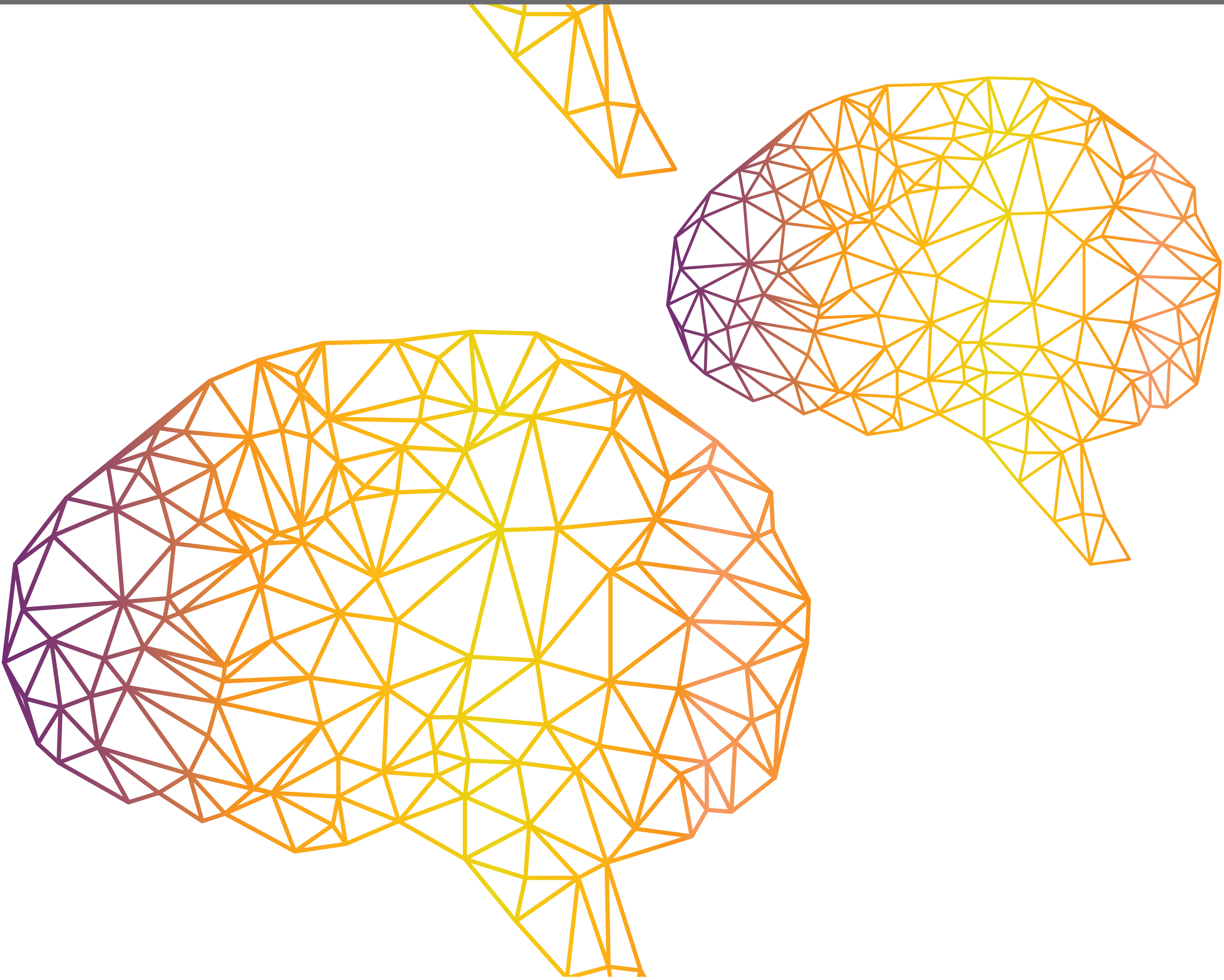
INTELLIGENT CONTROL AND APPLICATIONS FOR ROBOTICS

EDITED BY: Yimin Zhou, Chen Qiao, Lianghong Wu and Huiyu Zhou
PUBLISHED IN: Frontiers in Neurorobotics



frontiers Research Topics

ISSN 1664-8714
ISBN 978-2-83250-095-8
DOI 10.3389/978-2-83250-095-8

About Frontiers

Frontiers is more than just an open-access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers Journal Series

The Frontiers Journal Series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the Frontiers Journal Series operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to Quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews.
Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the Frontiers Journals Series: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area! Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers Editorial Office: frontiersin.org/about/contact

INTELLIGENT CONTROL AND APPLICATIONS FOR ROBOTICS

Topic Editors:
Yimin Zhou, Shenzhen Institutes of Advanced Technology (CAS), China
Chen Qiao, Xi'an Jiaotong University, China
Lianghong Wu, Hunan University of Science and Technology, China
Huiyu Zhou, University of Leicester, United Kingdom

Citation: Zhou, Y., Qiao, C., Wu, L., Zhou, H., eds. (2022). Intelligent Control and Applications for Robotics. Lausanne: Frontiers Media SA.
doi: 10.3389/978-2-83250-095-8

Table of Contents

frontiers
in Neurorobotics

ORIGINAL RESEARCH
published: 05 November 2021
doi: 10.3389/fnbot.2021.770361

An Improved PSO-GWO Algorithm With Chaos and Adaptive Inertial Weight for Robot Path Planning

Xuezhen Cheng [1], ***Jiming Li*** [1], ***Caiyun Zheng*** [2], ***Jianhui Zhang*** [1] and ***Meng Zhao*** [1]*

[1] College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao, China, [2] State Grid Dongying District of Dongying City Power Supply Company, Dongying, China

The traditional particle swarm optimization (PSO) path planning algorithm represents each particle as a path and evolves the particles to find an optimal path. However, there are problems in premature convergence, poor global search ability, and to the ease in which particles fall into the local optimum, which could lead to the failure of fast optimal path obtainment. In order to solve these problems, this paper proposes an improved PSO combined gray wolf optimization (IPSO-GWO) algorithm with chaos and a new adaptive inertial weight. The gray wolf optimizer can sort the particles during evolution to find the particles with optimal fitness value, and lead other particles to search for the position of the particle with the optimal fitness value, which gives the PSO algorithm higher global search capability. The chaos can be used to initialize the speed and position of the particles, which can reduce the prematurity and increase the diversity of the particles. The new adaptive inertial weight is designed to improve the global search capability and convergence speed. In addition, when the algorithm falls into a local optimum, the position of the particle with the historical best fitness can be found through the chaotic sequence, which can randomly replace a particle to make it jump out of the local optimum. The proposed IPSO-GWO algorithm is first tested by function optimization using ten benchmark functions and then applied for optimal robot path planning in a simulated environment. Simulation results show that the proposed IPSO-GWO is able to find an optimal path much faster than traditional PSO-GWO based methods.

Keywords: path planning, improved particle swarm optimization, robot, gray wolf algorithm, adaptive inertia weight, chaos

OPEN ACCESS

Edited by:
Yimin Zhou,
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences (CAS), China

Reviewed by:
Yanjiang Wang,
China University of Petroleum, China
Shubo Wang,
Qingdao University, China

****Correspondence:***
Meng Zhao
zhaomeng@sdust.edu.cn

Received: *03 September 2021*
Accepted: *11 October 2021*
Published: *05 November 2021*

Citation:
Cheng X, Li J, Zheng C, Zhang J and Zhao M (2021) An Improved PSO-GWO Algorithm With Chaos and Adaptive Inertial Weight for Robot Path Planning.
Front. Neurorobot. 15:770361.
doi: 10.3389/fnbot.2021.770361

INTRODUCTION

Along with the development of automation technology and robotics, path planning is important in robot task execution when searching for an optimal path from the starting position to the target position with obstacle avoidance based on certain criteria.

There have been many achievements in robot path planning. The current path planning algorithms mainly include the colony algorithms (Liu et al., 2019; Ye et al., 2020; Zhang et al., 2020; Zhu et al., 2020), PSO (Krell et al., 2019; Wang Y. B. et al., 2019; Liu X. H. et al., 2021; Song et al., 2021), A* algorithms (Xiong et al., 2020; Liu Z. H. et al., 2021; Tang et al., 2021; Tullu et al., 2021), artificial potential field methods (Wang P. W. et al., 2019; Azmi and Ito, 2020; Song et al., 2020; Yao et al., 2020), genetic algorithms (Hao et al., 2020; Li K. R. et al., 2021; Wen et al., 2021), fuzzy control algorithms (Guo et al., 2020; Zhi and Jiang, 2020), fast marching algorithms (Sun et al., 2021; Wang et al., 2021; Xu et al., 2021), and deep reinforcement learning algorithms (Li L. Y. et al., 2021; Lin et al., 2021; Xie et al., 2021). PSO is an evolutionary computation algorithm that can be used to find the optimal solution through collaboration and information sharing between individuals in the group, as in path planning, the optimal solution is to find the shortest path. The PSO algorithm is easy to implement and has fewer adjustable parameters, however, it still has problems such as being easy to fall into the local optimum and slow convergence.

In response to these problems, researchers have extensively studied PSO improvement in recent years. Das and Jena (2020) used a genetic algorithm that inherits multiple crossover operators and bee colony operators as two evolutionary operators to improve the optimization ability of the PSO. Shao et al. (2020) designed the constant acceleration coefficient and the maximum speed as the adaptive linear variation to adapt to the optimization process. Further, a particle mutation strategy has been proposed to enhance the convergence speed of the algorithm. Li and Chou (2018) applied different strategies to realize the adaptive learning of the PSO; they turned the problem of path planning into a minimizing multi-objective optimization problem and proposed a new adaptive learning mechanism to improve the search ability of the PSO algorithm.

Although the performance of the PSO algorithm has been improved greatly, there are still some shortcomings when it is applied for complex problems (Phung and Ha, 2021), i.e., premature convergence, poor global search capability, and slow convergence speed. To this end, an improved particle swarm optimization combined gray wolf optimization (IPSO-GWO) is proposed in this paper. The GWO can sort the particles during iteration to find the particles with the optimal fitness value and lead other particles to search for the position of the particles with the optimal fitness value, which can greatly improve the search ability of the PSO algorithm in the global searching space. The chaos is further adopted to initialize the speed and position of the swarm particles and a new adaptive inertia weight is designed to improve the global search capability and convergence speed of the IPSO-GWO. When the algorithm falls into the local optimum, chaos can make the algorithm quickly jump out of the local optimum. Experiments on benchmark functions optimization test and the robot path planning simulation tests demonstrate that the IPSO-GWO algorithm has faster convergence speed.

The remainder of the paper is organized as follows. The Proposed Method section describes the proposed algorithm including environment settings, IPSO-GWO, chaos based and new inertial weight design. Experiments and result analysis are explained in third section. The conclusion is given in fourth section.

THE PROPOSED METHOD

Environment Modeling

The working environment of the robot is established through a grid model, which can be divided into $N \times N$ squares, as seen in **Figure 1**. The black grid represents obstacles which are impassable, and the white grids represent feasible passing free areas, denoted as 0 and 1, respectively. The five-pointed star indicates the starting point and the green point is the target point. Then the grid model is placed in the coordinate system so as to establish the robot working environment.

It can be seen from **Figure 1** that the model is easy to construct, represent, and store data for processing and it is convenient for computer processing.

PSO Algorithm

The PSO algorithm is an intelligent optimization algorithm proposed by Kennedy and Eberhart (1995) based on the study of the living habits of animal flocks (Tang et al., 2020). Suppose the optimal solution of a certain problem exists in D dimensional space for a swarm with size m, and the population can be expressed as, $Swarm = \{x_1, x_2, \ldots, x_m\}$ where $x_i\ (i = 1, \cdots, m)$ is the particle without mass, k represents the total number of the required iterations, and the position information of the ith particle in the kth iteration can be represented by a d-dimensional vector $x_i^k = \left(x_{i1}^k, x_{i2}^k, \ldots, x_{id}^k\right), i = 1, 2, \ldots m$, the velocity of each particle can be represented as $v_i^k = \left(v_{i1}^k, v_{i2}^k, \cdots, v_{id}^k\right), i = 1, 2, \cdots, m$. In each iteration, the position and velocity of the particles are dynamically adjusted according to the historical optimal fitness values of each particle and the population. The calculation for the $(k+1)th$ iterations of the ith particle in d-dimensional space can be written as,

$$v_{id}^{k+1} = v_{id}^k + c_1 * rand() * \left(p_{id}^k - x_{id}^k\right) + c_2 * rand() * \left(p_{gd}^k - x_{id}^k\right) \tag{1}$$

$$x_{id}^{(k+1)} = x_{id}^{(k)} + v_{id}^{(k+1)} \tag{2}$$

where c_1 and c_2 represent the learning factors. c_1 and c_2 are the control variables to control the step lengths of the individual particle flying toward the local optimal value and the swarm optimal value, respectively. p_{id}^k is the historical optimal fitness value of each particle in the optimization process, p_{gd}^k is the optimal fitness value reached by all particles, that is, the optimal fitness value of the population; the $rand()$ function is to generate a random number between (0,1) to differentiate particles. The subscript $d (1 \leq d \leq D)$ represents the dimension of the searching space. In the above Equation (1) and (2), the speed of the PSO is composed of the local and global three parts: v_{id}^k represents the speed of the particle at the k^{th} iteration, $\left\{c_1 * rand() * \left(p_{id}^k - x_{id}^k\right)\right\}$ represents the information of the particle itself, and $\left\{c_2 * rand() * \left(p_{gd}^k - x_{id}^k\right)\right\}$ represents the part of the particle in the population for collaboration and information sharing.

PSO-GWO Algorithm

The PSO-GWO algorithm is an improved PSO version incentive inspired by gray wolf predation (Narinder and Singh, 2017; Teng et al., 2019; Gul et al., 2021). Different from bird flocks, the gray

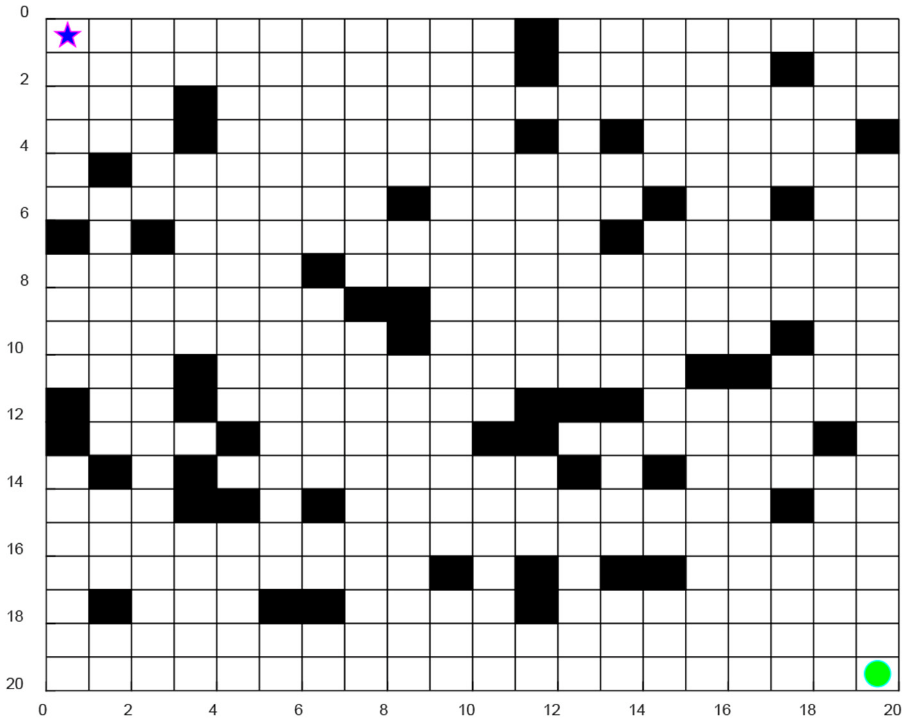


FIGURE 1 | Environment modeling.

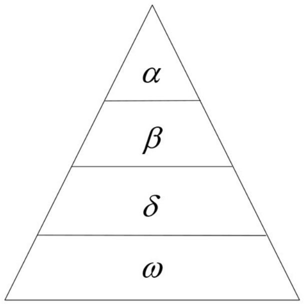


FIGURE 2 | Grey wolf social hierarchy.

wolf pack is quite a strict socially hierarchical organization; its hierarchical arrangement is illustrated in **Figure 2**.

The first layer α in **Figure 2** represents the leader in the population, where the leader α is the core of the wolf pack, being mainly responsible for leading and assigning tasks.

The second layer β in **Figure 2** represents the think tank team, which is used to assist the leader in management, i.e., when leader α is vacant, β will quickly take over the position of α. In the entire wolf pack, the status of β is only lower than that of α. If α is occupied, β is an advisor to α and discipliner for the group.

The third layer δ follows the command and management of leader α and think tank, and are mainly responsible for care and supervision.

The function of the fourth layer, ω, is to balance the membership within the population.

The essence of the GWO is that the particle with the highest fitness is taken as the leader α to manage other particles. The specific steps of the GWO are summarized as follows:

Step 1: To initialize particles of one population in the searching space;
Step 2: To rank the particles according to the historical best fitness values;
Step 3: Taking three particles with the highest fitness values set as α, β and δ, the other particles are arranged in sequence. If an individual with a higher fitness value appears in the iterative process, it is set as the new leader α and p_{gd} is updated with its individual fitness. $x_i = (x_{i1}, x_{i2}, ..., x_{iN})$ represents the position of the ith particle, and $v_i = (v_{i1}, v_{i2}, ..., v_{iN})$ is the speed of the ith particle. In the $k + 1$ iteration, the positions of the three particles with the best fitness values in the population are updated *via* Equation (3), and the positions of the rest particles are updated *via* Equations (4) and (5):

$$\begin{aligned}\vec{d}_{\alpha} &= \left|\vec{c}_{1} \cdot \vec{x}_{\alpha} - w * \vec{x}\right| \\ \vec{d}_{\beta} &= \left|\vec{c}_{2} \cdot \vec{x}_{\beta} - w * \vec{x}\right| \\ \vec{d}_{\delta} &= \left|\vec{c}_{2} \cdot \vec{x}_{\delta} - w * \vec{x}\right| \end{aligned} \tag{3}$$

$$\begin{aligned} v_i^{k+1} &= w * (v_i^k + c_1 rand()\left(x_1 - x_i^k\right) \\ &\quad + c_2 rand()\left(x_2 - x_i^k\right) + \ c_3 rand()\left(x_3 - x_i^k\right)\Big) \end{aligned} \tag{4}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{5}$$

As seen in Equations (3)–(5), the dimension of the spatial solution is d, and the current number of the iteration is k. c_1, c_2, and c_3 represent the learning factors, *rand*() is a random number between (0, 1), and w represents the inertia weight coefficient. The larger w makes the algorithm better in global search, in contrast, the smaller w is more suitable for local search. The core idea of the PSO-GWO is to arrange the particles according to their fitness values during each iteration, and set the three particles with the best fitness values to α, β, and δ, while these three particles can predict the approximate range of the location where the optimal solution may exist, and the remaining particles can search the optimal solution within the predicted range. In such a way, the particles can find the optimal solution more quickly and effectively with the improved convergence performance, so the path planning ability of the PSO algorithm can be improved accordingly.

IPSO-GWO Algorithm

In the previous section, GWO was added to PSO to form the PSO-GWO algorithm, and the search ability of the PSO algorithm can be enhanced to improve its path planning ability, but the PSO algorithm still has the problem of premature convergence, and its convergence speed and global search ability can be further strengthened. Hence chaos and a new adaptive inertia weight are added to provide solutions for these problems.

PSO With Chaos

The PSO can randomly distribute particles while the optimal solution is highly related to the particle initialization. The more uniform the initial particle distribution, the richer the diversity of the group, and the faster the optimal solution can be obtained.

Chaos (Demir et al., 2020; Lian et al., 2020; Lu et al., 2020; Wu et al., 2020; Guo et al., 2021; Ouertani et al., 2021) refers to a nonlinear motion that can traverse all situations within a specified range. A chaotic sequence can represent all states in a prescribed space, which is commonly generated by mapping. Many researchers have found that chaotic mapping has unpredictable characteristics when studying chaotic mapping relations. Although it is somewhat unpredictable, certain laws can still be used in the mapping process. The most commonly used form of chaotic mapping is logistic mapping, as shown in Equation (6):

$$Z_{i+1} = \mu Z_i \left(1 - Z_i\right) \quad i = 0, 1, 2, ...; \quad \mu \in (0, 4] \tag{6}$$

In Equation (6), $0 \leq Z_0 \leq 1$, Z_i is the value obtained by i times Logistic mapping of Z_0, and μ represents the control variable. When $\mu = 4$, the system is within a completely chaotic state and the range of the chaotic space is $[\,0, 1]$.

The steps of using chaos to initialize the PSO are summarized as follows. First, an n-dimensional vector $Z_1 = (z_1, z_2, \ldots, z_n)$ is randomly generated, and Equation (6) is used to map the other vectors so as to generate a chaotic sequence $Z_1, Z_2, \ldots, Z_N$. Then the chaotic sequence z_i is inversely mapped from the chaotic space$[0, 1]$ to the space $[a, b]$ where the optimal solution is located, and the particle position is $x_{ij} = a + (b - a)z_{ij}$, $j = 1, 2, \ldots, n$, $i = 1, 2, \ldots, N$. Finally, the particles with higher fitness values are determined as the initial particles of the population.

When the PSO is trapped in the local optimum, the algorithm will select the historical optimal value of the particles in the iterative process and convert it into a chaotic sequence through inverse mapping to obtain the optimal position of the particle, then randomly replace a certain particle position in the current search space so that the local optimum can be jumped out by the algorithm. Whether particles fall into precocity is determined by the variance of the population fitness, calculated as,

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{f_i - f_{avg}}{f} \right)^2$$
$$f = max(1, max \left| f_i - f_{avg} \right|) \tag{7}$$

where n is the size of the population, f_i represents the adaptability of the first particle, and f_{avg} represents the average adaptability of the current swarm particles. The population variance σ^2 reflects the precocious state of the particles. When σ^2 is less than a certain threshold, it is calculated that the particle algorithm will fall into precocity. Then the chaos is applied to process the optimal particles to increase the diversity of the population. The detailed steps are described as follows.

Step 1: Select the optimal position in the iterative process and use the function Logistic to map it into the chaotic space $[\,0, 1]$.
Step 2: Use logistics to generate a new sequence and inversely map the sequence to the population.
Step 3: Calculate the optimal adaptability of the particles and conclude whether the particle has jumped out of the local optimum; then record the optimal fitness value and set the corresponding particles to α, β, and δ.
Step 4: Use the current optimal chaotic particles to manage the particles in the particle swarm to make the particles leave the local optimum.

After the particle swarm performs the chaotic initialization operation, the particles are more evenly distributed in the search space, and the chaotic sequence can be used to reduce the prematurity, improve the diversity of particles, and enhance the convergence speed of the algorithm.

A New Adaptive Inertial Weight

It is known that the quality of PSO is closely related to inertia weight where the local search ability of the algorithm is higher with smaller inertial weight and global search capability is stronger with larger inertial weight. To enable the algorithm maintaining higher search ability during the entire operation process, many methods have been proposed to adjust the inertia weight (Li et al., 2019a,b; Gopal et al., 2020; Wang et al., 2020; Wang, 2021; Zhang et al., 2021). However, the current inertia weight improvement methods have a close relationship with the iteration number and cannot adapt to the nonlinear variations well. For this reason, this paper deals with the inertia weight *via* the particle adjacent fitness values. The inertia weight can be updated and calculated as,

$$w = (w_{max} + w_{min}) * a - \frac{w_{max} * k}{MaxIter}$$
$$a = \frac{p^{k}_{gbest}}{p^{k-1}_{gbest}} \tag{8}$$

In Equation (8), the global optimal fitness of the kth iteration is p^{k}_{gbest}, and the global optimal fitness of the $(k-1)th$ iteration is p^{k-1}_{gbest}; the maximum ω_{max} and minimum ω_{min} of ω is set as 0.9 and 0.4, respectively. k is the current iteration number; *MaxIter* represents the maximum number of the iterations. It can be seen from Equation (8) that a is larger at the beginning of the iteration, so the algorithm has strong global searching ability, and a becomes smaller gradually at the later iteration stage, so the algorithm has strong local search ability. In summary, the inertia weight combined with the fitness ratio of the neighboring particles can adaptively adjust the size of w with the number of iterations so that the algorithm has a higher global search ability.

Path Planning

The steps of the IPSO-GWO algorithm for path planning are summarized as follows:

Step 1: The velocities and positions of the swarm particles are initialized by chaos using logistic function, while the position of each particle represents a path and the fitness of the particle represents the length of the path;

Step 2: Collision detection is performed on the path represented by the particles. If the path collides with an obstacle, the path is adjusted without obstacle collision;

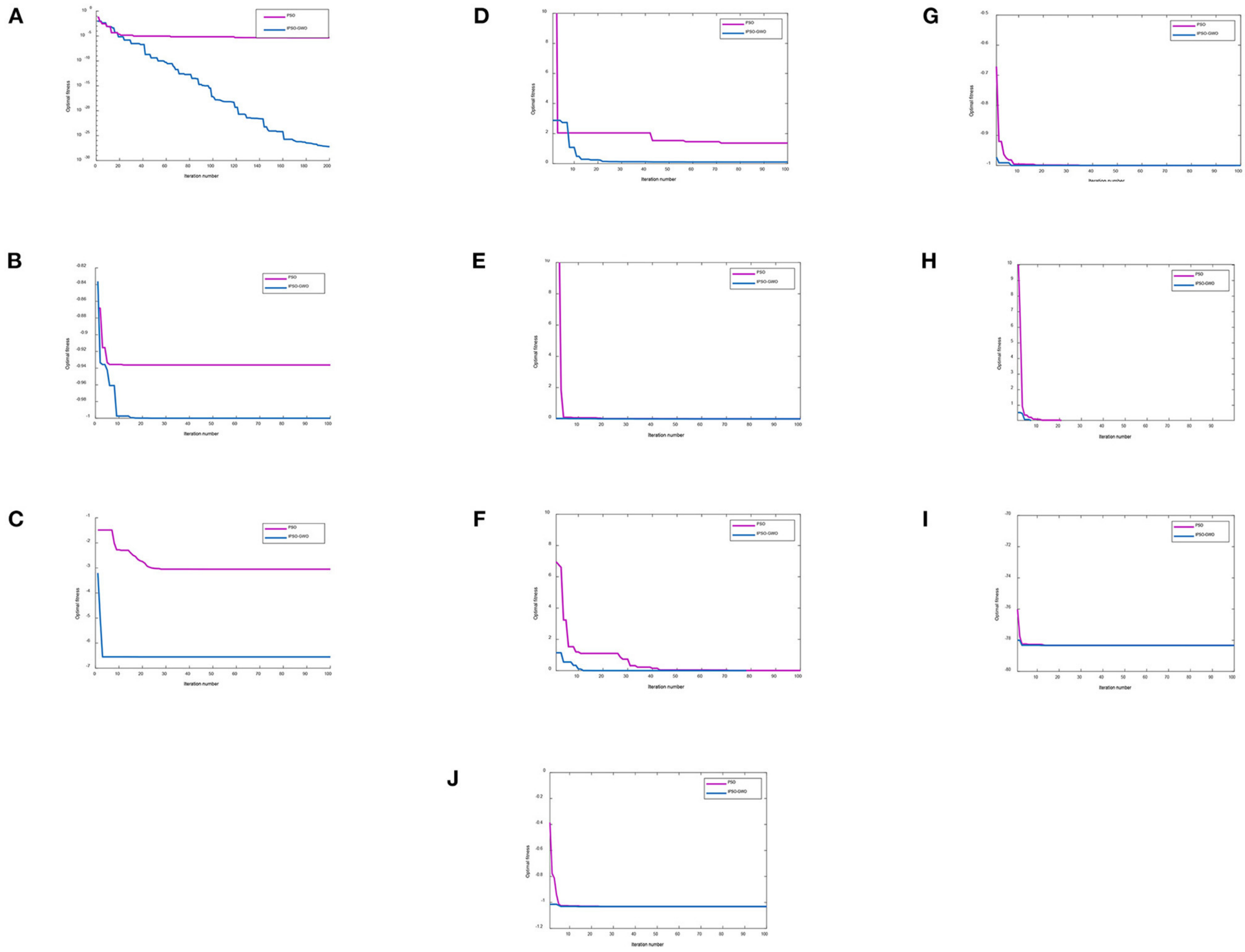


FIGURE 3 | Iteration comparison curves with different functions. **(A)** Iteration curve comparison diagram with Rosenbrock function. **(B)** Iteration curve comparison diagram with Drop Wave function. **(C)** Iteration curve comparison diagram with Peaks function. **(D)** Iteration curve comparison diagram with Bukin function. **(E)** Iteration curve comparison diagram with Booth function. **(F)** Iteration curve comparison diagram with Rastrigin function. **(G)** Iteration curve comparison diagram with Easom function. **(H)** Iteration curve comparison diagram with Levy function. **(I)** Iteration curve comparison diagram with Styblinski-Tang function. **(J)** Iteration curve comparison diagram with Six-Hump Camel function.

Step 3: The fitness values of the particles are evaluated to select the three particles with the largest fitness values, set as α, β, and δ;

Step 4: To update the positions of particles α, β, and δ based on Equation (3), update the velocities and the positions of the rest particles *via* Equations (4) and (5);

Step 5: Determine whether the algorithm has fallen into prematurity *via* Equation (7); if so, chaos is applied to process the premature particles and jump to Step 2;

Step 6: Determine whether the algorithm meets the termination condition. If it is satisfied, the iteration stops and the optimal path is obtained; otherwise, continue to Step 2 for calculation.

EXPERIMENT AND RESULT ANALYSIS

Benchmark Experiments

To verify the superiority of the IPSO-GWO algorithm, this paper uses MATLAB R2018b software to perform benchmark function tests on PSO and IPSO-GWO. The variables of the simulation experiments are set as follows: the population size is 50, the dimension of the optimization variable is 4, the learning factor $c_1 = c_2 = 2.05$, and the test functions of the simulation experiments are ten benchmark functions such as Drop Wave, Peaks, Rosenbrock, etc. For Rosenbrock function, the number of the iterations is 200, and the number of iterations of others is 100. Simulation experiments are performed on the above functions. The algorithm iteration curve is shown in **Figure 3**. The experimental results of the three test functions are analyzed and compared, as listed in **Tables 1–10**.

It can be seen from **Figure 3** that, compared with the PSO and PSO-GWO algorithms, the IPSO-GWO algorithm converges the fastest. From **Tables 1–10**, it is seen that the results obtained from IPSO-GWO is closer to the global minimum, which verify that the performance of the IPSO-GWO algorithm is higher than those of the other algorithms.

TABLE 1 | Rosenbrock function test results.

Algorithm	Maximum	Minimum	Average value	Standard deviation	Global Minimum
PSO	1.8143E−12	3.272E−16	1.70543E−13	4.4715E−13	0
IPSO-GWO	9.8215E−27	2.3419E−31	2.2383E−27	2.952E−27	0

TABLE 2 | Drop Wavefunction test results.

Algorithm	Maximum	Minimum	Average value	Standard deviation	Global minimum
PSO	−0.93625	−0.99997	−0.99192	0.015633	−1
IPSO-GWO	−0.93625	−1	−0.99636	0.009047	−1

TABLE 3 | Peaks function test results.

Algorithm	Maximum	Minimum	Average value	Standard deviation	Global minimum
PSO	−3.0395	−6.5511	−4.62282	1.741408	−6.5511
IPSO-GWO	−3.0496	−6.5511	−5.50012	1.604133	−6.5511

Path Planning Experiments

To verify the superiority of the IPSO-GWO optimization in robot global path planning, we have carried out two kinds of path planning simulation tests: one is the test of IPSO with PSO, and the other is the test of IPSO with Genetic algorithm (GA) and Ant Colony Optimization (ACO). Both tests use 20 × 20 and 30 × 30 map environments. For IPSO-GWO and PSO, the population size is set to 50 and $c_1 = c_2 = 1.6$. For GA, the crossover probability is set to 0.8, the mutation probability is set to 0.2, and the population size is set to 50. For ACO, the stimulating factor of the pheromone concentration α is set to 1, the stimulating factor of visibility β is set to 7, pheromone evaporation coefficient ρ is set to 0.3, pheromone intensity is set to 1, and the number of iterations of the four

TABLE 4 | Bukin function test results.

Algorithm	Maximum	Minimum	Average value	Standard deviation	Global minimum
PSO	27.5233	1.3670	2.1584	3.3801	0
IPSO-GWO	2.8783	0.1078	0.3540	0.7018	0

TABLE 5 | Booth function test results.

Algorithm	Maximum	Minimum	Average value	Standard deviation	Global minimum
PSO	16.1673	0.0020	0.3429	2.1682	0
IPSO-GWO	0.0278	0.0001	0.0042	0.0076	0

TABLE 6 | Rastrigin function test results.

Algorithm	Maximum	Minimum	Average value	Standard deviation	Global minimum
PSO	6.9599	0.0073	0.5837	1.2599	0
IPSO-GWO	1.1406	0	0.0835	0.2501	0

TABLE 7 | Easom function test results.

Algorithm	Maximum	Minimum	Average value	Standard deviation	Global minimum
PSO	−0.6713	−0.9999	−0.9934	0.0351	−1
IPSO-GWO	−0.9727	−1	−0.9992	0.0033	−1

TABLE 8 | Levy function test results.

Algorithm	Maximum	Minimum	Average value	Standard deviation	Global minimum
PSO	10.9906	0.0016	0.2517	1.3966	0
IPSO-GWO	0.5333	0.0001	0.0191	0.0890	0

TABLE 9 | Styblinski-Tang function test results.

Algorithm	Maximum	Minimum	Average value	Standard deviation	Global minimum
PSO	−76.0206	−78.3296	−78.2937	0.2377	−78.3322
IPSO-GWO	−77.9984	−78.3322	−78.3231	0.0468	−78.3322

TABLE 10 | Six-Hump Camel function test results.

Algorithm	Maximum	Minimum	Average value	Standard deviation	Global minimum
PSO	−0.3852	−1.0313	−1.0185	0.0726	−1.0316
IPSO-GWO	−1.0141	−1.0316	−1.0307	0.0035	−1.0316

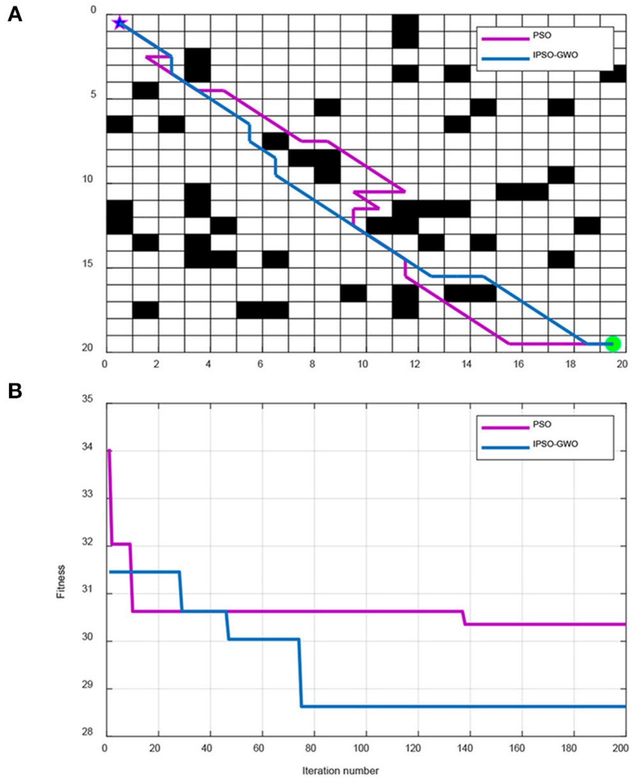


FIGURE 4 | Simulation results of IPSO-GWO and PSO in 20 × 20 map environment. **(A)** Path comparison with PSO and IPSO-GWO algorithms. **(B)** Iterative curves with PSO and IPSO-GWO algorithms.

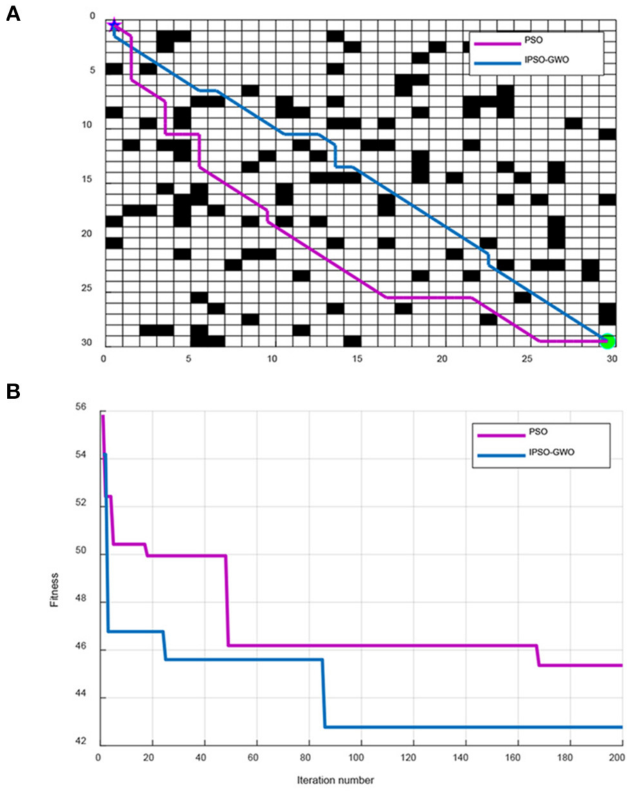


FIGURE 5 | Simulation results of IPSO-GWO and PSO in 30 × 30 map environment. **(A)** Path comparison with PSO and IPSO-GWO algorithms. **(B)** Iterative curves with PSO and IPSO-GWO algorithms.

algorithms is set to 200. The experimental results are illustrated in **Figures 4–7**.

From **Figure 4**, IPSO-GWO and PSO simulate the path and iterative convergence curve of the path planning in a 20 × 20 map environment; it can be concluded that the PSO can obtain the optimal path at the 138th iteration with path length 30.36. Whereas the proposed IPSO-GWO can acquire the optimization at the 75th iteration with the obtained path length 28.63.

From **Figure 5**, IPSO and PSO simulate the path and iterative convergence curve of the path planning in a 30 × 30 map environment. The PSO algorithm searches for the optimal path at the 169th iteration, and the obtained path length is 45.36. The proposed IPSO-GWO can find the optimal path in the 86th iteration with acquired path length 42.77.

From **Figure 6**, IPSO, ACO, and GA are used to simulate the path and iterative convergence curve of the path planning in a 20 × 20 map environment. It can be seen that IPSO-GWO can acquire the optimal path length of 28.63 in the 11th iteration, the optimal path length found by ACO in the 62th iteration is 29.21, and the optimal path length found by GA in the 22nd iteration is 29.21. It can be concluded that the proposed IPSO-GWO algorithm converges faster in a 20 × 20 map environment with shortest path acquirement.

From **Figure 7**, IPSO, ACO, and GA are used to simulate the path and iterative convergence curve of the path planning in a 30 × 30 map environment. It can be seen that IPSO-GWO can find the optimal path length of 42.77 in the 86th iteration, the optimal path length found by ACO in the 166th iteration is 42.77, and the optimal path length found by GA in the 33rd iteration is 45.11. It can also be concluded that although GA finds the optimal path faster, the path length is longer, whilst ACO finds the same optimal path as IPSO-GWO, but it is slower than IPSO-GWO. In summary, IPSO-GWO algorithm has the highest performance efficiency.

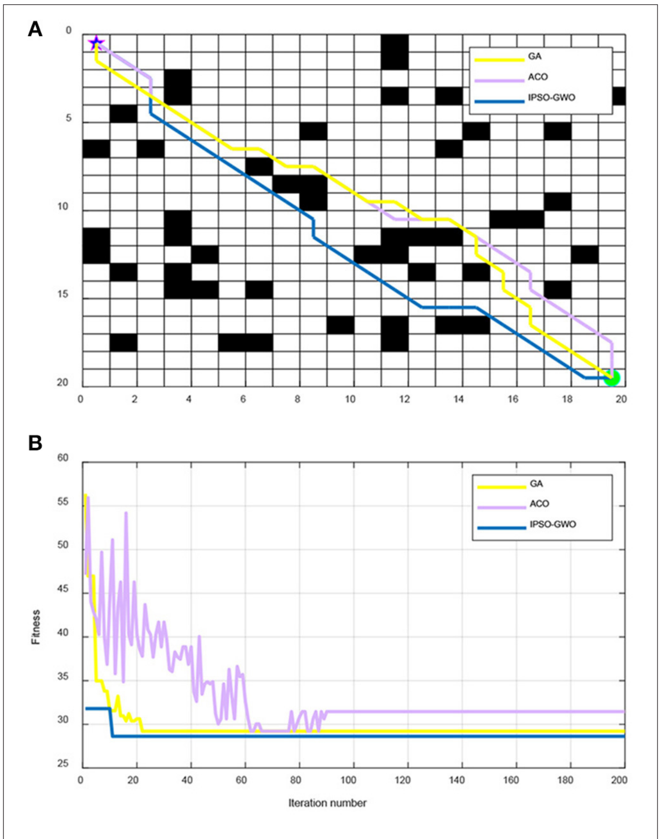


FIGURE 6 | Simulation results of IPSO-GWO, GA, ACO in 20 × 20 map environment. **(A)** Path comparison with GA, ACO, and IPSO-GWO algorithms. **(B)** Iterative curve with GA, ACO, and IPSO-GWO algorithms.

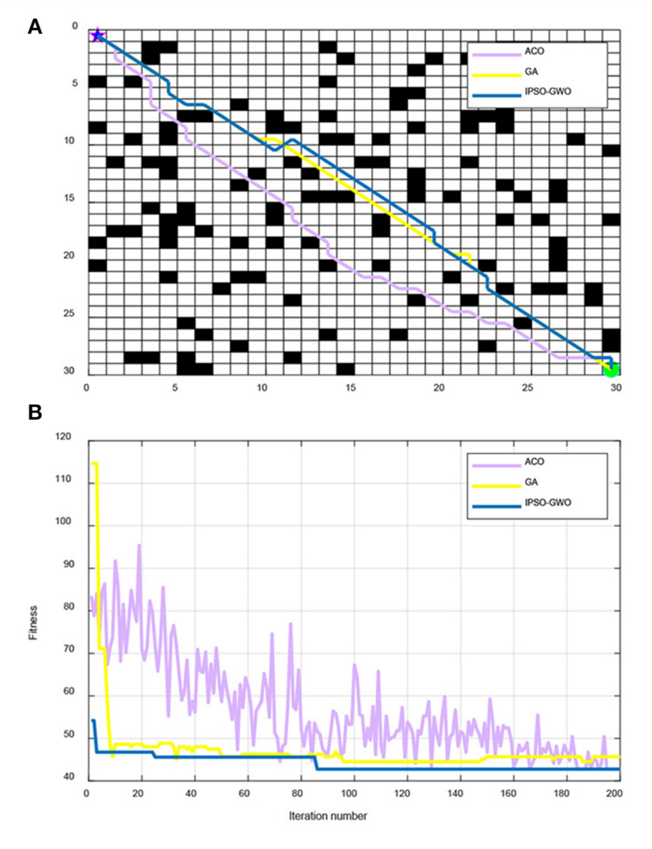


FIGURE 7 | Simulation results of IPSO-GWO, GA, and ACO in 30×30 map environment. **(A)** Path comparison with ACO, GA, and IPSO-GWO algorithms. **(B)** Iterative curves with ACO, GA, and IPSO-GWO algorithms.

CONCLUSION

This paper makes a valuable contribution to the improvement of PSO algorithm in robot path planning in terms of convergence speed and shortest path acquirement. Combining the traditional PSO with the GWO, chaos, and a new adaptive inertia weight, it can address the problem of premature convergence and poor global search ability, and improve the convergence speed for faster path searching. The proposed IPSO-GWO algorithm has been tested against the traditional PSO for ten benchmark functions, and optimization results show that the IPSO-GWO converges faster without premature convergence. Comparing the IPSO-GWO with PSO and two other algorithms for path planning, the IPSO-GWO can find an optimal path with faster speed. In summary, the proposed IPSO-GWO algorithm exhibits higher performance in path planning with the combination of chaos for premature convergence avoidance. In the future, we will continue to apply the proposed IPSO-GWO algorithms in more practical applications.

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

XC and JL: proposed the idea and designed the experiment in this study. CZ and JZ: performed the simulation experiments and analyzed the experiment results and wrote the manuscript. MZ: corrected the manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported by the National Natural Science Foundation of China Program under Grant 62073198, by the Major Research Development Program of Shandong Province of China under Grant 2016GSF117009, and by the China Postdoctoral Science Foundation under Grant 2018M642680.

ACKNOWLEDGMENTS

We acknowledge the support of the College of Electrical Engineering and Automation in Shandong University of Science and Technology and State Grid Dongying District of Dongying City Power Supply Company.

REFERENCES

Azmi, M. Z., and Ito, T. (2020). Artificial potential field with discrete map transformation for feasible indoor path planning. *Appl. Sci.-Basel*. 10:8987. doi: 10.3390/app10248987

Das, P. K., and Jena, P. K. (2020). Multi-robot path planning using improved particle swarm optimization algorithm through novel evolutionary operators. *Appl. Soft Comput*. 92:106312. doi: 10.1016/j.asoc.2020.106312

Demir, F. B., Tuncer, T., and Kocamaz, A. F. (2020). A chaotic optimization method based on logistic-sine map for numerical function optimization. *Neural Comput. Appl*. 32, 14227–14239. doi: 10.1007/s00521-020-04815-9

Gopal, A., Sultani, M. M., and Bansal, J. C. (2020). On stability analysis of particle swarm optimization algorithm. *Arab. J. Sci. Eng*. 45, 2385–2394. doi: 10.1007/s13369-019-03991-8

Gul, F., Rahiman, W., Alhady, S. S. N., Ali, A., Mir, I., and Jalil, A. (2021). Meta-heuristic approach for solving multi-objective path planning for autonomous guided robot using PSO-GWO optimization algorithm with evolutionary programming. *View Web Sci. Res. ORCID* 12, 7873–7890. doi: 10.1007/s12652-020-02514-w

Guo, J., Li, C. Y., and Guo, S. X. (2020). A novel step optimal path planning algorithm for the spherical mobile robot based on fuzzy control. *IEEE Access* 8, 1394–1405. doi: 10.1109/ACCESS.2019.2962074

Guo, Z. L., Zhang, W. S., and Wang, S. W. (2021). Improved gravitational search algorithm based on chaotic local search. *Int. J. Bio-Inspired Comput*. 17, 154–164. doi: 10.1504/IJBIC.2021.114873

Hao, K., Zhao, J. L., Yu, K. C., Li, C., and Wang, C. Q. (2020). Path planning of mobile robots based on a multi-population migration genetic algorithm. *Sensors* 20:5873. doi: 10.3390/s20205873

Kennedy, J., and Eberhart, R. (1995). "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, *Vol. 4,* 1942–1948. doi: 10.1109/ICNN.1995.488968

Krell, E., Sheta, A., Balasubramanian, A. P. R., and King, S. A. (2019). Collision-free autonomous robot navigation in unknown environments utilizing PSO for path planning. *J. Artif. Intell. Soft Comput. Res*. 9, 267–282. doi: 10.2478/jaiscr-2019-0008

Li, G. S., and Chou, W. S. (2018). Path planning for mobile robot using self-adaptive learning particle swarm optimization. *Sci. China-Inf. Sci*. 61:052204. doi: 10.1007/s11432-016-9115-2

Li, K. R., Hu, Q. Q., and Liu, J. P. (2021). Path planning of mobile robot based on improved multiobjective genetic algorithm. *Wireless Commun. Mobile Comput*. 2021:8836615. doi: 10.1155/2021/8836615

Li, L. Y., Wu, D. F., Huang, Y. Q., and Yuan, Z. M. (2021). A path planning strategy unified with a COLREGS collision avoidance function based on deep reinforcement learning and artificial potential field. *Appl. Ocean Res*. 113:102759. 10.1016/j.apor.2021.102759

Li, M., Chen, H., Shi, X., Liu, S., Zhang, M., and Lu, S. F. (2019a). A multi-information fusion "triple variables with iteration" inertia weight PSO algorithm and its application. *Appl. Soft Comput*. 84:105677. doi: 10.1016/j.asoc.2019.105677

Li, M., Chen, H., Wang, X. D., Zhong, N., and Lu, S. F. (2019b). An improved particle swarm optimization algorithm with adaptive inertia weights. *Int. J. Inf. Technol. Decis. Mak*. 3, 833–866. doi: 10.1142/S0219622019500147

Lian, J. F., Yu, W. T., Xiao, K., and Liu, W. R. (2020). Cubic spline interpolation-based robot path planning using a chaotic adaptive particle swarm optimization algorithm. *Math. Prob. Eng*. 2020:1849240. doi: 10.1155/2020/1849240

Lin, G. C., Zhu, L. X., and Li, J. H. (2021). Collision-free path planning for a guava-harvesting robot based on recurrent deep reinforcement learning. *Comput. Electron. Agric*. 188:106350. doi: 10.1016/j.compag.2021.106350

Liu, T. T., Song, C. Y., and Jiang, J. Q. (2019). Robotic path planning based on improved ant colony algorithm. *Lect. Notes Comput. Sci*. 11554, 351–358. doi: 10.1007/978-3-030-22796-8_37

Liu, X. H., Zhang, D. G., Zhang, T., Jie, Z., and Zhang, J. (2021). A new path plan method based on hybrid algorithm of reinforcement learning and particle swarm optimization. *Eng. Comput*. doi: 10.1108/EC-09-2020-0500

Liu, Z. H., Liu, H. B., Lu, Z. G., and Zeng, Q. L. (2021). A dynamic fusion pathfinding algorithm using delaunay triangulation and improved a-star for mobile robots. *IEEE Access* 9, 20602–20621. doi: 10.1109/ACCESS.2021.3055231

Lu, C., Gao, L., Li, X. Y., Hu, C. Y., Yan, X. S., and Gong, W. Y. (2020). Chaotic-based grey wolf optimizer for numerical and engineering optimization problems. *Memetic Comput*. 12, 371–398. doi: 10.1007/s12293-020-00313-6

Narinder, S., and Singh, S. B. (2017). Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance. *J. Appl. Math*. 2017, 1–15. doi: 10.1155/2017/2030489

Ouertani, M. W., Manita, G., and Korbaa, O. (2021). Chaotic lightning search algorithm. *Soft Comput*. 25, 2039–2055. doi: 10.1007/s00500-020-05273-0

Phung, M. D., and Ha, and, Q. P. (2021). Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization. *Appl. Soft Comput*. 107:107376. doi: 10.1016/j.asoc.2021.107376

Shao, S. K., Peng, Y., He, C. L., and Du, Y. (2020). Efficient path planning for UAV formation *via* comprehensively improved particle swarm optimization. *ISA Trans*. 97, 415–430. doi: 10.1016/j.isatra.2019.08.018

Song, B. Y., Wang, Z. D., and Zou, L. (2021). An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve. *Appl. Soft Comput*. 100:106960. doi: 10.1016/j.asoc.2020.106960

Song, J., Hao, C., and Su, J. C. (2020). Path planning for unmanned surface vehicle based on predictive artificial potential field. *Int. J. Adv. Robot. Syst*. 17:1729881420918461. doi: 10.1177/1729881420918461

Sun, X. J., Wang, G. F., Fan, Y. S., Mu, D. D., and Qiu, B. B. (2021). A formation autonomous navigation system for unmanned surface vehicles with distributed control strategy. *IEEE Trans. Intell. Transp. Syst*. 22, 2834–2845. doi: 10.1109/TITS.2020.2976567

Tang, B. W., Kui, X., Pang, M. Y., and Zhu, Z. X. (2020). Multi-robot path planning using an improved self-adaptive particle swarm optimization. *Int. J. Adv. Robot. Syst*. 17:1729881420936154. doi: 10.1177/1729881420936154

Tang, G., Tang, C. Q., Claramunt, C., Hu, X., and Zhou, P. P. (2021). Geometric A-star algorithm: an improved A-star algorithm for AGV path planning in a port environment. *IEEE Access*. 9, 59196–59210. doi: 10.1109/ACCESS.2021.3070054

Teng, Z. J., Lv, J. L., and Guo, L. W. (2019). An improved hybrid grey wolf optimization algorithm. *Soft Comput*. 23, 6617–6631. doi: 10.1007/s00500-018-3310-y

Tullu, A., Emdale, B., Wondosen, A., and Hwang, H. Y. (2021). Machine learning approach to real-time 3D path planning for autonomous navigation of unmanned aerial vehicle. *Appl. Sci.-Basel*. 11:4706. doi: 10.3390/app11104706

Wang, D., Zhang, J., Jin, J. C., Liu, D. Q., and Mao, X. P. (2021). Rapid global path planning algorithm for unmanned surface vehicles in large-scale and multi-island marine environments. *PeerJ. Comput. Sci*. 7:e612. doi: 10.7717/peerj-cs.612

Wang, P. W., Gao, S., Li, L., Sun, B. B., and Cheng, S. (2019). Obstacle avoidance path planning design for autonomous driving vehicles based on an improved artificial potential field algorithm. *Energies*. 12:2342. doi: 10.3390/en12122342

Wang, S. B. (2021). Asymptotic tracking control for nonaffine systems with disturbances. *IEEE Trans. Circuits Syst. II: Exp. Briefs*. 99, 1–1. doi: 10.1109/TCSII.2021.3080524

Wang, S. B., Na, J., and Chen, Q. (2020). Adaptive predefined performance sliding mode control of motor driving systems with disturbances. *IEEE Trans. Energy Conv*. 99, 1–1. doi: 10.1109/TEC.2020.3038010

Wang, Y. B., Bai, P., Liang, X. L., Wang, W. J., Zhang, J. Q., and Fu, Q. X. (2019). Reconnaissance mission conducted by UAV swarms based on distributed PSO path planning algorithms. *IEEE Access* 7, 105086–105099. doi: 10.1109/ACCESS.2019.2932008

Wen, T., Yang, D. C., Liu, W. F., Wen, C. L., and Cai, B. G. (2021). A novel integrated path planning algorithm for warehouse AGVs. *Chin. J. Electron*. 30, 331–338. doi: 10.1049/cje.2021.02.002

Wu, Y., Liu, J. F., and Li, D. F. (2020). Chaotic dynamically dimensioned search algorithm. *IEEE Access* 8, 152474–152499. doi: 10.1109/ACCESS.2020.3017210

Xie, R. L., Meng, Z. J., Wang, L. F., Li, H. C., Wang, K. P., and Wu, Z. (2021). Unmanned aerial vehicle path planning algorithm based on deep reinforcement learning in large-scale and dynamic environments. *IEEE Access* 9, 24884–24900. doi: 10.1109/ACCESS.2021.3057485

Xiong, X. Y., Min, H. T., Yu, Y. B., and Wang, P. Y. (2020). Application improvement of A* algorithm in intelligent vehicle trajectory planning. *Math. Biosci. Eng*. 18, 1–21. doi: 10.3934/mbe.2021001

Xu, J., Song, K. C., Zhang, D. F., Dong, H. W., Yan, Y. H., and Meng, Q. G. (2021). Informed anytime fast marching tree for asymptotically optimal motion planning. *IEEE Trans. Indus. Electron*. 68, 5068–5077. doi: 10.1109/TIE.2020.2992978

Yao, Q. F., Zheng, Z. Y., Qi, L., Yuan, H. T., Guo, X. W., Zhao, M., et al. (2020). Path planning method with improved artificial potential field-a reinforcement learning perspective. *IEEE Access* 8, 135513–135523. doi: 10.1109/ACCESS.2020.3011211

Ye, X., Luo, L., Hou, L., Duan, Y., and Wu, Y. (2020). Laser ablation manipulator coverage path planning method based on an improved ant colony algorithm. *Appl. Sci.-Basel* 10:8641. doi: 10.3390/app10238641

Zhang, D. H., You, X. M., Liu, S., and Pan, H. (2020). Dynamic multi-role adaptive collaborative ant colony optimization for robot path planning. *IEEE Access* 8, 129958–129974. doi: 10.1109/ACCESS.2020.3009399

Zhang, J., Sheng, J. N., Lu, J. W., and Shen, L. (2021). UCPSO: a uniform initialized particle swarm optimization algorithm with cosine inertia weight. *Comput. Intell. Neurosci*. 2021:8819333. doi: 10.1155/2021/8819333

Zhi, Y. H., and Jiang, Y. (2020). Design of basketball robot based on behavior-based fuzzy control. *Int. J. Adv. Robot. Syst*. 17:1729881420909965. doi: 10.1177/1729881420909965

Zhu, S. N., Zhu, W. Y., Zhang, X. Q., and Cao, T. (2020). Path planning of lunar robot based on dynamic adaptive ant colony algorithm and obstacle avoidance. *Int. J. Adv. Robot. Syst*. 17:1729881419898979. doi: 10.1177/1729881419898979

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.

frontiers
in Neurorobotics

ORIGINAL RESEARCH
published: 26 November 2021
doi: 10.3389/fnbot.2021.751037

Short Circuit Recognition for Metal Electrorefining Using an Improved Faster R-CNN With Synthetic Infrared Images

Xin Li, Yonggang Li, Renchao Wu, Can Zhou and Hongqiu Zhu*

School of Automation, Central South University, Changsha, China

This paper is concerned with the problem of short circuit detection in infrared image for metal electrorefining with an improved Faster Region-based Convolutional Neural Network (Faster R-CNN). To address the problem of insufficient label data, a framework for automatically generating labeled infrared images is proposed. After discussing factors that affect sample diversity, background, object shape, and gray scale distribution are established as three key variables for synthesis. Raw infrared images without fault are used as backgrounds. By simulating the other two key variables on the background, different classes of objects are synthesized. To improve the detection rate of small scale targets, an attention module is introduced in the network to fuse the semantic segment results of U-Net and the synthetic dataset. In this way, the Faster R-CNN can obtain rich representation ability about small scale object on the infrared images. Strategies of parameter tuning and transfer learning are also applied to improve the detection precision. The detection system trains on only synthetic dataset and tests on actual images. Extensive experiments on different infrared datasets demonstrate the effectiveness of the synthetic methods. The synthetically trained network obtains a mAP of 0.826, and the recall rate of small latent short circuit is superior to that of Faster R-CNN and U-Net, effectively avoiding short-circuit missed detection.

Keywords: sample synthesis, short circuit detection, infrared image, metal electrorefining, attention-based Faster R-CNN

OPEN ACCESS

Edited by:
Lianghong Wu,
Hunan University of Science and Technology, China

Reviewed by:
Zhichao Lian,
Nanjing University of Science and Technology, China
Sebastian Herzog,
Max Planck Society (MPG), Germany

****Correspondence:***
Can Zhou
zhoucan@csu.edu.cn

Received: *31 July 2021*
Accepted: *29 October 2021*
Published: *26 November 2021*

Citation:
Li X, Li Y, Wu R, Zhou C and Zhu H (2021) Short Circuit Recognition for Metal Electrorefining Using an Improved Faster R-CNN With Synthetic Infrared Images. Front. Neurorobot. 15:751037. doi: 10.3389/fnbot.2021.751037

1. INTRODUCTION

In the metal electrorefining process, short circuits between electrodes cause the temperature of the electrodes to rise, the electrochemical reaction to stop, and the further reduction of the electrolytic efficiency (Aqueveque et al., 2009). Infrared thermography technology has been become a promising method to detect short-circuit electrodes due to its visualization of heat distribution, non-invasive nature, and large-scale monitoring (Maekipaeae et al., 1997; Hong and Wang, 2017). But recognizing short-circuit electrodes from infrared images is still a challenge because of the occlusion above the electrolytic cell group and the complex heat conduction. Temperature of the fault electrode increases, but the canvas on the cell surface may hide the abnormal heat, leading to missed detection. The complex heat conduction between the canvas and the electrodes will interrupt and spread the short-circuit temperature distribution, which will deform the shape of the electrode in infrared image, resulting in inaccurate detection results. In addition, the inherent low resolution of infrared image degrades the detailed features of objects, making it difficult for the electrodes to distinguish and recognize (Xiao et al., 2017; Xing et al., 2019).

With the rise of deep learning, the Convolutional Neural Networks (CNN) have efficiently solved a number of object detection problems by learning more discriminative features (Peng and Chen, 2015; Hiary et al., 2018). CNN based object detection methods are classified into two classes: one-stage detectors like OverFeat (Sermanet et al., 2013), SSD (Liu et al., 2016), and U-Net (Ronneberger, 2017), perform specific classification immediately after feature extract. On the other hand, two-stage detectors such as R-CNN (Girshick et al., 2014), Faster R-CNN (Ren et al., 2017) generate region proposals with low-level cues first and then use the proposals to target existing judgments prior to the classification (Ce et al., 2018). Among recent deep learning methods, Faster R-CNN shows excellent detection performance as Faster R-CNN can capture more pixel-wise annotation information about objects. Also, infrared images usually present poor resolution, low contrast, and fuzzy visual effect, objects in infrared images tend to appear as a series of rough, indistinct areas which closely related to the surrounding complex background. A specific feature and the surrounding area in infrared image both affect the accuracy of target detection. Therefore, in this paper, we use Faster R-CNN model to perform infrared target recognition. However, two challenges are proposed for the task. First, the training of a detection network requires a large number of labeled images, which are labeled with the category and location information of objects in detail (Samadi et al., 2019). Usually such a dataset is collected and annotated manually. It requires the annotator to have expert knowledge of the task and capture the distribution of variables contributing to the varied representation of real world conditions. The laborious work is the first difficulty in deploying CNNs on practical applications. Second, Scale variance enforced by resampling operation in Faster R-CNN may result in information loss, which is even worse for small-scale infrared targets, leading to missed detection, such as the latent short circuit in our problem. Compared to visible images, the features of infrared images degrade, and the convolution layer needs to learn more meaningful features. A complete dataset and a well-designed network are a worthy pursuit.

In order to overcome the difficulty of manually annotating a sufficient number of images and meeting the accuracy requirements of short circuit recognition tasks, we first propose a framework for automatically generating labeled images, and then design an attention-based Faster R-CNN for short circuit detection. In the image synthesis process, we classify short-circuit objects into two categories: obvious short circuit and latent short circuit. Background, target gray scale distribution, and shape are proposed as three key variables after a series of reasonable assumptions and analyses. We simulate the three key variables with different methods to satisfy the diversity of the intra-class of samples. Single cells infrared images that without short circuits are collected as backgrounds; rectangles of random sizes and aspect ratios are exploited to simulate electrode targets; external illumination template and local signal-to-clutter ration (SCR) constraint method are introduced to simulate the multiple manifestations of objects; object locations and class labels are automatically annotated. Then, to increase the detection accurancy, our detection scheme improves the Faster R-CNN by introducing an attention module. This module fuses the semantic segment information of small-scale latent short circuits and the synthetic dataset, making the network focus on small objects during the extraction of features. Combined with anchor parameters fine-tuning and transfer learning strategy the attention-based Faster R-CNN can better avoid latent short circuit missed detection. Summarizing the above discussion, we aim at addressing the problem of automatic synthesis of labeled infrared images and apply it to the training of the short circuit recognition system. The contributions of the paper can be summarized as the following two aspects:

1) Propose an automatic sample synthesis method that can generate a sufficient number of labeled infrared images.
2) Improve the Faster R-CNN by introducing an attention module and design the short circuit recognition system for metal electrolysis, the system is trained only on synthetic samples and generalizes well to real images especially for the latent short circuit class.

The remainder of the paper is organized as follows. Section 2 provides related works about sample number increase method-data augmentation and background knowledge about metal electrorefining. Synthesis difficulties are also discussed in this section. Section 3 presents the details of our synthesis method and the short circuit recognition system based on attention-based Faster R-CNN. Section 4 provides three experiments to evaluate the synthesis method and demonstrate the effectiveness of the system. Finally, the conclusion of our research is presented in Section 5.

2. RELATED WORK

2.1. Methods About Data Augmentation

Various data augmentation methods have been studied to create additional training data. Generative adversarial networks (GANs) and its variations (Goodfellow et al., 2014; Odena et al., 2016; Zhang et al., 2016) show promising results for highly realistic image generation. The GANs-based method generates images by simultaneously training two models: generative model and discriminative model. But balancing the two models is a difficult task (Ngxande et al., 2019). Another method achieves data augmentation by combining multiple image transformation operations on an existing data set while preserving class labels (Ratner et al., 2017; Gao et al., 2020). For example, flipping, cropping, and color casting are applied to increase the number of marine organism images (Huang et al., 2019). The method mainly imitates the variable elements of the scene that contribute to the samples diversity, such as ambient illumination, target perspective and scale, etc. However, the above augmentations rely on a certain amount of images that have already been labeled. For some specific application scenarios in which the images are difficult to obtain, a third-part public dataset can be exploited, because the content of the dataset has similar features with the application scenario. The optical remote sensing images slices

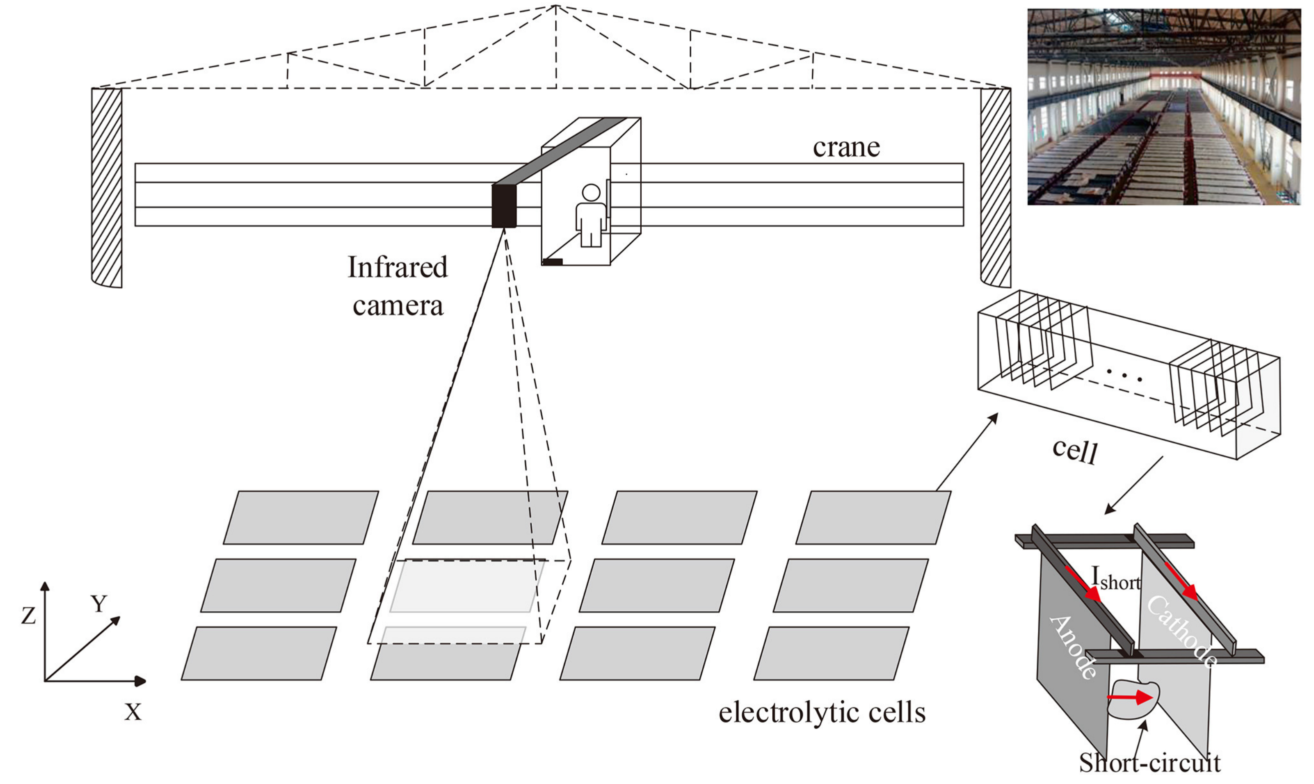


FIGURE 1 | Infrared camera above electrolytic cell groups and the structure of a cell.

of Google Earth are employed to train a deep model, and then the model is applied for ship detection (Jiang et al., 2019). The handwritten images of MNIST are used to simulate long-rang infrared images, in which dim targets are against background clutters (Fan et al., 2018). Nevertheless, a common dataset whose content is similar to the problem scenario is infrequent.

Actually, realistic-looking is not strictly necessary to train a discriminant algorithm. Key variables of the scene are drive, the diversity of sample, and effectiveness in training a neural network (Mayer et al., 2018). In an image, these key variables are usually related to the visual complexity of the scene and multiple manifestations of objects. Different geometric shapes are adopted to generate targets for machine learning (Silva et al., 2019). Irrelevant pictures are taken as complex backgrounds, and on the backgrounds vehicle license plates are synthesized, then the synthesis images are used for identification training (Björklund et al., 2019). Traditional data augmentation methods may be heavyweight and more expensive in metal electrolysis due to the complexity and uniqueness of infrared scenario. The above research provides valuable references and meaningful inspirations for our problem. The complexity comes from the heat conduction, and the uniqueness is because all cells have a similar structure. Identifying variables about the background and the targets of metal electrolysis infrared image makes sample synthesis possible.

2.2. Background of Metal Electrorefining

In our problem, take copper electrotrfining as example, the infrared imager is installed on the crane above the electrolytic cell groups to monitor the temperature distribution of electrodes in cells (**Figure 1**). In each cell, hundreds of electrode plates are parallel immersed into high temperature electrolytes. To prevent electrolyte evaporation, the cell surface is covered with a canvas. In the electrolytic process, the anode is dissolved into metal ions, and then the ions are crystallized out at the cathode plate. Due to the impurities, additive dosing problems, particulates in electrolyte, temperature control, etc. dendrites or nodules growing out from cathode surfaces until they reach anode plates, short circuits occur. The temperature of the short-circuit electrode is obviously higher than that of other electrodes, but the high-temperature electrodes manifest in various forms due to the shielding of the cover. **Figure 2** shows an actual infrared image with different types of short circuit and other components of a cell. Other metal electrolysis processes (lead, nickel, etc.) have similar infrared images.

Complexity and randomness of gray scale distribution and the diversity of short-circuit targets pose challenges for infrared image synthesis. Different from realistic visible image synthesis for which the threshold changes of visibility, color appearance, etc. are the most important (Ferwerda et al., 1996), infrared image is the visual result of the thermal distribution of a scene, and

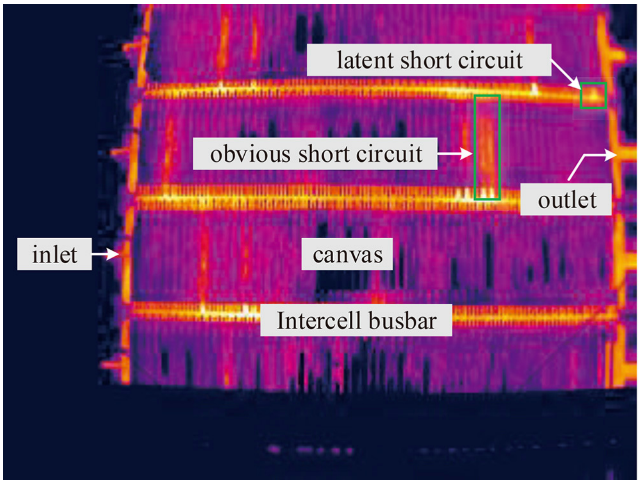


FIGURE 2 | Infrared image containing multiple copper electrolytic cells and different classes of short circuit.

it lacks color information. So the gray scale used to reflect the temperature is significant for synthesis.

With canvas, the heat environment in each electrolytic cell is independent, and heat transfer among components of a cell is complex. Thus, the gray distributions of every cell surface are different, although the structure and size of these cells are the same. Also the airflow above cell surfaces adds random attributes to the already complicated background.

Different kinds of short-circuit targets have various manifestations on images, like the short circuits marked in **Figure 2**. Some short circuits present obvious high gray intensity distribution on the canvas region, and the gray scale distributions are usually not uniform, but they show some common features: the silhouette of the electrode is hazy, the outline appears as a rectangle, and the gray intensity of the positive terminal is obviously higher than other parts. While for some other short circuits, there is no obvious gray scale change on the canvas area, only the end of the electrode which is located on the intercell busbar exhibits high gray intensity. This type of short circuit appears as a small area with uniform gray scale distribution on busbar.

Based on the above analysis, the background, shape, and gray scale distribution patterns are established as three key variables, that are responsible for the large inter-class variability of infrared images. Short-circuits targets are classified into two classes: obvious short circuit and latent short circuit. To determine the positive and negative terminals, we also take the electrolyte inlet and the outlet as another two classes of detection targets that will be annotated on the image (**Table 1**).

3. METHOD

The research route of this work is shown as the flow chart in **Figure 3**. The collected infrared images are firstly corrected for barrel distortion and segmented into individual cells. Then cell images without short circuit are used as background, and we synthesize and label short-circuit targets on it. At last the synthetic image dataset is used for training an improved faster R-CNN detection network to recognize different classes of short circuits, and the network is tested on real world infrared images. Our work mainly focuses on modules of preprocessing, sample images synthesis, and detection network improvement.

TABLE 1 | Target class labeled in the synthesis images.

Class number	Target class
1	obvious short circuit
2	latent short circuit
3	inlet
4	outlet

3.1. Preprocessing of the Infrared Images

Infrared electrolytic cells images suffer from barrel distortion due to the use of the wide-angle lens. So barrel distortion correction is first carried out to facilitate the acquisition of the image patch of a single electrolytic cell.

Distortion occurs because of the inconsistent transmittance of the lens. The refractive index at the edge of the lens is greater than that at the center of the lens. Thus, the same object looks smaller in the outer region of the image than in the central region because the outer region is more compressed than the region near the distorted center. Assume that the distortion rate is radial about the distortion center (Asari et al., 1999), through mapping pixels of the distorted image onto a corrected image, we can obtain the corrected pixel coordinates.

(x_c, y_c) represents the center of the distorted image, and (x, y) are coordinates of any pixel. Radius r and the angle θ of a vector from the distortion center to (x, y) are given by:

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2} \tag{1a}$$

$$\theta = arctan(\frac{y - y_c}{x - x_c}) \tag{1b}$$

The pixel location (x, y) in the distorted image can be transformed to a new location (x_{new}, y_{new}) in the corrected image. The corresponding radius r_{new} and angle θ_{new} of the vector from the corrected center (x_{nc}, y_{nc}) to (x_{new}, y_{new}) can be computed as:

$$r_{new} = \sqrt{(x_{new} - x_{nc})^2 + (y_{new} - y_{nc})^2} \tag{2a}$$

$$\theta_{new} = \theta \tag{2b}$$

The mapping relation between radius r_{new} and r is defined with a polynomial as:

$$r_{new} = \sum_{i=0}^{n} k_i r^i \tag{3}$$

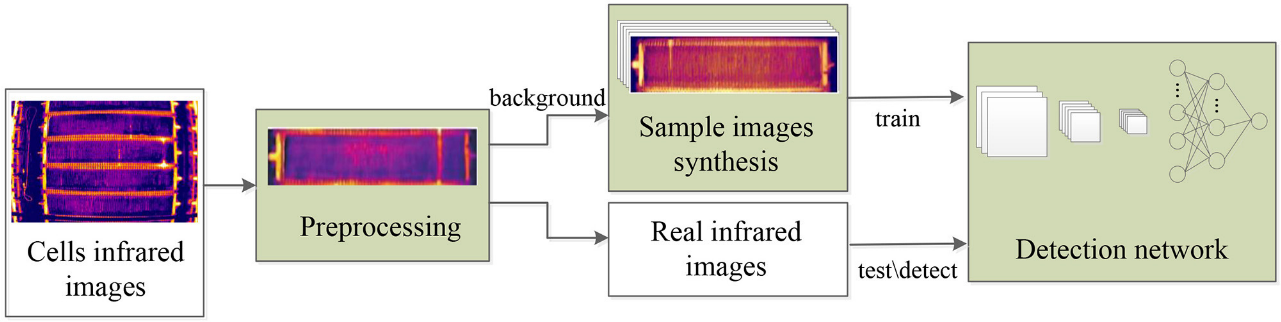


FIGURE 3 | Research route of the work.

TABLE 2 | Distortion coefficients estimated by fitting pixels on cells border.

Distortion coefficient	k_0	k_1	k_2
Value	1	2.355×10^{-4}	2.285×10^{-6}

Where n is the number of polynomial terms. k_i denotes the distortion coefficient. The effect of higher order terms can be ignored, because the distortion rate is very small, so the quadratic mapping relationship is adopted. The distortion coefficients can be estimated by fitting the pixel coordinates of the cell boundary in the distorted image In our work, the estimated distortion coefficients are as shown in **Table 2**.

Then, coordinates of the new location (x_{new}, y_{new}) can be obtained by:

$$x_{new} = x_{nc} + r_{new} cos\theta \tag{4a}$$

$$y_{new} = y_{nc} + r_{new} sin\theta \tag{4b}$$

The method corrects the distortion by shifting the pixels. Moving pixels causes vacancy in the original pixel position and thus form a grid of blank pixels on the corrected image. For ease of viewing, we use a black grid in **Figure 4B**. A bilinear interpolation method is employed to fill these vacant pixels. The final result of distortion correction is as shown in **Figure 4C**. With the corrected image, we can easily obtain image patches of single cells like in **Figure 5**.

3.2. Sample Images Synthesis

3.2.1. Simulate Background

The background is difficult to simulate because of its complex and random gray scale distribution. That is due to the complex electrochemical reactions in the cell and the heat conduction between the canvas, electrode, and electrolyte. It is unique for the metal electrorefining scene, the alternative of using other backgrounds fails here (Björklund et al., 2019).

The diversity of the background has a serious influence on target recognition. The complexity and randomness of the gray scale distribution of cells without short circuits gives the image diversity, and such cells can be used as backgrounds. After barrel distortion correction, we can easily obtain any number of single cell images with the same structure. Therefore, we collect enough images of single cells without short circuits as backgrounds (**Figure 6**) to satisfy the diversity of backgrounds. On these backgrounds, we further synthesize targets.

3.2.2. Simulate Shape

We chose a rectangle to simulate the shape of the electrode. For the obvious short circuit class, although the rod is invisible, the corresponding area on canvas is a strip of high gray intensity. The strip starts from the busbar area and has a larger aspect ratio. For the latent short circuit class, the high intensity area is small and with a small aspect ratio that is approximately 1:1. The latent short circuit is contained within the intercell busbar area. Hence, rectangles with different aspect ratios are used to simulate the shape variable of short-circuit targets.

Electrodes width is calculated by geometric method, and the width range on the image is $[4, 13]$ pixels. Similarly, the length range is $[4, 65]$ pixels. The length and width of the rectangle are randomly selected in the two intervals to construct a rectangle to simulate a short circuit. We set that when the aspect ratio is greater than 1.5, the rectangle is an image patch for obvious short circuit; when the aspect ratio is less than 1.5, the rectangle is an image patch for latent short circuit.

The location constraint for each target is that the coordinates of the upper-left corner are located within the scope of busbar region. So the intercell busbar needs to be located on the background image first. This can be implemented with a gray scale threshold. The busbar region of **Figure 7B** is as shown in **Figure 7A**. In the busbar region, a pixel position is randomly selected, and then the image patch of the target is determined by using the randomly selected size values. It means that the label information (class, location) of a synthetic target is deterministic.

The location information of a target label is expressed as $Patch = [x_p, y_p, w, l]$. (x_p, y_p), that is the upper-left coordinate. (w, l) are the width and length of the image patch. The pixel values of the patches in the raw background image are temporarily reserved for the next gray scale distribution simulation. Two example patches are as shown in **Figure 7B**.

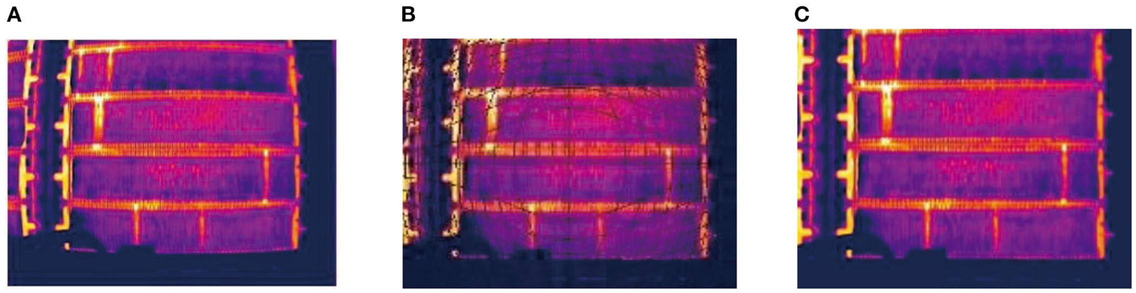


FIGURE 4 | Process of barrel distortion correction: **(A)** A barrel distorted IR image. **(B)** Vacant pixel grid produced by pixel radius mapping. **(C)** The final corrected image after bilinear interpolation.

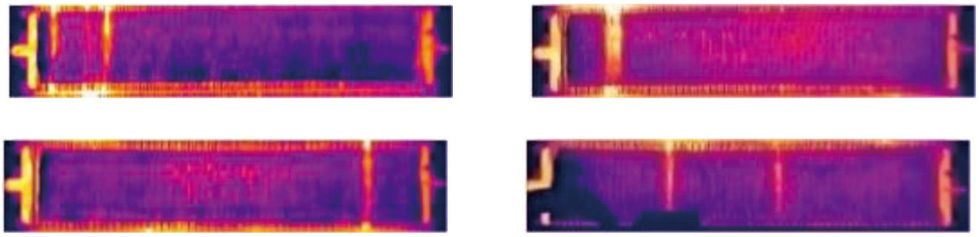


FIGURE 5 | Four single electrolytic cells cropped form the barrel distortion corrected image.

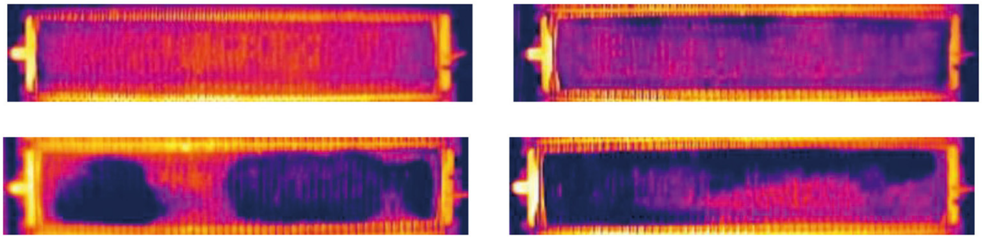


FIGURE 6 | Single cell images without short circuit used as background.

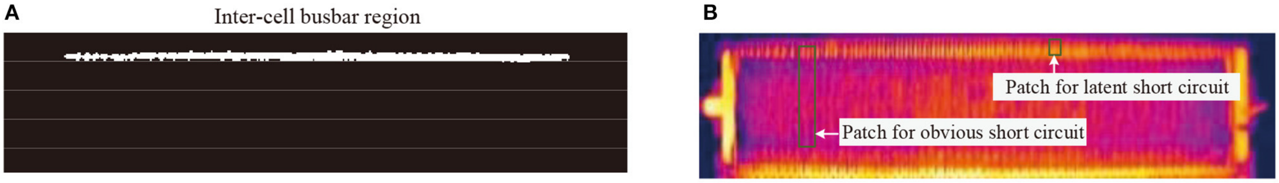

A
Inter-cell busbar region
B
Patch for latent short circuit
Patch for obvious short circuit

FIGURE 7 | (A) Intercell busbar region identified by gray threshold on background image. **(B)** Two example patches for latent short circuit class and obvious short circuit class, the upper-left coordinates of the patches are located in the intercell busbar region.

3.2.3. Simulate Gray Scale Distribution

After, the target patches are determined through assigning gray scale intensity for these patches to simulate the gray scale distribution. Short circuit gray scale intensity is usually higher than the surrounding area, but its distribution is characterized by complexity and diversity. Moreover, the two classes of targets are against different backgrounds. The gray scale continuity of the synthesized target and background should be considered. Therefore, we adopt two different gray scale assignment methods to simulate gray distribution for the two target classes.

For the latent short circuit class, the target should be located in the busbar background with high gray scale intensity and small scale area. The gray scale distribution is smooth. These features make it difficult to distinguish between a latent short circuit and

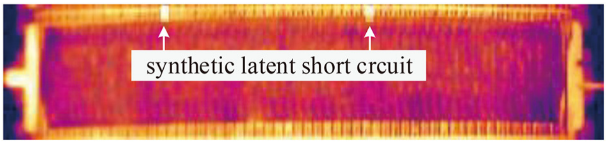


FIGURE 8 | Examples of synthetic latent short circuit instance.

a background. This difficulty can be quantitatively analyzed with SCR (Wang et al., 2017). SCR is a measure of target detectability, and the calculation formula is:

$$SCR = \frac{|\mu_T - \mu_B|}{\sigma_c} \tag{5}$$

Where, μ_T and μ_B represent the average intensity of the target and the background, respectively. σ_c is the standard deviation of the background. $|\mu_T - \mu_B|$ is used to evaluate the gray difference between target and background.

The gray level of the target is affected by the surrounding area, and the influence decreases with the increase of distance. So local background (Chen et al., 2013) is more suitable for infrared target simulation. We set the local background area to three times the target area. With a definite SCR value, we can calculate the μ_T of the target. Define a gray scale enhancement factor k as:

$$k = \mu_T / Ave_o \tag{6}$$

Where, Ave_o is the average intensity of the image patch for latent short circuit that originated from the background image. K is multiplied by the pixels in the image patch, the gray value of the image patch increases, and a latent short circuit with a certain SCR value can be obtained. To increase the diversity of samples, the local SCR is randomly selected from an interval $[1.5, 8]$ according the research of Kim et al. (2012). This method generates image patch from the infrared background image and preserves the randomness of its gray distribution. Two generated latent short circuit instances are shown in **Figure 8**.

For the obvious short circuit class, the gray scale distribution is uneven. Affected by random factors, the gray scale distributions are varied. But all the obvious short circuits have one thing in common, that is, there is at least one high gray scale spot in the gray scale distribution. An analogy between the spot and external light source is introduced (Huang et al., 2019). Therefore, a high-intensity spot in the obvious short-circuit area can be regarded as an external light source, affecting adjacent areas. Through adding an additional light source to the image patch for obvious short circuit, we can simulate the diversity of gray scale distribution.

The templates of light source are collected from real infrared images. Some of them are shown in **Figure 9**. A light source template O is randomly selected, then smooth the template with a mean filter as Eq. (7), a gray scale distribution template E can be obtained.

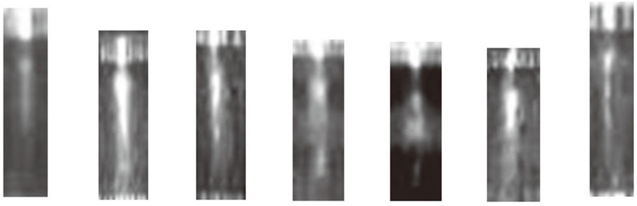


FIGURE 9 | Templates of external light source for obvious short circuit class.

$$E = O * X = \frac{1}{MN}\sum_i \sum_j O(x-i, y-j)X(i,j) \tag{7}$$

Where, X is a mean filter, the height M and the width N of X are set 3, and $*$ means the convolution operation.

The size of the synthetic patch for obvious short circuit P has been known. Resizing the template E to the same size of P, we obtained E_1. We calculated the average gray value Ave_{E_1} of E_1. Then by Eq. (8), the gray difference template D was obtained. Through adding pixels (Eq. (9)), the image patch P and gray difference template D were fused to generate a unique gray distribution for obvious short circuit.

$$D = E_1 - Ave_{E_1} \tag{8}$$

$$P_1 = P + D \tag{9}$$

The above operations strengthened the edge information of the synthetic target. Continuity between the synthesized target and its surrounding area should be maintained in image. Therefore, before adding an external light source, we first reduced the pixel value of P. In contrast to the simulation method of latent short circuit gray scale distribution, a smaller SCR value was used to reduce the gray scale. $[0.1, 0.3]$ is an appropriate range obtained by experience. **Figure 10** shows the synthesis process of obvious short circuit. The referenced light template is **Figure 10A**. The synthetic obvious short circuit is generated on the background in **Figure 10D**. **Figure 10E** shows some other synthetic images that contain both obvious short circuit and latent short circuit in pseudo color image.

The class of inlet and outlet can be annotated automatically through threshold segmentation or edge detection of the background image, that is not the focus of the work and we will not describe it in this work.

3.3. Short Circuit Recognition for Mental Electrorefining With Attention-Based Faster R-CNN

In this section, we first explain the core components of the Faster R-CNN in brief. Then the details of the proposed short circuit detection system with attention-based Faster R-CNN is described. Finally, the strategies of anchor parameters fine-tuning and transfer learning in the system is explained.

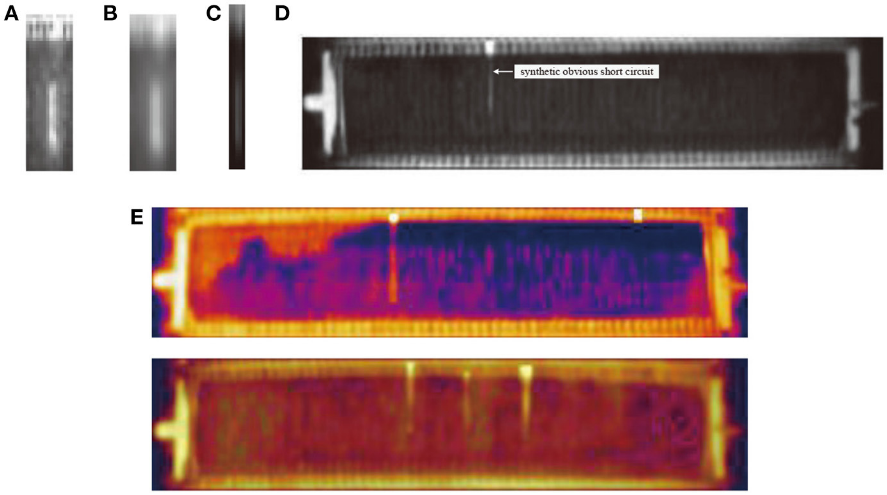


FIGURE 10 | Synthesis process of obvious short circuit class: **(A)** An external light source template. **(B)** Gray scale distribution template after average filtering. **(C)** Difference in template after being resized. **(D)** Synthetic obvious short circuit instance based on the external light source template. **(E)** Examples of synthetic infrared image with obvious short circuit instance and latent short circuit instance.

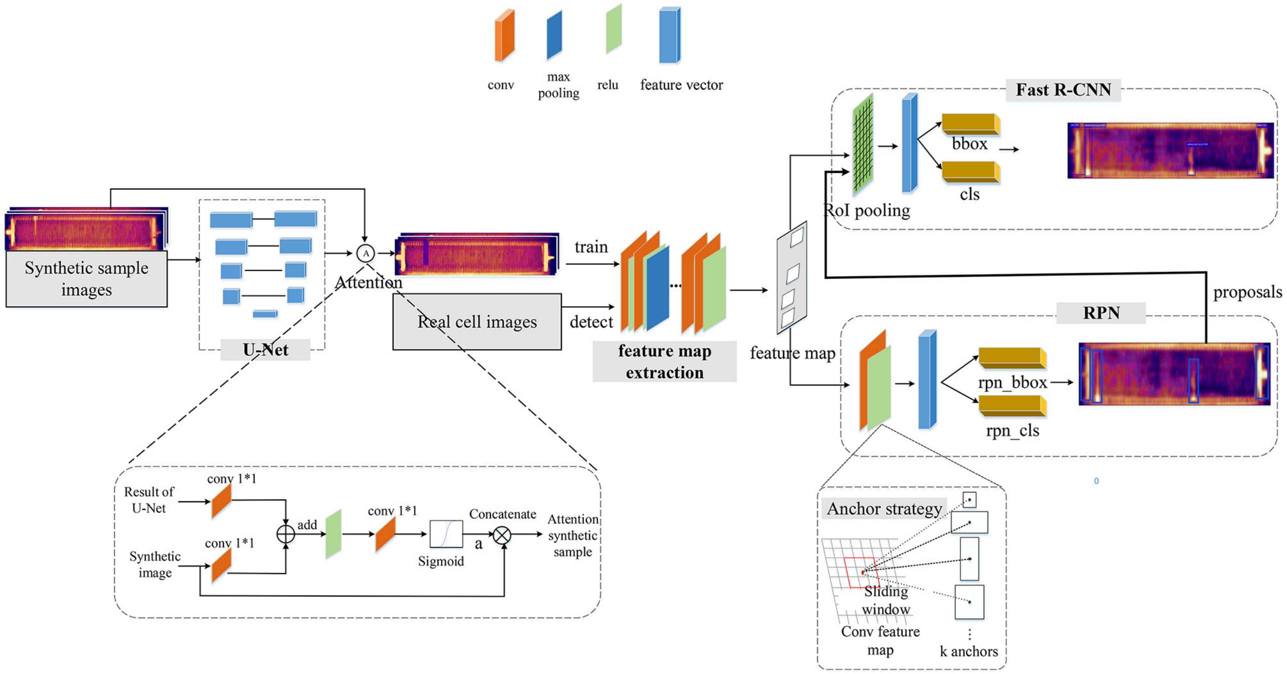


FIGURE 11 | The flowchart of our proposed short-circuit electrode recognition system with attention-based Faster R-CNN.

3.3.1. Architecture of Faster R-CNN

The Faster R-CNN firstly traverses the feature map of infrared image to distinguish objects from the background irrespective of class, accompanied by bounding box regression to generate region proposals of variable sizes. Then, region proposals are resampled to a fixed-sized box to ensure scale-invariance for categorization. Faster R-CNN consists of three functional components (**Figure 11**): feature map extraction, region proposal network (RPN) and Fast R-CNN. Convolutional network VGG-16 model (Simonyan and Zisserman, 2014) is used as a backbone to extract feature maps. The extracted feature maps are shared by the RPN and Fast R-CNN. RPN utilize multiscale anchors boxes strategy to generate region proposals from the feature maps. Firstly, the feature maps convolued with a 3 × 3 slide window. Then, on the resulting feature map, nine anchors boxes with 3 basic scale $(128^2, 256^2, 512^2)$ and 3 aspect rations

$(0.5, 1, 2)$ are simultaneously generated at each pixel. Feature maps of these anchor boxes are mapped to feature vectors by two $1 * 1$ convolutional kernels, and these feature vectors are used to perform preliminary regression and classification through a fully connected layer. The full connection layers preliminary judge whether there is a target in the anchor box and compute the coordinates of these bounding box. Positive anchor boxes are as recommended region proposals from the input feature maps. Region proposals and the feature maps obtained from convolution layers are fed into Fast R-CNN to perform object classification and bounding box regression.

Although the Faster R-CNN significantly improves detection performance and reduces calculation time through region proposals, it is difficult to detect the small scale targets, like latent short circuits in the infrared images. Because the infrared images lack detailed information, features of small scale latent short circuit are often lost in the sampling process of convolutional networks, which leads to missed detection of the latent short circuit. Compared with false detection, missed detection of the short circuit fault brings greater economic loss and security threat to the electrolytic process.

3.3.2. Attention-Based Faster R-CNN for Small Latent Short Circuit

We introduced attention mechanisms for small scale infrared object detection by combining U-Net and Faster R-CNN. Precisely, the attention module integrate the semantic information of U-Net and the synthetic infrared images to focus on small targets. The architecture is shown in **Figure 11**. U-Net has a good performance on pixel-wise predictions of small scale objects, so the net is adopted to handle the synthetic images first. U-Net's encoding features and decoding results are skip connected at different scales, realizing pyramid feature fusion and enriching the learned semantic features of small objects. And up sampling restores the edge information of feature image. Thus, the segmentation result of U-Net and the corresponding synthetic images are used as the two inputs of the attention mechanism. The attention module is also given in **Figure 11**, result of U-Net and synthetic image perform pixel added after 1×1 convolution. The result is activated by ReLu and then goes through sigmoid module, the attention coefficient α is obtained. The range of α is $[0, 1]$, if the coefficient is close to 1, the pixel is related to the target characteristics. By concatenating the attention coefficient and the synthetic images, we fuse the semantic information of small objects to the infrared synthetic images. These fused label infrared images then are fed into the Faster R-CNN as the second step of our training procedure.

3.3.3. Anchor Parameters Fine-Tuning for Small Latent Short Circuit

Note that the purpose of using a set of artificially anchored boxes in RPN is to deal with different scales and aspect ratios of objects (Ren et al., 2017). In the original Faster R-CNN, 9 anchor boxes with 3 scales and 3 ratios were used by default. The default basic scales are $(128^2, 256^2, 512^2)$ and the default aspect rations are $(0.5, 1, 2)$. A region proposal is identified by comparing the intersection-over-Union (IoU) overlap of each anchor box with a ground-truth target. The anchor boxes with a high IoU or which satisfy a criterion are assigned as positive.

Such parameter settings may be more applicable to an image datasets, in which targets come in relative large sizes and similar aspect ratios. However, in our application, the sizes of the four kinds of targets are relatively small and the aspect ratios vary greatly. The aspect ratios of latent short circuits are approximate to 1.5, while the aspect ratios of obvious short circuits are usually larger than 2. If the size of the anchor boxes is far larger than that of the ground-truth target, it may lead to any anchor box which can not meet the IoU requirements. Thus, there will be no regional proposal. Furthermore, it causes small-sized targets to fail detection. If the aspect ratio parameter of the anchor boxes are not set properly, the anchor boxes cannot reflect the target shape better, which will also affect the detection accuracy (Sun et al., 2018).

We tuned anchor parameters according to the actual target size. We increase the number of anchors from 9 to 15 by expanding the aspect ratio and reducing the basic size of anchors. The anchors sizes are $(8^2, 16^2, 32^2)$, and five aspect ratios $(0.5, 1, 2, 4, 5)$ are setted. In our problem, this improvement is mainly used to avoid missed detection of the latent short circuit class.

