Novel adaptive path-smoothening optimization method for mobile robots

Shuyong Duan
Hebei University of Technology
Linxin Zhang
Hebei University of Technology
Xu Han (xhan@hebut.edu.cn)
Hebei University of Technology
Yule Li
Hebei University of Technology
Fang Wang
Hebei University of Technology
G.R. Liu
University of Cincinnati

Research Article

Keywords: Mobile robot, Adaptive path optimization, Neural network, Bézier curve

Posted Date: April 13th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1501159/v1

Title page

Novel adaptive path-smoothening optimization method for mobile robots

Shu-Yong Duan, born in 1984, is currently an associate professor at *State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, China*. She received her PhD degree from *Hunan University, China*, in 2016. Her research interests include robot reliability, computational inverse techniques, intelligent robotics.
E-mail: duanshuyong@hebut.edu.cn

Lin-Xin Zhang, born in 1997, is currently a master candidate at *State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, China*.
E-mail: zlinxin@126.com

Xu Han, born in 1968, is currently a professor and a PhD candidate supervisor at *State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, Chin*a. His main research interests include computational inverse techniques, optimization theory and algorithm.
E-mail: xhan@hebut.edu.cn

Yu-Le Li, born in 1997, is currently a master candidate at State Key Laboratory of Reliability and Intelligence of Electrical Equipment, *Hebei University of Technology, China*.
E-mail: 15735717718@163.com

Fang Wang, born in 1965, is currently a professor at *Hebei University of Technology, China*. She received her PhD degree from *Nanyang Technological University, Singapore*, in 1998. Her research interests include protective structure, high performance computing.
E-mail: mmmwangf@yahoo.com.sg

G.R. Liu, born in 1958, is currently a professor and a PhD candidate supervisor at *Department of Aerospace Engineering and Engineering Mechanics, University of Cincinnati, USA*. His main research interests include computational inverse techniques, intelligent perception of robots, intelligent algorithm, optimization theory and algorithm.
E-mail: liugr@uc.edu

Corresponding author：Xu Han　E-mail：xhan@hebut.edu.cn

ORIGINAL ARTICLE

Novel adaptive path-smoothening optimization method for mobile robots

Shu-Yong Duan[1] • Lin-Xin Zhang[1] • Xu Han[1] • Yu-Le Li[1] • Fang Wang[1] • G.R. Liu[2]

Received June xx, 202x; revised February xx, 202x; accepted March xx, 202x

Abstract: A safe and smooth operating path is a prerequisite for mobile robots to accomplish tasks. Although the existing path optimization methods improve the smoothness of the planned path by introducing Bézier curve to locally optimize the path with regard to turning points, most of these methods manually select the position of control points and subjectively analyze the feasibility of the optimized path. It is argued unfavourably that it exhibits strong subjectivity and cumbersome selection process. To this gap, an adaptive path-smoothening optimization method is proposed in this study, which combines neural network, genetic algorithm, and Bézier curve to effectively resolve the problems of strong subjectivity, cumbersome steps, and thus low efficiency in the selection process of control points. To start with, the data set corresponding to the position of the control point and the path offset are constructed. Based on the actual working conditions, the value space of control point position is derived. Latin hypercube sampling is used to sample the control point position of the second-order Bézier curve, which is input into the Bézier curve solution model to calculate the corresponding path offset. The data set corresponding to the position of control point and path offset are thus acquired. Based on the data set, the neural network algorithm is used to train it, and the prediction model of path offset is constructed. Subsequently, with reference to the prediction model of path offset, a performance evaluation function is formulated by comprehending multiple influential factors of mobile robot motion safety and path smoothness. The genetic algorithm is then introduced to detect the optimal control points in different environments. The proposed method is verified by experiments in different operating environments. The study results show that the currently proposed adaptive path-smoothening optimization method exhibits remarkably superior applicability and effectiveness compared to the currently prevailing methods. It demonstrates advantages of fast path planning, reduced path turning points, and desirable path smoothness. In addition, it can also ensure the safety of mobile robot along the planned path as availed by a pre-set criterion.

Keywords: Mobile robot • Adaptive path optimization • Neural network • Bézier curve

✉ Xu Han
xhan@hebut.edu.cn

[1] State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, Tianjin 300401, China

[2] Department of Aerospace Engineering and Engineering Mechanics, University of Cincinnati, Ohio 45241-1886, USA

1 Introduction

With accelerating development of autonomous navigation technologies for mobile robots, various self-navigating mobile robots are being used extensively in fields, such as industrial production, fire rescue, and daily life [1]. The algorithm for controlling the motion of self-navigating mobile robot primarily comprises a positioning algorithm, motion control algorithm, and path planning algorithm [2]. A path planning algorithm is expected to specify a safe path from the initial position to the target position for a self-navigating mobile robot in a workspace. High effectiveness of path planning promotes the success of tasks assigned to the robot and is vital to operational safety, stability, and reliability, thus the service life of the robots [3].

Currently available path planning algorithms are divided mainly into graph search-based, random sampling-based, potential field-based, and objective optimization-based path planning algorithms. Specifically, graph search-based path planning algorithms (Dijkstra's algorithm [4], A* [5], and jump point search (JPS) [6]) feature advantages of high computational efficiency and short path. On the other hand, these algorithms are argued unfavourably for disadvantages, such as a large number of turning points and low path safety and smoothness [7, 8]. Therefore, researchers have endeavoured to improve or optimise the paths planned by such algorithms. For example, Duan et al. [9] reduced the turning points by introducing safe distance matrix and optimising heuristic function in A*, and thereby, improved the path safety. Boroujen et al. [10] proposed the flexible unit A* based on vehicle speed information to improve path smoothness. As alternative, another category of random sampling-based path planning algorithms, e.g.

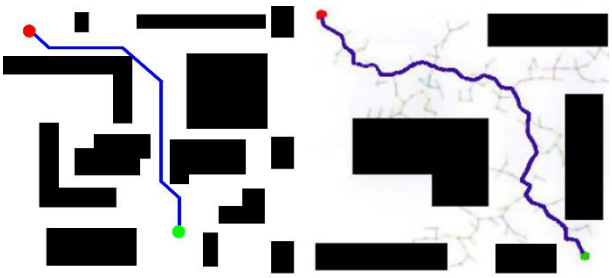
rapidly exploring random tree (RRT) and probabilistic roadmap (PRM), possess the superiorities of fast and strong search capacities. However, the paths planned using such algorithms exhibit low smoothness because of the innately ineffective random sampling characteristics as they simply fail to converge in a complex environment [11, 12]. To make amendment, Du et al. [13] proposed the modified RRT in terms of obstacles in the environment. Wang et al. [14] proposed the bidirectional RRT via the extension of target offset and Catmull–Rom spline interpolation. These researches have significantly improved the algorithm convergence rate and safety for the planned paths as well as shortened the path. On the other hand, the potential field-based path planning algorithms assign a gravitational field and a repulsive field in the environment, and plan the motion locus of a mobile robot by superposing gravitational and repulsive forces [15]. The merits of such algorithms are simple mechanism and low computational intensity to facilitate real-time underlying control. Still, the algorithms rely substantially on roadmap accuracy and display improficiency in a complex environment [16]. Zhang et al. [17] effectively addressed the issue of the low performance of the artificial potential field algorithm in complex environments by engaging the chaos optimization algorithm and adjusting the objective function and control variables. Jia [18] rectified the problem of trapping into a local optimum by the artificial potential field algorithm and slow convergence of RRT by combining the artificial potential field algorithm and RRT. The objective optimization-based path planning algorithms construct the corresponding objective functions via introducing an appropriate evaluation system to determine the optimal path [19]. Wang et al. [20] proposed a discrete optimization-based path planning algorithm. They reported that a mobile robot actively circumvented dynamic and static obstacles and the planned path was basically smooth and safe after a risk field and a cost function for design performance were engaged. Lyu et al. [21] proposed a modified path planning algorithm that integrated visibility graphs, B-spline curves, and particle swarm optimization to address issues of the discontinuous curvature of generated paths and trapping into a local optimum. Zhang et al. [22] proposed a path planning algorithm based on multi-stage and state space sampling. They achieved a smooth and safe path with a continuous curvature based on comprehension of geometric constraints and non-holonomic constraints.

To summarise, graph search-based and random sampling-based path planning algorithms are susceptive to issues of a large number of turning points and non-smoothness of paths as a result of their inadequate search approaches. On the other hand, potential field-based and objective optimization-based path planning algorithms effectively eliminate these issues. However, potential field-based and objective optimization-based path planning algorithms tend to yield very intensive computation and low convergence rate. To this gap, Bézier curve-based adaptive path optimization method is proposed in this study. The currently proposed method automatically identifies the position of the optimal control point in accordance with the criteria for determining such points, thus optimise the original path into a smooth and continuous locus.

2 Key issues in path planning algorithms

2.1 Graph search-based and random sampling-based path planning algorithms

Upon acquisition of complete workspace information for a mobile robot, the graph search-based and random sampling-based path planning algorithms can rapidly plan a safe path from the initial point to the target point. A case study has been performed on a path planned by A* [5] and RRT [12]. Fig. 1(a) shows a path planned by A*. It can be observed that the path is generally smooth. Nonetheless, apparent turning points exhibit at the corners. The safe distance between the path and obstacles at the corner is marginal. Thus, the safety of the robot traveling in the planned path may be at stake. Fig. 1(b) shows the path planned by RRT. This path is marginally superior over that planned by A* in terms of safety. Still, it is argued issues, such as a large number of turning points and generally low path smoothness, exhibit when RRT is applied. Although a four-wheel differential drive mobile robot can pass through turning points by pivot steering along a path generated by RRT, belt breakage and motor damage may be incurred during steering in scenarios involving low path smoothness and high friction coefficient. It severely negates the working stability, and thus service life of the robots. For mobile robots with an Ackermann steering mechanism, the front and rear wheels are generally the steering and driving wheels, respectively. While tracking a path containing apparent turning points, the robots may deviate from the planned path and even stuck at a turning point. Therefore, a smooth and continuous tracking path is fundamental to the improvement of the operational safety and stability, and thus the service life of mobile robots.

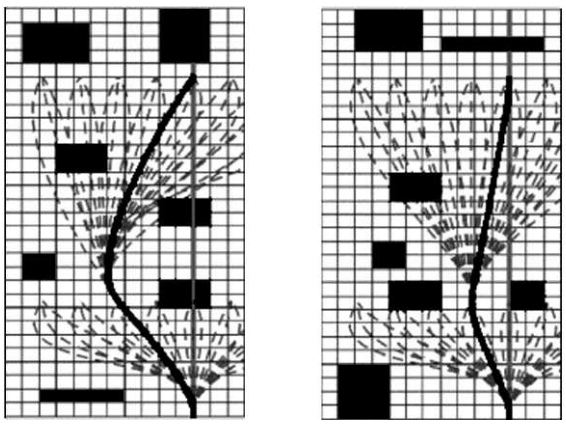


(a) A* algorithm (b) RRT algorithm

Figure 1 Paths by global path planning algorithms

2.2 Objective optimization-based path planning algorithms

Objective optimization-based path planning algorithms plan a smooth and continuous safe path for mobile robots in a workspace as well as construct an evaluation system. This category of algorithms consider the influential factors of locus curvature and kinematic constraints of mobile robots. Because a mobile robot is a system that is subjected to non-holonomic constraints, its kinematic constraints are approximated by the geometrical locus. Constraints, such as the continuous curvature of locus and the continuous motion velocity, are considered to plan a robot path using the optimization solution. These categories of algorithms have become the prevailing approaches to optimization solution for the motion path by the geometrical locus [23]. Based on a case study on two types of environments, as shown in Fig. 2, a smooth path satisfying the kinematic constraints of mobile robots was generated using the quartic Bézier curve [24]. Such a method can plan a smooth path with safety that satisfies the kinematic constraints of a mobile robot. However, unnecessary alternative paths, as indicated by dotted lines in Fig. 2, would be ensued simultaneously in the process of generating the final path to incur very demanding computational cost, which may not be manageable.

