Chew and Zakaria, 2019; Marchel et al., 2020), etc. There are also studies on error compensation based on artificial intelligence methods (Brossard et al., 2020) or combined with the kinematics of the robot (Batista et al., 2013). However, in the planning stage, the navigation error cannot be effectively reduced without a determined path. Therefore, the existing research generally solves such problems through fault-tolerant planning. Carlson et al. (2013) proposed and compared three different strategies for estimating the change of the robot's motion, which effectively reduced the probability of collisions and avoided sources of error in industrial scenarios. Eaton et al. (2017) proposed a robust *Partially Observable Markov Decision Process (POMDP)* formula, which provides the capability of planning and tracking with limited observations. Lv et al. (2019) cited the dense connection method to improve the Q-networks structure to solve the issue of robot drift by adopting the framework of a dense network. Sainte Catherine and Lucet (2020) combined with the improved *Hybrid Reciprocating Speed Obstacle (HRVO)* method of tracking error estimation, and adapting the speed obstacle paradigm to agents with dynamic constraints and unreliable velocity estimation. Yilmaz et al. (2021) uses the fuzzy logic network to model dynamic uncertainty, and proposes a new definition of the error-like vector containing the pseudo-inverse of the Jacobian matrix. The current method only considers the fault-tolerance of path planning and does not apply the mechanism of the cumulative error to avoid tracking drift, i.e., does not consider the impact of the motion after planning.

By considering the perceptual uncertainty, some planning methods consider the generation and elimination strategies of planned path errors, and thus, new planning methods are designed. Pilania and Gupta (2017) designs sensor measurements that depend only on the samples, achieving higher uncertainty reduction by placing more samples in regions with higher uncertainty reduction while maintaining enough samples in

regions with poor uncertainty reduction. It also uses uncertainty measures (instead of distance) to connect new samples to neighboring nodes, achieving an efficient and high-quality planning capability. Park et al. (2018) achieved collision avoidance path planning by considering the uncertainty of the time-varying trajectories of linearly increasing Autonomous Ground Vehicles (AGVs) and obstacles, modeling the error covariance using a tracking filter designed to estimate motion information, and employing a probabilistic approach to calculate the collision risk combined with the dynamic characteristics of AGVs. Papachristos et al. (2019) designed a paradigm that follows a hierarchical optimization objective and executes it in a backward horizon manner to implement an uncertainty-aware path planning strategy. Combining adaptive error sampling for generating possible path candidates with a utility-based approach, Lee et al. (2020) implements a path planning task for safe parking under perceptible uncertainty, which takes into account detection errors and makes optimal decisions under uncertainty. Uncertainty generation is mainly obtained through passive sensors, and unfortunately, the current capability to rely on inertial navigation alone for path planning under uncertainty needs to be further explored.

However, in practical applications, system errors and deviations are inevitable with sensor registration problems. Failure to use the control strategy to optimize the planning and motion process, a disastrous deviation will occur in the tracking process. Our previous study (Wang et al., 2021) applies reinforcement learning to address this issue and obtains a path with a relatively smaller cumulative error by generating a probability sampling. As the limitation of sampling, the global optimal solution cannot be obtained.

This article combines a qualitative and quantitative analysis of the ranging error and traversal advantage of the greedy search algorithm in the path planning process. To minimize the accumulated errors in navigation, we obtain an ideal path that can achieve high accuracy tracking. The key innovation is the theoretical modeling from the systematic perspective of error estimation and planning based on greedy search in a practical scene. In scenarios where measurement errors exist, the proposed algorithm is effective in reducing the path error concerning the underlying Dijkstra method. To the best of the author's knowledge, this is the first study that considers the cumulative error of tracking in the pre-planning process and performs global corrections to form paths with minimal cumulative error.

The main contributions of this article are as follows:

- Through the statistical qualitative and quantitative analysis of the cumulative error by odometry positioning, the qualitative and quantitative expressions for path planning are summarized.
- Improve the map exploration method of Dijkstra to adapt to the qualitative expression of reducing cumulative error.
- By iterating and optimizing the cumulative errors of the paths, the results of their statistics and the global optimal path are obtained.

This article is organized as follows: The second part analyzes the mathematical representation and statistical characteristics of the cumulation error. The third part proposes a path planning framework based on the improved Dijkstra method and optimized cumulative error. In the fourth part, the simulation planning results are compared and analyzed, and the results are discussed. Finally, the fifth part concludes the full article and discusses possible directions for future study.

## 2. METHODOLOGICAL BACKGROUND

When the global positioning system is unavailable, the robot has to utilize the attitude sensor and inertial sensor (gyro and accelerometer) to perform dead-reckoning. Assuming odometry sensor measurement is only presented in polar coordinates, and the corresponding noises are distributed with *Independent Identically Distribution (IID)*, which is determined based on the comparative statistics of the measured value and the true value in Fallon et al. (2010). As the presence of noise, robot positioning by heading projection will produce a continuous accumulation of errors. Hence, the robot has to calibrate its positions regularly.

The main challenge of numerical analysis of errors is the drift caused by relative noise measurements, i.e., the cumulative error increases nonlinearly with distance or time. This article uses statistical properties to study the growth rate of cumulative error in our previous article (Zhang et al., 2013). In this article, the robot is viewed as a mass, i.e., there are no kinematic constraints. This means that localization information can only be derived from inertial navigation measurements and cannot be corrected for localization based on kinematic models. Still, the proposed method applies to all types of robots, since it only considers planning paths and does not involve path tracking strategies.

The robot position is estimated based on angle and distance in polar coordinates, as shown in **Figure 1**. Define the corresponding metric:

$$\theta_n^m = \bar{\theta}_n + \tilde{\theta}_n; d_n^m = \bar{d}_n + \tilde{d}_n \quad (1)$$

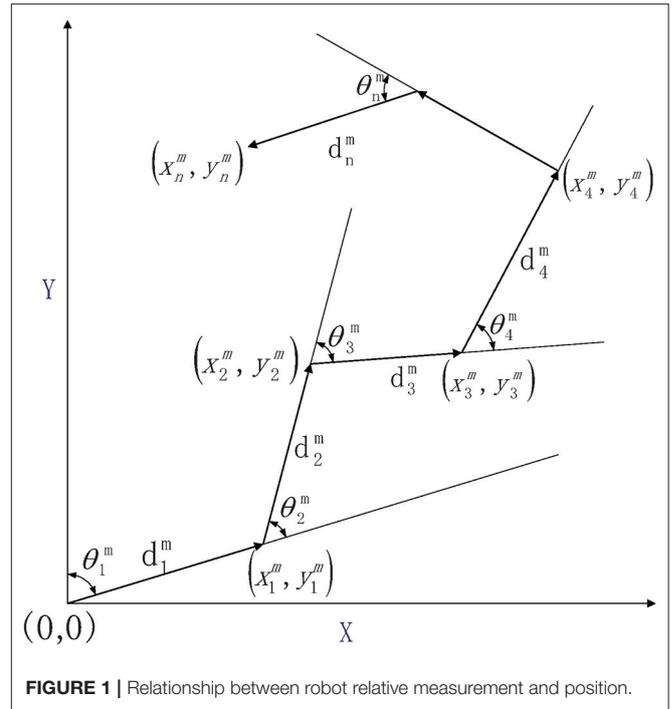
where  $n$  is the time index,  $d$  and  $\theta$  represents relative distance and direction between consecutive frames. The pose measurement  $(\theta_n^m, d_n^m)$  is then consisted of ground truth  $(\bar{\theta}_n, \bar{d}_n)$  and error  $(\tilde{\theta}_n, \tilde{d}_n)$  with SD  $\delta_\theta$  and  $\delta_d$ .

The principle of dead-reckoning in the Cartesian coordinate system is as follows:

$$x_n^m = \sum_{i=1}^n \left( d_i^m \sin \sum_{j=1}^i \theta_j^m \right) \quad (2)$$

$$y_n^m = \sum_{i=1}^n \left( d_i^m \cos \sum_{j=1}^i \theta_j^m \right) \quad (3)$$

The accumulation of drift by noise measurement is unbounded. The lower bound can be estimated by Cramer2Rao bound (Arrichiello et al., 2012), but the upper bound cannot be estimated by traditional methods, especially when there are no basic facts. However, the error distribution properties of multiple statistics can be used for the statistical estimation of errors.