3.3.4. Transfer Learning for Infrared Dataset

Infrared images contain less information compared to visible images, like information about color and texture. Convolutional layers are the most important part to extract feature information by multiple convolutional kernels. Concretely speaking, the shallower convolutional layers can extract lower-level features like edges and hot spots, but deeper layers can extract semantics information that are more important for object recognition. So a transfer learning strategy is adopted. The shallow layers weights of the pre-training model VGG-16 were frozen, and the deep layers weights were retrained. The number of frozen layers was obtained by comparing the results of multiple training. Architecture and the parameter settings of the shared convolutional layers of the VGG-16 are illustrated in **Table 3**, and the architecture of the RPN and Fast R-CNN are shown in **Tables 4**, **5**, respectively, the content in brackets is the input of the network.

4. EXPERIMENT

In this section, evaluation metrics are introduced first. Then we conduct three experiments. In the first experiment of section 4.2, with a variable-controlling approach we study how each variable of synthesis affects the detection performance of the neural network and verify the effectiveness of the proposed sample synthesis method. In the second experiment of section 4.3, different synthesis methods are compared. Finally, the comparison of original Faster R-CNN (Ori-Faster R-CNN) v.s. Faster R-CNN with anchor parameter fine-tuning (Fin-Faster R-CNN) v.s. Attention-based Faster R-CNN (Att-Faster R-CNN) v.s. U-Net is in section 4.4, also, the short circuit detection result are showed. All the experiments were trained on synthetic samples and texted on actual images.

TABLE 3 | Transfer learning settings in the feature extractor in **Figure 11**.

Layer type	Filters	Size of kernel	Parameter setting
Input image			
1-1st Conv	64	3 × 3 × 3	Frozen
1-2nd Conv	64	3 × 3 × 64	Frozen
Max pooling layer			
2-1st Conv	128	3 × 3 × 64	Frozen
2-2nd Conv	128	3 × 3 × 128	Frozen
Max pooling layer			
3-1st Conv	256	3 × 3 × 128	Frozen
3-2nd Conv	256	3 × 3 × 256	Trainable
3-3rd Conv	256	3 × 3 × 256	Trainable
Max pooling layer			
4-1st Conv	512	3 × 3 × 256	Trainable
4-2nd Conv	512	3 × 3 × 512	Trainable
4-3rd Conv	512	3 × 3 × 512	Trainable
5-1st Conv	512	3 × 3 × 512	Trainable
5-2nd Conv	512	3 × 3 × 512	Trainable
5-3rd Conv	512	3 × 3 × 512	Trainable

TABLE 4 | Architecture of the RPN in **Figure 11**.

Layer type	Filters	Size of kernel
[5-3rd Conv]Input layer		
RPN Conv	512	3 × 3 × 512
Classification convolutional layer(softmax)	30	1 × 1 × 512
[RPN Conv]Regression convolutional layer	60	1 × 1 × 512

TABLE 5 | Architecture of the classifier in Fast R-CNN in **Figure 11**.

Layer type	Size of output
[5-3rd Conv] [RPN proposal region]Input layer	
RoI Pooling	7 × 7 × 512 × 32
1st fully connected layer	4, 096 × 32
2nd fully connected layer	4, 096 × 32
Classification fully connected layer(softmax)	32 × 5
[2nd fully connected layer]Regression convolutional layer	32 × 16

4.1. Quantitative Metrics

Precision-recall (PR) curves of four classes of objects are used for result evaluation. The curve plots the precision against the recall rate of a detector, and it is a visual representation of an algorithm's performance. A detector with a higher precision and a higher recall rate indicates a better discrimination ability. Precision and recall rate are defined as follows:

$$Precision = tp/(tp + fp) \tag{10a}$$

$$Recallrate = tp/(tp + fn) \tag{10b}$$

Where tp represents the number of true positives, and fp represents the number of false positives. fn denotes the number of false negatives. Positive data and negative data mean the four types of detection objects and background region, respectively. A false positive case refers to the case where background is mistaken as a target or one kind of target is mistaken as another kind. A false negative case refers to the case where true positive data is error detected.

Mean Average Precision (mAP) of different classes of objects is used for result comparison. A mAP score is the mean of the average precision (Ap) for each class. The definition of AP is defined as:

$$AP = \int_0^1 p(r)dr \tag{11}$$

Where, p is the precision rate, and r is the recall rate. In our work, mAP score is reported using an intersection-over-union (IoU) threshold at 0.65. The bigger the mAP score, the better the detection result.

4.2. Key Variables Assessment for Synthesis

The significance of the background for target recognition has been studied in visual images, while the other two key variables (variable 1:shape and variable 2:gray scale distribution) effects on synthetic data are less understood. We designed a similar ablation experiment to study how each key variable influenced the synthesis process. The activation ratio of key variables in the dataset was controlled, and then the dataset was used to train the Faster R-CNN. The concept of activation here means that in a training set, the key variable of a specific proportion of samples satisfy diversity, and the variable of the remaining samples are kept constant. Through comparing the effects of different datasets on detection and recognition performance of Faster R-CNN, we analyze the robustness of the key variables.

We first generated an infrared cell image dataset (ICID) as a reference dataset using the procedure we described in section 3.2. ICID contains 1,4257 synthetic images (resolution≈ 70 × 280) and 4 categories of targets: inlet, outlet, obvious short circuit, latent short circuit. We also generated extra 8 annotated datasets where each key variable was active for 30, 50, 70, and 90% of the images, respectively, the numbers of samples are in parentheses in **Table 6**. To prove the effectiveness of the synthetic sample, we added 200 hand-annotated real images into the ICID to form a new dataset. The 10 datasets were fed to Faster R-CNN, respectively, and then the network tested on realistic electrolytic cells images to recognize the targets.

Table 6 shows the mAP scores of the Faster R-CNN trained on the 10 datasets. Reducing the activation ratio of the two key variables in dataset yields dramatic performance drop, which illustrates the significant impact of the two key variables toward the networks feature learning ability. But when more than 50% of the samples meet the diversity of each key variable, the growth value of mAP increases slowly. When 50% of the images are considered about the shape diversity (variable 1), the score is reduced by 8% compared to the reference dataset ICID; while for gray scale distribution (variable 2), the score is reduced only by 2%. It suggests that the networks learning is obviously more sensitive to target shape change, But the gray scale distributions

TABLE 6 | mAP score of the Faster R-CNN trained on synthetic datasets with different key variable activation ratio.

Datasets with key variable activated	Activation ratio			
	30% activated	50% activated	70% activated	90% activated
Variable 1	0.556 (15,649)	0.582 (15,640)	0.585 (14,578)	0.592(14,624)
Variable 2	0.578 (14,520)	0.617 (14,481)	0.619 (14,500)	0.621 (14,510)
ICID (14,257)	0.632			
ICID+Real images (14,457)	0.878			

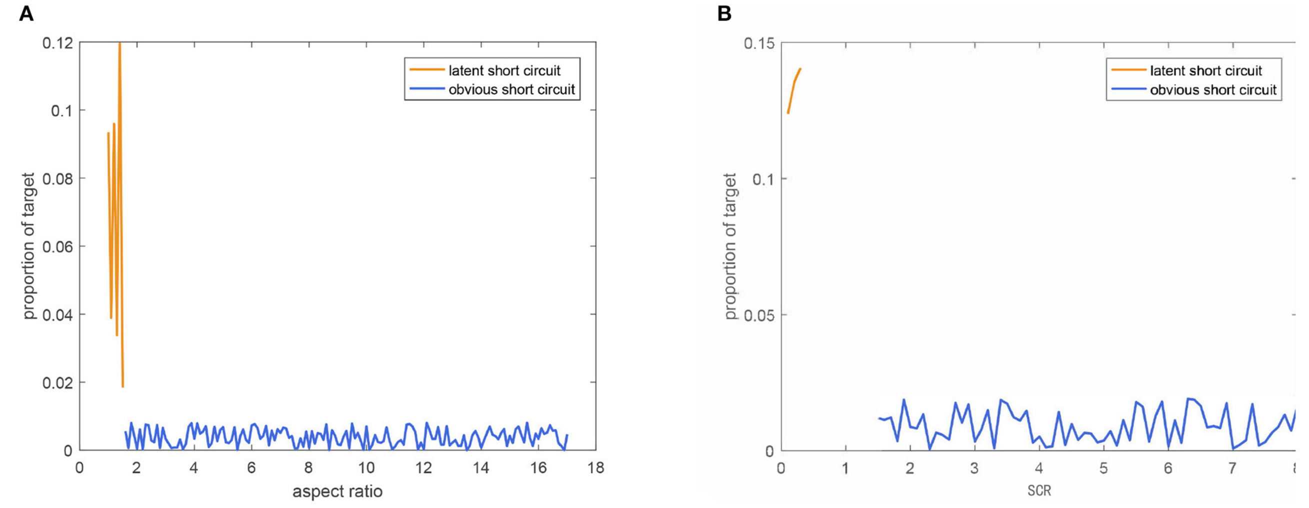


FIGURE 12 | (A) Distribution of the two types short circuits with respect to aspect ratio. **(B)** Distribution of the two types short circuits with respect to SCR value.

have a greater effect on learning than target shape, as the absolute value of the increase is larger. Especially when more samples meet the diversity of gray scale distribution, the mAP is close to that of ICID. It means that the key variables of the goal we defined are accurate, After training on ICID, a mAP score of 0.632 is obtained. The score increases to 0.878 when real images are added in. It suggests that the proposed synthetic method is effective and available at the beginning for deploying CNNs on practice metal scenario. The synthetic method helps avoid the laborious work of manually annotating large numbers of images.

Figure 12 shows the distribution of the two types of short circuits with respect to aspect ratio and SCR values in ICID. As aspect ratio determines the diversity of key variable 1, and similarly, SCR value determines the diversity of key variable 2 in our proposed synthetic method. Obvious short circuit class accounts for 60% of synthetic target number; the range of aspect ratio and SCR of obvious short circuit class are both obvious wider than latent short circuit, and the distribution is uniform.This indicates that the diversity of samples is guaranteed by a wide range of key variable values and uniform distribution of sample numbers. This is in accord with the actual engineering case.

4.3. Comparison of Data Synthesis Method

In this section, in order to compare the performance of different synthesis methods, Faster R-CNN trained on 8 training sets that generated with different methods listed below and tested on real cell images. We manually annotated 200 real images and used the four commonly methods M1, M2, M3, and M4 to augment the labeled images respectively. In M1, random visible light images were used to replace the electrolytic cell background while preserving targets. The set ICID was generated automatically without the aid of any labeled images by using our proposed method (OM).

- M1:Random background (Björklund et al., 2019)
- M2:Flipping
- M3:Color casting
- M4:Noise
- Fusion1: Fusing M2, M3 and M4.
- OM: ICID generated by the propsed synthetic method
- Fusion2: Fusing ICID and M2, M3 and M4.
- Baseline: 200 real image.

Table 7 shows the short circuit detection and recognition results with different data augmentation methods. Compared with the

TABLE 7 | Test results of different data sugmentation methods.

Methods	mAP	Obvious short circuit	Latent short circuit	Inlet	Outlet
M1	0.295	0.313	0.002	0.428	0.437
M2	0.493	0.606	0.006	0.679	0.680
M3	0.423	0.427	0.005	0.623	0.635
M4	0.473	0.613	0.006	0.628	0.646
Fusion1	0.552	0.620	0.007	0.791	0.790
OM	0.669	0.871	0.007	0.891	0.905
Fusion2	0.672	0.883	0.007	0.890	0.907
Baseline	0.421	0.425	0.006	0.621	0.633

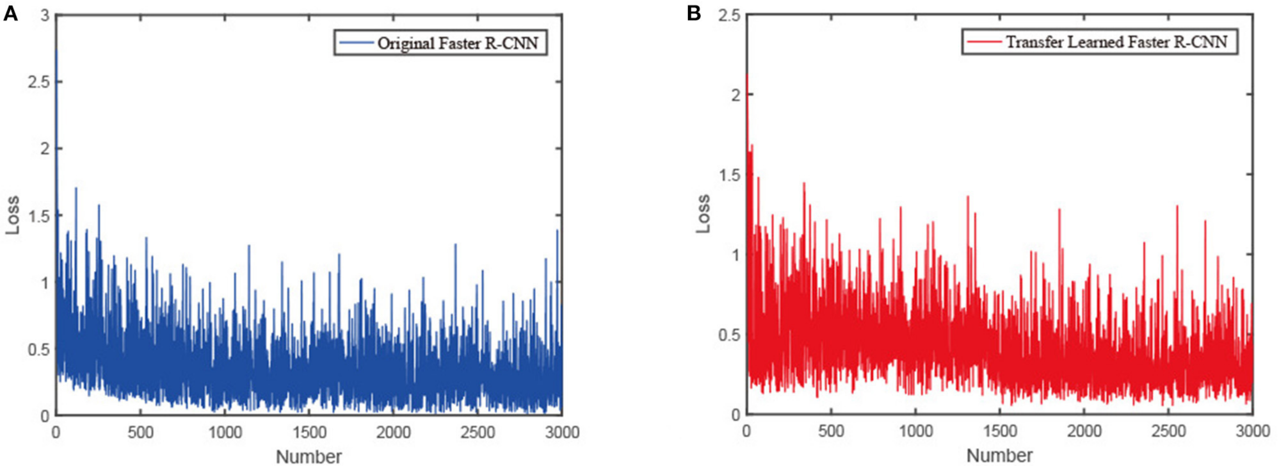


FIGURE 13 | The influence of transfer learning on the training of the Faster R-CNN: **(A)** Loss of the original Faster R-CNN with number of iterations. **(B)** Loss of the transfer learned Faster R-CNN with number of iterations.

Baseline, all methods improve the performance of short circuit detection except M1. That is because the backgrounds of short circuits are cell surfaces with complex thermal distribution. There is spatial continuity and gray correlation between targets and backgrounds, which can not be learned from other backgrounds. M2, M3, M4, and Fusion1 improve the mAP by 17.1%,4.75% 12.3%, and 31.1%, while the number improved by OM is 58.9%. Fusing the method OM with M2, M3, and M4 does not make a significant difference in detection performance, like the values with the gray shade in **Table 7**. Compared with other methods, our proposed synthesis method does not rely on any pre-annotated images at all, which is our original intention, that is, to solve the problem of lack of engineering samples effectively and labor-saving. At the same time, we note that in the table, the network's detection performance for latent short circuit has been kept at a low level. That is the subject to be discussed in our next experiment.

4.4. Accuracy of the Short Circuit Detection System

In this part, we verify our proposed short circuit detection system with attention-based Faster R-CNN(Att-Faster R-CNN). The method proposed in this paper can effectively improve the recall rate of latent short circuit while still maintaining a relative high detection accuracy about obvious short circuit class. To prove the advantage of our algorithm, we compared it with original Faster R-CNN (Ori-Faster R-CNN), Faster R-CNN with anchor parameter fine-tuning (Fin-Faster R-CNN), and U-Net. All experiments were performed under the same environment.

We evaluated the performance by training all models on ICID and tested on real cell infrared images. In the fine-tuning strategy, instead of 9 anchors with a basic size of 128 × 128, 256 × 256 and 512 × 512 and 3 aspect rations $(0.5, 1, 2)$ used for the PRN (Ren et al., 2017), we increased the number of anchors to 15 referring to the settings of Kim et al. (2018). The anchors sizes are 8 × 8, 16 × 16 and 32 × 32 and the five aspect ratios include $5:1$, $4:1$, $2:1$, $1:1$, and $1:1.5$. In the Fast R-CNN classification part, an RoI is treated as foreground with the threshold of $IoU = 0.65$. This choice respects the need for precise fault location in the engineering field. Transfer learning strategy is illustrated in **Table 3**, the weights of the first five layers from VGG-16 were frozen, and the rest of the layers were retrained. PR curves and RoC curves

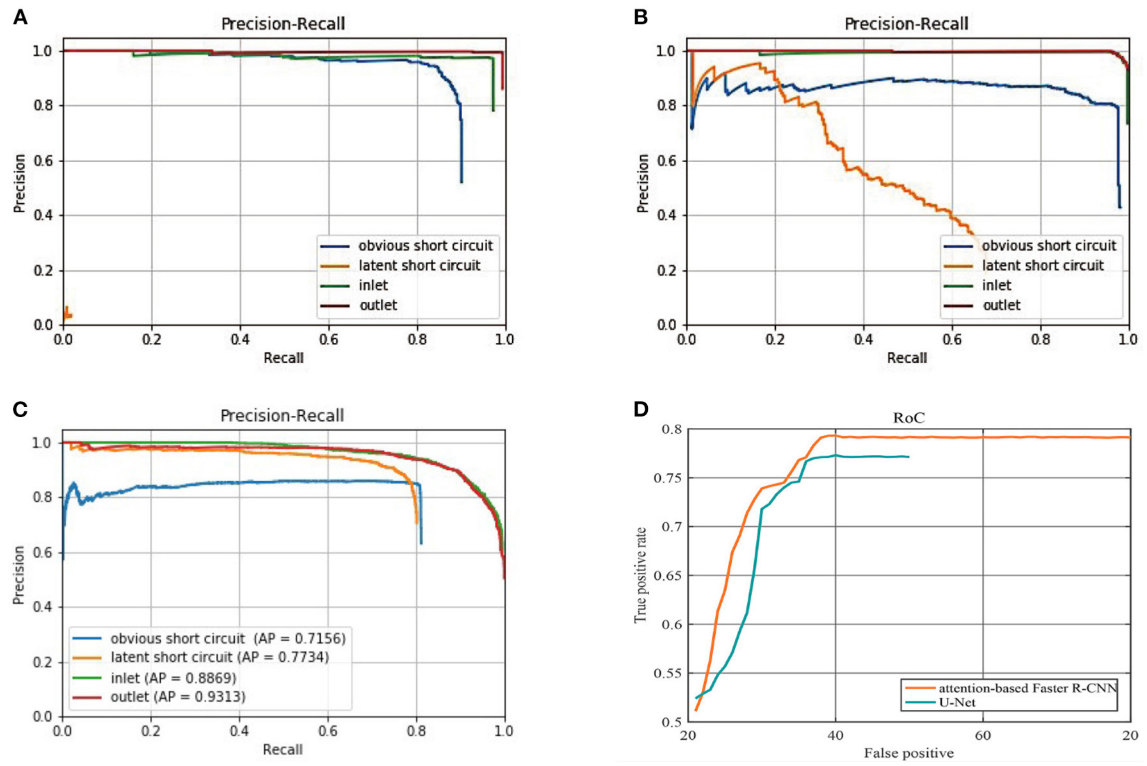


FIGURE 14 | (A) PR curves of the Ori-Faster R-CNN. **(B)** PR curves of the Fin-Faster R-CNN. **(C)** PR curves of the Att-Faster R-CNN. **(D)** RoC curves of Att-Faster R-CNN and U-Net for latent short circuit detection, on the x-coordinate, we replaced FPR with total false positive number.

TABLE 8 | The four class of targets average precision of Faster R-CNN with different anchor parameter setting.

Method	Obvious short circuit	Latent short circuit	Inlet	Outlet	mAP
Ori-Faster R-CNN	0.872	0.006	0.898	0.907	0.671
Fin-Faster R-CNN	0.815	0.460	0.907	0.908	0.772
Att-Faster R-CNN	0.715	0.773	0.887	0.931	0.826
U-Net	–	0.752	–	–	–

are adopted to illustrate the detection performance. Sum losses of the four class objects are used to indicate the efficiency of transfer learning.

Figure 13 shows the learning loss of the Faster R-CNN with and without transfer learning. The loss curve of the original Faster R-CNN converges around 1000 iterations, while the loss curve of the transfer learned Faster R-CNN converges at around 1500 iterations. At the beginning of the training, the gradient descent of our network is slow but stable, whereas the gradient of the original network soon drops to a stable value. It reveals that the transfer learned network can learn more general characteristic information from the dataset.

Figures 14A–C presents the detailed comparisons of three sets of PR curve for the four classes of objects, respectively with Ori-Faster R-CNN, Fin-Faster R-CNN, and Att-Faster R-CNN. Furthermore, comparision of small latent short circuit detection results with Att-Faster R-CNN and U-Net is also provided in **Figure 14D**. In conjunction with **Figure 14** and **Table 8** it can be observed that all models show the best detection performance on classes of inlet and outlet. Fin-Faster R-CNN can alleviate the difficult detection problems of latent short circuit, but keep a high recall rate need to sacrifice the precision. It is a dilemma for engineering management. Att-Faster R-CNN shows the best detection performance for latent short circuit. Because the attention mechanism integrates more semantic information about latent short circuits into the synthetic sample, aided by the variable anchor scale, the Att-Faster R-CNN can show a stable high precision at a high recall rate. Fin-Faster R-CNN and Att-Faster R-CNN both exhibit precision

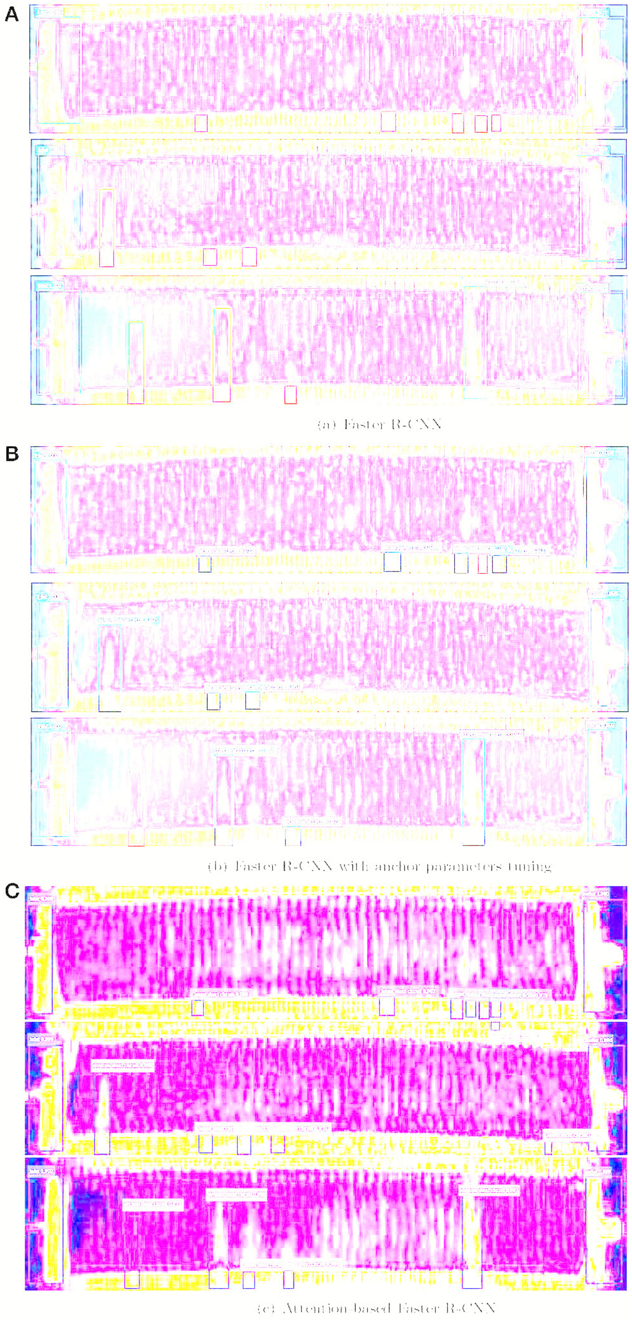

A
(a) Faster R-CNN
B
(b) Faster R-CNN with anchor parameters tuning
C
(c) Attention based Faster R-CNN

FIGURE 15 | (A) Short circuit detection result with original Faster R-CNN. **(B)** Short circuit detection result with Faster R-CNN with anchor parameter fine-tuning. **(C)** Short circuit detection result with attention-based Faster R-CNN.

decline for obvious short circuit class. It is an acceptable decrease of accuracy under the premise that miss detection of short circuits has a more serious impact on production. In **Figure 14D**, reliability of Att-Faster R-CNN is superior to U-Net, The promising results validate the effectiveness of the proposed attention mechanism for latent short circuit detection.

Figure 15 illustrates the detection results with the three Faster R-CNN network in three same scene, respectively. Attention-based Faster R-CNN able to accurately detect more latent short circuit. Whereas, the result with Ori-Faster R-CNN and Fin-Faster R-CNN lists more false negatives in the red box.

5. CONCLUSIONS

This work focused on short circuit detection in infrared image of metal electrolysis scene with CNNs. An infrared image synthetic method is proposed to automatically generate labeled infrared dataset ICID by simulating key variables of the scenario that affect the diversity of samples. Additionally, attention-based Faster R-CNN is proposed and used to design the short circuit detection system. In the system, an attention module integrates the semantic segment results of U-Net with the synthetic ICID to obtain rich representation ability on the infrared images. Combined with strategies of anchor parameters fine-tuning and transfer learning, the detection system can efficiently avoid the missed detection of a latent short circuit, and the performance is superior to the original Faster R-CNN and U-Net. The proposed method is specifically dedicated to metal electrolysis scenes, but the methodology of mining targets' key variables to automatically synthesize samples will be further extended to other application areas and training algorithms.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

XL: programming and preparation, creation and writing the initial draft. YL: review and editing, providing study materials. RW: programming. CZ: design of methodology. HZ: formulate research goals and aims. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported by the National Key R&D Program of China (grant no. 2019YFB1704700), the National Natural Science Foundation of China (grant no. 61890930-2), and the Postgraduate Research Innovation Project of Central South University (grant no. 2018zzts174).

REFERENCES

Aqueveque, P. E., Wiechmann, E. P., and Burgos, R. P. (2009). Short-circuit detection for electrolytic processes employing optibar intercell bars. *IEEE Trans. Ind. Appl*. 45, 1225–1231. doi: 10.1109/TIA.2009.2023357

Asari, K. V., Kumar, S., and Radhakrishnan, D. (1999). A new approach for nonlinear distortion correction in endoscopic images based on least squares estimation. *IEEE Trans. Med. Imaging* 18, 345–354. doi: 10.1109/42.768843

Björklund, T., Fiandrotti, A., Annarumma, M., Francini, G., and Magli, E. (2019). Robust license plate recognition using neural networks trained on synthetic images. *Pattern Recognit*. 93, 134–146. doi: 10.1016/j.patcog.2019.04.007

Ce, L., Baochang, Z., Hanwen, H., and Jing, D. (2018). Enhanced bird detection from low-resolution aerial image using deep neural networks. *Neural Proc. Lett*. 49, 1021–1039. doi: 10.1007/s11063-018-9871-z

Chen, C. L. P., Hong, L., Wei, Y., Tian, X., and Yuan, Y. T. (2013). A local contrast method for small infrared target detection. *IEEE Trans. Geosci. Remote Sens*. 52, 574–581. doi: 10.1109/TGRS.2013.2242477

Fan, Z., Bi, D., Xiong, L., Ma, S., He, L., and Ding, W. (2018). Dim infrared image enhancement based on convolutional neural network. *Neurocomputing* 272, 396–404. doi: 10.1016/j.neucom.2017.07.017

Ferwerda, J. A., Pattanaik, S., Shirley, P., and Greenberg, D. P. (1996). "A model of visual adaptation for realistic image synthesis," in *International Conference on Computer Graphics and Interactive Techniques* (New York, NY), 249–258.

Gao, Z., Zhang, Y., and Li, Y. (2020). Extracting features from infrared images using convolutional neural networks and transfer learning. *Infrared Phys. Technol*. 105:103237. doi: 10.1016/j.infrared.2020.103237

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition* (Columbus, OH: IEEE), 580–587.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial nets. *ArXiv* arXiv:2672.2680.

Hiary, H., Saadeh, H., Saadeh, M., and Yaqub, M. (2018). Flower classification using deep convolutional neural networks. *IET Comput. Vis*. 12, 855–862. doi: 10.1049/iet-cvi.2017.0155

Hong, Z., and Wang, Q. (2017). "A new processing method of infrared temperature images in copper electrolysis," in *IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society* (Beijing: IEEE).

Huang, H., Zhou, H., Yang, X., Zhang, L., Qi, L., and Zang, A. Y. (2019). Faster R-CNN for marine organisms detection and recognition using data augmentation. *Neurocomputing* 337, 372–384. doi: 10.1016/j.neucom.2019.01.084

Jiang, B., Ma, X., Lu, Y., Li, Y., Feng, L., and Shi, Z. (2019). Ship detection in spaceborne infrared images based on convolutional neural networks and synthetic targets. *Infrared Phys. Technol*. 97:229–234. doi: 10.1016/j.infrared.2018.12.040

Kim, J. H., Batchuluun, G., and Park, K. R. (2018). Pedestrian detection based on Faster R-CNN in nighttime by fusing deep convolutional features of successive images. *Expert Syst Appl*. 114, 15–33. doi: 10.1016/j.eswa.2018.07.020

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., et al. (2016). *Ssd: Single Shot Multibox Detector*. Cham: Springer.

Maekipaeae, E., Tanttu, J. T., and Virtanen, H. (1997). "Ir-based method for copper electrolysis short circuit detection," in *Proceedings of SPIE-The International Society for Optical Engineering* (Orlando, FL).

Mayer, N., Ilg, E., Fischer, P., Hazirbas, C., Cremers, D., Dosovitskiy, A., et al. (2018). What makes good synthetic training data for learning disparity and optical flow estimation? *Int. J. Comput. Vis*. 126, 942–960. doi: 10.1007/s11263-018-1082-6

Ngxande, M., Tapamo, J. R., and Burke, M. (2019). "DepthwiseGANs: fast training generative adversarial networks for realistic image synthesis," in *2019 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA)* (Bloemfontein: IEEE).

Odena, A., Olah, C., and Shlens, J. (2016). Conditional image synthesis with auxiliary classifier GANs. *arXiv e-prints* arXiv:1610.09585.

Peng, K., and Chen, T. (2015). "A framework of extracting multi-scale features using multiple convolutional neural networks," in 2015 *IEEE International Conference on Multimedia and Expo (ICME)* (Turin), 1–6.

Ratner, A. J., Ehrenberg, H. R., Hussain, Z., Dunnmon, J.and Ré, C. (2017). Learning to compose domain-specific transformations for data augmentation. *arXiv e-prints* arXiv:1709.01643.

Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell*. 39, 1137–1149. doi: 10.1109/TPAMI.2016.2577031

Ronneberger, O. (2017). "U-net: convolutional networks for biomedical image segmentation," in *Bildverarbeitung für die Medizin 2017*, eds Maier-Hein, geb. K. H. Fritzsche, Deserno, geb. T. M. Lehmann, H. Handels, and T. Tolxdorff (Berlin; Heidelberg: Springer Berlin Heidelberg), 3–3.

Samadi, F., Akbarizadeh, G., and Kaabi, H. (2019). Change detection in sar images using deep belief network: a new training approach based on morphological images. *IET Image Proc*. 13, 2255–2264. doi: 10.1049/iet-ipr.2018.6248

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and Lecun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *eprint arXiv*.

Silva, L. A. D., Bressan, P. O., Gonçalves, D. N., Freitas, D. M., and Gonçalves, W. N. (2019). Estimating soybean leaf defoliation using convolutional neural networks and synthetic images. *Comput. Electron. Agric*. 156, 360–368. doi: 10.1016/j.compag.2018.11.040

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv e-prints* arXiv:1409.1556.

Sun, X., Wu, P., and Hoi, S. C. (2018). Face detection using deep learning: An improved Faster R-CNN approach. *Neurocomputing* 299, 42–50. doi: 10.1016/j.neucom.2018.03.030

Sungho, K., and Lee, J. (2012). Scale invariant small target detection by optimizing signal-to-clutter ratio in heterogeneous background for infrared search and track. *Pattern Recognit*. 45, 393–406. doi: 10.1016/j.patcog.2011.06.009

Wang, X., Peng, Z., Kong, D., and He, Y. (2017). Infrared dim and small target detection based on stable multisubspace learning in heterogeneous scene. *IEEE Trans. Geosci. Remote Sens*. 55, 5481–5493. doi: 10.1109/TGRS.2017.2709250

Xiao, L.,Mao, Q., Lan, P., Zang, X., and Liao, Z. (2017). "A fault diagnosis method of insulator string based on infrared image feature extraction and probabilistic neural network," in *International Conference on Intelligent Computation Technology Automation* (Changsha), 80–85.

Xing, C., Wang, Z., Meng, F., and Dong, C. (2019). Fusion of infrared and visible images with gaussian smoothness and joint bilateral filtering iteration decomposition. *IET Comput. Vis*. 13, 44–52. doi: 10.1049/iet-cvi.2018.5027

Zhang, H., Xu, T., Li, H., Zhang, S., Huang, X., Wang, X., et al. (2016). StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv e-prints* arXiv:1612.03242. doi: 10.1109/ICCV.2017.629

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.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 06 December 2021
doi: 10.3389/fnbot.2021.753052

Multi-Target Coordinated Search Algorithm for Swarm Robotics Considering Practical Constraints

You Zhou[1,2]*, *Anhua Chen*[1], *Xinjie He*[3] *and Xiaohui Bian*[3]

[1] *Department of Mechanical and Electrical Engineering, Hunan University of Science and Technology, Xiangtan, China,* [2] *Intelligent Manufacturing College, Hunan Vocational Institute of Technology, Xiangtan, China,* [3] *Department of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan, China*

In order to deal with the multi-target search problems for swarm robots in unknown complex environments, a multi-target coordinated search algorithm for swarm robots considering practical constraints is proposed in this paper. Firstly, according to the target detection situation of swarm robots, an ideal search algorithm framework combining the strategy of roaming search and coordinated search is established. Secondly, based on the framework of the multi-target search algorithm, a simplified virtual force model is combined, which effectively overcomes the real-time obstacle avoidance problem in the target search of swarm robots. Finally, in order to solve the distributed communication problem in the multi-target search of swarm robots, a distributed neighborhood communication mechanism based on a time-varying characteristic swarm with a restricted random line of sight is proposed, and which is combined with the multi-target search framework. For the swarm robot kinematics, obstacle avoidance, and communication constraints of swarm robots, the proposed multi-target search strategy is more stable, efficient, and practical than the previous methods. The effectiveness of this proposed method is verified by numerical simulations.

Keywords: multi-target search, swarm robots, roaming search, coordinated search, simplified virtual force model, distributed neighborhood communication

OPEN ACCESS

Edited by:
Chen Qiao,
Xi'an Jiaotong University, China

Reviewed by:
Weichao Yue,
Zhengzhou University of Light Industry, China
Laihao Jiang,
Harbin Engineering University, China
Ping Zhou,
Northeastern University, China

***Correspondence:**
You Zhou
zhouyou613@163.com

Received: *04 August 2021*
Accepted: *01 November 2021*
Published: *06 December 2021*

Citation:
Zhou Y, Chen A, He X and Bian X (2021) Multi-Target Coordinated Search Algorithm for Swarm Robotics Considering Practical Constraints. Front. Neurorobot. 15:753052. doi: 10.3389/fnbot.2021.753052

INTRODUCTION

Inspired by the group behavior of social insects such as ants and bees, the concept of swarm intelligence is put forward by scholars (Bonabeau, 1999), which is defined as the collective intelligence emerging from a group of simple agents. The swarm robot system (Doty and Van Aken, 2002) is a typical artificial swarm intelligence system, which consists of a large number of homogeneous autonomous robots with a simple structure. By the coordination and cooperation of robots with limited individual capabilities under a specific mechanism, the system can present intelligent behavior and complete relatively complex tasks.

The common research contents of swarm robot systems include target search (Alfeo et al., 2019; Booth et al., 2020), task assignment (Liang et al., 2018), cluster avoidance (Khan et al., 2019), path planning (Ryan, 2008; Luo et al., 2017), and cluster formation (Anonymous, 1993; Alsamman, 2011). In this paper, the target search problem of swarm robots in unknown complex environments is mainly studied, such as forest fire detection

(Yao et al., 2018; Marzaeva, 2019), toxic gas leak detection (Zhang et al., 2010; Moshayedi and Gharpure, 2013), search and rescue of missing personnel (Goodrich et al., 2009; Kamegawa et al., 2020), military target detection (Ha and Cho, 2018; Jiong et al., 2019) and so on. In order to solve this type of search problem, there are mainly composed of two main categories of design strategies, namely, behavior-based search and learning-based search (Cizek and Faigl, 2019; Berscheid et al., 2020; Suzuki et al., 2020), and this article mainly discusses the former.

According to the number of search targets, searches can be divided into single-target searches and multi-target searches. When the swarm robot system is applied to single-target search, it is necessary to pay attention to the cooperation mechanism between individual robots. Gudise (2004) proposed an extended particle swarm optimization (EPSO) algorithm, which was successfully applied to single-target searches. Ducatelle et al. (2011) used the local wireless network communication strategy to strengthen the communication ability between robots and enhance the robustness of the swarm robot system. Majid and Arshad (2017) mainly focused on the performance indicators in the EPSO coordinated search algorithm such as trajectory smoothness, search success rate, and search time, and studied the impact of the inertial weight on the search performance of swarm robots. Tang et al. (2020) proposed an improved adaptive bat algorithm (IABA) search algorithm by focussing on the problem of obstacle avoidance and improving the performance of the algorithm in the single-target search process of swarm robots. Aiming at the distributed communication problem in the single-target search process of swarm robots, Yang et al. (2019) proposed a time-varying characteristics swarm of visual limited (V-TVCS) model.

However, when the swarm robot system is applied to the actual neighborhood search, the number of search targets is more than one. Therefore, how to set up a multi-target search algorithm considering the actual search environment is the focus of scholars at home and abroad. Manic (2009) proposed a multi-target task allocation model with response threshold (TRT) to realize self-organizing task allocation, and then robots with the same objective task used the EPSO algorithm for coordinated search. Zhang and Xue (2014) proposed a dynamic task division strategy with closed-loop adjustment for the problem of uneven subgroup size of the TRT model. Xinjie (2020) established a simplified virtual force model (SVFM) for the unknown and complex environment, and successfully solved the obstacle avoidance problem in the multi-target search process. Zhang and Xue (2015) proposed the strategies of competition and cooperation and cooperation for the problem of subgroup interaction in parallel search. Jie (2019) proposed a probabilistic finite state machine search framework for the multi-target search problem of swarm robots. Xinjie (2020) extended the two-dimensional SVFM (2D-SVFM) to 3D-space, and successfully implemented this type of search method to achieve multi-target search in the Unmanned Aerial Vehicle (UAV) cluster system.

Based on the above literature analysis, the above methods can be applied to specific target search scenarios, but there are the following problems. First of all, there is no standard multi-target search algorithm framework in these methods. Most of the algorithms' settings are only suitable for searching for a specific number of targets, not for searching for any number of targets. Secondly, most of the algorithms only start to study a specific performance index of swarm robots, and do not consider the algorithm performance, obstacle avoidance, and swarm communication problem of swarm robots in actual search scenarios at the same time.

Aiming at the static multi-target search problem of swarm robots in unknown complex environments, a multi-target coordinated search algorithm for swarm robots considering practical constraints (MSRCPC) is proposed in this paper. The main work of this paper is as follows. First, based on the mechanism of finite state machines, an ideal multi-target search framework for swarm robots is proposed. Then, on the basis of the entire framework, combined with the simplified virtual force model, the obstacle avoidance problem of the swarm robot in the multi-target search process is solved. Finally, considering the communication interaction problem in the coordination and cooperation of swarm robots and the random line-of-sight problem of individual robots in the actual communication process, the distributed neighborhood interaction model based on a time-varying characteristic swarm with a restricted random line of sight (RS-TVCS) is constructed. By embedding the sub-algorithms in the whole algorithm framework, the MSRCPC algorithm proposed in this paper can greatly improve the search performance of the swarm robot system, making the entire system more scalable and practical.

The remaining parts of this paper are summarized as follows. In section 1, the research background of this algorithm and the research progress at home and abroad was introduced. In section 2, the ideal multi-target search framework for swarm robots is introduced. In section 3, the obstacle avoidance mechanism and distributed communication mechanism of the swarm robot system are described, and the multi-objective search framework of swarm robots considering practical constraints is proposed. The simulation test analysis on the proposed algorithm is conducted in section 4. Finally, the main work is summarized.

THE FRAMEWORK OF IDEAL SEARCH ALGORITHM

In a closed two-dimensional space $\boldsymbol{R}^2$, the task environment for multi-target search of swarm robots can be described by the set $\{\boldsymbol{R}, \boldsymbol{T}, \boldsymbol{S}, \boldsymbol{D}\}$.where, $\boldsymbol{R} = \{\boldsymbol{R_1},..., \mathrm{R}_{\boldsymbol{i}},..., \mathrm{R}_{\boldsymbol{m}}, \boldsymbol{m} > 1\}$ is the search subject (swarm robots); $\boldsymbol{T} = \{\boldsymbol{T_1},..., \boldsymbol{T_j},..., \boldsymbol{T_n}, \boldsymbol{n} > \boldsymbol{m}\}$ is the searched target; $\boldsymbol{S} = \{\boldsymbol{S_1},..., \boldsymbol{S_o},..., \boldsymbol{S_p}, \boldsymbol{p} > 1\}$ is the static obstacle and $\boldsymbol{D} = \{\boldsymbol{D_1},..., \boldsymbol{D_l},..., \boldsymbol{D_q}, \boldsymbol{q} > \boldsymbol{l}\}$ is the dynamic obstacle. In addition, we let $\boldsymbol{w}$ exist in the task set $\{\boldsymbol{R}, \boldsymbol{T}, \boldsymbol{S}, \boldsymbol{D}\}$.

The set targets can send out a continuous specific signal, and are randomly distributed in the search map. The sensors carried by swarm robots can detect the strength of the target signal, which cannot determine the direction of the signal. The initial positions of swarm robots are randomly in a certain corner of the search map. In the case without considering obstacles and ideal communication interaction, the multi-target search algorithm framework of swarm robots can be described in the

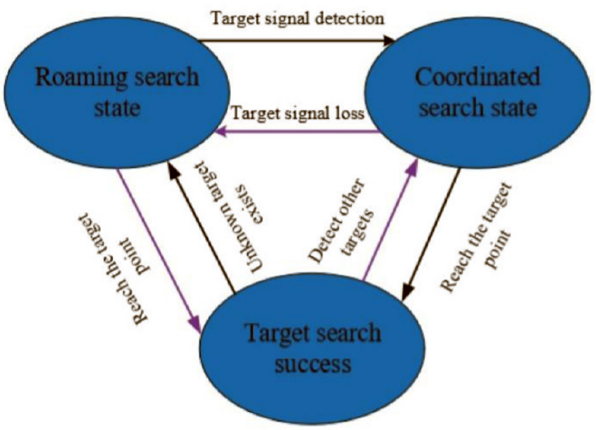


FIGURE 1 | Multi-target search algorithm framework diagram.

form of a finite state machine. The specific description is shown in **Figure 1**.

As shown in **Figure 1**, the basic multi-target search algorithm framework can be described as: when the robot detects the target signal, it enters the coordinated search state, and uses the swarm intelligence optimization algorithm to coordinate the search; when the robot does not detect the target signal, it will follow a certain roaming mechanism to detect the target signal.

The Multi-Assignment Model Based on Response Threshold

Sensor Detection Model

Sensors with different detection distances have different response strengths for target signal, and the function to describe the target signal strength can be set as follows (Manic, 2009):

$$I(i,j) = \begin{cases} \frac{sQ}{d_{ij}^2} + rand(), d_{ij} \le d_0 \\ 0, d_{ij} \ge d_0 \end{cases} \tag{1}$$

where Q is the constant power signal sent from the center of the target, $\boldsymbol{d_{ij}}$ denotes the distance between the robot and the target, $\boldsymbol{d_0}$ is the maximum detection distance of the sensor, $\boldsymbol{s}$ is the signal attenuation factor, ***rand*** is the random disturbance of the signal, and $\boldsymbol{I(i,j)}$ is the signal strength between robot and target.

Multi-Target Allocation and Design

In the robot roaming search process, the robot may detect multiple target signals. How to make self-organizing decisions on the target signals and find subgroup alliances is the key to the coordinated search of swarm robots. First, the target response function is used to calculate the detection of each target signal for each robot at time $\boldsymbol{t}$. Then, the probability that the robot selects the target is calculated via the target response signal strength, Finally, the decision about the target based on the roulette probability decision algorithm is made. As shown in **Table 1**, the induction about the target signal strength of the robot at the moment $\boldsymbol{t}$ is as follows:

The probability response process of the $\boldsymbol{i}$-th robot to the $\boldsymbol{j}$-th target is:

$$p(i,j) = \frac{I_j^2}{\sum\limits_{k=1}^{m} I_k^2} \tag{2}$$

where $\boldsymbol{I_j}$ is the signal strength of the target $\boldsymbol{T_j}$ detected by the robot $\boldsymbol{R_i}$. If the robot can detect the number of targets, i.e., m, the probability that $\boldsymbol{R_i}$ responds to the excitation from target $\boldsymbol{T_j}$ is $\boldsymbol{p}$ $(\boldsymbol{i},\boldsymbol{j})$. The $\boldsymbol{R_i}$ decision-making process of the robot $\boldsymbol{R_i}$ is as follows:

$$k = \min\left[\sum_{j=1}^{m} p(i,j) \ge rand()\right] \tag{3}$$

where ***rand* ()** is subject to a uniform score between 0 and 1, and k is the smallest target sequence number satisfying its condition. According to the processed decision-making method, it can be determined from **Table 1** that the subgroup alliances composed of the task target set are $\boldsymbol{T_1} = \{\boldsymbol{R_1}, \boldsymbol{R_5}\}$, $\boldsymbol{T_2} = \{\boldsymbol{R_2}, \boldsymbol{R_3}\}$, and $\boldsymbol{T_3} = \{\boldsymbol{R_4}\}$, and the members of $\boldsymbol{R_6}$ are in the roaming search state and do not participate in the coordinated search.

The Roaming Search Algorithm Based on Nearest Neighbor Exclusion Diffusion

At the initial moment, the robot cannot detect the target signal. Therefore, it is very important to design an effective individual roaming search model to detect the target signal at the fastest speed. Typical roaming search models include Levy Flight (Viswanathan et al., 1999) and Intermittent Search (Bénichou et al., 2006). However, the roaming search strategies of these models suffer from the following disadvantages: (1) the search efficiency is not high, and (2) the factor of obstacle avoidance is not considered in the search process. Therefore, a new roaming search algorithm, namely, the Nearest Neighbor Exclusion Diffusion (NNED) Algorithm is introduced in this section.

Suppose the position information of the $\boldsymbol{i}$-th robot in the search space at time $\boldsymbol{t}$ is expressed as $\boldsymbol{X_{ri}(t)} = [\boldsymbol{X_i(t)}, \boldsymbol{Y_i(t)}]^{\boldsymbol{T}}$, and the maximum speed of the roaming robot is $\boldsymbol{V_m}$. The NNED algorithm is described below.

Without considering obstacles, the distance matrix $\boldsymbol{D_{im}}$ between the $\boldsymbol{i}$-th robot and other robots at time $\boldsymbol{t}$ can be expressed as follows:

$$D_{im} = \left[d_{i1}, d_{i2}, \ldots, d_{ik}, \ldots, \mathrm{d}_{im}\right] \tag{4}$$

where $\boldsymbol{d_{ik}}$ is the Euclidean distance between the $\boldsymbol{i}$-th robot and the $\boldsymbol{k}$-th robot. Sort equation (4) by row from small to large to obtain the distance sorting matrix $\boldsymbol{D_{is}}$.

$$D_{is} = sort(D_{im}) \tag{5}$$

The position sequence information index of the neighboring robot can be expressed as follows:

$$index = find(D_{im}(1,:) == D_{is}(1,2)) \tag{6}$$

TABLE 1 | Detection of target signals by robots members at time *t*.

Robot	Perceived target type			Perceive target signal strength			Personalized task set
	T_1	T_2	T_3	T_4	T_5	T_6	
R_1	I-type	Unknown	I-type	0.9358	0	0.3346	$\{T_1,T_3\}$
R_2	Unknown	I-type	Unknown	0	0.6632	0	$\{T_2\}$
R_3	Unknown	I-type	Unknown	0	0.6632	0	$\{T_2\}$
R_4	Unknown	II-type	I-type	0	0.9358	0.6632	$\{T_2,T_3\}$
R_5	I-type	II-type	I-type	0.3346	0.6632	0.9358	$\{T_1,T_2,T_3\}$
R_6	Unknown	Unknown	Unknown	0	0	0	none

The repulsion angle $\boldsymbol{\theta}$ between the $\boldsymbol{i}$-th robot and the ***index***-th robot is expressed as follows:

$$\theta(t)=\begin{cases} ac\sin(\frac{X_i(t)-X_{index}(t)}{d_{iindex}}),\gamma_{index}(t)\geq\gamma_i(t) \\ \pi-ac\sin(\frac{X_i(t)-X_{index}(t)}{d_{iindex}}),\gamma_{index}(t)\leq\gamma_i(t) \end{cases} \tag{7}$$

where $\boldsymbol{d_{iindex}}$ is the Euclidean distance between the $\boldsymbol{i}$-th robot and the ***index***-th robot. Set the expected position of the robot at time $\boldsymbol{t}+1$ as $\boldsymbol{X'_{ri}}(\boldsymbol{t}+1)=[\boldsymbol{x_i'}\,(\boldsymbol{t}+1),\boldsymbol{y_i'}\,(\boldsymbol{t}+1)]^{\boldsymbol{T}}$, and the step size is updated as follows:

$$\begin{bmatrix} x_i'(t+1) \\ y_i'(t+1) \end{bmatrix} = \begin{bmatrix} x_i(t) \\ y_i(t) \end{bmatrix} + \begin{bmatrix} V_m\cos(\theta(t)) \\ V_m\sin(\theta(t)) \end{bmatrix} \tag{8}$$

Taking into account the boundary constraints, the actual position of the roaming robot is updated as follows:

$$\begin{cases} V_x{}'=-V_m\cos(\theta(t)), V_m\cos(\theta(t))\leq 0\cap x_i(t+1)\leq 0 \\ V_x{}'=-V_m\cos(\theta(t)), V_m\cos(\theta(t))\leq 0\cap x_i(t+1)\geq L \\ V_x{}'=-V_m\cos(\theta(t)), V_m\cos(\theta(t))\geq 0\cup 0\leq x_i(t+1)\leq L \end{cases} \tag{9}$$

where $\boldsymbol{L}$ is the search boundary. In the same way, the $\boldsymbol{y}$-axis velocity component considering the boundary limit can be updated. Set the actual updated position of the robot at time $\boldsymbol{t}+1$ as $\boldsymbol{X_{ri}(t+1)=[x_i(t+1),y_i(t+1)]^T}$, and the position update of the roaming robot considering boundary constraints is as follows:

$$\begin{bmatrix} x_i(t+1) \\ y_i(t+1) \end{bmatrix} = \begin{bmatrix} x_i(t) \\ y_i(t) \end{bmatrix} + \begin{bmatrix} V_x{}' \\ V_y{}' \end{bmatrix} \tag{10}$$

where $\boldsymbol{V_{ic}(t+1)=[V'(x),\ V'(y)]^T}$ is update step of the robot roaming speed.

Coordinated Search Algorithm of Particle Swarm Based on Kinematics Constraints

By analyzing and comparing several benchmark concepts in the cooperative search state of the particle swarm algorithm and swarm robots, it can be found that there is a certain mapping relationship between them. Based on the inertial weight particle swarm algorithm, kinematic constraints can be used to describe this mapping relationship, and the specific expression is as follows (Gudise, 2004):

$$\begin{cases} V_{ie}(t+1)=\omega V_{Ri}(t)+c_1r_1(X^*_{Ri}(t)-X_{Ri}(t)) \\ \quad +c_2r_2(g^*_{Ri}(t)-X_{Ri}(t)) \\ V_{Ri}(t+1)=V_{Ri}(t)+(V_{ie}(t+1)-V_{Ri}(t))\cdot\alpha \\ X_{Ri}(t+1)=X_{Ri}(t)+V_{Ri}(t+1)\cdot\delta \\ V_{Ri}(t+1)\leq V_m \end{cases} \tag{11}$$

where $\mathrm{V_{ie}(t+1)}$ is the expected speed of the robot at the next moment, $\mathrm{V_{Ri}(t+1)}$ is the speed of the robot at time t, $\mathrm{X^*_{Ri}(t)}$ is the historical optimal position of the individual robots, g * $_{\mathrm{Ri}}$ (t) is the optimal position of the robot at time t, $\mathrm{V_{Ri}(t+1)}$ is the actual expected speed considering the kinematics of the robot, $\mathrm{X_{Ri}(t)}$ is the position coordinate of the robot at time t, $\mathrm{X_{Ri}(t+1)}$ is the expected position of the robot at the next time, w is the inertial weight, c_1 and c_2 are the individual and social cognitive coefficients of the robot, r_1 and r_2 are random numbers uniformly distributed between 0 and 1, α is the inertia coefficient, δ is the step size control factor of the robot, and V_m is the limited maximum speed.

Setting the target position as $[\mathrm{X_{ot}},\mathrm{Y_{ot}}]^{\mathrm{T}}$, the fitness function of the coordinated search of the robot is as follows:

$$f(t)=\sqrt{(x_i(t)-X_{ot})^2+(y_i(t)-Y_{ot})^2} \tag{12}$$

Because the particle swarm optimization algorithm easily falls into the local best optimum, its inertia weight is improved by combining the actual search situation of the robot in this paper. The basic idea is as follows: when the distance between the particle and the target exceeds a certain threshold, w remains large and the global search is performed; when the distance between the particle and the target is less than a given threshold, w uses its fitness value to performs adaptive non-linear decrement value, fine-grained search and continuously approach the target point. The sigmoid function in the neural network has a strong non-linear approximation ability, whose extreme value ranges between 0 and 1. Since the value of the inertia weight $\boldsymbol{w}$ in the particle swarm is almost the same, the mapping relationship is as follows:

$$g(x)=\frac{2}{1+e^{-x}}-1, x\succ 0 \tag{13}$$

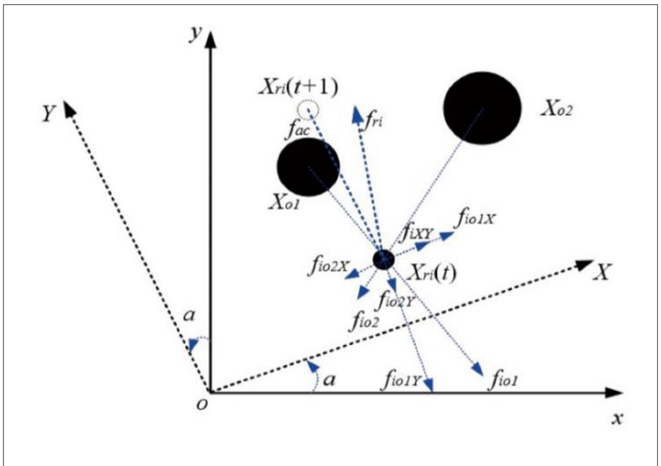


FIGURE 2 | SVFM obstacle avoidance model.

Then, a function is introduced into a distance-dependent robots system to adapt the value of inertia weight, and the specific expression is as follows:

$$\omega = \begin{cases} \frac{2}{1+e^{-5d/d_l}} - 1, d = f(i) \le d_l \\ 0.8, d = f(i) \succ d_l \end{cases} \tag{14}$$

where $\boldsymbol{d_l}$ is the set distance threshold and $\boldsymbol{d = f(i)}$ is the fitness value of the robot.

THE FRAMEWORK OF SWARM ROBOT SEARCH ALGORITHM CONSIDERING PRACTICAL CONSTRAINTS

On the basis of the ideal multi-target search algorithm framework in the previous section, in this section, the problems of real-time obstacle avoidance and distributed communication in the search process of swarm robots are considered, and a multi-target search algorithm framework for swarm robots considering practical constraints is designed.

Simplified Virtual Force Model

Aiming at the obstacle avoidance problem in the multi-target search process of swarm robots, introducing a simplified virtual force model can not only perfectly overcome the collision avoidance problem between robots, but also can be well integrated with the entire search algorithm framework, and the performance of the algorithm is also guaranteed.

The Construction of Obstacle Avoidance Model

The idea of this model is described in **Figure 2**. Supposing that the position of the *i*-th robot at time *t* is $X_{ri}(t)$, the position of the robot at time $t+1$ under the framework of the ideal multi-target search algorithm is $X_{ri}(t+1)$. It is obvious from **Figure 2** that the local path planned by the robot from t to $t+1$ will coincide with the position of the obstacle.

First, find out the position information of two neighboring obstacles or robots based on the information of obstacles or neighboring robots detected by the sensor of i-th robot sensor, which are X_{o1} and X_{o2} respectively.

Then it is assumed that the robot will be affected by the virtual introduction fac at the next moment and two neighboring obstacles or robot repulsion which are f_{io1} and f_{io2} respectively.

Now define the rotation matrix T_R of the new coordinate system XOY generated by rotating the xoy coordinate system counterclockwise by angle a as follows:

$$TR = \begin{bmatrix} \cos(a) & \sin(a) \\ -\sin(a) & \cos(a) \end{bmatrix} \tag{15}$$

Set gravity fac as follows:

$$f_{ac} = \begin{bmatrix} f_{acx}(t) \\ f_{acy}(t) \end{bmatrix} = \begin{bmatrix} x_i(t) \\ y_i(t) \end{bmatrix} - \begin{bmatrix} x_i(t+1) \\ y_i(t+1) \end{bmatrix} \tag{16}$$

The rotation matrix parameter a can be expressed as follows:

$$a = \arctan(\frac{f_{acy}(t)}{f_{acx}(t)}) \tag{17}$$

The force function of a given neighbor obstacle or robot is as follows:

$$f_{rep} = k_1 \cdot (\frac{1}{d_{ik}} - \frac{1}{d_a})^2 \tag{18}$$

where d_a is the obstacle avoidance distance of the object (static obstacles, robots, and dynamic obstacles) in the search process, and $\boldsymbol{d_{ik}}$ is the distance between the robot and the obstacle in the search process, and k_1 is the obstacle avoidance parameter of the robot.

Therefore, the coordinate components of obstacles (robots) X_{o1} and X_{o2} to robot i in the XOY coordinate system can be respectively obtained by the simultaneous equations (15)-(18), which are as follows:

$$\begin{bmatrix} f_{io1X}(t) \\ f_{io1Y}(t) \end{bmatrix} = \begin{bmatrix} \cos(a) & \sin(a) \\ -\sin(a) & \cos(a) \end{bmatrix} \cdot \begin{bmatrix} f_{io1x}(t) \\ f_{io1y}(t) \end{bmatrix} \tag{19}$$

or

$$\begin{bmatrix} f_{io2X}(t) \\ f_{io2Y}(t) \end{bmatrix} = \begin{bmatrix} \cos(a) & \sin(a) \\ -\sin(a) & \cos(a) \end{bmatrix} \cdot \begin{bmatrix} f_{io2x}(t) \\ f_{io2y}(t) \end{bmatrix} \tag{20}$$

where, f_{io1x} and f_{io1y} represent the components of the repulsive force f_{io1} in the coordinate system xoy, and f_{io2x} and f_{io2y} represent the components of the repulsive force f_{io2} in the coordinate system xoy. These components can be all obtained by equation (18).

Finally, only considering the deflection force in the X-axis direction and ignoring the resistance in the Y-axis direction, the final motion direction f_{ri} of the robot is expressed as follows:

$$\begin{cases} f_{iXY} = f_{io1X} + f_{io2X} \\ f_{ri} = f_{iXY} + f_{ac} \\ v_{ri}(t+1) = v_{if}(t) + v_{ie}(t+1) \end{cases} \tag{21}$$

where, $v_{ri}(t+1)$ is the actual speed required by the robot at the next moment, which can be regarded as the vector sum of the robot state update speed $v_{ie}(t+1)$ at time $t+1$ under the framework of the ideal search algorithm and the deflection speed $v_{if}(t)$ of the neighboring obstacles in the X-axis direction.

The Strategy of Robot State Step Update Considering Obstacle Avoidance Constraints

Based on the idea of the SVFM combined with the ideal search algorithm framework with SVFM, the step size update strategy of the robot in different search states is given in this section. When the robot is in the roaming search state, $r_{state} = 0$; when the robot is in the coordinated search state, $r_{state} = 1$. The speed step update strategy of swarm robots in different states is as follows:

$$\begin{cases} v_{ri}(t+1) = V_{ic}(t+1), r_{state} = 0 \cap d_{ij} \succ d_a \\ v_{ri}(t+1) = V_{ie}(t+1), r_{state} = 1 \cap d_{ij} \succ d_a \\ v_{ri}(t+1) = V_{ic}(t+1) + v_{if}(t), r_{state} = 0 \cap d_{ij} \le d_a \\ v_{ri}(t+1) = V_{ie}(t+1) + v_{if}(t), r_{state} = 1 \cap d_{ij} \le d_a \end{cases} \tag{22}$$

The position update strategy of swarm robots considering obstacle avoidance constraints is as follows:

$$x^{*}_{ri}(t+1) = x_{ri}(t) + v_{ri}(t+1) \tag{23}$$

The Distributed Neighborhood Communication Mechanism Based on Time-Varying Characteristic 179 Swarm With Restricted Random Line of Sight (RS-TVCS)

The Communication Model Based on RS-TVCS

In biological research, perception and communication between animal groups are often limited by perception distance. For example, when the Ouqiong bird population flies in formation, its individuals can only exchange information with neighboring individuals within its communication radius to form a local communication network. There is a common neighboring individual between two individuals, and they cannot directly communicate and interact. Through sharing the information of common neighboring individuals, it can spread to the individuals outside their neighbors to form a global communication network. Based on this idea, a representation based on distributed neighborhood communication is defined. The communication-based neighborhood of robot i is a set of teammates within a fixed radius dc to the position of robot i, which can be written as (Xue et al., 2009):

$$\Omega(r_i) = \left\{ r_{j\in m, j\neq i}, \left\| x_{ri} - x_{rj} \right\| \le d_c \right\} \tag{24}$$

where Ω is the communication-based neighborhood, m is the number of members in the swarm, and r_i denotes the robot i. x_{ri} and x_{rj} are the spatial positions of robots i and j, robots respectively. d_c is the maximum communication radius.

During swarm moving, the neighborhoods may change over time, causing the whole swarm to be divided into several dynamically changing sub-swarms. Xue et al. defined those sub-swarms with the concept of Time-Varying Characteristic Swarm (TVCS). The TVCS of robot i at time t can be represented as follows (Junior and Nedjah, 2016):

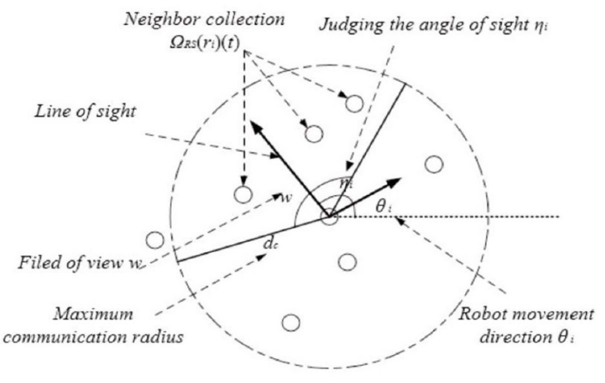

Neighbor collection
$\Omega_{RS}(r_i)(t)$
Judging the angle of sight η_i
Line of sight
η_i
w
θ_i
d_c
Filed of view w
Maximum communication radius
Robot movement direction θ_i

FIGURE 3 | Schematic diagram of individual neighbor collection.

$$\Omega(r_i)(t) = r_i \cup \left\{ r_{j\in m, j\neq i}, \left\| x_{ri}(t) - x_{rj}(t) \right\| \le d_c \right\} \tag{25}$$

where $\Omega(r_i)(t)$ represents the TVCS of robot i. The number of members in a TVCS is dynamically changing, i.e., ri can only able to communicate with other agents in $\Omega(r_i)(t)$ at the time t. Taking into account the limited field of view in the robot signal interaction process, Yang et al. (2019) defined a notation of visual limited TVCS (V-TVCS), which can be written as:

$$\Omega_v(r_i)(t) = r_i \cup \left\{ r_{j\in m, j\neq i}, \left\| x_{ri}(t) - x_{rj}(t) \right\| \le d_c \wedge \varphi_{i,j} \le \frac{\omega}{2} \right\} \tag{26}$$

where $\Omega_v(r_i)(t)$ is the V-TVCS. ω is the single of view of i-th robot, and its sight range is generally set to $\phi_{i,j} \epsilon (0, 2\pi]$. $\phi_{i,j}$ is the sight judgment vector of robots i and j, which is expressed as follows:

$$\varphi_{i,j} = \left\langle r_{ij}(t), v_{ri}(t) \right\rangle \tag{27}$$

where, $r_{ij}(t)$ is the location vector of robots i and j, $v_{ri}(t)$ is the speed vector of the ith robot, and $\langle r_{ij}(t), v_{ri}(t) \rangle$ is the angle between vectors $r_{ij}(t)$ and $v_{ri}(t)$.

Since the line of sight of the robot is not always in the direction of its speed in the process of motion, it is assumed that the line of sight of individual robots changes randomly along the direction of movement in this paper and that the change law obeys the normal distribution, namely, $\eta \sim N(0, \sigma^2)$, where σ is the standard deviation of the deflection angle of the line of sight, and the mean value is 0, indicating that the probability of the individual going straight ahead is greater than that of information interaction to the diagonal side. Considering the limitation of the random line of sight of the robot, the relationship structure diagram of the neighborhood distributed neighborhood communication based on RS-TVCS designed in this paper is shown in **Figure 3**.

The distributed neighborhood communication mechanism based on RS-TVCS is defined as followed:

$$\Omega_{RS}(r_i)(t) = r_i \cup \left\{ r_{j\in m, j\neq i}, \left\| x_{ri}(t) - x_{rj}(t) \right\| \le d_c \wedge \varphi_{i,j} \le \frac{\omega}{2} \right\} \tag{28}$$

where, the expression of $\phi_{i,j}$ is as follows:

$$\varphi_{i,j} = \langle r_{ij}(t), LOS_{ri}(t) \rangle \tag{29}$$

where $LOS_{ri}(t)$ is the vector of line of sight. When $LOS_{ri}(t) = v_{ri}(t)$, it indicates that the line of sight of the robot is consistent with its moving direction. Therefore, the V-TVCS distributed communication mechanism based on V-TVCS is only a special case of RS-TVCS. RS-TVCS has better scalability and practicability than V-TVCS.

RS-TVCS Distributed Network Connected Subset Judgment Based on BFS Algorithm

The global communication network based on the RS-TVCS will change with the dynamic migration of swarms. Under the ideal search algorithm framework, it will iteratively change with the position of the robots, which will make it impossible for some robots to interact with each other, thus forming connected subgroups. Therefore, based on graph theory, assuming that the position of each robot at a certain moment represents a dynamic node, the connected subgroup of each robot is determined based on the idea of the breadth first search (BFS) algorithm. Through this algorithm, the interactive information of each robot under the entire global communication network based on RS-TVCS can be obtained, so as to realize the coordinated search of swarm robots.

The specific ideas are as follows:

1) Taking the position of the robot at time *t* as the node, the weight matrix d_{ij} is constructed by using the distance between the two points as follows:

$$d_{i,j} = \begin{bmatrix} 0 & d_{1,2} & \cdots & d_{1,j} & \cdots & d_{1,m} \\ \vdots & 0 & \vdots & \vdots & \vdots & \vdots \\ d_{m,1} & d_{m,2} & \cdots & d_{m,j} & \cdots & 0 \end{bmatrix} \tag{30}$$

2) Through the neighborhood judgment conditions of equations (28) and (29), the neighborhood weight matrix is constructed. When the neighborhood judgment conditions are not satisfied between the robots i and j, the weight between the two robots (nodes) is 0; otherwise, the weight between the two robots (nodes) is Euclidean distance value.

3) Based on the idea of the BFS (Awerbuch and Gallager, 1987; Jia et al., 2008; Wang et al., 2020) algorithm, all the connected nodes of the neighborhood weight matrix are found to obtain the neighborhood communication information of each robot in the global network.

The Flow of Multi-Target Search Algorithm Swarm Robots Considering Practical Constraints

Under the ideal multi-target search algorithm framework, the distributed communication problem in the search environment is combined with the real-time obstacle avoidance problem. The flow chart of the multi-target search algorithm for swarm robots considering practical constraints (i.e., MSRCPC) is shown in **Figure 4**.

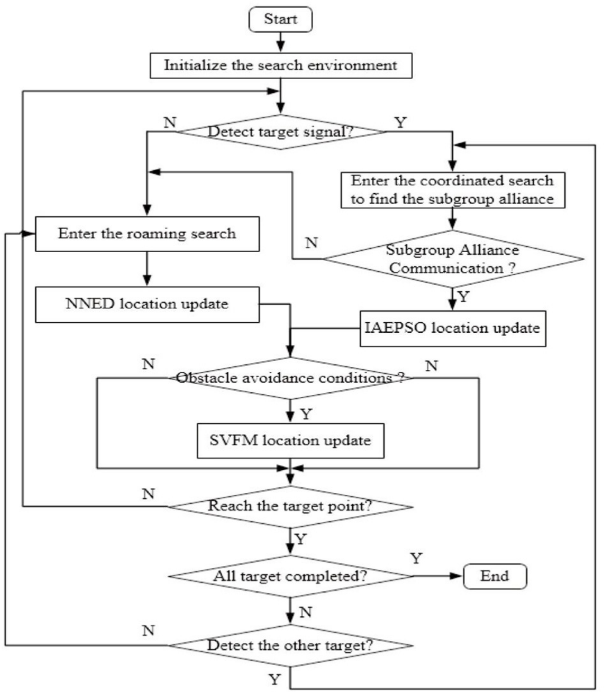


FIGURE 4 | MSRCPC algorithm flow diagram.

TABLE 2 | The table of MSRCPC algorithm parameter.

Symbol	Symbolic meaning	Parameter value
m	Swarm robotics	10–100
n	Search target	1–10
s	Signal attenuation factor	0.1
Q	Constant power signal	10,000
d0	Sensor maximum detection distance	100
Vm	Robot maximum speed	10
α	Inertia coefficient	0.4
δ	Step size control factor	0.6
dl	Adapted distance threshold	100
k1	Obstacle avoidance parameter	0.8
da	Obstacle avoidance distance	80
dc	Neighborhood communication distance	100
w	Robot sight range	150

The main sub-algorithms involved in the proposed algorithm include NNED roaming search algorithm, IAEPSO coordinated search algorithm, TRT multi-target task assignment, obstacle avoidance algorithm based on SVFM, and distributed communication algorithm based on RS-TVCS. The entire algorithm framework basically considers all the problems in the search process of swarm robots, which greatly enhances the scalability and usability of the algorithm.

SIMULATION

In this section, the proposed MSRCPC algorithm has been verified by several experiments in Matlab2019a. First, the

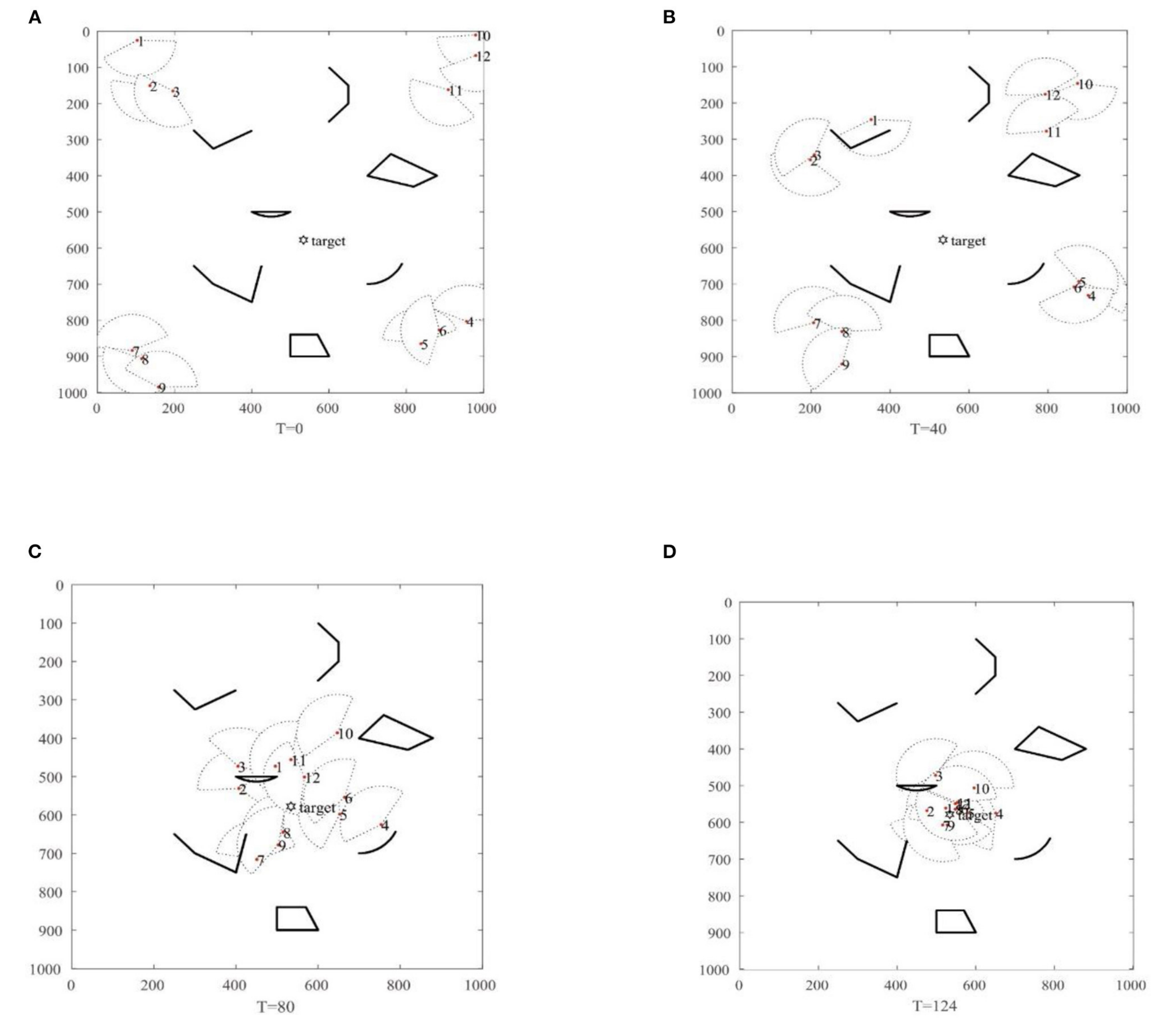


FIGURE 5 | The figure of MSRCPC single-target search. **(A)** T = 0, **(B)** T = 40, **(C)** T = 80, **(D)** T = 124.

MSRCPC algorithm is described in detail by simulating the search behavior of swarm robots in single-target environments and multi-target environments. Then, four multi-target search comparison modes are set up, and the simulation tests are carried out 30 times by using different modes simulation tests 30 times under different group sizes. The effectiveness of the MSRCPC algorithm is verified by comparing and analyzing the simulation results.

The MSRCPC Algorithm Test

In this part, the basic parameter settings of the MSRCPC algorithm are shown in **Table 2**.

With constant basic parameters, the algorithm is applied to single-target and multi-target simulation environments. In view of the randomness of the algorithm, an algorithm search process is randomly recorded to describe the search mechanism and show the performance of the algorithm in detail.

The Single-Target Search Test in Unknown Complex Environments

The initial environment settings of the swarm robotics single-target search simulation for swarm robots are shown in **Figure 5A**. As shown in **Figure 5A**, at $T = 0$, swarm robots are distributed in the corners of the search space, represented by red dots. The position of the target to be searched is set in the middle of the search space, represented by a black regular hexagon. The various black shapes in the figure represent obstacles in the search space. For the robot, the maximum speed is 10, the

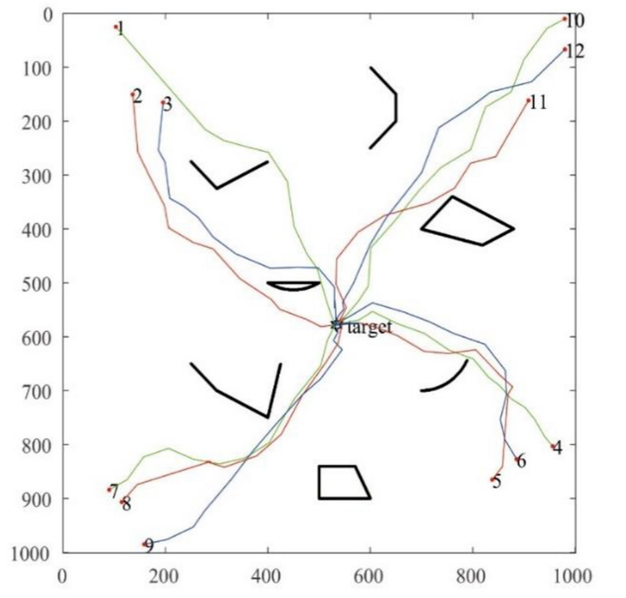


FIGURE 6 | MSRCPC single-target search path simulation diagram.

direction of its initial speed is random, the communication range is limited to 150 degrees, and the direction of moving speed is inconsistent with the direction of the line of sight, and meets the communication conditions of robot in RS-TVCS. Since the robot does not detect the target signal at the initial moment, the NNED algorithm is used to perform random search and diffusion. When $T = 40$, the robot still does not detect the target signal point, and the NNED algorithm continues to be used to randomize, as shown in **Figure 5B**. As shown in **Figure 5C**, at $T = 80$, the No. 1 robot detects the target signal, and then based on the RS-TVCS algorithm proposed in this paper, the number of the robots is learned that can communicate, and the group communication is conducted to form sub swarms. The robot that can detect the target signal through group information sharing switches from the roaming search state to the coordinated search state, and uses the IAEPSO algorithm to coordinate the search for the target point. Finally, as shown in **Figure 5D**, at $T = 128$, the robots numbered 5, 6, 7, 8, 9, 11, and 12 basically converge to the target point, and the target search is successful. The simulation search process with the MSRCPC algorithm can basically be divided into two stages: roaming search and coordinated search.

The search path of the robot recorded in this single-target simulation is shown in **Figure 6**, and it can be seen that the MSRCPC algorithm can not only search for targets quickly and accurately, but also can intelligently avoid obstacles, and has good cluster avoidance performance.

The Multi-Target Search Test in Unknown Complex Environments

Given that the initial number of robots is 30 and the number of targets is 5, other algorithm parameters are consistent with those of the single-target search algorithm in the previous section. The specific simulation search process is shown in **Figures 7A–D**.

In **Figure 7A**, at $T = 0$, the drone swarm is randomly distributed at 200 × 200 unit positions in the search space, and the target points are randomly distributed in the 1,000 × 1,000 search space. The black irregular shape represents the obstacles in the search environment, and the proposed RS-TVCS method is adopted by the robot group to communicate. Based on the RS-TVCS communication rules, using the BFS algorithm, it can be known that at $T = 0$, the 30 robots are neighbors and can maintain information sharing. As shown in **Figure 1**, the 30 fan-shaped shared areas of the robots are group global communication area of the robot group. The robot does not detect the target signal in the global communication area, and the robot is in a roaming search state, that is, it uses the NNED algorithm to perform a roaming search at its maximum speed.

In **Figure 7B**, when $T = 87$, some robots detect the No. 2 target signal and the No. 3 target signal. At this time, the robots in the RS-TVCS global communication neighborhood share local information, and then perform target assignment based on the TRT model to form a subgroup alliance and enter the coordinated search state. However, the robots that fail to communicate with their subgroups continue to maintain the roaming search state and perform roaming searches. In addition, the No. 4 robot detects the No. 2 target signal and the No. 13 robot detects the No. 3 target signal. Based on the RS-TVCS neighborhood communication algorithm, it can be seen that the robots 6, 8, 9, 11, 17, 21, 24, and 29 that share information with the No. 4 robot form a subgroup alliance. Their state changes to the coordinated search state, and then a collaborative search will be conducted on the No. 3 target. In the same way, the No. 29 robot that shares information with the No. 13 robot forms a subgroup alliance, and then performs an accurate collaborative search on the No. 3 target. Since the remaining robots cannot communicate with the two subgroup alliances, or detect the target signal, they continue to maintain the roaming search state for random diffusion.

As shown in **Figure 7C**, when the MSRCPC algorithm iterate to $T = 123$, the subgroup alliances that perform a coordinated search on the targets No. 2 and No. 3 converge to the vicinity of targets No. 2 and No. 3, respectively, and the search for targets No. 2 and No. 3 succeeds. At the same time, the search target information disappears, and the subgroup alliance is disbanded. The formation of robot is the No. 9 robot and the No. 26 robot detect the signal of the No. 1 target and the No. 5 target, respectively. Similarly, according to the solution of the RS-TVCS distributed communication model, it can be seen that 8 robots (2, 18, etc.), which can share information with the No. 9 robot form a subgroup alliance to conduct a collaborative search for the No. 1 target, while the No. 26 robot that fails to interact with other robots cannot obtain communication and maintains a coordinated search alone. In addition, the remaining individual robots that fail to communicate with the target groups No. 1 and No. 5 continue to roam and search using the NNED algorithm.

Finally, as shown in **Figure 7D**, at $T = 186$, the robots successfully detect the No. 4 target, and the search of swarm robots ends.

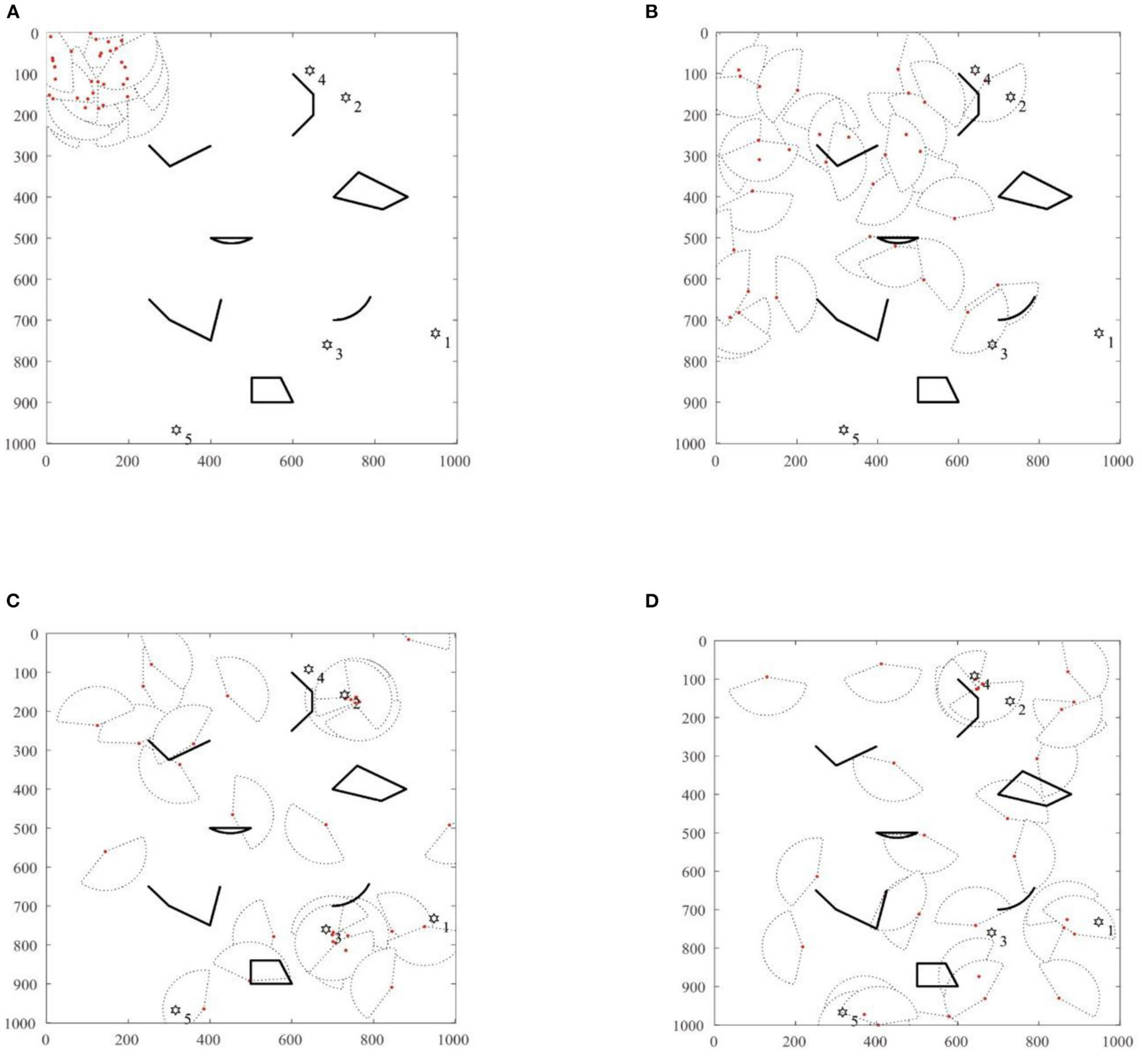


FIGURE 7 | The figure of MSRCPC multi-target search. **(A)** T = 0, **(B)** T = 87, **(C)** T = 123, **(D)** T = 186.

TABLE 3 | The four search algorithm mode table.

Mode	Task allocation	Roaming search	Coordinated search	Obstacle avoidance	Distributed communication
Mode1	ITRT	NNED	EPSO	SVFM	v–TVCS
Mode2	ITRT	NNED	IABA	SVFM	v-TVCS
Mode3	ITRT	NNED	IAEPSO	SVFM	v-TVCS
Mode4	ITRT	NNED	IAEPSO	SVFM	RS-TVCS

The Simulation Analysis of the MSRCPC Algorithm

In the test of the single-target and multi-target search process, the MSRCPC algorithm proposed in this paper has the following advantages. (1) The search process of the algorithm mainly includes roaming search processes and coordinated search processes. In the roaming search process, the robot cannot obtain the prior information of the target, and spreads the search space at the fastest speed; in the coordinated search

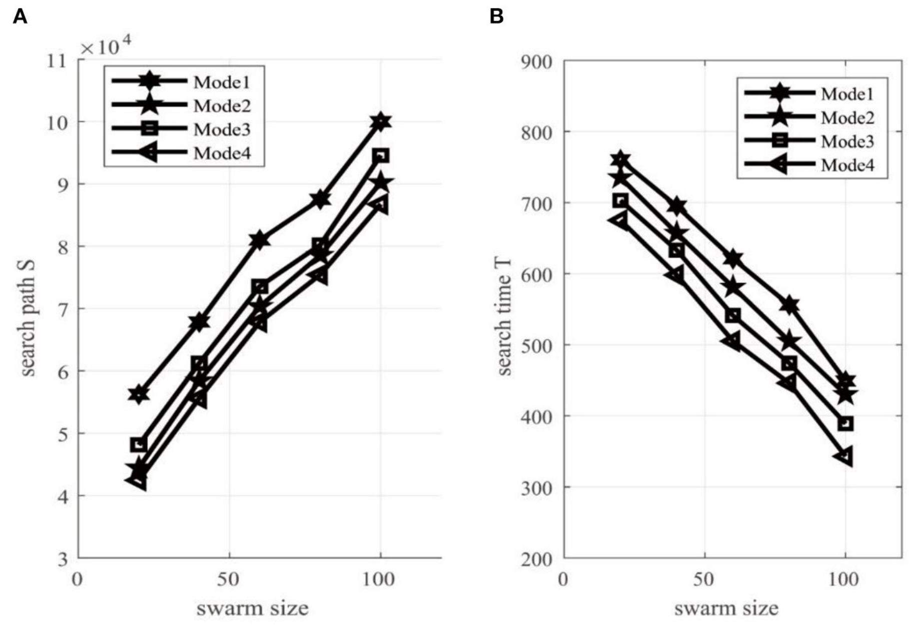


FIGURE 8 | (A) Searching time *T* and **(B)** total energy consumption S of the swarm robotics system.

TABLE 4 | System performance comparison statistics table of four search modes.

Swarm size	**Search**		***T***			***S***	
	model	**Max**	**Mean**	**Min**	**Max**	**Mean**	**Min**
20	Mode1	845	771.234	765	6.091E + 04	5.876E + 04	5.491E + 04
	Mode2	786	763.541	698	4.871E + 04	4.791E + 04	4.469E + 04
	Mode3	754	737.421	712	4.452E + 04	4.912E + 04	5.367E + 04
	Mode4	707	675.741	674	4.087E + 04	4.178E + 04	4.619E + 04
40	Mode1	746	698.39	678	7.189E + 04	6.654E + 04	6.291E + 04
	Mode2	645	645.48	610	5.908E + 04	5.769E + 04	5.491E + 04
	Mode3	631	654.31	631	6.598E + 04	6.235E + 04	5.561E + 04
	Mode4	607	587.431	571	5.798E + 04	5.668E + 04	5.247E + 04
60	Mode1	639	619.361	591	8.271E + 04	8.018E + 04	7.789E + 04
	Mode2	579	550.189	547	6.154E + 04	5.789E + 04	5.554E + 04
	Mode3	619	576.981	539	7.981E + 04	7.467E + 04	7.086E + 04
	Mode4	520	504.861	476	7.234E + 04	6.967E + 04	6.431E + 04
80	Mode1	581	543.187	538	8.913E + 04	8.761E + 04	8.531E + 04
	Mode2	489	471.67	468	8.318E + 04	7.971E + 04	7.689E + 04
	Mode3	549	518.60	471	8.241E + 04	7.987E + 04	7.618E + 04
	Mode4	459	449.356	423	8.089E + 04	7.709E + 04	7.136E + 04
100	Mode1	471	451.61	406	1.109E + 05	9.971E + 04	9.012E + 04
	Mode2	389	368.071	319	1.012E + 04	9.456E + 04	8.956E + 04
	Mode3	397	369.178	365	9.780E + 04	9.438E + 04	9.129E + 04
	Mode4	368	326.678	306	9.109E + 04	8.754E + 04	8.497E + 04

process, by obtaining the target information, the robots are determined by the RS-TVCS communication interaction model in the global communication range, and then the sub-group alliances approach the optimal position of the target point based on the group optimal information and individual optimal information in the IAEPSO algorithm. (2) Self-organization and adaptability are embodied in the process of the target search of swarm robots. In the process of target searching, swarm robots adaptively transform their own state by acquiring information of the external environment or sharing local information and participating in task collaboration. (3) The intelligence of swarm robots in the target search process is also reflected. In the process of roaming search and coordinated search, individual robots can realize intelligent obstacle avoidance by sensing the information of the external environment and successfully avoiding obstacles. In order to verify the performance of the MSRCPC algorithm, a series of comparative experiments are carried out in the next part.

Comparison and Discussion of MSRCPC Algorithm Simulation

In this part, the four sets of comparison modes are set up to further verify the superiority of the MSRCPC algorithm based on the multi-target search framework of the finite state machine. The settings of the four comparative search models are shown in **Table 3**.

Based on its framework, the search algorithm is divided into the following five parts, namely, multi-target task allocation model, roaming search algorithm, coordinated search algorithm, cluster obstacle avoidance, and distributed communication model.

For Mode 1, the NNED algorithm is adopted for roaming search, the traditional TRT model is used to assign tasks to the target, the EPSO algorithm proposed in Gudise (2004) is applied to coordinated search, the 2D-SVFM model (Xinjie, 2020) is applied to group obstacle avoidance, and the V-TVCS model proposed in Yang et al. (2019) is used for robot communication. For Mode 2, the IABA algorithm proposed in Tang et al. (2020) is applied to the robot coordinated search, and the other sub-algorithms remain constant. For Mode 3, the proposed IAEPSO algorithm is applied to the robot coordinated search, and the remaining sub-algorithms remain unchanged. For Mode 4, the proposed MSRCPC algorithm is used to set up the search experiment.

When the number of targets in the search environment of swarm robots is 10, by changing the number of swarm robots, these four modes were used to conduct 30 simulation search experiments. The change of the search path S and the mean value of the search time T of the swarm robots with the population number is shown in **Figure 8**; **Table 4**.

It can be seen from **Figure 8**; **Table 4** that when the number of constant search targets is 10, as the scale of the swarm robots increases, the search time of the swarm robot system will decrease, but the system energy consumption of the swarm robots will increase. Therefore, we are surprised to find that how to balance the search time and energy consumption of the entire system by balancing the scale of swarm robots is a basic problem in the practical application of swarm robot systems.

By comparing Mode 3 with Mode 1 and Mode 2, it can be seen that in the entire multi-target search framework, the proposed IAEPSO coordinated search algorithm has better performance than EPSO and IABA algorithms in different population sizes. The main reason is that the adaptive inertia weight set by the IAEPSO algorithm can satisfy the coordinated search behavior of the robot. However, when the target signal exceeds a certain threshold, the robot has a larger inertia weight and can conduct a large-scale coordinated search. When the target signal is less than a certain threshold, in order to avoid the robot oscillating around the target, the algorithm can adaptively adjust the motion behavior of the robot to avoid the oscillation of the path, thereby reducing system energy consumption.

From the performance comparison curves of Mode 4 and Mode 3, it can be seen that in the case of the other sub-algorithms being the same, the search performance of swarm robots using the RS-TVCS distributed communication algorithm is better than that of Mode 3. The main reason is that the RS-TVCS distributed neighborhood communication model can meet the communication interaction performance of actual swarm robots. Using the RS-TVCS model in the process of forming subgroup alliances will make the configuration of the robot members within each member more reasonable, which can greatly improve the utilization of robot members and provide a more efficient search for the entire algorithm framework.

All in all, compared with the first three modes, the search performance of the swarm robotics can be improved by at least 25 by using the proposed MSRCPC algorithm (Mode 4).

CONCLUSION

The multi-target search problem of swarm robots in unknown complex environments is studied in this paper. The main innovations are as follows. (1) Aiming at the target search problem of swarm robots in actual environments, a target search framework based on a finite state machine is proposed. The proposed framework can not only solve the single-target search problem, but also solve the multi-target search problem, which improves the applicability of this algorithm in actual search scenarios. (2) In this algorithm, the problem of cluster obstacle avoidance is considered as a problem in the actual search environment, and the intelligence of cluster search for the robot is reflected. (3) In order to solve the distributed communication interaction problem in the unknown environments, by considering the random communication between individual robots and the limited visual area, a RS-TVCS model is proposed, which overcomes the shortcomings of the V-TVCS communication model.

Simulation analysis and comparison experiments show that this proposed algorithm has good search performance and strong scalability and stability, and can adapt to any search environment. In addition, we find, surprisingly, that the balance of search performance of the swarm robot system is related to the number of swarm robots. Therefore, how to balance the search path and search time of swarm robot systems by setting a certain number of swarm robots is the focus of further research.

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/s.

AUTHOR CONTRIBUTIONS

YZ: conceptualization, methodology, and writing—original draft preparation. AC: methodology, software, investigation, and writing—reviewing and editing. XH and XB: software and writing—reviewing and editing. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported in part by the National Defense Basic Research Program of China under Grant JCKY2019403D006, the Outstanding Youth Project of the Education Department of Hunan Province of China under Grant 19B200, the Doctoral Scientific Research Initial Funds of the Human University of Science and Technology under Grant E56126, and the Special Project of Engineering Research Center (Item No: Lgy18gz006).

REFERENCES

Alfeo, A. L., Cimino, M. G. C. A., and Vaglini, G. (2019). Enhancing biologically inspired swarm behavior: Metaheuristics to foster the optimization of uavs coordination in target search. *Comput. Operat. Res.* 110, 34–47. doi: 10.1016/j.cor.2019.05.021

Alsamman, A. (2011). Arizona, all seven of the graduating seniors on the robotics team at an inner-city high school in phoenix will be attending college this year on full scholarships. *Optics Lett*. 36, 645–647. doi: 10.1364/OL.36.000645

Anonymous (1993). Innova holdings and mesa robotics team up to integrate their products for military, homeland security and law enforcement arenas. *Urology*. 41, 384–386.

Awerbuch, B., and Gallager, R. G. (1987). A new distributed algorithm to find breadth first search trees. *IEEE Trans. Inform. Theory*. 33, 315–322. doi: 10.1109/TIT.1987.1057314

Bénichou, O., Loverdo, C., Moreau, M., and Voituriez, R. (2006). Two-dimensional intermittent search processes: an alternative to lévy flight strategies. *Physic. Rev*. 74:020102. doi: 10.1103/PhysRevE.74.020102

Berscheid, L., Meibner, P., and Kroeger, T. (2020). Self-supervised learning for precise pick-and-place without object model. *IEEE Robot. Automat. Lett*. 99, 1–1. doi: 10.1109/LRA.2020.3003865

Bonabeau, E. (1999). Swarm intelligence: from natural to artificial systems. *Santa Fe Inst Stud ences Complex*. 9, 32–77. doi: 10.1093/oso/9780195131581.001.0001

Booth, K. E. C., Piacentini, C., Bernardini, S., and Beck, J. C. (2020). Target search on road networks with range-constrained uavs and ground-based mobile recharging vehicles. *IEEE Robot. Automat. Lett*. 99, 1–1. doi: 10.1109/LRA.2020.3015464

Cizek, P., and Faigl, J. (2019). Self-supervised learning of the biologically-inspired obstacle avoidance of hexapod walking robot. *Bioinspirat. Biomimetics*. 14:046002. doi: 10.1088/1748-3190/ab1a9c

Doty, K. L., and Van Aken, R. E. (2002). "Swarm robot materials handling paradigm for a manufacturing workcell," in *IEEE International Conference on Robotics and Automation*. New York, NY: IEEE.