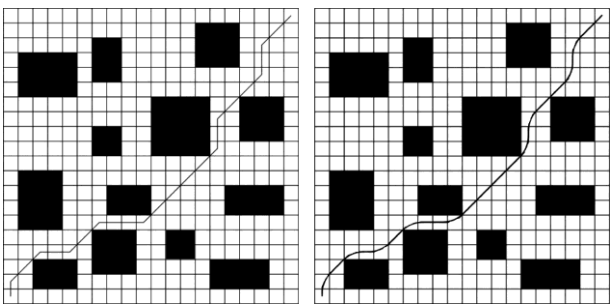


(a) A* algorithm (b) RRT algorithm

Figure 2 Path generation based on quartic Bézier curve

2.3 Post-processing optimization method

Because Bézier curve can effectively transit circular arcs and approximate the kinematic constraints of mobile robots, Bézier curve-based post-processing path optimization method has gradually become a preferential method for post-processing path optimization [25]. It is extensively used to improve the path smoothness by removing turning points. Fig. 3(b) shows the path by Bézier curve-based post-processing path optimization method. For comparison, Fig. 3(a) shows the path before optimization. The comparison of these paths indicates that the path smoothening performed via Bézier curve at the turning points notably improves the path smoothness. In addition, this method, regardless of the original path planning algorithms, features advantages of high computational efficiency and serializability. On the other hand, the order of Bézier curve is positively related to the number of necessary control points and the computational speed of the algorithm. Furthermore, different control point positions affect the shape of the Bézier curve at the corner. The transition of circular arcs is performed using cubic or even higher-order Bézier curves to determine the number of Bézier curve control points [26, 27]. Most Bézier curve control points are selected manually. Although a feasible path is generally yielded, the control point positions are selected manually, thus subjectively to render the selection process cumbersome and inefficient.



(a) Path before optimization (b) Path after optimization

Figure 3 Comparison of paths before and after Bézier curve post-processing

To summarize, rapid planning of safe, smooth, and high-quality tracking paths for mobile robots remains a challenge for currently available path planning methods. To this gap, this study presents an effective adaptive path optimization method by acquisition of feasible tracking paths for mobile robots.

3 Currently proposed adaptive path optimization method

Integrating the superior locus smoothening capacity of Bézier curve and the strong nonlinear fitting ability of the neural network algorithm [28, 29], this study derives the relationship between Bézier curve control points and path deviation via neural network algorithm. Specifically, smooth path is initially generated using Bézier curve and the positions of the optimal control points in the actual path are acquired by the genetic algorithm. Herein, the path deviation is considered as the governing factor of the performance evaluation function for the generated path, and the curvature of radius and path safety are considered as the constraints for the generated path.

3.1 Bézier curve

For Bézier curve-based locus optimization, a smooth curve satisfying the constraints can be generated only by selecting appropriate control points. A generic form of computational equation for control points on Bézier curve of an arbitrary order can be expressed as

$$P_i^k = \begin{cases} P_i^0 & i=0,1,...,n-k \\ (1-t)P_i^{k-1} + tP_{i+1}^{k-1} & i=0,1,...,n-k \end{cases} \quad (1)$$

where P_i^0 represents the *i*-th initial control point; P_i^k denotes the *i*-th k-order (*k*=1,2,3…n) control point; *t* is the normalized time.

According to Eq. (1), the computational equation for *n*-order Bézier curve control point contains two (*n*-1)-order Bézier curve control points. Therefore, the computational equation for *n*-order Bézier curve control points can finally be transformed into an equation associated with the initial control point. Although the high-order Bézier curve controls the turning heading angle of mobile robots and ensures path smoothness, it requires a substantial amount of computational expanse and displays low computational speed. Because the quadratic Bézier curve can reduce the computational intensity while ensuring path smoothness, quadratic Bézier curve is thus adopted in this study. As illustrated in Fig. 4, the computational equation for the quadratic Bézier curve takes the form of

$$B(t) = (1-t^2)P_i^0 + 2(1-t)P_{i+1}^0 + t^2 P_{i+2}^0 \ , t \in [0,1] \quad (2)$$

where P_i^0, P_{i+1}^0, P_{i+2}^0 stand for three consecutive initial control points.

In a two-dimensional plane, curvature of any point on Bézier curve can be calculated by

$$\kappa(t) = \frac{P_{ix}^k{}'(t)P_{iy}^k{}''(t) - P_{iy}^k(t)P_{ix}^k{}''(t)}{\sqrt[3]{(P_{ix}^k{}'(t)^2 + P_{iy}^k{}'(t)^2)^2}} \quad (3)$$

where $\kappa(t)$ is the curvature at time instant *t*; $P_{ix}^k{}'(t)$, $P_{iy}^k{}'(t)$, $P_{ix}^k{}''(t)$ and $P_{iy}^k{}''(t)$ stand for the first and second derivatives of the *i*-th *k*-order control point on Bézier curve with respect to *x* and *y*.

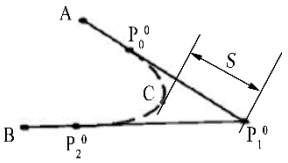


Figure 4 Second-order Bézier curve

Because Bézier curve has the characteristics of affine transformation invariance, the quadratic Bézier curve on the plane can be determined uniquely if three control point positions are specified. Fig. 4 shows the unique smooth curve $P_0^0P_2^0$ generated via the three control points P_0^0, P_1^0, and P_2^0. It can be observed that the curve is tangential to the original path at the entry and exit positions. Thus, the mobile robot can smoothly pass through the connecting points along the path. However, large deviations exhibit between $P_0^0P_2^0$ and the original path $\overline{P_0^0P_1^0P_2^0}$. As a result, the optimised path may render unsafe path to collide with obstacles.

To rectify, the shortest distance between Bézier curve and the control point P_1^0 is defined as the path deviation *S* in this study. As shown in Fig. 4, the magnitude of *S* is related to the position of the control points P_0^0 and P_2^0. Larger *S* indicates shorter optimised path. Thus, the control point position corresponding to a relatively large *S* should be selected, provided that path safety is ensured.

3.2 Construction of neural network model

The optimal control point positions can be specified by inferring the relationship between *S* and the control points P_0^0 and P_2^0. Because the back propagation (BP) neural network algorithm is a multi-layer feedforward neural network algorithm, an arbitrary nonlinear mapping from input to output can be achieved by setting the transformation function of neurons as a sigmoid function. The output can be a continuous value ranging from zero to one. Therefore, the relationships among Bézier curve control points, *S*, and the shortest distance point C are determined using BP neural network algorithm. This approach serves as the theoretical context for selecting the optimal control point.

3.2.1 Dataset establishment

Fig. 4 illustrates that the positions of the control points P_0^0 and P_2^0 determine the magnitude of *S* and the position of the shortest distance point C. The positions of the control points P_0^0 and P_2^0 are related primarily to the lengths of the line segments $\overline{P_0^0P_1^0}$, $\overline{P_1^0P_2^0}$ and the

angle $\angle P_0^0 P_1^0 P_2^0$. In this study, the numeral range of $\overline{P_0^0 P_1^0}$ and $P_1^0 P_2^0$ is specified as 0–100 and $\angle P_0^0 P_1^0 P_2^0$ is 0°–180° in the sampling space. Considering the characteristics of uniform dispersibility and symmetrical comparability of data obtained by Latin hypercube sampling (LHS), totally 6,000 sets for lengths of the line segments $\overline{P_0^0 P_1^0}$, $\overline{P_1^0 P_2^0}$ and angle $\angle P_0^0 P_1^0 P_2^0$ are acquired via LHS in the sampling space to be adopted as input data. The magnitude of *S* corresponding to each set of data is determined by Eq. (2). The data set to yield the minimum *S* is then identified. Table 1 lists typical data sets and the shortest *S*.

Table 1 Factors and their levels

$\angle P_0^0 P_1^0 P_2^0$	$\overline{P_0^0 P_1^0}$	$\overline{P_1^0 P_2^0}$	Shortest distance point	Path deviation
123.261°	58.826	14.22	69%	5.99
31.205°	10.618	22.82	34%	7.01
…	…	…	…	…
131.872°	86.864	34.32	63%	10.51

The data set corresponding to the shortest distance, as shown in Table 1, indicates the ratio of $P_0^0 C$ to $P_0^0 P_2^0$, which can be used to infer the deviation mode of a circular arc locus. A ratio less than 50% implies that the circular arc locus deviates towards point P_0^0. Otherwise, the circular arc locus deviates towards point P_2^0. The maximum *S* and minimum *S* determined by Eq. (2) are 48 and 0.001, respectively. It indicates a significant difference between the maximum *S* and minimum *S*. If the nonlinear fitting is performed directly using BP neural network algorithm, the training error would be measured by the mean squared error (MSE) as follows,

$$MES = \frac{1}{m \cdot o} \sum_{i=1}^{m} \sum_{k=1}^{o} (q_k^i - r_k^i)^2 \tag{4}$$

where *m* is the number of training samples; *o* represents the number of neurons in output layer; q_k^i stands for the predicted value of the *i*-th training sample at the *k*-th node of the output layer; r_k^i denotes the expected output value of the *i*-th training sample at the *k*-th node of the output layer. After the neural network is trained, its loss value can be controlled to be 10^{-6} and even smaller. However, the maximum relative error between the predicted and actual values could be as high as 900%. It is attributed to the difference in magnitude between the maximum and minimum values, which is as high as 10^4. In addition, the data normalisation ensues the magnitude of the data close to the minimum value to be 10^{-4}. It results in high relative error between the predicted and actual values due to extremely low base value.

In order to improve the training accuracy, the path deviation should be pre-processed. Firstly, the logarithm of *S* is taken to reduce the magnitude difference between the maximum and minimum values of *S*. A coefficient m_c is then added to each group of data to ensure that the processed data are all positive-definite, thus non-zero:

$$S' = \lg(S) + m_c \tag{5}$$

where *S* represents the path deviation; *S'* stands for the positive logarithmic path offset, which will serve as the path deviation after pre-processing; m_c denotes the migration coefficient and m_c=4. Some representative calculated results are listed in Table 2.

Table 2 Representative pre-processed data

The shortest distance point	*S*	lg(*S*)	*S*'
69%	5.99	0.777113	4.777113
49%	47.33	1.675119	5.675119
34%	7.01	0.84584	4.84584
…	…	…	…
5%	0.0084	-2.075721	1.922793
63%	10.51	1.021458	5.021458

Based on the computational results, *S'* spans a range of 1.9–5.7, and all the values are positive. It indicates a reduction in the relative error for network training. The datasets for network training are finally established. Table 3 lists some representative data. Specifically, $\angle P_0^0 P_1^0 P_2^0$, $\overline{P_0^0 P_1^0}$, and $\overline{P_1^0 P_2^0}$ are the input parameters for the network, and the shortest distance point and *S* are the output parameters.

Table 3 Summary of training data

$\angle P_0^0 P_1^0 P_2^0$ /°	$\overline{P_0^0 P_1^0}$	$\overline{P_1^0 P_2^0}$	Shortest distance point	*S'*
123.261°	58.826	14.22	69%	4.777113
22.624°	93.849	99.367	49%	5.675119
31.205°	10.618	22.82	34%	4.84584
…	…	…	…	…
147.655°	0.017	7.968	5%	1.922793
131.872°	86.864	34.32	63%	5.021458

3.2.2 Neural network structure and training parameters

The structure of the neural network model contains three layers: input, hidden, and output layers, as shown in

Fig. 5. The numbers of neurons in the input and output layers are determined by the feature dimension and the categorical number of the training samples. In this study, there are three independent variables and two dependent variables for the training samples. There are three neurons in the input layer and two neurons in the output layer. In general, the number of neurons in the hidden layer can be determined arbitrarily. However, according to an empirical formula, the fitting can be achieved only if the number of neurons in the hidden layer is larger than M, where M =sqrt (number of neurons in input layer + number of neurons in output layer). Only if the number of neurons in the hidden layer is sufficiently large, can then the arbitrary nonlinear mapping relationship be theoretically fitted using the three-layer network structure from input to output [30].