**FIGURE 1** | Relationship between robot relative measurement and position.

When the true value is known, the trajectory can also be expressed as:

$$\begin{aligned} x_n^m &= \bar{x}_n + \tilde{x}_n \\ &= \sum_{i=1}^n \left( d_i^m \sin \sum_{j=1}^i \theta_j^m \right) \\ &= \sum_{i=1}^n \left( (\bar{d}_i + \tilde{d}_i) \sin \sum_{j=1}^i (\bar{\theta}_j + \tilde{\theta}_j) \right) \\ &= \left( \sum_{i=1}^n \bar{d}_i + \sum_{i=1}^n \tilde{d}_i \right) \cdot \left[ \sin \sum_{j=1}^i \bar{\theta}_j \cos \sum_{j=1}^i \tilde{\theta}_j \right. \\ &\quad \left. + \cos \sum_{j=1}^i \bar{\theta}_j \sin \sum_{j=1}^i \tilde{\theta}_j \right] \end{aligned} \quad (4)$$

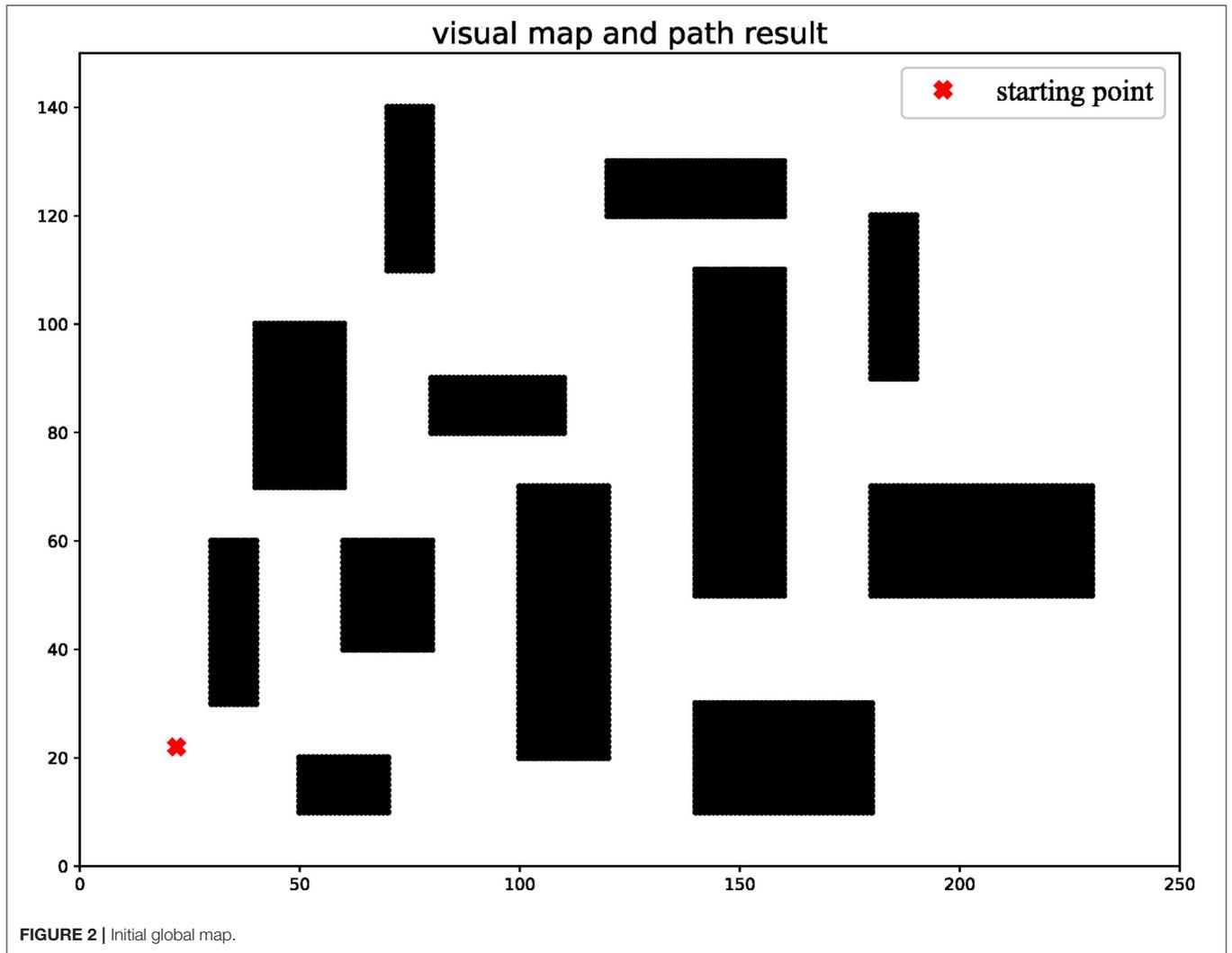
Then the mathematical expression of the cumulative error in the x-direction can be obtained:

$$\begin{aligned} \tilde{x}_n &= \sum_{i=1}^n \bar{d}_i \left[ \sin \sum_{j=1}^i \bar{\theta}_j \left( \cos \sum_{j=1}^i \tilde{\theta}_j - 1 \right) \right. \\ &\quad \left. + \cos \sum_{j=1}^i \bar{\theta}_j \sin \sum_{j=1}^i \tilde{\theta}_j \right] \\ &\quad + \sum_{i=1}^n \tilde{d}_i \left[ \sin \sum_{j=1}^i \bar{\theta}_j \cos \sum_{j=1}^i \tilde{\theta}_j \right. \\ &\quad \left. + \cos \sum_{j=1}^i \bar{\theta}_j \sin \sum_{j=1}^i \tilde{\theta}_j \right] \end{aligned} \quad (5)$$

In fact, the cumulative error depends to a large extent on basic facts. In addition, the expected and variance of the cumulative error are estimated based on statistical properties:

$$E \left[ \tilde{x} | \bar{\theta}, \bar{d} \right] = \sum_{i=1}^n \bar{d}_i \left[ \sin \sum_{j=1}^i \bar{\theta}_j \left( e^{-\frac{i\delta_\theta^2}{2}} - 1 \right) \right] \quad (6)$$

$$\begin{aligned} \text{var} \left( \tilde{x} | \bar{\theta}, \bar{d} \right) &= E \left[ \tilde{x} | \bar{\theta}, \bar{d} \right] - E^2 \left[ \tilde{x} | \bar{\theta}, \bar{d} \right] \\ &= A + B + C - E^2 \left[ \tilde{x} | \bar{\theta}, \bar{d} \right] \end{aligned} \quad (7)$$



where:

$$A = \sum_{i=1}^n \bar{d}_i^2 \left[ \sin^2 \sum_{j=1}^i \bar{\theta}_j \left( 0.5e^{-2i\delta_\theta^2} + 1.5 - 2e^{-\frac{i\delta_\theta^2}{2}} \right) + 0.5\cos^2 \sum_{j=1}^i \bar{\theta}_j \left( e^{-2i\delta_\theta^2} + 1 \right) \right] \quad (8)$$