Ducatelle, F., Di, G. A., Pinciroli, C., Mondada, F., and Gambardella, L. M. (2011). Communication assisted navigation in robotic swarms: Self-organization and cooperation. *IEEE/RSJ Int. Conf. Intell. Robots Syst.* 484:9. doi: 10.1109/IROS.2011.6094454

Goodrich, M. A., Morse, B. S., Engh, C., Cooper, J. L., and Adams, J. A. (2009). Towards using unmanned aerial vehicles (uavs) in wilderness search and rescue: Lessons from field trials. *Interact. Stud*. 10, 453–478. doi: 10.1075/is.10.3.08goo

Gudise, S. D. V. (2004). Optimal pso for collective robotics search applications. *IEEE Trans. Evol. Comput.* 2:15. doi: 10.1109/CEC.2004.1331059

Ha, I. K., and Cho, Y. Z. (2018). A probabilistic target search algorithm based on hierarchical collaboration for improving rapidity of drones. *Sensors*. 18:8. doi: 10.3390/s18082535

Jia, Y., Andrieu, C., Piechocki, R., and Sandell, M. (2008). Depth-first and breadth-first search based multilevel sga algorithms for near optimal symbol detection in mimo systems. *IEEE Trans. Wireless Commun*. 7, 1052–1061. doi: 10.1109/TWC.2008.060813

Jie, T. Y. L. (2019). A probabilistic finite state machine based strategy for multi-target search using swarm robotics. *Appl. Soft Comput*. 77, 467–483. doi: 10.1016/j.asoc.2019.01.023

Jiong, L. I., Jianshi, L. I., Feng, M., and Zhu, Y. (2019). Research on multi-target tracking method based on improved evolutionary hungarian algorithm. *J. Military Transport. Univ*. 8:298. doi: 10.3389/fbioe.2020.00298

Junior, L. S., and Nedjah, N. (2016). Efficient strategy for collective navigation control in swarm robotics. *Procedia Comput. Sci.* 80, 814–823. doi: 10.1016/j.procs.2016.05.371

Kamegawa, T., Akiyama, T., Sakai, S., Fujii, K., Une, K., Ou, E., et al. (2020). Development of a separable search-and-rescue robot composed of a mobile robot and a snake robot. *Int. J. Adv. Robotics*. 34, 132–139. doi: 10.1080/01691864.2019.1691941

Khan, Z. A., Awais, M., Alghamdi, T. A., Khalid, A., Fatima, A., Akbar, M., et al. (2019). Region aware proactive routing approaches exploiting energy efficient paths for void hole avoidance in underwater wsns. *IEEE Access*. 19:39155. doi: 10.1109/ACCESS.2019.2939155

Liang, G. Q., Kang, Y. H., Xing, Z. C., and Yin, G. Y. (2018). Uav cooperative multi-task assignment based on discrete particle swarm optimization algorithm. *Comput. Simulat*. 2, 81–86. doi: 10.3969/j.issn.1006-9348.2018.02.005

Luo, T., Liang, S., Zeyin, H. E., and Zhang, X. (2017). Path planning of robot based on glowworm swarm optimization algorithm of scene understanding. *J. Comput. Applicat*. 17:12. doi: 10.11772/j.issn.1001-9081.2017.12.3608

Majid, M., and Arshad, A. M. (2017). An analysis of pso inertia weight effect on swarm robot source searching efficiency. *IEEE* 447, 173–178. doi: 10.1109/I2CACIS.2017.8239053

Manic, K. D. (2009). Multi-robot, multi-target particle swarm optimization search in noisy wireless environments. *Conf. Hum. Syst. Interact.* 8, 76–87. doi: 10.1109/HSI.2009.5090958

Marzaeva, V. I. (2019). Mathematical modeling of canopy forest fire spread in the presence of fire breaks and barriers. *Technic. Phys*. 64, 1073–1081. doi: 10.1134/S1063784219080139

Moshayedi, A. J., and Gharpure, D. (2013). Implementing breath to improve response of gas sensors for leak detection in plume tracker robots. *Adv. Intell. Syst. Comput*. 259:13. doi: 10.1007/978-81-322-1768-8_31

Ryan, M. R. K. (2008). Exploiting subgraph structure in multi-robot path planning. *J.Artific. Intell. Res*. 31, 497–542. doi: 10.1613/jair.2408

Suzuki, K., Yokota, Y., Kanazawa, Y., and Takebayashi, T. (2020). "Online self-supervised learning for object picking:Detecting optimum grasping position using a metric learning approach," in *2020 IEEE/SICE International Symposium on System Integration* (SII).

Tang, H., Sun, W., Yu, H., Lin, A., and Xue, M. (2020). A multirobot target searching method based on bat algorithm in unknown environments. *Exp. Syst. Applic*. 141:112945. doi: 10.1016/j.eswa.2019.112945

Viswanathan, G. M., Buldyrev, S. V., Havlin, S., Luz, M. D., a., Raposo, E., et al. (1999). Optimizing the success of random searches. *Nature*. 401, 911–914. doi: 10.1038/44831

Wang, Y., Shi, K., Zheng, X., You, S., and Wang, N. (2020). Thermo-hydraulic coupled analysis of meshed district heating networks based on improved breadth first search method. *Energy*. 205:117950. doi: 10.1016/j.energy.2020.117950

Xinjie, W. (2020). Multiobjective coordinated search algorithm for swarm of uavs based on 3d-simplified virtual forced model. *Int. J. Syst. Sci*. 51, 367–386. doi: 10.1080/00207721.2020.1799110

Xue, S., Zhang, J., and Zeng, J. (2009). Parallel asynchronous control strategy for target search with swarm robots. *Int. J. Bio-inspired Comput.* 1, 151–163. doi: 10.1504/IJBIC.2009.023811

Yang, J., Wang, X., and Bauer, P. H. (2019). Extended pso based collaborative searching for robotic swarms with practical constraints. *IEEE Access*. 7, 328–76 341. doi: 10.1109/ACCESS.2019.2921621

Yao, J. Y., and Raffuse, S. M., Brauer, G. J., Bowman, D. M., and Johnston, J. S. (2018). Predicting the minimum height of forest fire smoke within the atmosphere using machine learning and data from the calipso satellite. *Rem. Sens. Environ*. 206, 98–106. doi: 10.1016/j.rse.2017.12.027

Zhang, X. J., Zhang, M. L., Zhang, J. H., and Sun, L. Y. (2010). A multi-sensory robot used for searching toxic gas leak source. *Adv. Mater. Res.* 139, 2150–2153. doi: 10.4028/www.scientific.net/AMR.139-141.2150

Zhang, Z. J. C., and Xue, Y. Z. (2014). Dynamic task allocation with closed-loop adjusting in swarm robot search for multiple targets. *Robot*. 36, 57–68. doi: 10.1017/S0263574717000157

Zhang, Z. J. C., and Xue, Y. Z. (2015). Cooperative and coordination in swarm robotic search for multiple targets. *Robot* 37, 142–151.

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.

frontiers in Neurorobotics

METHODS
published: 07 January 2022
doi: 10.3389/fnbot.2021.801956

Underwater Localization and Mapping Based on Multi-Beam Forward Looking Sonar

*Chensheng Cheng, Can Wang, Dianyu Yang, Weidong Liu and Feihu Zhang**

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

SLAM (Simultaneous Localization And Mapping) plays a vital role in navigation tasks of AUV (Autonomous Underwater Vehicle). However, due to a vast amount of image sonar data and some acoustic equipment's inherent high latency, it is a considerable challenge to implement real-time underwater SLAM on a small AUV. This paper presents a filter based methodology for SLAM algorithms in underwater environments. First, a multi-beam forward looking sonar (MFLS) is utilized to extract environmental features. The acquired sonar image is then converted to sparse point cloud format through threshold segmentation and distance-constrained filtering to solve the calculation explosion issue caused by a large amount of original data. Second, based on the proposed method, the DVL, IMU, and sonar data are fused, the Rao-Blackwellized particle filter (RBPF)-based SLAM method is used to estimate AUV pose and generate an occupancy grid map. To verify the proposed algorithm, the underwater vehicle is equipped as an experimental platform to conduct field tasks in both the experimental pool and wild lake, respectively. Experiments illustrate that the proposed approach achieves better performance in both state estimation and suppressing divergence.

Keywords: SLAM, multi-beam forward looking sonar, point cloud, grid map, underwater vehicle

OPEN ACCESS

Edited by:
Chen Qiao,
Xi'an Jiaotong University, China

Reviewed by:
Zhe Zhang,
Taiyuan University of Technology, China
Tian Wang,
Beihang University, China

****Correspondence:***
Feihu Zhang
feihu.zhang@nwpu.edu.cn

Received: *26 October 2021*
Accepted: *15 December 2021*
Published: *07 January 2022*

Citation:
Cheng C, Wang C, Yang D, Liu W and Zhang F (2022) Underwater Localization and Mapping Based on Multi-Beam Forward Looking Sonar. Front. Neurorobot. 15:801956. doi: 10.3389/fnbot.2021.801956

1. INTRODUCTION

AUVs (Autonomous Underwater Vehicles) have been widely applied to perform various complex underwater tasks such as resource exploration (Ohta et al., 2016), environmental monitoring (Williams et al., 2012; Barrera et al., 2018), underwater rescue (Venkatesan, 2016), and military operations (Hagen et al., 2005), etc. To satisfies the safety and reliability, AUVs should acquire accurate localization in underwater unknown environments.

To achieve this goal, Doppler Velocity Logging (DVL) and Inertial Measurement Unit (IMU) are fused with acoustic long baseline (Matos et al., 1999), short baseline (Vickery, 1998), and ultra-short baseline (Hao et al., 2020) to calculate the position of AUVs. However, these traditional methods have shortcomings regarding to error divergence. DVL measures the speed by integrating the acceleration and meanwhile further calculates localization from dead-reckoning, the final result may, therefore, contain cumulative errors; The method based on the acoustic baseline needs to arrange the equipment in the environment in advance; Therefore, it is essential to use a more robust and reliable method to solve above problems. On the other hand, SLAM enables AUVs to fuse sensor data and build a map of an unknown environment, while localizing simultaneously. So far, sensors applied in underwater slam include cameras, side-scan sonar (SSS), single-beam mechanical scanning sonar (SMSS), and multi-beam forward-looking sonar (MFLS).

Camera-based underwater SLAM estimates the ego-motion by extracting and matching features from adjacent images and optimizing the pose at the back-end (Kim and Eustice, 2015; Hong et al., 2016). Jongdae Jung et al. proposed a vision-based SLAM, where artificial underwater landmarks help visualize camera poses (Jung et al., 2017). Suresh et al. proposed a novel method for underwater localization using natural landmarks (Suresh et al., 2019). Sparse features were obtained via an onboard upward-facing stereo camera through water for underwater localization. Although the cost of the camera is low, vision-based underwater SLAM has significant limitations. The camera's detection range is close and can only work in a clean environment with good light.

Compare to cameras, sonar emits sound waves in single or multiple directions and obtains information about the surrounding environment by analyzing each echo's strength and return time. Sonar-based method is, therefore, the development trend of underwater SLAM (Wang et al., 2017; Wang and Cheng, 2020). Chen et al. proposed an RBPF SLAM algorithm to tackle the issues of scan distortion and data sparseness caused by the slow-sampling mechanical scanning sonar, by carefully designed a sliding window-based scan module (Chen et al., 2020). The formed scans are then fed into the modified RBPF to build a consistent grid-based map. Siantidis et al. described a SLAM system with a dead reckoning system and a side-scan sonar (Siantidis, 2016), which can compensate for the position drifts. Aulinas et al. proposed a feature-based sub-mapping SLAM approach, which considered side-scan salient objects as landmarks (Aulinas et al., 2010). However, the long scanning period is quite challenging to meet underwater real-time performance, as the return of the side-scan sonar and mechanical scanning sonar image is delayed.

Meanwhile, MFLS is becoming more and more popular in underwater perception because of its solid real-time performance, small size, and easy installation (Hurtós et al., 2014; Wright and Baldauf, 2016). Wang et al. proposed a novel approach for underwater SLAM using an MFLS for 3D terrain mapping tasks (Wang et al., 2019). Instead of repeatedly projecting extracted features into Euclidean space, they applied optical flow within bearing-range images for tracking extracted features and assumed these features are sampled from a Gaussian Process terrain map. Neves et al. introduced a novel multi-object detection system, which outputs object position and rotation from MFLS images (Neves et al., 2020). Pyo et al. proposed a novel localization method in shallow water, where localization is based on passive-type acoustic landmarks. Through modeling, the distance from landmark to MFLS could be calculated (Pyo et al., 2017). However, a complete occupancy grid map using underwater vehicles with MFLS is still missing.

This paper presents a methodology for the SLAM algorithm based on MFLS, by building an accurate occupancy grid map and providing an accurate estimation of AUV poses. The occupancy grid graph can be used for subsequent global positioning and path planning. The main contributions of the proposed algorithm are in two aspects. (1) Aiming at the slow processing speed caused by a large amount of MFLS image data, a method is proposed to convert the collected sonar image into sparse point cloud format data through threshold segmentation and distance-constrained filtering. (2) Based on the proposed method, the DVL, IMU, and MFLS data are fused, and then the RBPF-based SLAM method is used to generate an accurate occupancy grid map, and at the same time, the drift of the inertial navigation can be suppressed.

The structure of the proposed approach is as follows. Section 2 introduces the characteristics of the MFLS used in this article. The proposed SLAM method for underwater vehicles is detailed in section 3. The experimental results are shown in section 4. Section 5 presents a brief conclusion and section 6 is our future work.

2. PROBLEM DESCRIPTION OF MFLS SLAM

MFLS is an image sonar. It can emit multiple sound waves with a vertical width in the horizontal direction and detect the environment based on the echoes. The working principle is shown in **Figure 1**. However, it has no resolution in the vertical direction, so the result is a two-dimensional image. By measuring the flight time and intensity of the echo, images with different degrees of brightness will be obtained, as shown in **Figure 2**. The bright part indicates an obstacle with high echo intensity, and the dark part indicates that the echo intensity at that location is weak.

Generally, there are two MFLS data processing methods: image-level processing and echo intensity processing. However, due to the dense beams, high resolution, and the relatively large amount of image sonar data, processing directly from the image level will cause a large amount of calculation in the SLAM process, and it is difficult for the processor installed on the small AUV to process the data in real-time. In order to solve this problem, this paper converts the sonar image data into corresponding point cloud data and then uses distance-constrained filtering to extract the necessary information and reduce the amount of calculation. In the case of limited processor performance, the goal of real-time SLAM is achieved on a small AUV.

3. PROPOSED SLAM METHOD

Figure 3 is the overall framework of the SLAM algorithm proposed in this article. We first fuse DVL data and IMU data to obtain odometer data in the algorithm and use it for dead reckoning. At the same time, the MFLS sonar data is preprocessed. First, the sonar data is converted into point cloud data through threshold segmentation and data conversion. The obtained point cloud data is then subjected to sparse processing using distance constraints filtering. Finally, send the processed data into the RBPF-SLAM algorithm for positioning and composition.

3.1. Dead Reckoning

The function of the dead reckoning module is to apply the IMU and DVL to provide a rough estimate of the AUV pose. When the

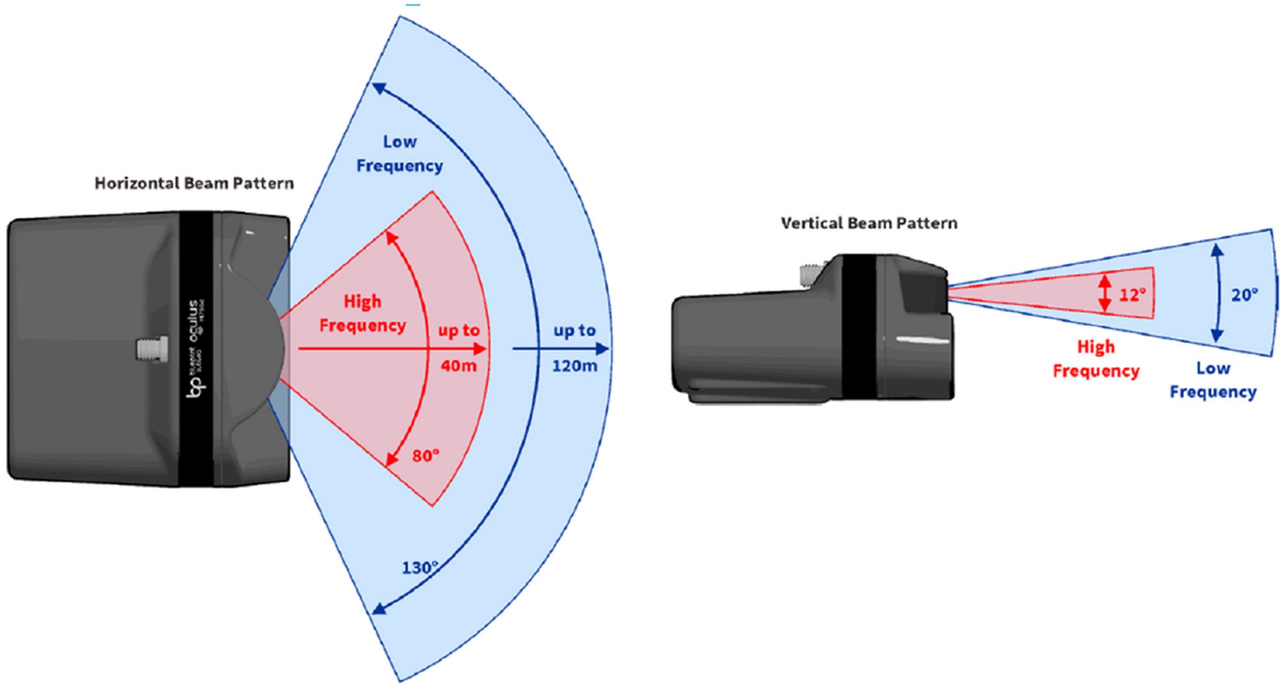


FIGURE 1 | Schematic diagram of MFLS working mode.

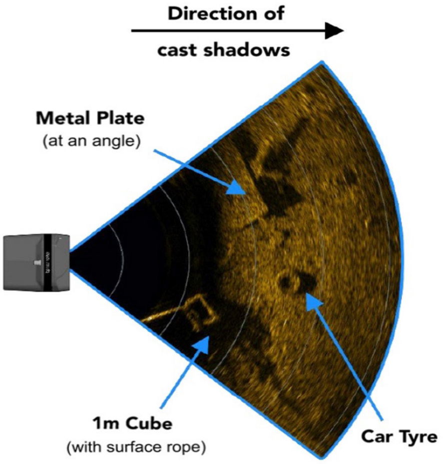


FIGURE 2 | Example of scanning effect of 3 targets. The bright part indicates an obstacle with high echo intensity, and the dark part indicates that the echo intensity at that location is weak.

sonar samples the data, the dead reckoning module applies the extended Kalman filter (EKF) to estimate the attitude by fusing the data from these sensors.

Suppose the AUV estimated state is $X_k = [p_k^T, \varphi_k^T]^T$ in the global coordinate system at time k, where p_k represents the position of the AUV, and φ_k represents the attitude of the AUV. p_k and φ_k are, respectively, defined as

$$p_k = [x_k \quad y_k \quad z_k]^T, \quad \varphi_k = [\phi_k \quad \theta_k \quad \psi_k]^T$$

where x_k, y_k, z_k are the position coordinates in each axis in the global coordinate frame, and ϕ_k, θ_k, ψ_k are the Euler angles roll, pitch, and yaw in each corresponding axis.

Assuming that the linear velocity and angular velocity of the AUV are υ_k and ω_k, they are jointly used as the control input $u_k = [\upsilon_k^T, \omega_k^T]^T$. Specifically, υ_k and ω_k are expressed as

$$\upsilon_k = [l_k \quad m_k \quad n_k]^T, \quad \omega_k = [o_k \quad q_k \quad r_k]^T$$

where each element in the two vectors is the linear velocity and angular velocity on each axis in the AUV coordinate system. Then the kinematics model of AUV can be expressed as

$$X_{k+1} = f(X_k, u_k) = X_k + \Delta_T * J(X_k) * u_k \tag{1}$$

where $J(X_k)$ is the transformation matrix, and Δ_T is the sampling time interval. u_k can be represented by DVL measurement value and IMU measurement value with Gaussian noise ω_k $N(0, Q_u)$. Due to this kind of noise, there will be error accumulation in dead reckoning. Therefore, other sensor information is called to correct the error during the update phase of EKF. Through formula (1), the AUV state can be estimated as

$$\hat{X}_{k+1|k} = f(\hat{X}_{k|k}, \hat{u}_k) \tag{2}$$

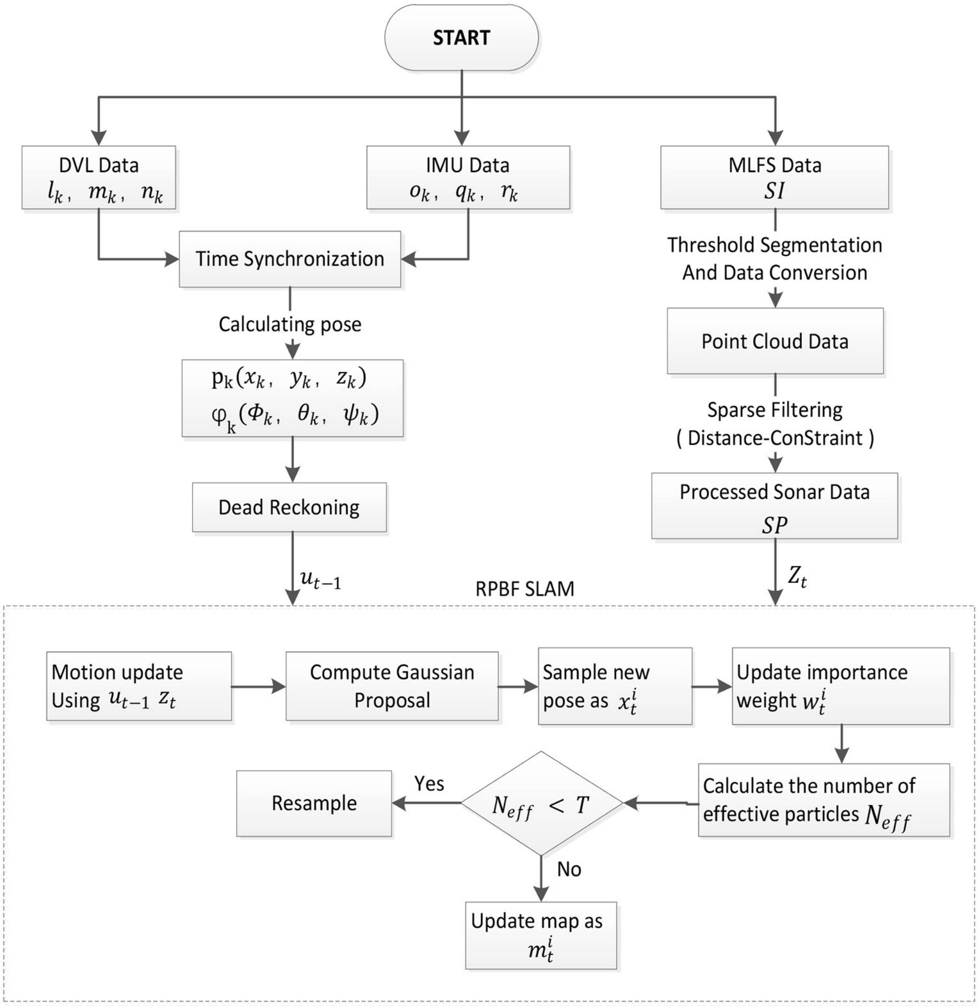

START
DVL Data
l_k, m_k, n_k
IMU Data
o_k, q_k, r_k
MLFS Data
SI
Time Synchronization
Calculating pose
$p_k(x_k, y_k, z_k)$
$\varphi_k(\Phi_k, \theta_k, \psi_k)$
Dead Reckoning
u_{t-1}
Threshold Segmentation And Data Conversion
Point Cloud Data
Sparse Filtering (Distance-ConStraint)
Processed Sonar Data
SP
Z_t
RPBF SLAM
Motion update Using u_{t-1} z_t
Compute Gaussian Proposal
Sample new pose as x_t^i
Update importance weight w_t^i
Calculate the number of effective particles N_{eff}
$N_{eff} < T$
Yes
Resample
No
Update map as m_t^i

FIGURE 3 | Architecture of the proposed SLAM algorithm.

The covariance matrix used for the prediction error can be expressed as

$$P_{k+1|k} = F_{k+1} * P_{k|k} * F_{k+1}^T + G_{k+1} * Q_u * G_{k+1}^T \qquad (3)$$

where F_{k+1} and G_{k+1} are the Jacobian matrices obtained by solving the partial derivative of the nonlinear model function f about the state X_k and the noise ω_k.

Finally, the model prediction is updated by applying the standard EKF update equation to generate the estimated pose of the AUV.

3.2. Threshold Segmentation and Data Conversion

The working principle of sonar is to generate an echo according to the sound wave encountering an object and then generate an image according to the time and intensity of the return of the echo. Due to water quality and acoustic interference, multi-beam sonar data will carry a lot of clutter and outliers in a natural environment. Direct conversion into lidar data for mapping will distort the resulting map. Therefore, it is necessary to filter according to the environment so that the data can better reflect the characteristics of the environment.

In this experiment, the raw sonar data were processed in three steps: threshold segmentation, data conversion, and distance-constrained filtering. The flow chart of the proposed algorithm is presented in **Algorithm 1**.

SI is sonar's original data, an image generated with parameters with a scanning angle of 130° and a scanning distance of $40m$. **Figure 4** is an image generated by aiming at the corner of the pool with sonar, which contains a lot of clutter. $T1$ is the threshold for filtering selection. In general, we use the average pixel value as the threshold for filtering. At the same time, the threshold can also be manually set according to the water quality environment. Under normal circumstances, we put several targets in the water or look for an environment with apparent characteristics in the background. Then, we use sonar to scan in real-time and continuously adjust the threshold manually until the generated

Algorithm 1: Sonar data process.

```
Data: [$SI$] Sonar's raw image data and the average value of
      pixels.
Result: [$SP$] Processed sonar data.
begin
    $SI$ = ReadSonarData();
    //Threshold segmentation and data conversion
    for $Pixel_{(i*j)} \in SI$ do
        if $Pixel_{(i*j)} < T_1$ then
            $Pixel_{(i*j)} = 0$;
        else
            $Pixel_{(i*j)} = Pixel_{(i*j)}$;
        end
    end
    for $k = 0$; $k < i$; $k++$ do
        for $Pixel_{(k*j_{th})} \in Pixel_{(k*j)}$ do
            $Pixel_{(k*j_{th})} = FindMaxValue(Pixel_{(k*j)})$;
        end
        $range_i = j_{th} * range\_step$;
        $angle_i = i * angle\_step$;
        $PointC.PushBack(range_i, angle_i)$;
    end
    //Distance-constrained filtering
    $num = 0, sum = 0$;
    for $p = 0$; $p < i$; $p++$ do
        $num = num + 1$;
        $sum = sum + range_p$;
        if $p\%T_2 == 0 \quad \&\& \quad p! = 0$ then
            $average\_value = sum/num$;
            for $n = p - num$; $n < num$; $n++$ do
                $dif = range_p - average\_value$
                if $dif > T_3$ then
                    $range_p = 0$
                else
                    $range_p = range_p$
                end
            end
            $num = 0, sum = 0$;
        else
        end
    end
    $SP.PushBack(range_p)$
end
```

point cloud data can better reflect the target profile. At the same time, when watching open waters, less noise is generated, and the current threshold is selected as the optimal threshold in the current environment.

The pixel value below $T1$ is assigned a value of 0, and the pixel value above $T1$ remains unchanged. Since only the features of the surface of the object are considered when constructing the map, we believe that the brightest point on the beam is formed by the sound wave hitting the surface of the object and returning. Therefore, we only select a point with the highest intensity on each beam as the target of interest. After preliminary filtering, we calculate the position of the target pixel in the sonar coordinate system according to the angular resolution *angle_step* and distance resolution *range_step* of the sonar. The resolution value can be changed by setting the sonar parameters. Calculating all the pixels of the sonar image, we can get the point cloud data, as shown in **Figure 5A**.

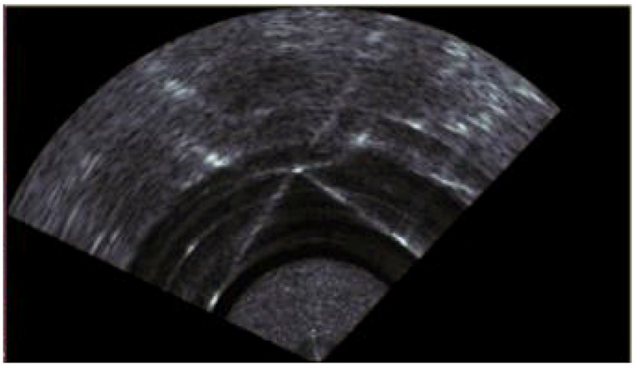


FIGURE 4 | Sonar raw data. This is the raw data picture of the sonar facing the corner of the experimental pool.

3.3. Distance-Constrained Filtering

From **Figure 5A**, we can find that many bright spots are generated behind the wall of the pool. This is not the data we want, and it will affect the positioning accuracy and mapping effect of the AUV. Aiming at these clutter interference, this paper uses a distance constraint-based method to filter out clutter while reducing the amount of data. Doing so can improve the real-time performance of the algorithm while ensuring positioning accuracy and mapping quality. The flow chart of this algorithm is shown in the lower part of **Algorithm 1**.

Process the obtained point cloud data *PointC*:

- Set a beam threshold $T2$, and divide all range data $range_i$ into $A = \{[range_{(0)} - range_{(T2-1)}], [range_{(T2)} - range_{(2*T2-1)}].....[range_{(i-T2-1)} - range_{(i)}]\}$ according to the angle order($-65° - 65°$).
- Set the distance threshold $T3$, calculate the average of all $range_i$ in A_k, and then calculate the difference dif between each $range_i$ and the average. If dif is greater than $T3$, the data is judged to be noise deleted, if it is less than $T3$, it is judged to be valid data.
- Save the valid data into the SP, and the sonar data processing ends.

Figure 5B is the point cloud image obtained after distance constraint filtering. It can be found that the clutter behind the wall of the pool is successfully filtered, and data that can truly reflect the environmental characteristics are obtained.

3.4. RBPF SLAM With MFLS

The flow chart of the proposed algorithm is presented in **Algorithm 2**. PS is the particle set generated according to the initial state of the AUV. According to the theory that the joint

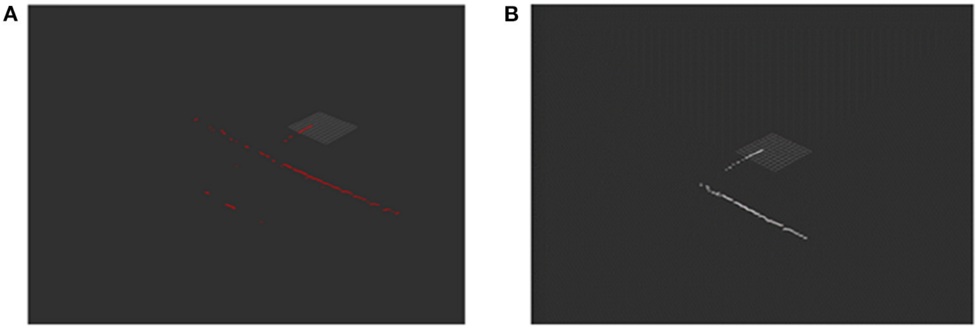


FIGURE 5 | Panel **(A)** shows the point cloud data of the sonar raw data after threshold segmentation and data conversion. Panel **(B)** is the result of filtering based on distance-constrained.

Algorithm 2: RBPF with multi-beam forward looking sonar.

```
Data: $[X_{t-1}, SP]$ The AUV's initial pose and processed sonar data.
Result: $[X_t]$ The pose of the most weighted particles.
begin
    $PS_{t-1}$ = Init();
    for $SP = ProcessSonarData()$ do
        $X_k = DeadReckoning()$;
        $z_t = SP$;
        $u_{t-1} = X_k - X_{k-1}$;
        for $PS_{t-1}^{(i)} \in PS_{t-1}$ do
            $< x_{t-1}^{(i)}, \omega_{t-1}^{(i)}, m_{t-1}^{(i)} >= PS_{t-1}^{(i)}$;
            $\tilde{x}_t^{(i)} = Sample(x_{t-1}^{(i)}, u_{t-1})$;    //sampling
            $\hat{x}_t^{(i)} = IW(m_{t-1}^{(i)}, z_t, \tilde{x}_{t-1}^{(i)})$;    //scan matching and importance weighting
            if $\hat{x}_t^{(i)} = \phi$ then
                $x_t^{(i)} = \tilde{x}_t^{(i)}$;
                $\omega_t^{(i)} = \omega_{t-1}^{(i)} p(z_t|m_{t-1}^{(i)}, x_t^i)$;
            else
                $[x_t^{(i)}, \omega_t^{(i)}] = Resample(\hat{x}_t^{(i)}, \Upsilon, u_{t-1}, m_{t-1}^{(i)})$; //resampling
            end
            $m_t^{(i)} = MapUpdate(m_{t-1}^{(i)}, z_t, x_t^{(i)})$;    //map updating
            $PS_t = PS_t \cup (x_t^{(i)}, \omega_t^{(i)}, m_t^{(i)})$;
        end
        $N_{eff} = \frac{1}{\Sigma_{i=1}^{N}(\tilde{\omega}^{(i)})^2}$;
        if $N_{eff} < T$ then
            $PS_t = resample(PS_t)$;
        else
            $PS_{t-1} = PS_t$;
        end
    end
end
```

probability can be converted into the product of conditional probabilities, the solution of RBPF SLAM is to decompose the original SLAM problem into separate positioning and mapping parts (Grisetti et al., 2007).

$$p(x_{1:t}, m|z_{1:t}, u_{1:t-1}) = p(m|x_{1:t}, z_{1:t}) \cdot p(x_{1:t}|z_{1:t}, u_{1:t-1}) \tag{4}$$

where $p(x_{1:t}|z_{1:t}, u_{1:t-1})$ is the posterior of potential trajectories $x_{1:t}$ given observations $z_{1:t}$ and odometry measurements $u_{1:t}$ of the AUV, $p(m|x_{1:t}, z_{1:t})$ is the posterior of maps, and $p(x_{1:t}, m|z_{1:t}, u_{1:t-1})$ is the posterior of maps and trajectories. Given the values of $x_{1:t}$ and $z_{1:t}$, $p(m|x_{1:t}, z_{1:t})$ can be calculated analytically, so the key to the problem is to calculate $p(x_{1:t}|z_{1:t}, u_{1:t-1})$.

To estimate the posterior $p(x_{1:t}, m|z_{1:t}, u_{1:t-1})$, a group of particles is first introduced. Each particle is composed of the pose x of the AUV, the grid map m, and the weight ω. The particle filter algorithm incrementally uses dead reckoning values and sonar scan data to update the particle set. This process can be divided into four steps, *sampling*, *scanmatchingandimportanceweighting*, *resampling*, and *mapupdate*. Function $\hat{x}_t^{(i)} = IW(m_{t-1}^{(i)}, z_t, \tilde{x}_{t-1}^{(i)})$ is a scan matching and importance weighting module, and its function is to calculate the pose that best matches the current map $m_{t-1}^{(i)}$ based on the current observation z_t and all samples $\tilde{x}_{t-1}^{(i)}$.

Then each particle is assigned a separate importance weighting based on the importance sampling principle $w_t^{(i)}$.

$$w_t^{(i)} = w_{t-1}^{(i)} \cdot \frac{\eta p(z_t|m_{t-1}^{(i)}, x_t^{(i)}) p(x_t^{(i)}|x_{t-1}^{(i)}, u_{t-1})}{p(x_t|m_{t-1}^{(i)}, x_{t-1}^{(i)}, z_t, u_{t-1})} \tag{5}$$

$$\propto w_{t-1}^{(i)} \frac{p(z_t|m_{t-1}^{(i)}, x_t^{(i)}) p(x_t^{(i)}|x_{t-1}^{(i)}, u_{t-1})}{\frac{p(z_t|m_{t-1}^{(i)}, x_t) p(x_t|x_{t-1}^{(i)}, u_{t-1})}{p(z_t|m_{t-1}^{(i)}, x_{t-1}^{(i)}, u_{t-1})}} \tag{6}$$

$$= w_{t-1}^{(i)} \cdot p(z_t|m_{t-1}^{(i)}, x_{t-1}^{(i)}, u_{t-1}) \tag{7}$$

TABLE 1 | The algorithm parameters.

Parameter	Value
Map-update-intervel	0.3
maxRange	25
maxUrange	24
Number of particles	150
Resample threshold	0.5
Number of iterations	7

$$= w_{t-1}^{(i)} \cdot \int p(z_t|x')p(x'|x_{t-1}^{(i)}, u_{t-1})dx' \quad (8)$$

Here, $\eta = 1/p(z_t|z_{1:t-1}, u_{1:t-1})$ is the normalization factor produced by Bayes' rule where all particles are equal.

In function $Resample(\hat{x}_t^{(i)}, \Upsilon, u_{t-1}, m_{t-1}^{(i)})$, the Gaussian approximation of the proposed distribution is calculated, and new particles are sampled for the next iteration based on the calculated result. Υ is the interval threshold for resampling in the vicinity area of $\hat{x}_t^{(i)}$.

Finally, the map $m_t^{(i)}$ is updated based on the estimated pose $x_t^{(i)}$ and the observed value z_t. Then, select the map and pose of the particle with the largest weight among all the particles as the final constructed map and estimated AUV pose.

4. EXPERIMENTAL RESULT

Both simulation and practical experiments are conducted to verify the effectiveness of the proposed SLAM algorithm. **Table 1** shows the main parameters of our algorithm for experiments.

4.1. Simulation Experiments

In the simulation experiment, we used UUV-Simulator to create a Rexrov2 model, which is a full-propeller-driven ROV, and it is equipped with four cameras, four lights, and a wide range of sensors, including sonar, DVL, IMU, etc.

Figure 6 shows the Rexrov2 model and simulation environment, an underwater maze. First, build a 3-D model of the environment and load it into the 3-D simulator Gazebo. To facilitate interaction with Gazebo, we used Robot Operating System (ROS) in our simulation. The drivers of IMU, DVL, and MFLS are loaded as Gazebo plug-ins and used to publish ROS-compatible data, which are subscribed by the proposed SLAM algorithm. Our ROV is controlled to navigate the maze for one round during the simulation. The algorithm's outputs include an occupancy grid map and an estimated ROV trajectory.

Figure 7 shows the occupancy grid map generated by the proposed SLAM algorithm. Based on this map, AUV can make path planning to avoid obstacles and reach the designated position.

The comparison between the ROV position calculated by the proposed algorithm and the ground truth is shown in **Figure 8**. Their trajectory error is very small, and the simulation experiment proves the effectiveness of our proposed method.

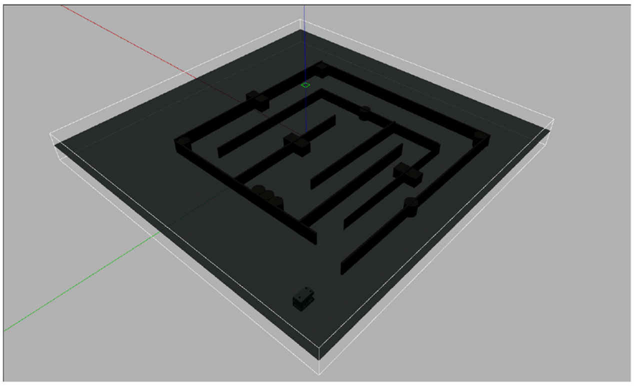


FIGURE 6 | Experimental environment and Rexrov2 model.

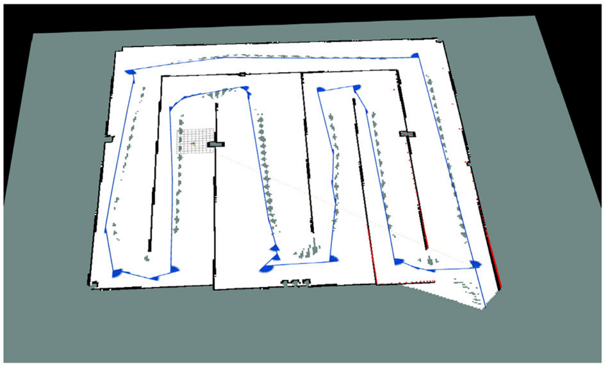


FIGURE 7 | Two-dimensional grid map.

4.2. Experimental Pool Experiment

In the practical experimental, the open-source underwater robot platform BlueROV2 was used to complete the SLAM experiment of the underwater vehicle. In this paper, we only focus on the positioning and surveying indicators of the underwater vehicle, without considering the control. Therefore, the effect is the same regardless of whether the AUV or ROV is used as the experimental carrier to carry out the verification experiment. To make BlueROV2 meet our experimental requirements, we installed a multi-beam forward looking sonar and a DVL based on the original BlueROV2. Also, we installed a pair of power cat modules to achieve underwater BlueROV2 and the shore PC Long-distance data transmission. The modified BlueROV2 (as shown in **Figure 9**) can well meet the SLAM experimental requirements.

We first tested in the multi-purpose pool indoor. The size of the pool is 70 x 44 m. In the experiment, we controlled the BlueROV2 to face the wall for scanning and mapping.

The experimental pool environment is shown in **Figure 10A**, and the positioning and mapping effect of the proposed algorithm is shown in **Figure 10B**.

In **Figure 10B**, the red part represents the trajectory of the odometer, the white part represents the detectable travelable area,

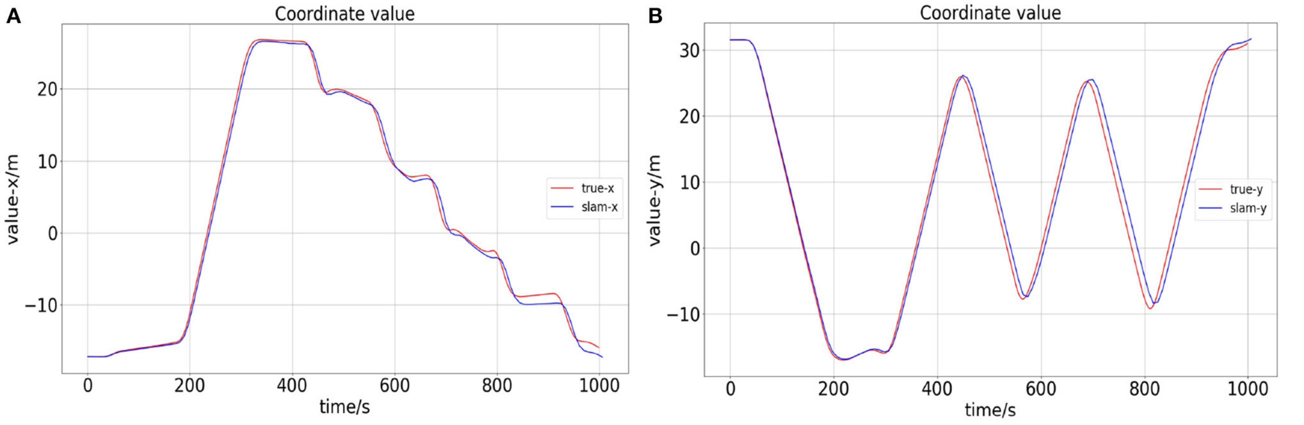


FIGURE 8 | Panel **(A)** is the x-axis error between the calculated value of slam and the true value. Panel **(B)** is the y-axis error.

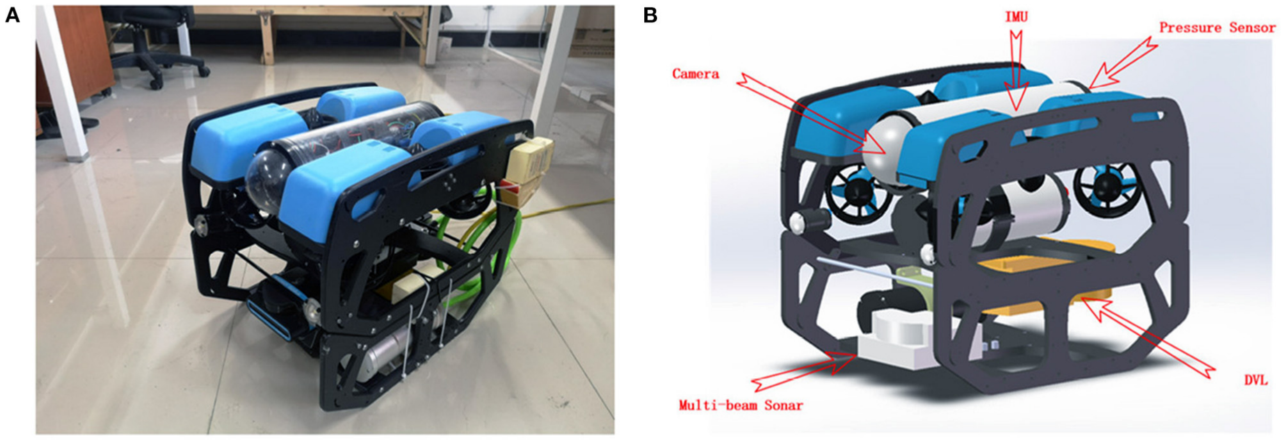


FIGURE 9 | Panel **(A)** is modified BlueROV2, and Panel **(B)** is the sensor layout.

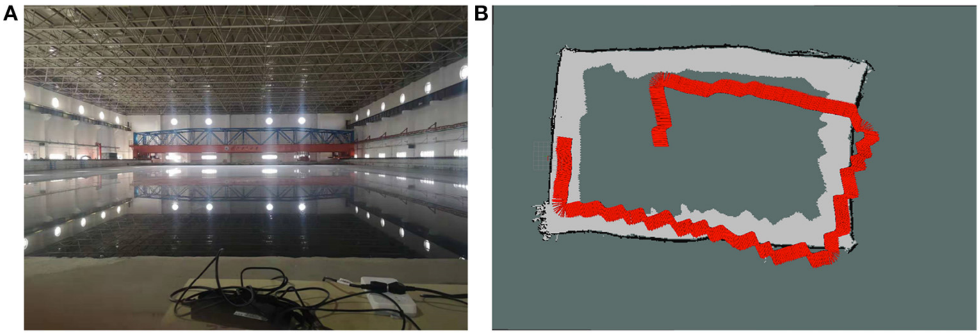


FIGURE 10 | Panel **(A)** is the experimental pool environment. Panel **(B)** is the positioning and mapping effect of SLAM.

the black part represents the obstacle, and the gray part represents the undetected area.

In this experiment, we used threshold segmentation and distance-constrained filtering to process the multi-beam sonar data and then used it to build the map. It can be seen from the mapping results that good results have been achieved. At the same time, it can be seen that the odometer error is continuously accumulating. When the scan is completed, the odometer has not completed closing, but the map established by the SLAM algorithm based on the multi-beam sonar has been closed, the odometer's position deviation is amended. Furthermore, the superiority of the SLAM algorithm based on multi-beam sonar used in this paper is proved.

4.3. Wild Lake Experiment

The field lake experiment was carried out in Liquan Lake, Xi'an, Shaanxi. During the experiment, we take a dinghy to approach the target environment, launch BlueROV2, and control it to scan the target environment for positioning and mapping experiments. Also, there is a GPS positioning antenna on the dinghy. When the BlueROV2 scanning environment, the boat closely followed BlueROV2 to obtain GPS coordinates near it, providing actual data for the quantitative analysis of the positioning surveying experiment.

Figure 11 is a satellite image of the two experimental locations we choose.

Figures 12, **13** are the positioning and mapping results of the two experimental scenes with our proposed algorithm.

It can be seen from the experimental results (**Figures 12A**, **13A**) that the mapping effect is mostly consistent with satellite

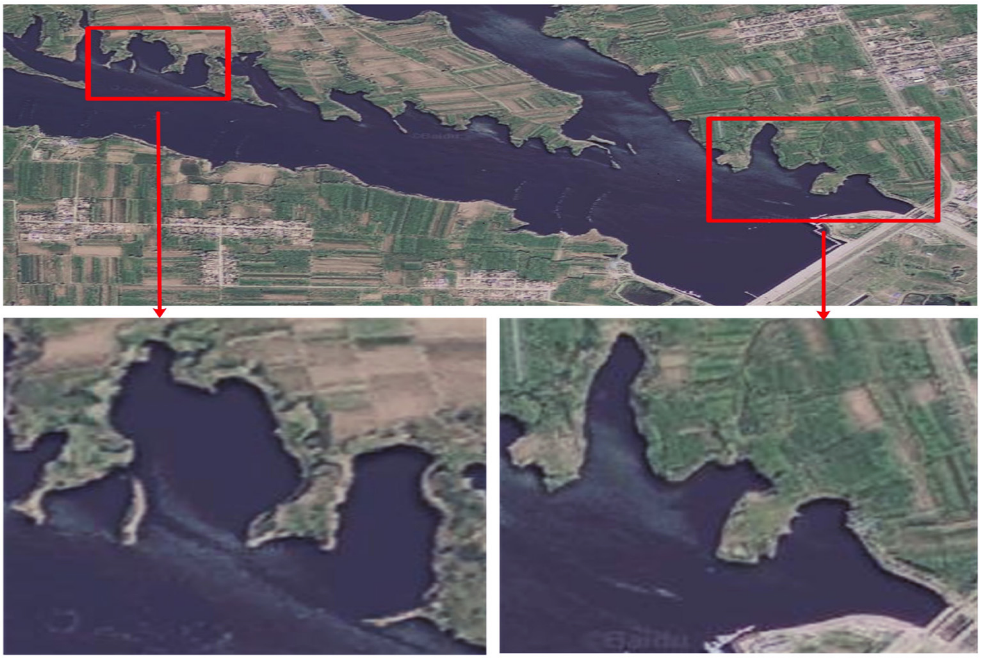


FIGURE 11 | Satellite picture of the place where the experiment was conducted.

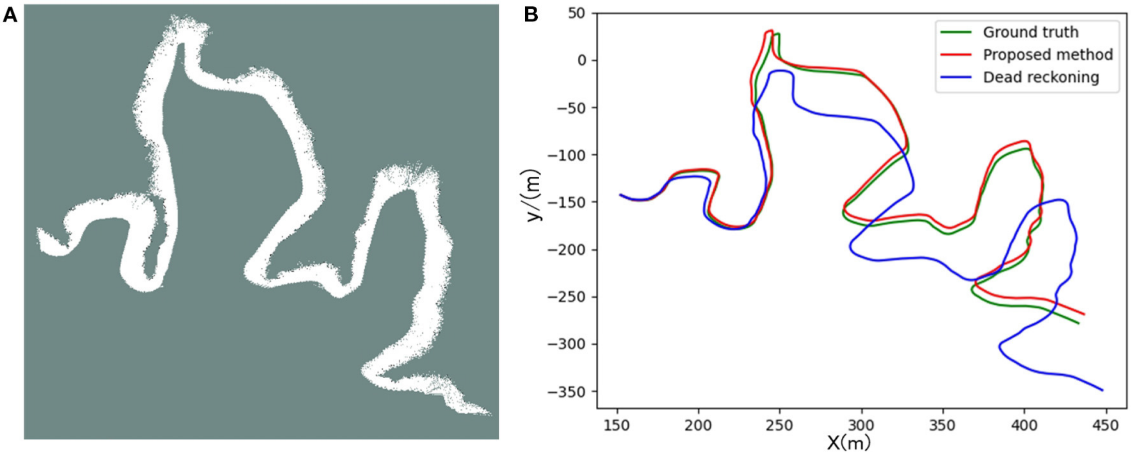


FIGURE 12 | Panel **(A)** is the result of the two-dimensional grid map created in scene 1. Panel **(B)** is the trajectory calculated by the proposed algorithm and dead-reckoning, respectively.

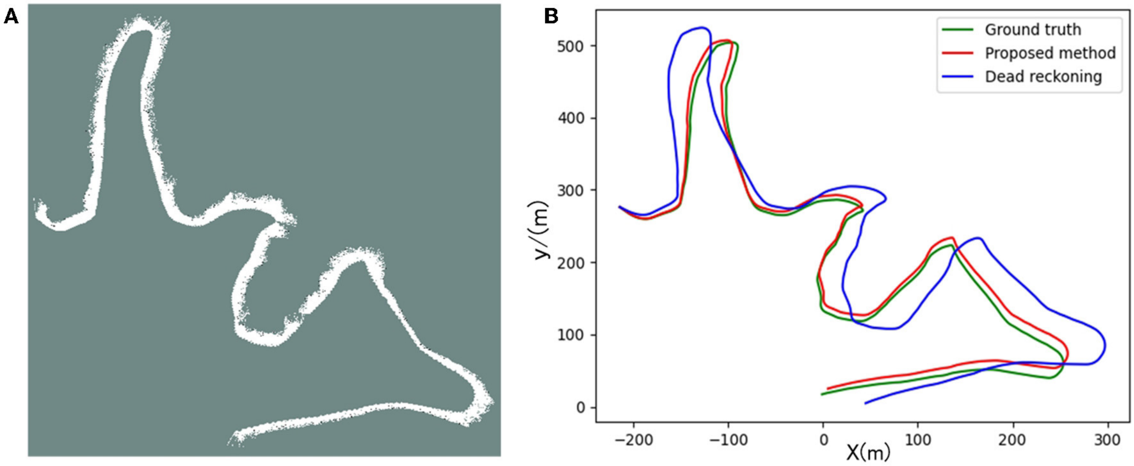


FIGURE 13 | Panel **(A)** is the result of the two-dimensional grid map created in scene 2. Panel **(B)** is the trajectory calculated by the proposed algorithm and dead-reckoning, respectively.

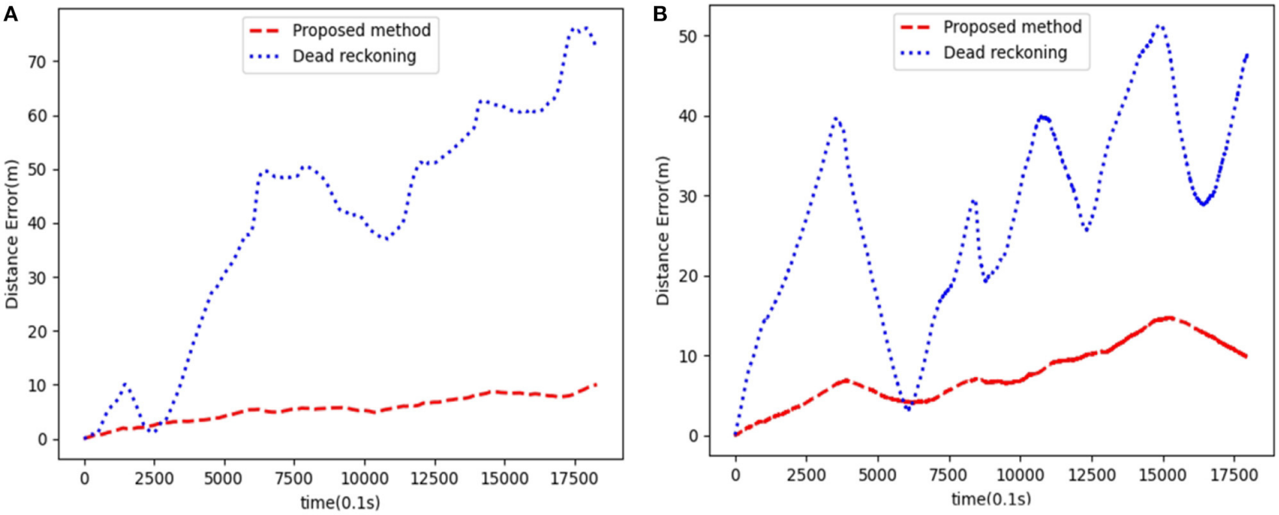


FIGURE 14 | Panel **(A)** is the relative localization errors among the dead-reckoning trajectories, the proposed SLAM algorithm and the ground truth using data from scene1. Panel **(B)** is the scene2.

images. There are several reasons for the deviation between the satellite image and the map created by SLAM:

- Due to changes in water level, satellite maps may deviate from actual maps.
- When BlueRov2 dives into the water, it scans the underwater extension of the lake bank, including objects such as branches and rocks. Unfortunately, the satellite image of this part of the underwater terrain is invisible, so the constructed map will be different from the satellite image.
- BlueRov2 cannot reach some waters, so the factual environmental information has not been thoroughly scanned, which will cause differences between the constructed map and the actual environment.

To quantitatively analyze the map's location accuracy, we compare and analyze the ground truth (GPS measurement values), the estimated value of the proposed algorithm, and the dead reckoning value.

Since the GPS measured value is the latitude and longitude information, we used the geodesy package provided by ROS in the experiment and completed the conversion of latitude and longitude coordinates to two-dimensional coordinates. In this way, we unified the above three coordinate values under the same reference system to evaluate the SLAM algorithm's positioning accuracy.

The trajectory result graph is shown in **Figures 12B**, **13B**, **14** shows the relative localization errors. From **Figure 14** we can find that with the accumulation of time, the deviation of the odometer will gradually increase. Using SLAM for simultaneous positioning and mapping will correct the odometer’s deviation, correct the pose, and build a better map.

The maps based on the multi-beam sonar SLAM algorithm proposed in this paper can better represent the environment’s characteristics. And positioning accuracy is also better than pure dead reckoning.

5. CONCLUSIONS

This paper proposed a SLAM algorithm using MFLS. Two problems are solved: Aiming at the slow processing speed caused by a large amount of MFLS image data, and a method is proposed to convert the collected sonar image into sparse point cloud format data through threshold segmentation and distance-constrained filtering; Based on the proposed method, the DVL, IMU, and MFLS data are fused, and then the RBPF-based SLAM method is used to suppress the accumulation of errors of the inertial unit and generate an accurate occupancy grid map. Finally, we used BlueROV2 as the experimental carrier, conducted field tests in the experimental pool and Liquan Lake, Xi’an, Shaanxi, and achieved good positioning and mapping results.

6. FUTURE WORK

This article mainly uses MFLS image data to be converted into sparse point cloud data format for SLAM experiments, which involves the problem of sonar filtering and data matching after filtering. There are still some problems currently, such as the inability to fully extract environmental features due to clutter interference and the map drifting due to data matching failure. For these problems, future work will continue to study better filtering methods and data matching methods to improve the accuracy of positioning and mapping.

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/s.

AUTHOR CONTRIBUTIONS

CC and FZ presented the idea in this article and wrote the paper together. FZ and WL provided technical support and experimental equipment. DY and CW helped complete the experimental verification. 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 would like to thank Songxiang Wang, Xijun Zhou, and Liyuan Chen et al. for their help during the experiment.

REFERENCES

Aulinas, J., Lladó, X., Salvi, J., and Petillot, Y. R. (2010). “Feature based slam using side-scan salient objects,” in *OCEANS 2010 MTS/IEEE SEATTLE* (Seattle, WA), 1–8.

Barrera, C., Real-Arce, D. A., Waldmann, C., Caldeira, R., Rueda, M. J., Hernández, J., et al. (2018). “Ocean-gliders contribution to improve monitoring capacity in the east-central north atlantic,” in *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)* (Porto), 1–5.

Chen, L., Yang, A., Hu, H., and Naeem, W. (2020). Rbpf-msis: toward rao-blackwellized particle filter slam for autonomous underwater vehicle with slow mechanical scanning imaging sonar. *IEEE Syst. J.* 14, 3301–3312. doi: 10.1109/JSYST.2019.2938599

Grisetti, G., Stachniss, C., and Burgard, W. (2007). Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Trans. Robot.* 23, 34–46. doi: 10.1109/TRO.2006.889486

Hagen, P. E., Storkersen, N., Marthinsen, B. E., Sten, G., and Vestgard, K. (2005). “Military operations with hugin auvs: lessons learned and the way ahead,” in *Europe Oceans 2005*, vol. 2 (Brest), 810–813.

Hao, R., Zhuo, C., Zhao, P., Nie, Q., and Li, B. (2020). “Research on optimized m-estimate arithmetic in ins/usbl integrated navigation system,” in *2020 Chinese Automation Congress (CAC)* (Shanghai), 939–944.

Hong, S., Kim, J., Pyo, J., and Yu, S.-C. (2016). A robust loop-closure method for visual slam in unstructured seafloor environments. *Auton. Robots* 40, 1095–1109. doi: 10.1007/s10514-015-9512-6

Hurtós, N., Nagappa, S., Palomeras, N., and Salvi, J. (2014). “Real-time mosaicing with two-dimensional forward-looking sonar,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hong Kong), 601–606.

Jung, J., Lee, Y., Kim, D., Lee, D., Myung, H., and Choi, H. (2017). “Auv slam using forward/downward looking cameras and artificial landmarks,” in *2017 IEEE Underwater Technology (UT)* (Busan), 1–3.

Kim, A., and Eustice, R. M. (2015). Active visual slam for robotic area coverage: theory and experiment. *The Int. J. Robot. Res.* 34, 457–475. doi: 10.1177/0278364914547893

Matos, A., Cruz, N., Martins, A., and Lobo Pereira, F. (1999). “Development and implementation of a low-cost lbl navigation system for an auv,” in *Oceans ’99. MTS/IEEE. Riding the Crest into the 21st Century. Conference and Exhibition. Conference Proceedings (IEEE Cat. No.99CH37008)*, vol. 2 (Seattle, WA), 774–779.

Neves, G., Ruiz, M., Fontinele, J., and Oliveira, L. (2020). Rotated object detection with forward-looking sonar in underwater applications. *Exp. Syst. Appl.* 140:112870. doi: 10.1016/J.ESWA.2019.112870

Ohta, Y., Yoshida, H., Ishibashi, S., Sugesawa, M., Fan, F. H., and Tanaka, K. (2016). “Seabed resource exploration performed by auv ‘yumeiruka,”’ in *OCEANS 2016 MTS/IEEE Monterey* (Monterey, CA), 1–4.

Pyo, J., Cho, H., and Yu, S. (2017). Beam slice-based recognition method for acoustic landmark with multi-beam forward looking sonar. *IEEE Sens. J.* 17, 7074–7085. doi: 10.1109/JSEN.2017.2755547

Siantidis, K. (2016). “Side scan sonar based onboard slam system for autonomous underwater vehicles,” in *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)* (Tokyo), 195–200.

Suresh, S., Westman, E., and Kaess, M. (2019). Through-water stereo slam with refraction correction for auv localization. *IEEE Robot. Autom. Lett.* 4, 692–699. doi: 10.1109/LRA.2019.2891486

Venkatesan, S. (2016). "Auv for search rescue at sea - an innovative approach," in *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)* (Tokyo), 1–9.

Vickery, K. (1998). "Acoustic positioning systems. a practical overview of current systems," in *Proceedings of the 1998 Workshop on Autonomous Underwater Vehicles (Cat. No.98CH36290)* (Cambridge, MA), 5–17.

Wang, J., Bai, S., and Englot, B. (2017). "Underwater localization and 3d mapping of submerged structures with a single-beam scanning sonar," in *2017 IEEE International Conference on Robotics and Automation (ICRA)* (Singapore), 4898–4905.

Wang, J., Shan, T., and Englot, B. (2019). "Underwater terrain reconstruction from forward-looking sonar imagery," in *2019 International Conference on Robotics and Automation (ICRA)* (Montreal, QC), 3471–3477.

Wang, W., and Cheng, B. (2020). "Augmented ekf based slam system with a side scan sonar," in *2020 12th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, vol. 1 (Hangzhou), 71–74.

Williams, S. B., Pizarro, O. R., Jakuba, M. V., Johnson, C. R., Barrett, N. S., Babcock, R. C., et al. (2012). Monitoring of benthic reference sites: using an autonomous underwater vehicle. *IEEE Robot. Autom. Mag.* 19, 73–84. doi: 10.1109/MRA.2011.2181772

Wright, R. G., and Baldauf, M. (2016). Hydrographic survey in remote regions: using vessels of opportunity equipped with 3-dimensional forward-looking sonar. *Mar. Geodesy.* 39, 439–457. doi: 10.1080/01490419.2016.1245226

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.

frontiers
in Neurorobotics

ORIGINAL RESEARCH
published: 11 February 2022
doi: 10.3389/fnbot.2021.798428

MPPTM: A Bio-Inspired Approach for Online Path Planning and High-Accuracy Tracking of UAVs

Xin Yi[1], *Anmin Zhu*[1]* and *S. X. Yang*[2]

[1] *Research Institute of Intelligence Technology and System Integration, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China,* [2] *Advanced Robotics and Intelligent Systems (ARIS) Laboratory, School of Engineering, University of Guelph, Guelph, ON, Canada*

The path planning and tracking problem of the multi-robot system (MRS) has always been a research hotspot and applied in various fields. In this article, a novel multi-robot path planning and tracking model (MPPTM) is proposed, which can carry out online path planning and tracking problem for multiple mobile robots. It considers many issues during this process, such as collision avoidance, and robot failure. The proposed approach consists of three parts: a neural dynamic path planner, a hyperbolic tangent path optimizer, and an error-driven path tracker. Assisted by Ultra-wideband positioning system, the proposed MPPTM is a low-cost solution for online path planning and high-accurate tracking of MRS in practical environments. In the proposed MPPTM, the proposed path planner has good time performance, and the proposed path optimizer improves tracking accuracy. The effectiveness, feasibility, and better performance of the proposed model are demonstrated by real experiments and comparative simulations.

Keywords: multi-robot system, path planning, neural dynamics, path tracking, neural network

OPEN ACCESS

Edited by:
Huiyu Zhou,
University of Leicester,
United Kingdom

Reviewed by:
Guodong Zeng,
University of Bern, Switzerland
Shuyang Chen,
ABB, United States

****Correspondence:***
Anmin Zhu
azhu@szu.edu.cn

Received: *20 October 2021*
Accepted: *30 December 2021*
Published: *11 February 2022*

Citation:
Yi X, Zhu A and Yang SX (2022)
MPPTM: A Bio-Inspired Approach for
Online Path Planning and
High-Accuracy Tracking of UAVs.
Front. Neurorobot. 15:798428.
doi: 10.3389/fnbot.2021.798428

1. INTRODUCTION

As the development of disciplines and technologies, robotics always involves numerous disciplines. It covers many aspects from control, mechanics, electronics to communication, computer science, materials, and so forth. Robotics has also developed from a simple single robot system (SRS) to a complex multi-robot system (MRS). For dealing with complex problems, MRS has more advantages than SRS. Large numbers of researches state that the cooperation of MRS has been applied to more practical fields, such as services (Morita et al., 2018; Krizmancic et al., 2020), therapy (Ali et al., 2019; Mehmood et al., 2019), rescue (Queralta et al., 2020), training (Xu and Tang, 2021), and so on.

Path planning and tracking problem of MRS has always been a research hotspot and applied in various fields, including delivery (Chen et al., 2021), monitoring (Silic and Mohseni, 2019; Koutras et al., 2020), task assignment (Chen and Zhu, 2019; Wang et al., 2020b), target tracking (Zhou et al., 2018), and so on. In order to solve this problem, many kinds of research can be divided into three aspects: task-level, control-level, and task-control-level (Zeng et al., 2015; Rubí et al., 2019). The task-level research focuses on finding the optimal solution to the problem without considering the hardware conditions, which is top-down. It includes path planning of MRS and task assignment of MRS. The control-level research needs to consider the hardware of MRS and the realization of the solution, which is down-top. Tracking the target or path with high accuracy is one of these kinds of research. The task-control-level research combines task-level and control-level, and the path planning and tracking problem is one of this kind of research.

In the task-level research of MRS, path planning and task assignment are the two mainstreams. Compared with task assignments, path planning is more focused on the time-space continuity and process. For example, these studies (Yi et al., 2017; Dai et al., 2019; Ali et al., 2020; Dong et al., 2020; Han and Yu, 2020; Wang et al., 2020a) only consider task-level without considering control-level details. Combining with deep learning, (Wang et al., 2020a) proposed a neural RRT* for path planning of MRS, but it needs a lot of processed data for training before path planning (Wang et al., 2020a). Han et al. used database heuristics for fast near-optimal path planning of MRS, it can carry out efficient path planning, but its applicable scene is only based on grid environment (Han and Yu, 2020). Yi et al. proposed a bio-inspired approach to plan the path of robots in 3-D environments. Also, it can make real-time path planning but can not avoid obstacles (Yi et al., 2017). Dong et al. proposed a path planning method of UAVs in the 3-D environment for the inspection of transmission lines. But it is only made available for a single target, not for multiple targets (Dong et al., 2020).

Path tracking for MRS is solved by control-level algorithms. These algorithms tend to reduce error during the tracking process while considering the difference of robot hardware in MRS. Ma proposed cooperative tracking of MRS with circular formation, but it can not track multi-target (Ma, 2020). Yu et al. proposed a formation tracking method based on an adaptive neural network, but it just makes MRS formate to track a single target (Yu et al., 2018). Zhou et al. presented a resilient tracking method for MRS. It is suitable for patrol and monitoring in the area but can not track the immovable target (Zhou et al., 2018).

Both task-level and control-level studies are very limited in practical application. Therefore, some studies focusing on both task-level and control-level have occurred.

Park et al. proposed a distributed approach combing alternating direction method of multipliers (ADMM) to non-myopic path planning for multi-target tracking, but it can not avoid obstacles in the environment (Park et al., 2019). Yordanova et al. proposed a path planning and tracking method for the area coverage of autonomous underwater vehicles, but in essence, the method is still only for the 2-D environment without obstacles (Yordanova and Gips, 2020). Penin et al. proposed a vision-based path planning and target tracking method for UAVs (Penin et al., 2018). It can deal with collision avoidance and obstacle avoidance, but its accuracy of vision-based positioning is still questionable for indoor environments.

Compared with these studies (Penin et al., 2018; Park et al., 2019; Yordanova and Gips, 2020; Yu et al., 2020), there are few studies for online path planning and tracking of MRS. During online path planning and tracking of MRS, the proposed model needs to plan the trajectories easy to track in real-time, which deals with dynamic environments and accidents, such as robot fault, moving obstacles, and so on. Online path planning and tracking need to solve the following three problems. (1) How can path planning meet the requirements of real-time; (2) How can the planned paths be transformed into trajectories easy to track; and (3) How to efficiently organize related processes?

In this study, a novel model named MPPTM (multi-robot path planning and tracking model) is proposed for online path planning and high-accuracy tracking of MRS. It does not depend on the sensors of the individual robot in MRS by using an Ultra-wideband (UWB) positioning system. Therefore, the proposed approach is a low-cost solution for warehouse or factory environments. The proposed model has the following innovations.

1. The proposed model uses superscalar pipelining mode to organize these processes more efficiently. Therefore, the process of path tracking does not need to wait until the end of path planning.
2. Compared with traditional path planners, the proposed neural dynamic path planner has better time performance.
3. In our proposed model, the hyperbolic tangent path optimizer bridges the planned paths and the trajectories easy to track, and it reduces the tracking error of MRS.

The remainder of the article is organized as follows. In section 2, the components and framework of the proposed model are introduced in detail. The experiments for MRS in a 3-D environment are present in section 3. Some further discussions about the comparative studies are given in section 4. Finally, the conclusion and future study are presented in section 5.

2. PROPOSED APPROACH

The proposed MPPTM is described in detail in this section. It is applicable for not only 2-D but also 3-D environments. It combines the task-level and control-level, which can deal with the path planning of MRS, and cope with the path tracking of MRS.

The proposed MPPTM mainly integrates three parts, including the path planner, the path tracker, and the path optimizer. The path planner is responsible for dealing with the online path planning of MRS at the task-level. During the path planning of MRS, obstacles avoidance, collision avoidance, and other robot accidents are considered. The path tracker is used to cope with the path tracking of MRS. During the path tracking of MRS, reducing tracking error and compatibility with different hardware are considered. The path optimizer is used to bridge the gap between the path planner and the path tracker, which makes the planned path to be easier tracked. It can process the online path planning and path tracking for MRS in complex environments.

The framework of MPPTM with environments is shown in **Figure 1**. The process of path planning and tracking, the cooperation mechanism between the path planner and the path tracker are given in the framework of MPPTM. In **Figure 1**, k and t are the iterations of the planner and the iteration of the tracker, respectively. For the path planner, $I(k)$ represents the environmental information, and $P_c(k)$ represents the current posture of the MRS. Both are used to generate the desired posture $P_d(k)$ through $Q(k)$. Through the path optimizer, the desired posture $P_d(k)$ is transferred into the desired trajectory $T_d(t)$ for

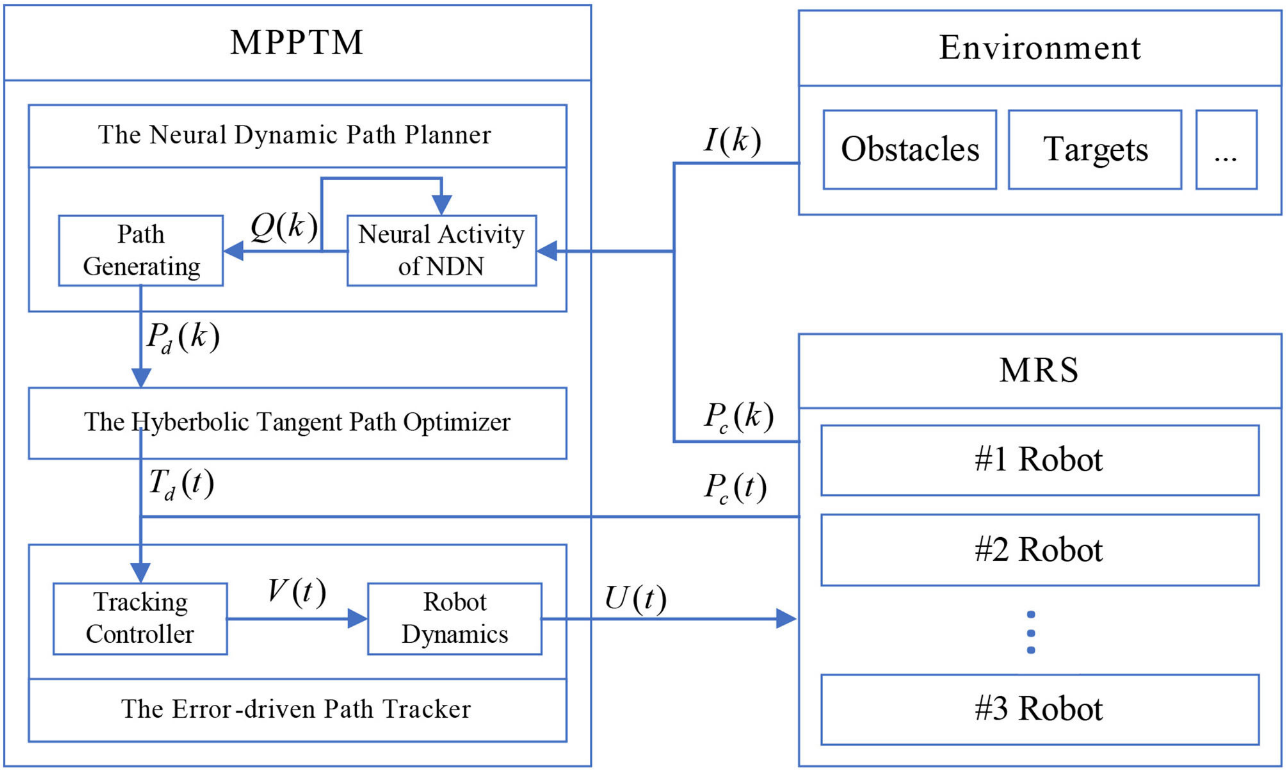


FIGURE 1 | Overall schematic diagram of multi-robot path planning and tracking model (MPPTM) with environments.

the path tracker to track. In the path tracker, the desired posture $T_d(k)$ and current posture $P_c(t)$ of MRS are used to generate the desired velocity $V(t)$ by path tracker for MRS. Additionally, then it is converted into the desired motor speed $U(t)$ by robot dynamics. In the MRS, each robot tries to achieve the desired motor speed $U(t)$ to move.

Usually, there are two different modes to deal with the path planning and tracking of MRS, which are superscalar pipelining mode and traditional mode.

In traditional mode, path tracking must wait until all path planning and optimization are completed. This mechanism handles these processes serially, which can save computing resources. However, it is only suitable for path planning and tracking of MRS in static environments.

Due to the time performance of the proposed path planner, the proposed MPPTM applies superscalar pipelining mode to deal with these processes, such as path planning, path optimizing, and path tracking, as shown in **Figure 2**.

In superscalar pipelining mode, part of the planned path is used to be optimized by the path optimizer, and tracked by the path tracker. Path tracking does not have to wait for all path planning and optimization to be completed. This mechanism parallels these processes to some extent, which can be suitable for online path planning and tracking of MRS in dynamic environments.

The proposed MPPTM is introduced in detail by three following parts: (A) The neural dynamic path planner; (B) The error-driven path tracker; and (C) The hyperbolic tangent path optimizer.

2.1. The Improved Neural Dynamic Path Planner

As an important part of MPPTM, the neural dynamic path planner is used to plan paths for MRS. It is based on neural dynamics and has the advantages and characteristics of the biological neural system. The neural dynamic path planner for 3-D environments is introduced in the section, and the proposed path planner for 2-D environments is similar to this. The neural dynamic path planner includes three following parts.

2.1.1. The Neural Dynamic Network (NDN)

In the neural dynamic path planner, the NDN is used to real-time describe the environment where MRS is located. In NDN, the distance between any two adjacent neurons is equal, its distance is 1, and any two adjacent neurons are also connected to each other. The structure of NDN describing the 3-D environment is as shown in **Figure 3**.

The activity of neuron q_{ijk} is in the ith row, the jth column, and the kth page of the NDN, which describes the environment that it maps. The activities of NDN Q

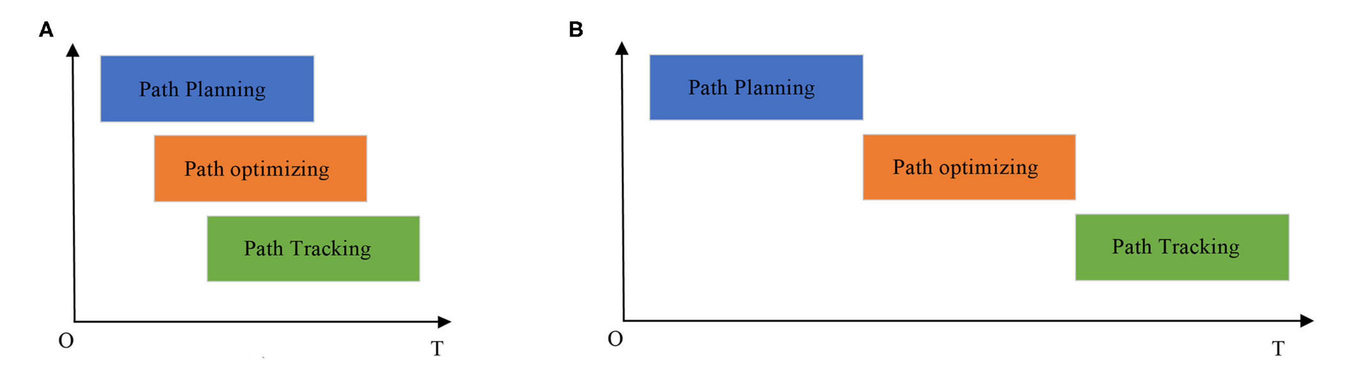


FIGURE 2 | Two different operation modes in MPPTM. **(A)** The superscalar pipelining mode. **(B)** The traditional mode.

describing 3-D environments is a 3-D matrix defined by Equation (1).

$$Q = \begin{bmatrix} \begin{bmatrix} q_{111} & \cdots & q_{1n1} \\ \vdots & \ddots & \vdots \\ q_{m11} & \cdots & q_{mn1} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} q_{11h} & \cdots & q_{1nh} \\ \vdots & \ddots & \vdots \\ q_{m1h} & \cdots & q_{mnh} \end{bmatrix} \end{bmatrix}_{m\times n\times h} \tag{1}$$

In **Figure 3**, r is the radius of the range that neurons can affect, x is horizontal offset, y is vertical offset, z is longitudinal offset, and x, y, z are integers. The radius of the search sphere r directly affects the computation performance and the accuracy of path planning. If radius r is set too large, the planned path will pass through the obstacles in the environment. In NDN, the activity of each neuron q represents the environment where it maps. Therefore, the activities of NDN Q describe the whole environment.

2.1.2. The Improved Neural Activity Algorithm

This algorithm states the activity of neural signals in NDN. Through multiple iterations of neural activity, the activities of NDN Q tend to be stable, and all kinds of signals have been fully spread in NDN. The neural activity of NDN is the core part of the proposed path planner, and it is also the most time-consuming process. Therefore, we proposed an improved neural activity algorithm of NDN, and it is defined by Equation (2).

$$\begin{aligned} \tfrac{dQ}{dt} = & -KQ + (D - Q)\left([I]^{+} + [F(x,y,z)]^{+}\right) \\ & -(J + Q)\left([I]^{-} + [F(x,y,z)]^{-}\right) \end{aligned} \tag{2}$$

Three parameters, K, D, J, are decay rate, upper bound, and lower bound, respectively, in the dynamic equation of neural activity. Meanwhile, two operators, $[a]^{+}$ and $[b]^{-}$, obtain, respectively, $max\{a, 0\}$ and $max\{-a, 0\}$. The function $F(x, y, z)$ is the weighted sum of the shifting matrix Q with these offsets x, y, z, and it is defined by Equation (3).

$$F(x,y,z) = \sum_{xyz} w_{xyz} shift(Q, x, y, z) \quad \sqrt{x^2 + y^2 + z^2} \le r \tag{3}$$

where $shift(Q, x, y, z)$ shifts the elements of matrix Q with the x rows, the y columns, and the z pages, but it satisfies the condition $\sqrt{x^2 + y^2 + z^2} \le r$. This weight w_{xyx} is defined by Equation (4).

$$w_{xyz} = \frac{u}{\sqrt{x^2 + y^2 + z^2}} \quad \sqrt{x^2 + y^2 + z^2} \le r \tag{4}$$

where w_{xyz} is the connection weight when the horizontal offset is x, the vertical offset is y, and longitudinal offset is z in NDN, u is the positive parameter and represents the intensity of the connection.

In Equation (2), environmental information I is a 3-D matrix with the same size as Q, its element is defined as Equation (5). Ex and In are positive parameters, which represent the intensity of excitatory nerve signal and inhibitory nerve signal, respectively, in NDN.

$$i_{ijk} = \begin{cases} Ex & \text{The neuron maps target} \\ -In & \text{The neuron maps robot} \\ 0 & \text{Others} \end{cases} \tag{5}$$

2.1.3. The Path Generation for 3-D Environments

After the multiple iterations of neural activity, the activities of NDN Q are used to generate the next position during the path planning of MRS. The next position of the i_{th} robot $P^i(k+1)$ is defined as Equation (6).

$$P^i(k+1) \Leftarrow q_{abc} = max\left\{q_{efg} \middle|\ 0 < \left\|q_{efg} - q_{ijk}\right\| \ < r\right\} \tag{6}$$

Assume that the current position of the i_{th} robot is mapped by the neuron q_{ijk}, q_{efg} is the set of neurons in the affected range of the neuron q_{ijk}. In the set q_{efg}, the maximum activity of the neuron q_{abc} is selected as the next position $P^i(k+1)$ of the i_{th} robot during path planning.

2.2. The Error-Driven Path Tracker

The proposed path tracker uses error driven method to track the path planned by the proposed path planner. The error-driven path tracker is capable of being compatible with a variety of controllers and the hardware of MRS. In order to better test the performance of the proposed MPPTM in 3-D environments, a

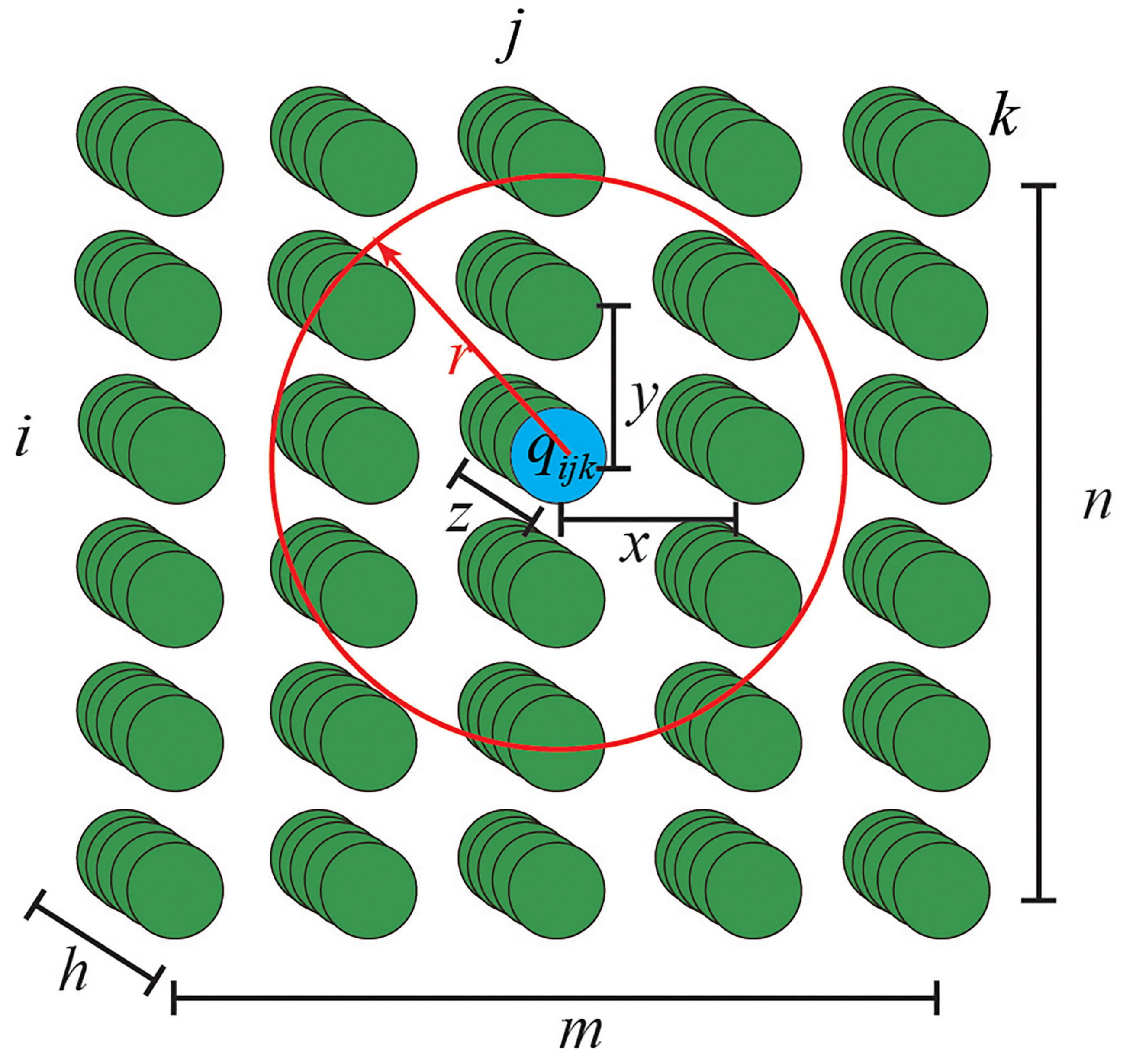


FIGURE 3 | The neural dynamic network (NDN) describing 3-D environments.

path tracker with PID controller for quadrotor is given briefly in the article, which is used in the following experiments.

Quadrotor, a helicopter with four rotors, is a small unmanned aerial vehicle (UAV). Additionally, it is used as an individual MRS in 3-D environments. Two control loops with PID control are used in the path tracker of MRS for path tracking in 3-D environments. The system architecture of the proposed path tracker for 3-D tracking with PID control is shown in **Figure 4**.

2.3. The Hyperbolic Tangent Path Optimizer

The path optimizer bridges the gap between the path planner with the path tracker, it translates the task-level paths planned by the path planner into the control-level paths for the path tracker. During this process, the task-level paths $P^i(k)$ with low frequency should be transformed into the control-level trajectory $T^i(t)$ with high frequency, which can make the planned paths easy to track for MRS. The planned paths are given by the path planner are defined as Equation (7), where $Paths^i(\gamma)$ is the continuous planned path of the i_{th} robot in MRS, k is the iteration of the path planner, and R is the number of robots in MRS.

$$\begin{aligned} P^i(k) \in \left\{Paths^i(\gamma) |\, i = 1, 2 \ldots, R \;\; \gamma \geq 0\right\} \\ k = 1, 2 \ldots, n \;\; i = 1, 2 \ldots, R \end{aligned} \tag{7}$$

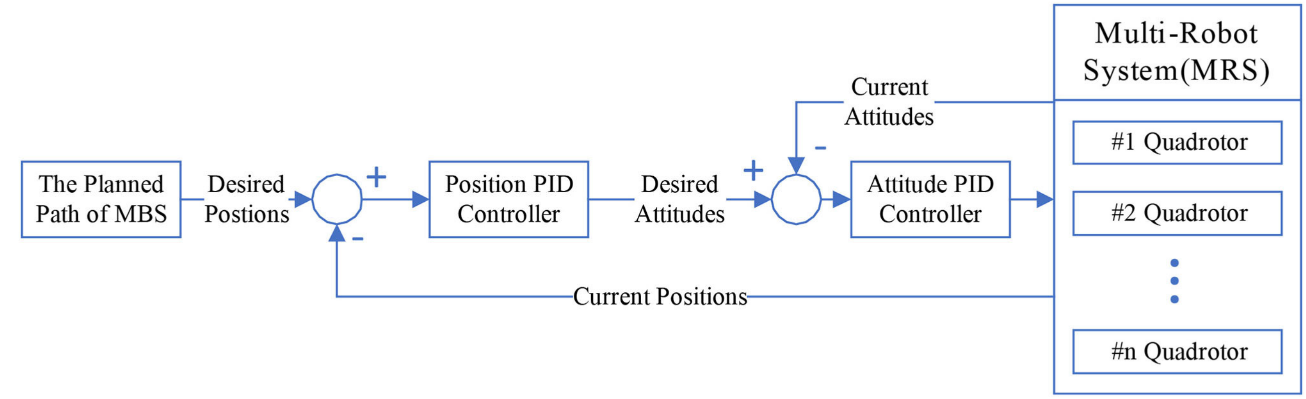


FIGURE 4 | The proposed path tracker for quadrotors in 3-D environments.

The control-level trajectory $T^i_{os}(t)$ given by the path optimizer with original sampling from $P^i(k)$ can be obtain by Equation (8), where the operator $\lfloor a \rfloor^b$ obtains the largest element in set b and less than or equal to a, and w is the sampling frequency.

$$\begin{cases} a = \lfloor {}^{t}\!/_{w} \rfloor^k \\ T^i_{os}(t) = [P^i(a+1) - P^i(a)] \times \frac{t\%w}{w} \quad i = 1, 2 \ldots R \end{cases} \tag{8}$$

In order to improve the tracking performance of MRS, an improved sampling method is proposed in the article. $P^i(\tau)$ is the set of elements in set $P^i(k)$ whose elements are not differentiable on the paths of MRS, which can describe as Equation (9).

$$P^i(\tau) \in \left\{ P^i(k) | \begin{array}{c} \lim\limits_{\Delta\delta \to 0^+} \dot{P}^i(k + \Delta\delta) \neq \lim\limits_{\Delta\delta \to 0^-} \dot{P}^i(k + \Delta\delta) \\ k \in 1, 2 \ldots, n \end{array} \right\}$$
$$i \in 1, 2 \ldots, R \tag{9}$$

The trajectory $T^i_{hts}(t)$ given by the path optimizer with hyperbolic tangent sampling from $P^i(\tau)$ can be obtain by Equation (10), where the operator $\lceil a \rceil^b$ obtains the smallest element in set b and more than or equal to a, w is the sampling frequency.

$$\begin{cases} a = \lceil {}^{t}\!/_{w} \rceil^\tau \\ b = \lfloor {}^{t}\!/_{w} \rfloor^\tau \\ T^i_{hts}(t) = [P^i(a) - P^i(b)] \times f(\frac{t - w \times b}{w \times (a-b)}) \quad i = 1, 2 \ldots R \end{cases} \tag{10}$$

where $f(x)$ is the hyperbolic tangent function defined as Equation (11). Optimal trajectory with hyperbolic tangent sampling T^i_{hts} is generated by optimizing the path $Paths^i(\gamma)$ according to the referenced velocity.

$$f(x) = 0.5 \times (\frac{e^{2x+1} - 1}{e^{2x+1} + 1} + 1) \tag{11}$$

Compared with the path planner with original sampling, the one with hyperbolic tangent sampling can give the trajectory which is easier to track for MRS, which is demonstrated in the following experiment.

3. EXPERIMENTS

For demonstrating the feasibility of the proposed MPPTM, online path planning and tracking experiments for multi-UAV are designed in this section. However, it should be noted that the proposed MPPTM is suitable for online path planning and tracking of MRS not only in 2-D environments but also in 3-D environments.

3.1. Experimental Preparations

In order to accurately locate UAV groups in 3-D environments, the UWB positioning system is used, as shown in **Figure 5**. At least four UWB locator nodes are used to locate UAV groups in 3-D environments, and the distance between two adjacent nodes is 15m. A micro quadrotor produced by Zeronetech is used as an individual of the UAV group in this experiment, and it has a built-in UWB tag and wireless communication module.

The laptop with R5800u CPU and 32GB ROM is responsible for collecting the location data of UAVs *via* the UWB positioning system, recoding the flight data of UAVs, and controlling the fight of UAVs *via* WiFi communication. The system is implemented by Matlab and C++, the proposed path planner is coded by Matlab, the proposed path optimizer and the proposed path tracker are coded by C++. The obstacles in 3-D environments are realized by marking the environmental information $I(k)$ in the proposed path planner.

3.2. Online Path Planning and Tracking for 3-D Environments

The size of the outdoor environment used for testing is 10 m×10 m× 10 m, which is mapped by the NDN with size $50 \times 50 \times 50$ in the neural dynamic path planner. In the proposed path planner, we set K to 50, D to 5, J to 3, u to 0.3, r to 2, Ex to 50, and In to 5. In the proposed path optimizer, hyperbolic tangent sampling is used at the condition $w = 100$. The initial positions of targets and quadrotors are randomly located. The initial sizes

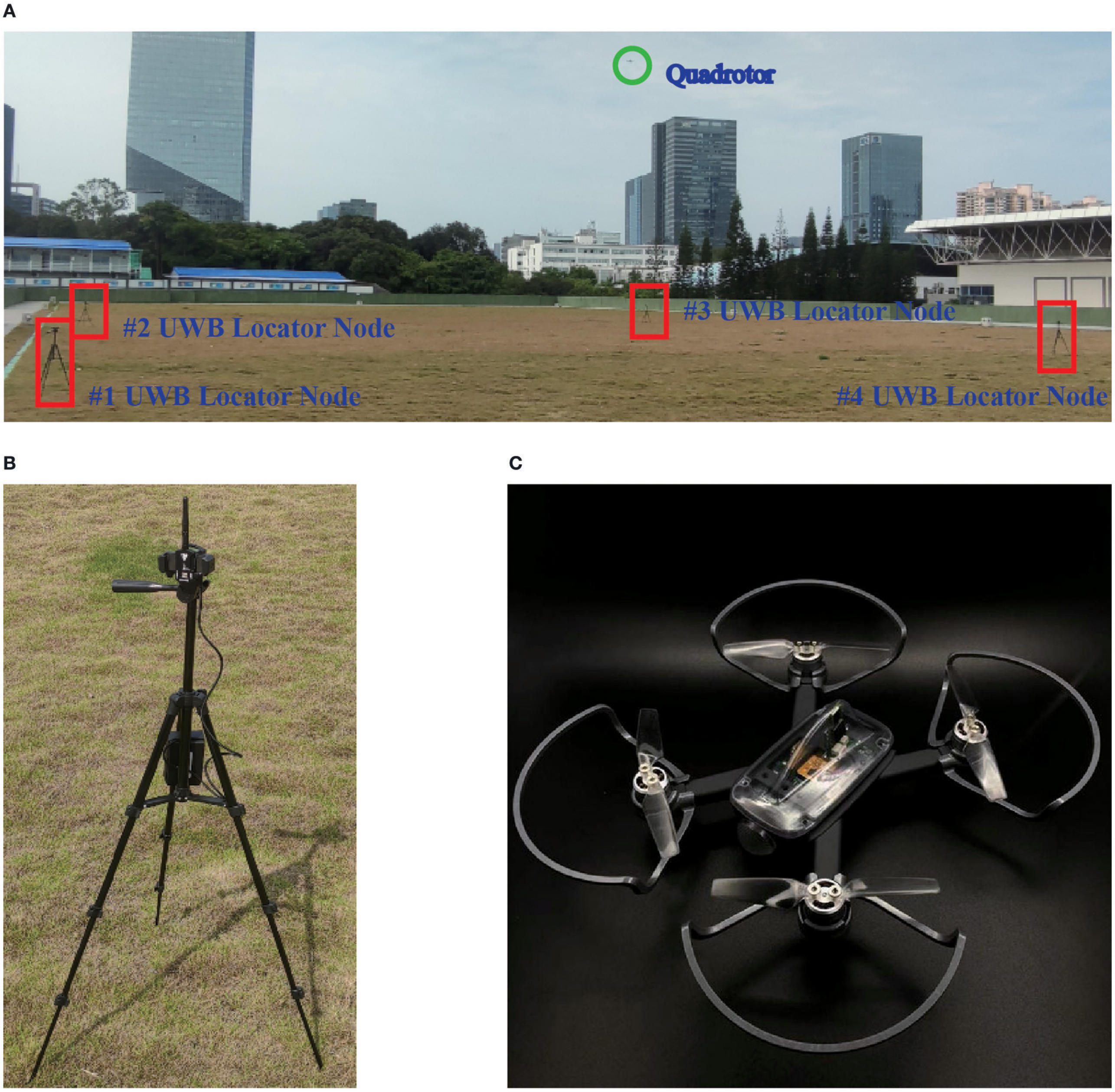


FIGURE 5 | The experimental preparations for online path planning and tracking. **(A)** The experimental site for online path planning and tracking of multi-UAVs. **(B)** The UWB locator node. **(C)** A micro quadrotor with UWB tag.

and initial positions of obstacles are randomly generated to mark the environmental information $I(k)$.