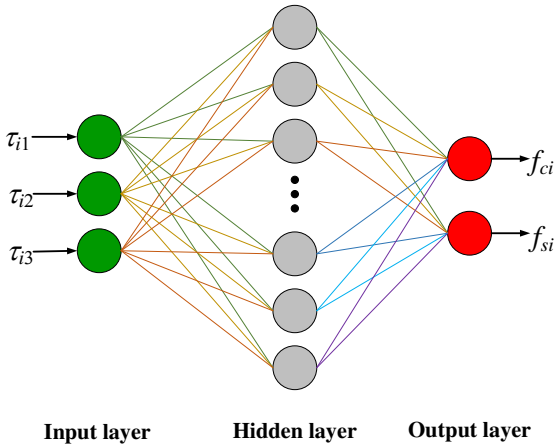


Figure 5 Neural network model

The hyperbolic tangent function is used as the activation function for each layer. Output neuron j in the hidden layer can be expressed as

$$h_j = \frac{e^{(\sum_{i=1}^{3} w_{ij}\tau_i - \theta_j)} - e^{(-\sum_{i=1}^{3} w_{ij}\tau_i - \theta_j)}}{e^{(\sum_{i=1}^{3} w_{ij}\tau_i - \theta_j)} + e^{(-\sum_{i=1}^{3} w_{ij}\tau_i - \theta_j)}} \quad j = 1,2,...,H \qquad (6)$$

where w_{ij} represents the weight from the i-th neuron in the input layer to the j-th neuron in the hidden layer; θ_j denotes the threshold of j-th neuron in the hidden layer; τ_i stands for the i-th input of the hidden layer neuron; H represents the number of hidden layer neurons.

By analogy, the output neuron k-th in the output layer can be derived as

$$f_k = \frac{e^{(\sum_{j=1}^{H} w_{jk}h_j - \theta_k)} - e^{(-\sum_{j=1}^{H} w_{jk}h_j - \theta_k)}}{e^{(\sum_{j=1}^{H} w_{jk}h_j - \theta_k)} + e^{(-\sum_{j=1}^{H} w_{jk}h_j - \theta_k)}} \quad k = 1,2 \qquad (7)$$

where w_{jk} represents the weight from the j-th neuron in the hidden layer to the k-th neuron in the output layer; θ_k denotes the threshold of k-th neuron in the output layer.

The training accuracy of a neural network is evaluated in terms of the mean squared error (MSE) in Eq. (4). Lower training MSE indicates higher accuracy of the neural network model. If only one hidden layer presents and "overfitting" does not occur in the neural network, the number of neurons in the hidden layer can be increased appropriately to improve the training accuracy of the neural network model. Therefore, an appropriate number of neurons that enables short training time and high training accuracy in the hidden layer should be specified. Table 4 shows the corresponding relationships between training time, accuracy, and the number of neurons in the hidden layer from multiple experiments.

Table 4 Influence of different numbers of hidden layer neurons on training time and training accuracy

Number of neurons in hidden layer	Number of iterations	Training error /10^{-5}
50	158	2.59
60	269	3.29
70	224	1.8
80	119	2.58
90	195	1.77
100	107	1.54
110	189	1.64
120	214	1.93

Table 4 indicates that an increase in the number of neurons in the hidden layer reduces the training error of the neural network while increasing the training time. Thus, appropriate number should be chosen to strike trade-off balance between accuracy and training time. From Table 4, 100 neurons in the hidden layer result in a high training accuracy as well as a reasonably short training time. Therefore, this number is selected in this study, and the neural network structure model is then specified as 3-100-2. Totally, 6,000 datasets are selected. Amongst, 70% datasets (i.e., 4,200 datasets) are assigned as the training sets, 20% (1,200 datasets) are considered as the test sets, and 10% (600 datasets) are the validation sets. The learning rate of neural network a is specified as 10^{-5}, and the hyperbolic tangent function is used as the activation function for each layer. Fig. 6 shows the loss curve for neural network

training, and Fig. 7 shows the correlation curve for input and output data. Fig. 6 reveals that algorithm convergence can be achieved after 107 iterations of the three-layer neural network structure model (3-100-2). From Fig. 7, the coefficients of determination R for the training set, test set, validation set, and overall data are higher than 0.99. It indicates that the dependent variables can be characterised effectively by the independent variable. In other words, $\angle P_0^0 P_1^0 P_2^0$, $\overline{P_0^0 P_1^0}$, and $\overline{P_1^0 P_2^0}$ can effectively infer variations in the shortest distance point and *S'*.

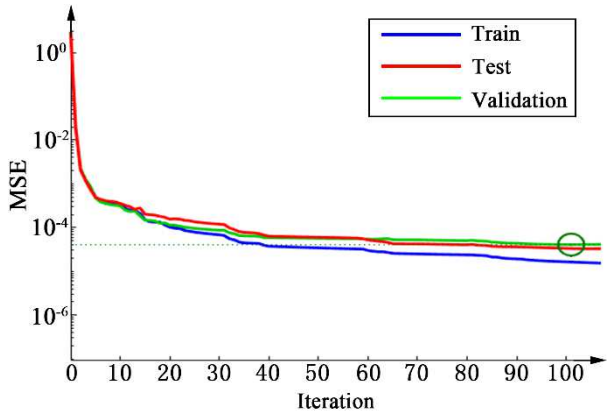


Figure 6 Loss curve for neural network training

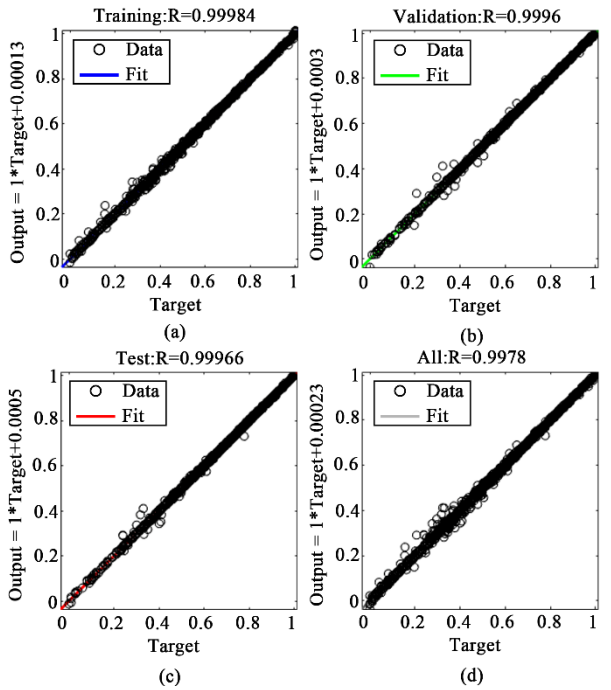


Figure 7 Correlation curves for input and output data

3.3 Determination of optimal control points by genetic algorithm

3.3.1 Construction of path performance evaluation function

In the genetic algorithm, the fitness function is the critical factor for the algorithm process. Higher degree of fitness of chromosomes indicates higher adaptability and probability of the selection for entries into the next generation. In this study, the path performance evaluation function is constructed and considered as the fitness function in the genetic algorithm. The relationships of $\overline{P_0^0 P_1^0}$, $\overline{P_1^0 P_2^0}$, and $\angle P_0^0 P_1^0 P_2^0$ with *S'* fitted by BP neural network algorithm are considered as the core of the path performance evaluation function, which takes the form of

$$S' = f(\beta, d_1, d_2) \tag{8}$$

where *S'* denotes path offset; β represents the angle of $\angle P_0^0 P_1^0 P_2^0$; d_1, d_2 stand for the lengths of $\overline{P_0^0 P_1^0}$, $\overline{P_1^0 P_2^0}$, respectively.

The key to the efficient optimization in this study is the appropriate selection of control point positions and the maximum sum of *S* at each section provided that the path safety and continuous curvature are ensured. Thereby, a smooth and continuous path can be generated.

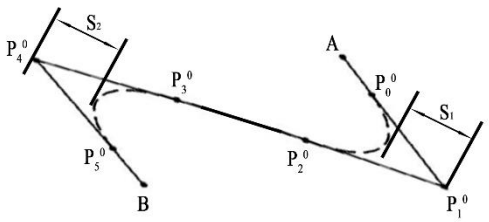


Figure 8 Optimization process of Bézier curves on continuous paths

Fig. 8 illustrates the optimization process of Bézier curves on a continuous path. It demonstrates that the two turning points P_1^0 and P_4^0 still exhibit. Therefore, appropriate control points, i.e., P_0^0, P_2^0, P_3^0, and P_5^0, should be selected so as to maximise $S_1 + S_2$ or $S_1' + S_2'$. Because the control points P_2^0 and P_3^0 are on the line segment $P_1^0 P_4^0$, the sum of the lengths of $\overline{P_1^0 P_2^0}$ and $\overline{P_3^0 P_4^0}$ should be less than the length of $\overline{P_1^0 P_4^0}$. If the sum of the lengths of $\overline{P_1^0 P_2^0}$ and $\overline{P_3^0 P_4^0}$ is larger than the length of $\overline{P_1^0 P_4^0}$, staggered contours occur. In this study, the influential factors, e.g. the path safety and smoothness, are considered comprehensively. The objective function and path performance evaluation function can be respectively expressed as

$$K = \frac{1}{\sum_{i=1}^{n} f(\beta_i, d_{1i}, d_{2i})} + p + c + r + z \tag{9}$$

$$F(\beta_i, d_{1i}, d_{2i}, p, c, r, z) = \frac{1}{K} \tag{10}$$

where $f(\beta_i, d_{1i}, d_{2i})$ represents the path offset for the i-th turn position; p is the penalty factor for distance, that is, the sum of the lengths of $\overline{P_1^0 P_2^0}$ and $\overline{P_3^0 P_4^0}$ is less than the length $\overline{P_1^0 P_4^0}$; p = 0, conversely, the value p is proportional to the square number that does not satisfy the side length condition. c denotes the penalty factor for collisions, in other words, if the path does not collide with obstacles, c = 0; otherwise, c = 1. r stands for the penalty factor for path curvature, which is calculated by Eq. (3). If the curvature is less than the maximum curvature, r = 0, otherwise, r = 1. z represents the penalty factor for arc trajectory deviation: if the offset value of the arc trajectory is higher than 10% and less than 90%, then z=0; otherwise, z=1.

According to Eqs. (9) and (10), lower K indicates higher fitness of the individual, and higher performance of the generated path, closer proximity of the control point position to the ideal point position. Therefore, Eq. (9) is considered as the objective function, and the minimum K value and optimal control point position are determined by genetic algorithms in this study.

3.3.2 Construction of path performance evaluation function

A genetic algorithm involves encoding and decoding processes. A conventional genetic algorithm applies binary encoding to store the chromosome information from the population, performs genetic operations, e.g. selection and crossover, on the encoded chromosomes, and decodes the chromosomes upon completion of the genetic operations. The encoding and decoding processes increase the algorithmic complexity, and thus negate the algorithmic computational efficiency. To make amendment, the encoding is performed directly using real numbers in this study to streamline the algorithmic structure and boost the algorithm-based solving speed. In addition, the conversion process between real numbers and binary numbers is spared to further improve the algorithm-based solving efficiency.

Because real numbers are encoded in this study, the range of genetic boundaries for each gene segment of chromosomes is detected by below equation after selection, crossover, and mutation of chromosomes:

$$J_{\text{test}} = \begin{cases} 0, & a_{ij} < a_{j\min} \\ 1, & a_{j\min} < a_{ij} < a_{j\max} \\ 0, & a_{ij} > a_{j\max} \end{cases} \tag{11}$$

where a_{ij} represents the j-th gene of individual i; $a_{j\min}$ denotes the minimum value of the j-th gene; $a_{j\max}$ is the maximum value of the j-th gene; $J_{\text{test}} = 1$ stands for the chromosomes that have been genetically manipulated to meet the requirements, otherwise $J_{\text{test}} = 0$.

The selection, crossover, and mutation procedures are specified as below.