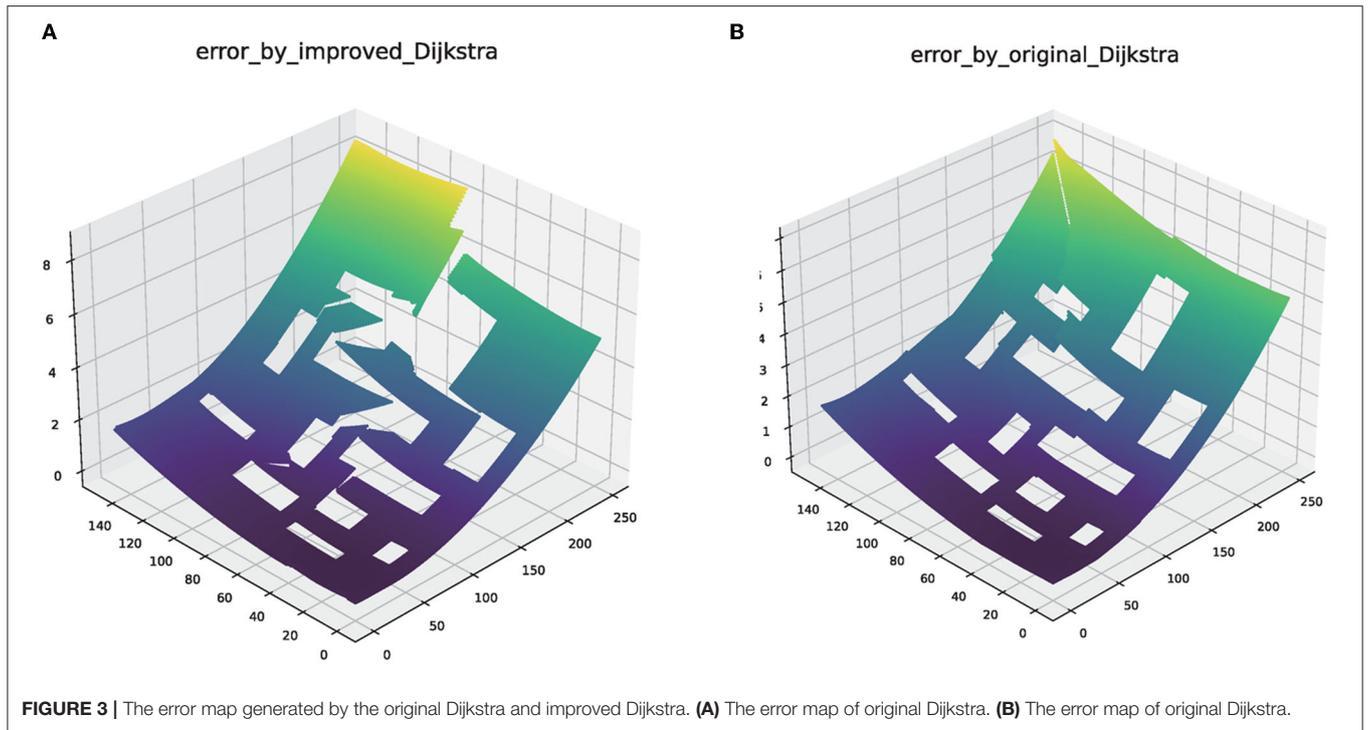
$$C = \sum_{i=1}^n \left[ 0.5\sin^2 \sum_{j=1}^i \bar{\theta}_j \left( e^{-2i\delta_\theta^2} + 1 \right) + 0.5\cos^2 \sum_{j=1}^i \bar{\theta}_j \left( 1 - e^{-2i\delta_\theta^2} \right) \right] \quad (10)$$

$$B = 2 \sum_{i=1}^{n-1} \sum_{p=1+i}^n \bar{d}_i \bar{d}_p \left\{ \begin{aligned} & \sin^2 \sum_{j=1}^i \bar{\theta}_j \cos \Delta\bar{\theta} \left[ \frac{1 + 0.5 \left( 1 + e^{-2i\delta_\theta^2} \right) e^{-0.5(p-i)\delta_\theta^2}}{e^{-0.5i\delta_\theta^2} - e^{-0.5i\delta_\theta^2} e^{-0.5(p-i)\delta_\theta^2}} \right] \\ & + \sin \sum_{j=1}^i \bar{\theta}_j \sin \Delta\bar{\theta} \cos \sum_{j=1}^i \bar{\theta}_j \\ & \left[ \frac{1 + 0.5 \left( 1 + e^{-2i\delta_\theta^2} \right) e^{-0.5(p-i)\delta_\theta^2} + 1}{-e^{-0.5i\delta_\theta^2} - e^{-0.5i\delta_\theta^2} e^{-0.5(p-i)\delta_\theta^2}} \right] \\ & \left[ \frac{-0.5 \left( 1 - e^{-2i\delta_\theta^2} \right) e^{-0.5(p-i)\delta_\theta^2}}{+ \cos^2 \sum_{j=1}^i \bar{\theta}_j \cos \Delta\bar{\theta} \cdot 0.5 \left( 1 - e^{-2i\delta_\theta^2} \right) e^{-0.5(p-i)\delta_\theta^2}} \right] \end{aligned} \right\} \quad (9)$$

The above formula is an explicit expression of expectation and variance of cumulative error. Since the global planning map is a priori, this article uses the true value to calculate expectation and variance. However, the ground truth is quite challenging to acquire in real scenarios. To effectively evaluate the error in a real scene, the expected values of the true moment are evaluated conditional on the noisy relative measurements:

$$E[\tilde{x}_n^m] = \sum_{i=1}^n \bar{d}_i^m \left( e^{-i\delta_\theta^2} - e^{-0.5i\delta_\theta^2} \right) \sin \sum_{j=1}^i \bar{\theta}_j^m \quad (11)$$

$$\text{var}(\tilde{x}_n^m) = A_1 + B_1 + C_1 - E^2[\tilde{x}_n^m] \quad (12)$$



where:

$$A_1 = \sum_{i=1}^n (d_i^m)^2 \left\{ \begin{array}{l} \left[ \begin{array}{l} 0.5e^{-2i\delta_\theta^2} + 1.5 \\ -2e^{-0.5i\delta_\theta^2} \\ 0.5(1 + e^{-2i\delta_\theta^2}) \sin^2 \sum_{j=1}^i \theta_j^m \\ + 0.5(1 - e^{-2i\delta_\theta^2}) \cos^2 \sum_{j=1}^i \theta_j^m \end{array} \right] \\ + 0.5(1 + e^{-2i\delta_\theta^2}) \\ \left[ \begin{array}{l} 0.5(1 + e^{-2i\delta_\theta^2}) \cos^2 \sum_{j=1}^i \theta_j^m \\ + 0.5(1 + e^{-2i\delta_\theta^2}) \sin^2 \sum_{j=1}^i \theta_j^m \end{array} \right] \end{array} \right\} \quad (13)$$

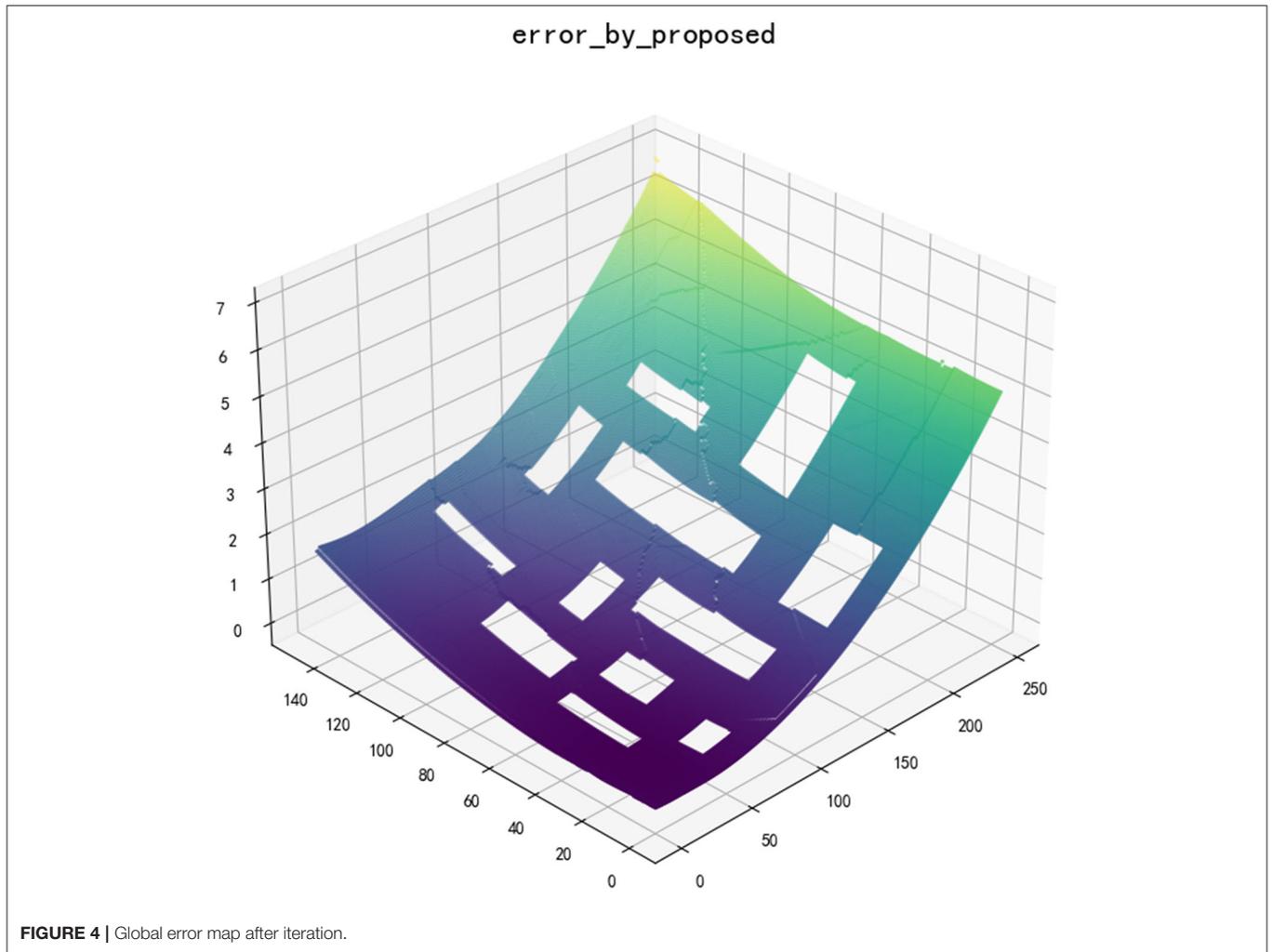
$$B_1 = 2 \sum_{i=1}^{n-1} \sum_{p=1+i}^n d_i^m d_p^m \left\{ \begin{array}{l} \left[ \begin{array}{l} 0.5(1 + e^{-2i\delta_\theta^2}) \cdot \sin^2 \sum_{j=1}^i \theta_j^m \\ + 0.5(1 - e^{-2i\delta_\theta^2}) \cdot \cos^2 \sum_{j=1}^i \theta_j^m \end{array} \right] \\ \cdot \left[ \cos \Delta\theta^m e^{-0.5(p-i)\delta_\theta^2} \right] [\dots] \\ + \left[ \sin \sum_{j=1}^i \theta_j^m \cdot \sin \Delta\theta^m \cdot \cos \sum_{j=1}^i \theta_j^m \right. \\ \left. \cdot e^{-2i\delta_\theta^2} e^{-0.5(p-i)\delta_\theta^2} \right] [\dots] \\ + \left[ \begin{array}{l} 0.5(1 + e^{-2i\delta_\theta^2}) \cdot \cos^2 \sum_{j=1}^i \theta_j^m \\ + 0.5(1 - e^{-2i\delta_\theta^2}) \cdot \sin^2 \sum_{j=1}^i \theta_j^m \end{array} \right] \\ \left[ \cos \Delta\theta^m e^{-0.5(p-i)\delta_\theta^2} \right] [\dots] \end{array} \right\} \quad (14)$$