The snapshot of path planning and tracking at 30 s is shown in **Figure 6**, and the 1th quadrotor has finished capturing the target. **Figure 7** shows the snapshot of path planning and tracking at 65 s, where all quadrotors have captured targets.

The velocities of quadrotors in X, Y, Z directions are recoded and shown in **Figure 8**. The experimental result shows that all quadrotors can avoid collision and obstacles, and capture targets. The experimental result demonstrates that MPPTM is capable of online path planning and tracking of MRS in 3-D complex environments.

3.3. Online Path Planning and Tracking With Robot Fault

This experiment uses the same parameters and the same environment from the experiment in section 3.2. But in the online path planning and tracking, there are two quadrotors that have a certain probability 0.05 of failure during the process. Additionally, MPPTM needs to allocate

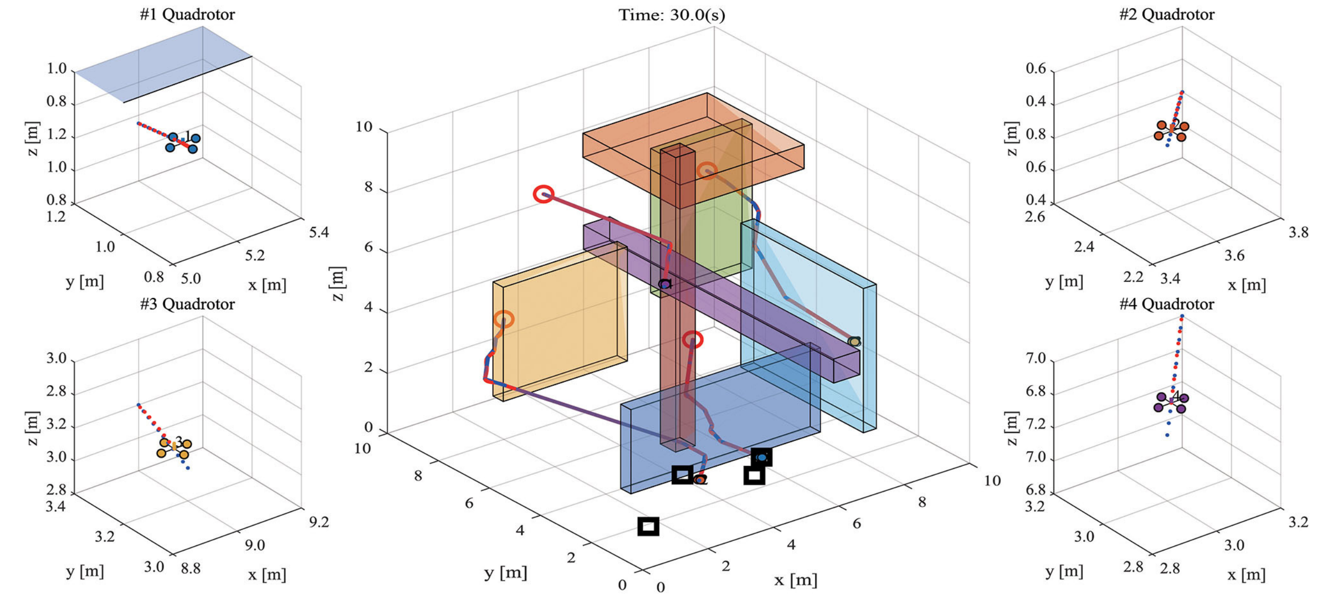


FIGURE 6 | The snapshot of path planning and tracking for UAVs in 3-D environments at 30 s. (Red circles and black squares present the initial positions of robots and targets respectively on the main diagram. Four sub diagrams zoom in the main diagram and show the current flight states of quadrotors. The red dot line presents the real path and the blue dot line presents the referenced path).

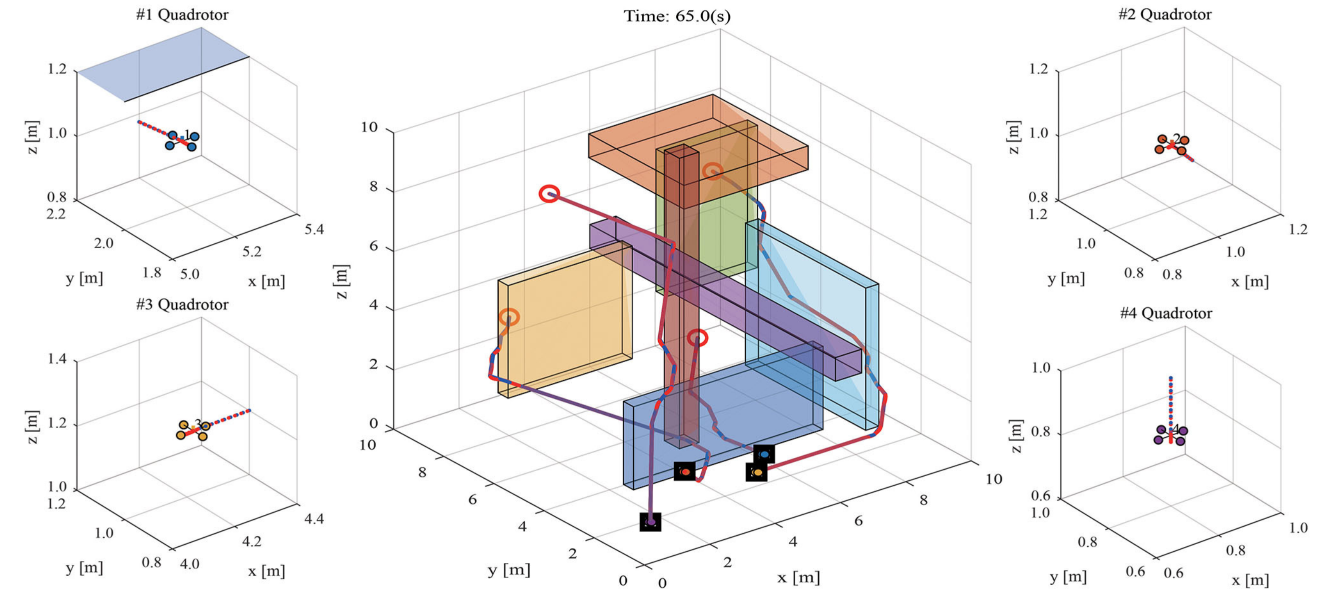


FIGURE 7 | The snapshot of path planning and tracking for UAVs in 3-D environments at 65 s. (The meaning of markers is the same as that in **Figure 6**).

these two targets to the other two quadrotors during the process.

As shown in **Figure 9**, after two quadrotors break down, the other quadrotors have also successfully completed four targets capturing. Therefore, the Experimental result indicates that MPPTM can deal with robot fault during online path planning and tracking, and it can reassign targets timely after robot fault.

3.4. Online Path Planning and Tracking With Dynamic Obstacles

This experiment is conducted in the 3-D workspace 8 m× 2.5 m× 2.5 m, and the experimental environment is mapped by the NDN with size 40 × 10 × 10. The experimental parameters are the same as those of the experiment in section 3.2. Obstacles in this experiment move at different times $(8\,s, 14\,s, 28s)$ in the order

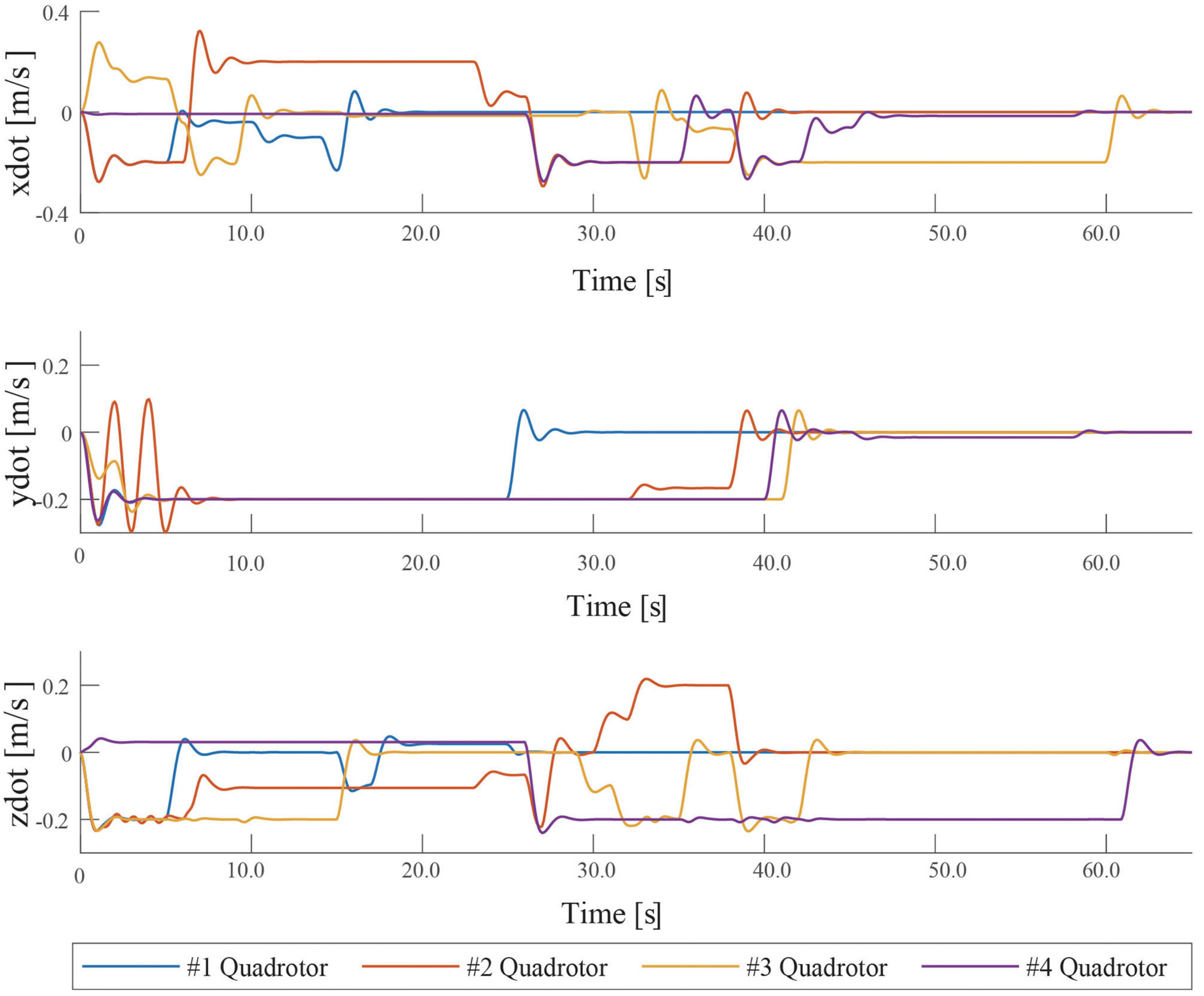


FIGURE 8 | The snapshot of path planning and tracking for UAVs in 3-D environments at 65 s.

shown in **Figure 10**. In order to better observe the experimental results, only two UAVs are used for the experiment. However, the proposed MPPTM can carry out online path planning and tracking for more robot individuals in an environment with dynamic obstacles.

Figure 10 shows that two quadrotors can avoid dynamic obstacles to capture targets during the path planning and tracking. The experimental result indicates that MPPTM carries out online path planning and tracking of MRS in an environment with dynamic obstacles.

4. DISCUSSION

Several comparative experiments on the proposed path optimizer and the proposed path planner are given in this section.

4.1. The Performance of the Proposed Path Optimizer

By using two different path optimizers, the actual paths and the desired paths are recoded in the experiment in section 3.2. The tracking error is the sum of the errors between the actual path and the desired path on the X, Y, and Z axes. The errors of the quadrotors are shown in **Figure 11**.

An indicator, defined as Equation (12), is given to measure the tracking error of UAVs during online path planning and tracking, where R is the number of quadrotors in UAVs, $error_i(t)$ is the error the i_{th} quadrotor, and t is the time.

$$E = \frac{1}{R}\sum_{i=1}^{R}\int_{0}^{+\infty} error_i(t)dt \tag{12}$$

In order to eliminate the influence of a series of factors as far as possible, such as wind speed, battery status, and measurement

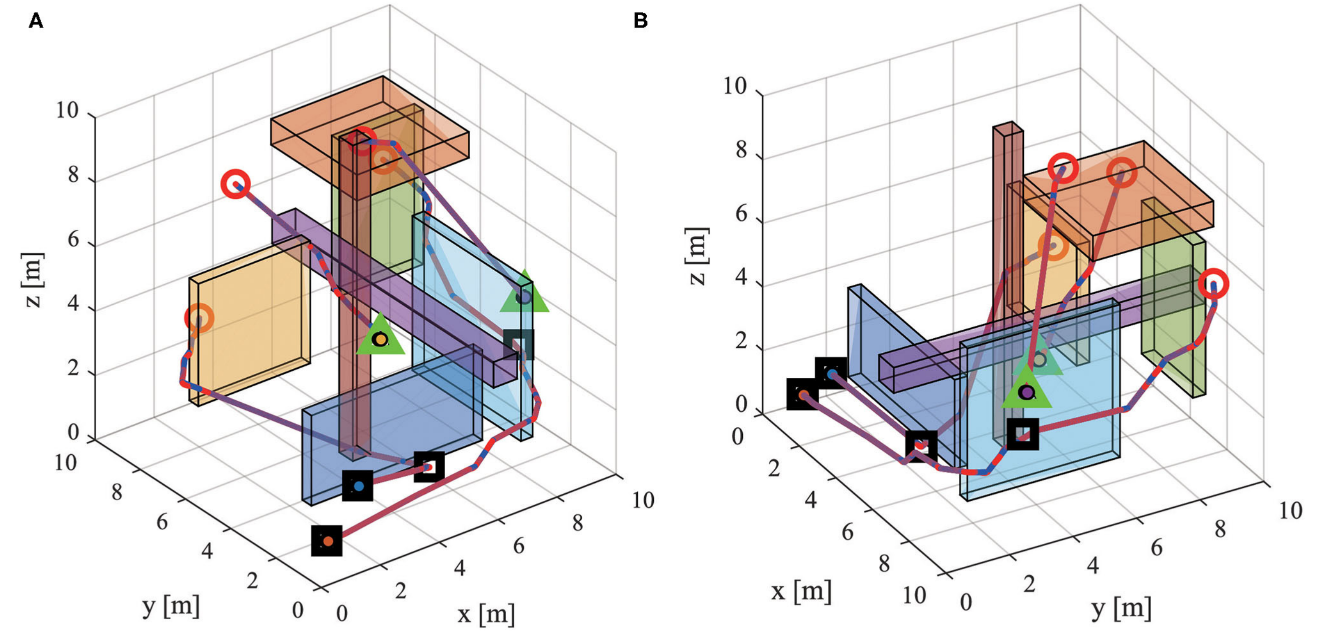


FIGURE 9 | The snapshot of path planning and tracking with quadrotor fault in 3-D environments. **(A)** Observation from view (-38, 30). **(B)** Observation from view (60, 30). (Green triangle represents the position of quadrotor fault. The meaning of other markers are the same as that in **Figure 6**).

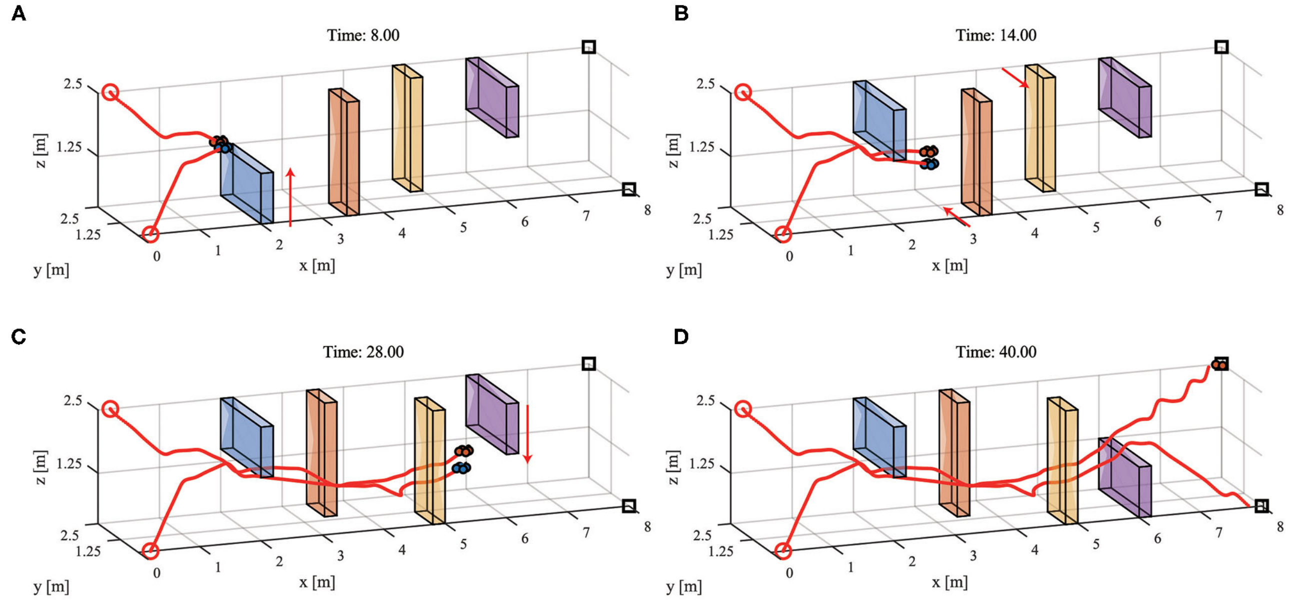


FIGURE 10 | The path planning and tracking with dynamic obstacles in 3-D environments. **(A)** The snapshot at 8 s. **(B)** The snapshot at 14 s. **(C)** The snapshot at 28 s. **(D)** The snapshot at 40 s. (Red arrow represents the direction of obstacles movement. The meaning of other markers are the same as that in **Figure 6**).

error, we test 20 experiments and collect the data of tracking error. These data are shown in **Table 1**.

Table 1 shows that, compared with the original path optimizer, the proposed path optimizer can reduce the tracking error of UAVs during online path planning and tracking.

4.2. The Time Performance of the Proposed Path Planner

By using the proposed neural activity algorithm, the proposed path planner has better time performance than other approaches (Li et al., 2009; Yi and Zhu, 2013; Sun et al., 2019;

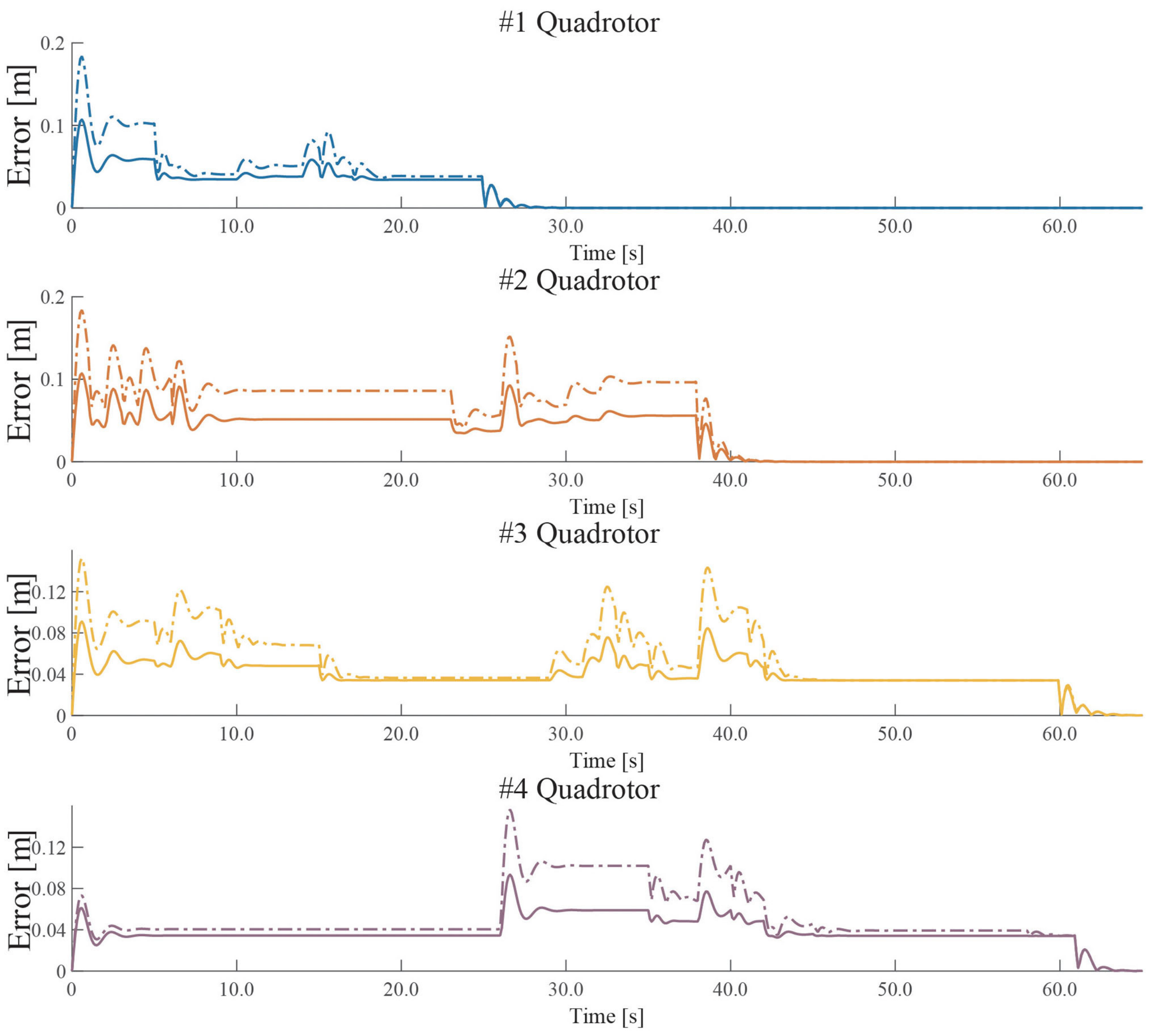


FIGURE 11 | The tracking errors of quadrotors by using two different path optimizers. (The solid line represents the tracking error by using the hyperbolic tangent path optimizer, and the dashed line represents the tracking error by using the original path optimizer).

TABLE 1 | The comparison of tracking error of unmanned aerial vehicle (UAVs).

Case	*Max(E)*	*Min(E)*	*Avg(E)*
The original path optimizer	1.5669	1.3768	1.4797
The proposed path optimizer	**1.0453**	**0.9765**	**1.0278**

Bold indicates that this indicator is the most superior to other cases.

Ni et al., 2020; Zhu et al., 2021). The comparative simulations are coded by Matlab, which is run on the PC with Intel i7-7700, 28GB ROM, and Win10 OS. The average time consumptions of single path planning by using different approaches and different sizes are shown in **Table 2**.

Table 2 indicates that the time performance of the proposed path planner is better than other approaches, and it is insensitive to the number of robots in MRS. The proposed path planner has excellent time performance, which makes it very suitable for superscalar pipelining mode in the proposed path planner of MPPTM for online path planning and tracking in 3-D environments.

5. CONCLUSION

The proposed model, MPPTM, can deal with task-level and control-level problems for path planning and tracking of MRS in 3-D environments. During the online path planning and tracking, our proposed model only needs to obtain the position

TABLE 2 | The average time consumptions of different approaches (s).

Case	Size	R=4	R=8	R=12	R=16
Our approach	30X30	**0.1738**	**0.1831**	**0.1686**	**0.1826**
	50X50	**0.8563**	**0.8322**	**0.8647**	**0.8534**
Li et al. (2009)	30X30	3.5184	7.4610	9.8962	15.7595
	50X50	18.3248	37.6820	46.9013	74.6896
Yi and Zhu (2013)	30X30	0.9502	0.8306	0.9999	0.9882
	50X50	5.0008	3.9363	5.0498	4.9412
Sun et al. (2019)	30X30	2.8010	5.6704	6.8507	10.5158
	50X50	13.2122	25.7745	34.4256	52.0586
Ni et al. (2020)	30X30	2.563	2.7275	2.9789	3.4007
	50X50	12.0897	13.8657	15.0516	18.0413
Zhu et al. (2021)	30X30	2.2681	4.5916	5.5473	8.5152
	50X50	10.6986	20.871	27.8763	42.1546

Bold indicates that this indicator is the most superior to other cases.

of MRS instead of relying on the complex sensor data of individual robots to plan paths. The cost of MRS equipped with such complex sensors is huge. Therefore, our proposed model, MPPTM, is a low-cost solution for the online path planning of MRS. Based on the UWB positioning system, MPPTM can carry out online path planning and high-accuracy path tracking for MRS in indoor or outdoor environments. It is suitable for application in manufacturing plants and industrial parks. Real experiments in this article demonstrate the applicability and effectiveness of the proposed model. In this model, the proposed path planner has excellent time performance to meet the requirement of superscalar pipelining mode. Meanwhile, the proposed path optimizer can guarantee the high-accuracy tracking of UAVs.

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/s.

AUTHOR CONTRIBUTIONS

XY and AZ contributed to the conception of the study. XY performed the experiment, the data analyses and wrote the manuscript. AZ contributed significantly to analysis and manuscript preparation. SY helped perform the analysis with constructive discussions. All authors listed have made a direct and intellectual contribution to the study, and approved it for publication.

FUNDING

This study was supported by National Natural Science Foundation of China (61273354) and Shenzhen stability support program for university (20200812112522002).

REFERENCES

Ali, H., Gong, D., Wang, M., and Dai, X. (2020). Path planning of mobile robot with improved ant colony algorithm and mdp to produce smooth trajectory in grid-based environment. *Front. Neurorobot.* 14:44. doi: 10.3389/fnbot.2020.00044

Ali, S., Mehmood, F., Dancey, D., Ayaz, Y., Khan, M. J., Naseer, N., et al. (2019). An adaptive multi-robot therapy for improving joint attention and imitation of asd children. *IEEE Access* 7, 81808–81825. doi: 10.1109/ACCESS.2019.2923678

Chen, M., and Zhu, D. (2019). A workload balanced algorithm for task assignment and path planning of inhomogeneous autonomous underwater vehicle system. *IEEE Trans. Cogn. Develop. Syst.* 11, 483–493. doi: 10.1109/TCDS.2018.2866984

Chen, Z., Alonso-Mora, J., Bai, X., Harabor, D. D., and Stuckey, P. J. (2021). Integrated task assignment and path planning for capacitated multi-agent pickup and delivery. *IEEE Robot. Autom. Lett.* 6, 5816–5823. doi: 10.1109/LRA.2021.3074883

Dai, X., Long, S., Zhang, Z., and Gong, D. (2019). Mobile robot path planning based on ant colony algorithm with a* heuristic method. *Front. Neurorobot.* 13:15. doi: 10.3389/fnbot.2019.00015

Dong, R., Liu, C., Wang, X., and Han, X. (2020). 3d path planning of uavs for transmission lines inspection. *Int. J. Robot. Autom.* 35, 146–158. doi: 10.2316/j.2020.206-0146

Han, S. D., and Yu, J. (2020). Ddm: Fast near-optimal multi-robot path planning using diversified-path and optimal sub-problem solution database heuristics. *IEEE Robot. Autom. Lett.* 5, 1350–1357. doi: 10.1109/LRA.2020.2967326

Koutras, D. I., Kapoutsis, A. C., and Kosmatopoulos, E. B. (2020). Autonomous and cooperative design of the monitor positions for a team of uavs to maximize the quantity and quality of detected objects. *IEEE Robot. Autom. Lett.* 5, 4986–4993. doi: 10.1109/LRA.2020.3004780

Krizmancic, M., Arbanas, B., Petrovic, T., Petric, F., and Bogdan, S. (2020). Cooperative aerial-ground multi-robot system for automated construction tasks. *IEEE Robot. Autom. Lett.* 5, 798–805. doi: 10.1109/LRA.2020.2965855

Li, H., Yang, S. X., and Seto, M. L. (2009). Neural-network-based path planning for a multirobot system with moving obstacles. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* 39, 410–419. doi: 10.1109/TSMCC.2009.2020789

Ma, L. (2020). Cooperative target tracking in balanced circular formation with time-varying radius. *Int. J. Robot. Autom.* 35, 86–100. doi: 10.2316/J.2020.206-0086

Mehmood, F., Ayaz, Y., Ali, S., De Cassia Amadeu, R., and Sadia, H. (2019). Dominance in visual space of asd children using multi-robot joint attention integrated distributed imitation system. *IEEE Access* 7, 168815–168827. doi: 10.1109/ACCESS.2019.2951366

Morita, T., Kashiwagi, N., Yorozu, A., Walch, M., Suzuki, H., Karagiannis, D., et al. (2018). "Practice of multi-robot teahouse based on printeps and evaluation of service quality," in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, Vol. 02, (Tokyo) 147–152.

Ni, J., Wang, X., Tang, M., Cao, W., Shi, P., and Yang, S. X. (2020). An improved real-time path planning method based on dragonfly algorithm for heterogeneous multi-robot system. *IEEE Access* 8, 140558–140568. doi: 10.1109/ACCESS.2020.3012886

Park, S., Min, Y., Ha, J., Cho, D., and Choi, H. (2019). A distributed admm approach to non-myopic path planning for multi-target tracking. *IEEE Access* 7, 163589–163603. doi: 10.1109/ACCESS.2019.2952235

Penin, B., Giordano, P. R., and Chaumette, F. (2018). Vision-based reactive planning for aggressive target tracking while avoiding collisions and occlusions. *IEEE Robot. Autom. Lett.* 3, 3725–3732. doi: 10.1109/LRA.2018.2856526

Queralta, J. P., Taipalmaa, J., Can Pullinen, B., Sarker, V. K., Nguyen Gia, T., Tenhunen, H., et al. (2020). Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision. *IEEE Access* 8, 191617–191643. doi: 10.1109/ACCESS.2020.3030190

Rubí, B., Pérez, R., and Morcego, B. (2019). A survey of path following control strategies for uavs focused on quadrotors. *J. Intell. Robot. Syst.* 1–25. doi: 10.1007/s10846-019-01085-z

Silic, M. and Mohseni, K. (2019). Field deployment of a plume monitoring uav flock. *IEEE Robot. Autom. Lett.* 4, 769–775. doi: 10.1109/LRA.2019.2893420

Sun, B., Zhu, D., Tian, C., and Luo, C. (2019). Complete coverage autonomous underwater vehicles path planning based on glasius bio-inspired neural network algorithm for discrete and centralized programming. *IEEE Trans. Cogn. Develop. Syst.* 11, 73–84. doi: 10.1109/TCDS.2018.2810235

Wang, J., Chi, W., Li, C., Wang, C., and Meng, M. Q. H. (2020a). Neural rrt*: Learning-based optimal path planning. *IEEE Trans. Autom. Sci. Eng.* 17, 1748–1758. doi: 10.1109/TASE.2020.2976560

Wang, J., Wang, J., and Che, H. (2020b). Task assignment for multivehicle systems based on collaborative neurodynamic optimization. *IEEE Trans. Neural Netw. Learn. Syst.* 31, 1145–1154. doi: 10.1109/TNNLS.2019.2918984

Xu, T., and Tang, L. (2021). Adoption of machine learning algorithm-based intelligent basketball training robot in athlete injury prevention. *Front. Neurorobot.* 14:117. doi: 10.3389/fnbot.2020.620378

Yi, X., and Zhu, A. (2013). "An improved neuro-dynamics-based approach to online path planning for multi-robots in unknown dynamic environments," in *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 1–6.

Yi, X., Zhu, A., Yang, S. X., and Luo, C. (2017). A bio-inspired approach to task assignment of swarm robots in 3-d dynamic environments. *IEEE Trans. Cybern.* 47, 974–983. doi: 10.1109/TCYB.2016.2535153

Yordanova, V., and Gips, B. (2020). Coverage path planning with track spacing adaptation for autonomous underwater vehicles. *IEEE Robot. Autom. Lett.* 5, 4774–4780. doi: 10.1109/LRA.2020.3003886

Yu, J., Dong, X., Li, Q., and Ren, Z. (2018). Practical time-varying formation tracking for second-order nonlinear multiagent systems with multiple leaders using adaptive neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* 29, 6015–6025. doi: 10.1109/TNNLS.2018.2817880

Yu, J., Su, Y., and Liao, Y. (2020). The path planning of mobile robot by neural networks and hierarchical reinforcement learning. *Front. Neurorobot.* 14:63. doi: 10.3389/fnbot.2020.00063

Zeng, Z., Lian, L., Sammut, K., He, F., Tang, Y., and Lammas, A. (2015). A survey on path planning for persistent autonomy of autonomous underwater vehicles. *Ocean Eng.* 110, 303–313. doi: 10.1016/j.oceaneng.2015.10.007

Zhou, L., Tzoumas, V., Pappas, G. J., and Tokekar, P. (2018). Resilient active target tracking with multiple robots. *IEEE Robot. Autom. Lett.* 4, 129–136. doi: 10.1109/LRA.2018.2881296

Zhu, D., Zhou, B., and Yang, S. X. (2021). A novel algorithm of multi-auvs task assignment and path planning based on biologically inspired neural network map. *IEEE Trans. Intell. Veh.* 6, 333–342. doi: 10.1109/TIV.2020.3029369

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.

frontiers in Neurorobotics

METHODS
published: 11 March 2022
doi: 10.3389/fnbot.2022.821991

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

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. Neurorobot. 16:821991. doi: 10.3389/fnbot.2022.821991

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.

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 reducting 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:

$$\theta_n^m = \overline{\theta}_n + \widetilde{\theta}_n; d_n^m = \overline{d}_n + \widetilde{d}_n \tag{1}$$

where n is the time index, d and θ represents relative distance and direction between consecutive frames. The pose measurement $(\theta_{\underset{\sim}{n}}^m, \underset{\sim}{d}_n^m)$ is then consisted of ground truth $(\bar{\theta}_n, \bar{d}_n)$ and error (θ_n, d_n) with SD δ_θ and δ_d.

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

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

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

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.

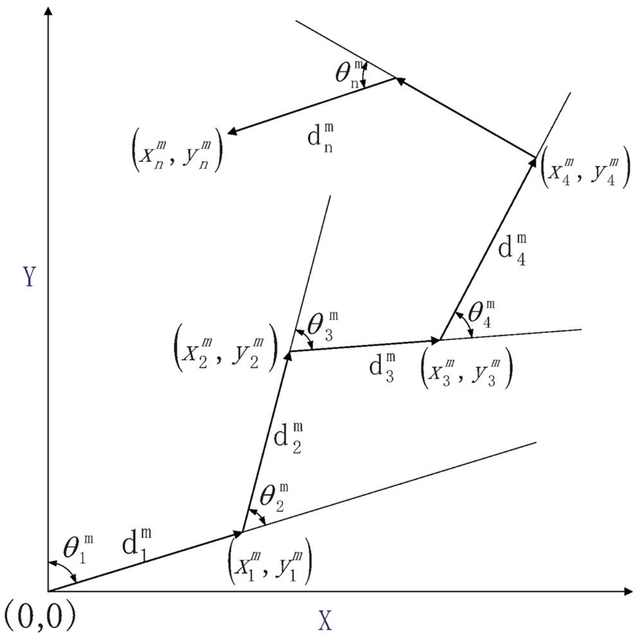


FIGURE 1 | Relationship between robot relative measurement and position.

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

$$\begin{aligned} x_n^m &= \overline{x_n} + \widetilde{x_n} \\ &= \sum\nolimits_{i=1}^{n} \left(d_i^m \sin \sum\nolimits_{j=1}^{i} \theta_j^m \right) \\ &= \sum\nolimits_{i=1}^{n} \left(\left(\overline{d_i} + \widetilde{d_i} \right) \sin \sum\nolimits_{j=1}^{i} \left(\overline{\theta_j} + \widetilde{\theta_j} \right) \right) \\ &= \left(\sum\nolimits_{i=1}^{n} \overline{d_i} + \sum\nolimits_{i=1}^{n} \widetilde{d_i} \right) \cdot \left[\sin \sum\nolimits_{j=1}^{i} \overline{\theta_j} \cos \sum\nolimits_{j=1}^{i} \widetilde{\theta_j} \right. \\ &\quad \left. + \cos \sum\nolimits_{j=1}^{i} \overline{\theta_j} \sin \sum\nolimits_{j=1}^{i} \widetilde{\theta_j} \right] \end{aligned} \tag{4}$$

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

$$\begin{aligned} \widetilde{x_n} = \sum\nolimits_{i=1}^{n} \overline{d_i} &\left[\begin{array}{l} \sin \sum\nolimits_{j=1}^{i} \overline{\theta_j} \left(\cos \sum\nolimits_{j=1}^{i} \widetilde{\theta_j} - 1 \right) \\ + \cos \sum\nolimits_{j=1}^{i} \overline{\theta_j} \sin \sum\nolimits_{j=1}^{i} \widetilde{\theta_j} \end{array} \right] \\ + \sum\nolimits_{i=1}^{n} \widetilde{d_i} &\left[\begin{array}{l} \sin \sum\nolimits_{j=1}^{i} \overline{\theta_j} \cos \sum\nolimits_{j=1}^{i} \widetilde{\theta_j} \\ + \cos \sum\nolimits_{j=1}^{i} \overline{\theta_j} \sin \sum\nolimits_{j=1}^{i} \widetilde{\theta_j} \end{array} \right] \end{aligned} \tag{5}$$

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:

$$E\left[\widetilde{x}|\overline{\theta}, \overline{d}\right] = \sum\nolimits_{i=1}^{n} \overline{d_i} \left[\sin \sum\nolimits_{j=1}^{i} \overline{\theta_j} \left(e^{-\frac{i\delta_\theta^2}{2} - 1} \right) \right] \tag{6}$$

$$\begin{aligned} \mathrm{var}\left(\widetilde{x}|\overline{\theta}, \overline{d}\right) &= E\left[\widetilde{x}|\overline{\theta}, \overline{d}\right] - E^2\left[\widetilde{x}|\overline{\theta}, \overline{d}\right] \\ &= A + B + C - E^2\left[\widetilde{x}|\overline{\theta}, \overline{d}\right] \end{aligned} \tag{7}$$

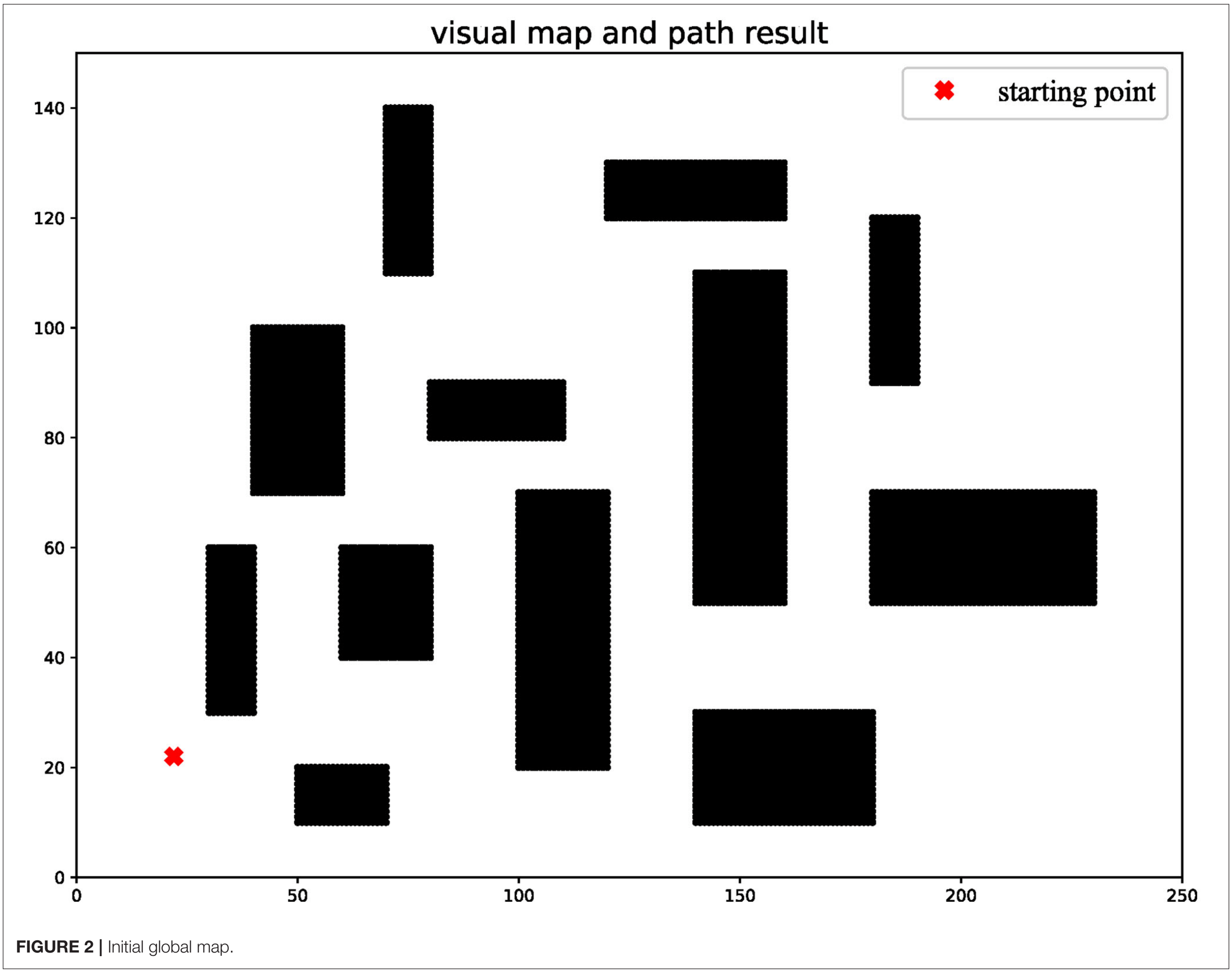


FIGURE 2 | Initial global map.

where:

$$A = \sum\nolimits_{i=1}^{n} \overline{d_i^2} \left[\sin^2 \sum\nolimits_{j=1}^{i} \overline{\theta_j} \left(0.5e^{-2i\delta_\theta^2} + 1.5 - 2e^{-\frac{i\delta_\theta^2}{2}} \right) \right. \\ \left. + 0.5\cos^2 \sum\nolimits_{j=1}^{i} \overline{\theta_j} \left(e^{-2i\delta_\theta^2} + 1 \right) \right] \quad (8)$$

$$B = 2 \sum\nolimits_{i=1}^{n-1} \sum\nolimits_{p=1+i}^{n} \overline{d_i d_p} \\ \left\{ \begin{array}{l} \sin^2 \sum_{j=1}^{i} \overline{\theta_j} \cos \Delta\overline{\theta} \left[\begin{array}{l} 1 + 0.5 \left(1 + e^{-2i\delta_\theta^2} \right) e^{-0.5(p-i)\delta_\theta^2} \\ e^{-0.5i\delta_\theta^2} - e^{-0.5i\delta_\theta^2} e^{-0.5(p-i)\delta_\theta^2} \end{array} \right] \\ + \sin \sum_{j=1}^{i} \overline{\theta_j} \sin \Delta\overline{\theta} \cos \sum_{j=1}^{i} \overline{\theta_j} \\ \left[\begin{array}{l} 1 + 0.5 \left(1 + e^{-2i\delta_\theta^2} \right) e^{-0.5(p-i)\delta_\theta^2} + 1 \\ -e^{-0.5i\delta_\theta^2} - e^{-0.5i\delta_\theta^2} e^{-0.5(p-i)\delta_\theta^2} \\ -0.5 \left(1 - e^{-2i\delta_\theta^2} \right) e^{-0.5(p-i)\delta_\theta^2} \end{array} \right] \\ + \cos^2 \sum_{j=1}^{i} \overline{\theta_j} \cos \Delta\overline{\theta} \cdot 0.5 \left(1 - e^{-2i\delta_\theta^2} \right) e^{-0.5(p-i)\delta_\theta^2} \end{array} \right\} \quad (9)$$

$$C = \sum\nolimits_{i=1}^{n} \left[0.5\sin^2 \sum\nolimits_{j=1}^{i} \overline{\theta_j} \left(e^{-2i\delta_\theta^2} + 1 \right) \right. \\ \left. + 0.5\cos^2 \sum\nolimits_{j=1}^{i} \overline{\theta_j} \left(1 - e^{-2i\delta_\theta^2} \right) \right] \quad (10)$$

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:

$$E\left[\widetilde{x_n^m}\right] = \sum\nolimits_{i=1}^{n} d_i^m \left(e^{-i\delta_\theta^2} - e^{-0.5i\delta_\theta^2} \right) \sin \sum\nolimits_{j=1}^{i} \theta_j^m \quad (11)$$

$$\mathrm{var}\left(\widetilde{x_n^m}\right) = A_1 + B_1 + C_1 - E^2\left[\widetilde{x_n^m}\right] \quad (12)$$

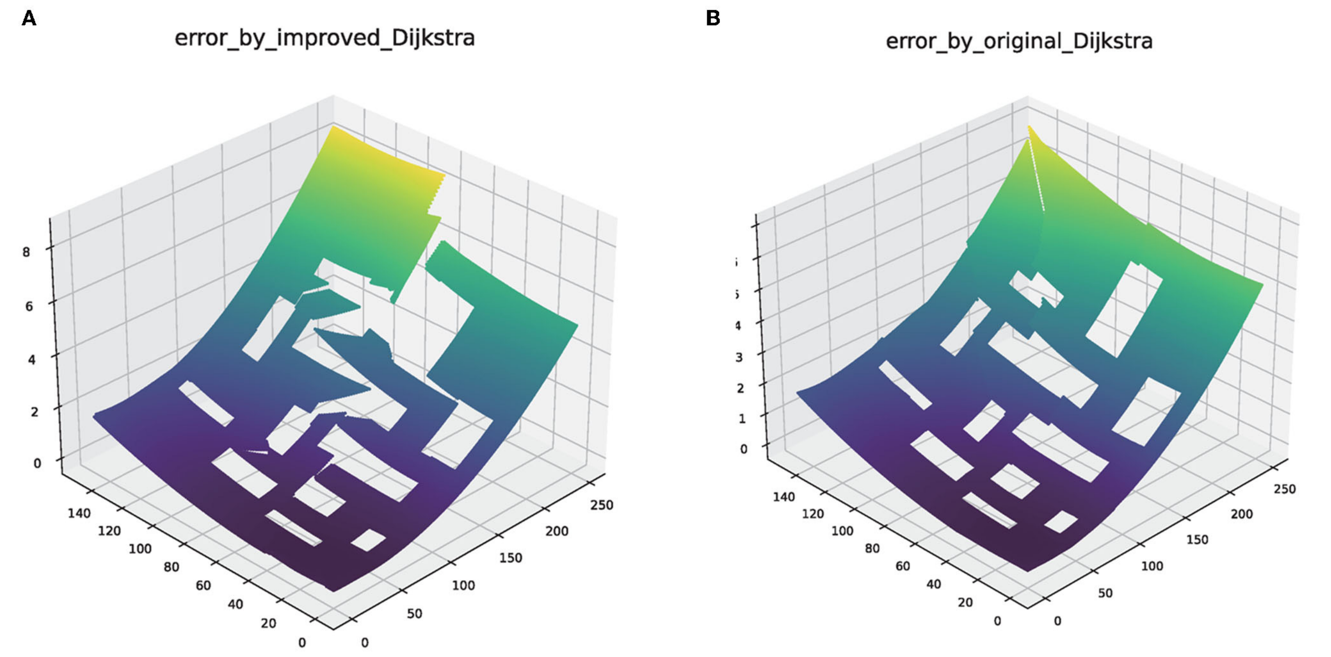


FIGURE 3 | The error map generated by the original Dijkstra and improved Dijkstra. **(A)** The error map of original Dijkstra. **(B)** The error map of original Dijkstra.

where:

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

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

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

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

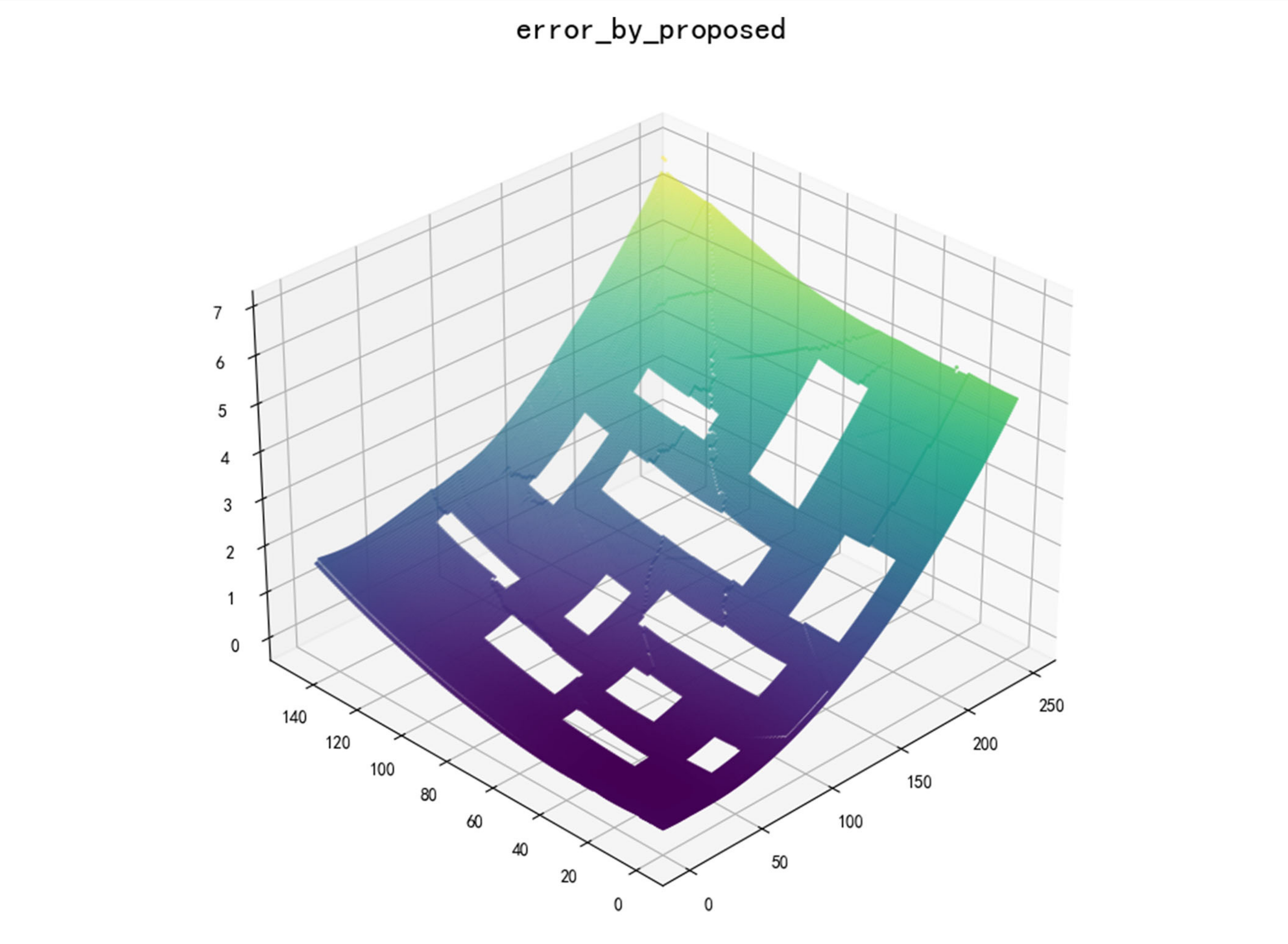


FIGURE 4 | Global error map after iteration.

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 $\boldsymbol{G}$ 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 $\boldsymbol{S}$ to store those vertices that have been traversed and initialize a set $\boldsymbol{Q}$ which includes all vertices $\boldsymbol{G}.V$. $\boldsymbol{Q}$ 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 *trun_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 $\boldsymbol{G}.Adj[u]$. 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, $\boldsymbol{G}.Adj\_limited[u]$ 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

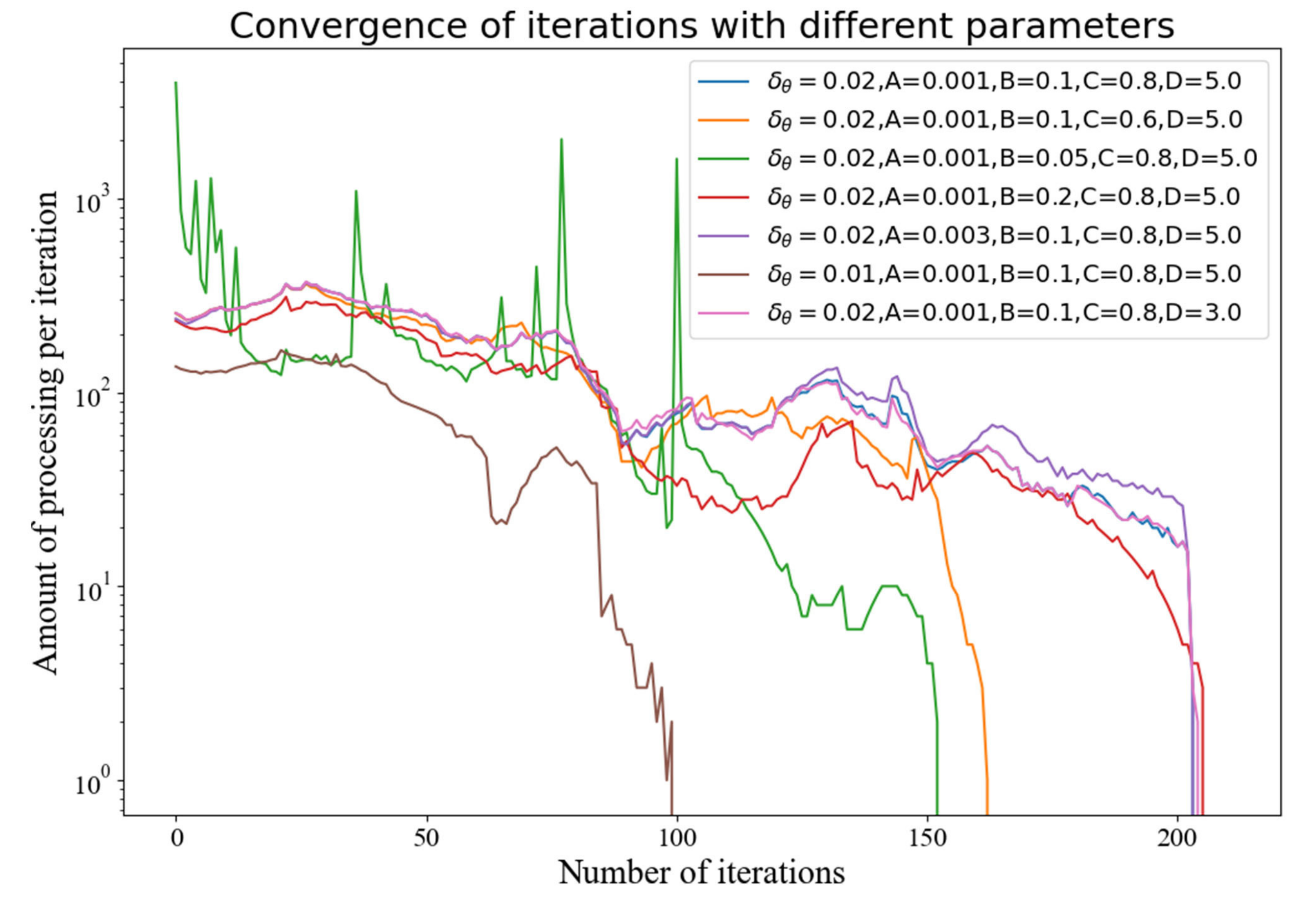


FIGURE 5 | The number of calculations per iteration under different parameters.

$\omega(u, v, u.\pi)$ 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 $(\bar{\theta}, \bar{d})$ 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: $v.value = \max(\boldsymbol{G}.Adj[u].error - v.error)/v.d$, 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 $v.value\_absolute = \max(\boldsymbol{G}.Adj[u].error - v.error)$. To adapt to different scale maps, we simultaneously apply two benchmarks to update the queue ***Q_dif***.

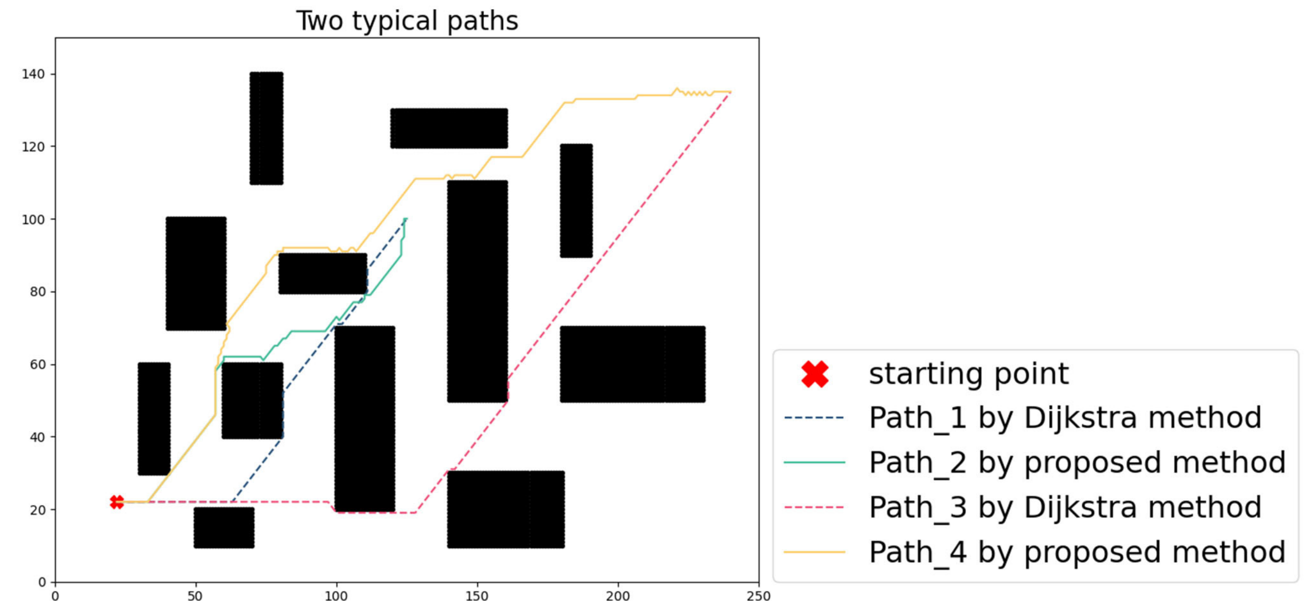


FIGURE 6 | Comparison of two typical paths.

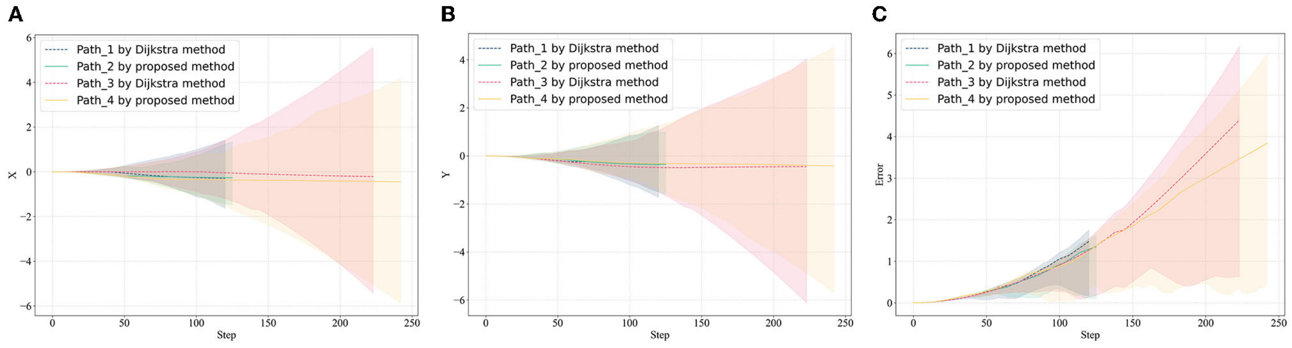


FIGURE 7 | Error estimation in path planning domain. **(A)** Average error in *X*-axis, **(B)** average error in *Y*-axis, and **(C)** average distance error.

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

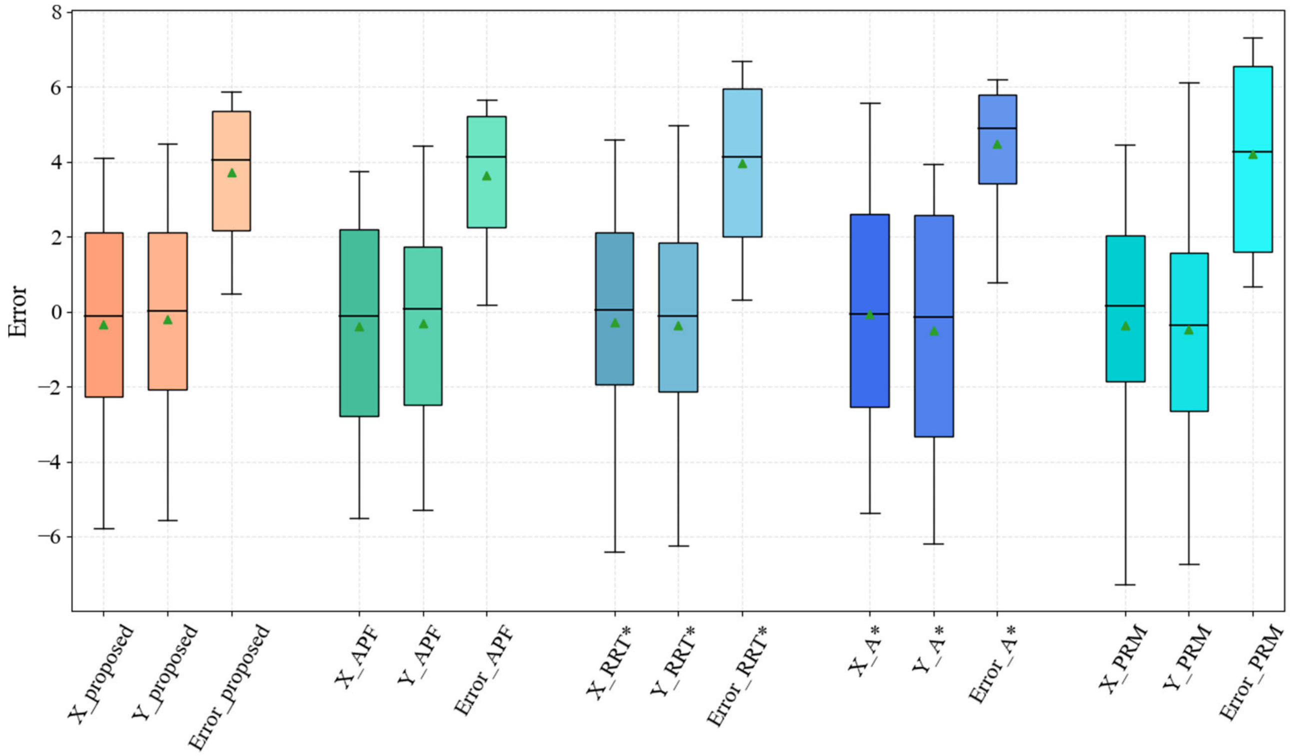


FIGURE 8 | Results of the proposed method compared with artificial potential field (APF), RRT*, A*, and probabilistic roadmap (PRM) methods.

For point set $\boldsymbol{Q\_dif}$, 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($\boldsymbol{G}, s$)
Ensure: Original global error map $\boldsymbol{G\_error}$
INITIALIZE-SINGLE-SOURCE($\boldsymbol{G}, s$)
for each node $n \in \boldsymbol{G}.V$ do
    $n.turn\_num = 0$
    $n.d = \infty$
    $n.\pi = NIL$
end for
$s.turn\_num = 0$
$\boldsymbol{S} = \varnothing$
$\boldsymbol{Q} = \boldsymbol{G}.V$
while $\boldsymbol{Q} \neq \varnothing$ do
    $u$ = heappop ($\boldsymbol{Q}$)
    $\boldsymbol{S} = \boldsymbol{S} \cup \{u\}$
    for each node $v \in \boldsymbol{G}.Adj\_limited[u]$ do
        if $v.turn\_num > u.turn\_num + \omega(u, v, u.\pi)$ then
            $v.turn\_num = u.turn\_num + \omega(u, v, u.\pi)$
            $v.\pi = u$
            $v.d = u.d + \omega\_d(u, v)$
            $v.error$ = Equation (6)
            $v.path = PATH(s, v)$
        end if
    end for
end while
```

Algorithm 2: Update Iteration Point Set.

```
Require: Original global error map $\boldsymbol{G\_error}$
Ensure: Point set to be iterated $\boldsymbol{Q\_dif}$
$\boldsymbol{Q\_dif} = \varnothing$
$\boldsymbol{Q\_temp} = \boldsymbol{G}.V$
while $\boldsymbol{Q\_temp} \neq \varnothing$ do
    $u$ = heappop ($\boldsymbol{Q\_temp}$)
    for each node $v \in \boldsymbol{G}.Adj[u]$ do
        if $v.value > A$ or $v.value\_absolute > B$ then
            $\boldsymbol{Q\_dif} = \boldsymbol{Q\_dif} \cup \{v\}$
        end if
    end for
end while
```

Algorithm 3: Error Map Iteration Strategy.

```
Require: Point set to be iterated $\boldsymbol{Q\_dif}$
Ensure: Final error map $\boldsymbol{G\_error}$
while $\boldsymbol{Q\_dif} \neq \varnothing$ do
    $u$ = heappop ($\boldsymbol{Q\_dif}$)
    for each node $v \in \boldsymbol{G}.Adj[u]$ do
        if $ERROR(v.path + u) < u.error$ and $CORRELATION(v.path + u, u.path) < C$ then
            $u.\pi = v$
            $u.path = v.path + u$
            $u.error = ERROR(v.path + u)$
        end if
    end for
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.

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 μAUV 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/RSJ 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.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 18 March 2022
doi: 10.3389/fnbot.2022.840332

LEACH Protocol Optimization Based on Weighting Strategy and the Improved Ant Colony Algorithm

Xuezhen Cheng[1], Chuannuo Xu[1], Xiaoqing Liu[1,2], Jiming Li[1] and Junming Zhang[3]*

[1] College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao, China, [2] Shandong Senter Electronic Co, Zibo, China, [3] College of Energy and Mining Engineering, Shandong University of Science and Technology, Qingdao, China

This article aims to address problems in the current clustering process of low-energy adaptive clustering hierarchy (LEACH) in the wireless sensor networks, such as strong randomness and local optimum in the path optimization. This article proposes an optimal combined weighting (OCW) and improved ant colony optimization (IACO) algorithm for the LEACH protocol optimization. First, cluster head nodes are updated *via* a dynamic replacement mechanism of the whole network cluster head nodes to reduce the network energy consumption. In order to improve the quality of the selected cluster head nodes, this article proposes the OCW method to dynamically change the weight according to the importance of the cluster head node in different regions, in accordance with the three impact factors of the node residual energy, density, and distance between the node and the sink node in different regions. Second, the network is partitioned and the transmission path among the clusters can be optimized by the transfer probability in IACO with combined local and global pheromone update mechanism. The efficacy of the proposed LEACH protocol optimization method has been verified with MATLAB simulation experiments.

Keywords: optimal combination weighting, improved ant colony optimization, path superiority, LEACH optimization, routing protocol

OPEN ACCESS

Edited by:
Yimin Zhou,
Shenzhen Institutes of Advanced Technology (CAS), China

Reviewed by:
Avishek Nag,
University College Dublin, Ireland
Alireza Mousavi,
Brunel University London, United Kingdom

****Correspondence:***
Junming Zhang
truegis@163.com

Received: *21 December 2021*
Accepted: *14 February 2022*
Published: *18 March 2022*

Citation:
Cheng X, Xu C, Liu X, Li J and Zhang J (2022) LEACH Protocol Optimization Based on Weighting Strategy and the Improved Ant Colony Algorithm. Front. Neurorobot. 16:840332. doi: 10.3389/fnbot.2022.840332

INTRODUCTION

It is known that wireless sensor networks (WSNs) are composed of many spatially distributed sensor nodes with limited energy (Efe et al., 2013), whereas the sensor nodes are usually powered by light batteries. Frequent charging or battery replacement of the sensor nodes would cause inconvenience to maintenance; hence, balancing the energy consumption of sensor nodes and prolonging the network lifetime are the two most important indicators to evaluate the performance of the WSNs (Tripathi et al., 2013; Mukherjee et al., 2018). The energy loss of the network can directly affect the performance and life of the network, which should be delicately dealt with to keep the low energy loss of the network in the communication process (Yan et al., 2018; Mohar et al., 2020; Lv et al., 2021). Many routing protocols can be used in WSNs, where hierarchical routing protocols are the most widely adopted. The typical hierarchical routing protocols include low-energy adaptive clustering hierarchy (LEACH), power-efficient gathering in sensor information systems (PEGASIS), threshold-sensitive energy-efficient sensor network (TEEN) protocol, and hybrid energy-efficient distributed (HEED) clustering approach, which can gather the nodes into clusters to form a specific hierarchy (Galkin, 2018). Particularly, the typical LEACH protocol was proposed by Heinzelman (2000) and adopted the "wheel" cycle mode for the first time, which is widely applied due to its low power consumption, node equality, and self-clustering adaptation. In such a protocol, each sensor node contains a clustering algorithm and a data transmission algorithm among nodes (Chen et al., 2015), where the clustering algorithm can randomly change the cluster head nodes by comparing the size of the random number and threshold. After deployment, all sensor nodes self-organize to form different clusters. Generally, each cluster contains a cluster head node and multiple sensor member nodes. The nodes of the specific cluster can only hop one step inside the cluster to the cluster head. The cluster head node of each cluster will also hop one step to transmit the received information to the sink node to complete an iteration circle. The structure of the LEACH protocol is illustrated in **Figure 1**.

In the LEACH protocol, the energy of the sensor nodes is evenly distributed, and the network duration can be prolonged by balancing the energy consumption of the sensor nodes (Wang et al., 2018). However, the random selection of the cluster head nodes in the protocol could lead to uneven distribution of the cluster head nodes, which is prone to a large or small number of member nodes in the cluster (Marappan and Rodrigues, 2016). Each cluster head node and the base station adopt a single-hop transmission mode, which could easily cause premature death of the remote cluster head, poor expansibility, and uneven energy consumption phenomena in the network. The replacement mechanism of clusters using wheels will speed up the replacement frequency of the cluster head, which could result in more times of the information transmission in the network and shortened network life (Rahman et al., 2013; Jameii and Maadani, 2016). Then, the LEACH protocol has been improved with the fractional lion (FLION) algorithm to generate the optimal route (Sirdeshpande and Udupi, 2017), whereas the fractional derivative is introduced to detect the neighbor solution, and the forward link algorithm is also used to select channels to improve the network survival duration. However, the distribution density of the nodes is not considered in the fitness function. Lalwani et al. (2018) used the harmony search algorithm (HSA) to determine the optimal routing, where the fitness function is designed considering the node density, energy, and distance factor, by choosing the smallest distance nodes for data transmission so as to achieve the energy consumption reduction of the nodes. However, when the cluster contains non-local nodes, the network performance will be affected. Ning et al. (2017), Ezhilarasi (2019) adopted an improved particle swarm optimization (PSO) to optimize the clusters of the WSN process. However, the selection rules of the cluster head node are relatively complex, and the cluster scale rapid update frequency consumes high energy due to fast convergence. At the same time, local optimum would easily occur. Hence, certain algorithms are proposed to tackle such problems, where the cuckoo search algorithm (CSA) (Huang and Hua, 2020) and the fruit fly algorithm (FFA) (Dai et al., 2020) are developed to optimize the cluster and routing protocol and to determine the transmission path according to the distance, energy, and trust value from the node to the base station such factors. However, the generalization capability of the surviving nodes during communication has not been considered (Maheshwari et al., 2021).

In recent years, some improved heuristic intelligent algorithms have become research hotspots. For instance, the ant colony optimization (ACO) algorithm has been extensively studied and applied since it can assist to select the optimal path for the fused data transmission and the network life cycle extension (Ding, 2020). Experts and scholars have explored the

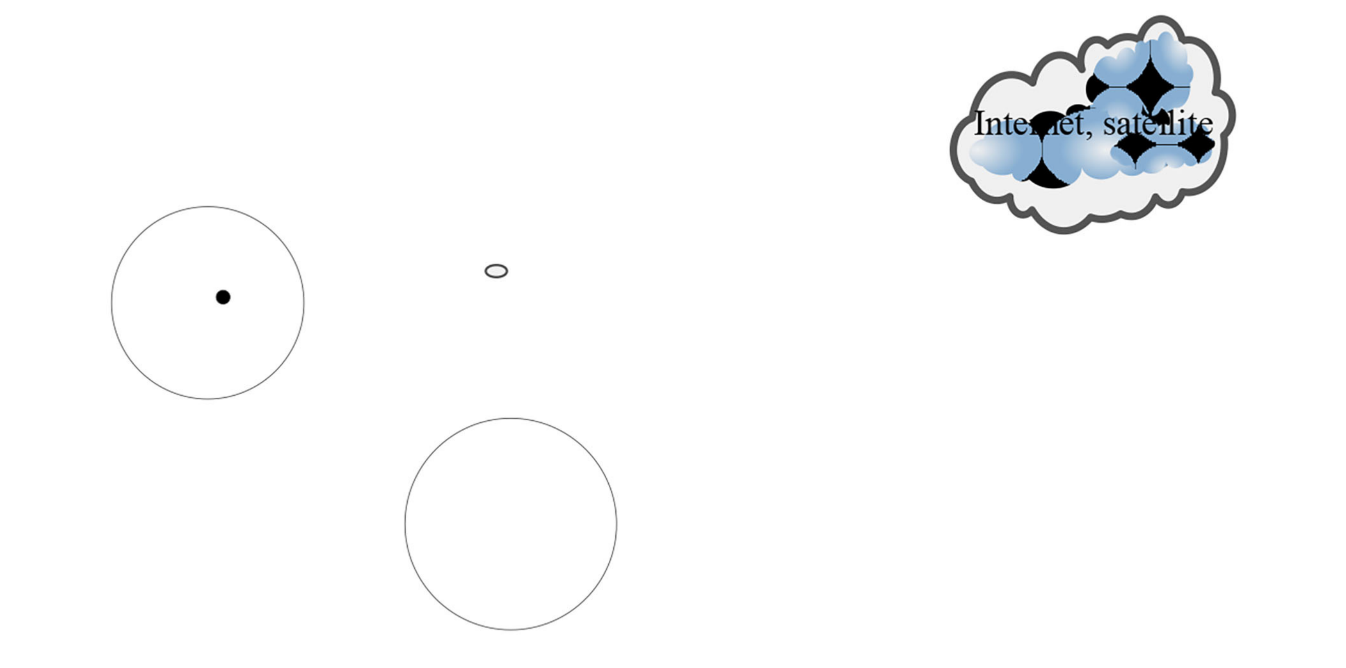


FIGURE 1 | Network structure diagram of LEACH protocol.

TABLE 1 | WSNs cluster routing protocol performance comparison.

Protocol	Energy saving performance	Cluster creation time	Uniform distribution of cluster head nodes	Path selection
LEACH	Poor	Quick	Difference	Single jump
TEEN	Good	Faster	Difference	Single jump
PEGASIS	Better	Slow	–	More jumps
HEED	Better	Slower	Good	More jumps
EEUC	Good	Medium	Medium	More jumps

ACO algorithms to reduce the energy consumption of the nodes. An ACO multipath routing protocol has been proposed based on the angle factor and entropy to optimize the cluster head nodes with high reliability in the process of data transmission (Hou et al., 2017; Liu and Li, 2018). To tackle the energy consumption balance problem, Zou and Qian (2019) proposed improved ACO (IACO) with a sensor node transfer function and pheromone updating routing rules for optimal WSN routing but with little consideration of the impact factors. Nayyar and Singh (2020) proposed an energy-saving multipath routing protocol based on the ACO algorithm for dynamic WSNs, in which the optimal path for the adjacent nodes is designed, but the scalability is poor. Therefore, it is of great significance to investigate the method to improve the transmission performance of the WSNs and select the optimal data transmission path quickly and effectively.

This article aims to propose an IACO algorithm for the LEACH protocol optimization. The specific sections are as follows: the first section is the improvement of the clustering algorithm based on optimal combination weighting. Furthermore, the designed transmission path between clusters based on the IACO algorithm is proposed in section Improved Clustering Algorithm Based on OCW. Simulation experiments for the proposed LEACH protocol optimization verification are described in section Energy Consumption Analysis of the Sensor Node. The Conclusion is given in section Optimization of the Transmission Path Between Clusters Based on IACO. **Table 1** compares the energy-saving, cluster establishment time, uniform distribution of cluster head nodes, and path selection of the representative typical routing protocols.

IMPROVED CLUSTERING ALGORITHM BASED ON OCW

The Update Mechanism of the Cluster Head Node

In the LEACH, cluster head nodes are replaced at each round, which would increase the network power consumption. Therefore, the cluster head nodes are updated within the existed clusters for the whole network clusters. The node whose energy is lower than the average energy E_a in the cluster loses the qualification to be the cluster head node, whereas the node with the highest remaining energy in the cluster is defined as the cluster head node of the next round, and the average energy E_a in the cluster is defined as,

$$\boldsymbol{E_a}=\sum \frac{\boldsymbol{E_i}}{\boldsymbol{N_{alive}}} \tag{1}$$

where E_i is the remaining energy of the surviving node i in the cluster and N_{alive} is the total number of the surviving nodes. If the energy of the current cluster head nodes is lower than this value, WSNs update the cluster size.

The average energy E_{all} of all the cluster head nodes in the whole network is calculated as,

$$\boldsymbol{E_{all}}=\sum \frac{\boldsymbol{E_r}}{\boldsymbol{B}} \tag{2}$$

where E_r and B are the residual energy and the total number of the cluster head nodes.

The Threshold of the Cluster Head Node Selection

Improved Threshold for Cluster Head Node Selection

The improved LEACH protocol has the same rules as cluster head selection in LEACH protocol, while the selection threshold is defined as $T_{imp}\left(n\right)$, written as,

$$\boldsymbol{T_{imp}\left(n\right)}=\begin{cases}\omega_1\frac{\boldsymbol{E_i}}{\boldsymbol{E_0}}+\omega_2\frac{\boldsymbol{E_{alive}}}{\boldsymbol{N_{alive}}}+\omega_3\frac{\boldsymbol{d}_{\max}-\boldsymbol{d_{is}}}{\boldsymbol{d}_{\max}-\boldsymbol{d}_{\min}}, & \boldsymbol{n}\in\boldsymbol{G}\\ \boldsymbol{0}, & \boldsymbol{n}\notin\boldsymbol{G}\end{cases} \tag{3}$$

$$\boldsymbol{R}=\sqrt{\frac{\boldsymbol{S}}{\pi\boldsymbol{Np}}}$$

$$\boldsymbol{l_{alive}}=\left\{\boldsymbol{S_j}\middle|\boldsymbol{d_{ij}}\leq\boldsymbol{R},\boldsymbol{S_j}\in\boldsymbol{S}\right\}$$

where E_0 is the initial energy of the survived node, l_{alive} is the number of the existing neighboring nodes, $d_{\max}$ is the distance from node i to the sink node, $d_{\max}$ and $d_{\min}$ are the farthest and closest distance from the surviving node to the sink node, respectively, and $\{\omega_1, \omega_2, \omega_3\}$ are the weights of the node energy consumption, density, and distance, respectively.

In the improved threshold, the probability of being selected as the cluster head node is related to the energy, density, and distance from the sink node. As the remaining energy of the node decreases and the smaller the energy ratio is, the lower the probability of the node being selected as the cluster head node. If node i is the closest to the sink node, the distance factor is 1, otherwise, the distance factor is 0. Therefore, the range of the distance factor is between 0 and 1, and the closer the distance to the sink node, the closer the value to be 1.

Threshold Weight Selection

To ensure the rationality for the cluster head nodes selection, the optimal combined weighting (OCW) is used to adjust the three weights of the improved threshold, which combines the analytic hierarchy process (AHP) and the entropy method to ensure the objectivity of the threshold weight.

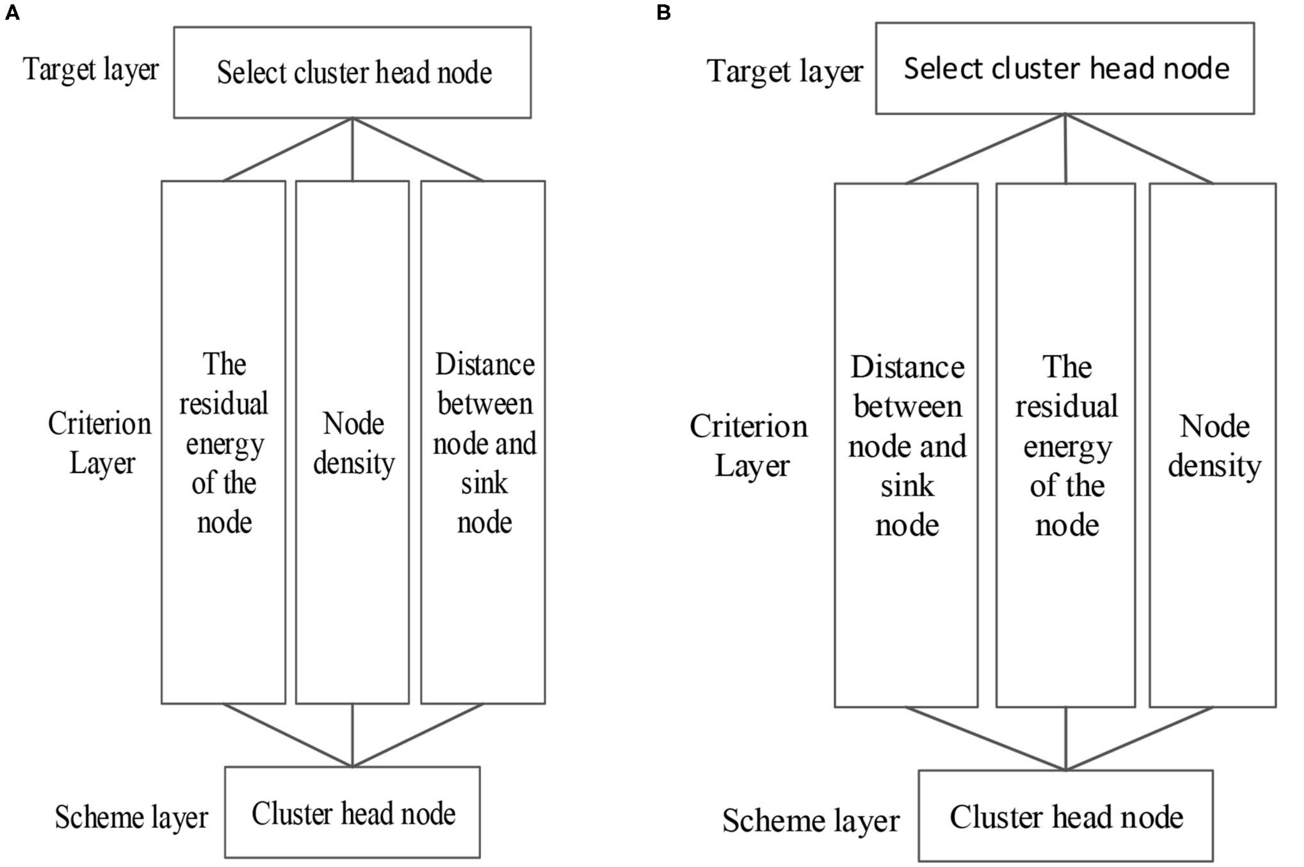


FIGURE 2 | Hierarchical model of threshold weight.

Weight of AHP

The AHP method can decompose the problems into different levels with an evaluation index matrix to solve the maximum eigenvalues of the matrix and the corresponding eigenvector, so as to conduct a consistency test to obtain the weight of different evaluation indices (Yang et al., 2017). However, if the AHP method is used alone, the weight cannot be reasonable due to the large subjective component and certain persuasiveness. Still, the AHP method can be applied to determine the threshold weight in WSNs, and the steps are described as follows:

(1) Construction of the structural model of the cluster head node selection. The hierarchical model of the threshold weight in Region 1 ($d_i \le d_0$) and Region 2 ($d_i > d_0$) are depicted in **Figures 2A,B**.

(2) Determination of the evaluation index matrix. The residual energy, distribution density, and distance of the nodes are taken as evaluation indices, which are composed of the evaluation index matrix C, written as,

$$\boldsymbol{C}=\begin{bmatrix} \boldsymbol{1} & \boldsymbol{c_{12}} & \boldsymbol{c_{13}} \\ \boldsymbol{c_{21}} & \boldsymbol{1} & \boldsymbol{c_{23}} \\ \boldsymbol{c_{31}} & \boldsymbol{c_{32}} & \boldsymbol{1} \end{bmatrix} \tag{4}$$

where c_{ij} is the measurement value of the evaluation index, and $c_{ji} = \frac{1}{c_{ij}}$. The value of the c_{ij} usually adopts Santy's 1–9 scale method as listed in **Table 2**.

According to the importance of the impact factors in different regions, evaluation index matrices corresponding to influencing factors can be obtained from different hierarchical models, as shown in Eq. (5).

$$\boldsymbol{C}=\begin{bmatrix} \boldsymbol{1} & \boldsymbol{2} & \boldsymbol{5} \\ \boldsymbol{\frac{1}{2}} & \boldsymbol{1} & \boldsymbol{3} \\ \boldsymbol{\frac{1}{5}} & \boldsymbol{\frac{1}{3}} & \boldsymbol{1} \end{bmatrix} \tag{5}$$

(3) The calculation of the AHP weight. According to Eq. (6), the maximum eigenvalue and eigenvector of the evaluation index matrix can be obtained, $\lambda_{\max}$ and W, respectively, where the corresponding eigenvector is $W = [W_1\ W_2\ W_3]^T$, and the obtained normalized weight is written in Eq. (7). The eigenvalue of Eq. (5) is 3.0037, and the weight of the energy, density, and distance are 0.581, 0.309, and 0.109 respectively.

$$\boldsymbol{CW}=\lambda_{\max} \tag{6}$$

$$\omega_{\boldsymbol{Ai}}=\frac{\boldsymbol{W_i}}{\sum_{j}^{3}\boldsymbol{W_j}} \quad (\boldsymbol{i = 1, 2, 3}) \tag{7}$$

TABLE 2 | Evaluation index matrix value measurement table.

Scale values	*Meaning*
1	The two evaluation indicators are of the same importance
3	The previous evaluation index is more important
5	The previous evaluation index is important
7	The previous evaluation index is very important
9	The previous evaluation index is extremely important
2,4,6,8	The importance is the median value of the above adjacent indicators
Reciprocal	If the evaluation index i and j are measured as c_{ij},but$c_{ji} = \frac{1}{c_{ij}}$

(4) Consistency test. When the weights of the different impact factors are obtained, the final consistency test can be carried out to ensure the effectiveness of the obtained weights. The test expressions are expressed in Eqs. (8), (9), and (10). The consistency index of the weight values of the tested regions 1 and 2 is 0.0018, and the consistency ratio is 0.0032.

$$\boldsymbol{CR}=\frac{\boldsymbol{CI}}{\boldsymbol{RI}} \tag{8}$$

$$\boldsymbol{CI}=\frac{\lambda_{\max}}{\boldsymbol{n-1}} \tag{9}$$

$$\boldsymbol{RI}=\frac{\lambda'_{\max}}{\boldsymbol{n-1}} \tag{10}$$

where, *RI* is the average consistency indicator, and *CR* is the consistency ratio. If *C* meets the consistency test, $CR < 0.1$. *CI* is a consistency indicator and $CI = 0$ with a high degree of consistency *C*.

Weight of Entropy Method

The entropy method is an objective weighting method without considering the correlation among factors. Here, the entropy method is applied to determine the threshold weight in WSNs.

Assuming WSNs containing n ($n = 1,2,\ldots, N$) sensor nodes, the impact factor of the *n*th node is f_{nt} t ($t = 1,2,\ldots, T$), to affect the threshold of the cluster head node selection.

(1) Analysis of the impact factors. According to Eq. (11), the impact factors are determined as the energy f_{n1}, distance f_{n2}, and relative density f_{n3} of the nth nodes,

$$f_{n1}=\frac{\boldsymbol{E_i}}{\boldsymbol{E_0}} \tag{11}$$

$$f_{n2}=\frac{\boldsymbol{dis}_{\max}}{\boldsymbol{dmin}_{\max}} \tag{12}$$

$$f_{n3}=\frac{\boldsymbol{l_{alive}}}{\boldsymbol{N_{alive}}} \tag{13}$$

(2) The entropy method for the weight of impact factors determination.

(a) The above three impact indicators are normalized as,

$$\boldsymbol{F'_{nt}}=\frac{\boldsymbol{f_{nt}}-\min(\boldsymbol{f_{nt}})}{\max(\boldsymbol{f_{nt}})-\min(\boldsymbol{f_{nt}})} \tag{14}$$

where $\max(f_{nt})$ and $\min(f_{nt})$ are the maximum and minimum of the impact factors, respectively. The evaluation matrix *R* of the normalized impact factor is obtained as,

$$\boldsymbol{R}=\begin{bmatrix} \boldsymbol{F'_{11}} & \boldsymbol{F'_{12}} & \boldsymbol{F'_{13}} \\ \boldsymbol{F'_{21}} & \boldsymbol{F'_{22}} & \boldsymbol{F'_{23}} \\ \cdots & \cdots & \cdots \\ \boldsymbol{F'_{N1}} & \boldsymbol{F'_{N2}} & \boldsymbol{F'_{N3}} \end{bmatrix} \tag{15}$$

(b) The weight of the *t*th impact factor, E_t is expressed as,

$$\boldsymbol{E_t}=\frac{\boldsymbol{1}}{\ln \boldsymbol{N}}\sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} \boldsymbol{P_{nt}} \ln \boldsymbol{P_{nt}} \tag{16}$$

$$\boldsymbol{P_{nt}}=\frac{\boldsymbol{F'_{nt}}}{\sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} \boldsymbol{F'_{nt}}} \tag{17}$$

where P_{nt} is the proportion of the t impact factor of the *n*th node under all indicators of the node.

(c) The entropy value (weight) of each impact factor is written as,

$$\omega_{\boldsymbol{St}}=\frac{\boldsymbol{1-E_n}}{\boldsymbol{T}-\sum_{\boldsymbol{t=l}}^{\boldsymbol{T}} \boldsymbol{E_n}} \quad 0<\omega_{\boldsymbol{S}}<1 \tag{18}$$

Optimal Combined Weighting

The OCW is a method where both the quantitative and qualitative analyses are used to reasonably allocate the weight, and how to allocate the weight of the AHP and entropy is the key of this method (Yang et al., 2017). The weight calculation of the OCW method can be expressed as,

$$\omega_{\boldsymbol{t}}=\lambda_{\boldsymbol{1}}\omega_{\boldsymbol{Ai}}+\lambda_{\boldsymbol{2}}\omega_{\boldsymbol{St}} \tag{19}$$

where ω_t, ω_{Ai}, and ω_{St} are the weights of the optimal combination, AHP, and entropy method, respectively. λ_1 and λ_2 are the importance degrees of the AHP and entropy methods, respectively.

A Bunch of Rules

Once the cluster head nodes are determined, they are broadcast inside the WSNs so that the rest sensor nodes are invited to join specific clusters. If a sensor node receives multiple invitation messages within a period of time, it will determine the distance between the cluster head nodes and itself according to the strength of the received information, then make the decision to join the closer cluster and transmit the requested information to the specific cluster head node. The cluster head node receives the distributed information from each member node and decides which nodes can be joined. Then, the clusters can be established, and it enters the data transmission stage.

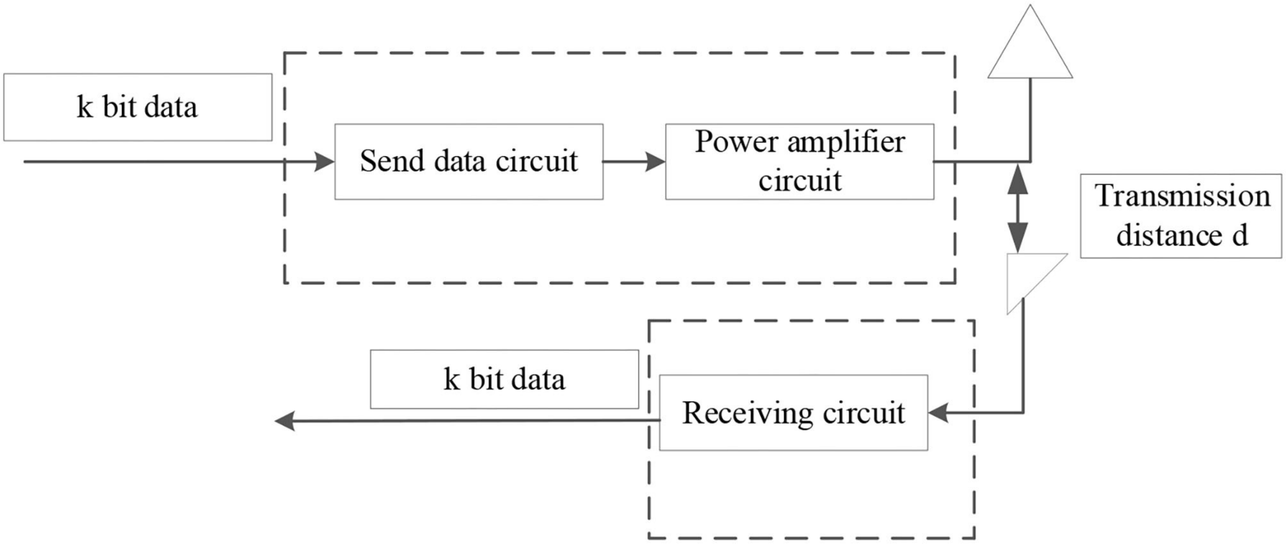


FIGURE 3 | Wireless communication model of sensor nodes.

ENERGY CONSUMPTION ANALYSIS OF THE SENSOR NODE

The differences in energy consumption with different routing protocols can be used to evaluate the transceiver characteristics of the sensor nodes. Hence, a typical radio energy consumption model is usually adopted, including the energy consumption of the data transmitting circuit and the power amplifier circuit (Yang et al., 2017), as demonstrated in **Figure 3**.

There are mainly two energy consumption models for the WSNs, i.e., free-space models and multipath fading models, the adoption of which depends on the distance between the sender and the receiver (Li et al., 2020) and written as,

$$\boldsymbol{E_{TX}\,(l,d) = E_{TX-elec}\,(l) + E_{TX-amp}\,(l,d) = \begin{cases} -l^{*}E_{elec} + l^{*}\varepsilon_{fs}{}^{*}d^{2},\ if & d \le d_0 \\ -l^{*}E_{elec} + l^{*}\varepsilon_{mp}{}^{*}d^{4},\ if & d > d_0 \end{cases}} \tag{20}$$

where $E_{TX-elec}(l)$ is the loss of the transmitting circuit energy, $E_{TX-amp}(l,d)$ is the loss of the amplification circuit energy, and E_{elec} is the energy loss of the sender/receiver when performing data transmission. The calculation of the threshold of the data transmission distance is written as,

$$\boldsymbol{d_0 = \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}}} \tag{21}$$

where ε_{fs} and ε_{mp} represent the amplification energy consumption parameters of the free-space model and the multipath fading model, respectively. The type of the transmitter amplifier determines the values of the two parameters.

OPTIMIZATION OF THE TRANSMISSION PATH BETWEEN CLUSTERS BASED ON IACO

Flow Pattern Division

According to the energy consumption model, the energy consumption of the nodes takes the distance threshold d_0 as the intermediate value, so the regions can be divided by the data transmission distance threshold d_0 in the LEACH protocol optimization. The range ($d_{\min} \le d_i \le d_0$) within the distance from the sink node d_0 is area 1, and the range within the range of WSN nodes ($d_0 \le d_i \le d_{\max}$) is area 2. The partitioning diagram is illustrated in **Figure 4**.

In WSNs, the node positions are usually fixed after deployment. When $d_i \le d_0$, the distance between nodes will not be regarded as the main impact factor, so the weight in the threshold will change. In contrast, when $d_0 \le d_i$, the energy consumption of the node transmission and the distance between nodes increase to the fourth power. Hence, distance is the main impact factor. The weight in Eq. (19) will change with the importance of the impact factor.

Transmission Path Optimization Among Clusters Based on the IACO Algorithm

The ACO algorithm is a heuristic optimization method, which can be used to search the optimal path through individual efforts and group cooperation *via* accumulated pheromones on the path with positive feedback (Song and Yao, 2017; Li et al., 2020). In WSNs, the optimization of the transmission data path of the routing protocol has the same characteristics as the ACO algorithm to obtain the best foraging route for ants. The application of the ACO algorithm in WSNs still has some limitations, including low path efficiency and the appearance of local optimal solution. Here, the developed IACO algorithm

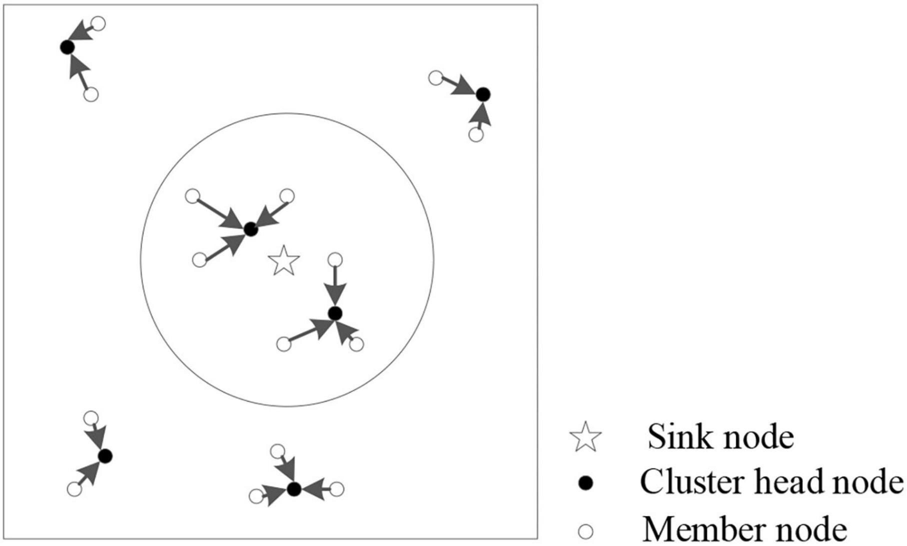


FIGURE 4 | Schematic diagram of the zoning of wireless sensor network deployment node range.

is applied to WSNs to conduct path optimization from three aspects, improved transition probability, path superiority, and pheromone updating mechanism.