Selecting: Based on the fitness value of individuals in the population, that is, the path performance evaluation function, the roulette method is used to select superior individuals from the previous generation population with specified probability to build a new population. The probability of individual i being selected can be calculated by

$$p_i = \frac{F_i}{\sum_{i=1}^{N} F_i} \tag{12}$$

where F_i represents the fitness value of individual i; $\sum_{i=1}^{N} F_i$ denotes the sum of the fitness of the population.

Crossing: Two individuals are randomly selected from the new population. A gene fragment is randomly selected for exchange and combination to generate new individuals, which are crossed with regard to below conditions:

$$\begin{cases} a_{ij} = a_{ij}(1-B) + a_{kj}(B) \\ a_{kj} = a_{kj}(1-B) + a_{ij}(B) \end{cases}, B \in [0,1] \tag{13}$$

where a_{ij} represents the j-th gene of individual i; a_{kj} denotes the j-th gene of individual k; B stands for the random number of the crossing.

Since the real number coding is adopted in this study, after completing the crossing, Eq. (11) should be engaged to detect the gene boundary range. If the requirements are met, the crossing is completed; otherwise, the crossing should be performed again.

Mutation: An individual is randomly selected from the new population, and a gene locus is also randomly selected. The mutation operation is performed by Eq. (14). After completing the mutation operation, gene boundary range detection should be implemented by Eq. (11). If the requirements are met, the mutation operation should be completed. Otherwise, the mutation operation should be carried out again. At the same time, with the evolution of the population, the individuals of the population will become superior. Therefore, the mutation rate adopted in this paper will gradually decline with the increase of evolution algebra, which takes the form of

$$a_{ij} = \begin{cases} a_{ij} - (a_{ij} - a_{j\min}) * (1-R), r_{\text{o}} < 0.5 \\ a_{ij} + (a_{j\max} - a_{ij}) * (1-R), r_{\text{o}} > 0.5 \end{cases} \tag{14}$$

where a_{ij} represents the j-th gene of individual i; $a_{j\min}$ denotes the lower limit of the value of the j-th gene;

$a_{j\max}$ is the upper limit of the value of the j-th gene; r_o represents the random number of the mutation. R stands for the mutation rate and there is

$$R = R_0(1 - \frac{G}{G_{\max}})^2 \tag{15}$$

where R_0 denotes the initial mutation rate; $G_{\max}$ stands for the maximum number of iterations; G represents the current number of iterations.

The flow chart for the stepwise solution process of the optimal control point position is shown in Fig. 9, which is exposited as follows:

1) Set the basic parameters, i.e. the initial parameters of the algorithm, population size, crossing rate, and mutation rate.

2) Initialize the population, specify the maximum evolutionary generation $G_{\max}$ of the population, and set the first-generation population $G = 0$.

3) Evaluate the feasibility of the path via calculations of the distance penalty factor p, the collision penalty factor c, the path curvature penalty factor r, and the arc trajectory offset penalty factor z.

4) Calculate the fitness of the population of current generation through Eq. (10).

5) Selecting: select superior individuals of the population into the next generation via Eq. (12).

6) Crossing: generate new individuals for the next generation via crossovering operation of population by Eq. (13).

7) Mutation: generate new individuals and enter the next generation through mutation operation by Eqs. (14) and (15).

8) Advance to the next generation population: $G = G+1$.

9) Check if the population G satisfies $G = G_{\max}$: if so, go to Step 10; otherwise, go back to Step 3.

10) Output the optimal solution: the optimal control points are solved and the positions of each control point are output.

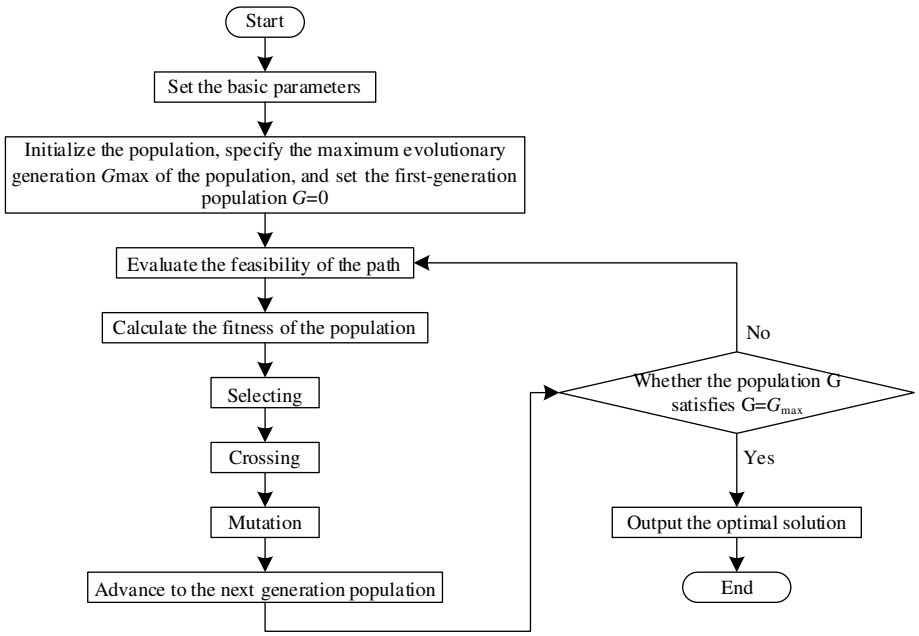


Figure 9 Flow chart of stepwise solution process for optimal control point position

After the optimal control point is determined by the genetic algorithm, a smooth curve is generated via Eq. (2) and the corresponding control point position at the turning position along the original path. To the objective, a smooth and collision-free global navigation path that satisfies the kinematic requirements of mobile robots is thus acquired. Fig. 10 illustrates the process of the currently proposed adaptive path optimization method.

The stepwise procedure for the adaptive path optimization method is exposited as below.

1) Input training data of 6000 sets extracted by Latin Hypercube.

2) Build a neural network model: according to the input and output data and the requirements of training accuracy, a 3-100-2 three-layer network model is constructed.

3) Set training parameters of learning rate, maximum number of iterations, etc.

4) Train the network model.

5) Output training results and evaluate network performance: according to the training results, the network performance is evaluated whether it fits for construction of the path performance evaluation function.
6) Input the actual path and use the genetic algorithm to solve the optimal control point.

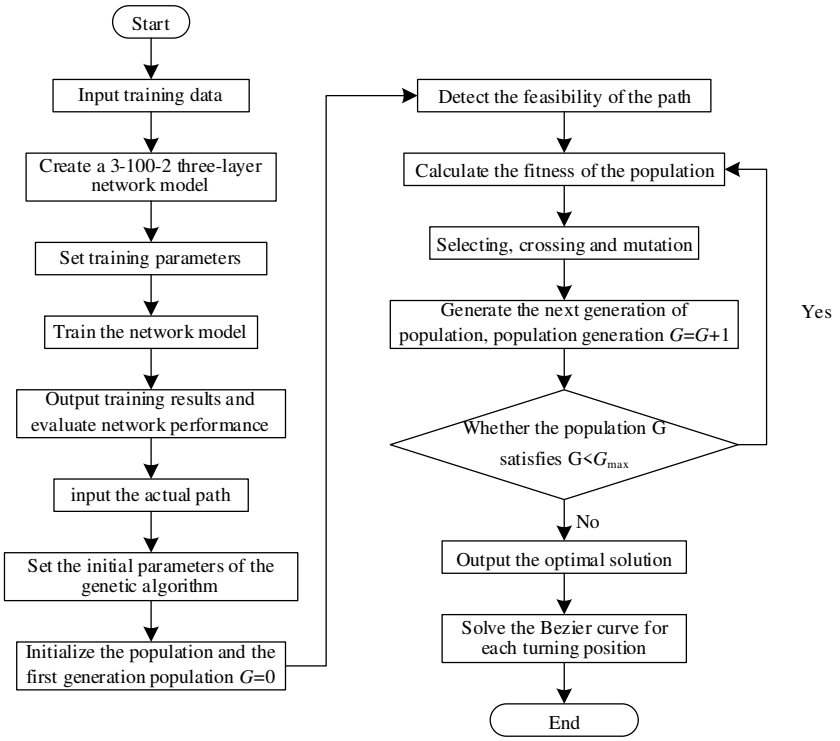


Figure 10 Flow chart of currently adaptive path optimization method

4 Validation of adaptive path optimization method

Simulation analysis and experimental validation are performed via the aforementioned processes to evaluate the applicability of the currently proposed adaptive path optimization method.

4.1 Validation of BP neural network model

4.1.1 Result analysis of test set

The relationships among $\overline{P_0^0P_1^0}$, $\overline{P_1^0P_2^0}$, $\angle P_0^0P_1^0P_2^0$, the shortest distance point, and S' fitted by BP neural network algorithm are fundamental for determining the optimal solution using the genetic algorithm. The accuracy of the equation specifying the relationships governs the appropriateness of the optimal control point position via the genetic algorithm. This study analyses the test results by BP neural network model. Table 5 shows the relative errors between the expected and predicted values for the shortest distance point and the path offset, respectively. From Table 5, the ratios of data with relative errors higher than 10% for the shortest distance point and the path offset are as low as 0.67% and 0.42%, respectively. Correspondingly, the ratios of data with relative errors less than 1% for the shortest distance point and the path offset are as high as 80.99% and 91.75%, respectively. It can be stated that among the 1,200 test datasets, which are persuasively sufficient, the ratios of data with slightly large relative errors are exceptionally low. The overall relative errors in the test sets are moderately low in view of that the ratios of data with relative errors within 10^{-5}–10^{-6} are high among the data with relative errors of less than 1%. Therefore, BP neural network model adopted in this study demonstrates desirable prediction accuracy.

Table 5 Relative errors for shortest distance point and S'

Classification	Relative error	Shortest distance point	S'
1	e>10%	0.67%	0.42%
2	5%<e<10%	1.17%	0.5%
3	1%<e<5%	17.17%	7.33%
4	e<1%	80.99%	91.75%

4.4.2 Validation of random sampling model

Because the accuracy of BP neural network model plays a pivotal role in the subsequent optimization operations, random sampling is again implemented in this study to further validate the computational accuracy of BP neural

network model. In this part of the study, 30 sets of data are selected randomly in the specified domain. According to the curve for the relative errors between the expected and predicted values for the shortest distance point, as shown in Fig. 11, the maximum relative error for the shortest distance point is as low as 3.32% in the extracted data, and the relative errors for the other 29 sets of data are also below 2%. According to the curve for the relative error between the expected and predicted values for the path offset, as shown in Fig. 12, the maximum relative error for the path offset is 0.54%, and the relative errors for most data are below 0.2%. A specific numeral comparison of the expected and predicted results in Table 6 reveals that the differences between the expected and predicted values are marginal. According to a case study on the data for the shortest distance point of the fifth sample in Group 3, the difference between the expected and predicted values is as low as 0.007, but the relative error is high. The reason is that the expected value is small, slight fluctuation of the predicted value will incur drastic change in the relative error, which can be deemed as non-indicative or irrelevant. Nonetheless, the difference between the expected and predicted values is controlled to be within 10-2–10-3. Thus, it can be stated that BP neural network model adopted in this study exhibits high prediction accuracy. It yields a highly accurate and reliable mapping relationship for determining the optimal control point by the genetic algorithm.