$$C_1 = \sum_{i=1}^n \left\{ \begin{array}{l} 0.25(e^{-2i\delta_\theta^2} + 1) \\ \left[ \begin{array}{l} (e^{-2i\delta_\theta^2} + 1) \cdot \sin^2 \sum_{j=1}^i \theta_j^m \\ + (1 - e^{-2i\delta_\theta^2}) \cos^2 \sum_{j=1}^i \theta_j^m \end{array} \right] \\ + 0.25(1 - e^{-2i\delta_\theta^2}) \\ \left[ \begin{array}{l} (1 + e^{-2i\delta_\theta^2}) \cdot \cos^2 \sum_{j=1}^i \theta_j^m \\ + (1 - e^{-2i\delta_\theta^2}) \cdot \sin^2 \sum_{j=1}^i \theta_j^m \end{array} \right] \end{array} \right\} \quad (15)$$

More details could be found in Zhang and Knoll (2016) in the same manner, the complete cumulative errors are, therefore, calculated. In the next section, the Dijkstra-based global exploration method will first be used to traverse the map and determine the error-minimizing path for each location by evaluating the error of each path, thus achieving the task of reducing path drift.

### 3. PATH PLANNING METHOD BASED ON DIJKSTRA

To obtain a globally optimal path with the smallest error in the prior map, it is necessary to traverse the entire map and generate an error map. That is, similar to the “breadcrumbs map,” the error map has nothing to do with the endpoint but only with the starting point. Meanwhile, a greedy algorithm means that only the locally optimal solution is selected, but the part relative to the starting point is known, which is conducive to the optimization of the algorithm. Therefore, this algorithm can only choose the global traversal method, not the heuristic method. This article improved the Dijkstra algorithm based on its principle and the



qualitative results of error statistical calculations. At the same time, the quantitative calculation of path error is applied to iterate and update the error map, and finally, obtain the global error map. In the case of a given endpoint, the minimum error path can be quickly obtained through the global error map.

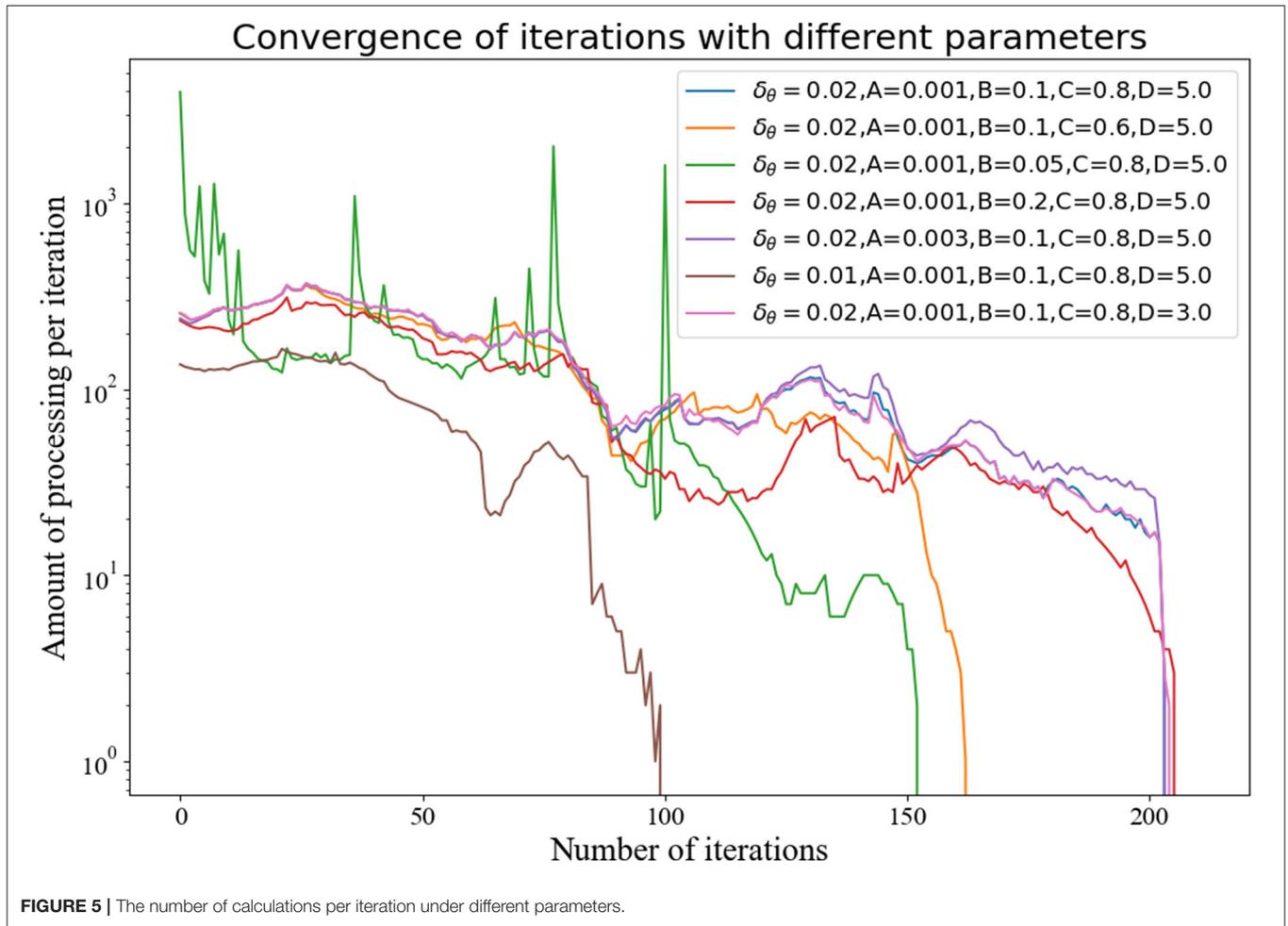
### 3.1. Improved Dijkstra

Breadth-First Search (BFS) (Broder et al., 2000) or Dijkstra (Kang et al., 2008) can be used to traverse the map, which is suitable for obtaining a global error map. However, the calculation of the cumulative error needs to be based on the entire path rather than part of the path segment, which does not apply to algorithms based on father-node exploration. Therefore, based on the results of Section II and previous study, the cumulative error has a great relationship with angle change of path measurement. That is, relative to starting point, the later the robot changes its angle, the smaller the cumulative error of path. The improved Dijkstra method will make the path generation of each point based on the latest turn path of starting point in the process of traversing the global map.