Transfer Probability Improvement

The transition probability of the ACO algorithm considers only the path pheromone concentration and the distance between two nodes. The ant in the process of the optimal path search can also increase the probability of invalid path search and reduce the efficiency of path construction. Because the impact factors are not comprehensive, the flow of the data transmission of the nodes is larger, which could result in premature death affecting the network operation.

By using the ACO algorithm in WSN, the distance between two nodes is only considered in the next hop of the cluster node selection, which could cause more candidate nodes and a large amount of data transmission accompanied by more energy consumption. In this article, the transition probability is fully considered with the node energy for the ants' node search ability enhancement, so as to speed up the convergence speed, and the developed transition probability function is formatted as,

$$P_{ij}^{k}(t) = \begin{cases} \dfrac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}(E_{ja})^{\gamma}}{\sum_{s\in allowed_k}(\tau_{is})^{\alpha}(\eta_{js})^{\beta}(E_{ja})^{\gamma}}, & s \in allowed_k \\ 0, & s \notin allowed_k \end{cases} \tag{22}$$

$$E_{ja} = \frac{E_j}{E_{iave}} \tag{23}$$

where $allowed_k$ is the cluster head node set where k nodes have not been reached, E_j is the remaining energy of the next hop node, and E_{iave} is the average energy of the adjacent nodes of node i. τ_{ij} is the pheromone concentration of the path from node i to node j, and η_{ij} is the heuristic function to be defined for the cluster head node

Based on the forwarding distance j node and the distance of the aggregation node, written as,

$$\eta_{ij} = \frac{1}{d_{ij} + d_{js}} \tag{24}$$

where d_{ij} is the Euclidean distance between node i and the next hop node j, and d_{js} is the Euclidean distance between the next hop node j and the sink node.

Path Superiority Determination

When all ants are transferred to the sink node, each ant corresponds to a transmission path, so the path superiority degree can be used as the standard to measure the optimal path. The ant with a higher superiority degree is the optimal transmission path. In the previous path optimization process, the path with higher average energy is highly likely to be the best transmission path; however, the mean energy cannot represent the current node energy level. For instance, if certain nodes have high energy mean, but the energy difference between the actual nodes is substantially large, it will cause premature death of the nodes and failure of the transmission path. The path superiority is thus reflected from the lower hop count, higher mean energy, and uniform energy distribution. The higher the path superiority, the higher quality of the selected transmission path.

The variation coefficient is defined by the ratio between the SD and mean of the data, which is used to evaluate the difference degree of the data distribution. The smaller the coefficient of the variation, the more uniform the data distribution, and the smaller difference between the data (Zhu, 2017). Therefore, the path variation coefficient P_{cv} is used to analyze the energy balance of the nodes.

$$P_{cv} = \frac{P_{sd}}{P_m} = \frac{\sqrt{\frac{1}{\delta}\sum_{i=1}^{\delta}(E(i) - E_{ave})^2}}{E_{ave}} \tag{25}$$

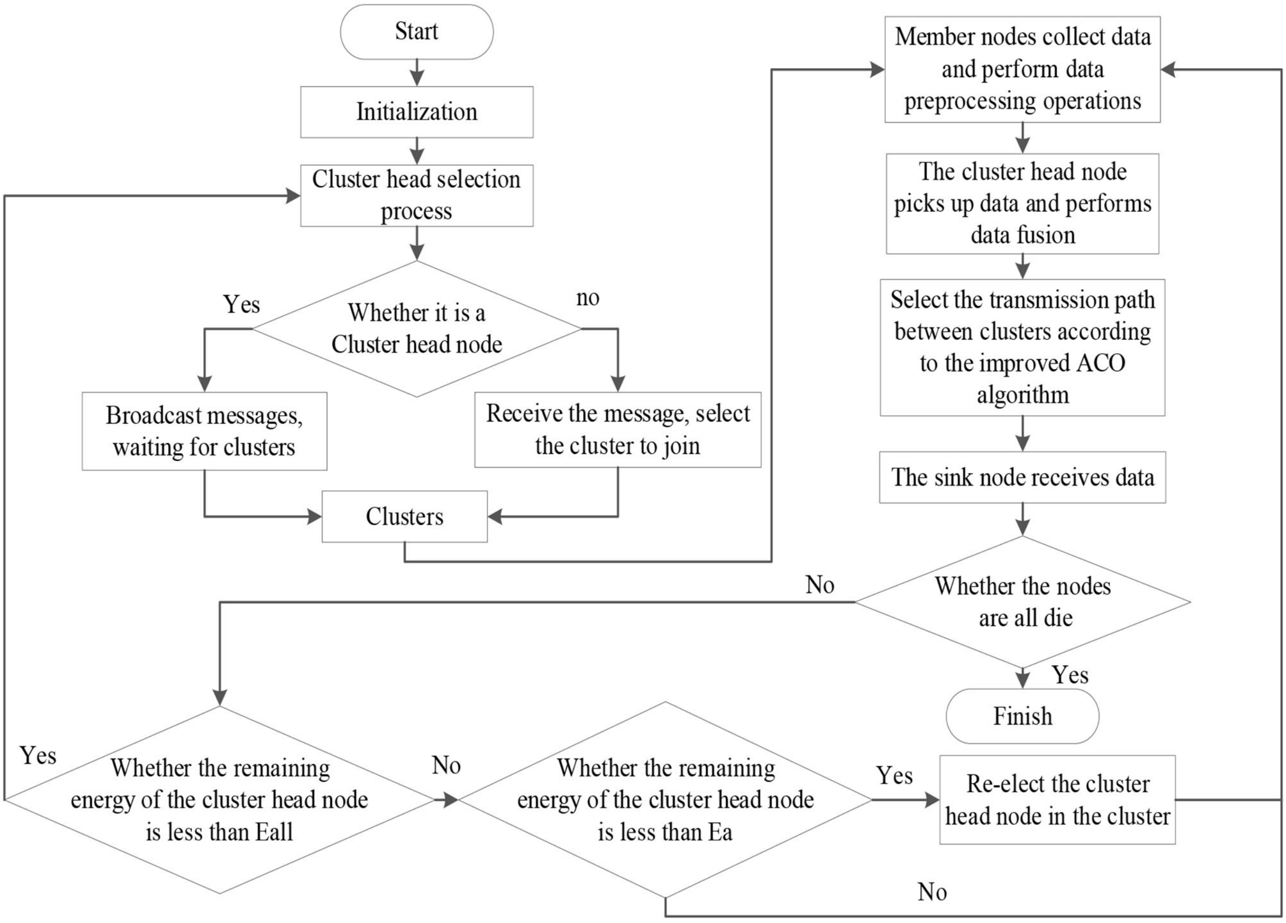

Start
Initialization
Cluster head selection process
Whether it is a Cluster head node
Yes
no
Broadcast messages, waiting for clusters
Receive the message, select the cluster to join
Clusters
Member nodes collect data and perform data preprocessing operations
The cluster head node picks up data and performs data fusion
Select the transmission path between clusters according to the improved ACO algorithm
The sink node receives data
Whether the nodes are all die
No
Yes
Finish
Yes
Whether the remaining energy of the cluster head node is less than Eall
No
Whether the remaining energy of the cluster head node is less than Ea
Yes
Re-elect the cluster head node in the cluster
No

FIGURE 5 | Improved LEACH protocol flowchart.

where P_{sd} is the SD of the energy of all nodes in the path P_m, E_{ave} is the mean energy of all nodes in the path, and δ is the number of nodes contained in the path.

Hence, the path superiority P_s can be written as,

$$\boldsymbol{P_s}=\frac{\boldsymbol{E}_{\min}}{\boldsymbol{E_{con}}\frac{\mathbf{1}}{\boldsymbol{P_{cv}}}+\frac{\mathbf{1}}{\boldsymbol{J}}} \tag{26}$$

where $E_{\min}$ is the minimum of the node energy, E_{con} is the sum of the node energy consumption, and J is the total hop number in the path.

Pheromone Updating

In the ACO algorithm, the pheromone concentration varies according to the length of the path, and ants plan the next route according to the pheromone concentration of different paths. Ants tend to choose paths with higher pheromone concentrations, and other paths are ignored. As a result, the path searching will fall into the local optimization, the ant will no longer search for new paths and the path searching process stops. Therefore, the IACO algorithm adds local and global pheromone updates to inter-cluster transmission path planning to avoid such local optimum problems. The developed pheromone updating mechanism is written as,

$$\tau_{\boldsymbol{ij}}(\boldsymbol{t}+\mathbf{1})=(\mathbf{1}-\rho)\tau_{\boldsymbol{ij}}(\boldsymbol{t})+\rho\Delta\tau_{\boldsymbol{ij}}(\boldsymbol{t}) \tag{27}$$

$$\Delta\tau_{\boldsymbol{ij}}(\boldsymbol{t})=\sum_{\boldsymbol{k}=\mathbf{1}}^{\boldsymbol{N}}\Delta\tau_{\boldsymbol{ij}}^{\boldsymbol{k}}(\boldsymbol{t}) \tag{28}$$

where ρ is the parameter regulating the pheromone volatilization speed, $\Delta\tau_{ij}(t)$ is the pheromone increment in the path, $\Delta\tau_{ij}(0)=0$ at the initial time, and $\Delta\tau_{ij}^{k}(t)$ is the pheromone concentration left by the kth ant in the path (i, j).

- Local pheromone update: If the ant node carries out data forwarding from the cluster head node $i \rightarrow j$, the pheromone concentrations of the corresponding paths should be locally updated,

$$\Delta\tau_{\boldsymbol{ij}}^{\boldsymbol{k}}(\boldsymbol{t})=\frac{\boldsymbol{E_r}(\boldsymbol{j})}{\boldsymbol{d}(\boldsymbol{i},\boldsymbol{j})} \tag{29}$$

where $E_r(j)$ is the number of the member nodes in the next hop node.

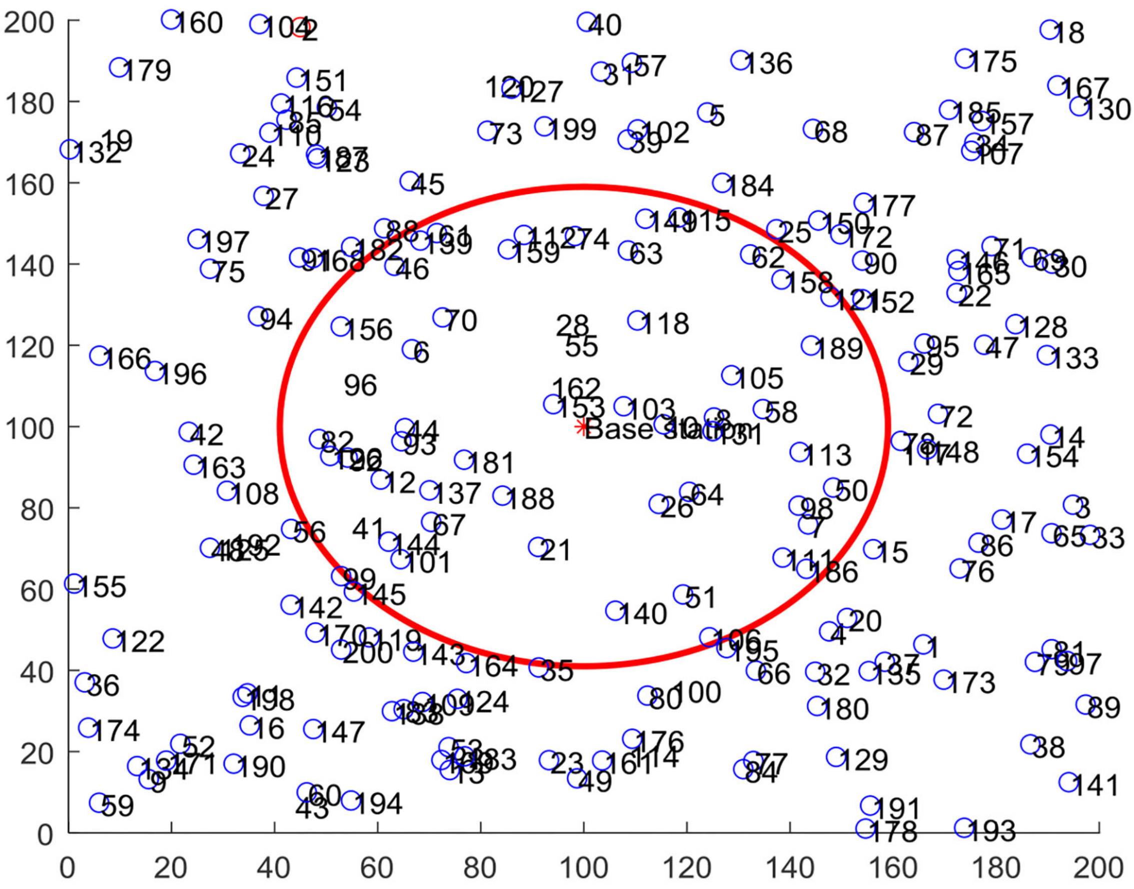


FIGURE 6 | Sensor node deployment diagram.

- Global pheromone update: When all ants move to the sink node, each ant corresponds to a transmission path, and the pheromone concentration of the path is updated globally. Based on the path superiority, the global update rules is written as,

$$\Delta\tau_{ij}^{k}(t) = \frac{1}{P_s} \tag{30}$$

The LEACH protocol is optimized with the improved clustering algorithm to cluster sensor nodes, and the influence factors of nodes in different network regions are different so as to replace the cluster head nodes dynamically. When the energy of the cluster head node reaches the limit value, the cluster is updated and the data are fused in the cluster head node. Although the IACO algorithm is used to find the optimal transmission path between clusters, it can effectively avoid the stagnation of the IACO algorithm in the local optimum. The specific process is depicted in **Figure 5**.

EXPERIMENTAL VERIFICATION AND ANALYSIS

Experimental Environment Setting

The experiment verification is carried out in a MATLAB simulation environment, and 200 sensor nodes are deployed in a 200 × 200 m network area. The node deployment within the network is shown in **Figure 6**, where the circle with sink node is the center, the area with the threshold distance *R* as the radius of the circle is area 1, and the rest is area 2. The parameters of the basic network and IACO algorithm in the experiments are listed in **Tables 3**, **4**.

Performance Indicators

The life cycle of the network, the total energy consumption of the network, and the data received by the sink node are taken as indicators to evaluate the quality of the routing protocol. The specific analysis is as follows:

1. Network life cycle: the duration from the normal operation of the WSN after the successful layout to the death of the last node. Three indicators were selected for evaluation, namely, the number of rounds in which the first node died (indicator

TABLE 3 | Basic parameter settings of the network.

Parameter	Parameter value
Network area	200 × 200 m
Number of nodes	200 individuals
The location of the sink node	(100, 100)
Node's initial energy E_0	0.2J
Energy consumption at the receiving end E_{RX}	50 nJ/bit*m^2
Energy consumption of the sender E_{TX}	50 nJ/bit
Energy consumption of data fusion E_{DA}	5 nJ/bit
Free space model energy consumption parameters ε_{fs}	10 pJ/bit*m^2
Amplified energy consumption parameters of the multipath model ε_{mp}	0.0013 pJ/bit*m^2
Control packet size	100 bits
Packet size	3,000 bits

TABLE 4 | Basic parameter setting of IACO algorithm.

Parameter	Parameter value
Pheromone concentration weight factor α	1
Heuristic function weight factor β	5
Node energy consumption weight factor λ	4
Pheromone Volatilization Coefficient ρ	0.1

1), the number of rounds in which 10% of nodes died (indicator 2), and the number of rounds in which all nodes died (indicator 3). In most cases, the appearance of the dead nodes leads to the deterioration of the network detection quality, so the number of rounds where the first node dies is of particular importance.

2. Total network energy consumption: the total energy consumption of all nodes in the network during implementation. This index can reflect the balance degree of energy consumption. In the simulation, the size of the packet and control packet is set, and the energy consumptions of different nodes are calculated through the network operation.
3. Data received by the sink node: data received by the sink node after each round of the network operation.

Experimental Results and Analysis

The simulation results of different routing protocols are analyzed based on the above performance indicators to verify the effectiveness of the ILEACH protocol.

Figures 7, **8** compare the number of surviving nodes and total network energy consumption of the LEACH protocol and the ILEACH protocol, respectively. To be specific, the impact of the ILEACH protocol on the network life cycle is analyzed *via* indicators 1 and 2, as shown in **Figure 8**.

Figure 9 indicates that indicator 1 of the LEACH protocol and the ILEACH protocol appears in rounds 141 and 199, respectively, and indicator 2 appears when the network operates to rounds 596 and 986. When the LEACH protocol reaches ∼600 rounds, most nodes have no power, but the ILEACH protocol can be extended to ∼1,200 rounds. The network needs to be initialized in the early stage, which consumes energy quickly, and there are fewer surviving nodes in the later stage. However, ILEACH can effectively balance the energy consumption of nodes by considering the energy consumption of nodes in the process of clustering and data transmission, and the network energy consumption varies slowly in the later period. It is proven that the improved threshold and the replacement mechanism of the cluster head nodes can reduce the energy consumption of the nodes, and the ILEACH protocol can prolong the network life cycle much longer compared with the LEACH protocol.

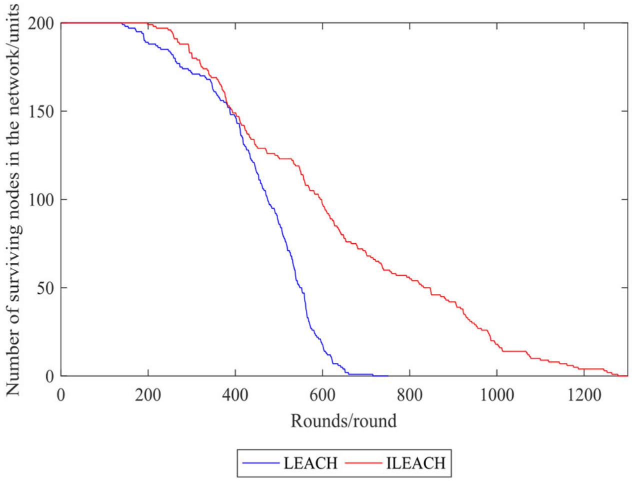


FIGURE 7 | Comparison curve of the number of surviving nodes in the network.

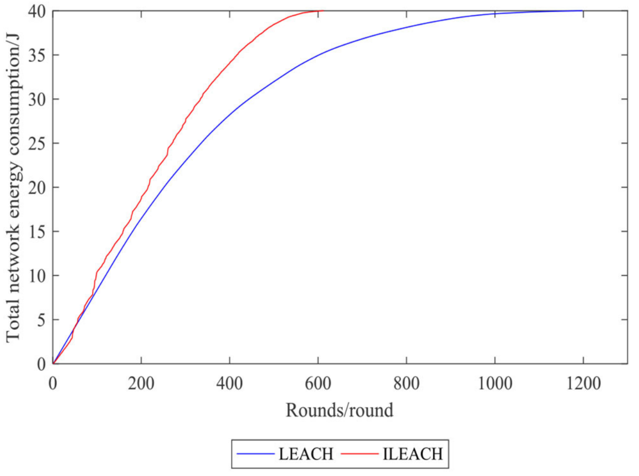


FIGURE 8 | Comparison curve of total network energy consumption.

Figures 10, **11** compare the number of the surviving nodes and total network energy consumption between the LEACH

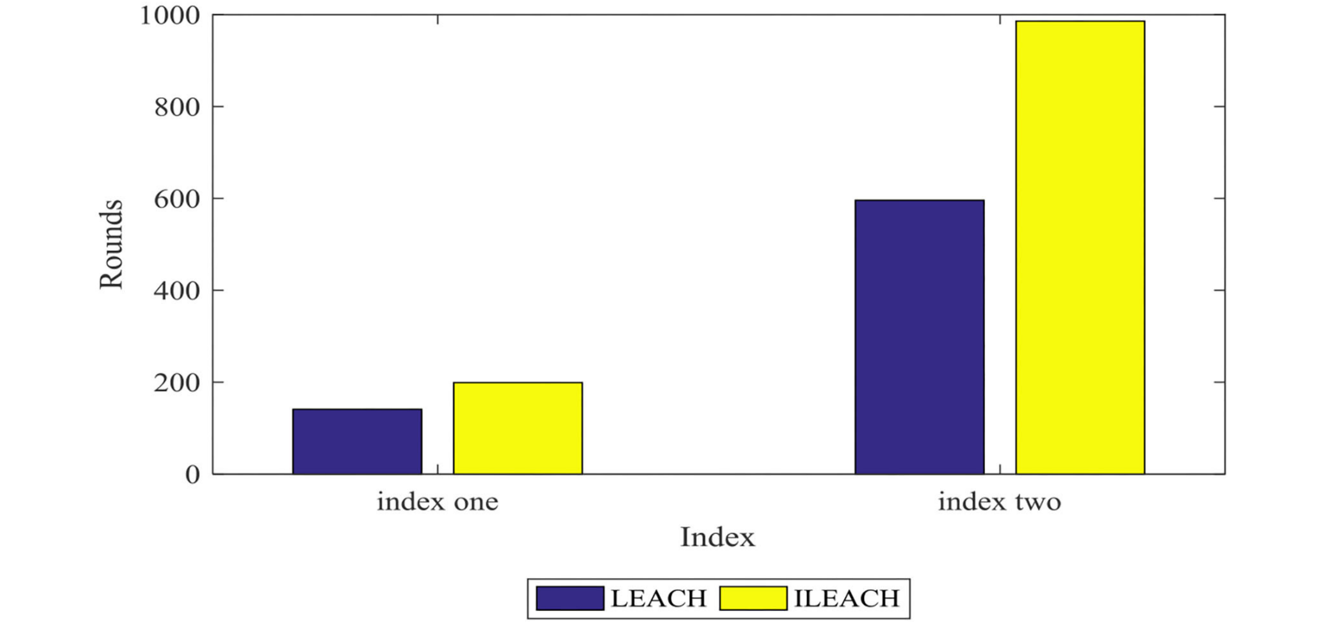


FIGURE 9 | Indicator analysis histogram.

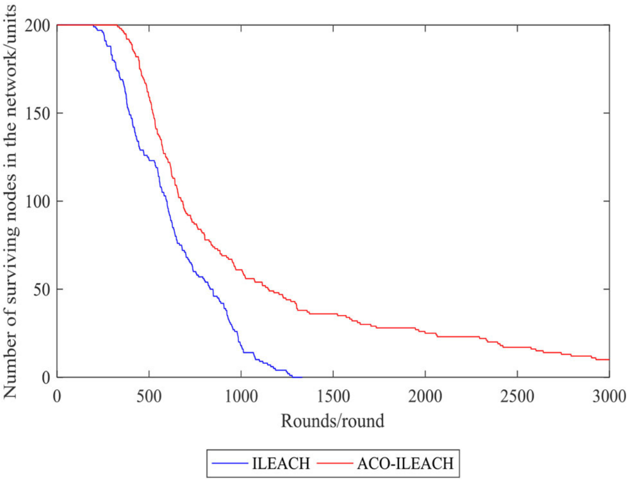


FIGURE 10 | Comparison curve of the number of surviving nodes in the network.

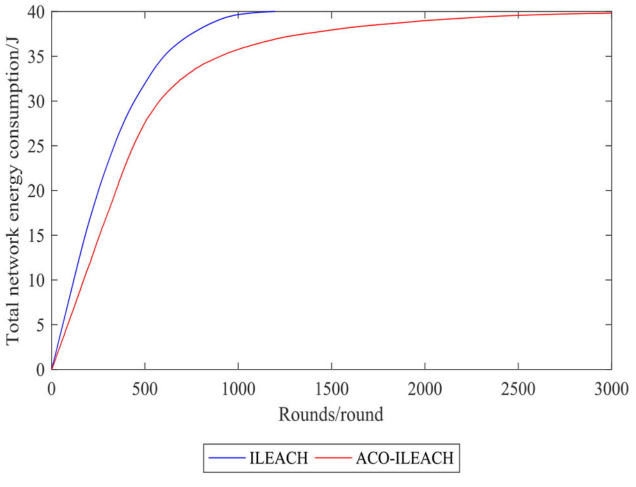


FIGURE 11 | Comparison curve of totalnetwork energy consumption.

of ACO (ACO-LEACH) protocol and the LEACH of IACO (IACO-LEACH) protocol, where **Figure 11** is the histogram of the index. **Figure 12** illustrates that indicator 1 of the ACO-LEACH protocol and IACO-LEACH protocol occurs in rounds 199 and 329, respectively, and indicator 2 occurs in rounds 986 and 2,338. **Table 5** displays the number of the first node deaths, 10% node deaths, and all node deaths under different protocols. In conclusion, the developed IACO algorithm can gradually find the optimal transmission path for the WSN operation and save energy in the data transmission stage effectively.

CONCLUSION

An algorithm based on OCW and IACO is proposed in the article to solve the problem of high energy consumption of the traditional LEACH protocol in WSNs. The ILEACH protocol adopts the cluster head node replacement mechanism to reduce the energy consumption considering the energy, density, and nodes distance for the threshold selection, which can effectively avoid the randomness of the clustering. Furthermore, the OCW is used to dynamically change the weight of the nodes according to the different impact factors of nodes in different regions. The developed IACO algorithm can optimize the transfer probability

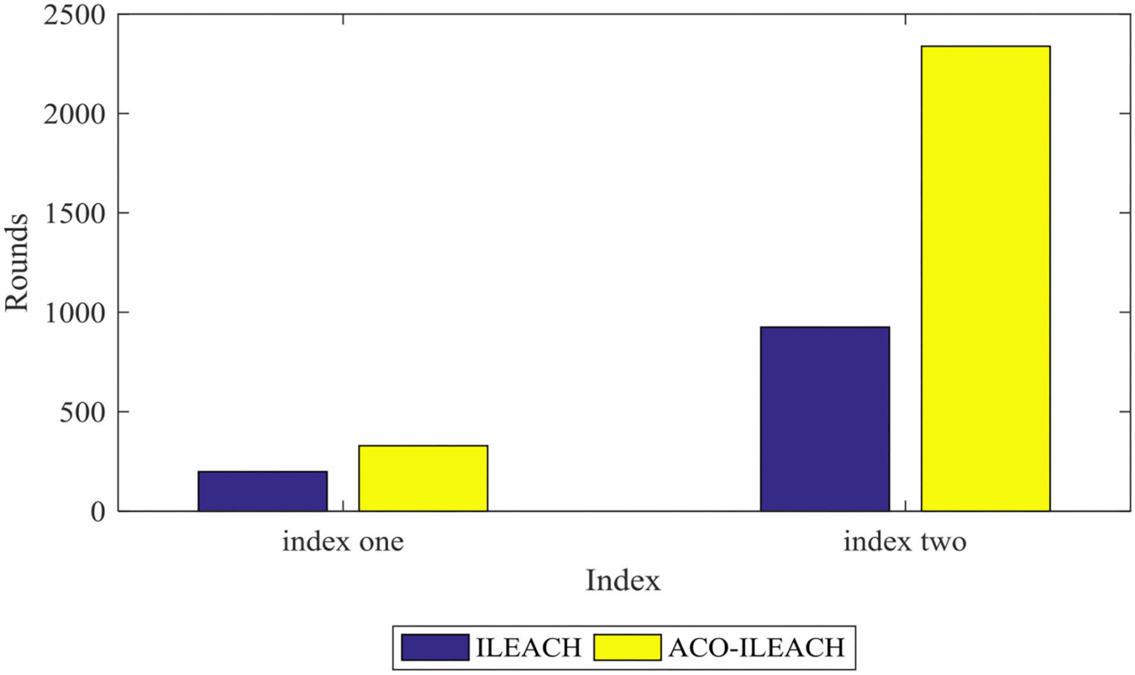


FIGURE 12 | Indicator analysis histogram.

TABLE 5 | Comparison table of the network life cycle of different protocols.

Protocol	Number of rounds in which the first node died	Number of rounds in which 10% of the nodes died	Number of rounds in which all nodes died
LEACH	141	596	659
ILEACH	199	986	1,296
ACO-ILEACH	329	2,338	3,487

of the sensor nodes with the local update and global update strategies, which can prolong the life cycle of the network to a certain extent. The network environment is deployed through MATLAB simulation software to verify the feasibility of the ILEACH protocol.

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

XC and JZ proposed the idea in this study. CX and XL designed the experiment, performed the simulation experiments, analyzed the experiment results, and wrote the manuscript. JL corrected the manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

The project (work) was supported by the National Natural Science Foundation of China Program (No. 62073198) and the Major Research Development Program of Shandong province of China (No. 2016GSF117009).

ACKNOWLEDGMENTS

We acknowledge the support of the College of Electrical Engineering and Automation and the College of Energy and Mining Engineering in Shandong University of Science and Technology and Shandong Senter Electronic Co., Ltd.

REFERENCES

Chen, S. X., He, Z. S., Xiong, H. L., and Liao, J. W. (2015). A wireless sensor signal reconstruction algorithm based on compressed sensing. *Journal of computer science, Papers* 38, 614–624. doi: 10.3724/SP.J.1016.2015.00614

Dai, J. Y., Deng, X. H., Wang, B., and Wang, H. H. (2020). Clustering routing protocol for wireless sensor networks based on improved firefly optimized neural network. *Journal of Beijing University of Posts and telecommunications, Papers* 43, 131–137. doi: 10.13190/j.jbupt.2019-161

Ding, H. (2020). Data fusion algorithm for wireless sensor networks based on ant colony optimization. *Journal of Shenyang University of technology, Papers* 42, 208–212. doi: 10.7688/j.issn.1000-1646.2020.02.16

Efe, K., Ibrahim, K., and Cevdet, A. (2013). Active node determination for correlated data gathering in wireless sensor networks. *Computer Networks* 57, 1124–1138. doi: 10.1016/j.comnet.2012.11.018

Ezhilarasi, M., and Krishnaveni, V. (2019). An evolutionary multipath energy-efficient routing protocol (emeer) for network lifetime enhancement in wireless sensor networks. *Soft Computing A Fusion of Foundations Methodologies and Applications. Papers* 23, 8367–8377. doi: 10.1007/s00500-019-03928-1

Galkin, P. (2018). Model of Reducing the Power Consumption for Node of Wireless Sensor Network in Embedded Control Systems. 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC SandT), pp. 252-256. doi: 10.1109/INFOCOMMST.2018.8631891

Heinzelman, W. R. (2000). Energy-efficient communication protocol for wireless microsensor networks. *Proc. IEEE Annu. Hawaii Int. Conf. Syst. Sci.* 8:8020.

Hou, M. T., Zhao, Z. P., Gao, M., and Zhang, N. N. (2017). Ant colony optimization multipath routing algorithm adopted angle factor. *Computer Engineering and Applications. Papers* 53, 107–112. doi: 10.3778/j.issn.1002-8331.1604-0176

Huang, Y., and Hua, Y. Q. (2020). Routing optimization of wireless sensor networks with energy and path constraints. *Journal of Xi'an University of Electronic Science and technology, Papers* 47, 113–120. doi: 10.19665/j.issn1001-2400.2020.03.016

Jameii, S. M., and Maadani, M. (2016). Intelligent dynamic Connectivity Control Algorithm for cluster-based wireless sensor networks. 2016 11th International Conference for Internet Technology and Secured Transactions (ICITST), pp. 416-420, doi: 10.1109/ICITST.2016.7856744

Lalwani, P., Das, S., Banka, H., and Kumar, C. (2018). Crhs: clustering and routing in wireless sensor networks using harmony search algorithm. *Neural Computing and Applications* 30, 1–21. doi: 10.1007/s00521-016-2662-4

Li, X. Q., Ma, R., Zhang, S., Hou, Y. Z., and Pei, Y. F. (2020). Improved design of Ant Colony Optimization and its application in Track Planning. *Journal of Aeronautics, Papers* 41, 213–219. doi: 10.7527/S1000-6893.2020.24381

Liu, H., and Li, H. W. (2018). Non uniform clustering routing algorithm based on Ant Colony Optimization," Journal of Huazhong University of science and Technology (Natural Science Edition), Papers 46(08), 50–54. doi: 10.13245/j.hust.180810

Lv, L., Peng, F., Chen, W., and Wang, J. (2021). Node Deployment of Wireless Sensor Networks Based on MOEA/P Algorithm. 2021 13th International Conference on Communication Software and Networks (ICCSN). pp. 173-178. doi: 10.1109/ICCSN52437.2021.9463630

Maheshwari, P., Sharma, A. K., and Verma, K. (2021). Energy efficient cluster based routing protocol for wsn using butterfly optimization algorithm and ant colony optimization. *Ad Hoc Networks* 110, 102317. doi: 10.1016/j.adhoc.2020.102317

Marappan, P., and Rodrigues, P. (2016). An energy efficient routing protocol for correlated data using CLLEACH,nd in WSNs. *Wireless Networks*. 22(04): 1415–1423. doi: 10.1007/s11276-015-1063-4

Mohar, S. S., Goyal, S., and Kaur, R. (2020). Optimized sensor nodes deployment in wireless sensor network using bat algorithm. *Wireless Personal Communications(*2), 1–19. doi: 10.1007/s11277-020-07823-z

Mukherjee, M., Lu, Y. R., Matam, R., and Choudhury, N. (2018). Energy trading in sleep scheduling for wireless rechargeable WSNs, 2018 10th International Conference on Communication Systems and Networks (COMSNETS). *pp.* 429–431. doi: 10.1109/COMSNETS.2018.8328235

Nayyar, A., and Singh, R. (2020). Ieemarp- a novel energy efficient multipath routing protocol based on ant colony optimization (aco) for dynamic sensor networks. *Multimedia Tools and Applications, Papers* 79, 35221–35252. doi: 10.1007/s11042-019-7627-z

Ning, W., Yuan, Z., and Jinghao, L. (2017). A hierarchical clustering protocol for wireless sensor Networks based on improved particle swarm Optimization. *Chinese Journal of Sensing Technology* 30, 120–125. doi: 10.3969/j.issn.1004-1699.2017.01.022

Rahman, A. U., Hasbullah, H., and Sama, N. U. (2013). Efficient energy utilization through optimum number of sensor node distribution in engineered corona-based (onsd-ec) wireless sensor network. *Wireless Personal Communications* 73, 1227–1243. doi: 10.1007/s11277-013-1275-9

Sirdeshpande, N., and Udupi, V. (2017). Fractional lion optimization for cluster head-based routing protocol in wireless sensor network. *Journal of the Franklin Institute* 354, 4457–4480. doi: 10.1016/j.jfranklin.2017.04.005

Song, Y., and Yao, X. (2017). Design of Routing Protocol and Node Structure in Wireless Sensor Network Based on Improved Ant Colony Optimization Algorithm. *2017 International Conference on Computer Network, Electronic and Automation (ICCNEA)* 2017, pp. 236–240. doi: 10.1109/ICCNEA.2017.54

Tripathi, R. K., Singh, Y. N., and Verma, N. K. (2013). Clustering algorithm for non-uniformly distributed nodes in wireless sensor network. *Electronics Letters* 49, 299–300. doi: 10.1049/el.2012.3512

Wang, C., Wang, Z., Zhu, Y., and Han, Z. (2018). The Application of Data-Level Fusion Algorithm Based on Adaptive-Weighted and Support Degree in Intelligent Household Greenhouse. In: Zhu Q., Na J., Wu X. (eds) Innovative Techniques and Applications of Modelling, Identification and Control. Lecture Notes in Electrical Engineering, vol 467. Springer, Singapore. doi: 10.1007/978-981-10-7212-3_6

Yan, Z., Goswami, P., Mukherjee, A., Yang, L., Routray, S., and Palai, G. (2018). Low-energy pso-based node positioning in optical wireless sensor networks. *Optik - International Journal for Light and Electron Optics. Papers* 181, 378–382. doi: 10.1016/j.ijleo.2018.12.055

Yang, M., Hu, W., and Wang, Y. S. (2017). Risk assessment of debris flow in the lower reaches of Zagunao River Based on Optimal Combination Weighting and fuzzy comprehensive evaluation method. *Journal of natural science of Heilongjiang University, Papers* 34, 271–278. doi: 10.13482/j.issn1001-7011.2016.07.230

Zhu, Z. C. (2017). Hybrid cross weighting method based on coefficient of variation G1 method. *Statistics and decision making, Papers* 12, 78–81. doi: 10.13546/j.cnki.tjyjc.2017.12.018

Zou, Z. F., and Qian, Y. (2019). Wireless sensor network routing method based on improved ant colony algorithm. *J Ambient Intell Human Comput* 10, 991–998. doi: 10.1007/s12652-018-0751-1

Conflict of Interest: XL is employed by Shandong Senter Electronic Co., Ltd.

The remaining 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.

frontiers
in Neurorobotics

ORIGINAL RESEARCH
published: 24 March 2022
doi: 10.3389/fnbot.2022.806898

Learning Suction Graspability Considering Grasp Quality and Robot Reachability for Bin-Picking

Ping Jiang, Junji Oaki, Yoshiyuki Ishihara, Junichiro Ooga, Haifeng Han, Atsushi Sugahara, Seiji Tokura, Haruna Eto, Kazuma Komoda and Akihito Ogawa*

Corporate Research & Development Center, Toshiba Corporation, Kawasaki, Japan

Deep learning has been widely used for inferring robust grasps. Although human-labeled RGB-D datasets were initially used to learn grasp configurations, preparation of this kind of large dataset is expensive. To address this problem, images were generated by a physical simulator, and a physically inspired model (e.g., a contact model between a suction vacuum cup and object) was used as a grasp quality evaluation metric to annotate the synthesized images. However, this kind of contact model is complicated and requires parameter identification by experiments to ensure real world performance. In addition, previous studies have not considered manipulator reachability such as when a grasp configuration with high grasp quality is unable to reach the target due to collisions or the physical limitations of the robot. In this study, we propose an intuitive geometric analytic-based grasp quality evaluation metric. We further incorporate a reachability evaluation metric. We annotate the pixel-wise grasp quality and reachability by the proposed evaluation metric on synthesized images in a simulator to train an auto-encoder–decoder called suction graspability U-Net++ (SG-U-Net++). Experiment results show that our intuitive grasp quality evaluation metric is competitive with a physically-inspired metric. Learning the reachability helps to reduce motion planning computation time by removing obviously unreachable candidates. The system achieves an overall picking speed of 560 PPH (pieces per hour).

Keywords: bin picking, grasp planning, suction grasp, graspability, deep learning

OPEN ACCESS

Edited by:
Yimin Zhou,
Shenzhen Institutes of Advanced Technology (CAS), China

Reviewed by:
Zhifeng Huang,
Guangdong University of Technology, China
Miao Li,
Swiss Federal Institute of Technology Lausanne, Switzerland

****Correspondence:***
Ping Jiang
ping2.jiang@toshiba.co.jp

Received: *01 November 2021*
Accepted: *23 February 2022*
Published: *24 March 2022*

Citation:
Jiang P, Oaki J, Ishihara Y, Ooga J, Han H, Sugahara A, Tokura S, Eto H, Komoda K and Ogawa A (2022) Learning Suction Graspability Considering Grasp Quality and Robot Reachability for Bin-Picking. Front. Neurorobot. 16:806898. doi: 10.3389/fnbot.2022.806898

1. INTRODUCTION

In recent years, growth in retail e-commerce (electronic-commerce) business has led to high demand for warehouse automation by robots (Bogue, 2016). Although the Amazon picking challenge (Fujita et al., 2020) has advanced the automation of the pick-and-place task, which is a common task in warehouses, picking objects from a cluttered scene remains a challenge.

The key to the automation of pick-and-place is to find the grasp point where the robot can approach *via* a collision free path and then stably grasp the target object. Grasp point detection methods can be broadly divided into analytical and data-driven methods. Analytical methods (Miller and Allen, 2004; Pharswan et al., 2019) require modeling the interaction between the object and the hand and have a high computation cost (Roa and Suárez, 2015). For those reasons, data-driven methods are preferred for bin picking.

Many previous studies have used supervised deep learning, which is one of the most widely used data-driven methods, to predict only grasp point configuration (e.g., location, orientation, and open width) without considering the grasp quality. Given an RGB-D image, the grasp configuration for a jaw gripper (Kumra and Kanan, 2017; Chu et al., 2018; Zhang et al., 2019) or a vacuum gripper (Araki et al., 2020; Jiang et al., 2020) can be directly predicted using a deep convolutional neural network (DCNN). Learning was extended from points to regions by Domae et al. (2014) and Mano et al. (2019), who proposed a convolution-based method in which the hand shape mask is convolved with the depth mask to obtain the region of the grasp points. Matsumura et al. (2019) later learned the peak among all regions for different hand orientations to detect a grasp point capable of avoiding multiple objects.

However, in addition to the grasp configuration, the grasp quality is also important for a robot to select the optimal grasp point for bin picking. The grasp quality indicates the graspable probability by considering factors such as surface properties. For example, for suction grasping, although an object with a complicated shape may have multiple grasp points, the grasp points located on flat surfaces need to be given a higher selection priority because they have higher grasp quality (easier for suction by vacuum cup) than do curved surfaces. Zeng et al. (2018b) empirically labeled the grasp quality in the RGB-D images of the Amazon picking challenge object set. They proposed a multi-modal DCNN for learning grasp quality maps (pixel-wise grasp quality corresponding to an RGB-D image) for jaw and vacuum grippers. However, preparing a dataset by manual labeling is time consuming and so the dataset was synthesized in a simulator to reduce the time cost. Dex-Net (Mahler et al., 2018, 2019) evaluated the grasp quality by a physical model and generated a large dataset by simulation. They used the synthesized dataset to train a grasp quality conventional neural network (GQ-CNN) to estimate the success probability of the grasp point. However, defining a precise physical model for the contact between gripper and object is difficult. Furthermore, the parameters of the model needed to be identified experimentally to reproduce the salient kinematics and dynamics features of a real robot hand (e.g., the deformation and suction force of a vacuum cup).

Unlike Dex-Net, this study proposes an intuitive suction grasp quality analytic metric based on point clouds without the need for modeling complicated contact dynamics. Furthermore, we incorporate a robot reachability metric to evaluate the suction graspability from the viewpoint of the manipulator. Previous studies have evaluated grasp quality only in terms of grasp quality for the hand. However, it is possible that although a grasp point has high grasp quality, the manipulator is not able to move to that point. It is also possible for an object to have multiple grasp points with same the level of graspability but varying amounts of time needed for the manipulator to approach due to differences in the goal pose and surrounding collision objects. Bin picking efficiency can therefore be improved by incorporating a reachability evaluation metric. We label suction graspability by the proposed grasp quality and reachability metric and generate a dataset by the physical simulator. An auto-encoder is trained to predict the suction graspability given the depth image input and a graspability clustering and the ranking algorithm is designed to propose the optimal grasp point.

Our primary contributions include (1) Proposal of an intuitive grasp quality evaluation metric without complicated physical modeling. (2) Proposal of a reachability evaluation metric for labeling suction grapability in addition to grasp quality. (3) Performance of a comparison experiment between the proposed intuitive grasp quality evaluation metric and a physically-inspired one (Dex-Net). (4) Performance of an experiment to investigate the effect of learning reachability.

2. RELATED WORKS

2.1. Pixel-Wise Graspability Learning

In early studies, deep neural networks were used to directly predict the candidate grasp configurations without considering the grasp quality (Asif et al., 2018; Zhou X. et al., 2018; Xu et al., 2021). However, since there can be multiple grasp candidates for an object that has a complicated shape or multiple objects in a cluttered scene, learning graspablity is required for the planner to find the optimal grasp among the candidates.

Pixel-wise graspablity learning uses RGB-D or depth-only images to infer the grasp success probability at each pixel. Zeng et al. (2018b) used a manually labeled dataset to train fully convolutional networks (FCNs) for predicting pixel-wise grasp quality (affordance) maps of four pre-defined grasping primitives. Liu et al. (2020) performed active exploration by pushing objects to find good grasp affordable maps predicted by Zeng's FCNs. Recently, Utomo et al. (2021) modified the architecture of Zeng's FCNs to improve the inference precision and speed. Based on Zeng's concept, Hasegawa et al. (2019) incorporated a primitive template matching module, making the system adaptive to changes in grasping primitives. Zeng et al. also applied the concept of pixel-wise affordance learning to other manipulation tasks such as picking by synergistic coordination of push and grasp motions (Zeng et al., 2018a), and picking and throwing (Zeng et al., 2020). However, preparing huge amounts of RGB-D images and manually labeling the grasp quality requires a large amount of effort.

Faced with the dataset generation cost of RGB-D based graspability learning, researchers started to use depth-image-only based learning. The merits of using depth images are that they are easier to synthesize and annotate in a physical simulator compared with RGB images. Morrison et al. (2020) proposed a generative grasping convolutional neural network (GG-CNN) to rapidly predict pixel-wise grasp quality. Based on a similar concept of grasp quality learning, the U-Grasping fully convolutional neural network (UGNet) (Song et al., 2019), Generative Residual Convolutional Neural Network (GRConvNet) (Kumra et al., 2020), and Generative Inception Neural Network (GI-NNet) (Shukla et al., 2021) were later proposed and were reported to achieve higher accuracy than GG-CNN. Le et al. (2021) extended GG-CNN to be capable of predicting the grasp quality of deformable objects by incorporating stiffness information. Morrison et al. (2019) also applied GG-CNN to a multi-view picking controller to avoid bad grasp poses caused by occlusion and collision. However, the grasp quality dataset of GG-CNN was generated by creating masks of the center third of each grasping rectangle of the Cornell Grasping dataset (Lenz et al., 2015) and Jacquard dataset (Depierre et al., 2018). This annotation method did not deeply analyze the interaction between hand and object, which is expected to lead to insufficient representation of grasp robustness.

To improve the robustness of grasp quality annotation, a physically-inspired contact force model was designed to label pixel-wise grasp quality. Mahler et al. (2018, 2019) designed a quasi-static spring model for the contact force between the vacuum cup and the object. Based on the designed compliant contact model, they assessed the grasp quality in terms of grasp robustness in a physical simulator. They further proposed GQ-CNN to learn the grasp quality and used a sampling-based method to propose an optimal grasp in the inference phase, and also extended their study by proposing a fully convolutional GQ-CNN (Satish et al., 2019) to infer pixel-wise grasp quality, which achieved faster grasping. Recently, (Cao et al., 2021) used an auto-encoder–decoder to infer the grasp quality, which was labeled by a similar contact model to that used in GQ-CNN, to generate the suction pose. However, the accuracy of the contact model depends on the model complexity and parameter tuning. High complexity may lead to a long computation cost of annotation. Parameter identification by real world experiment (Bernardin et al., 2019) might also be necessary to ensure the validity of the contact model.

Our approach also labeled the grasp quality in synthesized depth images. Unlike GQ-CNN, we proposed a more intuitive evaluation metric based on a geometrical analytic method rather than a complicated contact analytic model. Our results showed that the intuitive evaluation metric was competitive with GQ-CNN. A reachability heatmap was further incorporated to help filter pixels that had high grasp quality value but were unreachable.

2.2. Reachability Assessment

Reachability was previously assessed by sampling a large number of grasp poses and then using forward kinematics calculation, inverse kinematics calculation, or manipulability ellipsoid evaluation to investigate whether the sampled poses were reachable (Zacharias et al., 2007; Porges et al., 2014, 2015; Vahrenkamp and Asfour, 2015; Makhal and Goins, 2018). The reachability map was generated off-line, and the feasibility of candidate grasp poses was queried during grasp planning for picking static (Akinola et al., 2018; Sundaram et al., 2020) or moving (Akinola et al., 2021) objects. However, creating an off-line map with high accuracy for a large space is computationally expensive. In addition, although the off-line map considered only collisions between the manipulator and a constrained environment (e.g., fixed bin or wall) since the environment for picking in a cluttered scene is dynamic, collision checking between the manipulator and surrounding objects is still needed and this can be time consuming. Hence, recent studies have started to learn reachability with collision awareness of grasp poses. Kim and Perez (2021) designed a density net to learn the reachability density of a given pose but considered only self-collision. Murali et al. (2020) used a learned grasp sampler to sample 6D grasp poses and proposed a CollisionNet to assess the collision score of sampled poses. Lou et al. (2020) proposed a 3D CNN and reachability predictor to predict the pose stability and reachability of sampled grasp poses. They later extended the work by incorporating collision awareness for learning approachable grasp poses (Lou et al., 2021). These sampling-based methods have required designing or training a good grasp sampler for inferring the reachability. Our approach is one-shot, which directly infers the pixel-wise reachability from the depth image without sampling.

3. PROBLEM STATEMENT

3.1. Objective

Based on depth image and point cloud input, the goal is to find a grasp pose with high graspability for a suction robotic hand to pick items in a cluttered scene and then place them on a conveyor. The depth image and point cloud point are directly obtained from an Intel RealSense SR300 camera.

3.2. Picking Robot

As shown in **Figure 1A**, the picking robot is composed of a 6 degree-of-freedom (DoF) manipulator (TVL500, Shibaura Machine Co., Ltd.) and a 1 DoF robotic hand with two vacuum suction cups (**Figure 1B**). The camera is mounted in the center of the hand and is activated only when the robot is at its home position (initial pose) and, hence, can be regarded as a fixed camera installed above the bin. This setup has the merit that the camera can capture the scene of the entire bin from the view above the bin center without occlusion by the manipulator.

3.3. Grasp Pose

As shown in **Figure 1C**, the 6D grasp pose $\boldsymbol{G}$ is defined as $(\boldsymbol{p}, \boldsymbol{n}, \theta)$, where $\boldsymbol{p}$ is the target point position of the vacuum suction cup center, $\boldsymbol{n}$ is the suction direction, and θ is the rotation angle around $\boldsymbol{n}$. Given the point cloud of the target item and $\boldsymbol{p}$ position, the normal of $\boldsymbol{p}$ can be calculated simply by principal component analysis of a covariance matrix generated from neighbors of $\boldsymbol{p}$ using a point cloud library. $\boldsymbol{n}$ is the direction of the calculated normal of $\boldsymbol{p}$. As $\boldsymbol{n}$ determines only the direction of the center axis of the vacuum suction cup, a further rotation degree of freedom (θ) is required to determine the 6D pose of the hand. Note that the two vacuum suction cups are symmetric with respect to the hand center.

4. METHODS

The overall picking system diagram is shown in **Figure 2**. Given a depth image captured at the robot home position, the auto-encoder SG-U-Net++ predicts the suction graspability maps, including a pixel-wise grasp quality map and a robot reachability map. The auto-encoder SG-U-Net++ is trained using a synthesized dataset generated by a physical simulator without any human-labeled data. Cluster analysis is performed on two maps to find areas with graspbility higher than the

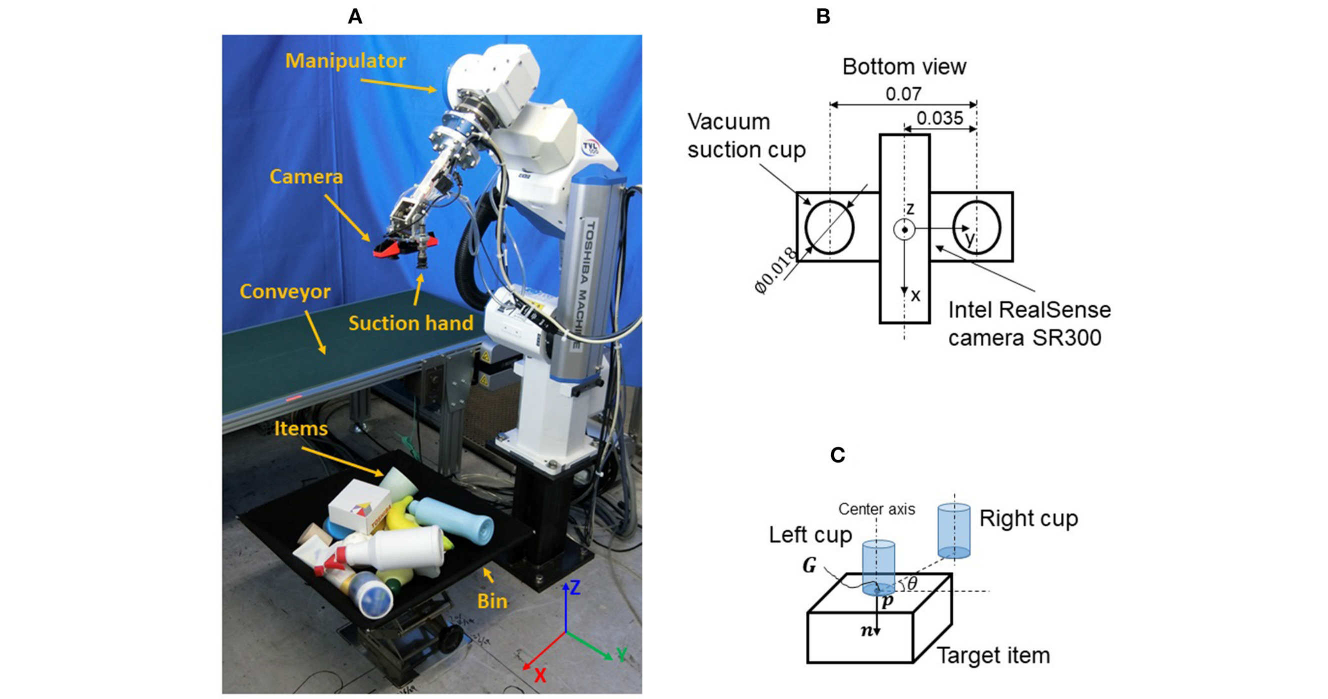


FIGURE 1 | Problem statement: **(A)** Picking robot; **(B)** Suction hand; **(C)** Grasp pose.

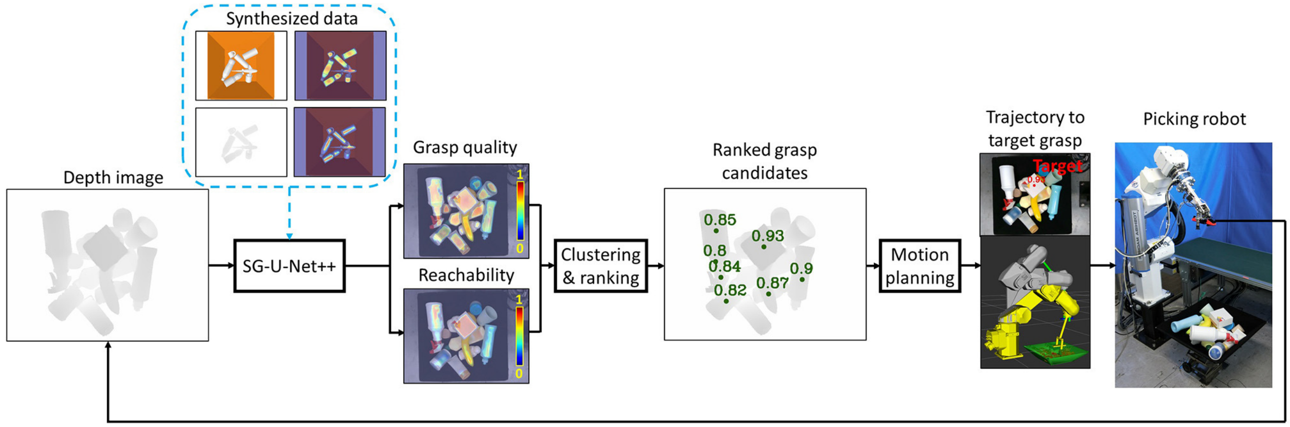


FIGURE 2 | System diagram.

thresholds. Local sorting is performed to extract the points with the highest graspbility values in each cluster as grasp candidates. Global sorting is further performed to sort the candidates of all clusters in descending order of graspbility value, and this is sent to the motion planner. The motion planner plans the trajectory for reaching the sorted grasp candidates in descending order of graspability value. The path search continues until the first successful solution of the candidate is found.

4.1. Learning the Suction Graspability

SG-U-Net++ was trained on a synthesized dataset to learn suction graspability by supervised deep learning. **Figure 3A** shows the overall dataset generation flow. A synthesized cluttered scene is first generated using pybullet to obtain a systematized depth image and object segmentation mask. Region growing is then performed on the point cloud to detect the graspable surfaces. A convolution-based method is further used to find the

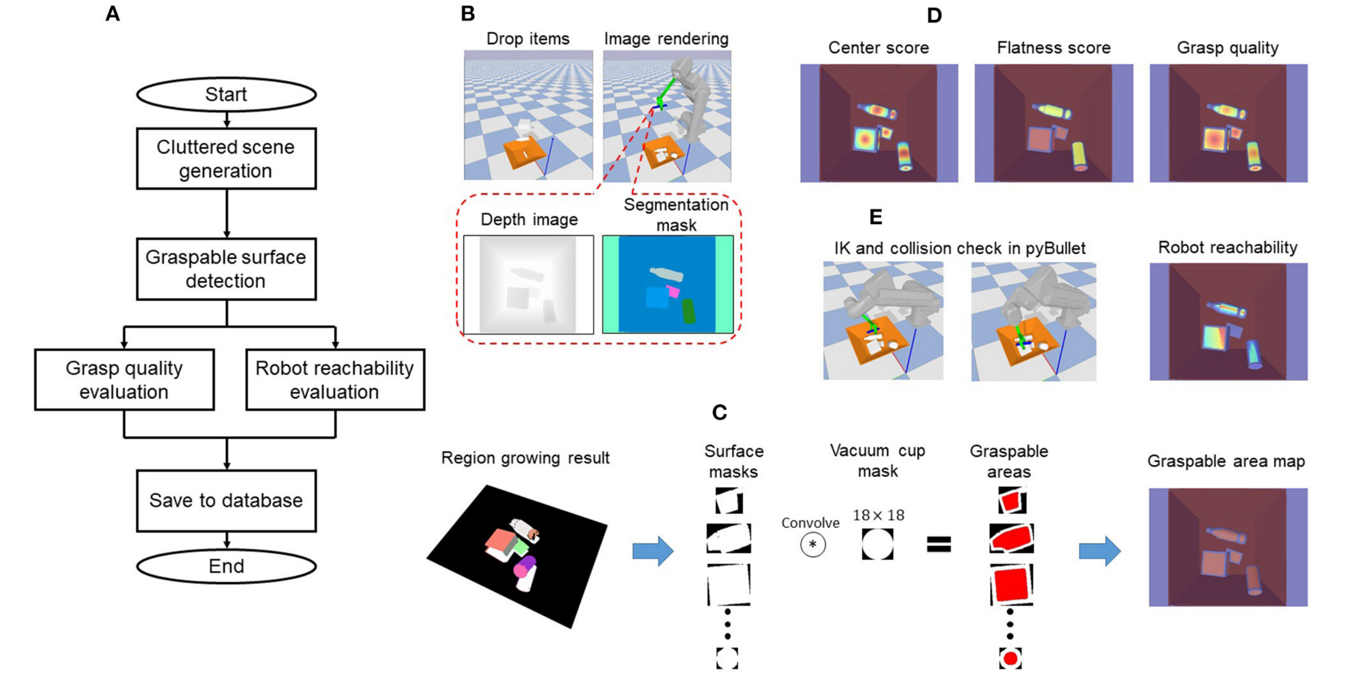


FIGURE 3 | Data generation pipeline: **(A)** Dataset generation flow; **(B)** Cluttered scene generation; **(C)** Graspable surface detection; **(D)** Grasp quality evaluation; **(E)** Robot reachability evaluation.

graspable areas of vacuum cup centers where the vacuum cup can make full contact with the surfaces. The grasp quality and robot reachability are then pixel-wise evaluated by the proposed metrics in the graspable area.

4.1.1. Cluttered Scene Generation

The object set used to synthesize the scene contains 3D CAD models from the 3DNet (Wohlkinger et al., 2012) and KIT Object database (Kasper et al., 2012). These models were used because they had previously been used to generate a dataset for which a trained CNN successfully predicted the grasp quality (Mahler et al., 2017). We empirically removed objects that are obviously difficult for suction to finally obtain 708 models. To generate cluttered scenes, a random number of objects were selected from the object set randomly and were dropped from above the bin in random poses. Once the state of all dropped objects was stable, a depth image and segmentation mask for the cluttered scene was generated, as in **Figure 3B**.

4.1.2. Graspable Surface Detection

As shown in **Figure 3C**, in order to find the graspable area of each object, graspable surface detection was performed. Given the camera intrinsic matrix, the point cloud of each object can be easily created from the depth image and segmentation mask. To detect surfaces that are roughly flat and large enough for suction by the vacuum cup, a region growing algorithm (Rusu and Cousins, 2011) was used to segment the point cloud. To stably suck an object, the vacuum cup needs to be in full contact with the surface. Hence, inspired by Domae et al. (2014), a convolution based method was used to calculate the graspable area (set of vacuum cup center positions where the cup could make full contact with the surface). Specifically, as shown in the middle of **Figure 3C**, each segmented point cloud was projected onto its local coordinates to create a binary surface mask. Each pixel of the mask represents 1 mm. The surface mask was then convolved with a vacuum cup mask (of size 18 × 18, where 18 is the cup diameter) to obtain the graspable area. At a given pixel, the convolution result is the area of the cup ($\pi * 0.009^2$ for our hand configuration) if the vacuum cup can make full contact with the surface. Refer to Domae et al. (2014) for more details. The calculated areas were finally remapped to a depth image to generate a graspable area map (right side of **Figure 3C**).

4.1.3. Grasp Quality Evaluation

Although the grasp areas of the surfaces were obtained, each pixel in the area may have a different grasp probability, i.e., grasp quality, owing to surface features. Therefore, an intuitive metric J_q (Equation 1) was proposed to assess the grasp quality for each pixel in the graspable area. The metric J_q is made up of J_c which evaluates the normalized distance to the center of the graspable area and J_s which evaluates the flatness and smoothness of the contact area between the vacuum cup and surface.

$$J_q = 0.5J_c + 0.5J_s \tag{1}$$

J_c (Equations 2, 3) was derived based on the assumption that the closer the grasp points are to the center of the graspable area, the closer they are to the center of mass of the object. Hence, grasp points close to the area center (higher J_c values) are considered to be more stable for the robot to suck and hold the object.

$$J_c = 1 - \text{maxmin}(\|\boldsymbol{p} - \boldsymbol{p}_c\|_2) \tag{2}$$

$$\text{maxmin}(\boldsymbol{x}) = \frac{\boldsymbol{x} - \min(\boldsymbol{x})}{\max(\boldsymbol{x}) - \min(\boldsymbol{x})} \tag{3}$$

where $\boldsymbol{p}$ is a point in a graspable area of a surface, $\boldsymbol{p}_c$ is the center of the graspable area, and maxmin($\boldsymbol{x}$) is a max-min normalization function.

J_s (Equations 4–6) was derived based on the assumption that a vacuum cup generates a higher suction force when in contact with a flat and smooth surface than a curved one. We defined $\boldsymbol{p}_s$ as the point set of the contact area between the vacuum cup and the surface when the vacuum cup is sucked at a certain point in the graspable area. As reported in Nishina and Hasegawa (2020), the surface flatness can be evaluated by the variance of the normals, the first term of J_s assesses the surface flatness by evaluating the variance of the normals of $\boldsymbol{p}_s$ as in Equation (5). However, it is not sufficient to consider only the flatness. For example, although a vicinal surface has a small normal variance, the vacuum cup cannot achieve suction to this kind of step-like surface. Hence, the second term (Equation 6) was incorporated to assess the surface smoothness by evaluating the residual error to fit $\boldsymbol{p}_s$ to a plane Plane($\boldsymbol{p_s}$) where the sum of the distance of each point in $\boldsymbol{p}_s$ to the fitted plane is calculated. Note that the weights in the equations were tuned manually by human observations. We adjusted the weights and parameters until we observed that the Jq map was physically plausible for grasping. We finally empirically set weights of J_c and J_s to 0.5, scaled res($\boldsymbol{p_s}$) by 5.0, and added weights 0.9 and 0.1 to two terms in Equation 4 to obtain plausible grasp quality values.

$$J_s = 0.9\text{var}(\boldsymbol{n_s}) + 0.1e^{-5\text{res}(\boldsymbol{p_s})} \tag{4}$$

$$\text{var}(\boldsymbol{n}_s) = \frac{\sum_{i=1}^{N} \boldsymbol{n}_{s,i} - \bar{\boldsymbol{n}}_s}{N-1} \tag{5}$$

$$\text{res}(\boldsymbol{p_s}) = \sum_{i=1}^{N} \|\boldsymbol{p}_{s,i} - \text{Plane}(\boldsymbol{p_s})\|_2 \tag{6}$$

where $\boldsymbol{p_s}$ are the points in the contact surface when the vacuum cup sucks at a point in the graspable area, N is the number of points in $\boldsymbol{p_s}$, $\boldsymbol{n_s}$ are the point normals of $\boldsymbol{p_s}$, var($\boldsymbol{n}_s$) is the function to calculate the variance of $\boldsymbol{n}_s$, Plane($\boldsymbol{p_s}$) is a plane equation fitted by $\boldsymbol{p_s}$ using the least squares method, and res($\boldsymbol{p_s}$) is the function to calculate the residual error of the plane fitting by calculating the sum of the distance from each point in $\boldsymbol{p_s}$ to the fitted plane.

Figure 3D shows an example of the annotated grasp quality. Points closer to the surface center had higher grasp quality values, and points located on flat surfaces had higher grasp quality (e.g., surfaces of boxes had higher grasp quality values than cylinder lateral surfaces).

4.1.4. Robot Reachability Evaluation

The grasp quality considers only the interaction between the object and the vacuum cup without considering the manipulator. As a collision check and inverse kinematics (IK) solution search for the manipulator are needed, online checking and searching for all grasp candidates is costly. Learning robot reachability helped to rapidly avoid the grasp points where the hand and manipulator may collide with the surroundings. It also assessed the ease of finding IK solutions for the manipulator.

As described in Section 3.3, $\boldsymbol{p}$ and $\boldsymbol{n}$ of a grasp pose $\boldsymbol{G}$ can be calculated from the point cloud. θ is the only undetermined variable for defining a $\boldsymbol{G}$. We sampled the θ from 0° to 355° in step intervals of 5°. IKfast (Diankov, 2010) and Flexible Collision Library (FCL) (Pan et al., 2012) were used to calculate the inverse kinematics solution and detect the collision check for each sampled θ. The reachability evaluation metric (Equations 7–8) assessed the ratio of the number of IK valid θ (had collision free IK solution) to the sampled size N_θ.

$$J_a = \frac{\sum_{i=1}^{N_\theta} \text{Solver}(\boldsymbol{p}, \boldsymbol{n}, \theta_i)}{N_\theta} \tag{7}$$

$$\text{Solver}(\boldsymbol{p}, \boldsymbol{n}, \theta_i) = \begin{cases} 1 & \text{if collision free and IK solution exists} \\ 0 & \text{else} \end{cases} \tag{8}$$

where N_θ is the size of sampled θ and Solver is the IK solver and collision checker for the robot.

Note that because the two vacuum cups are symmetric with respect to the hand center, we evaluated the reachability score of only one cup. **Figure 3E** shows an example of the robot reachability evaluation.

4.1.5. SG-U-Net++

As shown in **Figure 4**, a nest structured auto-encoder–decoder called suction graspability U-Net++ (SG-U-Net++) was used to learn the suction graspability. We used the nested architecture because it was previously reported to have high performances for semantic segmentation. Given a 256 × 256 depth image, SG-U-Net++ outputs 256 × 256 shape grasp quality and robot reachability maps. SG-U-Net++ resembles the structure of U-Net++ proposed by Zhou Z. et al. (2018). SG-U-Net++ consists of several sub encoder–decoders connected by skip connections. For example, $X^{0,0} \rightarrow X^{1,0} \rightarrow X^{0,1}$ is one of the smallest sub encoder–decoders, and $X^{0,0} \rightarrow X^{1,0} \rightarrow X^{2,3} \rightarrow X^{3,0} \rightarrow X^{4,0} \rightarrow X^{3,1} \rightarrow X^{2,2} \rightarrow X^{1,3} \rightarrow X^{0,4}$ is the largest encoder–decoder. The dense block for $X^{i,j}$ consists of two $3 \times 3 \times 32 * 2^i$ convolution (conv) layers, each of which is followed by batch normalization and rectified linear unit (ReLU) activation. The output layer connected to $X^{0,4}$ is a 1 × 1 × 2 conv layer. MSELoss (Equation 9) was used for supervised pixel-wise heatmap learning.

$$Loss = \frac{1}{H}\frac{1}{W}\sum_{i=0}^{H}\sum_{j=0}^{W} 0.5*(J_q(i,j)-\widehat{J_q}(i,j))^2+0.5*(J_a(i,j)-\widehat{J_a}(i,j))^2 \tag{9}$$

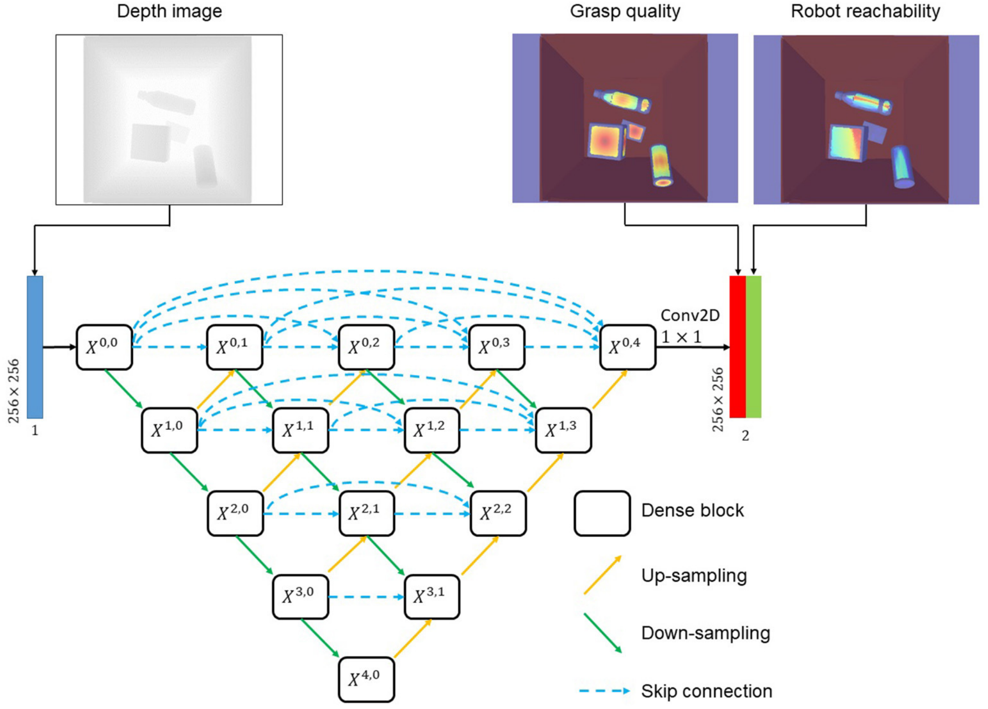


FIGURE 4 | The architecture of SG-U-Net++.

where H and W are the image height and width. $\widehat{J_q}$ and $\widehat{J_a}$ indicate the ground truth.

4.2. Clustering and Ranking

The clustering and ranking block in **Figure 2** outputs the ranked grasp proposals. To validate the role of learning reachability, we proposed two policies (Policy 1: use only grasp quality; Policy 2: use both grasp quality and reachability) to propose the grasp candidates. Policy 1 extracted the area of grasp quality values larger than threshold th_g. Policy 2 extracted the area of grasp quality score values larger than threshold th_g and the corresponding reachability score values larger than th_r. Filtering by reachability score value was assumed to help to remove pixels with high grasp quality values that are not reachable by the robot due to collision or IK error. The values of th_g and th_r were empirically set to 0.5 and 0.3, respectively. The extracted areas were clustered by scipy.ndimage.label (Virtanen et al., 2020). Points in each cluster were ranked (local cluster level) by the grasp quality values, and the point with the highest grasp quality was used as the grasp candidate for its owner clusters (refer to Ranked grasp candidates in **Figure 2**). Finally, the grasp candidates were further ranked (global level) and sent to the motion planner.

4.3. Motion Planning

Given the grasp candidates, goal poses were created for move. It (Chitta et al., 2012) to plan a trajectory. As described in 3.3, the values of $\boldsymbol{p}$ and $\boldsymbol{n}$ of a goal pose could be obtained from the corresponding point cloud information of the grasp candidates so that only θ was undetermined. As a cartesian movement path is required for the hand to suck the object, $\boldsymbol{p}$ was set to a 1 cm offset away from the object along the $\boldsymbol{n}$ direction. θ was sampled from 0° to 180° at step intervals of 5°. For each sampled goal pose, the trajectory was planned for left and right vacuum cups, respectively, and the shorter trajectory was selected as the final solution. The planned trajectory was further time parametrized by Time-Optimal Path Parameterization (toppra) (Pham and Pham, 2018) to realize position control for the robot to approach the goal pose. After reaching the goal pose, the robot hand moved down along $\boldsymbol{n}$ to suck the object. Once the contact force between the vacuum cup and object, which was measured by a force sensor, exceeded the threshold, the object was assumed to be sucked by the vacuum cup and was then lifted and placed on the conveyor.

5. EXPERIMENTS

5.1. Data Collection, Training, and Precision Evaluation

We used the proposed suction graspability annotation method in pyBullet to generate 15,000 data items, which were split into 10,000 for training and 5,000 for testing. The synthesized data was then used to train SG-U-Net++, which was implemented by pyTorch. The adam optimizer (learning rate = 1.0e−4) was used to update the parameters of the neural network during the training. The batch size was set to 16. Both data collection and training were conducted on an Intel Core i7-8700K 3.70 GHz PC with 64G RAM and 4 Nvidia Geforce GTX 1080 GPUs.

To evaluate the learning results, we used a similar evaluation method to that reported in Zeng et al. (2018b) on the testing set. For practical utilization, it is important for SG-U-Net++ to find at least one point in ground truth suction graspable area or manipulator reachable area. We defined suction graspable area as the pixels whose ground truth grasp quality scores are larger than 0.5 and approachable area as the pixels whose ground truth reachability scores are larger than 0.5. The inferred grasp quality and reachability scores were divided by thresholds into Top 1%, Top 10%, Top 25%, and Top 50%. If pixels larger than the threshold were located in the ground truth area, the pixels were considered true positive, otherwise, the pixels were considered false positive. We report the inference precision for the four thresholds above for SG-U-Net++ and compare them with Dex-Net.

5.2. Real World Picking Experiments

To evaluate and compare the performance of different policies for the picking system, a pick-and-placement task experiment was conducted. In order to investigate whether SG-U-Net++ could predict the graspability of objects with different shape complexities, we used primitive solids (a simple shape with large surfaces), commodities (general shape), and 3D-printed objects (a complex shape with small surfaces) as experimental object set (refer to **Figure 5**). All of the objects are novel objects that were not used during training. During each trial, the robot was required to pick 13 randomly posed objects (except for the cup) from a bin and then place them on the conveyor. Note that the cup was placed in the lying pose because it could not be grasped if it was in a standing pose. A grasp attempt was treated as a failure if the robot could not grasp the object in three attempts.

We conducted 10 trials for Policy 1, Policy 2, and Dex-Net 4.0 (suction grasp proposal by fully convolutional grasping policy), respectively. Note that because Dex-Net had its own grasp planning method, we directly sorted the inferred grasp quality values without clustering. To compare our proposed intuitive grasp quality evaluation metric (Equation 1) with the one used in Dex-Net, we evaluated and compared the grasp planning computation time cost and success rate of Policy 1 and Dex-Net. To evaluate the effect of incorporating the reachability score, we evaluated and compared the grasp planning computation time cost, motion planning computation time cost, and success rate of Policy 1 and Policy 2.

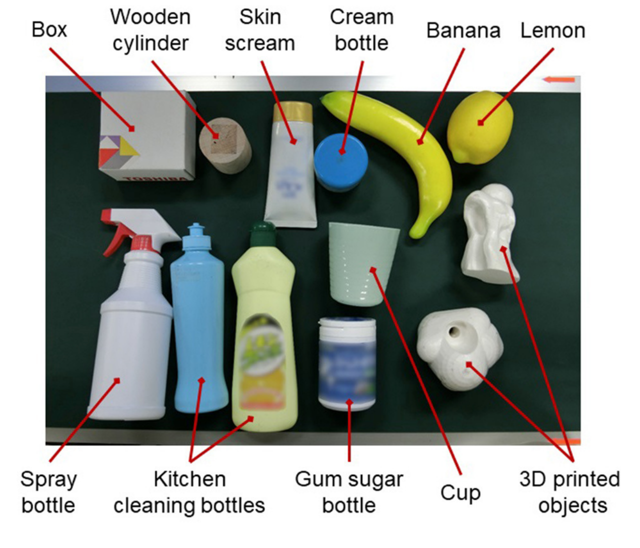


FIGURE 5 | Experiment object set.

6. RESULTS AND DISCUSSION

6.1. Inference Precision Evaluation

Table 1 shows the inference precision of grasp quality and reachability. Both SQ-U-Net++ and Dex-Net achieved high precisions for Top 1% and Top 10% but the precision of Dex-Net decreased to lower than 0.9 for Top 25% and Top 50%. This result indicates that the performance of our proposed intuitive grasp quality evaluation metric (Equation 1) was as good as a physically inspired evaluation metric. Learning the suction graspability annotation by point cloud analytic methods might not be so bad compared to dynamics analytic methods for the suction grasp task. However, the inference precision of the reachability for SQ-U-Net++ also achieved larger than 0.9 for Top 1% and Top 10%, but decreased sharply for Top 25% and Top 50%. The overall performance of reachability inference was poorer than grasp quality, indicating that reachability is more difficult to learn than grasp quality. This is probably because grasp quality can be learned from the surface features, but reachability learning requires more features such as the features of surrounding objects in addition to the surface features, leading to more difficult learning.

6.2. Picking Experiments

6.2.1. Overall Performance

Table 2 shows the experimental results of Dex-Net and our proposed method. Although all three methods achieved a high grasp success rate (>90%), our method took a shorter time for grasp planning. Moreover, the motion planning computation time was reduced by incorporating the learning of reachability. The SQ-U-Net++ Policy 2 achieved a high speed picking of approximately 560 PPH (piece per hour) (refer to **Supplementary Video**).

6.2.2. Comparison With Physically-Inspired Grasp Quality Evaluation Metric

As shown in **Table 2**, although our method was competitive with Dex-Net, it was faster for grasp planning. This result indicates that our geometric analytic based grasp quality evaluation is good enough for the picking task compared with a physically-inspired one. The evaluation of contact dynamics between a vacuum cup

TABLE 1 | Inference precision.

Score	Method	Top 1%	Top 10%	Top 25%	Top 50%
Grasp quality	Dex-Net	91.9	91.0	88.7	84.2
	SQ-U-Net++	99.8	99.6	99.2	97.5
Reachability	SQ-U-Net++	95.8	91.1	80.7	61.2

and the object surface might be simplified to just analyze the geometric features of the vacuum cup (e.g., the shape of the cup) and surfaces (e.g., surface curvature, surface smoothness, and distance from the cup center to the surface center). In addition, similar to the report in Zeng et al. (2018b), the grasp proposal of Dex-Net was farther from the center of mass. **Figure 6** shows an example of our method and Dex-Net. Our predicted grasps were closer to the center of mass of the object than the ones inferred by Dex-Net. This is because we incorporated J_c (Equation 2) to evaluate the distance from the vacuum cup center to the surface center, helping the SQ-U-Net++ to predict grasp positions much closer to the center of mass.

TABLE 2 | Experiment results.

Method	Success rate (%)	Grasp planning cost (s)	Motion planning cost (s)
Dex-Net 4.0 Suction (FC-GQCNN-4.0-SUCTION)	91.5	0.60	2.91
SQ-U-Net++ Policy1 (grasp quality only)	94.6	**0.15**	1.71
SQ-U-Net++ Policy2 (grasp quality + reachability)	**95.4**	0.17	**0.90**

Bold values indicates the best performance among three methods in the Table. For the success rate, the higher the better. For the cost (computation time) of grasp planning and motion planning, the shorter the better.

6.2.3. Role of Learning Reachability

Despite that the grasp success rate might be dominant by the grasp quality score, it is possible that although a grasp point has high grasp quality, the manipulator is not able to move to that point, leading to a longer time for motion planning. The success rate and overall system efficiency are both important for the task of bin picking. Hence, reachability learning was incorporated to assess the grasp success probability from the view point of the manipulator. The reachability heatmap helped to filter out the candidates which were with high grasp quality but the manipulator could not reach to improve the efficiency. As shown in **Table 2**, although learning reachability increased the grasp planning cost a little bit by 0.02 s due to the processes such as clustering and ranking of the reachability heatmap, it helped to reduce the motion planning cost (Policy 2: 0.90 s vs. Policy 1: 1.71s) to improve the overall system efficiency, indicating that learning reachability is worthy.

Figure 7 shows an example of the role of learning reachability. Policy 2 predicted grasps with lower collision risks with

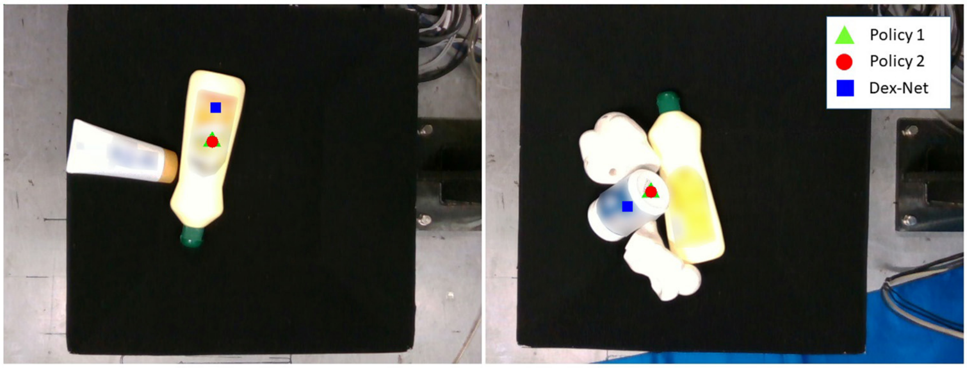


FIGURE 6 | Example of Dex-Net grasp prediction that is farther from the center of mass of the object.

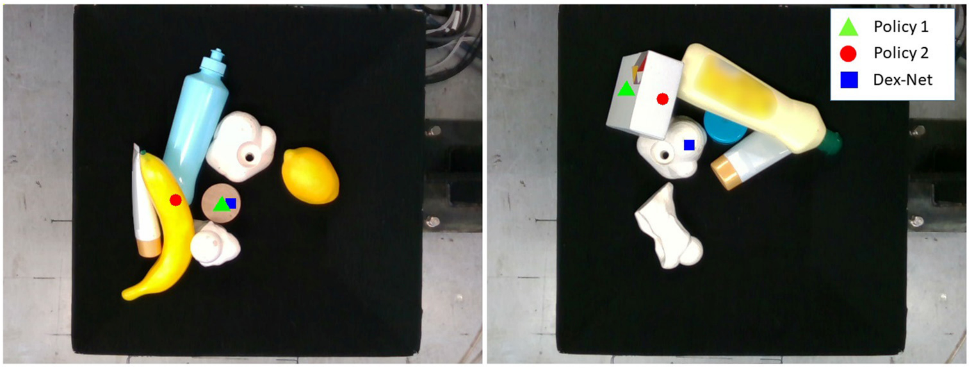


FIGURE 7 | Example of grasps predicted by Dex-Net and Policy 1 that are unreachable or difficult to reach.

neighboring objects than did Policy 1 and Dex-Net (e.g., **Figure 7** Left: Policy 1 and Dex-Net predicted grasps on a wooden cylinder that had high collision risks between the hand and 3D printed objects). Furthermore, an object might have surfaces with the same grasp quality (e.g., **Figure 7** Right: box with two flat surfaces). Whereas, Policy 2 selected the surface that was easier to reach, Policy 1 might select the one that is difficult to reach (**Figure 7** Right), since it does not consider the reachability. Therefore, Policy 2 was superior to Policy 1 and Dex-Net because it removed the grasp candidates that were obviously unable or difficult to approach. However, for Policy 1 and Dex-Net, as they considered only the grasp quality, the motion planner might first search the solutions for the candidates with high grasp quality, but those candidates might be unreachable for the manipulator and, thus, increase the motion planning effort.

6.2.4. Limitations and Future Work

Our study was not devoid of limitations. Several grasp failures occurred when picking 3D printed objects. Since the synthesized depth images differ from real ones because real images are noisy and incomplete, the neural network prediction error increased for real input depth images. This error was tolerable for objects with larger surfaces like cylinders and boxes but intolerable for 3D printed objects that have complicated shapes where the graspable areas are quite small. In the future, we intend to conduct sim-to-real (Peng et al., 2018) or depth missing value prediction (Sajjan et al., 2020) to improve the performance of our neural network. Another failure was that although not very often, the objects fell down during holding and placement because the speed of the manipulator was too high to hold the object stably. We addressed this problem by slowing down the manipulator movement during the placement action but this sacrificed the overall system picking efficiency. In the future, we want to consider a more suitable method for object holding and placement trajectory such as model based control.

Our study determined the grasping sequence by finding the grasp pose with the highest predicted grasp quality score among the filtered grasp pose candidates. The effect of other strategies such as the one that selects the target object which will not contact with the adjacent objects during the whole pick-and-place actions, or the reinforcement learning based policy (Mahler and Goldberg, 2017) will be investigated in the future.

Experiment results showed that our intuitive grasp quality evaluation metric was competitive with a physically-inspired metric, indicating that our method was plausible for bin picking of common rigid objects (e.g., primitive solids and commodities) in an electronic commerce warehouse. However, to apply our method to general industrial bin picking, object dynamics might need to be considered because the mass and materials of objects may vary in an industrial warehouse. We will investigate the effect of grasp quality metric incorporating object deformability (Xu et al., 2020; Huang et al., 2021), friction and mass distribution (Price et al., 2018; Zhao et al., 2018; Veres et al., 2020), and instability caused by robot acceleration (Khin et al., 2021) in the future.

Moreover, there is a trade-off between learning grasp quality and reachability. Increasing the weight of grasp quality loss in Equation (9) might improve the accuracy of grasp quality prediction and, thus, improve the success rate. However, it might also lead to an increased error of reachability, resulting in a long time for the motion planner to find the trajectory. Currently, we empirically set both weights to 0.5 in Equation 9, and the experimental result indicated that such a setup of weights was fine. In the future, we will investigate the influence of different weight values on the experimental result so as to find the optimal setup of weights to ensure both success rate and overall system efficiency.

Furthermore, the reachability heatmap considered the collision status of the hand goal pose for sucking the target object. The motion planner further checked whether the trajectory from the initial pose to the goal pose was collision free. This ensured that the robot could avoid colliding with other objects when grasping the target object. However, the grasped object might contact its neighboring objects when the robot lifted it after grasping. One way to avoid that is to learn the occlusion of the target object (Yu et al., 2020). If the target object was not occluded by any other objects, there would be a lower risk to make the movement of its neighboring objects when it was lifted. Another way is to predict the locations of objects by object segmentation (Araki et al., 2020; Hopfgarten et al., 2020) or object pose estimation (Tremblay et al., 2018) to make sure that there is a safe distance between the target object and its neighboring objects.

We will also extend the proposed framework for grasping by a gripper in the future. Previous studies reported that the grasp quality evaluation metric for a gripper could be designed based on geometric features (Domae et al., 2014), force closure (Miller and Allen, 2004; Roa and Suárez, 2015), or simulated gripper-object interaction (Eppner et al., 2019). For the reachability evaluation metric, the open width of a gripper should also be considered in addition to the grasp poses during evaluation.

7. CONCLUSION

We proposed an auto-encoder–decoder to infer the pixel-wise grasp quality and reachability. Our method is intuitive but competitive with CNN trained by data annotated using physically-inspired models. The reachability learning improved the efficiency of the picking system by reducing the motion planning effort. However, the performance of the auto-encoder–decoder deteriorated because of differences between synthesized and real data. In the future, sim-to-real technology will be adopted to improve performance under various environments.

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/s.

AUTHOR CONTRIBUTIONS

PJ made substantial contributions to conceiving the original ideas, designing the experiments, analyzing the results, and writing the original draft. JOa, YI, JOo, HH, AS, ST, HE, KK, and AO helped to conceptualize the final idea. PJ, YI, and JOa conducted the experiments and revised the manuscript. YI and AO supervised the project. All the authors contributed to the article and approved the submitted version.

ACKNOWLEDGMENTS

The preprint of this manuscript is available from https://arxiv.org/abs/2111.02571.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnbot.2022.806898/full#supplementary-material

REFERENCES

Akinola, I., Varley, J., Chen, B., and Allen, P. K. (2018). "Workspace aware online grasp planning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Madrid: IEEE), 2917–2924.

Akinola, I., Xu, J., Song, S., and Allen, P. K. (2021). Dynamic grasping with reachability and motion awareness. *arXiv preprint* arXiv:2103.10562. doi: 10.1109/IROS51168.2021.9636057

Araki, R., Onishi, T., Hirakawa, T., Yamashita, T., and Fujiyoshi, H. (2020). "Mt-dssd: deconvolutional single shot detector using multi task learning for object detection, segmentation, and grasping detection," in *2020 IEEE International Conference on Robotics and Automation (ICRA)* (Paris: IEEE), 10487–10493.

Asif, U., Tang, J., and Harrer, S. (2018). "Graspnet: an efficient convolutional neural network for real-time grasp detection for low-powered devices," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)* (Stockholm), *Vol.* 7, 4875–4882.

Bernardin, A., Duriez, C., and Marchal, M. (2019). "An interactive physically-based model for active suction phenomenon simulation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Macau: IEEE), 1466–1471.

Bogue, R. (2016). Growth in e-commerce boosts innovation in the warehouse robot market. *Ind. Robot*. 43, 583–587. doi: 10.1108/IR-07-2016-0194

Cao, H., Fang, H. -S., Liu, W., and Lu, C. (2021). Suctionnet-1billion: A large-scale benchmark for suction grasping. *IEEE Robot Autom. Lett*. 8, 8718–8725. Available online at: https://arxiv.org/pdf/2103.12311.pdf (accessed October 29, 2021).

Chitta, S., Sucan, I., and Cousins, S. (2012). Moveit![ros topics]. *IEEE Robot. Automat. Mag*. 19, 18–19. doi: 10.1109/MRA.2011.2181749

Chu, F.-J., Xu, R., and Vela, P. A. (2018). Real-world multiobject, multigrasp detection. *IEEE Robot. Automat. Lett*. 3, 3355–3362. doi: 10.1109/LRA.2018.2852777

Depierre, A., Dellandréa, E., and Chen, L. (2018). "Jacquard: a large scale dataset for robotic grasp detection," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Madrid: IEEE), 3511–3516.

Diankov, R. (2010). *Automated Construction of Robotic Manipulation Programs* (Ph.D. thesis). Carnegie Mellon University, Robotics Institute.

Domae, Y., Okuda, H., Taguchi, Y., Sumi, K., and Hirai, T. (2014). "Fast graspability evaluation on single depth maps for bin picking with general grippers," in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hong Kong: IEEE), 1997–2004.

Eppner, C., Mousavian, A., and Fox, D. (2019). A billion ways to grasp: an evaluation of grasp sampling schemes on a dense, physics-based grasp data set. *arXiv preprint* arXiv:1912.05604.

Fujita, M., Domae, Y., Noda, A., Garcia Ricardez, G., Nagatani, T., Zeng, A., et al. (2020). What are the important technologies for bin picking? technology analysis of robots in competitions based on a set of performance metrics. *Adv. Robot*. 34, 560–574. doi: 10.1080/01691864.2019.1698463

Hasegawa, S., Wada, K., Kitagawa, S., Uchimi, Y., Okada, K., and Inaba, M. (2019). "Graspfusion: realizing complex motion by learning and fusing grasp modalities with instance segmentation," in *2019 International Conference on Robotics and Automation (ICRA)* (Montreal, QC: IEEE), 7235–7241.

Hopfgarten, P., Auberle, J., and Hein, B. (2020). "Grasp area detection of unknown objects based on deep semantic segmentation," in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)* (Hong Kong,: IEEE), 804–809.

Huang, I., Narang, Y., Eppner, C., Sundaralingam, B., Macklin, M., Hermans, T., et al. (2021). Defgraspsim: Simulation-based grasping of 3d deformable objects. *arXiv preprint* arXiv:2107.05778.

Jiang, P., Ishihara, Y., Sugiyama, N., Oaki, J., Tokura, S., Sugahara, A., et al. (2020). Depth image-based deep learning of grasp planning for textureless planar-faced objects in vision-guided robotic bin-picking. *Sensors* 20, 706. doi: 10.3390/s20030706

Kasper, A., Xue, Z., and Dillmann, R. (2012). The kit object models database: An object model database for object recognition, localization and manipulation in service robotics. *Int. J. Rob. Res*. 31, 927–934. doi: 10.1177/0278364912445831

Khin, P. M., Low, J. H., Ang Jr, M. H., and Yeow, C. H. (2021). Development and grasp stability estimation of sensorized soft robotic hand. *Front. Rob. AI* 8, 619390. doi: 10.3389/frobt.2021.619390

Kim, S., and Perez, J. (2021). "Learning reachable manifold and inverse mapping for a redundant robot manipulator," in *2021 IEEE International Conference on Robotics and Automation (ICRA)* (Xi'an: IEEE).

Kumra, S., Joshi, S., and Sahin, F. (2020). "Antipodal robotic grasping using generative residual convolutional neural network," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Las Vegas, NV: IEEE), 9626–9633.

Kumra, S., and Kanan, C. (2017). "Robotic grasp detection using deep convolutional neural networks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Vancouver, BC: IEEE), 769–776.

Le, T. N., Lundell, J., Abu-Dakka, F. J., and Kyrki, V. (2021). Deformation-aware data-driven grasp synthesis. *arXiv preprint* arXiv:2109.05320. doi: 10.1109/LRA.2022.3146551

Lenz, I., Lee, H., and Saxena, A. (2015). Deep learning for detecting robotic grasps. *Int. J. Rob. Res*. 34, 705–724. doi: 10.1177/0278364914549607

Liu, H., Deng, Y., Guo, D., Fang, B., Sun, F., and Yang, W. (2020). An interactive perception method for warehouse automation in smart cities. *IEEE Trans. Ind. Inf*. 17, 830–838. doi: 10.1109/TII.2020.2969680

Lou, X., Yang, Y., and Choi, C. (2020). "Learning to generate 6-dof grasp poses with reachability awareness," in *2020 IEEE International Conference on Robotics and Automation (ICRA)* (Paris: IEEE), 1532–1538.

Lou, X., Yang, Y., and Choi, C. (2021). Collision-aware target-driven object grasping in constrained environments. *arXiv preprint* arXiv:2104.00776. doi: 10.1109/ICRA48506.2021.9561473

Mahler, J., and Goldberg, K. (2017). "Learning deep policies for robot bin picking by simulating robust grasping sequences," in *Conference on Robot Learning* (Mountain View, CA: PMLR), 515–524.

Mahler, J., Matl, M., Liu, X., Li, A., Gealy, D., and Goldberg, K. (2017). Dex-net 3.0: computing robust robot vacuum suction grasp targets in point clouds using a new analytic model and deep learning. *arXiv preprint* arXiv:1709.06670. doi: 10.1109/ICRA.2018.8460887

Mahler, J., Matl, M., Liu, X., Li, A., Gealy, D., and Goldberg, K. (2018). "Dex-net 3.0: computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (Brisbane, QLD: IEEE), 5620–5627.

Mahler, J., Matl, M., Satish, V., Danielczuk, M., DeRose, B., McKinley, S., et al. (2019). Learning ambidextrous robot grasping policies. *Sci. Rob*. 4, 4984. doi: 10.1126/scirobotics.aau4984

Makhal, A., and Goins, A. K. (2018). "Reuleaux: Robot base placement by reachability analysis," in *2018 Second IEEE International Conference on Robotic Computing (IRC)* (Laguna Hills, CA: IEEE), 137–142.

Mano, K., Hasegawa, T., Yamashita, T., Fujiyoshi, H., and Domae, Y. (2019). "Fast and precise detection of object grasping positions with eigenvalue templates," in *2019 International Conference on Robotics and Automation (ICRA)* (Montreal, QC: IEEE), 4403–4409.

Matsumura, R., Domae, Y., Wan, W., and Harada, K. (2019). "Learning based robotic bin-picking for potentially tangled objects," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Macau: IEEE), 7990–7997.

Miller, A. T., and Allen, P. K. (2004). Graspit! a versatile simulator for robotic grasping. *IEEE Rob. Automat. Mag.* 11, 110–122. doi: 10.1109/MRA.2004.1371616

Morrison, D., Corke, P., and Leitner, J. (2019). "Multi-view picking: Next-best-view reaching for improved grasping in clutter," in *2019 International Conference on Robotics and Automation (ICRA)* (Montreal, QC: IEEE), 8762–8768.

Morrison, D., Corke, P., and Leitner, J. (2020). Learning robust, real-time, reactive robotic grasping. *Int. J. Rob. Res.* 39, 183–201. doi: 10.1177/0278364919859066

Murali, A., Mousavian, A., Eppner, C., Paxton, C., and Fox, D. (2020). "6-dof grasping for target-driven object manipulation in clutter," in *2020 IEEE International Conference on Robotics and Automation (ICRA)* (Paris: IEEE), 6232–6238.