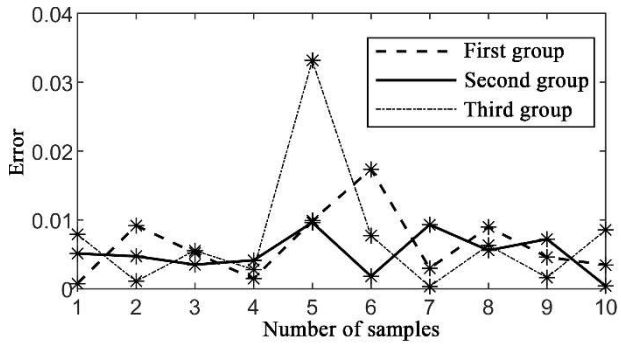


Figure 11 Relative error curves of expected values and predicted values for shortest distance points

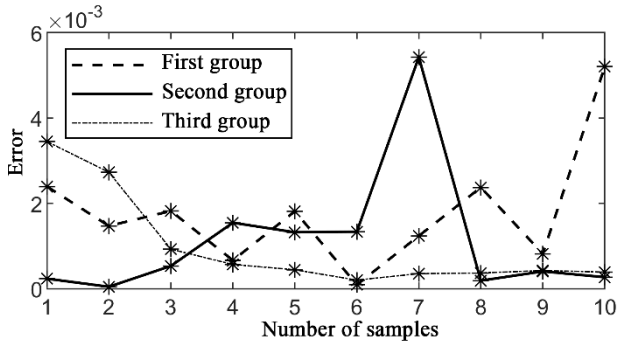


Figure 12 Relative error curves of expected values and predicted values for *S'*

Table 6 Comparison of expected values and predicted values

Classification	Serial number	Angle	Control side I	Control side II	Expected value		Predicted value		Error	
					Shortest distance point	*S'*	Shortest distance point	*S'*	Shortest distance point	*S'*
First group	1	159.117	51.259	42.057	53%	4.62	52.96%	4.61	0.08%	0.24%
	2	74.292	9.802	91.149	17%	4.90	16.84%	4.89	0.92%	0.15%
	3	149.155	93.449	75.796	53%	5.05	53.28%	5.04	0.52%	0.18%
	4	151.915	54.559	25.471	60%	4.64	60.09%	4.64	0.15%	0.07%
	5	1.560	75.629	32.555	70%	5.36	70.69%	5.37	0.99%	0.18%
	6	16.803	17.470	83.381	19%	5.16	18.67%	5.16	1.74%	0.01%
	7	76.183	43.724	83.864	39%	5.36	39.12%	5.37	0.30%	0.12%
	8	139.673	14.519	70.262	31%	4.67	30.72%	4.68	0.90%	0.24%
	9	54.939	24.104	32.972	44%	5.09	44.20%	5.09	0.46%	0.08%
	10	71.772	77.163	0.717	98%	3.84	97.66%	3.82	0.35%	0.52%
Second group	1	145.464	96.866	61.227	56%	5.05	56.44%	5.04	0.79%	0.34%
	2	114.499	98.800	94.866	51%	5.42	51.06%	5.40	0.11%	0.27%
	3	130.792	49.558	92.165	42%	5.14	42.23%	5.14	0.55%	0.09%
	4	21.814	56.693	24.221	70%	5.22	69.81%	5.23	0.28%	0.06%
	5	43.387	20.720	94.649	21%	5.21	21.70%	5.21	3.32%	0.04%
	6	48.308	31.555	55.826	39%	5.27	38.70%	5.27	0.77%	0.02%
	7	59.440	82.497	84.114	50%	5.56	50.02%	5.56	0.03%	0.04%
	8	27.605	91.232	15.653	84%	5.12	84.53%	5.12	0.63%	0.04%
	9	51.369	57.576	3.584	92%	4.52	91.85%	4.51	0.16%	0.04%
	10	141.474	77.796	66.028	53%	5.07	52.55%	5.07	0.85%	0.04%

Third group	1	145.464	96.866	61.227	56%	5.05	56.44%	5.04	0.79%	0.34%
	2	114.499	98.800	94.866	51%	5.42	51.06%	5.40	0.11%	0.27%
	3	130.792	49.558	92.165	42%	5.14	42.23%	5.14	0.55%	0.09%
	4	21.814	56.693	24.221	70%	5.22	69.81%	5.23	0.28%	0.06%
	5	43.387	20.720	94.649	21%	5.21	21.70%	5.21	3.32%	0.04%
	6	48.308	31.555	55.826	39%	5.27	38.70%	5.27	0.77%	0.02%
	7	59.440	82.497	84.114	50%	5.56	50.02%	5.56	0.03%	0.04%
	8	27.605	91.232	15.653	84%	5.12	84.53%	5.12	0.63%	0.04%
	9	51.369	57.576	3.584	92%	4.52	91.85%	4.51	0.16%	0.04%
	10	141.474	77.796	66.028	53%	5.07	52.55%	5.07	0.85%	0.04%

4.2 Applicability of current adaptive path optimization method in different obstacle environments

In environments with a uniform roadmap size of 10 × 10 and obstacle rates of 5%, 10%, 15%, and 20%, paths are planned using A*, Dijkstra's algorithm, JPS, and breadth-first search (BFS), respectively. The planned paths are then smoothened using the currently proposed adaptive path optimization method. Figs. 13–20 show the simulated and optimized paths. Specifically, Figs. 13, 15, 17, and 19 show the original paths planned using the aforementioned algorithms in environments with different obstacle rates, and Figs. 14, 16, 18, and 20 show the respective optimised paths. Figs. 21–24 depict the convergence curves of the genetic algorithm.