Since the Dijkstra algorithm is graph-based, we first initialize graph  $G$  and give a starting point. This algorithm is not to obtain the shortest path but to obtain a path that turns farther from starting point based on the nature of the cumulative error. The algorithm needs to initialize an empty set  $S$  to store those vertices that have been traversed and initialize a set  $Q$  which includes all vertices  $G.V$ .  $Q$  uses the data structure of the smallest priority queue, in which the key is the number of angle changes from starting point to vertex, expressed as  $turn\_num$ . Additionally, the vertex with the least number of angle changes is popped up each time.

In the rasterized map, the change of the robot's movement angle is discrete. This article chose  $\{(0, 1), (1, 1), (1, 0), (1, -1), (0, -1), (-1, -1), (-1, 0), (-1, 1)\}$  as optional movement directions  $G.Adj[u]$ . To reduce the cumulative error of each path, we limit the angle change of each vertex adjacent point, i.e., the angle of each movement  $|\theta| \leq \frac{\pi}{4}$ . In other words,  $G.Adj\_limited[u]$  has only 3 adjacent vertices.

For the weight of the edge, we first make the path go straight, and have to make a turn before turning. In the algorithm,  $\omega\_d(u, v)$  is the distance from  $u \rightarrow v$ , and



$\omega(u, v, u, \pi)$  is the number of turns of the edge  $u \rightarrow v$ . Since the traversal only considers the current node and adjacent nodes, the father node of the current node  $u, \pi$  is also required. The final algorithm will first traverse the nodes that have not turned, and then traverse the paths with fewer turns until the initial global error graph is generated. The complete pseudo-code of the improved Dijkstra algorithm is shown in **Algorithm 1**. The algorithm aims to facilitate subsequent point set updates and calculations by generating large error variances.

In the scenario where the sensor exists errors, the statistics of the cumulative error of the entire path are simple, but the error in the planning stage cannot be measured. Similar to the inability to obtain the best-first search strategy for the shortest path in concave obstacle environments, in addition, common planning methods are unable to move toward minimum error from the beginning. This is since larger errors may occur in the following trips, leading to larger overall deviations. This article evaluates the cumulative error based on the entire path from starting point to each point. Since the global map is a priori, the cumulative error of each path could be calculated through the true value of each measurement  $(\hat{\theta}, \hat{d})$  based on Equation (6).

### 3.2. Global Iteration Strategy

In the initial global error map, the path from starting point will pass through obstacles and intersect. That is, some points will be reached by the paths on both sides of the obstacle together, which results in different cumulative error values for this point. For the points where there are differences in cumulative error caused by different paths, this article initializes and updates the minimum priority queue  $Q_{dif}$  to determine the point set that needs to be iteratively calculated. In the iterative process, the error of the point set is recalculated and the path is updated to obtain a path with a smaller cumulative error for each point. The pseudo-code of strategy for updating queue is shown in **Algorithm 2**.

To accurately get each point that needs to be iterated, we need to update the key-value *value* of each point in the traversal map in advance. In this article, it is defined as:  $v.value = \max(G.Adj[u].error - v.error)/v.d$ , which is due to the smaller scale of the map and the absolute difference in cumulative error is not obvious. For maps with obvious error differences, we can judge whether the point needs iteration according to absolute error difference  $v.value\_absolute = \max(G.Adj[u].error - v.error)$ . To adapt to different scale maps, we simultaneously apply two benchmarks to update the queue  $Q_{dif}$ .

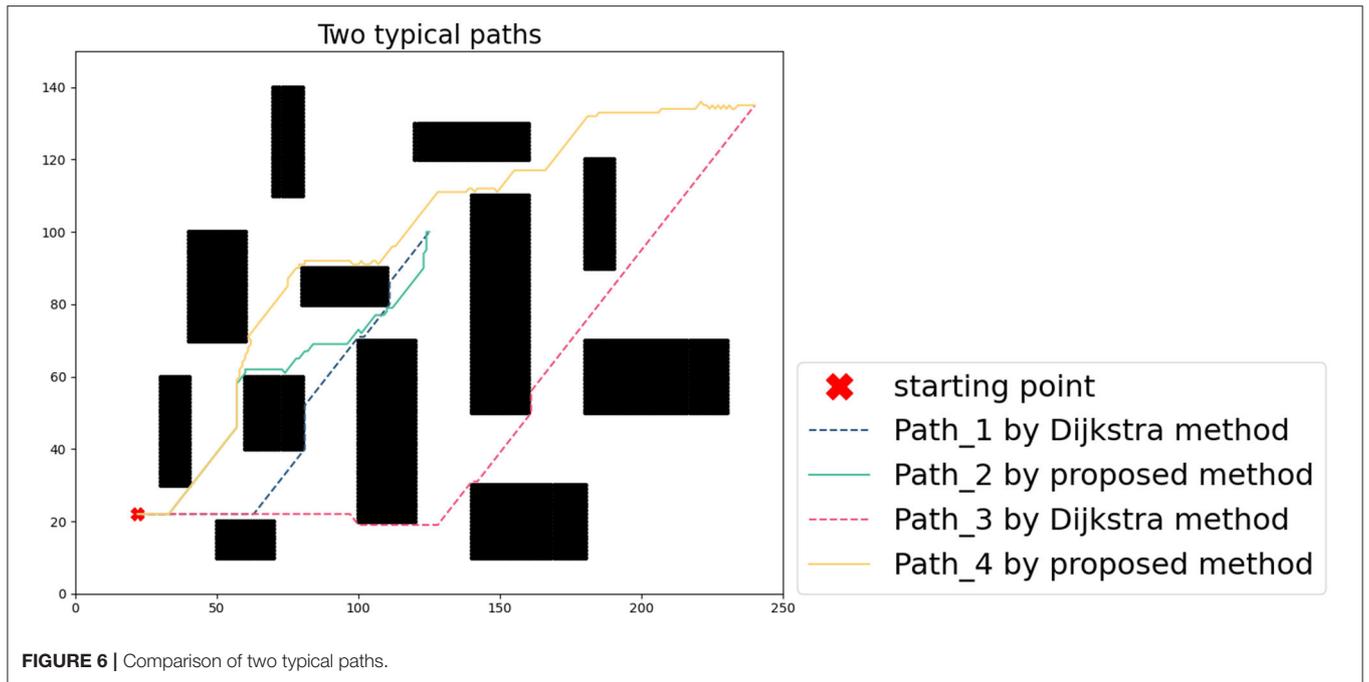


FIGURE 6 | Comparison of two typical paths.