Nishina, Y., and Hasegawa, T. (2020). Model-less grasping points estimation for bin-picking of non-rigid objects and irregular-shaped objects. *Omron Tech.* 52, 1–8. Available online at: https://www.omron.com/global/en/technology/omrontechnics/vol52/012.html (accessed March 9, 2022).

Pan, J., Chitta, S., and Manocha, D. (2012). "Fcl: a general purpose library for collision and proximity queries," in *2012 IEEE International Conference on Robotics and Automation* (Saint Paul, MN: IEEE), 3859–3866.

Peng, X. B., Andrychowicz, M., Zaremba, W., and Abbeel, P. (2018). "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (Brisbane, QLD: IEEE), 3803–3810.

Pham, H., and Pham, Q.-C. (2018). A new approach to time-optimal path parameterization based on reachability analysis. *IEEE Trans. Rob.* 34, 645–659. doi: 10.1109/TRO.2018.2819195

Pharswan, S. V., Vohra, M., Kumar, A., and Behera, L. (2019). "Domain-independent unsupervised detection of grasp regions to grasp novel objects," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Macau: IEEE), 640–645.

Porges, O., Lampariello, R., Artigas, J., Wedler, A., Borst, C., and Roa, M. A. (2015). "Reachability and dexterity: Analysis and applications for space robotics," in *Workshop on Advanced Space Technologies for Robotics and Automation-ASTRA* (Noordwijk).

Porges, O., Stouraitis, T., Borst, C., and Roa, M. A. (2014). "Reachability and capability analysis for manipulation tasks," in *ROBOT2013: First IBERIAN robOtics* Conference (Madrid: Springer), 703–718.

Price, A., Balakirsky, S., and Christensen, H. (2018). Robust grasp preimages under unknown mass and friction distributions. *Integr. Comput. Aided Eng.* 25, 99–110. doi: 10.3233/ICA-180568

Roa, M. A., and Suárez, R. (2015). Grasp quality measures: review and performance. *Auton. Rob.* 38, 65–88. doi: 10.1007/s10514-014-9402-3

Rusu, R. B., and Cousins, S. (2011). "3d is here: Point cloud library (pcl)," in *2011 IEEE International Conference on Robotics and Automation* (Shanghai: IEEE), 1–4.

Sajjan, S., Moore, M., Pan, M., Nagaraja, G., Lee, J., Zeng, A., et al. (2020). "Clear grasp: 3d shape estimation of transparent objects for manipulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)* (Paris: IEEE), 3634–3642.

Satish, V., Mahler, J., and Goldberg, K. (2019). On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks. *IEEE Rob. Automat. Lett.* 4, 1357–1364. doi: 10.1109/LRA.2019.2895878

Shukla, P., Pramanik, N., Mehta, D., and Nandi, G. (2021). Gi-nnet\rgi-nnet: Development of robotic grasp pose models, trainable with large as well as limited labelled training datasets, under supervised and semi supervised paradigms. *arXiv preprint* arXiv:2107.07452.

Song, Y., Fei, Y., Cheng, C., Li, X., and Yu, C. (2019). "Ug-net for robotic grasping using only depth image," in *2019 IEEE International Conference on Real-time Computing and Robotics (RCAR)* (Irkutsk: IEEE), 913–918.

Sundaram, A. M., Friedl, W., and Roa, M. A. (2020). "Environment-aware grasp strategy planning in clutter for a variable stiffness hand," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Las Vegas, NV,: IEEE), 9377–9384.

Tremblay, J., To, T., Sundaralingam, B., Xiang, Y., Fox, D., and Birchfield, S. (2018). Deep object pose estimation for semantic robotic grasping of household objects. *arXiv preprint* arXiv:1809.10790.

Utomo, T. W., Cahyadi, A. I., and Ardiyanto, I. (2021). Suction-based grasp point estimation in cluttered environment for robotic manipulator using deep learning-based affordance map. *Int. J. Automat. Comput.* 18, 277–287. doi: 10.1007/s11633-020-1260-1

Vahrenkamp, N., and Asfour, T. (2015). Representing the robot's workspace through constrained manipulability analysis. *Auton. Rob.* 38, 17–30. doi: 10.1007/s10514-014-9394-z

Veres, M., Cabral, I., and Moussa, M. (2020). Incorporating object intrinsic features within deep grasp affordance prediction. *IEEE Rob. Automat. Lett.* 5, 6009–6016. doi: 10.1109/LRA.2020.3010444

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., et al. (2020). SciPy 1.0: fundamental algorithms for scientific computing in python. *Nat. Methods* 17, 261–272. doi: 10.1038/s41592-020-0772-5

Wohlkinger, W., Aldoma, A., Rusu, R. B., and Vincze, M. (2012). "3dnet: Large-scale object class recognition from cad models," in *2012 IEEE International Conference on Robotics and Automation* (Saint Paul, MN: IEEE), 5384–5391.

Xu, J., Danielczuk, M., Ichnowski, J., Mahler, J., Steinbach, E., and Goldberg, K. (2020). "Minimal work: a grasp quality metric for deformable hollow objects," in *2020 IEEE International Conference on Robotics and Automation (ICRA)* (Paris: IEEE), 1546–1552.

Xu, R., Chu, F.-J., and Vela, P. A. (2021). Gknet: grasp keypoint network for grasp candidates detection. *arXiv preprint* arXiv:2106.08497. doi: 10.1177/02783649211069569

Yu, Y., Cao, Z., Liang, S., Geng, W., and Yu, J. (2020). A novel vision-based grasping method under occlusion for manipulating robotic system. *IEEE Sens. J.* 20, 10996–11006. doi: 10.1109/JSEN.2020.2995395

Zacharias, F., Borst, C., and Hirzinger, G. (2007). "Capturing robot workspace structure: representing robot capabilities," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems* (San Diego, CA: IEEE), 3229–3236.

Zeng, A., Song, S., Lee, J., Rodriguez, A., and Funkhouser, T. (2020). Tossingbot: Learning to throw arbitrary objects with residual physics. *IEEE Trans. Rob.* 36, 1307–1319. doi: 10.1109/TRO.2020.2988642

Zeng, A., Song, S., Welker, S., Lee, J., Rodriguez, A., and Funkhouser, T. (2018a). "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Madrid: IEEE), 4238–4245.

Zeng, A., Song, S., Yu, K.-T., Donlon, E., Hogan, F. R., Bauza, M., et al. (2018b). "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (Brisbane, QLD: IEEE), 3750–3757.

Zhang, H., Lan, X., Bai, S., Zhou, X., Tian, Z., and Zheng, N. (2019). "Roi-based robotic grasp detection for object overlapping scenes," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Macau: IEEE), 4768–4775.

Zhao, Z., Li, X., Lu, C., and Wang, Y. (2018). "Center of mass and friction coefficient exploration of unknown object for a robotic grasping manipulation," in *2018 IEEE International Conference on Mechatronics and Automation (ICMA)* (Changchun: IEEE), 2352–2357.

Zhou, X., Lan, X., Zhang, H., Tian, Z., Zhang, Y., and Zheng, N. (2018). "Fully convolutional grasp detection network with oriented anchor box," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Madrid,: IEEE), 7223–7230.

Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., and Liang, J. (2018). "Unet++: a nested u-net architecture for medical image segmentation," in *Deep Learning in*

Medical Image Analysis and Multimodal Learning for Clinical Decision Support (Springer), 3–11.

Conflict of Interest: PJ, JOa, YI, JOo, HH, AS, ST, HE, KK, and AO were employed by Toshiba Corporation.

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.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 31 March 2022
doi: 10.3389/fnbot.2022.817168

A Distributed Multi-Agent Formation Control Method Based on Deep Q Learning

Nianhao Xie[1,2], *Yunpeng Hu*[1,2] *and Lei Chen*[3*]

[1] College of Aerospace Science and Engineering, National University of Defense Technology, Changsha, China, [2] Hunan Key Laboratory of Intelligent Planning and Simulation for Aerospace Missions, Changsha, China, [3] National Innovation Institute and Defense Technology, Beijing, China

Distributed control method plays an important role in the formation of a multi-agent system (MAS), which is the prerequisite for an MAS to complete its missions. However, the lack of considering the collision risk between agents makes many distributed formation control methods lose practicability. In this article, a distributed formation control method that takes collision avoidance into account is proposed. At first, the MAS formation control problem can be divided into pair-wise unit formation problems where each agent moves to the expected position and only needs to avoid one obstacle. Then, a deep Q network (DQN) is applied to model the agent's unit controller for this pair-wise unit formation. The DQN controller is trained by using reshaped reward function and prioritized experience replay. The agents in MAS formation share the same unit DQN controller but get different commands due to various observations. Finally, through the min-max fusion of value functions of the DQN controller, the agent can always respond to the most dangerous avoidance. In this way, we get an easy-to-train multi-agent collision avoidance formation control method. In the end, unit formation simulation and multi-agent formation simulation results are presented to verify our method.

Keywords: multi-agent system, distributed control, formation control, deep Q learning, collision avoidance

Edited by:
Yimin Zhou,
Shenzhen Institutes of Advanced Technology (CAS), China

Reviewed by:
Zhijun Yang,
Middlesex University, United Kingdom
Jiahao Chen,
Institute of Automation (CAS), China

****Correspondence:***
Lei Chen
chenl@nudt.edu.cn

Received: *17 November 2021*
Accepted: *23 February 2022*
Published: *31 March 2022*

Citation:
Xie N, Hu Y and Chen L (2022) A Distributed Multi-Agent Formation Control Method Based on Deep Q Learning.
Front. Neurorobot. 16:817168.
doi: 10.3389/fnbot.2022.817168

1. INTRODUCTION

In recent years, with the development of manufacturing, microelectronics, and communication technology, unmanned multi-agent systems (MASs), such as unmanned land vehicles, unmanned underwater vehicles, and unmanned aerial vehicles have emerged. Taking the advantage of tireless, fearless, and infallible characters over a human being, MASs begin to be applied in many areas, e.g., express distribution, disaster search and rescue, ecological monitoring, entertainment performances, and military confrontation. As a primary guarantee for MAS coordination and cooperation during task execution, formation control has received more and more extensive attention. Although there are many theoretical achievements, lots of formation control methods for MAS still lack autonomy in practical applications, where manual remote control or trajectory planning is needed to coordinate the agents. This is the main reason that MAS cannot cope with many tasks that require high timelines.

Researchers usually turn the formation control problem into a consistency problem and model the topology among agents using the undirect (Eren et al., 2003) or direct (Falconi et al., 2011) graph. Based on the communication or observation topology, the stability and convergence of the designed formation control protocol can be proved. Nevertheless, this kind of method (Li et al., 2019a; Guo et al., 2020) takes the agents as a mass point and neglects their volume, causing these methods insecure for possible collision between agents. In addition, the obstacles in the environment are usually indescribable, which also raises challenges for these methods. The potential function is widely used to describe obstacles. Using leader-follower topology, Liang et al. (2020) came up with an adaptive leader-follower formation control method for unmanned aerial vehicle (UAV) swarms with motion constraints and unknown disturbances, where the collision avoidance between UAVs is achieved with the artificial potential method. Merheb et al. (2016) modeled the environment as an incompressible flow field and designed a potential function for obstacles. Then panel method was applied to generate formation trajectory, i.e., streamlines of flow. Wu et al. (2016) proposed an obstacle envelope modeling method to model the obstacles. Each obstacle can be regarded as a dipole where the positive pole attracts agents and the negative pole distracts agents. However, trajectory planning methods require complex pre-design and calculation, making them only applicable in the mission planning stage and becoming invalid in on-board formation control. Behavior-based methods can also work to deal with obstacles. Xu et al. (2014) made behavior rules for agents to bypass obstacles and move along the walls. Lee and Chwa (2018) defined the inner, middle, and outer boundaries to wrap the obstacles so that agents can take effective collision avoidance behaviors in different boundaries. Although many details need to be considered, the behavior-based method cannot ensure stability and optimal during formation (Kamel et al., 2020).

To reduce reliance on the experience of engineers to make behavior rules, behavior learning methods begin to be applied in the formation control. Jin (2019) achieved stable tracking of followers to the leader with iteration learning method, where the only angle of sight observation is needed. Zhao et al. (2020) considered the relative distance constraints between agents and planned collision avoidance trajectory by iteration learning. Sanz et al. (2008) took the first step to apply the reinforcement learning method in the formation control. The agent with a Q learning controller can learn when to move forward and backward to keep aligned with the other two agents. However, when the state or/and action space become continuous, the corresponding Q table will be too large to describe or to train. The appearance of deep Q network (DQN) (Mnih et al., 2013) and deep deterministic policy gradient (DDPG) (Lillicrap et al., 2016) have solved this problem because continuous state and/or action space can be modeled by a neural network with limited weights. Sui et al. (2019) built long short-term memory (LSTM) networks to learn the formation controller of a follower to track the leader. The training is divided into two-stage. First, the network is supervised to learn the trajectory from the optimal reciprocal collision avoidance (ORCA) method (Van Den Berg et al., 2011), which is a well-known formation control method to deal with collision avoidance. Then, the agent explores better control protocol using reinforcement learning. Wang (2019) equipped the DDPG with double prioritized experience replay. Without considering collision avoidance, the command of roll angle for a UAV is generated by the DDPG controller and executed by a traditional PID controller. Although trained in the simulation environment, the learned roll angle command also works on hardware-in-the-loop simulation. However, Sui et al. (2019) and Wang (2019) only focus on the situation of one leader with one follower. Li et al. (2019b) trained multi-agent collision avoidance controller under decomposition methodology. At first, they predicted the value function from one-to-one collision avoidance rules using the iterative policy evaluation method. Then, the one-to-one value functions of multi-agent are fused and corrected to a multi-agent collision avoidance policy.

In this article, based on the decomposition methodology, we train a DQN for formation with leader-follower topology. First, we extract the simplest environment from the multi-agent formation control environment, i.e., one agent tracks its follower and needs only to avoid one obstacle. Then, in this simplest environment, the agent with DQN controller is trained with reshaped reward function and prioritized experience replay. Finally, through the min-max fusion of the DQN value functions, the agent can avoid more than one obstacle during formation control. The main contributions of this article are as follows:

- The multi-agent formation problem is decomposed to the pair-wise control problem, called the unit formation problem, which reduces the state dimension of DQN and thus, simplifies the learning of control policy.
- The reward function of the DQN controller is reshaped, which improves the training performance of DQN.
- By min-max fusion of DQN value function, the pair-wise controller is upgraded to a multi-agent formation controller.
- The action field is proposed to visually compare the DQN formation controller before and after reinforcement learning.

This article is organized as follows. In section 2, the multi-agent formation control problem is modeled. In section 3, after the proposed decomposition-fusion learning framework is sketched out, we explained the details of the unit formation controller, and a min-max fusion method to deal with multiple obstacles in multi-agent formation. In section 4, simulations are presented to validate our method. Finally, we conclude this article in section 5.

2. PROBLEM DESCRIPTION

Oh et al. (2015) gave the general description of the formation control problem without considering collision avoidance, while, when considering the collision avoidance, the formation control problem can be modeled as follows. Supposed there are N agents in the formation, and let the state of agent i be $\mathbf{x}_i$ and the kinematics model and observation model are f_i and g_i, respectively. The multi-agent state set is $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]$, and the observation set is $\mathbf{Y} = \left[\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N\right]$, and the control output set is $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_N]$. The target of multi-agent formation controller at time t is calculating control output set $\mathbf{U}_t$ according to states sequence $\mathbf{X}_{t_0\,:\,t}$ and observations $\mathbf{Y}_{t_0\,:\,t}$ from starting time t_0 to current time t so that the agents can avoid collision with each other and form the expected geometric

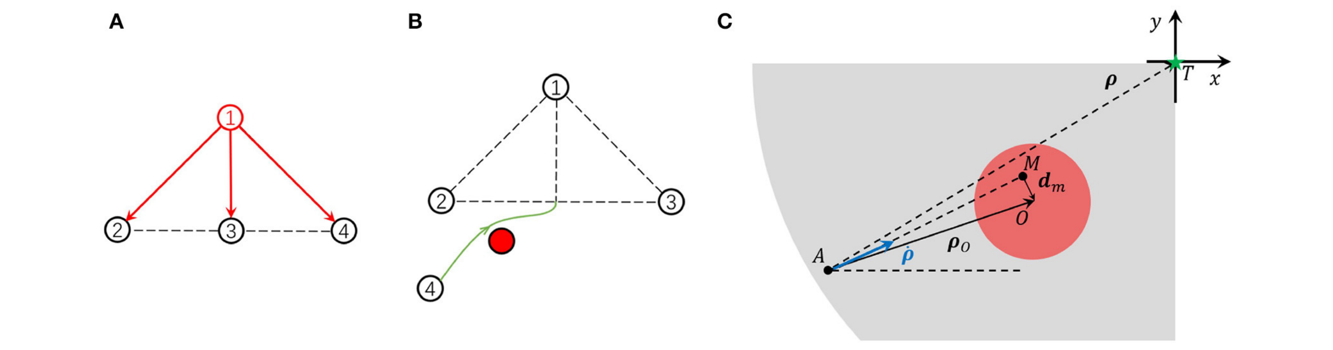


FIGURE 1 | (A) The leader-follower topology in the formation control. **(B)** The unit problem of formation control. **(C)** The relative kinematics model.

configuration. This problem can be described by optimization equations as follows.

$$\begin{cases} \min\limits_{\mathbf{U}_{t_0:t}} \; \|F(\mathbf{X}_t) - F(\mathbf{X}^*)\| \\ \text{s.t. } \; C(\mathbf{X}) < 0 \end{cases} \tag{1}$$

where the function $F(\cdot)$ maps the states of agents to geometric constraints and the function $C(\cdot)$ is collision function. When a collision happens, $C(\mathbf{X}) \geqslant 0$. The optimization objective is to make the geometric configuration $F(\mathbf{X})$ converge to the expected $F(\mathbf{X}^*)$. The states transformation and observation of agent i obey the following equation.

$$\begin{cases} \dot{\mathbf{x}}_i = f_i(\mathbf{x}_i, \mathbf{u}_i) \\ \mathbf{y}_i = g_i(\mathbf{X}) \end{cases} \tag{2}$$

3. FORMATION CONTROL METHOD

In this section, the decomposition-fusion framework to train the formation controller is proposed. Then, the unit controller is designed and learned by the improved deep Q learning method to get a pair-wise policy. Finally, the min-max fusion method that makes the pair-wise policy applicable for multi-agent formation is elaborated.

3.1. The Decomposition-Fusion Framework

In a multi-agent formation, as the number of agents increases, each agent needs to communicate and cooperate with more agents, which require higher computation capacity. By designing a suitable formation topology, the relationship among agents can be simplified so that the communication and calculation burden is relieved.

With leader-follower topology, the formation can be automatically kept and globally controlled by the leader. In the clustered MAS, considering that the follower in one cluster can become the leader in other clusters, this kind of hierarchical topology makes the control of a large-scale system possible. As shown in **Figure 1A**, the follower calculates its expected relative position by observing its leader and then moves toward the destination. At the same time, the follower is not allowed to collide with the other agents in the formation. From the agents' point of view, an agent takes other agents in the formation as moving or static obstacles. The agent aims to observe the leader, move toward the relative destination, and meanwhile, avoid collision with those obstacles. Thus, the formation control problem can be treated as an obstacle avoidance problem from this insight. A formation controller is expected to avoid multiple obstacles. Instead of using a one-step learning framework that directly takes multiple obstacles into account, we proposed the two-step decomposition-fusion learning framework which can give the agent the ability to deal with multiple obstacles.

As shown in **Figure 2A**, assuming that there are three obstacles, a direct way is to learn a controller that takes the observations of all obstacles as input. But this leads to two troubles. One is that, as the number of obstacles increase the input dimension, learning samples, and the parameters increase, which increases the learning difficulty. The other one is that, if the number of obstacles is not three, e.g., two or four, the learned controller will be inapplicable.

In this article, a decomposition-fusion framework is proposed to solve the above problems. Inspired by the pair-wise policy (Kuchar and Yang, 2000) and attention mechanism (Mnih et al., 2014), in the decomposition stage, we assume that the agent only focuses on a certain obstacle, and was "blind" to the rest obstacles. Such a "uint controller" has a fixed input dimension and is relatively simple to learn. However, it is clear that the unit controller cannot ensure collision avoidance to all obstacles at once. As shown in **Figure 2B1**, when the agent pays attention to obstacle 1, the agent may not be able to avoid obstacle 2. **Figures 2B2,B3** show similar things. Thus, a "fusion controller" will be designed to make the agent learn how to allocate attention and balance its pair-wise policy for different obstacles. In this way, an approximately global optimal solution can be gained.

3.2. Decomposition Stage: Reword Reshaped DQN for Unit Control

3.2.1. Modeling of Unit Control Problem

In the uint problem, with the assumption that agents in d-dimension space have a second-order linear kinematics model, only the relative movement of agent A, agent's target position T, and obstacle O need to be considered. As shown in **Figure 1C**, A, T, O is the agent's current position, expected relative position,

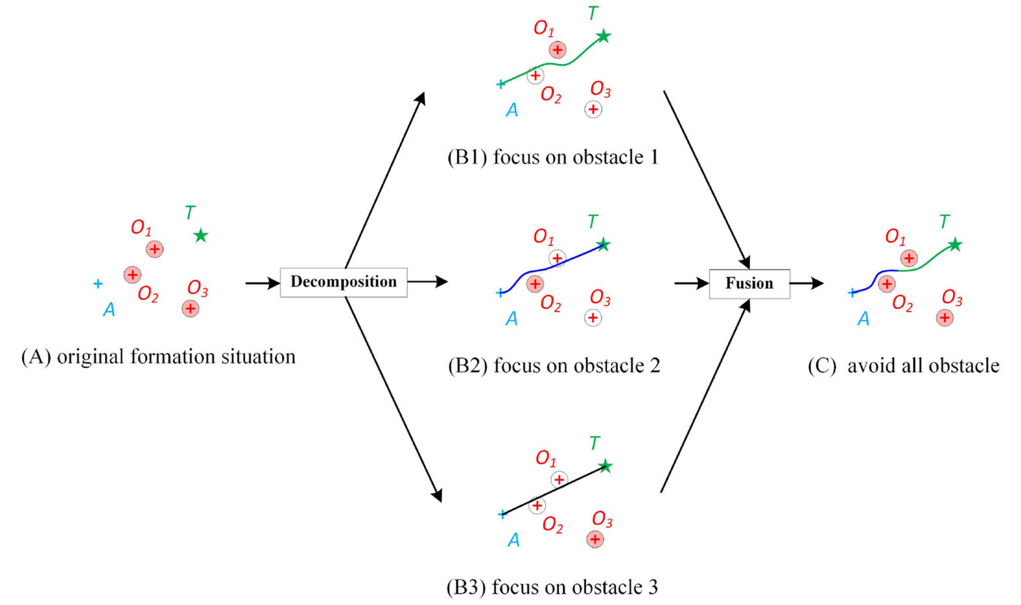


FIGURE 2 | Diagram of the proposed decomposition-fusion formation control framework. **(A)** A formation situation that the agent A is supposed to destinate target T but may be blocked by multiple obstacles O_1, O_2, and O_3. The circle of the obstacle represents its collision zone. If the trajectory of the agent does not intersect with the collision zone, it means that the agent has no collision with the obstacle. **(B)** The decomposition of original formation by focusing obstacles one by one. The "focused" obstacle is colored by red while the "ignored" obstacles are colored white. **(C)** The trajectory from the fused controller.

and obstacle center, respectively. The red circle is the threat zone and the gray zone is the motion permitted zone. $\mathbf{d}_m$ is the predicted minimum distance from the agent to the obstacle center if the agent keeps the current moving direction. We define a relative coordinate system in which the origin of the coordinates is fixed on the target position T and its axis is parallel to one inertial coordinate. Denote the relative position from the agent to the target as $\boldsymbol{\rho}$ and the relative position to the obstacle as $\boldsymbol{\rho}_O$. The agent's velocity in the relative coordinates is $\dot{\boldsymbol{\rho}}$. Then, the state of the agent i is $\mathbf{x}_i = \left[\boldsymbol{\rho}_i^\top \ \dot{\boldsymbol{\rho}}_i^\top\right]^\top$. The kinematics model of the agent is

$$\begin{cases} \dot{\mathbf{x}}_i = \mathbf{A}\mathbf{x}_i + \mathbf{B}\mathbf{u}_i \\ \mathbf{y}_i = \mathbf{C}\mathbf{x}_i \end{cases} \text{where } \mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \otimes \mathbf{I}_d, \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \mathbf{I}_d,\ \mathbf{C} = \mathbf{I}_{2n} \tag{3}$$

where $\otimes$ is Kronecker product and $\mathbf{u}_i \in \mathbb{R}^d$ is control output which has constraint $\mathbf{u}_i \in \mathcal{U}$. The agent has velocity constraints $\mathbf{v}_i \in \mathcal{V}$. The safe distance between agents is d_{safe} which means the formation would fail if any distance between two agents was less than d_{safe}. We also limit the agent to move inside a circle area with radius D. If the agent moves close enough to the target position, i.e., $|\boldsymbol{\rho}| \leqslant d_e$, the unit problem is solved and d_e is called formation error.

3.2.2. Buiding Markov Decision Process (MDP) for Unit Problem

The MDP is commonly used to describe continuous decision problems. An MDP can be defined by the tuple $M = \langle \mathcal{S}, \mathcal{A}, T_r, R, \gamma \rangle$, where $\mathcal{S}$ is state space, $\mathcal{A}$ is action space, T_r is state transition function, R is reward function, and γ is decay coefficient. A time t, the agent chooses action $a_t \in \mathcal{A}$ using policy π based on state observation $s_t \in \mathcal{S}$. Then, the state transits to s_{t+1} at time $t+1$, where the transition probability is $Pr\left(s_{t+1}|s_t, a_t\right) = T_r\left(s_t, a_t, s_{t+1}\right)$. Meanwhile, the agent gets reward $r_{t+1} = R\left(s_t, a_t, s_{t+1}\right)$. The goal of the continuous decision is finding the best policy π^* which maximizes the cumulative expected reward $\sum_{t=0}^{\infty} \gamma^t r_t$.

The state value function and state-action value function of MDP are briefly introduced for the convenience of explaining reward shaping and DQN training. Before training the policy of the unit formation MDP, the way of interaction between the agent and the designed environment needs to be decided. Then, the details of other elements of the unit formation MDP, including state, action, transition function, and reward function, are discussed.

State value function. The policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ gives the probability of choosing one action at the current state, and obviously, $\sum_{a \in \mathcal{A}} \pi\,(s, a) = 1$. The state value function of policy π at state s can be denoted as $V^\pi(s)$, which means no matter what policy the agent uses before the state s, if the agent always uses policy π from the state s to the end of the decision process, then the cumulative expected reward from state s_t to s_∞ is $V^\pi(s)$.

$$V^\pi\,(s) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s \right] \tag{4}$$

State-action value function. The state value after action a is state-action value $Q^\pi\,(s, a)$, which is the cumulative expected reward from state s_t to s_∞ is $V^\pi(s)$ when the agent transits to

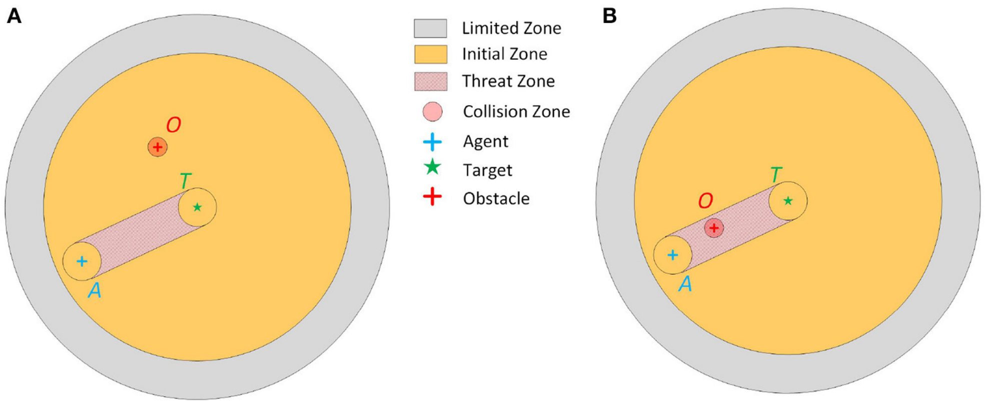


FIGURE 3 | (A) The episode mechanism of unit formation problem when the obstacle O is out of the threat zone. **(B)** The episode mechanism of unit formation problem when the obstacle O is in the threat zone.

new state s' after acting action a and keeps using policy π from the state s' to the end of the decision process.

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty}\gamma^{k} r_{t+k+1} \mid s_t = s, a_t = a\right] \tag{5}$$

Episode mechanism. The samples of reinforcement learning are generated during the agent's exploration in the environment. Therefore, how the agent interacts with the environment needs to be decided, which is called episode mechanism in this article. Consider the environment in two-dimension space, in order to simulate the collision avoidance during formation control, we design the episode mechanism as illustrated in **Figure 3**. Let the target position be the center of the initial zone and limited zone, where the initial zone and limited zone are circular area with radius d_3 and D, respectively. At the beginning of every episode, the agent is randomly initialized in the initial zone. A direct idea is to place the obstacle randomly on the limited zone. However, in most instances, the agent can simply move to the target along a straight line, which makes the agent lack of experience to learn how to avoid obstacle. Considering that an obstacle in the route of the agent to its target will threaten the agent, the threat zone is defined as an double-ended-wrench like area, where the width of the double-ended wrench is d_1. The initialization strategy of the obstacle is as follows. In 50% of cases (as shown in **Figure 3A**), the obstacle is initialized randomly in the limited zone except threat zone, in the other 50% cases (as shown in **Figure 3B**), the obstacle is initialize randomly in the threat zone. The initial velocity of the agent is also random but the obstacle is assumed to be static for simplification. At each control time step, the agent receives command from the DQN controller and executes this action. This process keeps going until the following events occur:

- the agent reaches its target position (finish)
- the agent collides with the obstacle (collision)
- the agent moves outside the limited zone (out of range)
- the agent moves more than $n_{\max}$ step (out of step)

Therefore, the four kinds of state, i.e., finish, collision, out of range, and out of step are the terminal states of one episode.

MDP State. The unit formation control involves the agent, the obstacle, and the target. Therefore, the MDP State $s_t = [\boldsymbol{\rho}^\top(t), \dot{\boldsymbol{\rho}}^\top(t), \boldsymbol{\rho}_O^\top(t)]^\top = [x_t, y_t, v_{x,t}, v_{y,t}, x_{O,t}, y_{O,t}]^\top$, i.e., the agent's relative position to target x, y, relative velocity v_x, v_y, and agent's relative position to the obstacle x_O, y_O at step t, which contains enough information to calculate control output. In addition, the velocity constraint is $v_x, v_y \in [-1, 1]$.

Transition function. The transition function of the MDP state is based on the agent's discrete kinematic Equation (3) but added the obstacle observation.

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ v_{x,t+1} \\ v_{y,t+1} \\ x_{O,t+1} \\ y_{O,t+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & dt & 0 & 0 & 0 \\ 0 & 1 & 0 & dt & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & dt & 0 & 0 & 0 \\ 0 & 1 & 0 & dt & 0 & 0 \end{bmatrix} \begin{bmatrix} x_t \\ y_t \\ v_{x,t} \\ v_{y,t} \\ x_{O,t} \\ y_{O,t} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \tag{6}$$

where u_x, u_y is the formation control command and dt is the time interval.

Action. The MDP action is directly defined to be the formation control command of the agent. Considering discrete action space, the action space is

$$\begin{aligned} a = [u_x, u_y] \in \{&[0,0], [-2,0], [-1,0], [1,0], [2,0], [0,-2], \\ &[0,-1], [0,1], [0,2]\} \end{aligned}$$

Reward function. As mentioned in the episode mechanism, one episode will be terminated under four situations, i.e. finish, collision, out of the range, and out of the step. Correspondingly, there are four kinds of terminal rewards for the unit formation MDP. Let the original reward function be:

$$R(s) = \begin{cases} 2, \textit{if finish} \\ -2, \textit{if collision} \\ -2, \textit{if out of range} \\ 0, \textit{otherwise} \end{cases} \tag{7}$$

3.2.3. Reward Shaping

Although the DQN can be trained by the original reward function Equation (7), the agent cannot get meaningful reward most of the time because the original reward is very sparse. Especially at the beginning of the training, it is hard for the agent to gain a way to the target. Therefore, the original reward function Equation (7) is not conducive to the convergence of training. In this article, we reshape the original reward to make the DQN get the reward at every step, which will improve the learning process.

For brevity, s_t, a_t, and s_{t+1} are abbreviated as s, a, and s'. Having original MDP $M = \langle \mathcal{S}, \mathcal{A}, T, \gamma, R \rangle$, the reward-reshaped MDP can be denoted as $M' = \left\langle \mathcal{S}, \mathcal{A}, T, \gamma, R' \right\rangle$, where $R'(s, a, s') : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ is reshaped reward

$$R'(s, a, s') = R(s, a, s') + F(s, a, s'), \tag{8}$$

and $F(s, a, s') : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ is an additional reward that need to be designed to ensure that the optimal solution of the original MDP is the same as the reward-shaped MDP. According to the reward reshaping principle (Ng et al., 1999), if exits $\Phi(s) : \mathcal{S} \rightarrow \mathbb{R}$ which makes $F(s, a, s') = \gamma\Phi(s') - \Phi(s)$, then the additional reward $F(s, a, s')$ is potential, which can ensure the invariance of optimal solution. Denote the state value function of the two equivalent MDP as V^{π}_{M}, $V^{\pi}_{M'}$ respectively, then

$$V^{\pi}_{M'} = V^{\pi}_{M} - \Phi(s) \tag{9}$$

If $\Phi(s) = V^{*}_{M}(s)$, then $V^{*}_{M'}(s) \equiv 0$. Equation (9) theoretically indicates that the learning of $V^{*}_{M'}(s)$ will be easier if we reshape the reward function by $\Phi(s)$ that predicts $V^{*}_{M}(s)$ (Ng et al., 1999). The agent has a higher state value when it approaches the target position, and the agent has a lower state value when the collision threats exist and the agent approaches the obstacle. Let

$$\Phi(s) = \begin{cases} -\rho & \text{if } \rho_O > 2d_{safe} \\ -\rho + (\rho_O - 2d_{safe}) & \text{if } \rho_O \leqslant 2d_{safe} \end{cases} \tag{10}$$

Finally, the additional reward function is defined as

$$F(s, a, s') = \begin{cases} -\gamma\rho(s') + \rho(s) & \text{if } \rho_O > 2d_{safe} \\ -\gamma\rho(s') + \rho(s) + \gamma\rho_O(s') & \\ -\rho_O(s) + 2(1-\gamma)d_{safe} & \text{if } \rho_O \leqslant 2d_{safe} \end{cases} \tag{11}$$

3.2.4. Q Learning for Optimal Policy

If the optimal state-action value function Q is known, the optimal policy is

$$\pi^{*}(s) = \arg\max_{a} Q^{*}(s, a) \tag{12}$$

The Q learning (Sutton and Barto, 1998) can iteratively make the Q function approaches the optimal because the current state-action value function can be presented using the next state-action value function according to the Bellman equation, i.e.,

$$Q^{\pi}(s, a) = \sum_{s'} P^{a}_{ss'} \left[R^{a}_{ss'} + \gamma \mathbb{E}_{\pi} \Big[\sum_{k=0}^{\infty} \gamma^{k} r_{t+k+2} | s_{t+1} = s' \Big] \right] \tag{13}$$

where $P^{a}_{ss'} = Pr\left(s_{t+1} = s' | s_t = s, a_t = a\right)$ $R^{a}_{ss'} = R\left(s_t = s, a_t = a, s_{t+1} = s'\right)$. Therefore, the optimal state-action value function satisfies the equation

$$Q^{*}(s, a) = \mathbb{E}_{s'} \left[R(s, a, s') + \gamma \max_{a'} Q^{*}(s', a') \right] \tag{14}$$

According to Equation (14), the Q function can be solved by temporal difference and eventually converge to Q^{*}. In traditional Q learning method, the Q function is defined by numerical table, which is unsuitable when the state space becomes larger or even infinite. Mnih et al. (2013) used a deep network to model the Q table so that it is possible to define infinite states and actions with finite weights of the network. They built two networks called evaluation network Q and target network Q^{-}, respectively. The structure of the two networks is the same, but they have different parameters. Denoting the parameter of evaluation network and target network as w and w^{-} respectively, the error of evaluation network to target network is

$$J(w) = \mathbb{E}_{s'} \left[\left(R^{a}_{ss'} + \gamma \max_{a'} Q^{-}(s', a') - Q(s, a) \right)^{2} \right] \tag{15}$$

the parameters of evaluation network can be updated by

$$w \leftarrow w + \alpha\nabla J = w + \alpha \left(R^{a}_{ss'} + \gamma \max_{a'} Q(s', a')^{-} - Q(s, a) \right) \nabla Q(s, a) \tag{16}$$

where α is the learning rate. The parameters of target network w^{-} are updated to parameters of the evaluation network w every N_{replace} training step, making the parameters of evaluation networks approach the optimal parameters w^{*}. In this way, the iterative temporal difference method is accomplished in DQN training.

The training processes of unit formation problems with and without reward shaping are shown in **Figures 4A,B**, respectively. The training samples come from the experience (s, a, r, s') of the agent, obtained by interacting with the environment. These samples are temporarily stored in the experience pool with size N_{pool}. However, at every training step, only N_{batch} samples will be trained during overpopulation of the output error between the evaluation network and target network. Therefore, the priority experience replay method (Schaul et al., 2016) is employed in this article to increase the probability of samples with large errors. The samples with a high error are more likely to be selected to train the networks, which can speed up the learning process.

3.3. Fusion Stage: Multi-Agent Formation Control by Min-Max Fusion of Unit Formation Control Policy

The unit formation DQN controller only equips the agent with the ability to avoid one certain obstacle during formation. However, there will be more than one potential threat in the multi-agent formation control. To make the agent knows which obstacle needs to be preferentially treated with, the

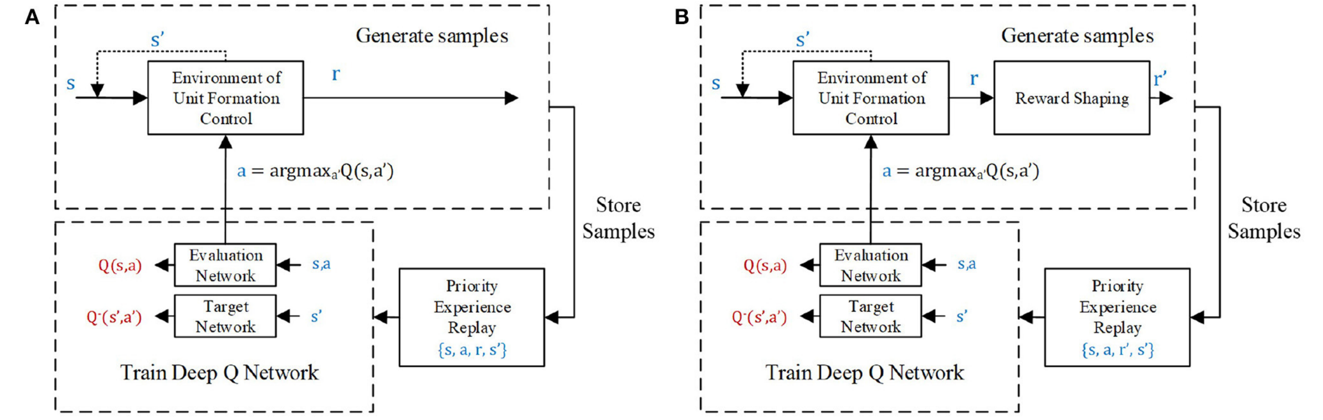


FIGURE 4 | The unit formation deep Q network (DQN) controller learning process **(A)** without and **(B)** with reward shaping.

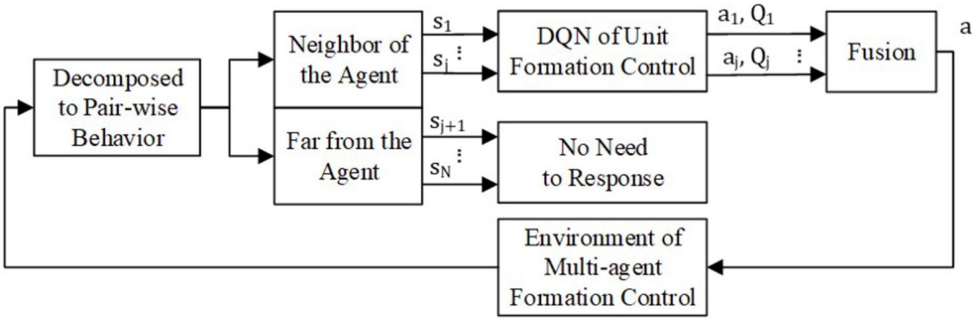


FIGURE 5 | Policy fusion for multi-agent formation control.

min-max fusion method proposed by Chryssanthacopoulos and Kochenderfer (2011) is employed to fuse pair-wise unit formation control policy.

To simplified the denotation, we omit the subscript i. The min-max fusion process is shown in **Figure 5**. The agent views other agent j in the formation as an obstacle. If the distance between agent i and j is beyond the threshold which makes the observation or communication impossible or the agent j is too far to threaten agent i, there is no need for agent i to respond to agent j. If not, having the state of agent i and j as input, the pair-wise policy can output the optimal action a_j and responding state-action function Q_j. From the definition of the state-action value function in section 3.2.2, Q_j predicts the cumulative expected reward after executing action a_j. A higher state-action value means lower collision threats. Thus, the lowest state-action value of all the optimal pair-wise policies most likely comes from the biggest threat. The min-max fusion method makes the agent respond first to the biggest threat. Therefore, the balanced global policy from the pair-wise policy is

$$a = \arg\min_{a_j \in \mathcal{A}_j} Q(s_j, a_j) \tag{17}$$

and

$$a_j = \arg\max_{a' \in \mathcal{A}} Q(s_j, a') \tag{18}$$

where $\mathcal{A}_j = \{a_j\}$ is all the pair-wise policy of the agent i to the other agents $j, j = 1, 2, \cdots, N, j \neq i$.

For every agent in the formation, it can get a global formation control policy without extra training by using Equations (17) and (18).

4. SIMULATIONS AND RESULTS

To verify our multi-agent formation algorithm step by step, we first present two demos of unit formation control in section 4.1. Then, in section 4.2, two more demos of multi-agent control are given to validate our method of multi-agent formation with collision avoidance.

4.1. Unit Formation Control Policy

4.1.1. Training

As shown in **Figure 6**, the evaluation network and target network are both composed of three fully connected layers. There are 256, 256, and 128 neurons in the first, second, and third layers, respectively. The weights are initialized using Gaussian random

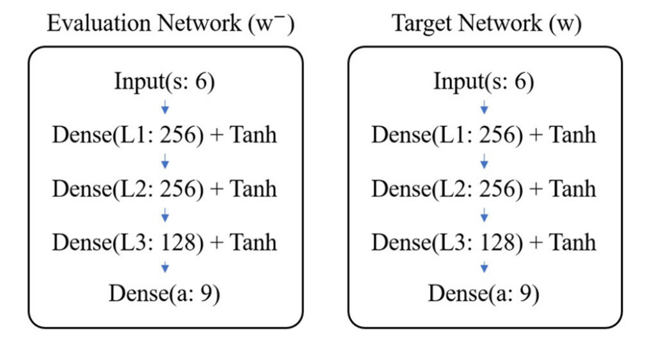


FIGURE 6 | Structure of DQN.

$N(0, 0.3^2)$ and the biases are initialized using uniform random $U(0, 0.1)$. Except for the last layer, the other layers' output is activated by tanh function.

The reward decay coefficient is set as 0.95 and the size of the experience pool to store samples is set as 50,000. For every $N_{\text{replace}} = 2,000$ training step, the parameters of the target network will be replaced by those of the evaluation network. At the beginning of training, the ϵ-greedy probability is 0.95 which allows the agent to explore the environment as far as possible. As the training goes on, the ϵ linearly decreases by 0.01 every 100 episodes to limit the exploration range of the agent until reaches the minimum value $\epsilon_{\min} = 0$. The weights are updated by Adam Optimizer with an initial learning rate $\alpha = 10^{-3}$. Like the probability of ϵ-greedy exploration, the learning rate also decreases every 100 episodes, not linearly but exponentially, i.e., the learning rate becomes 0.99 times the old learning rate ($\alpha \leftarrow 0.99\alpha$). At every training step, $N_{\text{batch}} = 32$ samples are selected by the priority experience replay method. The training stops when the number of trained episodes reaches 20,000.

We assume that the episode is finished if the distance between the agent and its target position is less than $d_e = 3$, and the agent is safe if the distance between the agent and the obstacle is more than $d_{\text{safe}} = 5$. To generate samples, the inner and outer radius of the obstacle zone are $d_1 = 10$ and $d_2 = 25$, respectively. The agent is limited to moving within the circular zone (radius $D = 100$) around the target position. If the episode goes more than $n_{\max} = 100$ control steps, or collision or crossing happens, the episode is forced to stop.

All the parameters related to the training of DQN is listed in **Table 1**.

To test the reward shaping in this article, we trained the original DQN and reward shaping DQN five times using the same episode mechanism, parameters, and network structure, but different network initial parameters, and different random seeds to initialize the agent's position, agent's velocity, and obstacle's position. During the training process, the DQN with and without reward shaping is tested. Let $\bar{R}_{\text{test}}(m)$ be the average reward of $N_{\text{test}} = 300$ test episodes after training DQN by m training episodes. Denote the DQN trained by m training episodes as mth DQN, the average reward of the mth DQN in one training process is

TABLE 1 | Parameter of deep Q network (DQN) and training.

Parameters	Value	Parameters	Value
Reward decay coefficient γ	0.95	update target network every N_{replace}*step*	2×10^3
Initial ϵ-greedy probability ϵ	0.95	Minimum ϵ-greedy probability $\epsilon_{\min}$	0
Initial learning rate α	10^3	Size of experience poolN_{pool}	5×10^4
Total episode N_e	2×10^4	Batch size N_{batch}	32
Maximum step in each episode $n_{\max}$	100	Formation error d_e	3
Safe distance d_{safe}	5	Width of threat zone d_1	10
Radius of limited zone D	100	Radius of initial zone d_3	50
Simulation interval dT	0.1	Control interval dT_c	1

$$\bar{R}_{\text{test}}(m) = \sum_{k=1}^{N_{\text{test}}} \sum_{j=j_{0,k}}^{n_k} r_{k,j}(m) \tag{19}$$

where $r_{k,j}(m)$ is the reward of the jth step in the kth test episode obtained by mth DQN. In addition, $j_{0,k} = \max\{1, n_k - 10\}$, meaning that the average reward is the average of the last 10 steps when the total steps of kth episode are more than 10. To make the reward of DQN with and without reshaping comparable, the average reward is normalized by the maximum average reward during the whole training process.

$$\bar{R}'_{\text{test}}(m) = \frac{\bar{R}_{\text{test}}(m) - \min_n\{\bar{R}_{\text{test}}(n)\}}{\max_n\{\bar{R}_{\text{test}}(n)\} - \min_n\{\bar{R}_{\text{test}}(n)\}}, n = 1, \cdots, N_e \tag{20}$$

The mean curve and SE of the normalized average reward with and without reshaping are recorded in **Figures 7A,B**. As shown in **Figure 7A**, the normalized average reward without reshaping reaches the maximum at about 5,000 training episodes by 0.8 ± 0.2. In **Figure 7B**, the normalized average reward with reshaping grows to the maximum value at about 5,000 training episodes by about 0.95 and the SE is small. Therefore, the convergence process with reward shaping is more stable.

Figures 7C,D present the terminal states of test episodes. We call the episodes out of step, out of range, and collision as unfinished episodes. Both terminal state curves show a rising trend of the finished episodes. As shown in **Figure 7C**, without reward shaping, most episodes terminate due to the state of out of step, and collision before 5,000 episodes because the sparse reward makes it hard for the agent to get a positive experience. The finish rate end by about 75% and there still is a 10% collision probability. In **Figure 7D**, benefiting from the reward shaping, the failure episodes of the reward shaping controller are mainly out of range, and the collision episodes only occur before 5,000

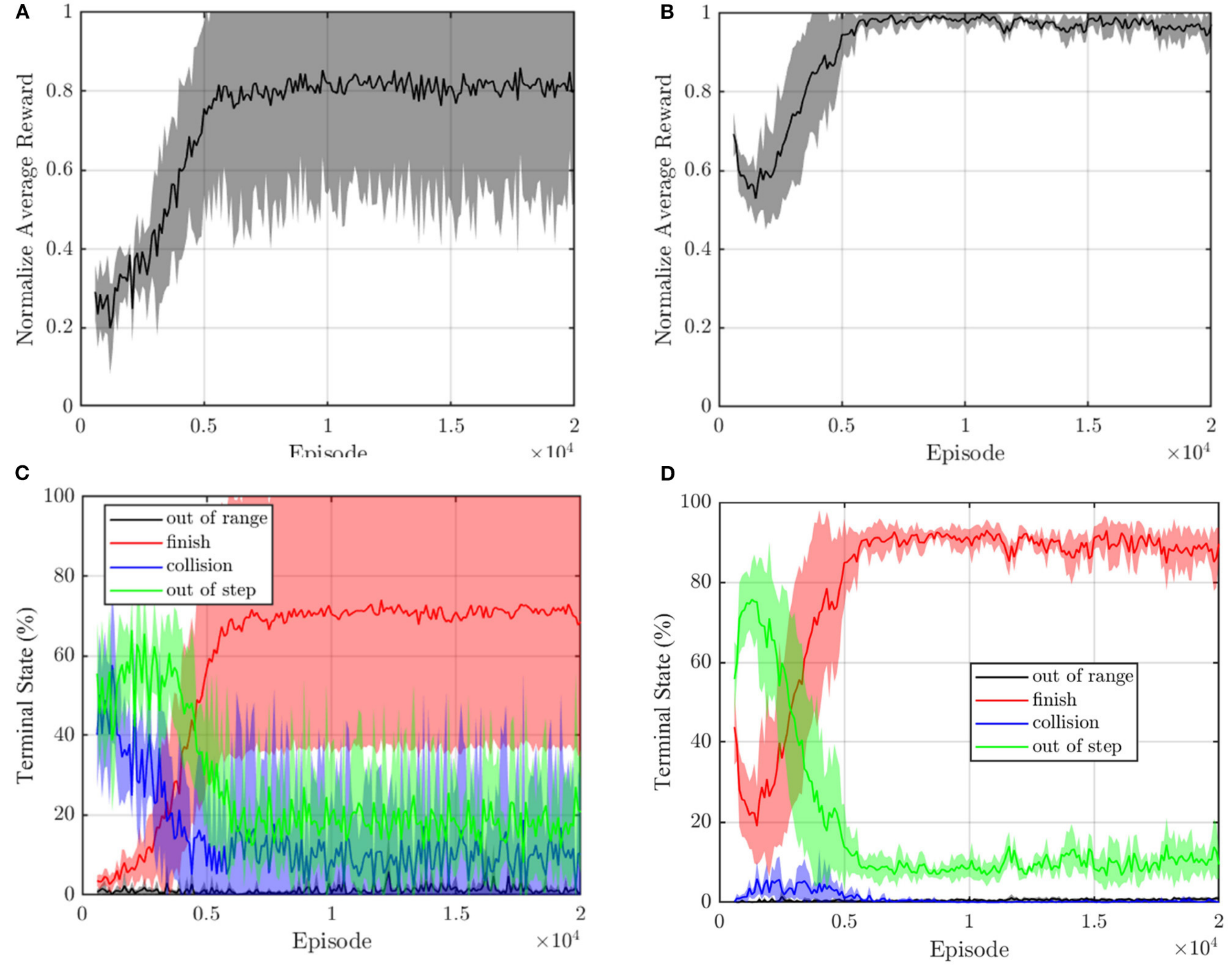


FIGURE 7 | The normalized average reward of DQN **(A)** without and **(B)** with reward shaping. The terminal state during train process **(C)** without and **(D)** with reward reshaping. The translucent shadow their SE.

steps which indicate that the agent effectively learns an obstacle avoidance strategy. The finish rate reaches about 90% with nearly zero collision probability. The reward shaping improves the convergence of terminal state curves. Therefore, we can conclude that the reward shaping method in this article improves the convergence of DQN.

4.1.2. Demo: Visualized Action Field

Sui et al. (2019) colored the action space according to the probability that action is optimal to analyze the learned policy. However, the action space only shows the policy in some keyframes. To globally visualize the learned policy by DQN, the action field is defined as follows. Supposed that the current position and velocity of the agent is $\boldsymbol{\rho}$ and $\dot{\boldsymbol{\rho}}$, the optimal action can be calculated by the DQN policy π. By fixing the velocity $\dot{\boldsymbol{\rho}}$ but traversing the position $\boldsymbol{\rho}$ of the agent, the function mapping $F_A(\boldsymbol{\rho}|\pi, \dot{\boldsymbol{\rho}})$ represents the action field. In other words, the point in the action field is the optimal action when the agent of velocity $\dot{\boldsymbol{\rho}}$ locates in the same position. **Figure 8** shows the action field of $\dot{\boldsymbol{\rho}} = \mathbf{0}$ with the target position at point $[0, 0]$ and limited zone in the square of $[-40, 40] - -[-40, 40]$.

As noted, the green, red, purple, and blue colors represent up, right, down, and left action, respectively, in which deeper color indicates bigger acceleration. Since the parameters of DQN are randomly initialized before training, the zero-velocity action field at this time is chaotic, as shown in **Figure 8A**. After 20,000 episodes of training, as illustrated in **Figure 8B**, the action field becomes regular. More precisely, the action field is composed of four triangle zones, which make the agent always move toward the expected position.

4.1.3. Demo: Unit Formation Control

In this subsection, two scenarios are presented to show the performance of the unit formation control. They represent two typical situations, i.e., the obstacle is or is not on the line between the initial position of the agent and the target position. When

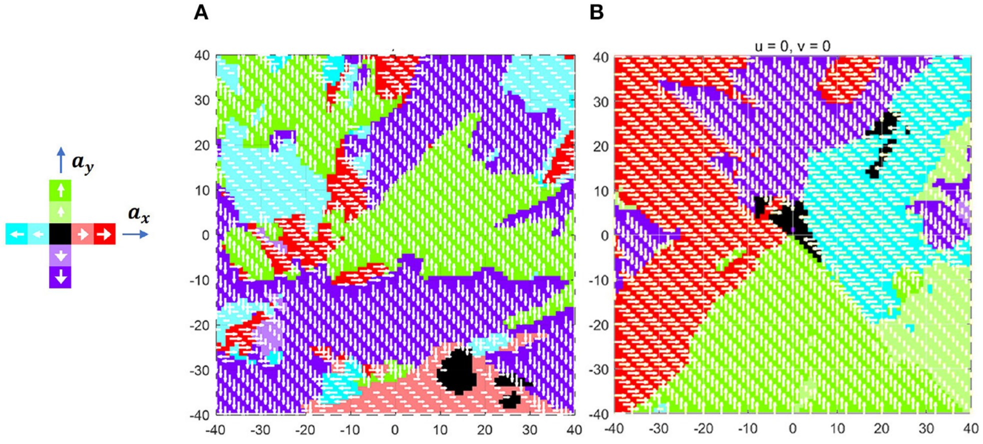


FIGURE 8 | Action field **(A)** before and **(B)** after training.

there is no obstacle in the direction to the target position, as shown in **Figures 9A,B**, the agent adjusts its direction at about the 15th and the 30th control step to aim at the target. If the obstacle blocks the way, the agent moves toward the target until the distance to the obstacle is close enough to alarm the agent. As shown in **Figures 10A,B**, the agent turns left at about the 22nd control step and turns right at about the 37th control step to avoid the obstacle.

Due to discrete acceleration and fixed time step, the zero-velocity constraint is not satisfied, when the agent arrives at the target position. This makes the trajectory of the agent fluctuate near the target position. It is noted that the agent does not need to avoid other agents when it is close enough to the target position. Therefore, to eliminate the continuously small fluctuation, in the following demos, the simple proportional derivative (PD) control method is employed when the agent is close enough to the target (assuming that the distance to the target is less than 10 in this article).

4.2. Multi-Agent Formation Control Policy

4.2.1. Demo: Avoid Multiple Obstacles

To show how the proposed multi-agent formation control method avoids multiple obstacles, the scenario as shown in **Figure 11A** is presented, where six static obstacles locate in point $[-20, 40]$, $[0, 40]$, $[20, 40]$, $[-20, 20]$, $[0, 20]$, and $[20, 20]$ and the agent is initialized in $[0, 60]$ with velocity $[0, 0]$. The target is $[0, 0]$. The blue dots represent the trajectory of the agent which is remarked by blue circles every 10 control steps. The collision zone of obstacles are represented by six colored circles.

In the beginning, the agent moves downward but turns right at the 10th control step to avoid Obstacle 2. At control step 20, the agent corrects its direction to approach target position. However, it must change direction at the 30th control step to avoid Obstacle 5. Finally, the agent faces the target again at the 40th control step and becomes stable at the target position after 60 steps. **Figure 11B** records the deviation of the agent from the target position. It is observed that the agent always approaches the target in y-direction but adjusts its velocity in x-direction to avoid the obstacles which are faster than any policy that changes the vertical velocity.

Next, we illustrate how the agent uses the unit controller to avoid multiple obstacles by fusion. By fusing the pair-wise state-action value of the six obstacles using equation (17), the agent can respond to the obstacle, that has a bigger threat, with a higher priority and thus, avoids more than one obstacle in the environment. To testify that the min-max fusion method indeed guides the agent responding to the most likely threat, we compare the min-max state-action value with two other kinds of most likely threat, i.e., minimum distance threat and minimum left time threat.

The minimum distance threat comes from the nearest obstacle, as shown in **Figure 12A**. However, the nearest obstacle may not have the biggest threat because the agent may move far away from this obstacle. Thus, the direction of the motion needs to be considered.

Supposed that the agent keeps moving at current speed in a straight line, it will reach the position which is the nearest point M (as shown in **Figure 1C**) to the obstacles in the straight line. Let $\mathbf{d}_m$ be the minimum distance to the obstacle in the straight line and Δt be the left time for the agent moving from the current position to point M, we have

$$\mathbf{d}_m = \boldsymbol{\rho}_O - \frac{\dot{\boldsymbol{\rho}}_O \left(\boldsymbol{\rho}_O \cdot \dot{\boldsymbol{\rho}}_O\right)}{\left|\dot{\boldsymbol{\rho}}_O\right|^2} \tag{21}$$

$$\Delta t = \frac{\boldsymbol{\rho}_O \cdot \dot{\boldsymbol{\rho}}_O}{\left|\dot{\boldsymbol{\rho}}_O\right|^2} \tag{22}$$

For all the obstacles, whose minimum distances $\mathbf{d}_m$ are less than the safe distance ($|\mathbf{d}_m| < d_{\text{safe}}$), the biggest threat to the agent comes from the obstacle, for which it has the minimum left time Δt to the point M. This is the second most likely threat criterion called minimum left time threat. According to equation

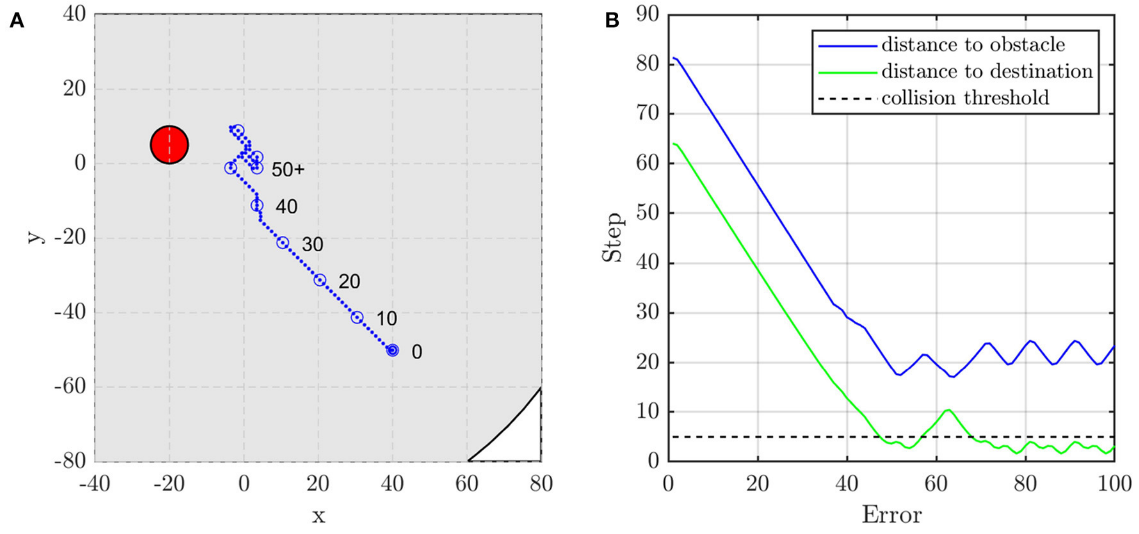


FIGURE 9 | The agent's **(A)** trajectory, and **(B)** distance to obstacle and destination when the obstacle is not on the line between the agent's initial and target position. The red circle is the obstacle with the radius of d_{safe} and the blue dots are the trace of the agent.

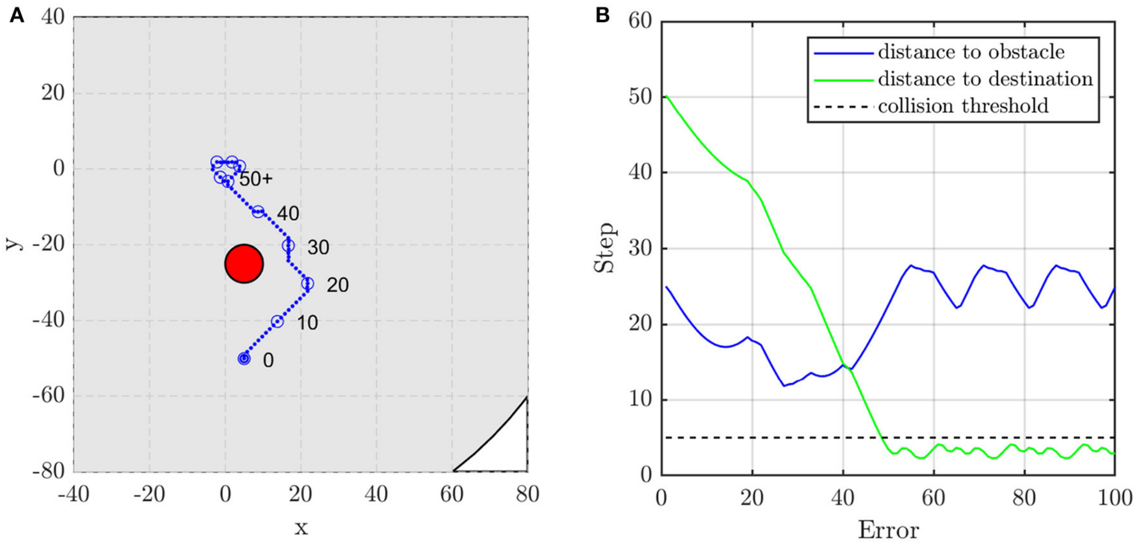


FIGURE 10 | The agent's **(A)** trajectory and **(B)** distance to obstacle and destination when the obstacle is on the line between the agent's initial and target position. The red circle is the obstacle with the radius of d_{safe} and the blue dots are the trace of the agent.

(22), the left time (equivalent to left steps) for all the obstacles is shown in **Figure 12B**. In a similar way, according to equation (18), the maximum state-action value Q for the obstacle is shown in **Figure 12C**.

The aforementioned three kinds of threats are predicted and compared in **Figure 12D**. It is observed that the most likely threat indicated by minimum-maximum Q value is consistent with that of minimum left time threat in most of the steps, which validates our multi-agent formation control method.

4.2.2. Demo: Multi-Agent Line Formation

Line formation is one of the most common formations in MAS. However, many formation control methods may be unreliable the line formation because they do not consider collision avoidance (Li et al., 2019a; Guo et al., 2020). The controller trained by static obstacles cannot ensure that the agent successfully avoids the moving obstacles. However, in some engineering scenarios, taking the other agents in the formation as static is reasonable. On the one hand, without filtering technique, the estimation of other agents' velocity may be unusable due to the observation noise. On the other hand, in most cases, like ground robots and quadrotors, the safe distance d_{safe} between agents is much larger than the agent's moving distance Δd within decision interval Δt. When $d_{safe} \gg \Delta d$, the dynamic obstacle can be approximated as static because the internal logic of the controller is that if command makes the agent go away from the obstacle, then it

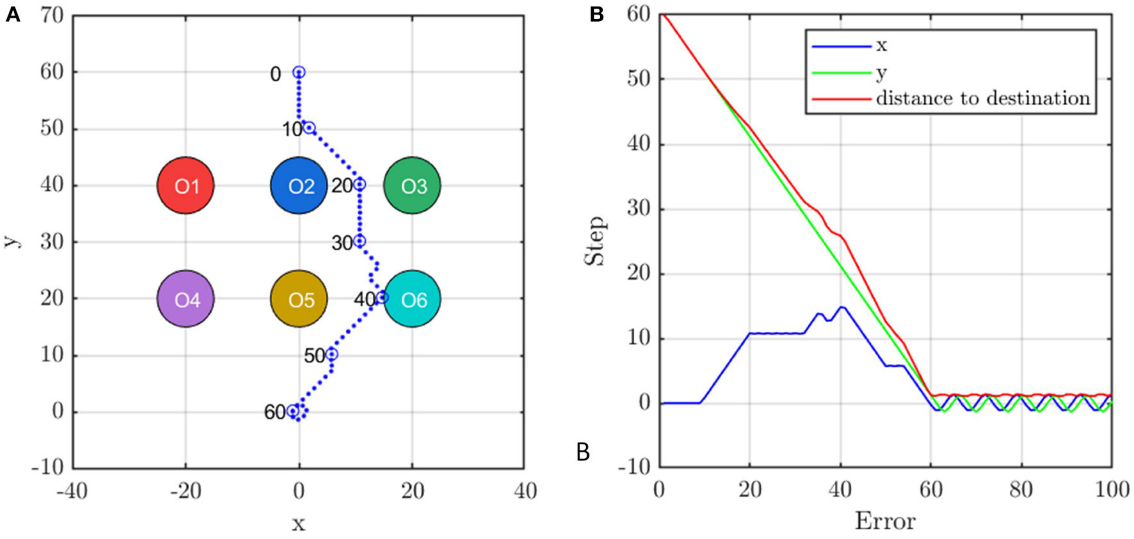


FIGURE 11 | The agent's **(A)** trajectory and **(B)** the distance to the destination during avoiding obstacles. The six colored circles are the collision zone of obstacles and the blue dots are the trace of the agent.

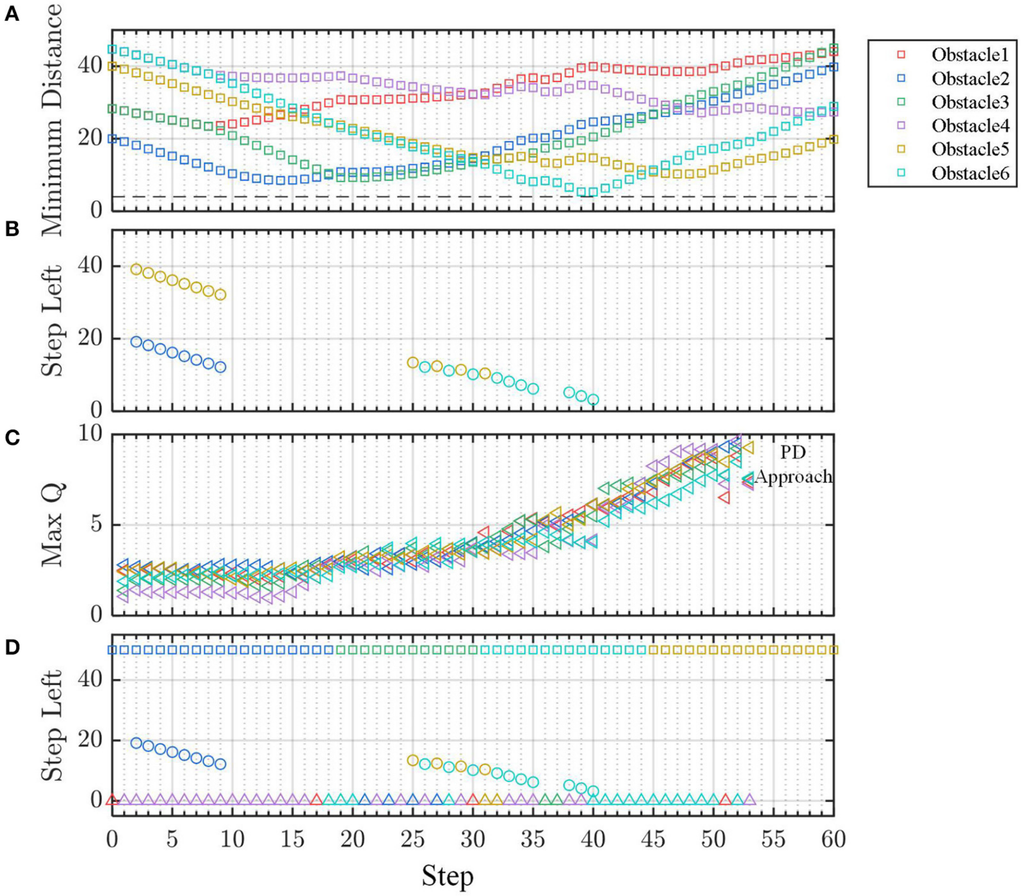


FIGURE 12 | The **(A)** distance, **(B)** prediction of left steps, **(C)** *Q* value of the fused policy for obstacles. **(D)** Comparison of three kinds of threat prediction.

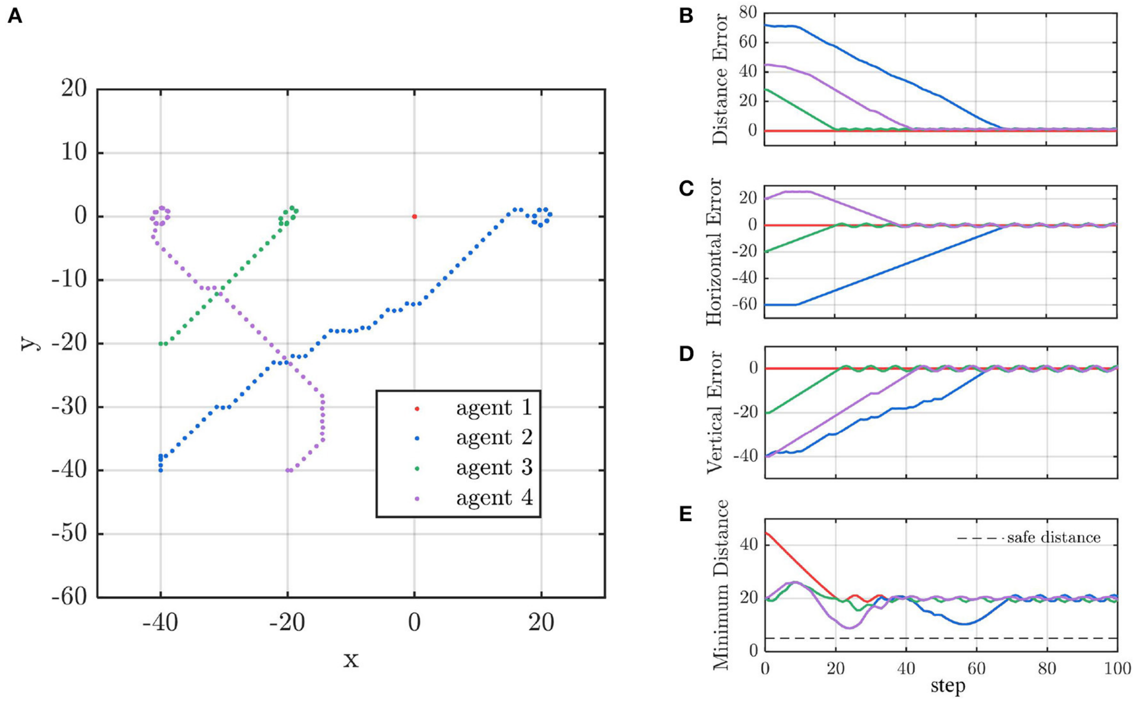


FIGURE 13 | (A) The agent's trajectory. **(B–D)** The agent's distance to destination and **(E)** the distance between agents.

is good; otherwise, it needs to be adjusted. In other words, the agent has enough time to find what works by trial and error, which is the advantage of the controller by learning. In the last demo, we present a line formation control scenario for four agents. Each agent is equipped with the DQN controller trained in section 4.1.1. We designate agent 1 as the leader and agent 2, 3, and 4 as followers. The expected target of the four agents are set as $[0, 0], [0, 20], [0, -20]$, and $[0, -40]$, respectively. In addition, their initial positions are set as $[0, 0], [0, 20], [0, -20]$, and $[0, -40]$ respectively, and their initial velocities are all zeros.

The trajectory of the agents is shown in **Figure 13A**. Agent 3 moves toward to its target position because there is no obstacle in its way. To avoid agent 2, agent 4 turns right and then turns back. Accordingly, agent 2 does not aim at its target position at the beginning to avoid collision with agent 4. Finally, the four agents form a linear formation. **Figures 13B–D** indicate that the distance between any two agents is more than the safe distance $d_{\text{safe}} = 5$, which validates the safety of our control method. As shown in **Figure 13E**, agent 2 changes its vertical speed instead of horizontal speed to avoid a collision. In the contrast, agent 4 adjusts its horizontal but not vertical speed.

5. CONCLUSION

Aiming at the problem of potential collision among agents in multi-agent formation control, an intelligent decomposition and fusion formation control method is proposed in this article. The multi-agent formation control is decomposed to the pair-wise unit formation control method where only one obstacle is considered. Then, the DQN controller for unit formation is trained following our episode mechanism design and reward shaping. Finally, by min-max fusion of all the pair-wise state-action values, the agent can first respond to the most likely threat among multiple obstacles without extra training. The demo of action field and unit formation control validates our unit formation DQN controller. The simulation results of avoiding obstacles and line formation show that our control method based on deep reinforcement learning can realize multi-agent formation with collision avoidance. In the future, obstacles with high dynamics can be taken into account, and the reward function can include optimal conditions like minimizing energy, and the fusion method also can be trained by the reinforcement learning method.

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

NX did research progress, simulation, and result in analysis and wrote the original draft with YH. LC supervised the work and revised this article. All authors contributed to the article and approved the submitted version.

REFERENCES

Chryssanthacopoulos, J. P. and Kochenderfer, M. J. (2011). "Decomposition methods for optimized collision avoidance with multiple threats," in *2011 IEEE/AIAA 30th Digital Avionics Systems Conference* (Seattle, WA), 1D2–1–1D2–11.

Eren, T., Anderson, B. D. O., Morse, A. S., Whiteley, W., and Belhumeur, P. N. (2003). Operations on rigid formations of autonomous agents. *Commun. Inf. Syst.* 3, 223–258. doi: 10.4310/CIS.2003.V3.N4.A2

Falconi, R., Sabattini, L., Secchi, C., Fantuzzi, C., and Melchiorri, C. (2011). A graph–based collision–free distributed formation control strategy. *IFAC Proc. Vol.* 44, 6011–6016. doi: 10.3182/20110828-6-IT-1002.02450

Guo, S., Li, Z., Niu, Y., and Wu, L. (2020). Consensus disturbance rejection control of directed multi-agent networks with extended state observer. *Chin. J. Aeronaut.* 33, 1486–1493. doi: 10.1016/j.cja.2019.07.018

Jin, X. (2019). Nonrepetitive leader-follower formation tracking for multiagent systems with LOS range and angle constraints using iterative learning control. *IEEE Trans. Cybern.* 49, 1748–1758.

Kamel, M. A., Yu, X., and Zhang, Y. (2020). Formation control and coordination of multiple unmanned ground vehicles in normal and faulty situations: a review. *Ann. Rev. Control* 49, 128–144. doi: 10.1109/TCYB.2018.2817610

Kuchar, J. K. and Yang, L. C. (2000). A review of conflict detection and resolution modeling methods. *IEEE Trans. Intell. Transp. Syst.* 1, 179–189. doi: 10.1016/j.arcontrol.2020.02.001

Lee, G., and Chwa, D. (2018). Decentralized behavior-based formation control of multiple robots considering obstacle avoidance. *Intell. Service Robot.* 11, 127–138. doi: 10.1007/s11370-017-0240-y

Li, D., Ge, S. S., He, W., Ma, G., and Xie, L. (2019a). Multilayer formation control of multi-agent systems. *Automatica* 109, 108558. doi: 10.1016/j.automatica.2019.108558

Li, S., Egorov, M., and Kochenderfer, M. J. (2019b). "Optimizing collision avoidance in dense airspace using deep reinforcement learning," in *13th USA/Europe Air Traffic Management Research and Development Seminar 2019, Vol. 3* (Vienna).

Liang, Y., Dong, Q., and Zhao, Y. (2020). Adaptive leader–follower formation control for swarms of unmanned aerial vehicles with motion constraints and unknown disturbances. *Chin. J. Aeronaut.* 33, 2972–2988.

Lillicrap, T., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., et al. (2016). "Continuous control with deep reinforcement learning," in *International Conference on Learning Representations* (San Juan).

Merheb, A., Gazi, V., and Sezer-Uzol, N. (2016). Implementation studies of robot swarm navigation using potential functions and panel methods. *IEEE/ASME Trans. Mechatron.* 21, 2556–2567.

Mnih, V., Heess, N., and Graves, A. (2014). "Recurrent models of visual attention," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., et al. (2013). Playing atari with deep reinforcement learning. *arXiv: Learning*.

Ng, A. Y., Harada, D., and Russell J., S. (1999). "Policy invariance under reward transformations: theory and application to reward shaping," in *Proceedings of the Sixteenth International Conference on Machine Learning ICML '99* (San Francisco, CA: Morgan Kaufmann Publishers Inc.), 278–287.

Oh, K. K., Park, M. C., and Ahn, H. S. (2015). A survey of multi-agent formation control. *Automatica* 53, 424–440. doi: 10.1016/j.automatica.2014.10.022

Sanz, Y., de Lope, J., and Martín H., J. A. (2008). "Applying reinforcement learning to multi-robot team coordination," in *Hybrid Artificial Intelligence Systems* (Berlin: Springer), 625–632.

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. (2016). "Prioritized experience replay," in *4th International Conference on Learning Representations, ICLR 2016 Conference Track Proceedings*, eds Y. Bengio, and Y. LeCun (San Juan).

Sui, Z., Pu, Z., Yi, J., and Xiong, T. (2019). "Formation control with collision avoidance through deep reinforcement learning," in *Proceedings of the International Joint Conference on Neural Networks* (Budapest), 1–8.

Sutton, R. S. and Barto, A. G. (1998). *Temporal-Difference Learning*. MIT Press. p. 133–160.

Van Den Berg, J., Guy, S. J., Lin, M., and Manocha, D. (2011). Reciprocal n-body collision avoidance. in *Springer Tracts in Advanced Robotics*, vol. 70 (Berlin: Springer), 3–19.

Wang, C. (2019). "A continuous actor-critic reinforcement learning approach to flocking with fixed-wing UAVs," in *Asian Conference on Machine Learning, eds W. S. L. Suzuki, and Taiji (Nagoya: Journal of Machine Learning Research)*, 64–79.

Wu, Z., Hu, G., Feng, L., Wu, J., and Liu, S. (2016). Collision avoidance for mobile robots based on artificial potential field and obstacle envelope modelling. *Assembly Autom.* 36, 318–332. doi: 10.1108/AA-01-2016-008

Xu, D., Zhang, X., Zhu, Z., Chen, C., and Yang, P. (2014). Behavior-based formation control of swarm robots. *Math. Problems Eng.* 2014, 205759. doi: 10.1155/2014/205759

Zhao, Z., Wang, J., Chen, Y., and Ju, S. (2020). Iterative learning-based formation control for multiple quadrotor unmanned aerial vehicles. *Int. J. Adv. Robot. Syst.* 17, 1–12. doi: 10.1177/1729881420911520

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.

ORIGINAL RESEARCH
published: 25 April 2022
doi: 10.3389/fnbot.2022.845858

Dynamic Video Image Segmentation Based on Dual Channel Convolutional Kernel and Multi-Frame Feature Fusion

Zuguo Chen [1,2,3], *Chaoyang Chen* [1,2] *and Ming Lu* [2]*

[1] Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, [2] School of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan, China, [3] CAS Key Laboratory of Human-Machine Intelligence-Synergy Systems, Shenzhen, China

The color image of the fire hole is key for the working condition identification of the aluminum electrolysis cell (AEC). However, the image of the fire hole is difficult for image segmentation due to the nonuniform distributed illuminated background and oblique beam radiation. Thus, a joint dual channel convolution kernel (DCCK) and multi-frame feature fusion (MFF) method is developed to achieve dynamic fire hole video image segmentation. Considering the invalid or extra texture disturbances in the edge feature images, the DCCK is used to select the effective edge features. Since the obtained edge features of the fire hole are not completely closed, the MFF algorithm is further applied to complement the missing portion of the edge. This method can assist to obtain the complete fire hole image of the AEC. The experiment results demonstrate that the proposed method has higher precision, recall rate, and lower boundary redundancy rate with well segmented image edge for the aid of working condition identification of the AEC.

Keywords: dynamic video image segmentation, dual channel convolution kernel, multi-frame feature fusion, fire hole, aluminum electrolysis cell

Edited by:
Chen Qiao,
Xi'an Jiaotong University, China

Reviewed by:
Xu Yang,
University of Science and Technology Beijing, China
Alireza Mousavi,
Brunel University London, United Kingdom
Nan Jiang,
East China Jiaotong University, China

Correspondence:
Ming Lu
mlu@hnust.edu.cn

Received: *30 December 2021*
Accepted: *14 March 2022*
Published: *25 April 2022*

Citation:
Chen Z, Chen C and Lu M (2022) Dynamic Video Image Segmentation Based on Dual Channel Convolutional Kernel and Multi-Frame Feature Fusion.
Front. Neurorobot. 16:845858.
doi: 10.3389/fnbot.2022.845858

1. INTRODUCTION

The aluminum electrolysis (AE) production process is a complex and continuous process, where any trivial failure in each step may affect the quality of the whole production process, resulting in poor uniformity of the products, low production efficiency, or extra energy resources consumption (Chen et al., 2021). It is known that the aluminum electrolysis cell (AEC) is the main production equipment in the AE production, which should be closely monitored online *via* the video as the currently adopted main measure. Then, image segmentation is the first essential dealt step to obtain visual features (Yue et al., 2020). The video image segmentation methods can be divided into the static feature and dynamic feature two categories (Bragantini et al., 2020).

The static features mainly include color, shape, contour, and texture. Wang proposes an Otsu image threshold segmentation method based on an improved particle swarm optimization (PSO) (Wang et al., 2019), where the inter-class variance of the Otsu is selected as the fitness function so as to increase the diversity of the particles with new particles supplement. Furthermore, a fast threshold image segmentation based on 2D fuzzy fisher and random local optimized quantum particle swarm optimization is proposed to reduce the redundant computation and improve the processing speed of the image segmentation (Zhang et al., 2016). Seyedhosseini and Tasdizen (2015) propose a semantic image segmentation based on the upper and lower hierarchical model to optimize the joint posterior probability. Dhanachandra and Chanu (2020) propose an image segmentation algorithm based on fuzzy c-means clustering algorithm (FCM) for noise image segmentation. However, it is rather difficult for static features to adapt to image segmentation with interference in complex scenes.

The dynamic features of the video images cannot be extracted with a single image but require to process continuous image frames consecutively in real-time. Karunanayake et al. (2020) has proposed a segmentation method based on multiple walking particles bouncing from the image edge to handling single or multiple objects characterized by a noisy background and broken boundaries. To balance the spatiotemporal coherence in scenes with deformation or large motion, a segmentation method is proposed based on the Markov chain model (Xixi and Chengmao, 2016). In terms of consecutive multi-frame images, an adaptive-domain network based on CycleGAN is proposed to improve the quality of the generated images in the feature space, making the translated images more informative for semantic segmentation (Cao et al., 2019). For moving object detection and segmentation, a feature extraction based adaptive neuro-fuzzy inference system (ANFIS) classifier is proposed (Guo et al., 2020), where the extracted features are trained and classified using the ANFIS classification module to improve the accuracy and recall rate of the image segmentation.

It is known that the fire hole images of the AEC are quite different under various working conditions, with high degree coupling between the target area and the background due to the background reflection, aerial fog, dust, and other interferences. Here, a dynamic video image segmentation method is proposed based on dual channel convolutional kernel (DCCK) and multi-frame feature fusion (MFF) to tackle these problems. The DCCK is proposed to select the effective edge features from the edge feature image. The core of the DCCK is a 4×4 convolution kernel which is used to obtain the edge image with smaller edges and interference texture. The other 6×6 convolutional kernel is used to obtain the salient edge with rough edges and interference texture. The multiplication of the two dealt results is applied to acquire the neat and smooth edge image. Since the edge features of the fire hole are not completely closed, an MFF algorithm is further developed to complement the missing portion by fusing two or more different frame images together.

The main contributions of the article are summarized as:

- A DCCK is proposed to select the effective edge features from the edge image *via* two different sized convolution kernels.
- A MFF method is developed to enhance the edge features and complement the missing edge portion by fusing more different frame images.
- Comparison experiments have been performed to demonstrate that the proposed method has higher mean pixel accuracy and lower boundary redundancy rate with satisfied image segmentation performance.

The remainder of the article is organized as follows. Section 2 presents the edge features obtainment based on the Prewitt operator and effective edge feature information selection based on DCCK. Section 3 develops a novel edge feature continuity processing algorithm based on MFF. Experiments are provided to verify the efficacy of the proposed method in Section 4. The conclusion is given in Section 5.

2. VIDEO IMAGE SEGMENTATION OF THE FIRE HOLE

When the working condition of the AEC changes, the fire hole features of the AEC will change. Accordingly, the fire hole features are the key criteria to identify the work conditions, while image segmentation is the first essential dealt step to obtain visual features of the fire hole.

2.1. Edge Feature Obtainment

The dealt image data are obtained from a 400KA aluminum electrolytic plant in Dengfeng City, Henan Province. The industrial cameras are installed to obtain the video data stream of the fire hole of the AEC. In order to obtain the object region in the fire hole image, edge features of the fire hole image should be extracted by the Prewitt operator before image segmentation. The RGB image, gray image, and the obtained edge feature image with the Prewitt operator of the fire hole are shown in **Figure 1**.

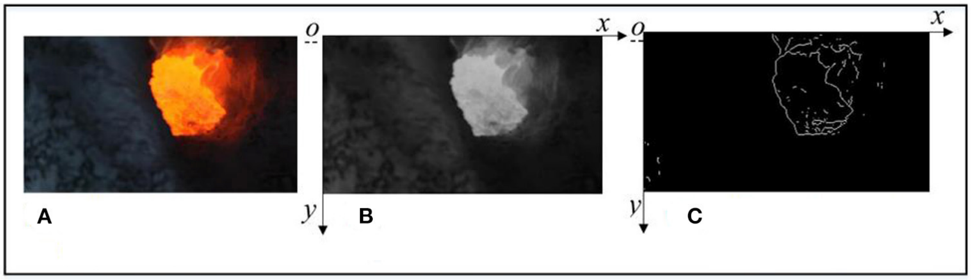
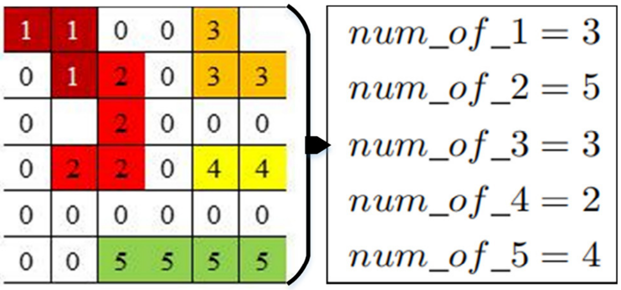


FIGURE 1 | The obtained edge feature images (Prewitt operator). **(A)** The original image of the fire hole. **(B)** The gray image of the fire hole. **(C)** The obtained edge feature image.

2.2. Effective Edge Features Selection Based on DCCK

The obtained edge feature image would contain a large number of invalid edge features of the fire hole. In order to acquire effective edge features of the fire hole, the convolutional neural network (CNN) is applied to deal with the edge features of the fire hole (Bui et al., 2019). With the different sizes of the convolutional kernels, the edge image can be dealt to connect the closer pixels and add the cover area of the edge. The sizes of the convolutional kernels are selected based on human experience with the trial and error method. In this article, a 4×4 convolutional kernel and another 6× 6 convolutional kernel are selected to deal with the edge feature image, where the pixel values (x, y) at the convolutional kernels are set as $f_1(x, y)$ and $f_2(x, y)$. As seen on the left side of Equation (1), the selected 4×4 convolutional kernel sk_1 is used to filter out the detailed images so as to connect scattered points of the image as much as possible. Moreover, the 6×6 convolution kernel sk_2 is written on the right side of Equation (1),

$$sk_1 = \begin{bmatrix} 0&0&0&0 \\ 0&1&1&0 \\ 0&1&1&0 \\ 0&0&0&0 \end{bmatrix}; \quad sk_2 = \begin{bmatrix} 0&0&0&0&0&0 \\ 0&1&1&1&1&0 \\ 0&1&1&1&1&0 \\ 0&1&1&1&1&0 \\ 0&1&1&1&1&0 \\ 0&0&0&0&0&0 \end{bmatrix} \tag{1}$$



1	1	0	0	3	
0	1	2	0	3	3
0		2	0	0	0
0	2	2	0	4	4
0	0	0	0	0	0
0	0	5	5	5	5

$num\_of\_1 = 3$
$num\_of\_2 = 5$
$num\_of\_3 = 3$
$num\_of\_4 = 2$
$num\_of\_5 = 4$

FIGURE 2 | The image processed with different convolutional kernels.

For the purpose of expending edge width to enlarge the range of the boundary connection, such dual kernel operation can obtain more edge breakpoints to be closed or connected. After being processed with the 4×4 convolutional kernel of the edge image, the number of all continuous image blocks can be calculated, and each pixel in the block is labeled, as shown in the left graph of **Figure 2**.

Next, the number of each non-zero digital tag is calculated in the continuous pixel labeling. In **Figure 2**, $num\_of\_i, (i = 1, 2, \cdots, 5)$ is the number of different digital tags in the box, which can easily obtain the number of pixels in each block, and removes pixel blocks with fewer pixels less than the $threshold_1$ (set as n_l) and turn their label as 0. After filtering, the unfiltered label is turned to 1 again (set as N_g), written as,

$$\begin{aligned} [1] num\_of\_n_l \leq threshold_1 \rightarrow f(x_j, y_j) = 0 \\ else f(x_k, y_k) = 1 \end{aligned} \tag{2}$$

where $f(x_j, y_j) = num\_of\_n_l$ is the pixel number in the pixel block n_l. l is the serial number of the pixel block; j is the serial number of the pixel point less than the $threshold_1$. N_g is the label of the pixel block; g is the serial number of the pixel point greater than the $threshold_1$. After processing by the 4×4 convolutional kernel, the obtained image is shown in **Figure 3A**. The image processing procedure *via* the 6×6 convolutional kernel is similar to that of the 4×4 convolutional kernel, and the processed image with the 6×6 convolutional kernel is shown in **Figure 3B**.

Afterward, the corresponding positions of the processed images with the 4×4 convolutional kernel $f_1(x, y)$ and the 6×6 convolutional kernel $f_2(x, y)$ are multiplied, where the multiplication process of the pixel values is shown in **Table 1** and the resulted image is depicted in **Figure 3C**. It can be seen from **Figure 3C** that there is a relatively complete

TABLE 1 | The multiplication process of the pixel values.

$f_1(x,y)$	$f_2(x,y)$	$f_1(x,y) \times f_2(x,y)$
0	0	0
0	1	0
1	0	0
1	1	1

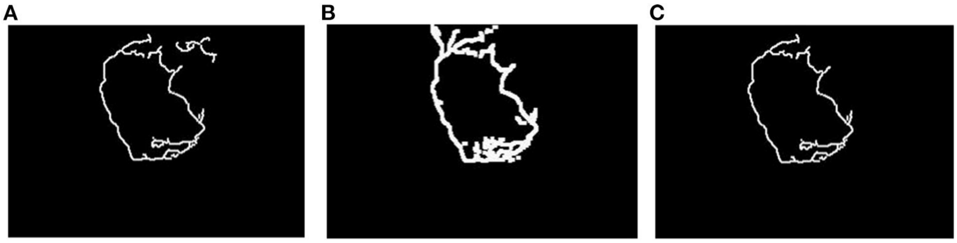


FIGURE 3 | The illustration of the continuous pixel labeling. **(A)** The image processed with a 4 × 4 convolutional kernal. **(B)** The image processed with a 6 × 6 convolutional kernal. **(C)** The processed image with double channel convolutional kernal.

edge image containing little texture features. However, the edge is not completely enclosed for each frame image, which is unsuitable for the image obtainment of the fire hole. Hence, the edge feature continuity has to be further processed.

3. EDGE FEATURE CONTINUITY PROCESSING BASED ON MFF

Since it is difficult to acquire a closed edge image for each frame image with large impurities to be filtered from only edge

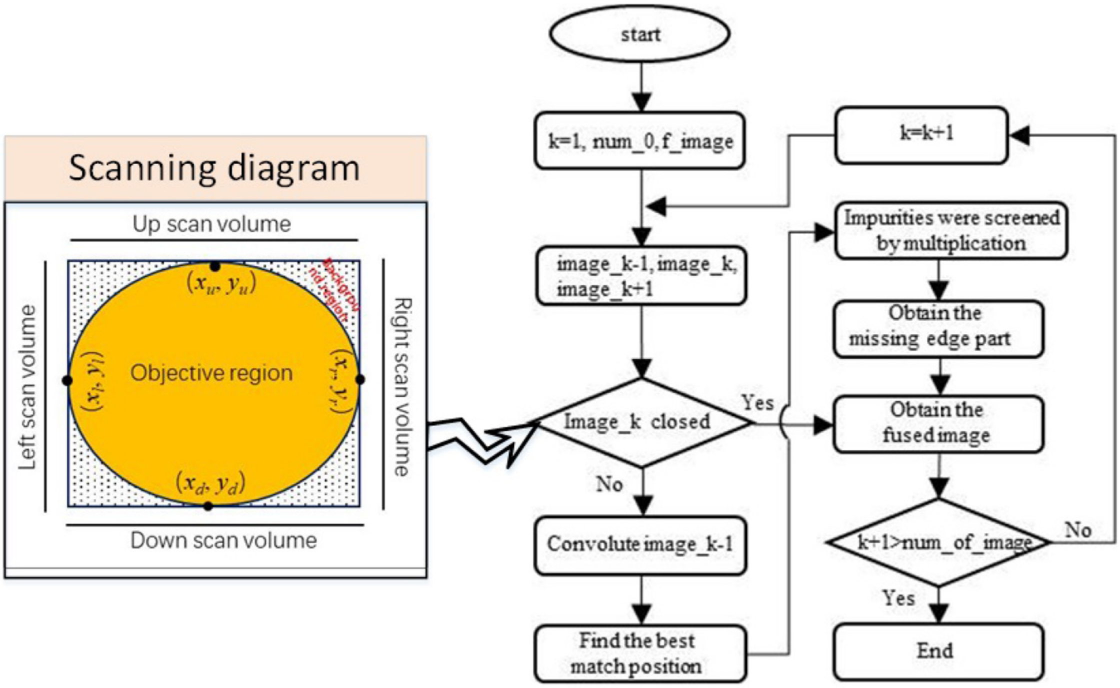


FIGURE 4 | The flow chart of the edge feature continuity processing.

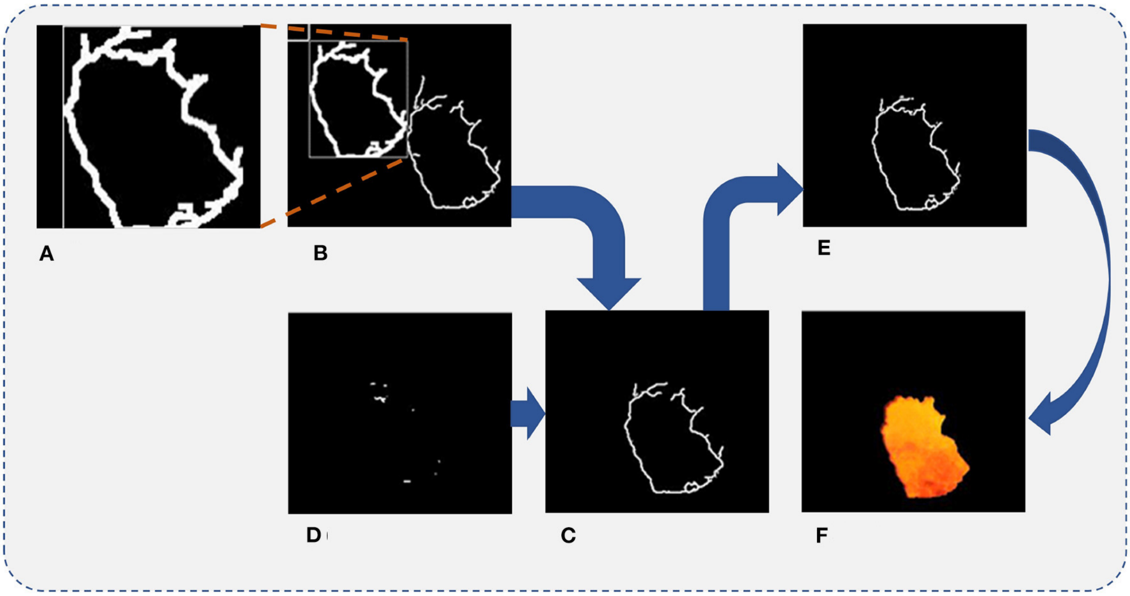


FIGURE 5 | Final separated result of the fire hole. **(A)** The processed previous frame edge image. **(B)** Matching diagram of the edge image. **(C)** The overlapped part of the two images. **(D)** The missing part of the next frame image. **(E)** The continuous edge feature image. **(F)** Final separated result of the fire hole.

feature selection, an MFF method is proposed to complement the missing portion of the edge. The flow chart of the MFF method is illustrated in **Figure 4**.