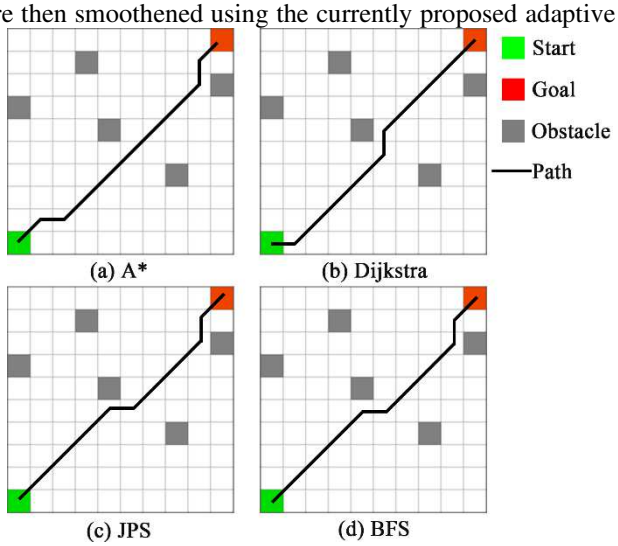


Figure 13 Original paths in environment with 5% obstacles

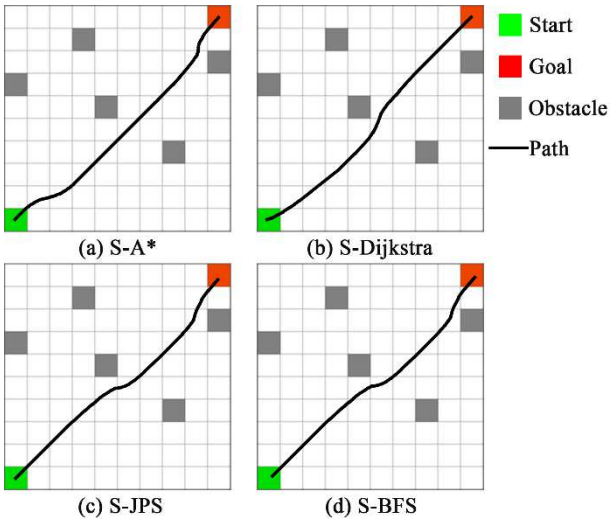


Figure 14 Optimised paths in environment with 5% obstacles

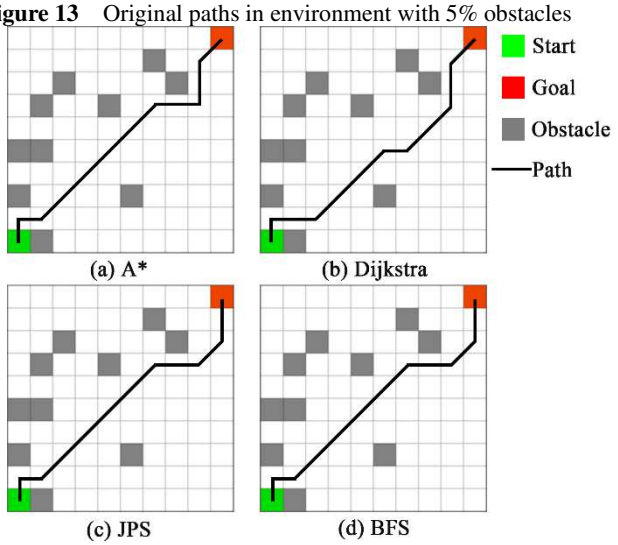


Figure 15 Original paths in environment with 10% obstacles

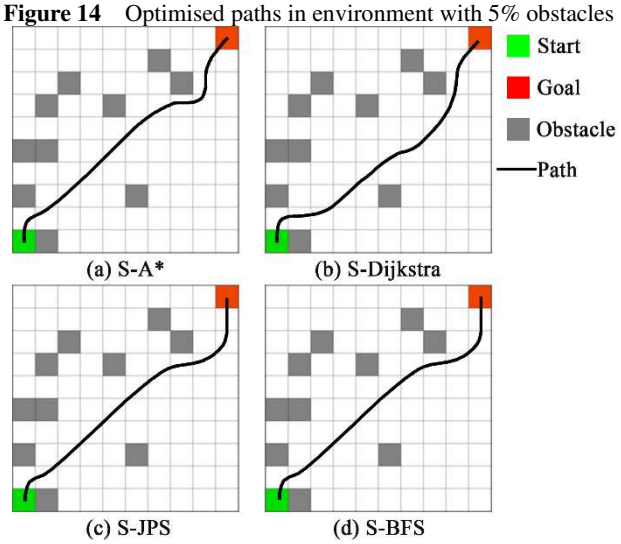


Figure 16 Optimised paths in environment with 10% obstacles

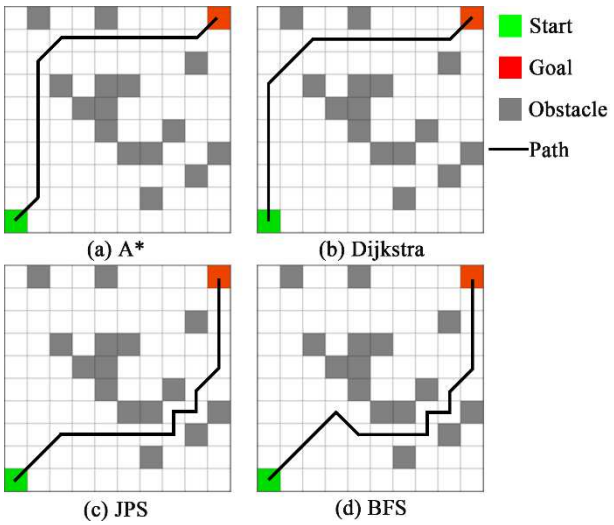


Figure 17 Original paths in environment with 15% obstacles

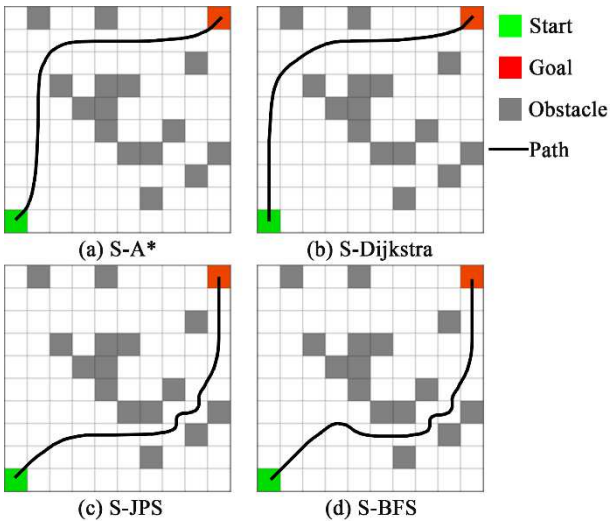


Figure 18 Optimised paths in environment with 15% obstacles

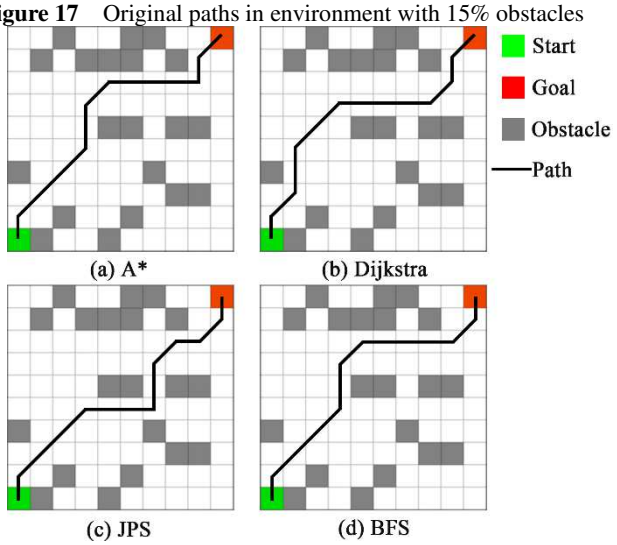


Figure 19 Original paths in environment with 20% obstacles

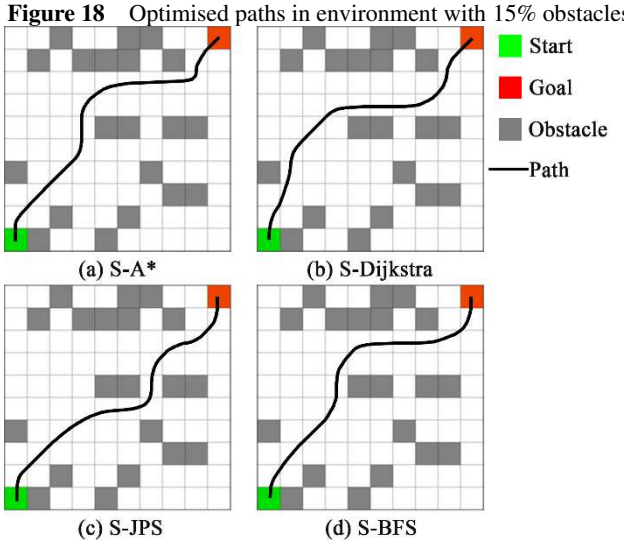


Figure 20 Optimised paths in environment with 20% obstacles

According to the simulated paths, as shown in Figs. 13–20, the current adaptive path optimization method yields smooth paths that satisfy the kinematic constraints of mobile robot in spite of different obstacle environments and original path plan algorithms. In particular, the path optimized by the current adaptive path optimization method is remarkably smooth if the original included angle between control sides is large and the control side is long. Although the optimal solution can also be derived using the current adaptive path optimization method when the original control side is short and the included angle between control sides is marginal, a large curvature of radius presents in the smooth path. In the case of the continuous turns at a marginal angle, the selection of the optimal control point would be affected significantly, and the algorithm may not converge. According to the case study on the environment with obstacle rate of 15%, the paths generated by JPS and BFS pass through the complex area at the bottom right. As a result, Z-shaped road sections with continuous 90° turns and a short spacing between turning points exhibit in the path. According to Fig. 18, the path is smoothened using the current adaptive path optimization method for the Z-shaped road sections. The path at the turning position has a larger curvature of radius of transited circular arcs and larger variations in heading angle compared with the paths at other turning positions. However, the paths generated by A* and Dijkstra's algorithm do not pass through the complex area in the roadmap. The optimised path exhibits overall smoothness, which facilitates the subsequent tracking and navigation of mobile robots. Therefore, the currently proposed adaptive path optimization method relies substantially on the original paths. It can be used to select the optimal control point and smoothen the path when Z-shaped road sections

present along the original paths. In addition, because the current adaptive path optimization method relies on the original paths and is used to select the optimal control point at the turning position and smoothen the path based on Bézier curve, the optimised path should be shorter than the original path. Moreover, the current adaptive path optimization method exhibits robustness in view of that it can be used in combination with post-processing optimization to smoothen paths planned using various path planning algorithms.

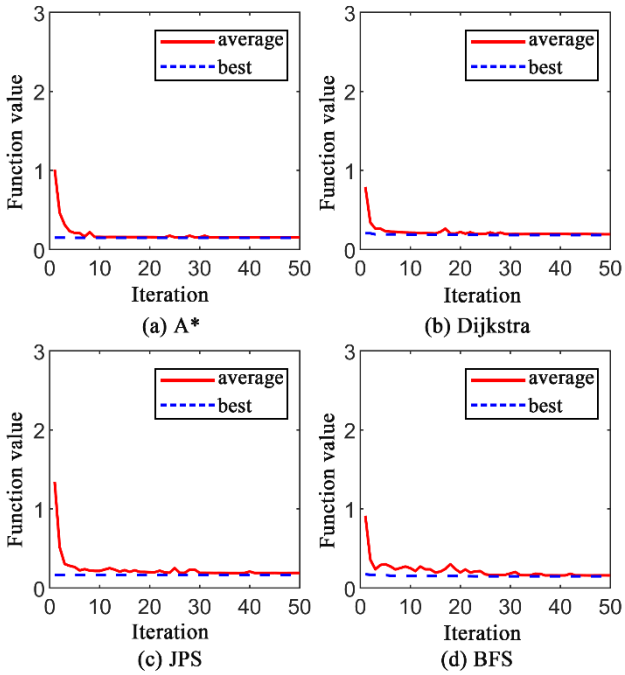


Figure 21 Convergence curves of genetic algorithm in environment with 5% obstacles

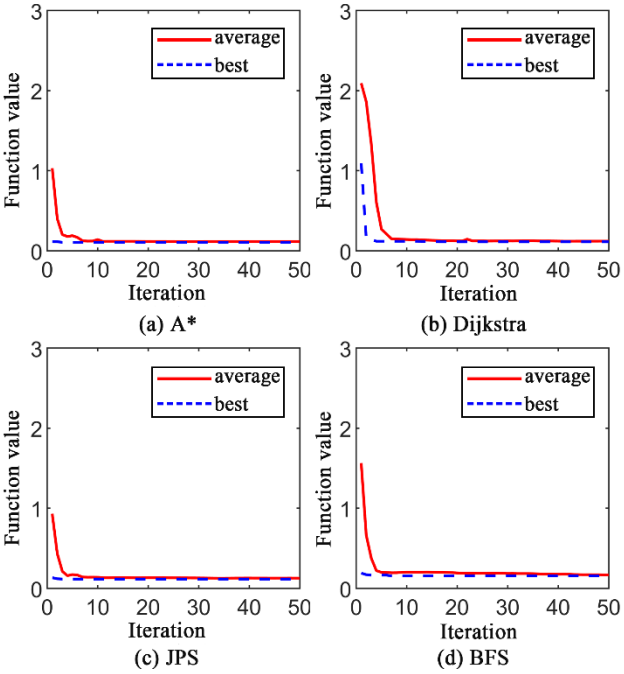


Figure 22 Convergence curves of genetic algorithm in environment with 10% obstacles

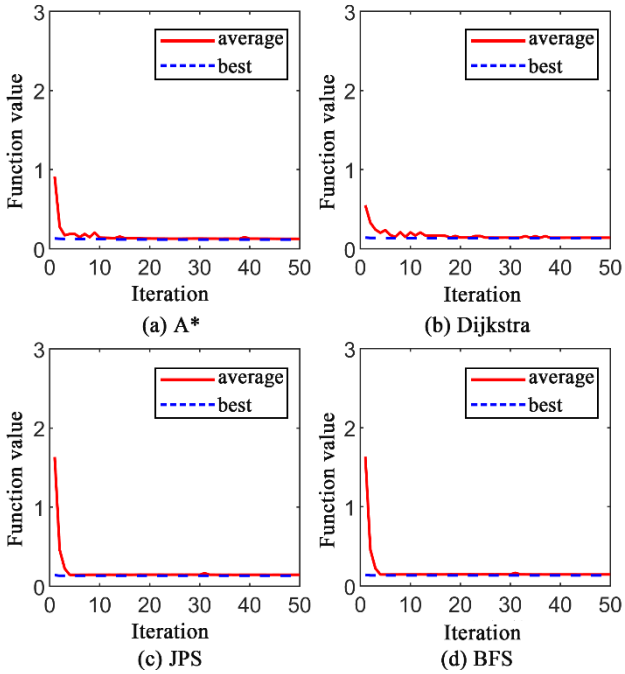


Figure 23 Convergence curves of genetic algorithm in environment with 5% obstacles

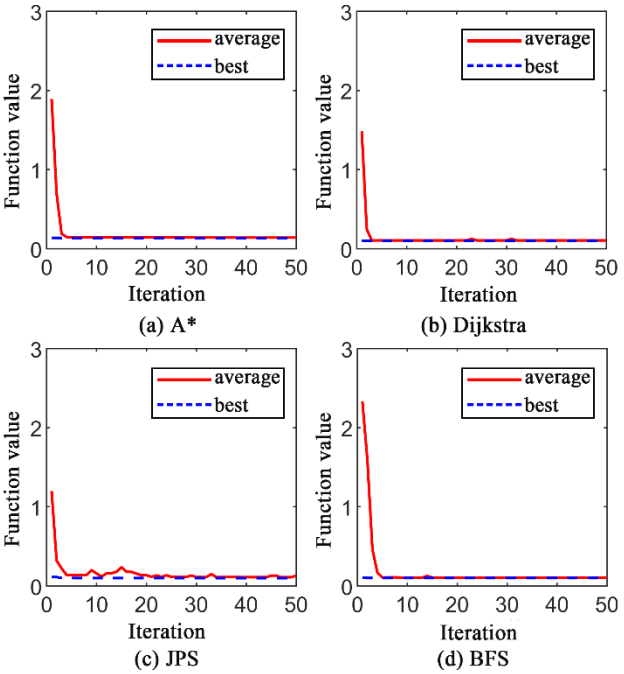


Figure 24 Convergence curves of genetic algorithm in environment with 10% obstacles

According to the convergence curves of the optimization algorithm, as shown in Figs. 21–24, the current adaptive

path optimization method displays advantages of high convergence speed in determining the optimal point. It can essentially achieve convergence after only 10 iterations in various environments. With increasing obstacle rate in the environment and the number of turning points in the original path, a large function value appears at the beginning of the algorithm iteration. It is attributed to the penalty mechanism for the fitness function triggered by the initial population, which is generated randomly by the genetic algorithm. After the genetic operations of selection and crossover are performed, the function value reduces rapidly and the algorithm converges quickly in determining the optimal control point.

4.3 Applicability of current adaptive path optimization method in extreme environment

Applicability of the current adaptive path optimization method in an extreme environment is then explored through a case study. The environment has a roadmap size of 20 × 20 and obstacle rate of 40%. Fig. 25 shows the path planned using A*. It can be observed that A* enables smooth path. Figs. 25–26 show the simulation results. In an environment with numerous obstacles and few feasible paths (see Fig. 25), still a smooth path from the initial position to the target position can be generated rapidly using A*, as shown in Fig. 25(a). In this original path, most turning angles and lengths of control sides are large. Nonetheless, Z-shaped road sections with continuous turns exhibit. Fig. 25(b) shows the improved path via the current adaptive path optimization method. According to Fig. 25(b), the original path has been smoothened. The overall smoothness of the optimized path is adequate, and the long control sides at the entry and exit positions exhibit moderate turning amplitude in the smoothened path even in the original Z-shaped road sections with continuous turns at (10, 5) and (10, 7). Thereby, the kinematic constraints of mobile robots are satisfied effectively.

According to the convergence curve of the optimization algorithm, as shown in Fig. 26, the large function value at the beginning of iteration is attributed to the substantial variation in quality among the individuals generated randomly at the beginning of iteration. Most individuals trigger the penalty mechanism of the algorithm, which results in a relatively high objective function value of the population. The algorithm converges rapidly with iterations because of genetic operations. It achieves convergence after 75 iterations.