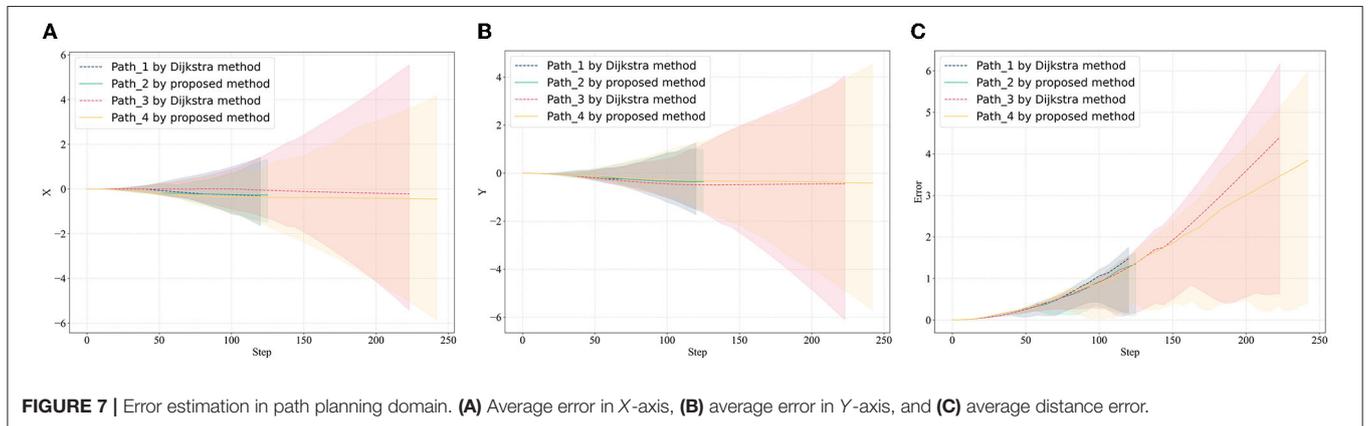
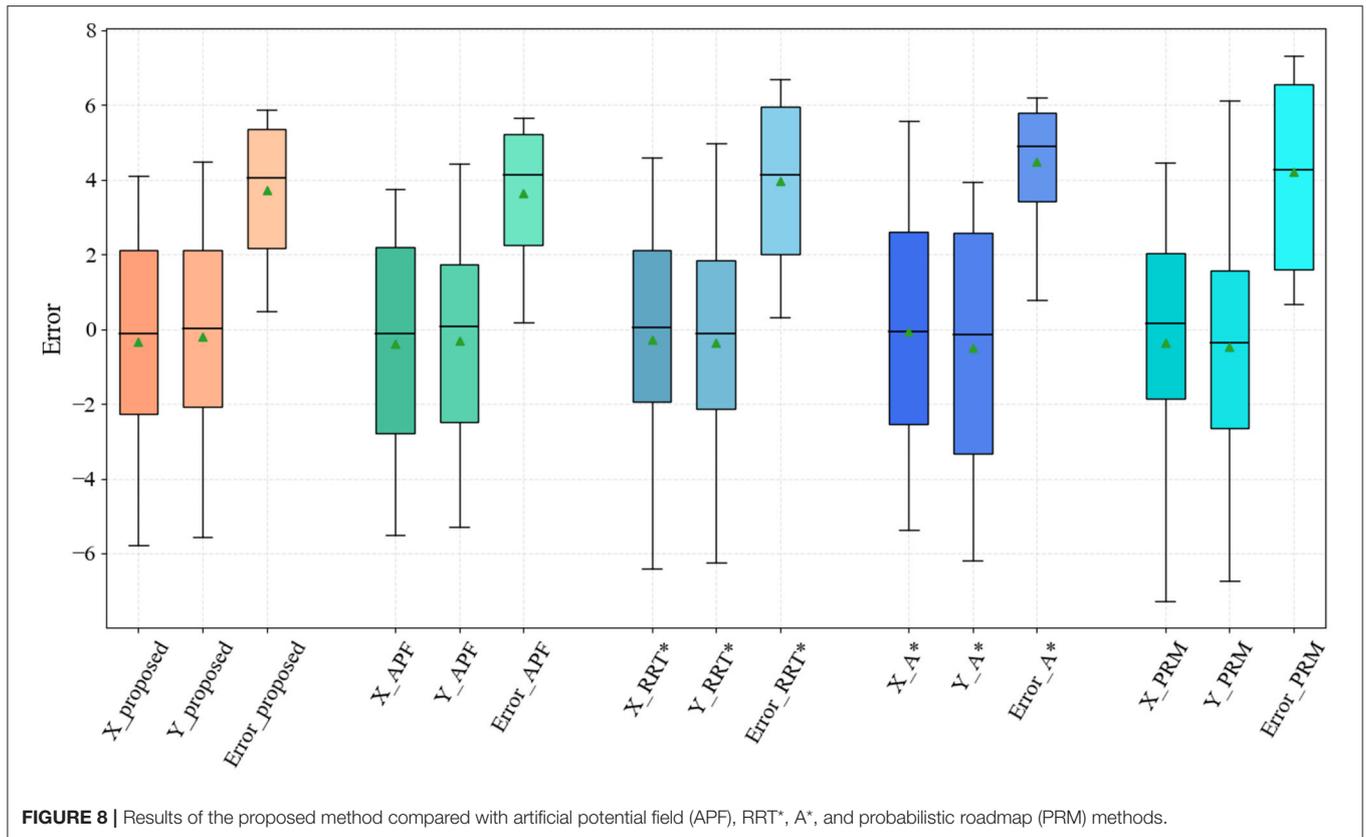


FIGURE 7 | Error estimation in path planning domain. (A) Average error in X-axis, (B) average error in Y-axis, and (C) average distance error.

TABLE 1 | Comparison of cumulative error of different methods for typical endpoints.

Endpoint	Dijkstra method			Proposed method			Error reduction ratio (%)
	x	y	Error	x	y	Error	
(170, 90)	-0.07575	-0.41536	2.324235	-0.46782	-0.33134	2.171227	7.0471
(90, 60)	-0.22252	-0.28665	0.621104	-0.17048	-0.28777	0.541869	14.62261
(130, 105)	-0.30649	-0.44412	1.634435	-0.29347	-0.36458	1.485441	10.03032
(175, 140)	-0.28543	-0.38564	2.698455	-0.43593	-0.45941	2.675029	0.875752
(240, 135)	-0.02745	-0.39179	4.497923	-0.37665	-0.36357	3.914085	14.91635
(200, 110)	-0.07465	-0.32521	3.181955	-0.32244	-0.37822	2.857053	11.37192
(125, 100)	-0.28511	-0.4254	1.49236	-0.2927	-0.4051	1.369209	8.994302
(87, 145)	-0.45045	-0.28108	1.596313	-0.49804	-0.25758	1.564632	2.024825



For point set  $Q_{dif}$ , the algorithm traverses its adjacent nodes each time to calculate the minimum error. The algorithm requires that error difference is caused by the path passing through two sides of an obstacle, so the correlation between the current node path and adjacent node path needs to be calculated. To be logical, we define  $CORRELATION(path1, path2) = \sum(dis(path1, path2) < D)/LEN(path)$ . It should be noted that, as the short distance between adjacent nodes, if their paths pass on the same side of an obstacle, the path correlation will be close to 1. The complete pseudo-code of the iteration strategy is shown in **Algorithm 3**.

In the algorithm,  $A, B, C$ , and  $D$  each represent a threshold constant, which is only for adjusting the algorithm effect and has no other representative meaning. Better convergence can be achieved by dynamically setting the threshold size according to the map size and task requirements.

## 4. SIMULATION

### 4.1. Implementation Details

A 2D grid map is used as a graphical basis for algorithmic simulations. In this article, a map with a scale of 250/150 was chosen and the starting point was randomly set to (22, 22). The priority map consists of obstacles, driveable areas, and boundaries as shown in **Figure 2**. In general, the robot can accurately reach the end-point through the tracking process, with the help of high-precision GPS. Considering cumulative

error generated by the noise-ranging sensor when GPS-denied, this article assumed that sensor error satisfies the Gaussian distribution, i.e., the error distributions in distance and angle are  $N(0, 0.01)$  and  $N(0, 0.02)$ .

The original Dijkstra method can only find the shortest path, which is not the path with the smallest cumulative error in individual scenarios. Especially every time robots pass an obstacle, it will cause a fault in the error map. The improved Dijkstra can delay turns from the starting point to each point, which is achieved by turning restrictions. The initial path graph generated by improved Dijkstra is conducive to the realization of later iterative convergence. The improved error graph is shown in **Figure 3**.

It is necessary to set a reasonable threshold in the update point set and algorithm iteration. According to the map scale in this article, we set the minimum allowable *value* that is  $A = 0.001$ , the path absolute error difference  $B = 0.1$ , the minimum allowable correlation between two paths  $C = 0.8$ , and the path correlation judgment distance is based on  $D = 5.0$ . After multiple iterations of the algorithm, the smallest error global map is finally generated, as shown in **Figure 4**. Additionally, the path of each point in the error map can be obtained by the way of parent node search, namely  $PATH(s, u)$ .

For different thresholds, there are some differences in the convergence ability of the algorithm, although convergence results can be obtained for all. Additionally, this algorithm is also suitable for sensor calculation with different error distributions.