In order to determine whether the previous frame edge image has formed a complete edge pattern, a scanning algorithm (from four directions: up, down, left, and right) is adopted (refer to the left graph in **Figure 4**). If the edge feature image is closed, its label is marked as 1, and the frame is used as the reference edge feature image to mend the next frame of the edge feature image. Otherwise, the fused edge image of the previous frame with a complete edge pattern is used as the reference to complement the next frame edge image. When two frame edge images are fused, the previous frame of the edge image has first to be expanded by the convolutional operation to retain as many edge features as possible from the previous edge image (refer to **Figure 5A**).

The processed previous frame edge images by the convolutional operation are marked as $image_{(k-1)}\_handled$. To find the optimal matching position, the $image_{(k-1)}\_handled$ edge image moves to the next frame edge image to find the optimal matching position of the next frame edge feature, the matching diagram is shown in **Figure 5B**. The object function is designed to determine the optimal matching position, written as,

$$
\begin{aligned}
max(\sum_{i=1}^{im_{(k-1)}\_r} \sum_{j=1}^{im_{(k-1)}\_c} part(i_{k-1}, j_{k-1}) \times part(i_k, j_k) \\
i = 1, 2, \cdots, im_{(k-1)}\_r - im_{(k)}\_r \\
j = 1, 2, \cdots, im_{(k-1)}\_c - im_{(k)}\_c
\end{aligned}
\tag{3}
$$

where the $im_{(k-1)}\_r$ and $im_{(k-1)}\_c$ are the pixel row and column number of the $image_{(k-1)}$. The $im_{(k)}\_r$ and $im_{(k)}\_c$ are the pixels row and column number of the next frame edge image. The $part(i_{k-1}, j_{k-1})$, $part(i_k, j_k)$ are the pixel value of the $image_{(k-1)}$ and the next frame edge image $image_{(k)}$.

After obtaining the optimal matching position, the overlapped region of the two images is obtained by multiplying each pixel of the $image_{(k-1)}$ and $image_{(k)}$, while other positions are filled with 0. The image is marked as $Im\_d_{(k)}$, as shown in **Figure 5C**.

The $Im\_d_{(k)}$ is dealt with by the convolutional operator to extract the effective part for optimal matching by subtracting the pixels of the $Im\_d_{(k)}$ from the effective edge pixels at the corresponding position of $Im\_d_{(k-1)}$, and the corresponding subtraction relationship is shown in **Table 2**. Thus, the missing part of the $Im\_d_{(k)}$ is obtained and marked as $Im\_sup_{(k)}$, as shown in **Figure 5D**.

Finally, the edge feature image of the $Im_{(k+1)}$ is superimposed by adding the pixel values in the same position of the $Im\_d_{(k)}$ and the $Im\_sup_{(k)}$. The corresponding relationship of the superimpose processing is shown in **Table 3**, and the continuous edge feature image is shown in **Figure 5E**.

In order to fill the image, the interior of the edge feature should be completely filled using the white color, and the two-dimensional binary image can be converted into a three-dimensional RGB image (Guan et al., 2019). To remain the color feature of the separated fire image as the same as the original image, the three-dimensional RGB image is merged with the original image, as shown in **Figure 5F**. It can be seen from the final dealt result that the dynamic video image segmentation method based on DCCK and MFF can effectively segment the image and obtain a completely smooth edge image without much impurity interference.

TABLE 2 | The corresponding relationship of the substract processing.

$Im_{(k-1)}(x,y)$	$Im_{(k)}(x,y)$	Substract	Standardization
0	0	0	0
0	1	-1	0
1	0	1	1
1	1	0	0

TABLE 3 | The corresponding relationship of the superimpose processing.

$Im\_d_{(k)}(x,y)$	$Im\_sup_{(k)}(x,y)$	Superimpose result	Standardization
0	0	0	0
0	1	1	1
1	0	1	1
1	1	2	1

4. EXPERIMENTAL RESULTS AND ANALYSIS

Three indicators(precision, recall rate, and F1-Measure) are used to evaluate the segmentation effect, and precision represents the proportion of samples identified as positive categories that are indeed positive categories. Recall rate represents the proportion of all positive class samples that are correctly identified. F1-Measure represents the harmonic average evaluation index of precision and recall rate.

The Roberts operator (Albdour and Zanoon, 2020) detects edge lines by local difference calculation and is often used to process low-noise images with steepness. The edge localization effect of the Sobel operator (Zhou and Liu, 2019) is good, and it is better for image processing with more noise, but the detected edge is prone to multi-pixel width. The Prewitt operator (Song et al., 2019) has a better effect on the image edge extraction of grayscale gradient and does not consider the influence of the distance of adjacent points on the current pixel point. Compared with other classical edge detection operators, the Canny operator (Bu et al., 2019) has higher accuracy, detects finer edges and requires more computation, and is the most representative edge detection operator.

In order to verify the effectiveness of the proposed method, the Roberts operator, Prewitt operator, Sobel operator, Canny operator, fuzzy-Sobel operator (Sivaranjani and Kalaiselvi, 2021), and bilateral filter based Canny operator (Zhang et al., 2019) are used for comparison. The experimental results are shown in **Figure 6**, and it can be seen that most methods are unable to segment the closed image edge completely.

When the Canny operator is used to obtain the edge images of the fire hole, most of the noise can be removed, but the obtained edge images are discontinuous with less remarkable edge lines. The fuzzy Sobel operator can filter out most of

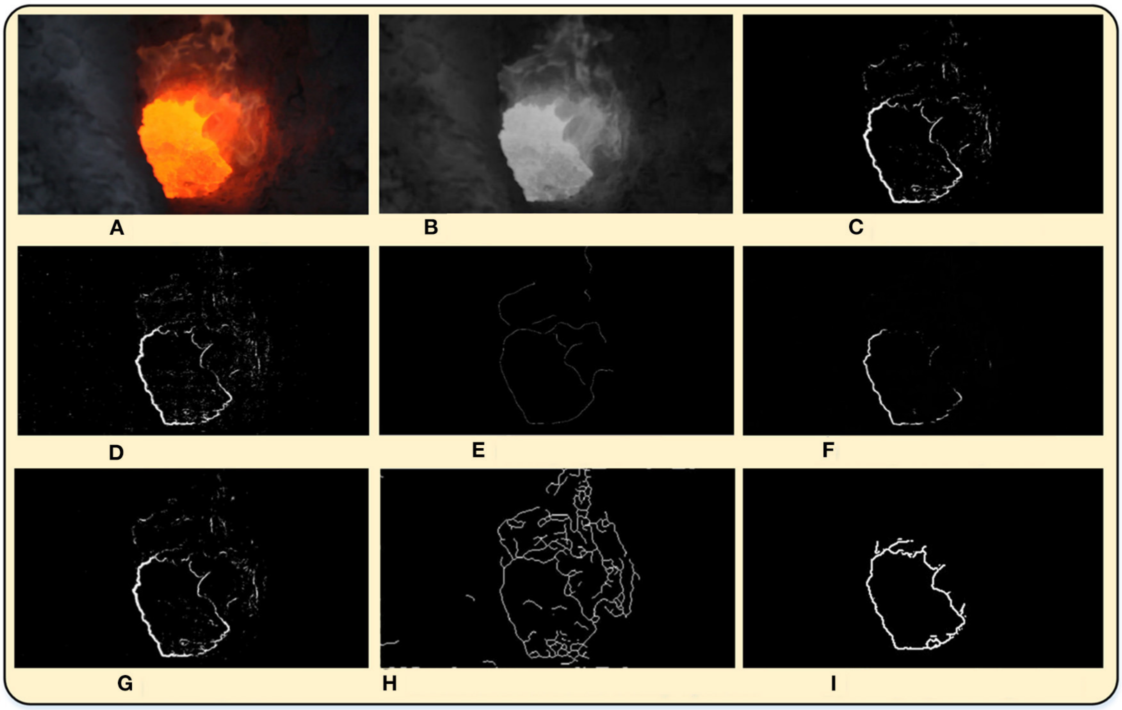


FIGURE 6 | The segmentation results with different algorithms for comparison. **(A)** Original image. **(B)** Gray-scale map image. **(C)** Prewitt operator. **(D)** Roberts operator. **(E)** Canny operator. **(F)** Funny-sobel operator. **(G)** Sobel operator. **(H)** Bilateral filter based Canny operator. **(I)** The proposed algorithm.

TABLE 4 | The evaluation indexes of the image segmentation with different algorithms comparison.

Evaluation index Algorithms	Precision	Recall rate	F1-Measure
Canny operator	0.0429	0.0598	0.0179
Bilateral filter based Canny operator	0.1676	0.2851	0.8554
Fuzzy sobel operator	0.4151	0.6530	1.9591
Roberts operator	0.4430	0.5464	1.6391
Prewitt operator	0.5363	0.5962	1.7886
Sobel operator	**0.6102**	0.5613	1.6839
The proposed algorithm	0.5132	**0.8554**	**2.5663**

Bold value means that these values are the best.

the noise of the edge images and the dust textures, however, the edge images of the fire hole still have a lot of missing parts. While the bilateral filtering based Canny operator can obtain the complete edge images, many invalid textures still remain. As the proposed DCCK joint MFF method can not only remove most of the noise of the edge images of the fire hole but also can obtain continuous and complete edge images of the fire hole. Therefore, the performance of the proposed method is significantly higher than those of traditional algorithms in the processing of the segmentation of the fire-hole of the AEC. The evaluation indexes (precision, recall rate, F1-Measure) of the image segmentation are calculated as well, listed in **Table 4**. The recall rate and F1-Measure of the obtained edge image using the proposed algorithm are the best among these algorithms. Although the precision of the obtained edge image with the proposed algorithm is slightly lower than that of the Prewitt and Sobel operator, it can be ignored without causing performance degradation in practice. In all, the proposed algorithm ranks highest comprehensively compared with other image segmentation algorithms. Compared with other methods of image segmentation, the algorithm of this article has a remarkable improvement in applicability and general utilization.

5. CONCLUSION

This article proposes a dynamic video image segmentation method based on the DCCK joint MFF algorithms to segment the images of the fire hole of the AEC. The Prewitt operator is first used to extract the edge features of the fire image. Due to the extra texture in the edge feature image, the

DCCK is proposed to select the effective edge features. Then the MFF algorithm is further proposed to complement the missing portion of the edge image. Finally, the performance of the proposed method is verified with a comparison of other segmentation methods in dealing with images of the fire hole AEC under heavy dust and complex background interference. Compared with the conventional method of image segmentation, the proposed method has high precision with wide applicability.

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/s.

AUTHOR CONTRIBUTIONS

ZC determined the research program and writed the article. The experimental data were analyzed by CC. The theory of the model was given and analyzed by ML. All authors contributed to the article and approved the submitted version.

FUNDING

The research has been supported by grants from the China Postdoctoral Science Foundation (Ref. 2020M672890), the National Natural Science Foundation of China (Ref. 61903137), the Natural Science Foundation Of Hunan Province (Ref. 2020JJ5201), and the Shenzhen Basic Research Program (Ref. Jcyj20170818153635759).

REFERENCES

Albdour, N., and Zanoon, N. (2020). A steganographic method based on roberts operator. *Jordan J. Elect. Eng.* 6, 266. doi: 10.5455/jjee.204-1583873433

Bragantini, J., Moura, B., Falcao, A. X., and Cappabianco, F. A. (2020). Grabber: a tool to improve convergence in interactive image segmentation. *Pattern Recognit. Lett.* 140, 267–273. doi: 10.1016/j.patrec.2020.10.012

Bu, C., Sun, Z., Tang, Q., Liu, Y., and Mei, C. (2019). Thermography sequence processing and defect edge identification of tbc structure debonding defects detection using long-pulsed infrared wave non-destructive testing technology. *Russian J. Nondestruct. Test.* 55, 80–87. doi: 10.1134/S1061830919010030

Bui, Q.-T., Pham, M. V., Nguyen, Q.-H., Nguyen, L. X., and Pham, H. M. (2019). Whale optimization algorithm and adaptive neuro-fuzzy inference system: a hybrid method for feature selection and land pattern classification. *Int. J. Rem. Sens.* 40, 5078–5093. doi: 10.1080/01431161.2019.1578000

Cao, Y., Sun, L., Han, C., and Guo, J. (2019). Video segmentation scheme based on amc. *IET Image Process.* 14, 407–416. doi: 10.1049/iet-ipr.2018.6659

Chen, Z., Lu, M., Zhou, Y., and Chen, C. (2021). Information synergy entropy based multi-feature information fusion for the operating condition identification in aluminium electrolysis. *Inf. Sci.* 548, 275–294. doi: 10.1016/j.ins.2020.07.031

Dhanachandra, N., and Chanu, Y. J. (2020). An image segmentation approach based on fuzzy c-means and dynamic particle swarm optimization algorithm. *Multimedia Tools Appl.* 79, 18839–18858. doi: 10.1007/s11042-020-08699-8

Guan, C., Wang, S., and Liew, A. W.-C. (2019). Lip image segmentation based on a fuzzy convolutional neural network. *IEEE Trans. Fuzzy Syst.* 28, 1242–1251. doi: 10.1109/TFUZZ.2019.2957708

Guo, X., Wang, Z., Yang, Q., Lv, W., Liu, X., Wu, Q., et al. (2020). Gan-based virtual-to-real image translation for urban scene semantic segmentation. *Neurocomputing* 394, 127–135. doi: 10.1016/J.NEUCOM.2019.01.115

Karunanayake, N., Aimmanee, P., Lohitvisate, W., and Makhanov, S. S. (2020). Particle method for segmentation of breast tumors in ultrasound images. *Math. Comput. Simulat.* 170, 257–284. doi: 10.1016/j.matcom.2019.10.009

Seyedhosseini, M., and Tasdizen, T. (2015). Semantic image segmentation with contextual hierarchical models. *IEEE Trans. Pattern Anal. Mach. Intell.* 38, 951–964. doi: 10.1109/TPAMI.2015.2473846

Sivaranjani, B., and Kalaiselvi, C. (2021). Sobel operator and pca for nearest target of retina images. *ICTACT J. Image Video Process.* 11, 2483–2491. doi: 10.21917/ijivp.2021.0353

Song, Y., Ma, B., Gao, W., and Fan, S. (2019). Medical image edge detection based on improved differential evolution algorithm and prewitt operator. *Acta Microscopica* 28, 30–39. Available online at: https://web.s.ebscohost.com/abstract?site=ehost&scope=site&jrnl=07984545&AN=137805076&h=fcVP4QGR%2b8HqjEIz55NLYTkMFShGfHSBaysqvKLksLFrf4Rg826RoH%2b6GHgiF2evcEMDAnTp7oL9EZ3lw%2fv2UA%3d%3d&crl=c&resultLocal=ErrCrlNoResults&resultNs=Ehost&crlhashurl=login.aspx%3fdirect%3dtrue%26profile%3dehost%26scope%3dsite%26authtype%3dcrawler%26jrnl%3d07984545%26AN%3d137805076

Wang, C., Yang, J., and Lv, H. (2019). "Otsu multi-threshold image segmentation algorithm based on improved particle swarm optimization," in *2019 IEEE 2nd International Conference on Information Communication and Signal Processing (ICICSP)* (Weihai: IEEE), 440–443.

Xixi, C., and Chengmao, W. (2016). Neutrosophic c-means clustering in kernel space and its application in image segmentation. *J. Image Graph.* 21, 1316–1327. doi: 10.11834/jig.20161006

Yue, W., Gui, W., and Xie, Y. (2020). Experiential knowledge representation and reasoning based on linguistic petri nets with application to aluminum electrolysis cell condition identification. *Inf. Sci.* 529, 141–165. doi: 10.1016/j.ins.2020.03.079

Zhang, B., Shi, Y., and Gu, S. (2019). Narrow-seam identification and deviation detection in keyhole deep-penetration tig welding. *Int. J. Adv. Manuf. Technol.* 101, 2051–2064. doi: 10.1007/s00170-018-3089-0

Zhang, C., Xie, Y., Liu, D., and Wang, L. (2016). Fast threshold image segmentation based on 2d fuzzy fisher and random local optimized qpso. *IEEE Trans. Image Process.* 26, 1355–1362. doi: 10.1109/TIP.2016.2621670

Zhou, R.-G., and Liu, D.-Q. (2019). Quantum image edge extraction based on improved sobel operator. *Int. J. Theor. Phys.* 58, 2969–2985. doi: 10.1007/s10773-019-04177-6

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.

frontiers | Frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 22 July 2022
doi: 10.3389/fnbot.2022.820389

Non-Fragile Observer-Based Adaptive Integral Sliding Mode Control for a Class of T-S Fuzzy Descriptor Systems With Unmeasurable Premise Variables

Xiaorong Huang[1], *Anca L. Ralescu*[2], *Yiqiang Peng*[1], *Hongli Gao*[3] *and Shulei Sun*[1]*

[1] Vehicle Measurement, Control and Safety Key Laboratory of Sichuan Province, School of Automobile and Transportation, Xihua University, Chengdu, China, [2] Department of Electrical Engineering and Computer Science, University of Cincinnati, Cincinnati, OH, United States, [3] School of Mechanical Engineering, Southwest Jiaotong University, Chengdu, China

The issue of non-fragile observer-based adaptive integral sliding mode control for a class of Takagi–Sugeno (T-S) fuzzy descriptor systems with uncertainties and unmeasurable premise variables is investigated. General nonlinear systems are represented by nonlinear T-S fuzzy descriptor models, because premise variables depend on unmeasurable system states and fuzzy models have different derivative matrices. By introducing the system state derivative as an auxiliary state vector, original fuzzy descriptor systems are transformed into augmented systems for which system properties remain the same. First, a novel integral sliding surface, which includes estimated states of the sliding mode observer and controller gain matrices, is designed to obtain estimation error dynamics and sliding mode dynamics. Then, some sufficient linear matrix inequality (LMI) conditions for designing the observer and the controller gains are derived using the appropriate fuzzy Lyapunov functions and Lyapunov theory. This approach yields asymptotically stable sliding mode dynamics. Corresponding auxiliary variables are introduced, and the Finsler's lemma is employed to eliminate coupling of controller gain matrices, observer gain matrices, Lyapunov function matrices, and/or observer gain perturbations. An observer-based integral sliding mode control strategy is obtained to assure that reachability conditions are satisfied. Moreover, a non-fragile observer and a non-fragile adaptive controller are developed to compensate for system uncertainties and perturbations in both the observer and the controller. Finally, example results are presented to illustrate the effectiveness and merits of the proposed method.

Keywords: T-S fuzzy descriptor systems, non-fragile control, observer design, integral sliding mode control, nonparallel distributed compensation control, unmeasurable premise variables

OPEN ACCESS

Edited by:
Yimin Zhou,
Shenzhen Institutes of Advanced Technology (CAS), China

Reviewed by:
Hoda Moodi,
Quchan University of Advanced Technology, Iran
Yong Xu,
Guangdong University of Technology, China

****Correspondence:***
Shulei Sun
shuleisun@foxmail.com

Received: *23 November 2021*
Accepted: *20 June 2022*
Published: *22 July 2022*

Citation:
Huang X, Ralescu AL, Peng Y, Gao H and Sun S (2022) Non-Fragile Observer-Based Adaptive Integral Sliding Mode Control for a Class of T-S Fuzzy Descriptor Systems With Unmeasurable Premise Variables. Front. Neurorobot. 16:820389. doi: 10.3389/fnbot.2022.820389

INTRODUCTION

In recent decades, control synthesis, stability analysis, and observer design for nonlinear systems have received important consideration because of their wide application in practice, and demands for reliability and performance have been increasingly enhanced. Nevertheless, it has become challenging to systematically design and analyze such systems. The Takagi–Sugeno (T-S) fuzzy system, also known as the type III fuzzy model, proposed by Japanese scholars in 1985 (Takagi and Sugeno, 1985) provides a general approach to approximate any smooth nonlinear system with an arbitrary degree of accuracy but without complex mathematical equations. Through the use of the T-S fuzzy model approach, systematic analysis and synthesis of nonlinear systems can be performed based on classical control theory (Lv et al., 2019), modern control theory (Zhang Z. et al., 2019), and intelligent control theory (Sun et al., 2007; Cervantes et al., 2016). Due to their strong approximation capabilities and good tolerance to uncertainty and imprecision, T-S fuzzy control techniques have been widely used in the area of intelligent control of robotics, i.e., for robot manipulators (Fan et al., 2020), nonlinear flexible link robots (Shams and Seyedtabaii, 2020), and planar parallel robots (Vermeiren et al., 2012) among others. Therefore, the T-S fuzzy model is an effective intelligent method for modeling and analyzing robotic systems.

In practical systems, all state variables cannot always be measured by sensors, and in other cases, the sensors used to measure system states are expensive; however, the state variables are indispensable to the design of system controllers and analyses of system stability. Thus, observer design plays an important role in stability analysis and control synthesis for nonlinear systems. A fuzzy observer was first proposed by Tanaka and Sano in 1994 (Tanaka and Sano, 1994), and observers have since received extensive research attention. Various problems involving fuzzy observers have been studied in the literature. The researchers in (Tong and Li, 2002; Asemani and Majd, 2013) studied an observer-based controller design, and the observer and controller were built simultaneously. The sliding mode fuzzy observers in (Shen et al., 2011) were designed to address the problem of fault-tolerant control for T-S fuzzy systems with actuator faults. Based on the Lyapunov method, sufficient conditions for an unknown input T-S observer (Chadli and Karimi, 2012) were given in a linear matrix inequality (LMI) formulation. The novel fuzzy learning observer in (You et al., 2019) was constructed to achieve simultaneous reconstruction of system states and actuator faults for T-S fuzzy systems with time-varying delays. However, the above methods for T-S fuzzy systems are difficult to implement in practice because of their high complexity. Compared with T-S fuzzy systems, T-S fuzzy descriptor systems have the following advantages: they effectively reduce the number of fuzzy rules in a nonlinear system, and they can describe a wider variety of nonlinear systems. Hence, the observer design problem for the traditional nonlinear T-S descriptor system is considered in this study.

In Li and Zhang (2018), Zhang et al. (2018), Zhang J. et al. (2019), the authors designed a reduced-order robust observer, a robust adaptive sliding mode observer, and a robust H∞ sliding mode observer; additionally, an observer-based sliding mode controller was proposed for T-S fuzzy descriptor systems with time-varying delay. Observer-based integral sliding mode control strategies were developed in (Li et al., 2018). An adaptive fuzzy observer in (Kharrat et al., 2018), a novel fuzzy descriptor learning observer in (Jia et al., 2015), a robust fuzzy descriptor observer in (Brahim et al., 2017), and some T-S descriptor observers in (López-Estrada et al., 2017; Haj Brahim et al., 2019) were constructed to achieve simultaneous reconstruction of system states and actuator/sensor faults. A fault-tolerant control scheme was derived based on Lyapunov asymptotic stability. Robust observer-based output feedback control and the robustness issue were addressed in (Liu et al., 2013) to avoid control performance deterioration or instability due to disturbances or approximation errors in the system. The design process of a T-S fuzzy observer was extended to a class of T-S descriptor systems with unmeasurable premise variables in (Soulami et al., 2015). When designing a fuzzy observer, it is important to have a clear understanding of the relationship between premise variables and estimated system states. From the literature (Liu et al., 2013; Jia et al., 2015; Soulami et al., 2015; Brahim et al., 2017; López-Estrada et al., 2017; Kharrat et al., 2018; Li and Zhang, 2018; Li et al., 2018; Zhang et al., 2018; Haj Brahim et al., 2019; Zhang J. et al., 2019), we can conclude that two cases exist to describe this relationship. Case A: premise variables are not dependent on system states estimated by the fuzzy observer, and case B: premise variables depend on system states estimated by the fuzzy observer. The design process of the controller and the observer is relatively simple in case A compared to the approach in case B; therefore, more studies have focused on case A such as (Liu et al., 2013; Jia et al., 2015; Kharrat et al., 2018; Li and Zhang, 2018; Li et al., 2018; Zhang et al., 2018; Zhang J. et al., 2019). However, it should be noted that premise variables usually depend on unmeasurable system states in practical systems. Therefore, case A has more restrictive conditions that limit its application to various systems. Some researchers in (Soulami et al., 2015; Brahim et al., 2017; López-Estrada et al., 2017; Haj Brahim et al., 2019) studied the observer design for T-S fuzzy descriptor systems with unmeasurable premise variables.

Sliding mode control (SMC), as a type of variable structure control method, can effectively control systems with nonlinearities and uncertainties because of its beneficial characteristics such as fast response, good transience, and strong robustness. In recent decades, many significant results (Vu et al., 2012; Van et al., 2013; Li and Zhang, 2018; Zhang et al., 2018; Wu et al., 2019; Zhang J. et al., 2019) based on fuzzy systems have been reported in studies on sliding mode observer design and observer-based SMC. A novel fuzzy second-order sliding mode observer was designed to estimate robot velocity, and a new fuzzy second-order sliding mode strategy based on T-S fuzzy models was proposed to track the expected motion in (Van et al., 2013). A T-S fuzzy-model-based sliding mode controller was developed for surface-mounted permanent-magnet synchronous motors in Vu et al. (2012) considering motor parameter uncertainties and unknown external noise. In (Wu et al., 2019), the disturbance in T-S fuzzy discrete time systems was monitored by a disturbance observer. The core objective of a sliding mode observer is to design an SMC strategy for an observer system or a dynamic estimation error system. Hence, a sliding mode observer designed with the SMC method has high robustness to nonlinearities and uncertainties. From the above literature, it is clear that the traditional SMC theory requires a reaching phase to drive state trajectories to the desired sliding surface. The integral sliding mode technique can eliminate this process by implementing sliding mode motion from the initial time of the control action. Consequently, the matched uncertainties can be compensated for throughout the integral sliding mode control (ISMC) process. Therefore, the issue of applying an integral sliding mode technique to design observers and controllers for fuzzy systems has received significant attention, such as in (Jiang et al., 2018; Li et al., 2018; Kuppusamy and Joo, 2019). In (Jiang et al., 2018), a novel integral sliding surface function was proposed for the observer space of T-S fuzzy systems with semi-Markov switching and immeasurable premise variables. An integral-type fuzzy switching surface function was defined that simultaneously involved a state-dependent input matrix and a memory parameter in Kuppusamy and Joo (2019).

The above methods of controller and/or observer design for T-S descriptor systems are based on an implicit assumption that a desired controller and/or observer can be realized exactly. However, in practical applications, it is impossible to implement an ideal designed controller or observer because of round-off errors in numerical computations, digital-to-analog conversion errors, the finite word length used in digital computer systems, and other factors. Therefore, a significant issue is determining how to design a controller and an observer that are able to tolerate some uncertainties in various processes, and is called non-fragile control. The problem of non-fragile controller design has been addressed, and a non-fragile guaranteed cost controller (Chen and Li, 2013), a non-fragile fuzzy dissipative static output feedback control (Guan and Liu, 2016), and a non-fragile robust H∞ control (Zhang et al., 2007) have been investigated. The researchers in (Li X. et al., 2017; Duan et al., 2019) focused on the issue of non-fragile observer design.

Although a considerable effort has been devoted to fuzzy observer analysis, ISMC design, and non-fragile control for fuzzy systems and some effective solutions have been developed, there are still some limitations in the existing research. First, the above studies (Jia et al., 2015; Soulami et al., 2015; Brahim et al., 2017; López-Estrada et al., 2017; Kharrat et al., 2018; Li and Zhang, 2018; Zhang et al., 2018; Haj Brahim et al., 2019; Zhang J. et al., 2019) mainly discussed T-S fuzzy descriptor systems with the same derivative matrix E. Other studies (Taniguchi et al., 2000; Chen et al., 2009) showed that T-S fuzzy descriptor systems with different derivative matrices are more useful for modeling and analyzing the complexity of nonlinear systems than T-S fuzzy descriptor systems with the same derivative matrices. To the best of the authors' knowledge, to date, the problem of observer design for T-S fuzzy descriptor systems with unmeasurable premise variables and a different derivative matrix E has not been previously studied. Second, from what we can ascertain, the existing integral sliding mode observer and controller in (Jiang et al., 2018; Li et al., 2018; Kuppusamy and Joo, 2019) were designed only for T-S fuzzy systems or T-S descriptor systems with measurable premise variables. The problem of observer-based adaptive ISMC for T-S fuzzy descriptor systems with unmeasurable premise variables and uncertainties has not been previously studied. Finally, fruitful results have been obtained for non-fragile controllers and/or observers for T-S fuzzy systems such as those in (Zhang et al., 2007; Chen and Li, 2013; Guan and Liu, 2016; Li X. et al., 2017; Duan et al., 2019), but these results generally lack corresponding techniques for T-S fuzzy descriptor systems.

Motivated by the abovementioned discussion, in this article, we study non-fragile observer-based ISMC problems for T-S fuzzy descriptor systems with unmeasurable premise variables and uncertainties. The main contributions of the proposed control method are as follows.

1) In accordance with the nonparallel distributed compensation control (non-PDC) method and the ISMC theory, an observer-based integral sliding mode controller is developed for T-S fuzzy descriptor systems with unmeasurable premise variables.
2) A non-fragile integral sliding mode observer and a non-fragile observer-based sliding mode controller are constructed. In addition, system uncertainties and perturbations in both the observer structure and the controller structure are compensated for by an adaptive controller.
3) Auxiliary variables are introduced into the system with an augmented method to eliminate the coupling of Lyapunov function matrices, observer gain matrices, and/or observer gain perturbations. Moreover, a fuzzy Lyapunov function containing information for system state estimation and system state estimation error is designed to guarantee the asymptotic stability of the closed-loop system. The auxiliary variables and the fuzzy Lyapunov function produce unconservative results.

This article is organized as follows. Section Problem Formulation and Preliminary Analysis describes the system, clarifies the problem formulation, and gives relevant preliminary information. Then, in Section Non-Fragile Observer-Based ISMC for T-S Fuzzy Descriptor Systems, a non-fragile observer and a non-fragile controller are constructed. A simulation example is presented to validate the accuracy and effectiveness of the proposed method in Section Examples. Finally, conclusions are drawn in Section Conclusions.

Notation: in this study, R^m and $R^{n\times m}$ denote the n-dimensional real Euclidean space and the set of $n \times m$ matrices with real elements, respectively. I is the identity matrix with appropriate dimensions. For clarity, the following definitions are given: $\gamma_h = \sum_{i=1}^{r} h_i(\centerdot)\gamma_i$, $\gamma_h^{-1} = (\sum_{i=1}^{r} h_i(\centerdot)\gamma_i)^{-1}$, $\gamma_{h,h} = \sum_{i=1}^{r}\sum_{j=1}^{r} h_i(\centerdot)h_j(\centerdot)\gamma_{i,j}$, $\gamma_{\widehat{h}} = \sum_{i=1}^{r} h_i(\widehat{\xi}(t))\gamma_i$, $A + * = A + A^T$, $\mathrm{He}(A) = A + A^T$, and $\begin{bmatrix} A & * \\ B & C \end{bmatrix} = \begin{bmatrix} A & B^T \\ B & C \end{bmatrix}$.

PROBLEM FORMULATION AND PRELIMINARY ANALYSIS

In this study, T-S fuzzy descriptor systems are used to approximate various complex nonlinear robotics, i.e., robot manipulators (Fan et al., 2020), planar parallel robots (Vermeiren et al., 2012), an overhead crane system (Chen et al., 2009), a ball and beam system (Li H. et al., 2017), and a nonlinear active vehicle suspension system (Li et al., 2012). Therefore, consider a class of uncertain robotic systems that can be represented by the following T-S fuzzy descriptor systems with uncertainties:

$$\sum\nolimits_{k=1}^{r_e} v_k(\xi(t))(E_k + \Delta E)\dot{x}(t)$$

$$= \sum\nolimits_{i=1}^{r} h_i(z(t)) \left\{(A_i + \Delta A)x(t) + Bu(t)\right\}, \qquad (1)$$

$$y(t) = \sum\nolimits_{i=1}^{r} h_i(z(t))(C_i x(t)),$$

where $x(t) = \left[x_1(t) \cdots x_n(t)\right] \in R^n$is the system state vector, $u(t) \in R^m$ is the control input, $y(t) \in R^p$is the system output vector, and $\xi(t) = \left[\xi_1(t) \cdots \xi_l(t)\right] \in R^l$ and$z(t) = \left[z_1(t) \cdots z_q(t)\right] \in R^q$are the unmeasurable premise vectors. $v_k(\xi(t))$, $k = 1, 2, \cdots, r_e$ and $h_i(z(t))$, $i = 1, 2, \cdots, r$ are fuzzy membership functions on the left and right-hand sides, respectively. $A_i \in R^{n\times n}, E_k \in R^{n\times n}, B \in R^{n\times m}$, and $C_i \in R^{p\times n}$ are the system matrices. $\Delta E \in R^{n\times n}$ and $\Delta A \in R^{n\times n}$ are the system uncertainties. In many practical cases, there are two main sources of system uncertainties. On the one hand, mechanical devices may carry various tools or goods for various operations; thus, the system's mass, center of mass, and other coefficients tend to change with load. On the other hand, dynamic model and system parameters are challenging to accurately obtain either through theoretical methods or by experimental measurements. In many practical cases, $h_i(z(t))$ and $v_k(\xi(t))$ are different, i.e., an inverted pendulum on a cart (Li et al., 2018), an overhead crane system (Li H. et al., 2017), a ball and beam system (Li X. et al., 2017), or a nonlinear active vehicle suspension system (Li et al., 2012).

Without loss of generality, some assumptions are introduced as follows:

Assumption 1. $\sum_{k=1}^{r_e} v_k(\xi(t))(E_k + \Delta E)$is nonsingular.

Assumption 2. $\Delta E$and$\Delta A$are uncertainties satisfying $\Delta E = M_E F_E(t) N_E$ and$\Delta A = M_A F_A(t) N_A$, where M_E, N_E, M_A, and $N_A$are known real constant matrices and $F_E(t)$and$F_A(t)$ are unknown time-varying matrices that satisfy $F_E^T(t)F_E(t) \leq I$ and $F_A^T(t)F_A(t) \leq I$, respectively.

By defining $X^*(t) = \left[x^T(t)\ \dot{x}^T(t)\right]^T$, the T-S fuzzy descriptor system (1) can be transformed into an augmented form as follows:

$$E^*\dot{X}^*(t) = \sum\nolimits_{k=1}^{r_e} \sum\nolimits_{i=1}^{r} v_k(\xi(t))h_i(z(t)) \left\{(A_{ki}^* + \Delta A^*)X^*(t) + B^*u(t)\right\}, \quad (2)$$

$$y(t) = \sum\nolimits_{i=1}^{r} h_i(z(t))C_i^* X^*(t),$$

where $E^* = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$,$A_{ki}^* = \begin{bmatrix} 0 & I \\ A_i & -E_k \end{bmatrix}$,$M = \begin{bmatrix} 0 & 0 \\ M_A & -M_E \end{bmatrix}$,$\Delta A^* = \begin{bmatrix} 0 & I \\ \Delta A & -\Delta E \end{bmatrix} = MF(t)N$, $F(t) = \begin{bmatrix} F_A(t) & 0 \\ 0 & F_E(t) \end{bmatrix}$, $N = \begin{bmatrix} N_A & 0 \\ 0 & N_E \end{bmatrix}$, $B^* = \begin{bmatrix} 0 \\ B \end{bmatrix}$, and $C_i^* = \left[C_i\ 0\right]$.

Assumption 3. The output matrices C_i of the ith rule of the T-S fuzzy descriptor system are full row rank for all $i = 1, \cdots, r$, and therefore nonsingular matrices T_i exist such that

$$C_i T_i = \left[I\ 0\right].$$

Remark 1. For any given C_i, the corresponding T_i is not unique in general. One solution for T_i, as discussed in Du and Yang (2009), is:

$$T_i = \left[C_i^T (C_i C_i^T)^{-1}\ C_i^{\perp}\right],$$

where $C_i^{\perp}$ is called an orthogonal basis for the null space of C_i and$C_i C_i^{\perp} = 0$.

Some essential lemmas are introduced to facilitate stability analysis.

Lemma 1 (Boyd et al., 1994). (Schur Complement) The appropriate dimensional matrices $S_{1,1}$, $S_{1,2}$, $S_{2,1}$, and$S_{2,2}$ satisfy $S_{1,1} = S_{1,1}^T$, $S_{1,2} = S_{2,1}^T$, and $S_{2,2} = S_{2,2}^T$such that the following conditions are equivalent:

$$S = \begin{bmatrix} S_{1,1} & S_{1,2} \\ S_{2,1} & S_{2,2} \end{bmatrix}$$

1) $S < 0$,
2) $S_{1,1} < 0$ and$S_{2,2} - S_{1,2}^T S_{1,1}^{-1} S_{1,2} < 0$, and
3)$S_{2,2} < 0$ and $S_{1,1} - S_{1,2} S_{2,2}^{-1} S_{1,2}^T < 0$.

Lemma 2 (Petersen, 1987). Let $P = P^T$, M, and N be real matrices of appropriate dimensions. Then, $P + MF(t)N + N^T F^T(t) M^T < 0$ for all variable matrix functions $F(t)$ satisfying $F^T(t)F(t) \leq I$ if and only if there is a scalar $\varepsilon > 0$such that the following inequality holds:

$$P + \varepsilon MM^T + \varepsilon^{-1} N^T N < 0.$$

Lemma 3 (Gahinet and Apkarian, 1994). (Finsler's lemma) The following conditions are equivalent:

1) $x^T \Omega x < 0$, $\forall Wx = 0$, and $x \neq 0$, where x is an augmented state vector;

2) ${W^{\perp}}^T \Omega W^{\perp} < 0$, where $W^{\perp}$is any null space basis matrix for W;

3) there is a scalar μ that satisfies $\Omega - \mu W^T W < 0$; and

4) there is a matrix X that satisfies $\Omega + XW + W^T X^T < 0$.

Remark 2. $\sum_{k=1}^{r_e} v_k(\xi(t))E_k$is required to be nonsingular to ensure that the augmented systems (2) maintain the impulse-free and regularization properties of the original system.

Remark 3. In this study, a nonlinear mechanical system is represented as a T-S fuzzy descriptor system instead of a T-S fuzzy system to effectively avoid the artificial introduction of different input matrices. By setting the system matrix $E_k = I$, a T-S fuzzy descriptor system can be transformed into a normal T-S fuzzy system. Therefore, the proposed controller is also feasible for a normal T-S fuzzy system.

NON-FRAGILE OBSERVER-BASED ISMC FOR T-S FUZZY DESCRIPTOR SYSTEMS

When uncertainties of robotic system are considered such that $\Delta A^* \neq 0$, a robust observer is constructed to estimate the system states. In contrast, a non-fragile observer-based adaptive integral sliding mode controller for T-S fuzzy descriptor systems, as shown in **Figure 1**, is designed to address robotic system uncertainties and perturbations for both observer and controller structures. The control system includes two parts: a non-fragile observer and a non-fragile adaptive integral sliding mode controller. Moreover, the non-fragile adaptive integral sliding mode controller has three parts, namely, the equivalent control strategy, the switching control strategy, and the adaptive control strategy. The equivalent control strategy guarantees system convergence to the designed sliding surface, and the switching control strategy makes the closed-loop control system asymptotically stable. System uncertainties and perturbations associated with controller and observer gains are compensated for by the adaptive control strategy.

Structure of the Non-Fragile Sliding Mode Observer

To estimate the states of system (2), the following non-fragile observer for estimating system states is considered:

$$E^*\dot{\hat{X}}^*(t) = \sum\nolimits_{k=1}^{r_e}$$

$$\sum\nolimits_{i}^{r} v_k(\hat{\xi}(t))h_i(\hat{z}(t))\left\{A^*_{k,i}\hat{X}^*(t) + B^*u(t) + (L^*_{k,i} + \Delta L^*_{k,i})(y(t) - \hat{y}(t))\right\}, \quad (3)$$

$$\hat{y}(t) = \sum\nolimits_{i=1}^{r} h_i(\hat{z}(t))C^*_i\hat{X}^*(t),$$

where $\hat{X}^*(t)$ is the state estimate of $X^*(t)$, $\hat{y}(t)$ is the estimated value of the system output vector $y(t)$, and $\hat{\xi}(t)$and$\hat{z}(t)$denote the estimated values of the premise variables$\xi(t)$ and $z(t)$, respectively. Using the notation presented in Introduction, Equation (3) can be represented as:

$$E^*\dot{\hat{X}}^*(t) = A^*_{\hat{h},\hat{v}}\hat{X}^*(t) + B^*u(t) + (L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})(y(t) - \hat{y}(t)), (4)$$

$$\hat{y}(t) = C^*_{\hat{h}}\hat{X}^*(t),$$

where $L^*_{\hat{h},\hat{v}} = \begin{bmatrix} 0 & L^T_{\hat{h},\hat{v}} \end{bmatrix}^T$ and $\Delta L^*_{\hat{h},\hat{v}} = \begin{bmatrix} 0 & \Delta L^T_{\hat{h},\hat{v}} \end{bmatrix}^T$. $L_{\hat{h},\hat{v}}$ denotes the observer gain, which will be determined later. $\Delta L_{\hat{h},\hat{v}} = M_L S_L N_L L_{\hat{h},\hat{v}}$ is the observer gain perturbation, where M_L and N_L are known real constant matrices.S_L is an unknown time-varying matrix that satisfies $S_L^T S_L \leq I$.$\Delta L^*_{\hat{h},\hat{v}}$ satisfies the following norm-bounded multiplicative relation:

$$\Delta L^*_{\hat{h},\hat{v}} = M^*_L S_L N^*_L L^*_{\hat{h},\hat{v}}, \quad (5)$$

where $M^*_L = \begin{bmatrix} 0 & M_L^T \end{bmatrix}^T$ and $N^*_L = \begin{bmatrix} 0 & N_L \end{bmatrix}$.

The system state estimation error is defined as $e(t) = X^*(t) - \hat{X}^*(t)$; by considering the T-S fuzzy descriptor system (2) and the non-fragile observer system (4), the estimation error dynamic is obtained as:

$$\begin{aligned} E^*\dot{e}(t) &= (A^*_{h,v} - (L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})C^*_h)e(t) \\ &+ (A^*_{h,v} - A^*_{\hat{h},\hat{v}})\hat{X}^*(t) \\ &- (L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})(C^*_h - C^*_{\hat{h}})\hat{X}^*(t) \\ &+ \Delta A^* X^*(t). \end{aligned} \quad (6)$$

Remark 4. Since we consider unmeasurable premise variables for T-S fuzzy descriptor systems, the membership functions of the T-S fuzzy descriptor system ($v_k(\xi(t))$ and $h_i(z(t))$) should be allowed to depend on the estimated system state $\hat{x}_i(t)$ rather than the original system state $x_i(t)$. Furthermore, the system output matrix $C_i(t)$ is allowed to be a function of the system state $x_i(t)$ instead of a constant matrix.

Construction of the Integral Sliding Surface

Based on the non-fragile observer (4) and ISMC theory, the integral sliding surface function is designed as

$$\begin{aligned} s_{\hat{X}^*}(t) &= S^*E^*\hat{X}^*(t) - S^*E^*\hat{X}^*(0) \\ &- S^*\int_0^t (A^*_{\hat{h},\hat{v}} + B^*(F^*_{\hat{h},\hat{v}} + \Delta F^*_{\hat{h},\hat{v}})K^{-1}_{\hat{h},\hat{v}})\hat{X}^*(\tau)d\tau, \end{aligned} \quad (7)$$

where $F^*_{\hat{h},\hat{v}} = \begin{bmatrix} F_{\hat{h},\hat{v}} & 0 \end{bmatrix}$,$K_{\hat{h},\hat{v}} = \begin{bmatrix} K_{1,\hat{v}} & K_{2,\hat{v}} \\ K_{3,\hat{h}} & K_{4,\hat{h}} \end{bmatrix}$,$F_{\hat{h},\hat{v}} \in R^{m\times n}$, $K_{1,\hat{v}} \in R^{n\times n}$, $K_{2,\hat{v}} \in R^{n\times n}$, $K_{3,\hat{h}} \in R^{n\times n}$, and $K_{4,\hat{h}} \in R^{n\times n}$ are the system controller gains, which will be determined later. S^* is a constant matrix that satisfies $\det(S^*B^*) \neq 0$. $\Delta F^*_{\hat{h},\hat{v}} = \begin{bmatrix} \Delta F_{\hat{h},\hat{v}} & 0 \end{bmatrix}$ is the controller gain perturbation and satisfies the following norm-bounded multiplicative relation:

$$\Delta F^*_{\hat{h},\hat{v}} = M_F S_F N_F F^*_{\hat{h},\hat{v}}, \quad (8)$$

where M_F and N_F are known matrices with appropriate dimensions, and S_F is an unknown time-varying matrix that satisfies $S_F^T S_F \leq I$.

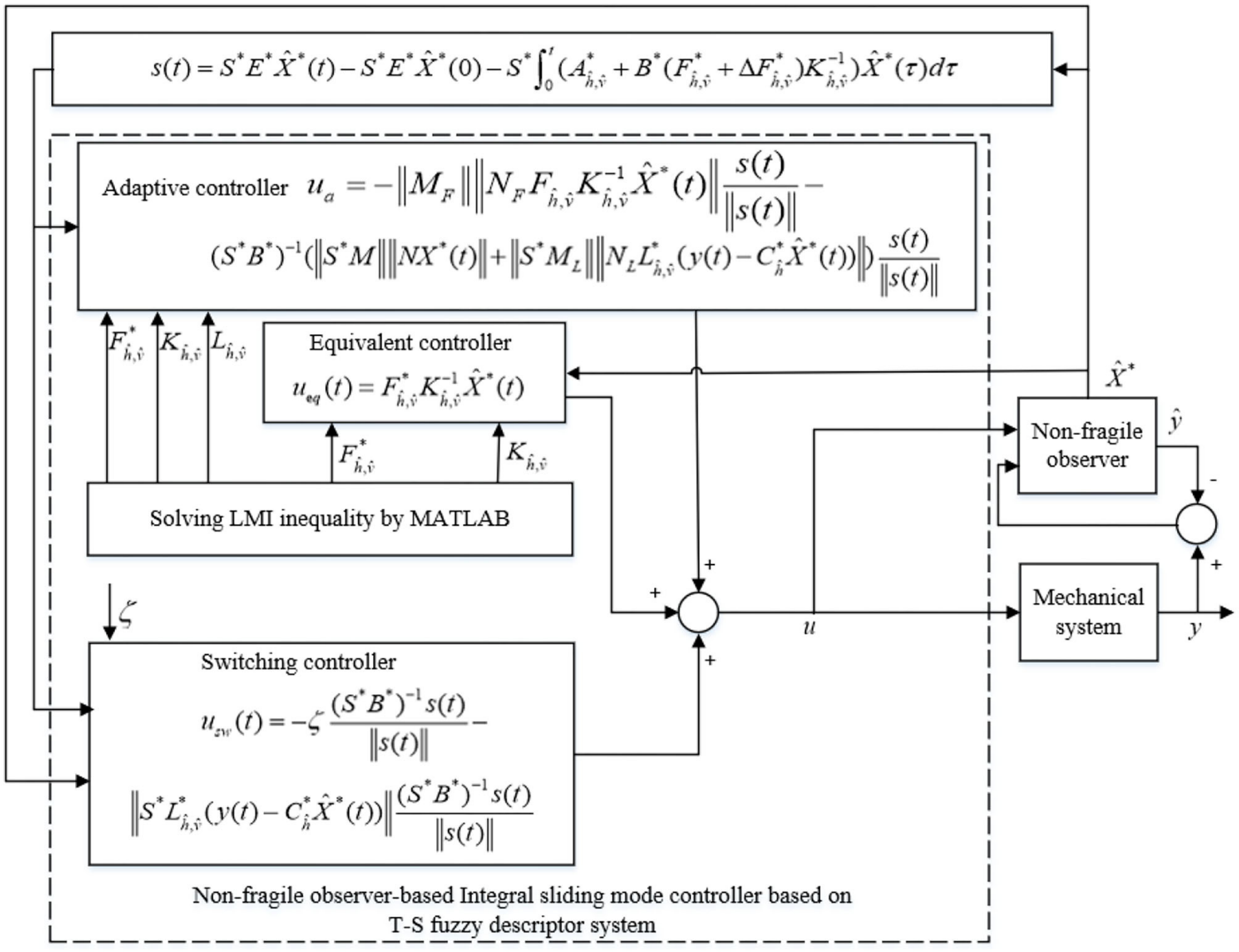

$s(t) = S^*E^*\hat{X}^*(t) - S^*E^*\hat{X}^*(0) - S^*\int_0^t (A^*_{\hat{h},\hat{v}} + B^*(F^*_{\hat{h},\hat{v}} + \Delta F^*_{\hat{h},\hat{v}})K^{-1}_{\hat{h},\hat{v}})\hat{X}^*(\tau)d\tau$
Adaptive controller $u_a = -\left\|M_F\right\|\left\|N_F F_{\hat{h},\hat{v}} K^{-1}_{\hat{h},\hat{v}}\hat{X}^*(t)\right\|\frac{s(t)}{\left\|s(t)\right\|} - (S^*B^*)^{-1}(\left\|S^*M\right\|\left\|NX^*(t)\right\| + \left\|S^*M_L\right\|\left\|N_L L^*_{\hat{h},\hat{v}}(y(t) - C^*_{\hat{h}}\hat{X}^*(t))\right\|)\frac{s(t)}{\left\|s(t)\right\|}$
$F^*_{\hat{h},\hat{v}}$
$K_{\hat{h},\hat{v}}$
$L_{\hat{h},\hat{v}}$
Equivalent controller
$u_{eq}(t) = F^*_{\hat{h},\hat{v}}K^{-1}_{\hat{h},\hat{v}}\hat{X}^*(t)$
$\hat{X}^*$
$F^*_{\hat{h},\hat{v}}$
$K_{\hat{h},\hat{v}}$
Solving LMI inequality by MATLAB
Non-fragile observer
$\hat{y}$
-
+
+
+
+
u
Mechanical system
y
ζ
Switching controller
$u_{sw}(t) = -\zeta\frac{(S^*B^*)^{-1}s(t)}{\left\|s(t)\right\|} - \left\|S^*L^*_{\hat{h},\hat{v}}(y(t) - C^*_{\hat{h}}\hat{X}^*(t))\right\|\frac{(S^*B^*)^{-1}s(t)}{\left\|s(t)\right\|}$
Non-fragile observer-based Integral sliding mode controller based on T-S fuzzy descriptor system

FIGURE 1 | Non-fragile observer-based ISMC for T-S fuzzy descriptor systems.

Combining (4) with (7), the derivative of the integral sliding mode surface (7) can be obtained as:

$$\begin{aligned}&\dot{s}_{\hat{X}^*}(t) = \\ &S^*(A^*_{\hat{h},\hat{v}}\hat{X}^*(t) + B^*u(t) + (L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})(y(t) - C^*_{\hat{h}}\hat{X}^*(t))) \\ &-S^*(A^*_{\hat{h},\hat{v}} + B^*(F^*_{\hat{h},\hat{v}} + \Delta F^*_{\hat{h},\hat{v}})K^{-1}_{\hat{h},\hat{v}})\hat{X}^*(t). \qquad (9)\end{aligned}$$

When system trajectories reach the ideal sliding surface, the following conditions must be satisfied: $s_{\hat{X}^*}(t) = 0$ and $\dot{s}_{\hat{X}^*}(t) = 0$. Therefore, the equivalent control can be designed as follows:

Case 1:$C_h = C_{\hat{h}} = C^*$

$$\begin{aligned}u_{eq}(t) &= (F^*_{\hat{h},\hat{v}} + \Delta F^*_{\hat{h},\hat{v}})K^{-1}_{\hat{h},\hat{v}}\hat{X}^*(t) \\ &-(S^*B^*)^{-1}S^*(L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})C^*e(t). \qquad (10)\end{aligned}$$

Case 2:$C_h \neq C_{\hat{h}}$

$$\begin{aligned}u_{eq}(t) &= (F^*_{\hat{h},\hat{v}} + \Delta F^*_{\hat{h},\hat{v}})K^{-1}_{\hat{h},\hat{v}}\hat{X}^*(t) \\ &-(S^*B^*)^{-1}S^*(L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})C^*_h e(t) \\ &-(S^*B^*)^{-1}S^*(L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})(C^*_h - C^*_{\hat{h}})\hat{X}^*(t). \qquad (11)\end{aligned}$$

By substituting the equivalent controller (10-11) into the sliding mode observer system (4), the sliding mode dynamics are established as (12) and (14). Meanwhile, the dynamic estimation error Equation (6) is redescribed as (13) and (15).

Case 1:$C_h = C_{\hat{h}} = C^*$

$$\begin{aligned}E^*\dot{\hat{X}}^*(t) &= (A^*_{\hat{h},\hat{v}} + B^*(F^*_{\hat{h},\hat{v}} + \Delta F^*_{\hat{h},\hat{v}})K^{-1}_{\hat{h},\hat{v}})\hat{X}^*(t) \\ &+\overline{BB}(L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})C^*e(t), \qquad (12)\end{aligned}$$

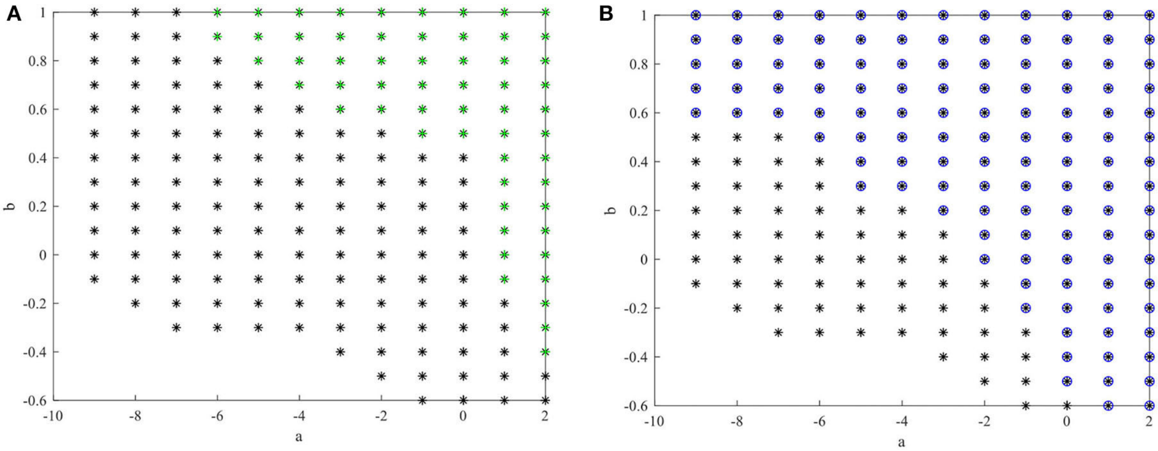


FIGURE 2 | Feasible area for theorem 3 (*) compared with the other methods. **(A)** Feasible area for Theorem 3 (*) and Theorem 2(+) [Ichalal et al., 2011]. **(B)** Feasible area for Theorem 3 (*) and Theorem 1(o) [Asemani and Majd, 2013].

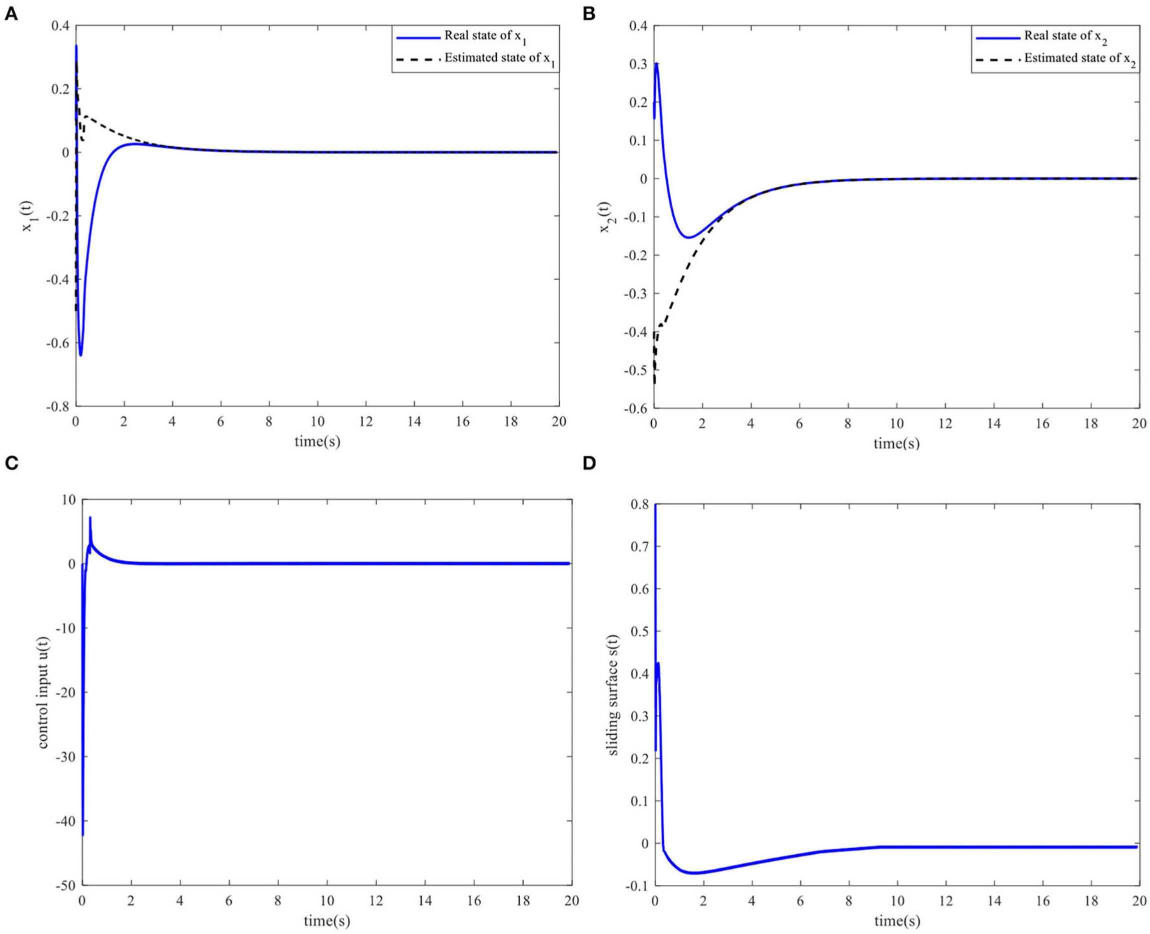


FIGURE 3 | Time responses of the system. **(A)** State $x_1(t)$ and the estimated $\hat{x}_1(t)$. **(B)** State $x_2(t)$ and the estimated $\hat{x}_2(t)$. **(C)** Control input. **(D)** Sliding surface.

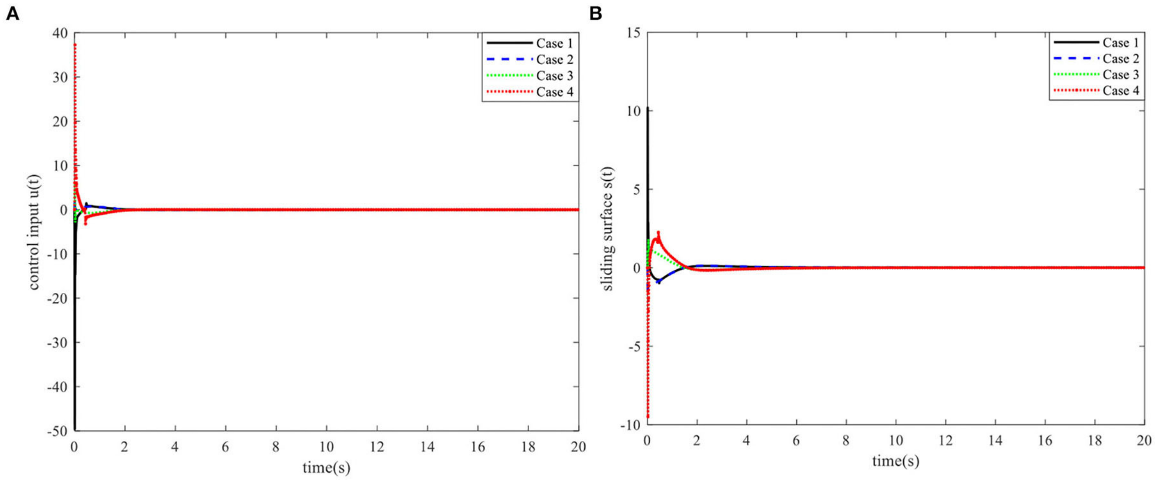


FIGURE 4 | Time responses of the control input and the sliding surface. **(A)** Control input. **(B)** Sliding surface.

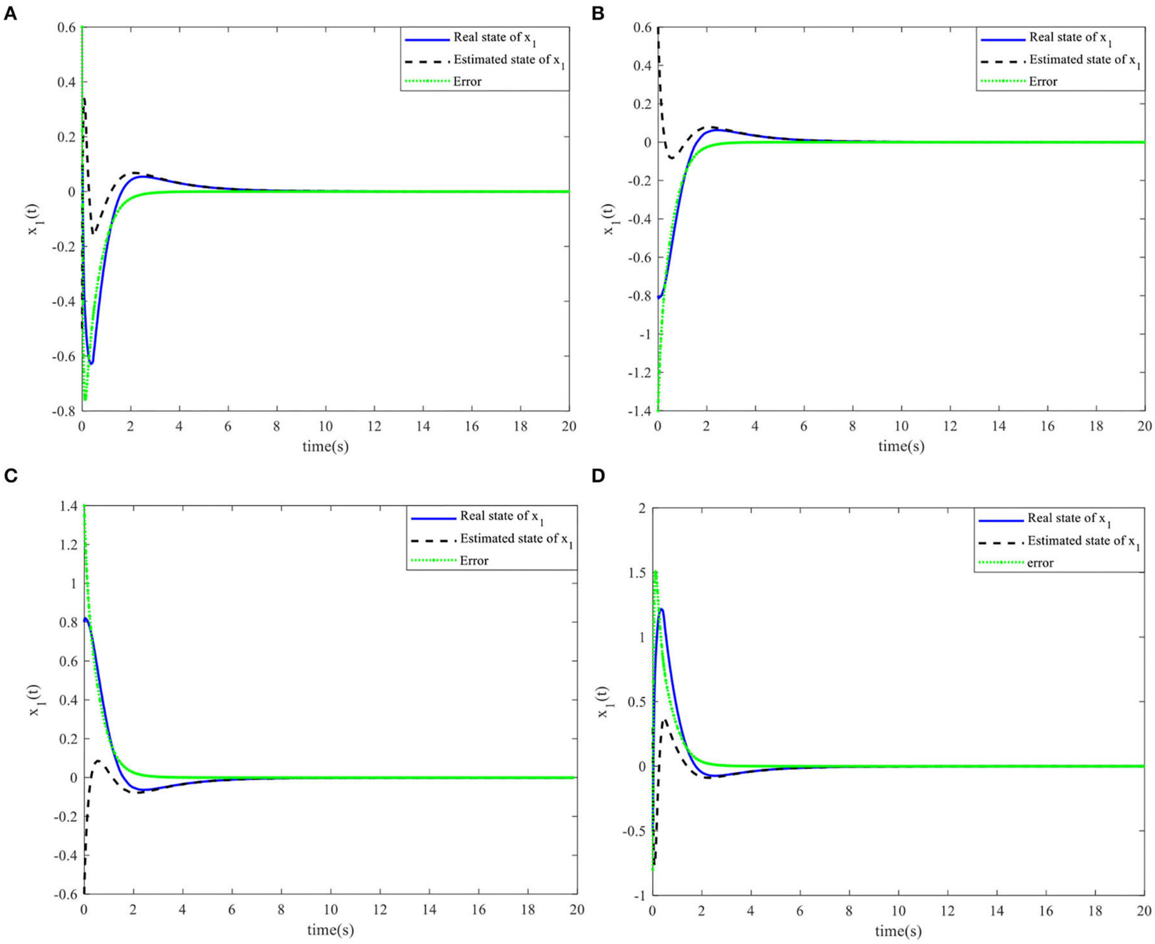


FIGURE 5 | Nonlinear state $x_1(t)$ and the estimated$\hat{x}_1(t)$. **(A)** Initial simulation case 1. **(B)** Initial simulation case 2. **(C)** Initial simulation case 3. **(D)** Initial simulation case 4.

$$E^*\dot{e}(t) = (A^*_{h,v} - (L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})C^* + \Delta A^*)e(t) + (A^*_{h,v} - A^*_{\hat{h},\hat{v}} + \Delta A^*)\hat{X}^*(t). \tag{13}$$

Case 2:$C_h \neq C_{\hat{h}}$

$$E^*\dot{\hat{X}}^*(t) = (A^*_{\hat{h},\hat{v}} + B^*(F^*_{\hat{h},\hat{v}} + \Delta F^*_{\hat{h},\hat{v}})K^{-1}_{\hat{h},\hat{v}} + \overline{BB}(L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})(C^*_h - C^*_{\hat{h}}))\hat{X}^*(t) + \overline{BB}(L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})C^*_h e(t), \tag{14}$$

$$E^*\dot{e}(t) = (A^*_{h,v} - (L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})C^*_h + \Delta A^*)e(t) + (A^*_{h,v} - A^*_{\hat{h},\hat{v}} - (L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})(C^*_h - C^*_{\hat{h}}) + \Delta A^*)\hat{X}^*(t), \tag{15}$$

where $\overline{BB} = I - B^*(S^*B^*)^{-1}S^*$. In this study, the control goal is to obtain the observer gains and controller gains such that $\hat{X}^*(t) \rightarrow 0$ and $e(t) \rightarrow 0$ for $t \rightarrow \infty$.

Equations (12) and (14) can be rearranged as follows:

Case 1:$C_h = C_{\hat{h}} = C^*$

$$\begin{bmatrix} A^*_{\hat{h},\hat{v}} + B^*(F^*_{\hat{h},\hat{v}} + \Delta F^*_{\hat{h},\hat{v}})K^{-1}_{\hat{h},\hat{v}} & -I & \overline{BB}(L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})C^* \end{bmatrix} \begin{bmatrix} \hat{X}^*(t) \\ E^*\dot{\hat{X}}^* V(t) \\ e(t) \end{bmatrix} = 0. \tag{16}$$

Case 2:$C_h \neq C_{\hat{h}}$

$$\begin{bmatrix} A^*_{\hat{h},\hat{v}} + B^*(F^*_{\hat{h},\hat{v}} + \Delta F^*_{\hat{h},\hat{v}})K^{-1}_{\hat{h},\hat{v}} + \overline{BB}(L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})(C^*_h - C^*_{\hat{h}}) & -I & \overline{BB}(L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})C^*_h \end{bmatrix} \begin{bmatrix} \hat{X}^*(t) \\ E^*\dot{\hat{X}}^*(t) \\ e(t) \end{bmatrix} = 0. \tag{17}$$

Admissibility Analysis of the Dynamic Sliding Mode System

Based on the LMI theory and the Lyapunov stability theory, the controller gains and observer gains are determined in this section such that sliding mode dynamics are asymptotically stable.

Theorem 1: Suppose that the T-S fuzzy descriptor system matrices in (2) satisfy $\Delta A^* \neq 0$ and $C_h = C_{\hat{h}} = C^*$. Given constants $\varepsilon > 0$ and $\varepsilon_1 > 0$, the closed-loop systems (12,13) are asymptotically stable if positive definite matrices P_1 and Q_1, scalars $\varepsilon_{2,i,k}$, $\varepsilon^E_{3,i,k}$, $\varepsilon^A_{3,i,k}$, and $\varepsilon_{4,i,k}$, and a set of matrices $P_{3,i,k}$, $P_{4,i,k}$, $F_{i,k}$, $K_{1,k}$, $K_{2,k}$, $K_{3,i}$, $K_{4,i}$, $L_{i,k}$, R_1, $R^{11}_{1,i,k}$, $R^{12}_{1,i,k}$, $R^2_{1,i,k}$, $R^3_{1,i,k}$, and $R^4_{1,i,k}$ exist, where $k \in \{1, \cdots, r_e\}$ and $i \in \{1, \cdots, r\}$, such that the following LMIs hold:

$$\begin{bmatrix} \Omega_1 & * \\ \Omega_2 & \Omega_3 \end{bmatrix} < 0, \tag{18}$$

$$\Omega_1 = \begin{bmatrix} \Delta_{1,1} & * & * & * & * & * & * & * \\ \Delta_{2,1} & \Delta_{2,2} & * & * & * & * & * & * \\ \Delta_{3,1} & \Delta_{3,2} & -\varepsilon \mathrm{He}(P_1) & * & * & * & * & * \\ \Delta_{4,1} & \Delta_{4,2} & -\varepsilon P_{3,\hat{h},\hat{v}} & \Delta_{4,4} & * & * & * & * \\ 0 & (\overline{B}Y_{\hat{h},\hat{v}})^T & 0 & \varepsilon(\overline{B}Y_{\hat{h},\hat{v}})^T & \mathrm{He}(Q_{3,\hat{h}}) & * & * & * \\ \Delta_{6,1} & \Delta_{6,2} & 0 & -\varepsilon M_L \varepsilon_{4,\hat{h},\hat{v}} M_L^T \overline{B}^T & \Delta_{6,5} & \Delta_{6,6} & * & * \\ 0 & \varepsilon_1(\overline{B}Y_{\hat{h},\hat{v}})^T & 0 & \varepsilon_1\varepsilon(\overline{B}Y_{\hat{h},\hat{v}})^T & Q_1 - TR^1_{1,\hat{h},\hat{v}} & \Delta_{7,6} & \Delta_{7,7} & * \\ 0 & 0 & 0 & 0 & Q_{3,\hat{h}} - R^3_{1,\hat{h},\hat{v}} & \Delta_{8,6} & \Delta_{8,7} & -\varepsilon_1 \mathrm{He}(R^4_{1,\hat{h},\hat{v}}) \end{bmatrix},$$

$$\Omega_2 = \begin{bmatrix} N_F F_{\hat{h},\hat{v}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ N_A K_{1,\hat{v}} & N_A K_{2,\hat{v}} & 0 & 0 & N_A Q_1 & 0 & 0 & 0 \\ N_E K_{3,\hat{h}} & N_E K_{4,\hat{h}} & 0 & 0 & N_E Q_{3,\hat{h}} & N_E Q_{4,\hat{h}} & 0 & 0 \\ 0 & 0 & 0 & 0 & N_L Y_{\hat{h},\hat{v}} & 0 & \varepsilon_1 N_L Y_{\hat{h},\hat{v}} & 0 \end{bmatrix},$$

$$\Omega_3 = \begin{bmatrix} -\varepsilon_{2,\hat{h},\hat{v}}I & * & * & * \\ 0 & -\varepsilon^{A}_{3,\hat{h},\hat{v}}I & * & * \\ 0 & 0 & -\varepsilon^{E}_{3,\hat{h},\hat{v}}I & * \\ 0 & 0 & 0 & -\varepsilon_{4,\hat{h},\hat{v}}I \end{bmatrix},$$

where $\Delta_{1,1} = \mathrm{He}(K_{3,\hat{h}})$, $\Delta_{2,1} = A_{\hat{h}}K_{1,\hat{v}} - E_{\hat{v}}K_{3,\hat{h}} + BF_{\hat{h},\hat{v}} + K^{T}_{4,\hat{h}}$, $\Delta_{2,2} = \mathrm{He}(A_{\hat{h}}K_{2,\hat{v}}) + BM_{F}\varepsilon_{2,\hat{h},\hat{v}}M^{T}_{F}B^{T} - \mathrm{He}(E_{\hat{v}}K_{4,\hat{h}}) + \overline{B}M_{L}\varepsilon_{4,\hat{h},\hat{v}}M^{T}_{L}\overline{B}^{T}$, $\Delta_{3,1} = K_{1,\hat{v}} + \varepsilon K_{3,\hat{h}} - P_{1}$, $\Delta_{3,2} = K_{2,\hat{v}} + \varepsilon K_{4,\hat{h}}$, $\Delta_{4,1} = K_{3,\hat{h}} - P_{3,\hat{h},\hat{v}} + \varepsilon(A_{\hat{h}}K_{1,\hat{v}} - E_{\hat{v}}K_{3,\hat{h}} + BF_{\hat{h},\hat{v}})$, $\Delta_{4,2} = \varepsilon BM_{F}\varepsilon_{2,\hat{h},\hat{v}}M^{T}_{F}B^{T} + \varepsilon\overline{B}M_{L}\varepsilon_{4,\hat{h},\hat{v}}M^{T}_{L}\overline{B}^{T} + K_{4,\hat{h}} + \varepsilon A_{\hat{h}}K_{2,\hat{v}} - P_{4,\hat{h},\hat{v}} - \varepsilon E_{\hat{v}}K_{4,\hat{h}}$, $\Delta_{4,4} = \varepsilon^{2}BM_{F}\varepsilon_{2,\hat{h},\hat{v}}M^{T}_{F}B^{T} + \varepsilon^{2}\overline{B}M_{L}\varepsilon_{4,\hat{h},\hat{v}}M^{T}_{L}\overline{B}^{T} - \varepsilon\mathrm{He}(P_{4,\hat{h},\hat{v}})$, $\Delta_{6,1} = (A_{h} - A_{\hat{h}})K_{1,\hat{v}} - E_{v}K_{3,\hat{h}} + E_{\hat{v}}K_{3,\hat{h}}$, $\Delta_{6,2} = (A_{h} - A_{\hat{h}})K_{2,\hat{v}} - (E_{v} - E_{\hat{v}})K_{4,\hat{h}} - M_{L}\varepsilon_{4,\hat{h},\hat{v}}M^{T}_{L}\overline{B}^{T}$, $\Delta_{6,5} = Q^{T}_{4,\hat{h}} - Y_{\hat{h},\hat{v}} + A_{h}Q_{1} - E_{v}Q_{3,\hat{h}}$, $\Delta_{6,6} = -\mathrm{He}(E_{v}Q_{4,\hat{h}}) + M_{L}\varepsilon_{4,\hat{h},\hat{v}}M^{T}_{L} + M_{A}\varepsilon^{A}_{3,\hat{h},\hat{v}}M^{T}_{A} + M_{E}\varepsilon^{E}_{3,\hat{h},\hat{v}}M^{T}_{E}$, $\Delta_{7,6} = -\varepsilon_{1}(Y_{\hat{h},\hat{v}})^{T} - TR^{2}_{1,\hat{h},\hat{v}}$, $\Delta_{7,7} = -\varepsilon_{1}\mathrm{He}(TR^{1}_{1,\hat{h},\hat{v}})$, $\Delta_{8,6} = Q_{4,\hat{h}} - R^{4}_{1,\hat{h},\hat{v}}$, $\Delta_{8,7} = -\varepsilon_{1}R^{3}_{1,\hat{h},\hat{v}} - \varepsilon_{1}(TR^{2}_{1,\hat{h},\hat{v}})^{T}$, $\overline{B} = B(S^{*}_{2}B)^{-1}S^{*}_{2}$, $Y_{\hat{h},\hat{v}} = \begin{bmatrix} (L_{\hat{h},\hat{v}}R_{1})_{n\times p} & 0_{n\times(n-p)} \end{bmatrix}$, $R^{1}_{1,\hat{h},\hat{v}} = \begin{bmatrix} R_{1} & 0_{p\times(n-p)} \\ R^{11}_{1,\hat{h},\hat{v}} & R^{12}_{1,\hat{h},\hat{v}} \end{bmatrix}$, and $R^{2}_{1,\hat{h},\hat{v}} = \begin{bmatrix} 0_{p\times n} \\ R^{21}_{1,\hat{h},\hat{v}} \end{bmatrix}$.

Proof: The Lyapunov function candidate is constructed as:

$$\mathrm{V}(\hat{x}(t),e_{1}(t)) = \hat{x}^{T}(t)P_{1}^{-1}\hat{x}(t) + e_{1}^{T}(t)Q_{1}^{-1}e_{1}(t) = \hat{X}^{*T}(t)E^{*T}P^{-1}_{\hat{h},\hat{v}}\hat{X}^{*}(t) + e^{T}(t)E^{*T}Q^{-1}_{\hat{h},\hat{v}}e(t), \tag{19}$$

where $P_{\hat{h},\hat{v}} = \begin{bmatrix} P_{1} & 0 \\ P_{3,\hat{h},\hat{v}} & P_{4,\hat{h},\hat{v}} \end{bmatrix}$, $P^{-1}_{\hat{h},\hat{v}} = \begin{bmatrix} P_{1}^{-1} & 0 \\ -P^{-1}_{4,\hat{h},\hat{v}}P_{3,\hat{h},\hat{v}}P_{1}^{-1} & P^{-1}_{4,\hat{h},\hat{v}} \end{bmatrix}$, $P_{1} = P_{1}^{T}$, $Q_{\hat{h},\hat{v}} = \begin{bmatrix} Q_{1} & 0 \\ Q_{3,\hat{h},\hat{v}} & Q_{4,\hat{h},\hat{v}} \end{bmatrix}$, $Q_{1} = Q_{1}^{T}$, $P_{1} \in R^{n\times n}$, $P_{3,\hat{h},\hat{v}} \in R^{n\times n}$, $P_{4,\hat{h},\hat{v}} \in R^{n\times n}$, $Q_{1} \in R^{n\times n}$, $Q_{3,\hat{h},\hat{v}} \in R^{n\times n}$, $Q_{4,\hat{h},\hat{v}} \in R^{n\times n}$, and $e_{1}(t) = x(t) - \hat{x}(t)$.The matrix and its derivative satisfy $E^{*T}P^{-1}_{\hat{h},\hat{v}} = P^{-T}_{\hat{h},\hat{v}}E^{*} = \begin{bmatrix} P_{1} & 0 \\ 0 & 0 \end{bmatrix}$, $E^{*T}Q^{-1}_{\hat{h},\hat{v}} = Q^{-T}_{\hat{h},\hat{v}}E^{*} = \begin{bmatrix} Q_{1} & 0 \\ 0 & 0 \end{bmatrix}$, $\frac{d}{dt}(E^{*T}P^{-1}_{\hat{h},\hat{v}}) = 0$, and $\frac{d}{dt}(E^{*T}Q^{-1}_{\hat{h},\hat{v}}) = 0$. Therefore, the derivative of the Lyapunov function (19) is calculated as:

$$\dot{V}(\hat{x}(t), e_{1}(t)) \;= \dot{\hat{X}}^{*T}(t)E^{*T}P^{-1}_{\hat{h},\hat{v}}\hat{X}^{*}(t) + \hat{X}^{*T}(t)P^{-T}_{\hat{h},\hat{v}}E^{*}\dot{\hat{X}}^{*}(t) + \dot{e}^{T}(t)E^{*T}Q^{-1}_{\hat{h},\hat{v}}e(t) + e^{T}(t)Q^{-T}_{\hat{h},\hat{v}}E^{*}\dot{e}(t) \tag{20}$$

$$= \begin{bmatrix} \hat{X}^{*}(t) \\ E^{*}\dot{\hat{X}}^{*}(t) \\ e(t) \end{bmatrix}^{T} \begin{bmatrix} 0 & * & * \\ P^{-1}_{\hat{h},\hat{v}} & 0 & * \\ Q^{-T}_{\hat{h},\hat{v}}(A^{*}_{h,v} - A^{*}_{\hat{h},\hat{v}} + \Delta A^{*}) & 0 & \mathrm{He}((A^{*}_{h,v} - (L^{*}_{\hat{h},\hat{v}} + \Delta L^{*}_{\hat{h},\hat{v}})C^{*} + \Delta A^{*})^{T}Q^{-1}_{\hat{h},\hat{v}}) \end{bmatrix} \begin{bmatrix} \hat{X}^{*}(t) \\ E^{*}\dot{\hat{X}}^{*}(t) \\ e(t) \end{bmatrix} < 0,$$

If (20) holds, then the following inequality can be obtained by Finsler's lemma (lemma 3) under the constraint in the sliding mode dynamics Equation (16):

$$\begin{bmatrix} 0 & * & * \\ P^{-1}_{\hat{h},\hat{v}} & 0 & * \\ Q^{-T}_{\hat{h},\hat{v}}(A^{*}_{h,v} - A^{*}_{\hat{h},\hat{v}} + \Delta A^{*}) & 0 & \mathrm{He}((A^{*}_{h,v} - (L^{*}_{\hat{h},\hat{v}} + \Delta L^{*}_{\hat{h},\hat{v}})C^{*} + \Delta A^{*})^{T}Q^{-1}_{\hat{h},\hat{v}}) \end{bmatrix} + \Xi_{1} < 0, \tag{21}$$

$$\Xi_{1} = \mathrm{He}(\begin{bmatrix} U_{\hat{h},\hat{v}} \\ V_{\hat{h},\hat{v}} \\ W_{\hat{h},\hat{v}} \end{bmatrix} \begin{bmatrix} A^{*}_{\hat{h},\hat{v}} + B^{*}(F^{*}_{\hat{h},\hat{v}} + \Delta F^{*}_{\hat{h},\hat{v}})K^{-1}_{\hat{h},\hat{v}} & -I & \overline{BB}(L^{*}_{\hat{h},\hat{v}} + \Delta L^{*}_{\hat{h},\hat{v}})C^{*} \end{bmatrix}),$$

where $U_{\hat{h},\hat{v}}$, $V_{\hat{h},\hat{v}}$, and $W_{\hat{h},\hat{v}}$ are matrix variables with appropriate dimensions. By pre- and post-multiplying (21) by $\begin{bmatrix} K^{T}_{\hat{h},\hat{v}} & 0 & 0 \\ 0 & P_{\hat{h},\hat{v}} & 0 \\ 0 & 0 & Q^{T}_{\hat{h},\hat{v}} \end{bmatrix}$ and $\begin{bmatrix} K_{\hat{h},\hat{v}} & 0 & 0 \\ 0 & P^{T}_{\hat{h},\hat{v}} & 0 \\ 0 & 0 & Q_{\hat{h},\hat{v}} \end{bmatrix}$, respectively, the following inequality is obtained:

$$\begin{bmatrix} 0 & * & * \\ K_{\hat{h},\hat{v}} & 0 & * \\ (A^{*}_{h,v} - A^{*}_{\hat{h},\hat{v}} + \Delta A^{*})K_{\hat{h},\hat{v}} & 0 & \mathrm{He}(Q^{T}_{\hat{h},\hat{v}}(A^{*}_{h,v} - (L^{*}_{\hat{h},\hat{v}} + \Delta L^{*}_{\hat{h},\hat{v}})C^{*} + \Delta A^{*})^{T}) \end{bmatrix} + \Xi_{2} < 0, \tag{22}$$

$$\Xi_2=\mathrm{He}(\begin{bmatrix} K_{\hat{h},\hat{v}}^{T}U_{\hat{h},\hat{v}} \\ P_{\hat{h},\hat{v}}V_{\hat{h},\hat{v}} \\ Q_{\hat{h},\hat{v}}^{T}W_{\hat{h},\hat{v}} \end{bmatrix}\begin{bmatrix} A_{\hat{h},\hat{v}}^{*}K_{\hat{h},\hat{v}}+B^{*}(F_{\hat{h},\hat{v}}^{*}+\Delta F_{\hat{h},\hat{v}}^{*}) & -P_{\hat{h},\hat{v}}^{T} & \overline{BB}(L_{\hat{h},\hat{v}}^{*}+\Delta L_{\hat{h},\hat{v}}^{*})C^{*}Q_{\hat{h},\hat{v}} \end{bmatrix})$$

where $U_{\hat{h},\hat{v}}=K_{\hat{h},\hat{v}}^{-T}$, $V_{\hat{h},\hat{v}}=\varepsilon P_{\hat{h},\hat{v}}^{-1}$, and $W_{\hat{h},\hat{v}}=0$ are defined for $\varepsilon>0$. Then, inequality (22) can be represented as:

$$\Xi_3+\mathrm{He}(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ (\overline{BB}(L_{\hat{h},\hat{v}}^{*}+\Delta L_{\hat{h},\hat{v}}^{*})C^{*}Q_{\hat{h},\hat{v}})^{T} & \varepsilon(\overline{BB}(L_{\hat{h},\hat{v}}^{*}+\Delta L_{\hat{h},\hat{v}}^{*})C^{*}Q_{\hat{h},\hat{v}})^{T} & -((L_{\hat{h},\hat{v}}^{*}+\Delta L_{\hat{h},\hat{v}}^{*})C^{*}Q_{\hat{h},\hat{v}})^{T} \end{bmatrix})<0, \tag{23}$$

$$\Xi_3=\begin{bmatrix} \mathrm{He}(A_{\hat{h},\hat{v}}^{*}K_{\hat{h},\hat{v}}+B^{*}(F_{\hat{h},\hat{v}}^{*}+\Delta F_{\hat{h},\hat{v}}^{*})) & * & * \\ K_{\hat{h},\hat{v}}-P_{\hat{h},\hat{v}}+\varepsilon(A_{\hat{h},\hat{v}}^{*}K_{\hat{h},\hat{v}}+B^{*}(F_{\hat{h},\hat{v}}^{*}+\Delta F_{\hat{h},\hat{v}}^{*})) & -\varepsilon\mathrm{He}(P_{\hat{h},\hat{v}}^{T}) & * \\ (A_{h,v}^{*}-A_{\hat{h},\hat{v}}^{*})K_{\hat{h},\hat{v}}+\Delta A^{*}K_{\hat{h},\hat{v}} & 0 & \mathrm{He}(\Delta A^{*}Q_{\hat{h},\hat{v}}+A_{h,v}^{*}Q_{\hat{h},\hat{v}}) \end{bmatrix},$$

The corresponding auxiliary variables are introduced into the system, and the dimension of the system is increased with the augmented method to eliminate the coupling of the Lyapunov function matrix $Q_{\hat{h},\hat{v}}$, the observer gain matrix $L_{\hat{h},\hat{v}}^{*}$, and the observer gain perturbation $\Delta L_{\hat{h},\hat{v}}^{*}$ in inequality (23). The following equations are defined:

$$x_4(t)=(\overline{BB}(L_{\hat{h},\hat{v}}^{*}+\Delta L_{\hat{h},\hat{v}}^{*})C^{*})^{T}x_1(t)+\varepsilon(\overline{BB}(L_{\hat{h},\hat{v}}^{*}+\Delta L_{\hat{h},\hat{v}}^{*})C^{*})^{T}x_2(t)-((L_{\hat{h},\hat{v}}^{*}+\Delta L_{\hat{h},\hat{v}}^{*})C^{*})^{T}x_3(t). \tag{24}$$

Thus, inequality (23) can be expressed as:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix}^{T}\Xi_4\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix}<0, \tag{25}$$

$$\Xi_4=\begin{bmatrix} \mathrm{He}(A_{\hat{h},\hat{v}}^{*}K_{\hat{h},\hat{v}}+B^{*}(F_{\hat{h},\hat{v}}^{*}+\Delta F_{\hat{h},\hat{v}}^{*})) & * & * & * \\ \Xi_4^{1} & -\varepsilon\mathrm{He}(P_{\hat{h},\hat{v}}^{T}) & * & * \\ (A_{h,v}^{*}-A_{\hat{h},\hat{v}}^{*})K_{\hat{h},\hat{v}}+\Delta A^{*}K_{\hat{h},\hat{v}} & 0 & \mathrm{He}(\Delta A^{*}Q_{\hat{h},\hat{v}}+A_{h,v}^{*}Q_{\hat{h},\hat{v}}) & * \\ 0 & 0 & Q_{\hat{h},\hat{v}} & 0 \end{bmatrix}$$

where $\Xi_4^{1}=K_{\hat{h},\hat{v}}+\varepsilon(A_{\hat{h},\hat{v}}^{*}K_{\hat{h},\hat{v}}+B^{*}(F_{\hat{h},\hat{v}}^{*}+\Delta F_{\hat{h},\hat{v}}^{*}))-P_{\hat{h},\hat{v}}$.

If (25) holds, then the following inequality can be obtained by Finsler's lemma (lemma 3) under the constraint in Equation (24).

$$\Xi_4+\Xi_5<0, \tag{26}$$

$$\Xi_5=\mathrm{He}(\begin{bmatrix} 0 \\ 0 \\ (T^{*}R_{1,\hat{h},\hat{v}})^{T} \\ \varepsilon_1(T^{*}R_{1,\hat{h},\hat{v}})^{T} \end{bmatrix}\begin{bmatrix} (\overline{BB}(L_{\hat{h},\hat{v}}^{*}+\Delta L_{\hat{h},\hat{v}}^{*})C^{*})^{T} & \varepsilon(\overline{BB}(L_{\hat{h},\hat{v}}^{*}+\Delta L_{\hat{h},\hat{v}}^{*})C^{*})^{T} & -((L_{\hat{h},\hat{v}}^{*}+\Delta L_{\hat{h},\hat{v}}^{*})C^{*})^{T} & -I \end{bmatrix})$$

where$R_{1,\hat{h},\hat{v}}=\begin{bmatrix} R_{1,\hat{h},\hat{v}}^{1} & R_{1,\hat{h},\hat{v}}^{2} \\ R_{1,\hat{h},\hat{v}}^{3} & R_{1,\hat{h},\hat{v}}^{4} \end{bmatrix}$, $R_{1,\hat{h},\hat{v}}^{1}=\begin{bmatrix} R_1 & 0_{p\times(n-p)} \\ R_{1,\hat{h},\hat{v}}^{11} & R_{1,\hat{h},\hat{v}}^{12} \end{bmatrix}$, $R_{1,\hat{h},\hat{v}}^{2}=\begin{bmatrix} 0_{p\times n} \\ R_{1,\hat{h},\hat{v}}^{21} \end{bmatrix}$, $T^{*}=\begin{bmatrix} T & 0 \\ 0 & I \end{bmatrix}$,$R_1\in R^{p\times p}$, $R_{1,\hat{h},\hat{v}}^{11}\in R^{(n-p)\times p}$, $R_{1,\hat{h},\hat{v}}^{12}\in R^{(n-p)\times(n-p)}$, $R_{1,\hat{h},\hat{v}}^{21}\in R^{(n-p)\times n}$, $R_{1,\hat{h},\hat{v}}^{3}\in R^{n\times n}$, $R_{1,\hat{h},\hat{v}}^{4}\in R^{n\times n}$, and T satisfies assumption 3.

Inequality (26) can be rewritten as follows:

$$\Xi_7 + \Xi_8 < 0, \tag{27}$$

$$\Xi_7 = \begin{bmatrix} \mathrm{He}(A^*_{\hat{h},\hat{v}}K_{\hat{h},\hat{v}} + B^*F^*_{\hat{h},\hat{v}}) & * & * & * \\ \Xi_7^1 & -\varepsilon\mathrm{He}(P^T_{\hat{h},\hat{v}}) & * & * \\ \Xi_7^2 & \varepsilon(\overline{BB}\varpi_1)^T & \mathrm{He}(A^*_{h,v}Q_{\hat{h},\hat{v}} - \varpi_1) & * \\ \varepsilon_1(\overline{BB}\varpi_1)^T & \varepsilon_1\varepsilon(\overline{BB}\varpi_1)^T & Q_{\hat{h},\hat{v}} - \varepsilon_1\varpi_1^T - T^*R_{1,\hat{h},\hat{v}} & -\varepsilon_1\mathrm{He}(T^*R_{1,\hat{h},\hat{v}})^T \end{bmatrix},$$

$$\Xi_8 = \mathrm{He}(\begin{bmatrix} B^*\Delta F^*_{\hat{h},\hat{v}} & 0 & 0 & 0 \\ \varepsilon B^*\Delta F^*_{\hat{h},\hat{v}} & 0 & 0 & 0 \\ \Delta A^*K_{\hat{h},\hat{v}} + (\overline{BB}\varpi_2)^T & \varepsilon(\overline{BB}\varpi_2)^T & \Delta A^*Q_{\hat{h},\hat{v}} - \varpi_2^T & 0 \\ \varepsilon_1\left(\overline{BB}\varpi_2\right)^T & \varepsilon_1\varepsilon\left(\overline{BB}\varpi_2\right)^T & -\varepsilon_1\varpi_2^T & 0 \end{bmatrix}),$$

where $\Xi_7^1 = K_{\hat{h},\hat{v}} + \varepsilon(A^*_{\hat{h},\hat{v}}K_{\hat{h},\hat{v}} + B^*F^*_{\hat{h},\hat{v}}) - P_{\hat{h},\hat{v}}$, $\Xi_7^2 = (A^*_{h,v} - A^*_{\hat{h},\hat{v}})K_{\hat{h},\hat{v}} + (\overline{BB}\varpi_1)^T$, $\varpi_1 = L^*_{\hat{h},\hat{v}}C^*T^*R_{1,\hat{h},\hat{v}}$, and$\varpi_2 = \Delta L^*_{\hat{h},\hat{v}}C^*T^*R_{1,\hat{h},\hat{v}}$.

Because $\Delta L^*_{\hat{h},\hat{v}} = M_L^*S_LN_L^*L^*_{\hat{h},\hat{v}}$, $\Delta F^*_{\hat{h},\hat{v}} = M_FS_FN_FF^*_{\hat{h},\hat{v}}$, and $\Delta A^* = MF(t)N$, inequality (27) can be rewritten as:

$$\Xi_7 + \mathrm{He}(M_1S_FN_1{+}M_2F(t)N_2{+}M_3S_LN_3) < 0, \tag{28}$$

where $M_1 = \begin{bmatrix} B^*M_F & \varepsilon B^*M_F & 0 & 0 \end{bmatrix}^T$, $N_1 = \begin{bmatrix} N_FF^*_{\hat{h},\hat{v}} & 0 & 0 & 0 \end{bmatrix}$, $M_2 = \begin{bmatrix} 0 & 0 & M & 0 \end{bmatrix}^T$, $N_2 = \begin{bmatrix} NK_{\hat{h},\hat{v}} & 0 & NQ_{\hat{h},\hat{v}} & 0 \end{bmatrix}$, $N_3 = \begin{bmatrix} 0 & 0 & N_L^*L^*_{\hat{h},\hat{v}}C^*T^*R_{1,\hat{h},\hat{v}} & \varepsilon_1N_L^*L^*_{\hat{h},\hat{v}}C^*T^*R_{1,\hat{h},\hat{v}} \end{bmatrix}$, and $M_3 =\begin{bmatrix} \overline{BB}M_L^* & \varepsilon\overline{BB}M_L^* & -M_L^* & 0 \end{bmatrix}^T$.

Based on lemma 2, (28) holds if there are positive real scalars $\varepsilon_{2,\hat{h},\hat{v}}$, $\varepsilon^A_{3,\hat{h},\hat{v}}$, $\varepsilon^E_{3,\hat{h},\hat{v}}$, and $\varepsilon_{4,\hat{h},\hat{v}}$ that satisfy the following relation:

$$\Xi_7 + M_1\varepsilon_{2,\hat{h},\hat{v}}M_1^T + N_1^T\varepsilon^{-1}_{2,\hat{h},\hat{v}}N_1 + M_2\varepsilon_{3,\hat{h},\hat{v}}M_2^T + N_2^T\varepsilon^{-1}_{3,\hat{h},\hat{v}}N_2 + M_3\varepsilon_{4,\hat{h},\hat{v}}M_3^T + N_3^T\varepsilon^{-1}_{4,\hat{h},\hat{v}}N_3 < 0, \tag{29}$$

where $\varepsilon_{3,\hat{h},\hat{v}} = \begin{bmatrix} \varepsilon^A_{3,\hat{h},\hat{v}}I & 0 \\ 0 & \varepsilon^E_{3,\hat{h},\hat{v}}I \end{bmatrix}$. Using the Schur complement in lemma 1, (29) holds if and only if the following relation set is satisfied:

$$\begin{bmatrix} \Xi_9^1 & * & * & * & * & * & * \\ \Xi_9^2 & \Xi_9^3 & * & * & * & * & * \\ \Xi_9^4 & \Xi_9^5 & \Xi_9^6 & * & * & * & * \\ \varepsilon_1\left(\overline{BB}\varpi_1\right)^T & \varepsilon_1\varepsilon\left(\overline{BB}\varpi_1\right)^T & \Xi_9^7 & -\varepsilon_1\mathrm{He}(T^*R_{1,\hat{h},\hat{v}}) & * & * & * \\ N_FF^*_{\hat{h},\hat{v}} & 0 & 0 & 0 & -\varepsilon_{2,\hat{h},\hat{v}}I & * & * \\ NK^*_{\hat{h},\hat{v}} & 0 & NQ_{\hat{h},\hat{v}} & 0 & 0 & -\varepsilon_{3,\hat{h},\hat{v}}I & * \\ 0 & 0 & N_L^*\varpi_1 & \varepsilon\varpi_1 & 0 & 0 & -\varepsilon_{4,\hat{h},\hat{v}}I \end{bmatrix} < 0, \tag{30}$$

where $\Xi_9^1 = \mathrm{He}(A^*_{\hat{h},\hat{v}}K_{\hat{h},\hat{v}} + B^*F^*_{\hat{h},\hat{v}}) + B^*M_F\varepsilon_{2,\hat{h},\hat{v}}M_F^TB^{*T} + \overline{BB}M_L^*\varepsilon_{4,\hat{h},\hat{v}}M_L^{*T}\overline{BB}^T$, $\Xi_9^2 = \varepsilon A^*_{\hat{h},\hat{v}}K_{\hat{h},\hat{v}} - P_{\hat{h},\hat{v}}{+}K_{\hat{h},\hat{v}} + \varepsilon B^*F^*_{\hat{h},\hat{v}} + \varepsilon B^*M_F\varepsilon_{2,\hat{h},\hat{v}}M_F^TB^{*T} + \varepsilon\overline{BB}M_L^*\varepsilon_{4,\hat{h},\hat{v}}M_L^{*T}\overline{BB}^T$, $\Xi_9^3 = \varepsilon^2B^*M_F\varepsilon_{2,\hat{h},\hat{v}}M_F^TB^{*T} - \varepsilon\mathrm{He}\left(P^T_{\hat{h},\hat{v}}\right){+}\varepsilon^2\overline{BB}M_L^*\varepsilon_{4,\hat{h},\hat{v}}M_L^{*T}\overline{BB}^T$, $\Xi_9^4 = \left(\overline{BB}\varpi_1\right)^T - M_L^*\varepsilon_{4,\hat{h},\hat{v}}M_L^{*T}\overline{BB}^T + A^*_{h,v}K_{\hat{h},\hat{v}} - A^*_{\hat{h},\hat{v}}K_{\hat{h},\hat{v}}$, $\Xi_9^5 = \varepsilon\left(\overline{BB}\varpi_1\right)^T{-}\varepsilon M_L^*\varepsilon_{4,\hat{h},\hat{v}}M_L^{*T}\overline{BB}^T$, $\Xi_9^6 = \mathrm{He}\left(A^*_{h,v}Q_{\hat{h},\hat{v}} - \varpi_1\right) + M\varepsilon_{3,\hat{h},\hat{v}}M^T + M_L^*\varepsilon_{4,\hat{h},\hat{v}}M_L^{*T}$, $\Xi_9^7 = Q_{\hat{h},\hat{v}} - \varepsilon_1\varpi_1^T{-}T^*R_{1,\hat{h},\hat{v}}$, and $\varpi_1 = L^*_{\hat{h},\hat{v}}C^*T^*R_{1,\hat{h},\hat{v}}$.