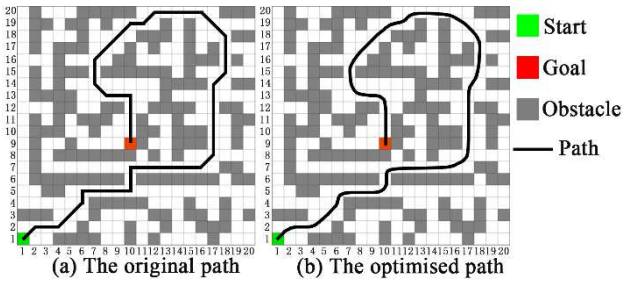


Figure 25 Original path and optimised path in extreme environment

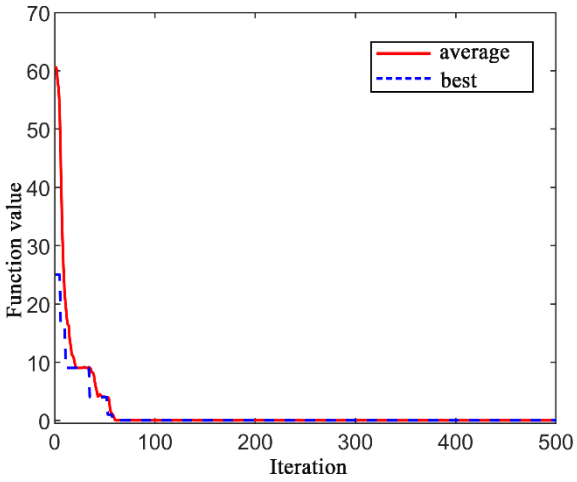


Figure 26 Convergence curves of genetic algorithm in extreme environment

4.4 Superior comprehensive performance of current adaptive path optimization method

The adaptive path optimization method proposed in this study is compared with the most prevailing path optimization methods, i.e., the quartic Bézier curve-based locus optimization method [24] and the path planning method that integrates the genetic algorithm and Bézier curve [25].

4.4.1 Comparison with quartic Bézier curve-based locus optimization method

The paths optimized by the currently proposed adaptive path optimization method and the quartic Bézier curve-based locus optimization method are compared with regard to aforementioned different environmental conditions. Table 7 shows the data by different methods pertaining to path length, number of corners, and path safety. Fig. 26 shows the paths optimized by the two methods. Specifically, paths 1 and 2 denote the optimal path by the currently proposed adaptive path optimization method and the quartic Bézier curve-based locus optimization method, respectively. Fig. 27 shows the convergence curves for the current adaptive path

optimization method in two scenarios.

Table 7 Comparison between simulation results by current adaptive path optimization method and quartic Bézier curve-based locus optimization method

	Methods	Path length	Number of corners	Path safety
Scene one	Paths 1	26.2	0	Moderate
	Paths 2	27.68	0	Good
Scene two	Paths 1	26.15	0	Moderate
	Paths 2	26.96	0	Good

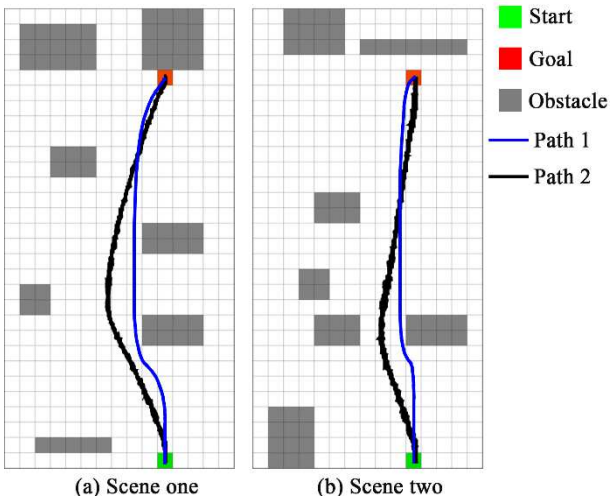


Figure 27 Comparison diagram between current adaptive path optimization method and the quartic Bézier curve-based locus optimization method

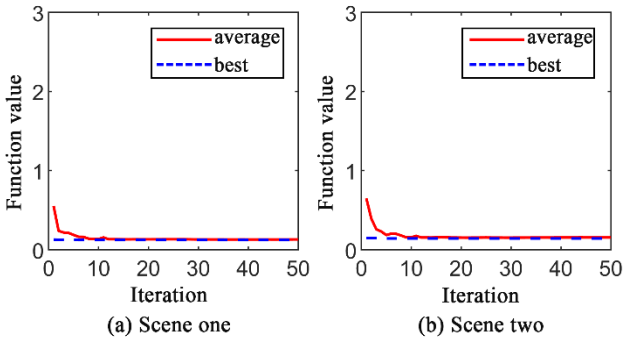


Figure 28 Convergence curves of genetic algorithm in two scenarios

Fig. 27 and Table 7 show that the optimal paths by the two methods are generally smooth and satisfy the kinematic constraints of mobile robots. Path 1 is marginally superior to path 2 in terms of path length. In addition, the quartic Bézier curve-based locus optimization method generates more alternative paths during the generation of path 2, and consumes substantial extra operation time and memory. On the other hand, the generation of path 1 depends significantly on the original paths planned by A* and JPS. After generation of the original paths, the paths can be smoothened rapidly at the turning positions. According to the convergence curves, as shown in Fig. 28, the genetic algorithm converges after only 10 iterations in the two scenarios. Therefore, less time is spent on the determination of the optimal control point during the generation of path 1. With regard to path safety, path 1 is closer to obstacles than path 2 notwithstanding that collision with obstacles does not occur. Therefore, path 2 is marginally superior to path 1 in terms of safety.

4.4.2 Comparison with path planning method integrating genetic algorithm and Bézier curve

The current adaptive path optimization method and the path planning method integrating the genetic algorithm and Bézier curve are then compared. Table 8 shows the simulation data of path length, number of corners, and path safety. Fig. 29 shows the paths optimized by the two methods. Specifically, paths 1 and 2 are the optimal paths by the current adaptive path optimization method and the path planning method integrating the genetic algorithm and Bézier curve, respectively. Fig. 30 shows the convergence curve for the current adaptive path optimization method.

Table 8 Comparison of simulation results by current adaptive path optimization method and path planning method integrating genetic algorithm and Bézier curve

	Paths 1	Paths 2
Path length	29.8871	29.6132
Number of corners	0	0
Path safety	Moderate	Bad

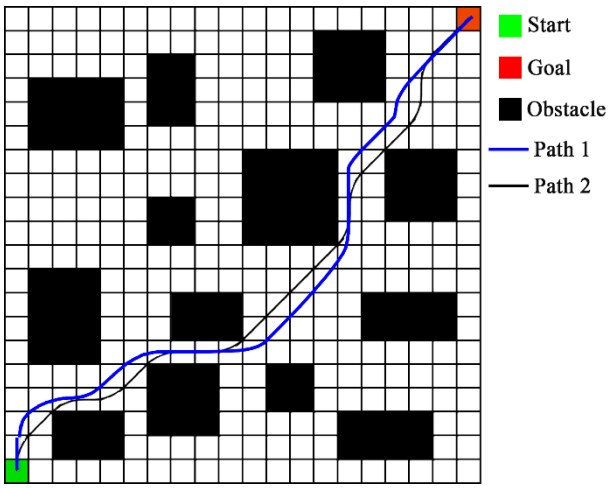


Figure 29 Comparison diagram between current adaptive path

optimization method and path planning method integrating genetic algorithm and Bézier curve

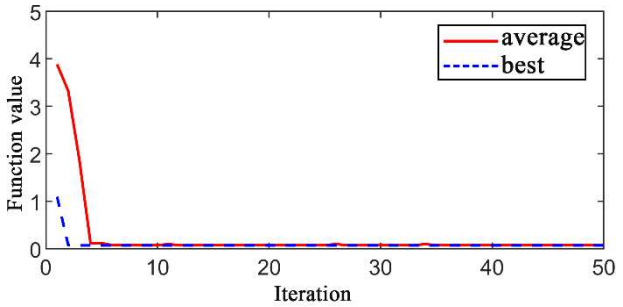


Figure 30 Genetic algorithm convergence curve of current adaptive path optimization method

Fig. 29 and Table 8 show the optimal paths by the current adaptive path optimization method and the path planning method integrating with the genetic algorithm and Bézier curve, respectively. Both the optimal paths are generally smooth. They are similar in terms of path length and direction of the planned path. Specifically, the path planning method integrating with the genetic algorithm and Bézier curve determines the original path using the genetic algorithm, and uses the quartic Bézier curve to smoothen the path at the turning point along the original path. In addition, it is similar to the current adaptive path optimization method because both the methods are used to smoothen the path based on the original path. However, the path planning method integrating with the genetic algorithm and Bézier curve uses the quartic Bézier curve to smoothen the path. Therefore, it requires significantly more operation time than the current adaptive path optimization method. Furthermore, the path planning method is used to select the control points for smoothening the path based on experience or the effect of generation of smooth path rather than on a scientifically credible and effective principle. The control point position determined by such a method is not necessarily optimal. It is also noteworthy that the safety of the path generated by the path planning method integrating with the genetic algorithm and Bézier curve is low because the path intersects with obstacles at the turning point. Therefore, the current adaptive path optimization method can be stated significantly superior to the path planning method integrating with the genetic algorithm and Bézier curve in terms of computational efficiency and path safety.

4.5 Experimental validation of current adaptive path optimization method

The applicability of the current adaptive path optimization method is tested experimentally using NanoPro series mobile robot as shown in Fig. 31. The mobile robot is equipped with devices of lidar, IMU, Jetson Nano, and motor. Its front and rear wheels are Ackermann steering wheel and driving wheel, respectively. Specifically, lidar is used to detect the external environmental information in real time; IMU acquires the position information of the mobile robot in real time; and Jetson Nano establishes communication with the upper computer, processes the environmental information obtained by sensors, sends the information to as well as receives commands from the upper computer.

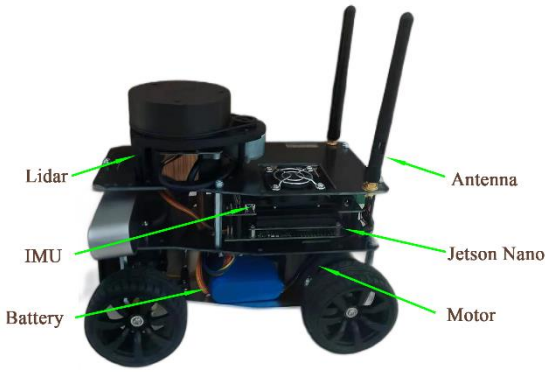


Figure 31 Mobile robot from Bingda Intelligent (Shenzhen) Co., Ltd

The experimental scenarios, namely scenario one and scenario two, are established as in Fig. 32. Furthermore, the raster maps, as shown in Fig. 33, is constructed via gmapping-based method and the actual scenarios in Fig. 32.

(a) Scenario one (b) Scenario two

Figure 32 Experimental scenarios

(a) Scenario one (b) Scenario two

Figure 33 Experimental scene maps constructed by gmapping

4.5.1 *Experimental scenario 1*

In experimental scenario 1, the base link coordinate system is constructed with the origin of geometric center of the mobile robot. The world coordinate system is established with the origin of the initial point of the mobile robot. For the mobile robot, the maximum velocity v_{max}, maximum angular velocity w_{max}, maximum acceleration a_{max}, and safe distance are set to be 0.2 m/s, 0.4 rad/s, 0.1 m/s^2, and 0.1 m, respectively. Two points are randomly selected as the target points to plan the path of the mobile robot. The position and motion curves and the path planned by the algorithm are plotted on the raster map based on the information from the mobile robot wheel odometer, as shown in Fig. 34. It can be observed that the path planned by the current adaptive algorithm roughly approximates the actual path of the mobile robot. However, it deviates substantially from the actual path at certain turning points and positions in the vicinity of obstacles. The deviations at the turning points can be attributed to that the current adaptive optimization algorithm optimises the locus based on the original locus only. Thus, the neglection of the constraints, e.g. maximum curvature and safety, and the influential factors of the velocity and acceleration of the mobile robot give rise to the deviations. The deviations between the planned and actual paths occur at positions in the vicinity of obstacles because a safe distance is artificially introduced in the experiment and the mobile robot thus deviates autonomously from the planned path to ensure its motion safety. Fig. 35 shows the convergence curve of the drawing algorithm. The algorithm convergence speed in scenario 1(a) is marginally higher than that in scenario 1(b). It is because the number of turning points in the planned path in Fig. 34(a) is larger than that in Fig. 34(b), whereby the algorithm convergence speed in scenario 1(a) is marginally lower. In addition, it is observed that the objective function value is relatively high at the beginning of algorithm operation. It is attributed to the presence of numerous turning points in the original path. Moreover, the penalty mechanism can be triggered directly in the initial population generated by the algorithm, which results in a relatively high objective function value in the initial population.