**Algorithm 1:** Improved Dijkstra.

---

**Require:** Dijkstra( $G, s$ )  
**Ensure:** Original global error map  $G\_error$

```

1 INITIALIZE-SINGLE-SOURCE( $G, s$ )
2 for each node  $n \in G.V$  do
3    $n.turn\_num = 0$ 
4    $n.d = \infty$ 
5    $n.\pi = NIL$ 
6 end for
7  $s.turn\_num = 0$ 
8  $S = \emptyset$ 
9  $Q = G.V$ 
10 while  $Q \neq \emptyset$  do
11    $u = \text{heappop}(Q)$ 
12    $S = S \cup \{u\}$ 
13   for each node  $v \in G.Adj\_limited[u]$  do
14     if  $v.turn\_num > u.turn\_num + \omega(u, v, u.\pi)$  then
15        $v.turn\_num = u.turn\_num + \omega(u, v, u.\pi)$ 
16        $v.\pi = u$ 
17        $v.d = u.d + \omega_d(u, v)$ 
18        $v.error = \text{Equation (6)}$ 
19        $v.path = \text{PATH}(s, v)$ 
20     end if
21   end for
22 end while
```

---

**Algorithm 2:** Update Iteration Point Set.

---

**Require:** Original global error map  $G\_error$   
**Ensure:** Point set to be iterated  $Q\_dif$

```

1  $Q\_dif = \emptyset$ 
2  $Q\_temp = G.V$ 
3 while  $Q\_temp \neq \emptyset$  do
4    $u = \text{heappop}(Q\_temp)$ 
5   for each node  $v \in G.Adj[u]$  do
6     if  $v.value > A$  or  $v.value\_absolute > B$  then
7        $Q\_dif = Q\_dif \cup \{v\}$ 
8     end if
9   end for
10 end while
```

---

We changed the settings of the relevant values, and the number of points processed in each iteration eventually tended to 0, as shown in Figure 5.

## 4.2. Analysis

Using the classic path planning algorithm, all points in drivable areas can be reached from starting point through at least one path. However, considering the influence of cumulative error caused by the above-mentioned sensor noise, the actual path will deviate from the original path and endpoint to a large extent. By adding measurement noise, we select a few representative path results and apply Equation (11) calculation to compare the effect of the planning algorithm in this article to reduce cumulative error.

**Algorithm 3:** Error Map Iteration Strategy.

---

**Require:** Point set to be iterated  $Q\_dif$   
**Ensure:** Final error map  $G\_error$

```

1 while  $Q\_dif \neq \emptyset$  do
2    $u = \text{heappop}(Q\_dif)$ 
3   for each node  $v \in G.Adj[u]$  do
4     if  $\text{ERROR}(v.path + u) < u.error$  and
        $\text{CORRELATION}(v.path + u, u.path) < C$  then
5        $u.\pi = v$ 
6        $u.path = v.path + u$ 
7        $u.error = \text{ERROR}(v.path + u)$ 
8     end if
9   end for
10 end while
```

---

Set the starting point as (170, 90), (240, 135), two different paths are obtained through the classic Dijkstra method and the method in this article (the path can also be the same in some scenarios, especially the scene where the path does not pass through obstacles). In Figure 6, the results of different algorithms are represented by dashed and solid lines, respectively. In the case that the noise of the measuring sensor conforms to the Gaussian distribution, the path error calculated by 1,000 Monte Carlo runs is shown in Figure 7. The cumulative error only considers the starting point and the endpoint, and the problem of large error boundaries caused by the path process will not be within the scope of this article.

To reflect the effectiveness of the algorithm, this article selects 8 typical points, and compares the cumulative error statistics of algorithm results and the classical Dijkstra planning results, as shown in Table 1. The proposed algorithm can effectively reduce cumulative error when the sensor is biased. When a robot relies on its inertial navigation, it is easy to deviate from the default path. During the tracking process based on the proposed algorithm path, the endpoint is closer to the target point. The reduction of cumulative error verifies the effectiveness of the proposed algorithm.

The proposed method is an iterative extension of Dijkstra, and the level of cumulative error in its planning results is significantly improved compared with the original method. To evaluate the effectiveness of the proposed method, the planning results of the proposed algorithm were evaluated in comparison with typical path planning methods, such as the artificial potential field method (APF) (Wang et al., 2020), Grid-based RRT\* (RRT\*) (Chao et al., 2018), A\* (Zafar et al., 2021), and probabilistic roadmap method (PRM) (Agha-mohammadi et al., 2014) methods. The error levels of the different methods were analyzed by bypassing 1,000 Monte Carlo tests under measurement white noise, as shown in Figure 8.

The proposed method has an advantage over the existing probability-based, graph search-based planning methods at the path error level. Note that the APF method falls into local optimum several times in the test, especially in maps containing

recessed obstacles, while the method in this article does not have this problem. Among the compared methods, the PRM-based planning method has the largest path accumulation error on account of the probabilistic uncertainty.

## 5. DISCUSSION

The results of the proposed method are the same as those of the Dijkstra method when the path from the starting point does not pass through obstacles, i.e., the shortest path is also the path with the smallest error. The advantage of the proposed planning method is demonstrated after the path encounters and bypasses the obstacles. Unfortunately, the computational effort of the proposed algorithm increases exponentially as the number of obstacles increases. The algorithm in this article is suitable for applications in scenarios with sparse obstacles (e.g., underwater obstacle avoidance for AUVs). The discretization of the map contributes to the lack of smoothness of the planned paths, which can be optimized at a later stage by smoothing algorithms. This will also be a problem that we need to solve in the future. Theoretically, the proposed algorithm achieves pathfinding with minimum estimation error by traversing the global map.

## 6. CONCLUSION

To address the problem of path planning in the absence of missing global positioning, a path planning algorithm with minimum cumulative error considering sensor drift is proposed. First, the statistical characteristics of sensor noise relative to the cumulative error of the measurement are analyzed. Second, considering the cumulative error in the positioning process, the greedy search algorithm is used to traverse the global map and generate an initial error map. Finally, the proposed algorithm is iterated to generate a smooth global error map, and the path planning task is carried out accordingly. Through simulation

analysis and comparison of results, the algorithm significantly improves the safety of collision avoidance during tracking and effectively reduces the cumulative error in complex conditions.

The motion of robots is continuous and regular. Future study will need to accommodate continuous motion strategies and complex path planning tasks in multidimensional spaces and incorporate robot kinematic models to accommodate more types of robots.

## DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

## AUTHOR CONTRIBUTIONS

CW provided the original motivation and idea. CC performed the writing of the manuscript and data analysis. DY completed the design of the simulation experiment. GP provided financial support and FZ is responsible for the resources and revision of the manuscript. All authors contributed to the article and approved the submitted version.

## FUNDING

This study was supported by the National Natural Science Foundation of China (52171322), the National Key Research and Development Program (2020YFB1313200), and the Fundamental Research Funds for the Central Universities (D5000210944).

## ACKNOWLEDGMENTS

The authors appreciated the participation of all the subjects in the experiment.

## REFERENCES

- Agha-mohammadi, A.-A., Chakravorty, S., and Amato, N. (2014). FIRM: sampling-based feedback motion-planning under motion uncertainty and imperfect measurements. *Int. J. Robot. Res.* 33, 268–304. doi: 10.1177/0278364913501564
- Arrichiello, F., Heidarsson, H. K., and Sukhatme, G. S. (2012). “Opportunistic localization of underwater robots using drifters and boats,” in *IEEE International Conference on Robotics & Automation* (Saint Paul, MN).
- Batista, P., Silvestre, C., and Oliveira, P. (2013). Globally exponentially stable filters for source localization and navigation aided by direction measurements. *Syst. Control Lett.* 62, 1065–1072. doi: 10.1016/j.sysconle.2013.07.010
- Bidot, J., Karlsson, L., Lagriffoul, F., and Saffiotti, A. (2013). Geometric backtracking for combined task and motion planning in robotic systems. *Artif. Intell.* 247, 229–265. doi: 10.1016/j.artint.2015.03.005
- Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., et al. (2000). Graph structure in the web. *Comput. Netw.* 33, 309–320. doi: 10.1016/S1389-1286(00)00083-9
- Brossard, M., Barrau, A., and Bonnabel, S. (2020). AI-IMU dead-reckoning. *IEEE Trans. Intell. Veh.* 5, 585–595. doi: 10.1109/TIV.2020.2980758
- Carlson, J., Spensieri, D., Söderberg, R., Bohlin, R., and Lindkvist, L. (2013). Non-nominal path planning for robust robotic assembly. *J. Manuf. Syst.* 32, 429–435. doi: 10.1016/j.jmsy.2013.04.013
- Chao, N., Liu, Y.-K., Xia, H., Ayodeji, A., and Bai, L. (2018). Grid-based RRT\* for minimum dose walking path-planning in complex radioactive environments. *Ann. Nucl. Energy* 115, 73–82.
- Chew, W. K., and Zakaria, M. A. (2019). Outdoor localisation for navigation tracking using differential global positioning system estimation (DGPS): positioning errors analysis. *Mekatronika*. 1, 103–114. doi: 10.15282/mekatronika.v1i2.4994
- Choset, H., Lynch, K., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., et al. (2005). *Principles of Robot Motion: Theory, Algorithms and Implementation*. Robotics & Automation Magazine IEEE.
- Dai, Z., Liu, C., X. Sheng, and Zhang, D. (2016). The error analysis and calibration of delta parallel robot. *Mechatronics* 22, 8–12. doi: 10.16413/j.cnki.issn.1007-080x.2016.03.002
- Eaton, C. M., Chong, E. K. P., and Maciejewski, A. A. (2017). “Robust UAV path planning using POMDP with limited FOV sensor,” In *2017 IEEE Conference on Control Technology and Applications (CCTA)* (Maui, HI), 1530–1535.