It is assumed that $S^* = \begin{bmatrix} S_1^* & S_2^* \end{bmatrix}$; thus, the matrix $\overline{BB}$can be obtained as:

$$\overline{BB} = \begin{bmatrix} I & 0 \\ -B(S_2^*B)^{-1}S_1^* & I - B(S_2^*B)^{-1}S_2^* \end{bmatrix}.$$

Then, the matrices in inequality (31) can be formulated as follows:

$$L^*_{\hat{h},\hat{v}} = \begin{bmatrix} 0 & L^T_{\hat{h},\hat{v}} \end{bmatrix}^T, T^* = \begin{bmatrix} T & 0 \\ 0 & I \end{bmatrix}, \; T^* R_{1,\hat{h},\hat{v}} = \begin{bmatrix} TR^1_{1,\hat{h},\hat{v}} & TR^2_{1,\hat{h},\hat{v}} \\ R^3_{1,\hat{h},\hat{v}} & R^4_{1,\hat{h},\hat{v}} \end{bmatrix},$$

$$L^*_{\hat{h},\hat{v}} C^* T^* R_{1,\hat{h},\hat{v}} = \begin{bmatrix} 0 & 0 \\ L_{\hat{h},\hat{v}} CTR^1_{1,\hat{h},\hat{v}} & 0 \end{bmatrix} = \begin{bmatrix} 0_{n\times n} & 0_{n\times n} \\ \begin{bmatrix} (L_{\hat{h},\hat{v}} R_1)_{n\times p} & 0_{n\times(n-p)} \end{bmatrix} & 0_{n\times n} \end{bmatrix},$$

$$\overline{BB} L^*_{\hat{h},\hat{v}} C^* T^* R_{1,\hat{h},\hat{v}} = \begin{bmatrix} 0_{n\times n} & 0_{n\times n} \\ \begin{bmatrix} ((I - B(S_2^* B)^{-1} S_2^*) L_{\hat{h},\hat{v}} R_1)_{n\times p} & 0_{n\times(n-p)} \end{bmatrix} & 0_{n\times n} \end{bmatrix}.$$

Therefore, inequality (18) is easily obtained from the above equalities and inequality (30).

Remark 5. To avoid introducing the derivative of the membership function in the derivative of the Lyapunov function, P_1 is chosen as a constant matrix. The $P_{3,\hat{h},\hat{v}}$ and $P_{4,\hat{h},\hat{v}}$ in $P_{\hat{h},\hat{v}}$ are related to the membership functions $v_k(\hat{\xi}(t))$ and $h_i(\hat{z}(t))$. Consequently, the conservativeness of this approach is significantly reduced. Moreover, the Lyapunov matrices and the observer gain matrices are decoupled by Finsler's lemma.

T-S fuzzy descriptor systems may occasionally have different output matrices, so the following theorem gives the relaxed sufficient LMI conditions for T-S fuzzy descriptor systems with different output matrices to expand the application scope of the design theory given in this study.

Theorem 2: Suppose that the T-S fuzzy descriptor system matrices in (2) satisfy $\Delta A^* \neq 0$ and $C_h \neq C_{\hat{h}}$. Given constants $\varepsilon > 0$, $\varepsilon_1 > 0$and $\varepsilon_2 > 0$, the closed-loop systems (14, 15) are asymptotically stable if positive definite matrices P_1 and Q_1, scalars $\varepsilon_{3,i,k}$, $\varepsilon^E_{4,i,k}$, $\varepsilon^A_{4,i,k}$, and $\varepsilon_{5,i,k}$, and a set of matrices $P_{3,i,k}$, $P_{4,i,k}$, $F_{i,k}$, $K_{1,k}$, $K_{2,k}$, $K_{3,i}$, $K_{4,i}$, $L_{i,k}$, R_1, $R^{11}_{1,i,k}$, $R^{12}_{1,i,k}$, $R^{11}_{2,i,k}$, $R^{12}_{2,i,k}$,$R^2_{1,i,k}$, $R^3_{1,i,k}$, $R^4_{1,i,k}$, $R^2_{2,i,k}$, $R^3_{2,i,k}$, and $R^4_{2,i,k}$exist, where $k \in \{1, \cdots, r_e\}$ and $i \in \{1, \cdots, r\}$, such that the following LMIs hold:

$$\begin{bmatrix} \Omega_1 & * \\ \Omega_2 & \Omega_3 \end{bmatrix} < 0, \tag{31}$$

$$\Omega_1 = \begin{bmatrix} \Delta_{1,1} & * & * & * & * & * & * & * & * & * \\ \Delta_{2,1} & \Delta_{2,2} & * & * & * & * & * & * & * & * \\ \Delta_{3,1} & \Delta_{3,2} & \Delta_{3,3} & * & * & * & * & * & * & * \\ \Delta_{4,1} & \Delta_{4,2} & \Delta_{4,3} & \Delta_{4,4} & * & * & * & * & * & * \\ 0 & \Delta_{5,2} & 0 & \Delta_{5,4} & \Delta_{5,5} & * & * & * & * & * \\ \Delta_{6,1} & \Delta_{6,2} & -Y_{\hat{h},\hat{v}} & \Delta_{6,4} & \Delta_{6,5} & \Delta_{6,6} & * & * & * & * \\ K_{1,\hat{v}} & \Delta_{7,2} & 0 & \Delta_{7,4} & \Delta_{7,5} & \Delta_{7,6} & \Delta_{7,7} & * & * & * \\ K_{3,\hat{h}} & K_{4,\hat{h}} & 0 & 0 & \Delta_{8,5} & \Delta_{8,6} & \Delta_{8,7} & \Delta_{8,8} & * & * \\ -K_{1,\hat{v}} & \Delta_{92} & -T_{\hat{h}} R^1_{2,\hat{h},\hat{v}} & \Delta_{9,4} & 0 & \Delta_{9,6} & 0 & 0 & \Delta_{9,9} & * \\ -K_{3,\hat{h}} & K_{4,\hat{h}} & -R^3_{2,\hat{h},\hat{v}} & -R^4_{2,\hat{h},\hat{v}} & 0 & 0 & 0 & 0 & \Delta_{10,9} & -\varepsilon_2 \mathrm{He}(T_{\hat{h}} R^4_{2,\hat{h},\hat{v}}) \end{bmatrix},$$

$$\Omega_2 = \begin{bmatrix} N_F F_{\hat{h},\hat{v}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ N_A K_{1,\hat{v}} & N_A K_{2,\hat{v}} & 0 & 0 & N_A & 0 & 0 & 0 & 0 & 0 \\ N_E K_{3,\hat{h}} & N_E K_{4,\hat{h}} & 0 & 0 & N_E Q_{3,\hat{h}} & N_E Q_{4,\hat{h}} & 0 & 0 & 0 & 0 \\ 0 & 0 & N_L Y_{\hat{h},\hat{v}} & 0 & N_L Y_{\hat{h},\hat{v}} & 0 & \varepsilon_1 N_L Y_{\hat{h},\hat{v}} & 0 & \varepsilon_2 N_L Y_{\hat{h},\hat{v}} & 0 \end{bmatrix},$$

$$\Omega_3 = \begin{bmatrix} -\varepsilon_{3,\hat{h},\hat{v}} I & * & * & * \\ 0 & -\varepsilon^A_{4,\hat{h},\hat{v}} I & * & * \\ 0 & 0 & -\varepsilon^E_{4,\hat{h},\hat{v}} I & * \\ 0 & 0 & 0 & -\varepsilon_{5,\hat{h},\hat{v}} I \end{bmatrix},$$

where $\Delta_{1,1} = \mathrm{He}(K_{3,\hat{h}})$, $\Delta_{2,1} = A_{\hat{h}}K_{1,\hat{v}} - E_{\hat{v}}K_{3,\hat{h}} + BF_{\hat{h},\hat{v}} + K^T_{4,\hat{h}}$, $\Delta_{2,2} = BM_F\varepsilon_{3,\hat{h},\hat{v}}M_F^TB^T + \mathrm{He}(A_{\hat{h}}K_{2,\hat{v}}) - \mathrm{He}(E_{\hat{v}}K_{4,\hat{h}}) + \overline{B}M_L\varepsilon_{5,\hat{h},\hat{v}}M_L^T\overline{B}^T$, $\Delta_{3,1} = K_{1,\hat{v}} + \varepsilon K_{3,\hat{h}} - P_1$, $\Delta_{3,2} = K_{2,\hat{v}} + \varepsilon K_{4,\hat{h}}$, $\Delta_{3,3} = -\varepsilon\mathrm{He}(P_1)$, $\Delta_{4,1} = K_{3,\hat{h}} + \varepsilon(A_{\hat{h}}K_{1,\hat{v}} - E_{\hat{v}}K_{3,\hat{h}} + BF_{\hat{h},\hat{v}}) - P_{3,\hat{h},\hat{v}}$,$\Delta_{4,2} = K_{4,\hat{h}} + \varepsilon(A_{\hat{h}}K_{2,\hat{v}} - E_{\hat{v}}K_{4,\hat{h}}) + \varepsilon BM_F\varepsilon_{3,\hat{h},\hat{v}}M_F^TB^T - P_{4,\hat{h},\hat{v}} + \varepsilon\overline{B}M_L\varepsilon_{5,\hat{h},\hat{v}}M_L^T\overline{B}^T$, $\Delta_{4,3} = -\varepsilon P_{3,\hat{h},\hat{v}} + \varepsilon\overline{B}Y_{\hat{h},\hat{v}}$, $\Delta_{4,4} = \varepsilon^2BM_F\varepsilon_{3,\hat{h},\hat{v}}M_F^TB^T + \varepsilon^2\overline{B}M_L\varepsilon_{5,\hat{h},\hat{v}}M_L^T\overline{B}^T - \varepsilon\mathrm{He}(P_{4,\hat{h},\hat{v}})$, $\Delta_{5,2} = (\overline{B}Y_{\hat{h},\hat{v}})^T$, $\Delta_{5,4} = \varepsilon(\overline{B}Y_{\hat{h},\hat{v}})^T$, $\Delta_{5,5} = \mathrm{He}(Q_{3,\hat{h}})$, $\Delta_{6,1} = A_hK_{1,\hat{v}} - A_{\hat{h}}K_{1,\hat{v}} - (E_v - E_{\hat{v}})K_{3,\hat{h}}$, $\Delta_{6,2} = -(E_v - E_{\hat{v}})K_{4,\hat{h}} - M_L\varepsilon_{5,\hat{h},\hat{v}}M_L^T\overline{B}^T + (A_h - A_{\hat{h}})K_{2,\hat{v}}$, $\Delta_{6,4} = -\varepsilon M_L\varepsilon_{5,\hat{h},\hat{v}}M_L^T\overline{B}^T$, $\Delta_{6,5} = A_hQ_1 - E_vQ_{3,\hat{h}} + Q^T_{4,\hat{h}} - Y_{\hat{h},\hat{v}}$, $\Delta_{6,6} = -\mathrm{He}(E_vQ_{4,\hat{h}}) + M_L\varepsilon_{5,\hat{h},\hat{v}}M_L^T + M_A\varepsilon^A_{4,\hat{h},\hat{v}}M_A^T + M_E\varepsilon^E_{4,\hat{h},\hat{v}}M_E^T$,$\Delta_{7,2} = K_{2,\hat{v}} + \varepsilon_1(\overline{B}Y_{\hat{h},\hat{v}})^T$, $\Delta_{7,4} = \varepsilon_1\varepsilon(\overline{B}Y_{\hat{h},\hat{v}})^T$, $\Delta_{7,5} = Q_1 - TR^1_{1,\hat{h},\hat{v}}$, $\Delta_{7,6} = -\varepsilon_1(Y_{\hat{h},\hat{v}})^T - TR^2_{1,\hat{h},\hat{v}}$, $\Delta_{7,7} =-\varepsilon_1\mathrm{He}(TR^1_{1,\hat{h},\hat{v}})$, $\Delta_{8,5} = Q_{3,\hat{h}} - R^3_{1,\hat{h},\hat{v}}$, $\Delta_{8,6} = Q_{4,\hat{h}} - R^4_{1,\hat{h},\hat{v}}$, $\Delta_{8,7} = -\varepsilon_1R^3_{1,\hat{h},\hat{v}} - \varepsilon_1(TR^2_{1,\hat{h},\hat{v}})^T$, $\Delta_{8,8} =-\varepsilon_1\mathrm{He}(R^4_{1,\hat{h},\hat{v}})$, $\Delta_{9,2} = -K_{2,\hat{v}} + \varepsilon_2(\overline{B}Y_{\hat{h},\hat{v}})^T$, $\Delta_{9,4} = -T_{\hat{h}}R^2_{2,\hat{h},\hat{v}} + \varepsilon_2\varepsilon(\overline{B}Y_{\hat{h},\hat{v}})^T$, $\Delta_{9,6} = -\varepsilon_2(Y_{\hat{h},\hat{v}})^T$, $\Delta_{9,9} =-\varepsilon_2\mathrm{He}(T_{\hat{h}}R^1_{2,\hat{h},\hat{v}})$, $\Delta_{10,9} = -\varepsilon_2R^3_{2,\hat{h},\hat{v}} - \varepsilon_2(T_{\hat{h}}R^2_{2,\hat{h},\hat{v}})^T$, $Y_{\hat{h},\hat{v}} = \begin{bmatrix}(L_{\hat{h},\hat{v}}R_1)_{n\times p} & 0_{n\times(n-p)}\end{bmatrix}$,$\overline{B} = -B(S_2^*B)^{-1}S_2^*+I$, $R^1_{1,\hat{h},\hat{v}} = \begin{bmatrix}R_1 & 0_{p\times(n-p)}\\ R^{11}_{1,\hat{h},\hat{v}} & R^{12}_{1,\hat{h},\hat{v}}\end{bmatrix}$, $R^1_{2,\hat{h},\hat{v}} = \begin{bmatrix}R_1 & 0_{p\times(n-p)}\\ R^{11}_{2,\hat{h},\hat{v}} & R^{12}_{2,\hat{h},\hat{v}}\end{bmatrix}$, $R^2_{1,\hat{h},\hat{v}} = \begin{bmatrix}0_{p\times n}\\ R^{21}_{1,\hat{h},\hat{v}}\end{bmatrix}$, and $R^2_{2,\hat{h},\hat{v}} = \begin{bmatrix}0_{p\times n}\\ R^{21}_{2,\hat{h},\hat{v}}\end{bmatrix}$.

Proof: Under the conditions of $C_h \neq C_{\hat{h}}$, the constructed Lyapunov function candidate is the same as in Equation (19). However, the sliding mode dynamics (Equations 14 and 15) under the condition of $C_h \neq C_{\hat{h}}$are different from the sliding mode dynamics (Equations 12 and 13) under the conditions of $C_h = C_{\hat{h}} = C^*$. The derivative of the Lyapunov function is calculated based on the sliding mode dynamics. Therefore, the derivative of the Lyapunov function candidate (19) under the condition of $C_h \neq C_{\hat{h}}$can be obtained as follows:

$$\dot{V}(\hat{x}(t), e_1(t)) = \dot{\hat{X}}^{*T}(t)E^{*T}P^{-1}_{\hat{h},\hat{v}}\hat{X}^*(t) + \hat{X}^{*T}(t)P^{-T}_{\hat{h},\hat{v}}E^*\dot{\hat{X}}^*(t) + \dot{e}^T(t)E^{*T}Q^{-1}_{\hat{h},\hat{v}}e(t) + e^T(t)Q^{-T}_{\hat{h},\hat{v}}E^*\dot{e}(t) \tag{32}$$

$$= \begin{bmatrix}\hat{X}^*(t)\\ E^*\dot{\hat{X}}^*(t)\\ e(t)\end{bmatrix}^T \Psi_1 \begin{bmatrix}\hat{X}^*(t)\\ E^*\dot{\hat{X}}^*(t)\\ e(t)\end{bmatrix} < 0.$$

$$\Psi_1 = \begin{bmatrix} 0 & * & * \\ P^{-1}_{\hat{h}\hat{v}} & 0 & * \\ Q^{-T}_{\hat{h},\hat{v}}(A^*_{h,v} - A^*_{\hat{h},\hat{v}} + \Delta A^* - (L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})(C^*_h - C^*_{\hat{h}})) & 0 & \mathrm{He}(Q^{-T}_{\hat{h},\hat{v}}(A^*_{h,v} - (L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})C^*_h)) \end{bmatrix}$$

If (32) holds, then the following inequality can be obtained by Finsler's lemma (lemma 3) under the constraint in Equation (17).

$$\Psi_1 + \mathrm{He}(\begin{bmatrix}U_{\hat{h},\hat{v}}\\ V_{\hat{h},\hat{v}}\\ W_{\hat{h},\hat{v}}\end{bmatrix}\begin{bmatrix}A^*_{\hat{h},\hat{v}} + B^*(F^*_{\hat{h},\hat{v}} + \Delta F^*_{\hat{h},\hat{v}})K^{-1}_{\hat{h},\hat{v}} + \overline{BB}(L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})(C^*_h - C^*_{\hat{h}}) & -I & \overline{BB}(L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})C^*_h\end{bmatrix}) < 0, \tag{33}$$

where $U_{\hat{h},\hat{v}}$, $V_{\hat{h},\hat{v}}$, and $W_{\hat{h},\hat{v}}$ are matrix variables with appropriate dimensions. By pre- and post-multiplying (33) by $\begin{bmatrix}K^T_{\hat{h},\hat{v}} & 0 & 0\\ 0 & P_{\hat{h},\hat{v}} & 0\\ 0 & 0 & Q^T_{\hat{h},\hat{v}}\end{bmatrix}$ and $\begin{bmatrix}K_{\hat{h},\hat{v}} & 0 & 0\\ 0 & P^T_{\hat{h},\hat{v}} & 0\\ 0 & 0 & Q_{\hat{h},\hat{v}}\end{bmatrix}$, respectively, the following inequality is obtained:

$$\Psi_2 + \Psi_3 < 0, \tag{34}$$

$$\Psi_2 = \begin{bmatrix} 0 & * & * \\ K_{\hat{h},\hat{v}} & 0 & * \\ (A^*_{h,v} - A^*_{\hat{h},\hat{v}} + \Delta A^* - (L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})(C^*_h - C^*_{\hat{h}}))K_{\hat{h},\hat{v}} & 0 & \mathrm{He}((A^*_{h,v} - (L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})C^*_h)Q_{\hat{h},\hat{v}} + \Delta A^*) \end{bmatrix},$$

$$\Psi_3\text{=He}(\begin{bmatrix} K^T_{\hat{h},\hat{v}}U_{\hat{h},\hat{v}} \\ P_{\hat{h},\hat{v}}V_{\hat{h},\hat{v}} \\ Q^T_{\hat{h},\hat{v}}W_{\hat{h},\hat{v}} \end{bmatrix}\begin{bmatrix} \Psi^1_3 & -P^T_{\hat{h},\hat{v}} & \overline{BB}(L^*_{\hat{h},\hat{v}}+\Delta L^*_{\hat{h},\hat{v}})C^*_hQ_{\hat{h},\hat{v}} \end{bmatrix}),$$

where $\Psi^1_3\text{=}A^*_{\hat{h},\hat{v}}K_{\hat{h},\hat{v}}+B^*(F^*_{\hat{h},\hat{v}}+\Delta F^*_{\hat{h},\hat{v}})+\overline{BB}(L^*_{\hat{h},\hat{v}}+\Delta L^*_{\hat{h},\hat{v}})(C^*_h-C^*_{\hat{h}})K_{\hat{h},\hat{v}}$.

Define $U_{\hat{h},\hat{v}}=K^{-T}_{\hat{h},\hat{v}}$, $V_{\hat{h},\hat{v}}=\varepsilon P^{-1}_{\hat{h},\hat{v}}$, $W_{\hat{h},\hat{v}}=0$, and $\varepsilon>0$. Then, inequality (34) can be represented as:

$$\Psi_4+\Psi_5<0, \tag{35}$$

$$\Psi_4=\begin{bmatrix} \text{He}(A^*_{\hat{h},\hat{v}}K_{\hat{h},\hat{v}}+B^*(F^*_{\hat{h},\hat{v}}+\Delta F^*_{\hat{h},\hat{v}})) & * & * \\ K_{\hat{h},\hat{v}}+\varepsilon(A^*_{\hat{h},\hat{v}}K_{\hat{h},\hat{v}}+B^*(F^*_{\hat{h},\hat{v}}+\Delta F^*_{\hat{h},\hat{v}}))-P_{\hat{h},\hat{v}} & -\varepsilon\text{He}(P_{\hat{h},\hat{v}}) & * \\ (A^*_{h,v}-A^*_{\hat{h},\hat{v}})K_{\hat{h},\hat{v}}+\Delta A^*K_{\hat{h},\hat{v}} & 0 & \text{He}(\Delta A^*Q_{\hat{h},\hat{v}}+A^*_{h,v}Q_{\hat{h},\hat{v}}) \end{bmatrix},$$

$$\Psi_5=\begin{bmatrix} \text{He}(\overline{BB}(L^*_{\hat{h},\hat{v}}+\Delta L^*_{\hat{h},\hat{v}})(C^*_h-C^*_{\hat{h}})K_{\hat{h},\hat{v}}) & * & * \\ \varepsilon\overline{BB}(L^*_{\hat{h},\hat{v}}+\Delta L^*_{\hat{h},\hat{v}})(C^*_h-C^*_{\hat{h}})K_{\hat{h},\hat{v}} & 0 & * \\ \Psi^1_5 & \varepsilon(\overline{BB}(L^*_{\hat{h},\hat{v}}+\Delta L^*_{\hat{h},\hat{v}})C^*_hQ_{\hat{h},\hat{v}})^T & -\text{He}((L^*_{\hat{h},\hat{v}}+\Delta L^*_{\hat{h},\hat{v}})C^*_hQ_{\hat{h},\hat{v}})^T \end{bmatrix},$$

where$\Psi^1_5=(\overline{BB}(L^*_{\hat{h},\hat{v}}+\Delta L^*_{\hat{h},\hat{v}})C^*_hQ_{\hat{h},\hat{v}})^T-(L^*_{\hat{h},\hat{v}}+\Delta L^*_{\hat{h},\hat{v}})(C^*_h-C^*_{\hat{h}})K_{\hat{h},\hat{v}}$.

The corresponding auxiliary variables are introduced into the system, and the dimension of the system is increased with the augmented method to eliminate the coupling between the Lyapunov function matrix $Q_{\hat{h},\hat{v}}$ and the observer gain matrix $L^*_{\hat{h},\hat{v}}$ in inequality (35). The following equations are defined:

$$x_4(t)=(\overline{BB}(L^*_{\hat{h},\hat{v}}+\Delta L^*_{\hat{h},\hat{v}})C^*_h)^Tx_1(t)+\varepsilon(\overline{BB}(L^*_{\hat{h},\hat{v}}+\Delta L^*_{\hat{h},\hat{v}})C^*_h)^Tx_2(t)-((L^*_{\hat{h},\hat{v}}+\Delta L^*_{\hat{h},\hat{v}})C^*_h)^Tx_3(t), \tag{36}$$

$$x_5(t)=(\overline{BB}(L^*_{\hat{h},\hat{v}}+\Delta L^*_{\hat{h},\hat{v}})C^*_{\hat{h}})^Tx_1(t)+\varepsilon(\overline{BB}(L^*_{\hat{h},\hat{v}}+\Delta L^*_{\hat{h},\hat{v}})C^*_{\hat{h}})^Tx_2(t)-((L^*_{\hat{h},\hat{v}}+\Delta L^*_{\hat{h},\hat{v}})C^*_{\hat{h}})^Tx_3(t). \tag{37}$$

Thus, inequality (35) can be expressed as:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \end{bmatrix}^T \Psi_6 \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \end{bmatrix}<0, \tag{38}$$

$$\Psi_6=\begin{bmatrix} \text{He}(A^*_{\hat{h},\hat{v}}K_{\hat{h},\hat{v}}+B^*(F^*_{\hat{h},\hat{v}}+\Delta F^*_{\hat{h},\hat{v}})) & * & * & * & * \\ K_{\hat{h},\hat{v}}+\varepsilon(A^*_{\hat{h},\hat{v}}K_{\hat{h},\hat{v}}+B^*(F^*_{\hat{h},\hat{v}}+\Delta F^*_{\hat{h},\hat{v}}))-P_{\hat{h},\hat{v}} & -\varepsilon\text{He}(P_{\hat{h},\hat{v}}) & * & * & * \\ (A^*_{h,v}-A^*_{\hat{h},\hat{v}})K_{\hat{h},\hat{v}}+\Delta A^*K_{\hat{h},\hat{v}} & 0 & \text{He}(\Delta A^*Q_{\hat{h},\hat{v}}+A^*_{h,v}Q_{\hat{h},\hat{v}}) & * & * \\ K_{\hat{h},\hat{v}} & 0 & Q_{\hat{h},\hat{v}} & 0 & * \\ -K_{\hat{h},\hat{v}} & 0 & 0 & 0 & 0 \end{bmatrix}$$

If (38) holds, then the following inequality can be obtained by Finsler's lemma (lemma 3) under the constraints in Equations (36, 37).

$$\Psi_7 + \mathrm{He}(\Psi_8) < 0, \tag{39}$$

$$\Psi_7 = \begin{bmatrix} \mathrm{He}(A^*_{\hat{h},\hat{v}}K_{\hat{h},\hat{v}} + B^*(F^*_{\hat{h},\hat{v}} + \Delta F^*_{\hat{h},\hat{v}})) & * & * & * & * \\ K_{\hat{h},\hat{v}} - P_{\hat{h},\hat{v}} + \varepsilon(A^*_{\hat{h},\hat{v}}K_{\hat{h},\hat{v}} + B^*(F^*_{\hat{h},\hat{v}} + \Delta F^*_{\hat{h},\hat{v}})) & -\varepsilon\mathrm{He}(P_{\hat{h},\hat{v}}) & * & * & * \\ (A^*_{h,v} - A^*_{\hat{h},\hat{v}})K_{\hat{h},\hat{v}} + \Delta A^* K_{\hat{h},\hat{v}} & 0 & \mathrm{He}(A^*_{h,v}Q_{\hat{h},\hat{v}} + \Delta A^* Q_{\hat{h},\hat{v}}) & * & * \\ K_{\hat{h},\hat{v}} & 0 & Q_{\hat{h},\hat{v}} & 0 & * \\ -K_{\hat{h},\hat{v}} & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\Psi_8 = \begin{bmatrix} 0 & 0 \\ 0 & (T^*_{\hat{h}}R^*_{2,\hat{h},\hat{v}})^T \\ (T^*_h R^*_{1,\hat{h},\hat{v}})^T & 0 \\ \varepsilon_1(T^*_h R^*_{1,\hat{h},\hat{v}})^T & 0 \\ 0 & \varepsilon_2(T^*_{\hat{h}}R^*_{2,\hat{h},\hat{v}})^T \end{bmatrix} \begin{bmatrix} \Psi_8^1 & \varepsilon\Psi_8^1 & -((L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})C^*_h)^T & -I & 0 \\ \Psi_8^2 & \varepsilon\Psi_8^2 & -((L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})C^*_{\hat{h}})^T & 0 & -I \end{bmatrix},$$

where $\Psi_8^1 = (\overline{BB}(L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})C^*_h)^T, \Psi_8^2 = (\overline{BB}(L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})C^*_{\hat{h}})^T$, $R_{1,\hat{h},\hat{v}} = \begin{bmatrix} R^1_{1,\hat{h},\hat{v}} & R^2_{1,\hat{h},\hat{v}} \\ R^3_{1,\hat{h},\hat{v}} & R^4_{1,\hat{h},\hat{v}} \end{bmatrix}$, $R_{2,\hat{h},\hat{v}} = \begin{bmatrix} R^1_{2,\hat{h},\hat{v}} & R^2_{2,\hat{h},\hat{v}} \\ R^3_{2,\hat{h},\hat{v}} & R^4_{2,\hat{h},\hat{v}} \end{bmatrix}$, $R^1_{1,\hat{h},\hat{v}} = \begin{bmatrix} R_1 & 0_{p\times(n-p)} \\ R^{11}_{1,\hat{h},\hat{v}} & R^{12}_{1,\hat{h},\hat{v}} \end{bmatrix}$, $R^1_{2,\hat{h},\hat{v}} = \begin{bmatrix} R_1 & 0_{p\times(n-p)} \\ R^{11}_{2,\hat{h},\hat{v}} & R^{12}_{2,\hat{h},\hat{v}} \end{bmatrix}$, $T^*_h = \begin{bmatrix} T_h & 0 \\ 0 & I \end{bmatrix}$, $T^*_{\widehat{h}} = \begin{bmatrix} T_{\widehat{h}} & 0 \\ 0 & I \end{bmatrix}$, $R_1 \in R^{p\times p}$, $R^{11}_{1,\hat{h},\hat{v}} \in R^{(n-p)\times p}$, $R^{12}_{1,\hat{h},\hat{v}} \in R^{(n-p)\times(n-p)}$, $R^{11}_{2,\hat{h},\hat{v}} \in R^{(n-p)\times p}$, $R^{12}_{2,\hat{h},\hat{v}} \in R^{(n-p)\times(n-p)}$, $R^2_{1,\hat{h},\hat{v}} \in R^{n\times n}$, $R^3_{1,\hat{h},\hat{v}} \in R^{n\times n}$, $R^4_{1,\hat{h},\hat{v}} \in R^{n\times n}$, $R^2_{2,\hat{h},\hat{v}} \in R^{n\times n}$, $R^3_{2,\hat{h},\hat{v}} \in R^{n\times n}$, $R^4_{2,\hat{h},\hat{v}} \in R^{n\times n}$, and T_h and $T_{\hat{h}}$ satisfy assumption 3.

Inequality (39) can be rewritten as follows:

$$\Psi_9 + \mathrm{He}(\begin{bmatrix} B^*\Delta F^*_{\hat{h},\hat{v}} & 0 & 0 & 0 & 0 \\ \varepsilon B^*\Delta F^*_{\hat{h},\hat{v}} + (\overline{BB}\varpi_4)^T & \varepsilon(\overline{BB}\varpi_4)^T & -\varpi_4{}^T & 0 & 0 \\ \Delta A^* K_{\hat{h},\hat{v}} + (\overline{BB}\varpi_3)^T & \varepsilon(\overline{BB}\varpi_3)^T & \Delta A^* Q_{\hat{h},\hat{v}} - \varpi_3{}^T & 0 & 0 \\ \varepsilon_1(\overline{BB}\varpi_3)^T & \varepsilon_1\varepsilon(\overline{BB}\varpi_3)^T & -\varepsilon_1\varpi_3{}^T & 0 & 0 \\ \varepsilon_2(\overline{BB}\varpi_4)^T & \varepsilon_2\varepsilon(\overline{BB}\varpi_4)^T & -\varepsilon_2\varpi_4{}^T & 0 & 0 \end{bmatrix}) < 0, \tag{40}$$

$$\Psi_9 = \begin{bmatrix} \Psi_9^1 & * & * & * & * \\ \Psi_9^2 & \Psi_9^3 & * & * & * \\ \Psi_9^4 & \Psi_9^5 & \mathrm{He}(A^*_{h,v}Q_{\hat{h},\hat{v}} - \varpi_1) & * & * \\ \Psi_9^6 & \Psi_9^7 & -T^*_h R_{1,\hat{h},\hat{v}} + Q_{\hat{h},\hat{v}} - \varepsilon_1\varpi_1^T & -\varepsilon_1\mathrm{He}(T^*_h R_{1,\hat{h},\hat{v}})^T & * \\ \Psi_9^8 & \Psi_9^9 & -\varepsilon_2\varpi_2^T & 0 & -\varepsilon_2\mathrm{He}(T^*_{\hat{h}}R_{2,\hat{h},\hat{v}})^T \end{bmatrix},$$

where $\Psi_9^1 = \mathrm{He}(A^*_{\hat{h},\hat{v}}K_{\hat{h},\hat{v}} + B^*F^*_{\hat{h},\hat{v}})$, $\Psi_9^2 = (\overline{BB}\varpi_2)^T + \varepsilon A^*_{\hat{h},\hat{v}}K_{\hat{h},\hat{v}} - P_{\hat{h},\hat{v}} + K_{\hat{h},\hat{v}} + \varepsilon B^*F^*_{\hat{h},\hat{v}}$, $\Psi_9^3 = -\varepsilon\mathrm{He}(P_{\hat{h},\hat{v}} - \overline{BB}\varpi_2)$, $\Psi_9^4 = (\overline{BB}\varpi_1)^T + A^*_{h,v}K_{\hat{h},\hat{v}} - A^*_{\hat{h},\hat{v}}K_{\hat{h},\hat{v}}$, $\Psi_9^5 = \varepsilon(\overline{BB}\varpi_1)^T - L^*_{\hat{h},\hat{v}}C^*_{\hat{h}}T^*_{\hat{h}}R_{2\hat{h},\hat{v}}$, $\Psi_9^6 = K_{\hat{h},\hat{v}} + \varepsilon_1(\overline{BB}\varpi_1)^T$, $\Psi_9^7 = \varepsilon_1\varepsilon(\overline{BB}\varpi_1)^T$, $\Psi_9^8 = -K_{\hat{h},\hat{v}} + \varepsilon_2(\overline{BB}\varpi_2)^T$, $\Psi_9^9 = \varepsilon_2\varepsilon(\overline{BB}\varpi_2)^T - T^*_{\hat{h}}R_{2,\hat{h},\hat{v}}$, $\varpi_1 = L^*_{\hat{h},\hat{v}}C^*_h T^*_h R_{1,\hat{h},\hat{v}}$, $\varpi_2 = L^*_{\hat{h},\hat{v}}C^*_{\hat{h}}T^*_{\hat{h}}R_{2,\hat{h},\hat{v}}$, $\varpi_3 = \Delta L^*_{\hat{h},\hat{v}}C^*_h T^*_h R_{1,\hat{h},\hat{v}}$, and $\varpi_4 = \Delta L^*_{\hat{h},\hat{v}}C^*_{\hat{h}}T^*_{\hat{h}}R_{2,\hat{h},\hat{v}}$.

Because $\Delta L^*_{\hat{h},\hat{v}} = M^*_L S_L N^*_L L^*_{\hat{h},\hat{v}}$, $\Delta F^*_{\hat{h},\hat{v}} = M_F S_F N_F F^*_{\hat{h},\hat{v}}$, and $\Delta A^* = MF(t)N$, inequality (40) can be rewritten as:

$$\Psi_9 + \mathrm{He}(\mathrm{M_1 S_F N_1 + M_2 F(t) N_2 + M_3 S_L N_3}) < 0, \tag{41}$$

$$M_1 = \begin{bmatrix} B^*M_F & \varepsilon B^*M_F & 0 & 0 & 0 \end{bmatrix}^T, N_1 = \begin{bmatrix} N_F F^*_{\hat{h},\hat{v}} & 0 & 0 & 0 & 0 \end{bmatrix},\ M_2 = \begin{bmatrix} 0 & 0 & M & 0 & 0 \end{bmatrix}^T,$$

$$N_2 = \begin{bmatrix} NK_{\hat{h},\hat{v}} & 0 & NQ_{\hat{h},\hat{v}} & 0 & 0 \end{bmatrix}, M_3 = \begin{bmatrix} \overline{BB}M^*_L & \varepsilon\overline{BB}M^*_L & -M^*_L & 0 & 0 \end{bmatrix}^T,$$

$$N_3 = \begin{bmatrix} 0 & N^*_L L^*_{\hat{h},\hat{v}} C^*_{\hat{h}} T^*_{\hat{h}} R_{2,\hat{h},\hat{v}} & N^*_L L^*_{\hat{h},\hat{v}} C^*_h T^*_h R_{1,\hat{h},\hat{v}} & \varepsilon_1 N^*_L L^*_{\hat{h},\hat{v}} C^*_h T^*_h R_{1,\hat{h},\hat{v}} & \varepsilon_2 N^*_L L^*_{\hat{h},\hat{v}} C^* T^* R_{1,\hat{h},\hat{v}} \end{bmatrix}.$$

Based on lemma 2, (41) holds if there are positive real scalars $\varepsilon_{3,\hat{h},\hat{v}}$, $\varepsilon^A_{4,\hat{h},\hat{v}}$, $\varepsilon^E_{4,\hat{h},\hat{v}}$, and $\varepsilon_{5,\hat{h},\hat{v}}$that satisfy the following relation:

$$\Psi_9 + M_1\varepsilon_{3,\hat{h},\hat{v}}M_1^T + N_1^T\varepsilon^{-1}_{3,\hat{h},\hat{v}}N_1 + M_2\varepsilon_{4,\hat{h},\hat{v}}M_2^T + N_2^T\varepsilon^{-1}_{4,\hat{h},\hat{v}}N_2 + M_3\varepsilon_{5,\hat{h},\hat{v}}M_3^T + N_3^T\varepsilon^{-1}_{5,\hat{h},\hat{v}}N_3 < 0, \tag{42}$$

where $\varepsilon_{4,\hat{h},\hat{v}} = \begin{bmatrix} \varepsilon^A_{4,\hat{h},\hat{v}}I & 0 \\ 0 & \varepsilon^E_{4,\hat{h},\hat{v}}I \end{bmatrix}$. Using the Schur complement in lemma 1, (42) holds if and only if the following relation set is satisfied:

$$\begin{bmatrix} \Psi^1_{10} & * & * & * & * & * & * & * \\ \Psi^2_{10} & \Psi^3_{10} & * & * & * & * & * & * \\ \Psi^4_{10} & \Psi^5_{10} & \Psi^6_{10} & * & * & * & * & * \\ \Psi^7_{10} & \varepsilon_1\varepsilon\left(\overline{BB}\varpi_1\right)^T & \Psi^8_{10} & \Psi^9_{10} & * & * & * & * \\ \Psi^{10}_{10} & \Psi^{11}_{10} & -\varepsilon_2\varpi_2^T & 0 & -\varepsilon_2\mathrm{He}(T^*_{\hat{h}}R_{2,\hat{h},\hat{v}})^T & * & * & * \\ N_F F^*_{\hat{h},\hat{v}} & 0 & 0 & 0 & 0 & -\varepsilon_{3,\hat{h},\hat{v}}I & * & * \\ NK^*_{\hat{h},\hat{v}} & 0 & NQ_{\hat{h},\hat{v}} & 0 & 0 & 0 & -\varepsilon_{4,\hat{h},\hat{v}}I & * \\ 0 & \Psi^{12}_{10} & \Psi^{13}_{10} & \Psi^{14}_{10} & \varepsilon_2 N^*_L L^*_{\hat{h},\hat{v}} C^* T^* R_{1,\hat{h},\hat{v}} & 0 & 0 & -\varepsilon_{5,\hat{h},\hat{v}}I \end{bmatrix} < 0 \tag{43}$$

where $\Psi^1_{10} = \mathrm{He}(A^*_{\hat{h},\hat{v}}K_{\hat{h},\hat{v}} + B^*F^*_{\hat{h},\hat{v}}) + B^*M_F\varepsilon_{3,\hat{h},\hat{v}}M_F^TB^{*T} + \overline{BB}M^*_L\varepsilon_{5,\hat{h},\hat{v}}M^{*T}_L\overline{BB}^T$, $\Psi^2_{10} = \left(\overline{BB}\varpi_2\right)^T + K_{\hat{h},\hat{v}} + \varepsilon A^*_{\hat{h},\hat{v}}K_{\hat{h},\hat{v}} - P_{\hat{h},\hat{v}} + \varepsilon B^*F^*_{\hat{h},\hat{v}} + \varepsilon B^*M_F\varepsilon_{3,\hat{h},\hat{v}}M_F^TB^{*T} + \varepsilon\overline{BB}M^*_L\varepsilon_{5,\hat{h},\hat{v}}M^{*T}_L\overline{BB}^T$, $\Psi^3_{10} = \varepsilon^2B^*M_F\varepsilon_{3,\hat{h},\hat{v}}M_F^TB^{*T} + \varepsilon^2\overline{BB}M^*_L\varepsilon_{5,\hat{h},\hat{v}}M^{*T}_L\overline{BB}^T - \varepsilon\mathrm{He}\left(P^T_{\hat{h},\hat{v}} - \left(\overline{BB}\varpi_2\right)^T\right)$, $\Psi^4_{10} = \left(\overline{BB}\varpi_1\right)^T - M^*_L\varepsilon_{5,\hat{h},\hat{v}}M^{*T}_L\overline{BB}^T + A^*_{h,v}K_{\hat{h},\hat{v}} - A^*_{\hat{h},\hat{v}}K_{\hat{h},\hat{v}}$, $\Psi^5_{10} = \varepsilon\left(\overline{BB}\varpi_1\right)^T - \varepsilon M^*_L\varepsilon_{5,\hat{h},\hat{v}}M^{*T}_L\overline{BB}^T - L^*_{\hat{h},\hat{v}}C^*_{\hat{h}}T^*_{\hat{h}}R_{2\hat{h},\hat{v}}$, $\Psi^6_{10} = \mathrm{He}\left(A^*_{h,v}Q_{\hat{h},\hat{v}} - \varpi_1\right) + M\varepsilon_{4,\hat{h},\hat{v}}M^T + M^*_L\varepsilon_{5,\hat{h},\hat{v}}M^{*T}_L$, $\Psi^7_{10} = K_{\hat{h},\hat{v}} + \varepsilon_1\left(\overline{BB}\varpi_1\right)^T$, $\Psi^8_{10} = Q_{\hat{h},\hat{v}} - \varepsilon_1\varpi_1^T - T^*_hR_{1,\hat{h},\hat{v}}$, $\Psi^9_{10} = -\varepsilon_1\mathrm{He}(T^*_hR_{1,\hat{h},\hat{v}})^T$, $\Psi^{10}_{10} = -K_{\hat{h},\hat{v}} + \varepsilon_2(\overline{BB}\varpi_2)^T$, $\Psi^{11}_{10} = \varepsilon_2\varepsilon(\overline{BB}\varpi_2)^T - T^*_{\hat{h}}R_{2,\hat{h},\hat{v}}$, $\Psi^{12}_{10} = N^*_LL^*_{\hat{h},\hat{v}}C^*_{\hat{h}}T^*_{\hat{h}}R_{2,\hat{h},\hat{v}}$, $\Psi^{13}_{10} = N^*_LL^*_{\hat{h},\hat{v}}C^*_hT^*_hR_{1,\hat{h},\hat{v}}$, $\Psi^{14}_{10} = \varepsilon_1N^*_LL^*_{\hat{h},\hat{v}}C^*_hT^*_hR_{1,\hat{h},\hat{v}}$.

It is assumed that $S^* = \begin{bmatrix} S^*_1 & S^*_2 \end{bmatrix}$; thus, the matrix $\overline{BB}$ can be formulated as:

$$\overline{BB} = \begin{bmatrix} I & 0 \\ -B(S^*_2B)^{-1}S^*_1 & I - B(S^*_2B)^{-1}S^*_2 \end{bmatrix}. \tag{44}$$

Then, the matrices in inequality (43) can be obtained as follows:

$$L^*_{\hat{h},\hat{v}} = \begin{bmatrix} 0 & L^T_{\hat{h},\hat{v}} \end{bmatrix}^T, T^*_h = \begin{bmatrix} T_h & 0 \\ 0 & I \end{bmatrix}, T^*_{\hat{h}} = \begin{bmatrix} T_{\hat{h}} & 0 \\ 0 & I \end{bmatrix},\ T^*_hR_{1,\hat{h},\hat{v}} = \begin{bmatrix} T_hR^1_{1,\hat{h},\hat{v}} & T_hR^2_{1,\hat{h},\hat{v}} \\ R^3_{1,\hat{h},\hat{v}} & R^4_{1,\hat{h},\hat{v}} \end{bmatrix}, *$$

$$L^*_{\hat{h},\hat{v}}C^*_hT^*_hR_{1,\hat{h},\hat{v}} = \begin{bmatrix} 0 & 0 \\ L_{\hat{h},\hat{v}}C_hT_hR^1_{1,\hat{h},\hat{v}} & 0 \end{bmatrix} = \begin{bmatrix} 0_{n\times n} & 0_{n\times n} \\ \begin{bmatrix} (L_{\hat{h},\hat{v}}R_1)_{n\times p} & 0_{n\times(n-p)} \end{bmatrix} & 0_{n\times n} \end{bmatrix},$$

$$\overline{B}BL^*_{\hat{h},\hat{v}}C^*_h T^*_h R_{1,\hat{h},\hat{v}} = \begin{bmatrix} 0_{n\times n} & 0_{n\times n} \\ \begin{bmatrix} ((I - B(S_2^* B)^{-1} S_2^*) L_{\hat{h},\hat{v}} R_1)_{n\times p} & 0_{n\times(n-p)} \end{bmatrix} & 0_{n\times n}, \end{bmatrix}$$

$$T^*_{\hat{h}} R_{2,\hat{h},\hat{v}} = \begin{bmatrix} T_{\hat{h}} R^1_{2,\hat{h},\hat{v}} & T_{\hat{h}} R^2_{2,\hat{h},\hat{v}} \\ R^3_{2,\hat{h},\hat{v}} & R^4_{2,\hat{h},\hat{v}} \end{bmatrix},$$

$$L^*_{\hat{h},\hat{v}} C^*_{\hat{h}} T^*_{\hat{h}} R_{2,\hat{h},\hat{v}} = \begin{bmatrix} 0 & 0 \\ L_{\hat{h},\hat{v}} C_{\hat{h}} T_{\hat{h}} R^1_{2,\hat{h},\hat{v}} & 0 \end{bmatrix} = \begin{bmatrix} 0_{n\times n} & 0_{n\times n} \\ \begin{bmatrix} (L_{\hat{h},\hat{v}} R_1)_{n\times p} & 0_{n\times(n-p)} \end{bmatrix} & 0_{n\times n} \end{bmatrix},$$

$$\overline{B}BL^*_{\hat{h},\hat{v}}C^*_h T^*_h R_{2,\hat{h},\hat{v}} = \begin{bmatrix} 0_{n\times n} & 0_{n\times n} \\ \begin{bmatrix} ((I - B(S_2^* B)^{-1} S_2^*) L_{\hat{h},\hat{v}} R_1)_{n\times p} & 0_{n\times(n-p)} \end{bmatrix} & 0_{n\times n} \end{bmatrix}.$$

Therefore, inequality (27) is easily obtained from the above equalities and inequality (43), which completes the proof.

In some special cases, T-S fuzzy descriptor systems may not contain uncertainties, so the following theorem gives sufficient LMI conditions for the stability of the closed-loop system in this case.

Theorem 3: Suppose that the T-S fuzzy descriptor system matrices in (2) satisfy ΔA^*=0 and$C_h = C_{\hat{h}} = C^*$. Given the constants $\varepsilon > 0$ and $\varepsilon_1 > 0$, the closed-loop systems are asymptotically stable if positive definite matrices P_1 and Q_1, scalars ε and ε_1, and a set of matrices $P_{3,i,k}$, $P_{4,i,k}$, $F_{i,k}$, $K_{1,k}$, $K_{2,k}$, $K_{3,i}$, $K_{4,i}$, $L_{i,k}$, R_1, $R^{11}_{1,i,k}$, $R^{12}_{1,i,k}$, $R^2_{1,i,k}$, $R^3_{1,i,k}$, and $R^4_{1,i,k}$exist, where $k \in \{1, \cdots, r_e\}$ and $i \in \{1, \cdots, r\}$, such that the following LMIs hold:

$$\begin{bmatrix} \mathrm{He}(K_{3,\hat{h}}) & * & * & * & * & * & * & * \\ \Lambda_{2,1} & \Lambda_{2,2} & * & * & * & * & * & * \\ \Lambda_{3,1} & \Lambda_{3,2} & -\varepsilon \mathrm{He}(P_1) & * & * & * & * & * \\ \Lambda_{4,1} & \Lambda_{4,2} & -\varepsilon P_{3,\hat{h},\hat{v}} & -\varepsilon \mathrm{He}(P_{4,\hat{h},\hat{v}}) & * & * & * & * \\ 0 & \Lambda_{5,2} & 0 & \varepsilon(\overline{B}Y_{\hat{h},\hat{v}})^T & \mathrm{He}(Q_{3,\hat{h}}) & * & * & * \\ \Lambda_{6,1} & \Lambda_{6,2} & 0 & 0 & \Lambda_{6,5} & \Lambda_{6,6} & * & * \\ 0 & \Lambda_{7,2} & 0 & \varepsilon_1 \varepsilon(\overline{B}Y_{\hat{h},\hat{v}})^T & \Lambda_{7,5} & \Lambda_{7,6} & \Lambda_{7,7} & * \\ 0 & 0 & 0 & 0 & \Lambda_{8,5} & \Lambda_{8,6} & \Lambda_{8,7} & \Lambda_{8,8} \end{bmatrix} < 0, \tag{45}$$

where $\Lambda_{2,1} = A_{\hat{h}} K_{1,\hat{v}} - E_{\hat{v}} K_{3,\hat{h}} + BF_{\hat{h},\hat{v}} + K^T_{4,\hat{h}}$, $\Lambda_{2,2} = \mathrm{He}(A_{\hat{h}} K_{2,\hat{v}} - E_{\hat{v}} K_{4,\hat{h}})$, $\Lambda_{3,1} = K_{1,\hat{v}} - P_1 + \varepsilon K_{3,\hat{h}}$, $\Lambda_{3,2} = K_{2,\hat{v}} + \varepsilon K_{4,\hat{h}}$, $\Lambda_{4,1} = K_{3,\hat{h}} + \varepsilon(A_{\hat{h}} K_{1,\hat{v}} - E_{\hat{v}} K_{3,\hat{h}} + BF_{\hat{h},\hat{v}}) - P_{3,\hat{h},\hat{v}}$, $\Lambda_{4,2} = \varepsilon(A_{\hat{h}} K_{2,\hat{v}} - E_{\hat{v}} K_{4,\hat{h}}) + K_{4\hat{h}} - P_{4,\hat{h},\hat{v}}$, $\Lambda_{5,2} = (\overline{B}Y_{\hat{h},\hat{v}})^T$, $\Lambda_{6,1} = (A_h - A_{\hat{h}})K_{1,\hat{v}} - (E_v - E_{\hat{v}})K_{3,\hat{h}}$, $\Lambda_{6,2} = (A_h - A_{\hat{h}})K_{2,\hat{v}} - (E_v - E_{\hat{v}})K_{4,\hat{h}}$, $\Lambda_{6,5} = A_h Q_1 - E_v Q_{3,\hat{h}} + Q^T_{4,\hat{h}} - Y_{\hat{h},\hat{v}}$, $\Lambda_{6,6} = -\mathrm{He}(E_v Q_{4,\hat{h}})$, $\Lambda_{7,2} = \varepsilon_1(\overline{B}Y_{\hat{h},\hat{v}})^T$, $\Lambda_{7,5} = Q_1 - TR^1_{1,\hat{h},\hat{v}}$, $\Lambda_{7,6} = -\varepsilon_1(Y_{\hat{h},\hat{v}})^T - TR^2_{1,\hat{h},\hat{v}}$, $\Lambda_{7,7} = -\varepsilon_1 \mathrm{He}(TR^1_{1,\hat{h},\hat{v}})$, $\Lambda_{8,5} = Q_{3,\hat{h}} - R^3_{1,\hat{h},\hat{v}}$, $\Lambda_{8,6} = Q_{4,\hat{h}} - R^4_{1,\hat{h},\hat{v}}$, $\Lambda_{8,7} = -\varepsilon_1(TR^2_{1,\hat{h},\hat{v}})^T - \varepsilon_1 R^3_{1,\hat{h},\hat{v}}$, $\Lambda_{8,8} = -\varepsilon_1 \mathrm{He}(R^4_{1,\hat{h},\hat{v}})$, $\overline{B} = I - B(S_2^* B)^{-1} S_2^*$, $R^1_{1,\hat{h},\hat{v}} = \begin{bmatrix} R_1 & 0_{p\times(n-p)} \\ R^{11}_{1,\hat{h},\hat{v}} & R^{12}_{1,\hat{h},\hat{v}} \end{bmatrix}$,$R^2_{1,\hat{h},\hat{v}} = \begin{bmatrix} 0_{p\times n} \\ R^{21}_{1,\hat{h},\hat{v}} \end{bmatrix}$, and $Y_{\hat{h},\hat{v}} = \begin{bmatrix} (L_{\hat{h},\hat{v}} R_1)_{n\times p} & 0_{n\times(n-p)} \end{bmatrix}$.

Theorem 4: Suppose that the T-S fuzzy descriptor system matrices in (2) satisfy ΔA^*=0 and $C_h \neq C_{\hat{h}}$. Given constants $\varepsilon > 0$, $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$, the closed-loop systems are asymptotically stable if positive definite matrices P_1 and Q_1 and a set of matrices $P_{3,i,k}$, $P_{4,i,k}$, $F_{i,k}$, $K_{1,k}$, $K_{2,k}$, $K_{3,i}$, $K_{4,i}$, $L_{i,k}$, R_1, $R^{11}_{1,i,k}$, $R^{12}_{1,i,k}$, $R^{11}_{2,i,k}$, $R^{12}_{2,i,k}$,$R^2_{1,i,k}$, $R^3_{1,i,k}$, $R^4_{1,i,k}$, $R^2_{2,i,k}$, $R^3_{2,i,k}$, and $R^4_{2,i,k}$ exist, where $i \in \{1, \cdots, r\}$ and $k \in \{1, \cdots, r_e\}$, such that the following LMIs hold:

$$\begin{bmatrix} \mathrm{He}(K_{3,\hat{h}}) & * & * & * & * & * & * & * & * & * \\ \Lambda_{2,1} & \Lambda_{2,2} & * & * & * & * & * & * & * & * \\ \Lambda_{3,1} & \Lambda_{3,2} & -\varepsilon \mathrm{He}(P_1) & * & * & * & * & * & * & * \\ \Lambda_{4,1} & \Lambda_{4,2} & \Lambda_{4,3} & \Lambda_{4,4} & * & * & * & * & * & * \\ 0 & \Lambda_{5,2} & 0 & \Lambda_{5,4} & \Lambda_{5,5} & * & * & * & * & * \\ \Lambda_{6,1} & \Lambda_{6,2} & -Y_{\hat{h},\hat{v}} & 0 & \Lambda_{6,5} & \Lambda_{6,6} & * & * & * & * \\ K_{1,\hat{v}} & \Lambda_{7,2} & 0 & \Lambda_{7,4} & \Lambda_{7,5} & \Lambda_{7,6} & \Lambda_{7,7} & * & * & * \\ K_{3,\hat{h}} & K_{4,\hat{h}} & 0 & 0 & \Lambda_{8,5} & \Lambda_{8,6} & \Lambda_{8,7} & \Lambda_{8,8} & * & * \\ K_{1,\hat{v}} & \Lambda_{9,2} & \Lambda_{9,3} & \Lambda_{9,4} & 0 & \Lambda_{9,6} & 0 & 0 & \Lambda_{9,9} & * \\ K_{3,\hat{h}} & K_{4,\hat{h}} & -R^3_{2,\hat{h},\hat{v}} & \Lambda_{10,4} & 0 & 0 & 0 & 0 & \Lambda_{10,9} & \Lambda_{10,10} \end{bmatrix} < 0, \tag{46}$$

where $\Lambda_{2,1} = A_{\hat{h}}K_{1,\hat{v}} - E_{\hat{v}}K_{3,\hat{h}} + BF_{\hat{h},\hat{v}} + K^T_{4,\hat{h}}$, $\Lambda_{2,2} = \mathrm{He}(A_{\hat{h}}K_{2,\hat{v}} - E_{\hat{v}}K_{4,\hat{h}})$, $\Lambda_{3,1} = K_{1,\hat{v}} + \varepsilon K_{3,\hat{h}} - P_1$, $\Lambda_{3,2} = K_{2,\hat{v}} + \varepsilon K_{4,\hat{h}} + (\overline{B}Y_{\hat{h},\hat{v}})^T$, $\Lambda_{4,1} = K_{3,\hat{h}} + \varepsilon(A_{\hat{h}}K_{1,\hat{v}} - E_{\hat{v}}K_{3,\hat{h}} + BF_{\hat{h},\hat{v}}) - P_{3,\hat{h},\hat{v}}$, $\Lambda_{4,2} = K_{4,\hat{h}} - P_{4,\hat{h},\hat{v}} + \varepsilon A_{\hat{h}}K_{2,\hat{v}} - \varepsilon E_{\hat{v}}K_{4,\hat{h}}$, $\Lambda_{4,3} = -\varepsilon P_{3,\hat{h},\hat{v}} + \varepsilon\overline{B}Y_{\hat{h},\hat{v}}$, $\Lambda_{4,4} = -\varepsilon \mathrm{He}(P_{4,\hat{h},\hat{v}})$, $\Lambda_{5,2} = (\overline{B}Y_{\hat{h},\hat{v}})^T$, $\Lambda_{5,4} = \varepsilon(\overline{B}Y_{\hat{h},\hat{v}})^T$,$\Lambda_{5,5} = \mathrm{He}(Q_{3,\hat{h}})$, $\Lambda_{6,1} = (A_h - A_{\hat{h}})K_{1,\hat{v}} - (E_v - E_{\hat{v}})K_{3,\hat{h}}$, $\Lambda_{6,2} = (A_h - A_{\hat{h}})K_{2,\hat{v}} - (E_v - E_{\hat{v}})K_{4,\hat{h}}$, $\Lambda_{6,5} = A_hQ_1 - E_vQ_{3,\hat{h}} + Q^T_{4,\hat{h}} - Y_{\hat{h},\hat{v}}$, $\Lambda_{6,6} = -\mathrm{He}(E_vQ_{4,\hat{h}})$, $\Lambda_{7,2} = K_{2\hat{v}} + \varepsilon_1(\overline{B}Y_{\hat{h},\hat{v}})^T$,$\Lambda_{7,4} = \varepsilon_1\varepsilon(\overline{B}Y_{\hat{h},\hat{v}})^T$, $\Lambda_{7,5} = Q_1 - T_hR^1_{1,\hat{h},\hat{v}}$, $\Lambda_{7,6} = -\varepsilon_1(Y_{\hat{h},\hat{v}})^T - T_hR^2_{1,\hat{h},\hat{v}}$, $\Lambda_{7,7} = -\varepsilon_1 \mathrm{He}(T_hR^1_{1,\hat{h},\hat{v}})$, $\Lambda_{8,5} = Q_{3,\hat{h}} - R^3_{1,\hat{h},\hat{v}}$, $\Lambda_{8,6} = Q_{4,\hat{h}} - R^4_{1,\hat{h},\hat{v}}$, $\Lambda_{8,7} = -\varepsilon_1R^3_{1,\hat{h},\hat{v}} - \varepsilon_1\left(T_hR^2_{1,\hat{h},\hat{v}}\right)^T$, $\Lambda_{8,8} = -\varepsilon_1 \mathrm{He}(R^4_{1,\hat{h},\hat{v}})$, $\Lambda_{9,2} = K_{2,\hat{v}} + \varepsilon_2(\overline{B}\overline{B}Y_{\hat{h},\hat{v}})^T$, $\Lambda_{9,3} = -T_{\hat{h}}R^1_{2,\hat{h},\hat{v}}$, $\Lambda_{9,4} = \varepsilon_2\varepsilon(\overline{B}Y_{\hat{h},\hat{v}})^T - T_{\hat{h}}R^2_{2,\hat{h},\hat{v}}$, $\Lambda_{9,6} = -\varepsilon_2(Y_{\hat{h},\hat{v}})^T$, $\Lambda_{9,9} = -\varepsilon_2 \mathrm{He}(T_{\hat{h}}R^1_{2,\hat{h},\hat{v}})$, $\Lambda_{10,4} = -R^4_{2,\hat{h},\hat{v}}$,$\Lambda_{10,9} = -\varepsilon_2(T_{\hat{h}}R^2_{2,\hat{h},\hat{v}})^T - \varepsilon_2R^3_{2,\hat{h},\hat{v}}$, $\Lambda_{10,10} = -\varepsilon_2 \mathrm{He}(T_{\hat{h}}R^4_{2,\hat{h},\hat{v}})$, $\overline{B} = I - B(S_2^*B)^{-1}S_2^*$, $R^1_{1,\hat{h},\hat{v}} = \begin{bmatrix} R_1 & 0_{p\times(n-p)} \\ R^{11}_{1,\hat{h},\hat{v}} & R^{12}_{1,\hat{h},\hat{v}} \end{bmatrix}$, $R^1_{2,\hat{h},\hat{v}} = \begin{bmatrix} R_1 & 0_{p\times(n-p)} \\ R^{11}_{2,\hat{h},\hat{v}} & R^{12}_{2,\hat{h},\hat{v}} \end{bmatrix}$, $R^2_{1,\hat{h},\hat{v}} = \begin{bmatrix} 0_{p\times n} \\ R^{21}_{1,\hat{h},\hat{v}} \end{bmatrix}$, $R^2_{2,\hat{h},\hat{v}} = \begin{bmatrix} 0_{p\times n} \\ R^{21}_{2,\hat{h},\hat{v}} \end{bmatrix}$, and $Y_{\hat{h},\hat{v}} = \begin{bmatrix} (L_{\hat{h},\hat{v}}R_1)_{n\times p} & 0_{n\times(n-p)} \end{bmatrix}$.

The corresponding proof can be obtained according to the proofs given for theorems 1 and 2, because theorems 3 and 4 are special cases of theorems 1 and 2, respectively. Therefore, the specific process is omitted here.

Non-Fragile Observer-Based Adaptive Integral Sliding Mode Controller Design

In practical applications, it is difficult to accurately obtain the bounds of unknown uncertainties and perturbations in controller and observer gains. Hence, an adaptive integral sliding mode controller is designed for T-S fuzzy descriptor systems with uncertainties and perturbations.

Theorem 5: Suppose that the T-S fuzzy descriptor system matrices in (2) satisfy $\Delta A^* \neq 0$, and $C_h = C_{\hat{h}} = C^*$. Assume that matrices $F_{i,k}$, $K_{1,k}$, $K_{2,k}$, $K_{3,i}$, $K_{4,i}$, and $L_{i,k}$, where $k \in \{1, \cdots, r_e\}$ and $i \in \{1, \cdots, r\}$, satisfy theorem 1 and $\zeta > 0$. System (2) can be driven to the sliding surface (7) and maintain sliding motion based on the following ISMC equation:

$$u(t) = F^*_{\hat{h},\hat{v}}K^{-1}_{\hat{h},\hat{v}}\hat{X}^*(t) - (S^*B^*)^{-1}(\left\|S^*L^*_{\hat{h},\hat{v}}(y(t) - C^*\hat{X}^*(t))\right\| + \left\|S^*M\right\| \left\|N\hat{X}^*(t)\right\| +$$

$$\left\|S^*M_L\right\| \left\|N_LL^*_{\hat{h},\hat{v}}(y(t) - C^*\hat{X}^*(t))\right\| + \zeta)\frac{s_{\hat{X}^*(t)}}{\left\|s_{\hat{X}^*(t)}\right\|} - \|M_F\| \left\|N_FF^*_{\hat{h},\hat{v}}K^{-1}_{\hat{h},\hat{v}}\hat{X}^*(t)\right\| \frac{s_{\hat{X}^*(t)}}{\left\|s_{\hat{X}^*(t)}\right\|}. \tag{47}$$

Proof: Consider the following Lyapunov function candidate:

$$V(s_{\hat{X}^*}(t)) = \frac{1}{2}s^T_{\hat{X}^*}(t)s_{\hat{X}^*}(t). \tag{48}$$

The derivative of the Lyapunov function candidate (48) can be obtained as:

$$\dot{V}(s_{\hat{X}^*}(t)) = s^T_{\hat{X}^*}(t)\dot{s}_{\hat{X}^*}(t)$$

$$= s^T_{\hat{X}^*}(t)\left\{S^*(B^*u(t) + (L^*_{\hat{h},\hat{v}} + \Delta L^*_{\hat{h},\hat{v}})(y(t) - C^*\hat{X}^*(t))) - S^*B^*(F^*_{\hat{h},\hat{v}} + \Delta F^*_{\hat{h},\hat{v}})K^{-1}_{\hat{h},\hat{v}}\hat{X}^*(t)\right\}. \tag{49}$$

By substituting (47) to (49), Equation (49) can be rewritten as

$$\begin{aligned}\dot{V}(s_{\hat{X}^*}(t)) = s_{\hat{X}^*}^T(t)\Big\{&S^*((L_{\hat{h},\hat{v}}^* + \Delta L_{\hat{h},\hat{v}}^*)(y(t) - C^*\hat{X}^*(t))) - S^*B^*\Delta F_{\hat{h},\hat{v}}^* K_{\hat{h},\hat{v}}^{-1}\hat{X}^*(t) + \\ &S^*B^*\left\|M_F\right\|\left\|N_F F_{\hat{h},\hat{v}} K_{\hat{h},\hat{v}}^{-1}\hat{X}^*(t)\right\|\frac{s_{\hat{X}^*}(t)}{\left\|s_{\hat{X}^*}(t)\right\|} - \left\|S^* L_{\hat{h},\hat{v}}^*(y(t) - C^*\hat{X}^*(t))\right\|\frac{s_{\hat{X}^*}(t)}{\left\|s_{\hat{X}^*}(t)\right\|} - \\ &\left\|S^* M\right\|\left\|N\hat{X}^*(t)\right\|\frac{s_{\hat{X}^*(t)}}{\left\|s_{\hat{X}^*(t)}\right\|} - \left\|S^* M_L\right\|\left\|N_L L_{\hat{h},\hat{v}}^*(y(t) - C^*\hat{X}^*(t))\right\|\frac{s_{\hat{X}^*(t)}}{\left\|s_{\hat{X}^*(t)}\right\|}\Big\}.\end{aligned} \tag{50}$$

Then, from $\Delta A^* = MF(t)N$, $F^T(t)F(t) \leq I$, $\Delta L_{\hat{h},\hat{v}}^* = M_L^* S_L N_L^* L_{\hat{h},\hat{v}}^*$, $S_L^T S_L \leq I$, $\Delta F_{\hat{h},\hat{v}}^* = M_F S_F N_F F_{\hat{h},\hat{v}}^*$, and $S_F^T S_F \leq I$, it follows that:

$$\dot{V}(s_{\hat{X}}(t)) \leq s_{\hat{X}}^T(t)\left\{-\zeta S^* B^* \frac{(S^*B^*)^{-1} s_{\hat{X}^*}(t)}{\left\|s_{\hat{X}^*}(t)\right\|}\right\} \leq -\zeta \left\|s_{\hat{X}^*}(t)\right\|. \tag{51}$$

From the above analysis, T-S fuzzy descriptor systems can reach the desired sliding mode surface in finite time even in the presence of uncertainties and external disturbances.

In some special cases, T-S fuzzy descriptor systems may not contain uncertainties, so the following theorem gives an observer-designed ISMC strategy in this case.

Theorem 6: Suppose that the T-S fuzzy descriptor system matrices in (2) satisfy $\Delta A^* = 0$ and $C_h = C_{\hat{h}} = C^*$. Additionally, assume that matrices $F_{i,k}$, $K_{1,k}$, $K_{2,k}$, $K_{3,i}$, $K_{4,i}$, and $L_{i,k}$, where $k \in \{1, \cdots, r_e\}$ and $i \in \{1, \cdots, r\}$, satisfy theorem 3 and that $\zeta > 0$. System (2) can be driven to the sliding surface and maintain sliding motion based on the following ISMC equation:

$$u(t) = F_{\hat{h},\hat{v}}^* K_{\hat{h},\hat{v}}^{-1}\hat{X}^*(t) - (S^*B^*)^{-1}(\left\|S^* L_{\hat{h},\hat{v}}^*(y(t) - C^*\hat{X}^*(t))\right\| + \zeta)\frac{s_{\hat{X}^*}(t)}{\left\|s_{\hat{X}^*}(t)\right\|}. \tag{52}$$

The corresponding proof can be obtained according to the proofs given for theorem 5, because theorem 6 is a special case of theorem 5. Therefore, the specific process is omitted here.

EXAMPLES

In this section, a simple numerical example is simulated to verify the effectiveness and superiority of the proposed method. Consider the following T-S fuzzy descriptor system:

$$\sum\nolimits_{k=1}^{2} v_k(\xi(t))(E_k + \Delta E)\dot{x}(t) = \sum\nolimits_{i=1}^{2} h_i(z(t))((A_i + \Delta A)x(t) + Bu(t)), \tag{53}$$

where $E_1 = \begin{bmatrix} 1.1 & -0.1 \\ -0.2 + b & 1.5 \end{bmatrix}$, $E_2 = \begin{bmatrix} 0.9 & -0.1 \\ 0.2 & 0.2 \end{bmatrix}$, $A_1 = \begin{bmatrix} -0.2 & -1 \\ -0.1 & -1.9 \end{bmatrix}$, $A_2 = \begin{bmatrix} 1 + a & 0.6 \\ 1.7 & -0.3 \end{bmatrix}$,$B = \begin{bmatrix} 0.7 \\ 0 \end{bmatrix}$, $C = \begin{bmatrix} 1 & 0 \end{bmatrix}$, $a \in [-10, 2]$, $b \in [-0.6, 1]$,$x(t) = \begin{bmatrix} x_1(t) & x_2(t) \end{bmatrix}^T$,$h_1(z(t)) = x_2^2(t)/4$, $h_2(z(t)) = 1 - h_1(z(t))$, $v_1(\xi(t)) = 1/(1 + x_2^2(t))$, and $v_2(\xi(t)) = 1 - v_1(\xi(t))$.

Case 1: $\Delta E = 0 \text{and} \Delta A = 0$

The state feedback controller in (Lin et al., 2006) and the sliding mode controller in (Kchaou et al., 2014) for a class of fuzzy descriptor systems are designed under conditions for which the system states must be measured. Moreover, in (Guerra et al., 2015; Li et al., 2018), the premise variables were dependent on measurable vectors, e.g., the system states $x_1(t)$ and$x_2(t)$ based on the observer strategy used. However, it is challenging to directly obtain the value of $x_2(t)$ with sensors. Therefore, the methods in (Lin et al., 2006; Kchaou et al., 2014; Guerra et al., 2015; Li et al., 2018) cannot be directly applied to T-S fuzzy descriptor systems with unmeasurable system states and premise variables. Several combinations of a and b are selected to compare the feasible solution region size of theorem 3 with those of theorems 2 (Ichalal et al., 2011) and 1 (Asemani and Majd, 2013), as shown in **Figures 2A,B**. These Figures show that the H∞ control method (Asemani and Majd, 2013) is less conservative than the PDC control method based on the traditional quadratic Lyapunov function (Ichalal et al., 2011) for a T-S fuzzy system. The feasible area of the proposed observer-based non-PDC ISMC method for a T-S fuzzy descriptor system is larger than that of the previous two methods. Therefore, in the non-PDC ISMC method, the fuzzy Lyapunov function and descriptor redundancy lead to less conservative results.

When $\Delta A^{*} = 0$, the following values are set: $a = 1$, $b = 0.5$, and $\varepsilon = 0.0001$. Based on theorem 3 and the MATLAB LMI toolbox, the coefficient matrices of the observer-based non-PDC integral sliding mode controller are obtained as follows:

$$L_{11} = \begin{bmatrix} 34.8381 & 4.3355 \end{bmatrix}^{T}; L_{12} = \begin{bmatrix} 34.4535 & 1.2693 \end{bmatrix}^{T};$$
$$L_{21} = \begin{bmatrix} 31.8954 & 2.2550 \end{bmatrix}^{T}; L_{22} = \begin{bmatrix} 31.3404 & 0.4542 \end{bmatrix}^{T};$$
$$F_{11} = \begin{bmatrix} -0.0306 & 0.0082 \end{bmatrix}; F_{12} = \begin{bmatrix} -0.0200 & 0.0024 \end{bmatrix};$$
$$F_{21} = \begin{bmatrix} -0.0280 & 0.0039 \end{bmatrix}; F_{22} = \begin{bmatrix} -0.0175 & 0.0051 \end{bmatrix};$$

$$K_{11} = \begin{bmatrix} 0.0006 & -0.0007 \\ -0.0007 & 0.0022 \end{bmatrix}; K_{12} = \begin{bmatrix} 0.0006 & -0.0007 \\ -0.0007 & 0.0022 \end{bmatrix};$$
$$K_{21} = \begin{bmatrix} -1.3053e^{-6} & -1.9676e^{-7} \\ -1.2372e^{-7} & 9.3139e^{-8} \end{bmatrix};$$
$$K_{22} = \begin{bmatrix} -1.9118e^{-6} & -2.4874e^{-7} \\ 1.4374e^{-8} & 2.4554e^{-7} \end{bmatrix};$$
$$K_{31} = \begin{bmatrix} -0.0031 & 0.0005 \\ 0.0008 & -0.0033 \end{bmatrix}; K_{32} = \begin{bmatrix} -0.0031 & 0.0004 \\ 0.0005 & -0.0018 \end{bmatrix};$$
$$K_{41} = \begin{bmatrix} 0.0181 & -0.0007 \\ 3.4515e^{-5} & 0.0040 \end{bmatrix}; K_{42} = \begin{bmatrix} 0.0166 & -0.0006 \\ -0.0011 & 0.0021 \end{bmatrix}.$$

The controller parameters are selected as $\zeta = 0.003 \text{and} S^{*} = \begin{bmatrix} 1 & 1 & 0.7 & 0 \end{bmatrix}$. Assuming the initial states of $x(0) = \begin{bmatrix} x_1(0) & x_2(0) \end{bmatrix}^{T} = \begin{bmatrix} 0.1 & 0.2 \end{bmatrix}^{T}$and $\hat{x}(0) = \begin{bmatrix} \hat{x}_1(0) & \hat{x}_2(0) \end{bmatrix}^{T} = \begin{bmatrix} -0.5 & -0.4 \end{bmatrix}^{T}$, the state responses, control input, and sliding surface for the closed-loop system are shown in **Figure 3**. The simulation results show that the proposed sliding mode observer can accurately estimate the system state after four s, and that the closed-loop system is asymptotically stable.

Case 2: $\Delta E \neq 0 \text{and} \Delta A \neq 0$.

Assume that the system uncertainties, observer gain perturbation, and controller gain perturbation are expressed by $M_A = \begin{bmatrix} 0.06 & 0.02 \end{bmatrix}^{T}$, $M_E = \begin{bmatrix} 0.04 & 0.01 \end{bmatrix}^{T}$, $N_A = \begin{bmatrix} 0.02 & 0.1 \end{bmatrix}$, $N_E = \begin{bmatrix} 0.03 & 0.05 \end{bmatrix}$, $F_A(t) = 0.5\sin(x_1(t))$, $F_E(t) = 0.2\cos(x_1(t))$, $M_L = \begin{bmatrix} 0.05 & 0.2 \end{bmatrix}^{T}$, $N_L = \begin{bmatrix} 0.01 & 0.12 \end{bmatrix}$, $M_F = 0.1$, $N_F = 0.2$, $S_L = 0.5\sin(x_1(t))$, and $S_F = 2\sin(x_1(t))\cos(x_1(t))$. In this case, the parameters are selected as $a = 1$, $b = 0.5$, $\varepsilon = 0.1$, and $\varepsilon_1 = 0.001$. Moreover, using theorem 1 and the MATLAB LMI toolbox, the following coefficient matrices are obtained:

$$L_{11} = \begin{bmatrix} 42.1615 & 4.6701 \end{bmatrix}^{T}; L_{12} = \begin{bmatrix} 41.5437 & 1.4550 \end{bmatrix}^{T};$$
$$L_{21} = \begin{bmatrix} 39.0960 & 2.5140 \end{bmatrix}^{T}; L_{22} = \begin{bmatrix} 37.9514 & 0.5004 \end{bmatrix}^{T};$$
$$F_{11} = \begin{bmatrix} -0.0349 & 0.0160 \end{bmatrix}; F_{12} = \begin{bmatrix} -0.0237 & 0.0075 \end{bmatrix};$$
$$F_{21} = \begin{bmatrix} -0.0293 & 0.0070 \end{bmatrix}; F_{22} = \begin{bmatrix} -0.0126 & 0.0030 \end{bmatrix};$$

$$K_{11} = \begin{bmatrix} 0.0018 & -0.0015 \\ -0.0015 & 0.0046 \end{bmatrix}; K_{12} = \begin{bmatrix} 0.0018 & -0.0015 \\ -0.0017 & 0.0039 \end{bmatrix};$$
$$K_{21} = \begin{bmatrix} -0.0015 & -0.0004 \\ 0.0001 & 0.0002 \end{bmatrix};$$
$$K_{22} = \begin{bmatrix} -0.0022 & -0.0004 \\ 0.0012 & 0.0006 \end{bmatrix};$$
$$K_{31} = \begin{bmatrix} -0.0058 & 0.0022 \\ 0.0010 & -0.0059 \end{bmatrix}; K_{32} = \begin{bmatrix} -0.0054 & 0.0015 \\ 0.0007 & -0.0031 \end{bmatrix};$$
$$K_{41} = \begin{bmatrix} 0.0227 & -0.0006 \\ 0.0006 & 0.0073 \end{bmatrix}; K_{42} = \begin{bmatrix} 0.0186 & -0.0005 \\ -0.0012 & 0.0038 \end{bmatrix}.$$

It is obvious that different initial system states will lead to different simulation results. Therefore, if the proposed method does not have a wide operating range, the system will not be stable when changing the initial parameters. Therefore, the following initial values of the systems are set to verify the effectiveness of the proposed method for a wide operating range: initial simulation case 1:$x(0) = \begin{bmatrix} 0.1 & 0.2 \end{bmatrix}^{T}$, $\hat{x}(0) = \begin{bmatrix} -0.5 & -0.4 \end{bmatrix}^{T}$; initial simulation case 2: $x(0) = \begin{bmatrix} -0.8 & 0.5 \end{bmatrix}^{T}$, $\hat{x}(0) = \begin{bmatrix} 0.6 & -0.7 \end{bmatrix}^{T}$; initial simulation case 3: $x(0) = \begin{bmatrix} 0.8 & -0.5 \end{bmatrix}^{T}$, $\hat{x}(0) = \begin{bmatrix} -0.6 & 0.7 \end{bmatrix}^{T}$; initial simulation case 4: $x(0) = \begin{bmatrix} -0.5 & -0.7 \end{bmatrix}^{T}$, $\hat{x}(0) = \begin{bmatrix} 0.3 & 0.5 \end{bmatrix}^{T}$.

When the controller parameters are set as $\zeta = 0.003 \text{and} S^{*} = \begin{bmatrix} 1 & 1 & 0.7 & 0 \end{bmatrix}$, the time responses for nonlinear states, the control input, and the sliding mode surface are as shown in **Figures 4–6**, respectively. The simulation results demonstrate that the system state models display good convergence performance; even when the system has uncertain characteristics, the sliding mode observer can accurately estimate the real states for a nonlinear system with different initial state values, and the designed controller has good robustness and is not fragile to system uncertainties, observer perturbations, and controller perturbations.

CONCLUSIONS

The problem of non-fragile observer-based adaptive ISMC for a class of T-S fuzzy descriptor systems with unmeasurable premise variables is considered in this study. For unmeasurable states, a sliding mode observer is designed, and an integral sliding mode surface is constructed considering the features of the fuzzy sliding mode observer system. Using the Lyapunov theory and designing a fuzzy Lyapunov function, sufficient conditions in terms of LMIs are obtained; additionally, asymptotically stable dynamic estimation error and sliding mode dynamics are achieved. An observer-based ISMC strategy is obtained to meet the reachability conditions. Moreover, a non-fragile observer and a non-fragile adaptive controller are

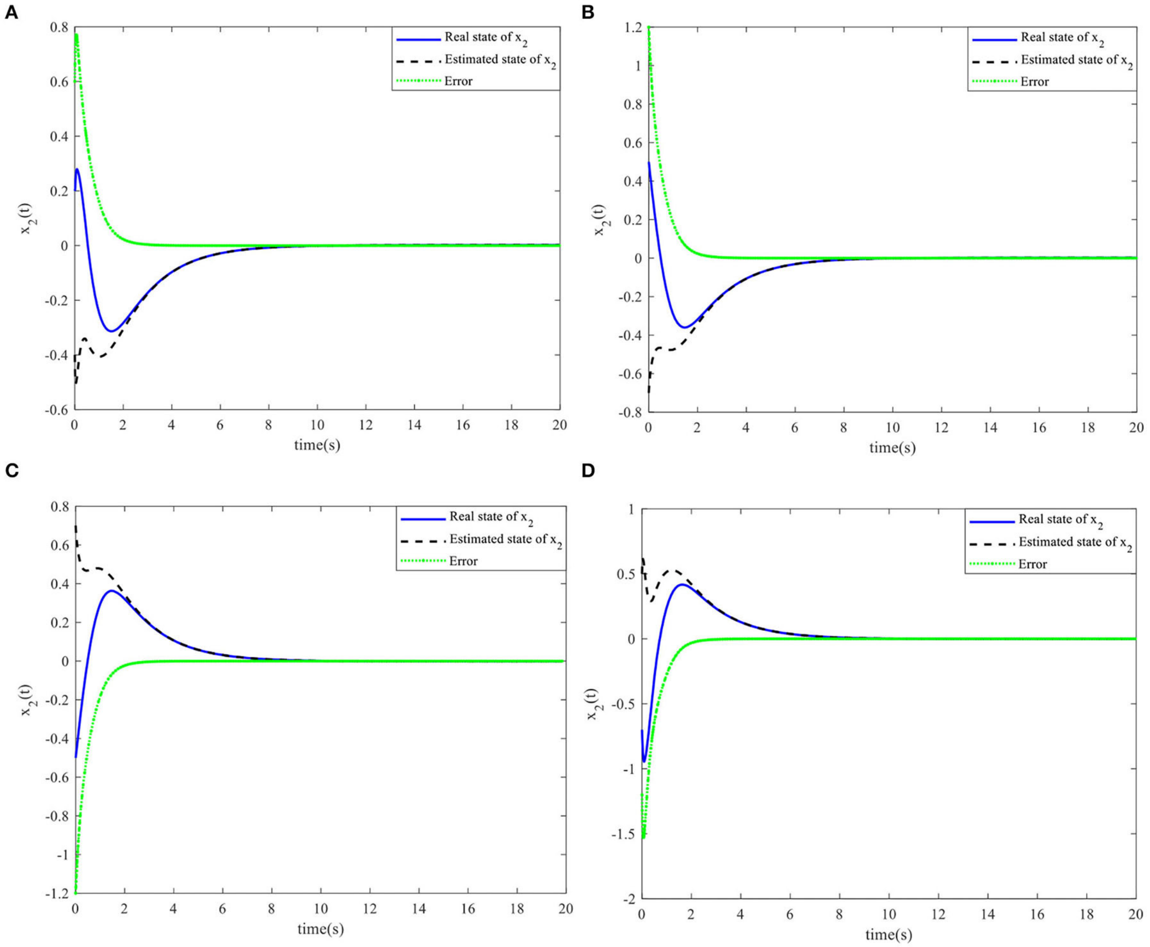


FIGURE 6 | Nonlinear state $x_2(t)$ and the estimated$\hat{x}_2(t)$. **(A)** Initial simulation case 1. **(B)** Initial simulation case 2. **(C)** Initial simulation case 3. **(D)** Initial simulation case 4.

developed such that system uncertainties and perturbations associated with both the observer and the controller can be mitigated. Simulation examples are presented to demonstrate the excellent state estimation performance and effectiveness of the controller.

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

XH: conceptualization, writing (original draft), and software. AR: validation and writing (review and editing). YP: investigation and conceptualization. HG: resources and formal analysis. SS: conceptualization, methodology, and supervision.

FUNDING

This study was supported in part by the Sichuan Science and Technology Program under grant 2020YFG0211, 2022YFG0094, and 2018GZ0110, the SWJTU Basic Research Cultivation Support Program under grant 2682021ZTPY068, the SWJTU Science and Technology Innovation Project under grant 2682022CX008, the Scientific Research Foundation of the Education Department of Sichuan Province under grant 18CZ0017, the National Key R&D Program of China under grant 2018YFB1201603, and the Innovation Fund of Postgraduate of Xihua University under grants YCJJ2021093 and YCJJ2021088.

ACKNOWLEDGMENTS

The authors would like to thank Prof Michio Sugeno, Tokyo Institute of Technology, Japan, for his contribution to this research and the associate editor and reviewers for their insightful comments on how to improve this article.

REFERENCES

Asemani, M. H., and Majd, V. J. A. (2013). robust H∞ observer-based controller design for uncertain T–S fuzzy systems with unknown premise variables via LMI. *Fuzzy Sets Syst.* 212, 21–40. doi: 10.1016/j.fss.2012.07.008

Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V. (1994). Linear matrix inequalities in system and control theory. *Siam.* 185. doi: 10.1137/1.9781611970777

Brahim, I. H., Chaabane, M., and Mehdi, D. (2017). Fault-tolerant control for T–S Fuzzy descriptor systems with sensor faults: an LMI approach. *Int. J. Fuzzy Syst.* 19, 516–527. doi: 10.1007/s40815-016-0154-4

Cervantes, J., Yu, W., Salazar, S., and Chairez, I. (2016). Takagi–Sugeno dynamic neuro-fuzzy controller of uncertain nonlinear systems. *IEEE Trans. Fuzzy Syst.* 25, 1601–1615. doi: 10.1109/TFUZZ.2016.2612697

Chadli, M., and Karimi, H. R. (2012). Robust observer design for unknown inputs Takagi–Sugeno models, *IEEE Trans. Fuzzy Syst.* 21, 158–164. doi: 10.1109/TFUZZ.2012.2197215

Chen, M., and Li, J. (2013). Non-fragile guaranteed cost control for Takagi–Sugeno fuzzy hyperbolic systems. *Int. J. Syst. Sci.* 46, 1–14. doi: 10.1080/00207721.2013.827259

Chen, Y. J., Wang, W. J., and Chang, C. L. (2009). Guaranteed cost control for an overhead crane with practical constraints: fuzzy descriptor system approach. *Eng. Appl. Artif. Intell.* 22, 639-645. doi: 10.1016/j.engappai.2009.03.005

Du, X., and Yang, G. H. (2009). New characterisations of positive realness and static output feedback control of discrete-time systems. *Int. J. Control.* 82, 1485-1495. doi: 10.1080/00207170802549586

Duan, R., Li, J., and Chen, J. (2019). Mode-dependent non-fragile observer-based controller design for fractional-order T–S fuzzy systems with Markovian jump via non-PDC scheme. *Nonlinear Analysis: Hybrid Systems.* 34, 74–91. doi: 10.1016/j.nahs.2019.05.006

Fan, Y., An, Y., Wang, W., and Yang, C. (2020). TS fuzzy adaptive control based on small gain approach for an uncertain robot manipulators. *Int. J. Fuzzy Syst.* 22, 930–942. doi: 10.1007/s40815-019-00793-w

Gahinet, P., and Apkarian, P. (1994). A linear matrix inequality approach to H∞ control. *Int. J. Robust Nonlinear Control.* 4, 421–448. doi: 10.1002/rnc.4590040403

Guan, W., and Liu, F. (2016). Non-fragile fuzzy dissipative static output feedback control for Markovian jump systems subject to actuator saturation. *Neurocomputing.* 193, 123–132. doi: 10.1016/j.neucom.2016.02.004

Guerra, T. M., Estrada-Manzo, V., and Lendek, Z. (2015). Observer design for Takagi–Sugeno descriptor models: An LMI approach. *Automatica.* 52, 154–159. doi: 10.1016/j.automatica.2014.11.008

Haj Brahim, I., Mehdi, D., and Chaabane, M. (2019). Sensor fault and state estimation for uncertain fuzzy descriptor systems: an LMI approach. *Trans. Inst. Meas.* 41, 135–144. doi: 10.1177/0142331218755235

Ichalal, D., Marx, B., Maquin, D., and Ragot, M. (2011). On observer design for nonlinear Takagi-Sugeno systems with unmeasurable premise variable[C]//2011 International Symposium on Advanced Control of Industrial Processes (ADCONIP). *IEEE.* 353–358. doi: 10.1109/MED.2012.6265744

Jia, Q., Chen, W., Zhang, Y., et al. (2015). Fault reconstruction and fault-tolerant control via learning observers in Takagi–Sugeno fuzzy descriptor systems with time delays. *IEEE Trans. Ind. Electron.* 62, 3885–3895. doi: 10.1109/TIE.2015.2404784

Jiang, B., Karimi, H. R., Kao, Y., and Gao, C. (2018). Adaptive control of nonlinear semi-Markovian jump TS fuzzy systems with immeasurable premise variables via sliding mode observer. *IEEE. Trans. Cybern.* (2018) 1–11.

Kchaou, M., Gassara, H., El-Hajjaji, A., and Toumi, A. (2014). Dissipativity-based integral sliding-mode control for a class of Takagi–Sugeno fuzzy singular systems with time-varying delay. *IET Control. Theory Appl.* 8, 2045–2054. doi: 10.1049/iet-cta.2014.0101

Kharrat, D., Gassara, H., Hajjaji, A., and Chaabane, M. (2018). Adaptive observer and fault tolerant control for takagi-sugeno descriptor nonlinear systems with sensor and actuator faults. *Int. J. Control Autom. Syst.* 16, 972–982. doi: 10.1007/s12555-017-0546-8

Kuppusamy, S., and Joo, Y. H. (2019). Memory-based integral sliding-mode control for ts fuzzy systems with pmsm via disturbance observer. *IEEE. Trans. Cybern.* 99–108. doi: 10.1109/TCYB.2019.2953567

Li, H., Wang, J., Du, H., Karmi, H. R. (2017). Adaptive sliding mode control for Takagi–Sugeno fuzzy systems and its applications. *IEEE Trans. Fuzzy Syst.* 26, 531–542. doi: 10.1109/TFUZZ.2017.2686357

Li, H., Yu, J., Hilton, C., and Liu, H. (2012) Adaptive sliding-mode control for nonlinear active suspension vehicle systems using T–S fuzzy approach. *IEEE Trans. Ind. Electron.* 60, 3328–3338. doi: 10.1109/TIE.2012.2202354

Li, J., Zhang, Q., Yan, X. G., Spurgeon, S. K. (2018). Observer-based fuzzy integral sliding mode control for nonlinear descriptor systems. *IEEE Trans. Fuzzy Syst.* 26, 2818–2832. doi: 10.1109/TFUZZ.2018.2802458

Li, R., and Zhang, Q. (2018). Robust H∞ sliding mode observer design for a class of Takagi–Sugeno fuzzy descriptor systems with time-varying delay. *J. Comput. Appl. Math.* 337, 158–178. doi: 10.1016/j.amc.2018.05.008

Li, X., Lu, D., Zeng, G., Liu, J., and Zhang, W. (2017). Integrated fault estimation and non-fragile fault-tolerant control design for uncertain Takagi–Sugeno fuzzy systems with actuator fault and sensor fault. IET Control Theory and Applications. 11, 1542–1553. doi: 10.1049/iet-cta.2016.1192

Lin, C., Wang, Q. G., and Lee, T. H. (2006). Stability and stabilization of a class of fuzzy time-delay descriptor systems. *IEEE Trans. Fuzzy Syst.* 14, 542–551. doi: 10.1109/TFUZZ.2006.876738

Liu, P., Yang, W. T., and Yang, C. E. (2013). Robust observer-based output feedback control for fuzzy descriptor systems. *Expert Syst. Appl.* 40, 4503–4510. doi: 10.1016/j.eswa.2013.01.053

López-Estrada, F. R., Astorga-Zaragoza, C. M., Theilliol, D., Ponsart, J. C., Palomo, G. V., and Torres, L. (2017). Observer synthesis for a class of Takagi–Sugeno descriptor system with unmeasurable premise variable. Application to fault diagnosis. *Int. J. Syst. Sci.* 48, 3419–3430. doi: 10.1080/00207721.2017.1384517

Lv, X., Fei, J., Sun, Y., and Fuzzy, P. I. D. (2019). Controller design for uncertain networked control systems based on T–S fuzzy model with random delays. *Int. J. Fuzzy Syst.* 21, 571–582. doi: 10.1007/s40815-018-0583-3

Petersen, I. R. (1987). A stabilization algorithm for a class of uncertain linear systems, Systems and Control Letters, 8, 351–357. doi: 10.1016/0167-6911(87)90102-2

Shams, Z., and Seyedtabaii, S. (2020) Nonlinear flexible link robot joint-fault estimation using Ts fuzzy observers. *Int. J. Robot. Autom.* 35. doi: 10.2316/J.2020.206-0214

Shen, Q., Jiang, B., and Cocquempot, V. (2011). Fault-tolerant control for T–S fuzzy systems with application to near-space hypersonic vehicle with actuator faults. *IEEE Trans. Fuzzy Syst.* 20, 652–665. doi: 10.1109/TFUZZ.2011.2181181

Soulami, J., Assoudi, A. E., Essabre, M., Habibi, M., and Yaagoubi, E. (2015). Observer design for a class of nonlinear descriptor systems: a Takagi-Sugeno approach with unmeasurable premise variables. *J. Control Sci. Eng.* 16, 1–10. doi: 10.1155/2015/636909

Sun, F., Li, L., Li, H. X., and Liu, H. (2007). Neuro-fuzzy dynamic-inversion-based adaptive control for robotic manipulators—discrete time case. *IEEE Trans. Ind. Electron.* 54, 1342–1351. doi: 10.1109/TIE.2007.893056

Takagi, T., and Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. Syst. Man Cybern. Syst.* 15, 116–132. doi: 10.1109/TSMC.1985.6313399

Tanaka, K., and Sano, M. (1994). On the concepts of regulator and observer of fuzzy control systems. *Proc. IEEE Int. Conf. Fuzzy Syst.* 767–772. doi: 10.1109/FUZZY.1994.343832

Taniguchi, T., Tanaka, K., and Wang, H. O. (2000). Fuzzy descriptor systems and nonlinear model following control. *IEEE Trans. Fuzzy Syst.* 8, 442–452. doi: 10.1109/91.868950

Tong, S., and Li, H. H. (2002). Observer-based robust fuzzy control of nonlinear systems with parametric uncertainties. *Fuzzy Sets Syst*, 131 (2), 165–184. doi: 10.1016/S0165-0114(01)00216-0

Van, M., Kang, H. J., and Suh, Y. S. (2013). A novel fuzzy second-order sliding mode observer-controller for a TS fuzzy system with an application for robot control. *Int. J. Precis. Eng. Manuf.* 14, 1703–1711. doi: 10.1007/s12541-013-0229-1

Vermeiren, L., Dequidt, A., Afroun, M., and Guerra, T-M. (2012). Motion control of planar parallel robot using the fuzzy descriptor system approach. *ISA Trans.* 51, 596–608. doi: 10.1016/j.isatra.2012.04.001

Vu, N. T. T., Yu, D. Y., Choi, H. H., and Jung, J. W. (2012). T–S fuzzy-model-based sliding-mode control for surface-mounted permanent-magnet synchronous motors considering uncertainties. *IEEE Trans. Ind. Electron.* 60, 4281–4291. doi: 10.1109/TIE.2012.2213554

Wu, B., Chen, M., and Zhang, L. (2019). Disturbance-observer-based sliding mode control for T–S fuzzy discrete-time systems with application to circuit system. *Fuzzy Sets Syst.* 374, 138–151. doi: 10.1016/j.fss.2018.10.022

You, F., Cheng, S., Tian, K., and Zhang, X. (2019). Robust fault estimation based on learning observer for Takagi-Sugeno fuzzy systems with interval time-varying delay. *Int. J. Adapt. Control Signal Process.* 1–18. doi: 10.1002/acs.3070

Zhang, B., Zhou, S., and Li, T. (2007). A new approach to robust and non-fragile H∞ control for uncertain fuzzy systems. *Informat. Sci.* 177, 5118–5133. doi: 10.1016/j.ins.2007.05.004

Zhang, J., Zhu, F., Karimi, H. R., and Wang, F. (2019). Observer-based Sliding Mode Control for TS Fuzzy Descriptor Systems with Time-delay. *IEEE Trans. Fuzzy Syst.* 27, 2009–2023. doi: 10.1109/TFUZZ.2019.2893220

Zhang, Q., Li, R., and Ren, J. (2018). Robust adaptive sliding mode observer design for ts fuzzy descriptor systems with time-varying delay. IEEE *Access.* 6, 46002–46018. doi: 10.1109/ACCESS.2018.2865618

Zhang, Z., Lin, D., Wang, X., and Shao, Z. (2019). A periodic event-triggered design of robust filtering for ts fuzzy discrete-time systems. *Front. Neurosci.* 318, 1–11. doi: 10.3389/fnins.2019.00001

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.