- Fallon, M. F., Papadopoulos, G., and Leonard, J. J. (2010). "Cooperative AUV navigation using a single surface craft," in *Field and Service Robotics*, eds A. Howard, K. Iagnemma, and A. Kelly (Berlin: Springer), 331–340.
- He, M., Zhu, C., Huang, Q., Ren, B., and Liu, J. (2020). A review of monocular visual odometry. *Vis. Comput.* 36, 1053–1065. doi: 10.1007/s00371-019-01714-6
- Ibraheem, I., and Hassan Ajeil, F. (2018). Path planning of an autonomous mobile robot in a dynamic environment using modified bat swarm optimization. *arXiv arXiv:1807.05352*.
- Kang, H. I., Lee, B., and Kim, K. (2008). "Path planning algorithm using the particle swarm optimization and the improved Dijkstra algorithm," in *2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, vol. 2 (Wuhan), 1002–1004.
- Lee, S., Lim, W., and Sunwoo, M. (2020). Robust parking path planning with error-adaptive sampling under perception uncertainty. *Sensors* 20, 3560. doi: 10.3390/s20123560
- Li, G., Svogor, I., and Beltrame, G. (2019). Long-term pattern formation and maintenance for battery-powered robots. *Swarm Intell.* 13, 21–57. doi: 10.1007/s11721-019-00162-1
- Lv, L., Zhang, S., Ding, D., and Wang, Y. (2019). Path planning via an improved DQN-based learning policy. *IEEE Access* 7, 67319–67330. doi: 10.1109/ACCESS.2019.2918703
- Mansouri, S. S., Kanellakis, C., Lindqvist, B., Pourkamali-Anaraki, F., and Nikolakopoulos, G. (2020). A unified NMPC scheme for mavs navigation with 3D collision avoidance under position uncertainty. *IEEE Robot. Autom. Lett.* 5, 5740–5747. doi: 10.1109/LRA.2020.3010485
- Marchel, U., Naus, K., and Specht, M. (2020). Optimisation of the position of navigational aids for the purposes of SLAM technology for accuracy of vessel positioning. *J. Navig.* 73, 282–295. doi: 10.1017/S0373463319000584
- Miller, P., A., Farrell, J., A., Zhao, Y., and Djapic, V. (2010). Autonomous underwater vehicle navigation. *IEEE J. Ocean. Eng. J. Devoted Appl. Elect. Electron. Eng. Ocean. Environ.* 35, 663–678. doi: 10.1109/JOE.2010.2052691
- Papachristos, C., Mascarich, F., Khattak, S., Dang, T., and Alexis, K. (2019). Localization uncertainty-aware autonomous exploration and mapping with aerial robots using receding horizon path-planning. *Auton. Robots* 43, 2131–2161. doi: 10.1007/s10514-019-09864-1
- Park, J., Choi, J., and Choi, H.-T. (2018). COLREGS-compliant path planning considering time-varying trajectory uncertainty of autonomous surface vehicle. *Electron. Lett.* 55, 222–224. doi: 10.1049/el.2018.6680
- Paull, L., Saeedi, S., Seto, M., and Li, H. (2014). AUV navigation and localization: a review. *IEEE J. Ocean. Eng.* 39, 131–149. doi: 10.1109/JOE.2013.2278891
- Peng, Y., and Green, P. N. (2019). Environment mapping, map constructing, and path planning for underwater navigation of a low-cost  $\mu$ AUV in a cluttered nuclear storage pond. *IAES Int. J. Robot. Autom. (IJRA)* 8, 277–292. doi: 10.11591/ijra.v8i4.pp277-292
- Pilania, V., and Gupta, K. (2017). Localization aware sampling and connection strategies for incremental motion planning under uncertainty. *Auton. Robot.* 41, 111–132. doi: 10.1007/s10514-015-9536-y
- Sainte Catherine, M., and Lucet, E. (2020). "A modified hybrid reciprocal velocity obstacles approach for multi-robot motion planning without communication," in *Proceedings of the 2020 IEEE/RISJ International Conference on Intelligent Robots and Systems (IROS)* (Las Vegas, NV: IEEE). doi: 10.1109/IROS45743.2020.9341377
- Thomson, D., Dosso, S. E., and Barclay, D. R. (2017). Modeling AUV localization error in a long baseline acoustic positioning system. *IEEE J. Ocean. Eng.* 43, 955–968. doi: 10.1109/JOE.2017.2771898
- Tick, D., Satici, A., C., Shen, J., Gans, and N. (2013). Tracking control of mobile robots localized via chained fusion of discrete and continuous epipolar geometry, IMU and odometry. *IEEE Trans. Cybern.* 43, 1237–1250. doi: 10.1109/TSMCB.2012.2227720
- Wang, C., Cheng, C., Yang, D., Zhang, F., and Pan, G. (2021). "Path Planning and Simulation Based on Cumulative Error Estimation," in *Cognitive Systems and Signal Processing. ICCSIP 2020. Communications in Computer and Information Science, Vol. 1397*, eds F. Sun, H. Liu, and B. Fang (Singapore: Springer), 131–141. doi: 10.1007/978-981-16-2336-3\_12
- Wang, D., Wang, P., Zhang, X., Guo, X., Shu, Y., and Tian, X. (2020). An obstacle avoidance strategy for the wave glider based on the improved artificial potential field and collision prediction model. *Ocean Eng.* 206, 107356. doi: 10.1016/j.oceaneng.2020.107356
- Xu, B., Liu, Y., Shan, W., and Wang, G. (2014). Error analysis and compensation of gyrocompass alignment for SINS on moving base. *Math. Problems Eng.* 2014, 1–18. doi: 10.1155/2014/373575
- Yilmaz, B. M., Tatlicioglu, E., Savran, A., and Alci, M. (2021). Self-adjusting fuzzy logic based control of robot manipulators in task space. *IEEE Trans. Ind. Electron.* 69, 1620–1629. doi: 10.1109/TIE.2021.3063970
- Yilmaz, N. K., Evangelinos, C., Lermusiaux, P., and Patrikalakis, N. M. (2008). Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming. *IEEE J. Ocean. Eng.* 33, 522–537. doi: 10.1109/JOE.2008.2002105
- Yin, X., Sun, Y., and Wang, C. (2013). Positioning errors predicting method of strapdown inertial navigation systems based on PSO-SVM. *Abstract Appl. Anal.* 2013, 1401–1429. doi: 10.1155/2013/737146
- Zafar, M., Anjum, M., and Hussain, W. (2021). LTA\*: local tangent based a\* for optimal path planning. *Auton. Robots* 45, 209–227. doi: 10.1007/s10514-020-09956-3
- Zhang, F., and Knoll, A. (2016). Systematic error modeling and bias estimation. *Sensors* 16, 729. doi: 10.3390/s16050729
- Zhang, F., Simon, C., Chen, G., Buckl, C., and Knoll, A. (2013). "Cumulative error estimation from noisy relative measurements," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)* (The Hague), 1422–1429.

**Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

**Publisher's Note:** All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Wang, Cheng, Yang, Pan and Zhang. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.