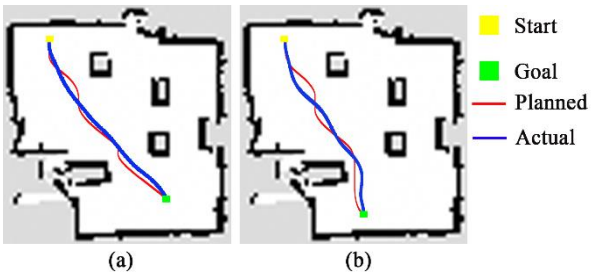


Figure 34 Results of experiment in experimental scenario 1

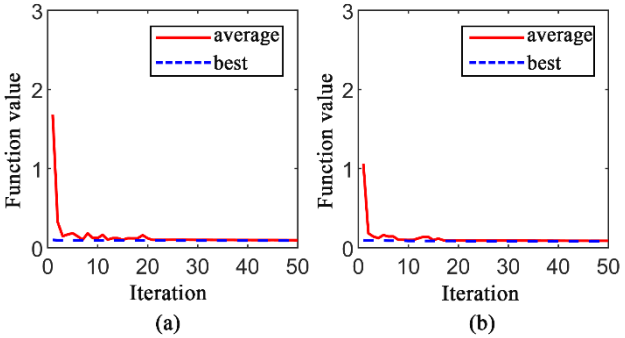


Figure 35 Convergence curve of genetic algorithm in experimental scenario 1

4.5.2 *Experimental scenario 2*

The experimental parameters in experimental scenario 2 are consistent with those in experimental scenario 1. The position and motion curves and the path planned by the algorithm are plotted on the raster map. Fig. 36 shows that marginal deviations between the path planned by the current adaptive algorithm and the actual path of the mobile robot for experimental scenario 2. In particular, the planned and actual paths are almost coincident in scenario 2(a). It is attributed to that the path planned in such a scenario is simple and has few turning points, and the original path is highly safe. The experimental results for scenario 2(b) are similar to those for scenario 1. Because the planned original path is highly safe, the deviations of the planned from actual paths are marginal at the positions in the vicinity of obstacles. However, significant deviations between planned and actual paths occur at the turning points along the path. This is because the influential factors, e.g. the velocity of the mobile robot, are omitted. Fig. 37 shows the convergence curve of the drawing algorithm. The algorithm convergence speed is high in scenario 2(a), and the penalty mechanism is not triggered by the algorithm at the beginning of algorithm operations. It is attributed to that the planned original path is relatively simple. However, the algorithm convergence speed is moderately high in scenario 2(b). The penalty mechanism is triggered at the beginning of algorithm operations owing to the presence of numerous turning points along the original path. It yields a relatively large objective function value at the beginning of iteration.

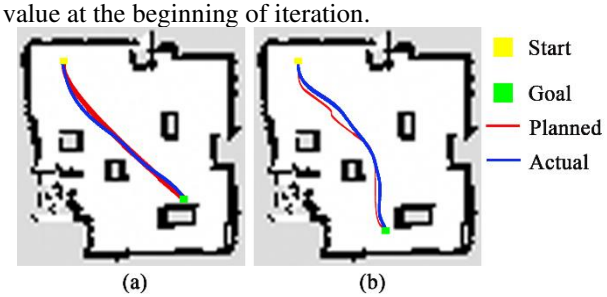


Figure 36 Results of experiment in experimental scenario 2

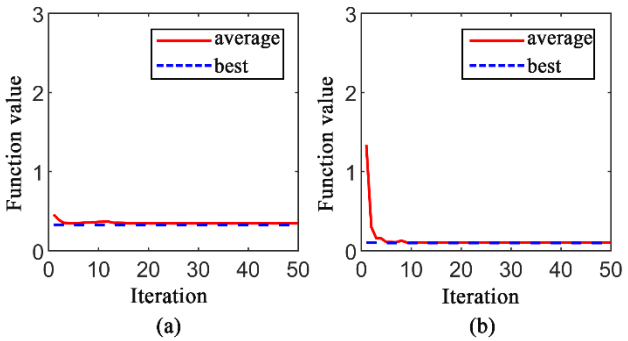


Figure 37 Convergence curve of genetic algorithm in experimental scenario 2

4.5.3 Tracking experiment

Considering the issues with regard to the turning points and at the positions close to obstacles along the planned path in the aforementioned experiments, the motion parameters of the mobile robot are modified in the tracking experiment to improve its tracking accuracy. Accordingly, v_{max}, w_{max}, and a_{max} are reset as 0.1 m/s, 0.2 rad/s, and 0.5 m/s^2, respectively, and the path is planned by the drawing algorithm as shown in Fig. 38. In the tracking experiment, the path planned by the currently proposed adaptive algorithm coincides with the actual path of the mobile robot without any deviation. This is attributed to three factors:

1.The motion parameters are modified and the velocity is reduced prior to the start of the experiment to prevent the motion path from deviating from the planned path as a result of the excessively high velocity at the entry and exit positions.

2.The randomly generated tracking path is relatively simple, and a large distance between the tracking path and obstacles exhibits in the tracking experiment. It prevents deviations from the planned path incurred by insufficient safety distance.

3.The tracking coordinate points are densely distributed along the tracking path, and the mobile robot sequentially passes through such points. It ensures the high tracking accuracy of the mobile robot in the tracking process.

Fig. 39 shows the convergence curve of the drawing algorithm. The algorithm convergence speed in scenario 1 is marginally higher than that in scenario 2. Furthermore, the initial objective function values in the algorithm are relatively low and the penalty mechanism is triggered in the two scenarios. The underlying reason is that the tracking paths in scenarios 1 and 2 are relatively simple, and the tracking path in scenario 1 is simpler than that in scenario 2 to yield a marginally lower initial objective function value and relatively higher convergence speed.

Figure 38 Results of tracking experiment

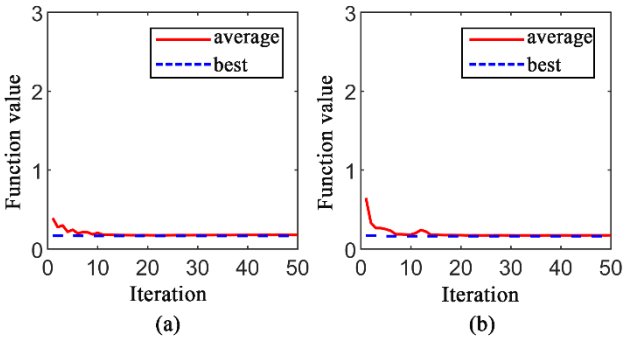


Figure 39 Convergence curves of genetic algorithm in tracking experiment

5 Conclusions

In this study, an adaptive path optimization method is proposed to address the issues of large number of turning points and non-smoothness of paths with regard to conventional path planning algorithms. The relationship between the length and included angle of the control side and the path offset in the quadratic Bézier curve is investigated. This relationship is adopted as the major component of the fitness function in the genetic algorithm. The optimal control point position satisfying the safety and kinematic constraints is determined at the turning point along the path. Furthermore, the transition is performed on the optimization effect of the mobile robot's working path using the quadratic Bézier curve and the control point position. Main conclusions of the study are drawn as below.

(1) BP neural network model constructed by the current adaptive path optimization method exhibits a high level of prediction accuracy. It provides a highly accurate and reliable mapping relationship for determining the optimal control point using the genetic algorithm.

(2) The current adaptive path optimization method transforms the path smoothening problem into an

optimization solution problem. It rapidly determines the position of the optimal control point based on comprehension of constraints, e.g. path safety and curvature, and the genetic algorithm. It lays the theoretical groundwork for smoothening the locus and path.

(3) The current adaptive path optimization method introduces the quadratic Bézier curve to avail smoothening the path at the turning points along the path based on the optimal control point position determined by the genetic algorithm. The optimized path satisfies the kinematic constraints of the mobile robot. It prevents the mobile robot motor from being powered on and off frequently, thus reduces the acceleration amplitude of the mobile robot and the peak value of the input current of the motor, which thus promotes the service life of the mobile robot.

6 Declaration

Acknowledgements

Not applicable.

Funding

Supported by National Natural Science Foundation of China (Grant No. 52175222), and Tianjin Science and Technology Program Project (Grant No. 19ZXZNGX00100)

Availability of data and materials

The datasets supporting the conclusions of this article are included within the article.

Authors' contributions

The author' contributions are as follows: SD was in charge of the whole trial; LZ wrote the manuscript; YL assisted with editing. All authors read and approved the final manuscript

Competing interests

The authors declare no competing financial interests.

Consent for publication

Not applicable

Ethics approval and consent to participate

Not applicable

References

[1] Zhang L, Wang Y, Wang Z. Robust lateral motion control for in-wheel-motor-drive electric vehicles with network induced delays. *IEEE Transactions on Vehicular Technology*, 2019, 68(11): 10585-10593.

[2] Jiang K, Yang D, Liu C, et al. A flexible multi-layer map model designed for lane-level route planning in autonomous vehicles. *Engineering*, 2019, 5(2): 305-318.

[3] Mathew R, Hiremath S S. Control of velocity-constrained stepper motor-driven hilare robot for waypoint navigation. *Engineering*, 2018, 4(4): 491-499.

[4] Dijkstra E W. A note on two problems in connexion with graphs. *Numerische mathematik*, 1959, 1(1): 269-271.

[5] Hart P E, Nilsson N J, Raphael B. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 1968, 4(2): 100-107.

[6] Harabor D, Grastien A. Improving jump point search. *24th International Conference on Automated Planning and Scheduling*, Menlo Park, USA: AAAI, 2014: 128-135.

[7] Traish J, Tulip J, Moore W. Optimization using boundary lookup jump point search. *IEEE Transactions on Computational Intelligence and AI in Games*, 2015, 8(3): 268-277.

[8] Zhao Jing, Wang Zheng, Huang Chengkan, et al. Mobile Robot Path Planning Based on an Improved A* Algorithm. *Robot*, 2018, 40(06): 903-910.

[9] Duan Shuyong, Wang Qifan, Han Xu, et al. An improved A-star algorithm for safety insured optimal path with smoothed corner turns. *Journal of Mechanical Engineering*, 2020, 56(18): 205-215.

[10] Boroujeni Z, Goehring D, Ulbrich F, et al. Flexible unit A-star trajectory planning for autonomous vehicles on structured road maps. *2017 IEEE international conference on vehicular electronics and safety (ICVES)*. 2017: 7-12.

[11] LIU Jiashun, LIU Jianhua, ZHANG Zhijing, et al. Anytime RRT Based Cable Automatic Routing under Three-dimensional Environment. *Journal of Mechanical Engineering*, 2016, 52(13): 156-1654.

[12] Wei Wu, Han Jin, Li Yanjie, et al. Path Planning of Mobile Robots Based on Dual-Tree Quick-RRT* Algorithm. *Journal of South China University of Technology (Natural Science Edition)* ,2021, 49(07): 51-58.

[13] Du M, Chen J, Zhao P, et al. An improved RRT-based motion planner for autonomous vehicle in cluttered environments. *2014 IEEE International Conference on Robotics and Automation (ICRA).* 2014: 4674-4679.

[14] Wang Haifang, Zhang Yao, Zhu Yakun, et al. Mobile Robot Path Planning Based on Improved Bidirectional RRT. *Journal of Northeastern University: Natural Science*, 2021, 42(8): 1065-1070.

[15] Khatib O. Real-Time Obstacle Avoidance System for Manipulators and Mobile Robots. *International Journal of Robotics Research*, 1986, 5(1): 90-98.

[16] Li C, Jiang X, Wang W, et al. A simplified car-following model based on the artificial potential field. *Procedia engineering*, 2016, 137: 13-20.

[17] Zhang C. Path planning for robot based on chaotic artificial potential field method. *IOP Conference Series: Materials Science and Engineering*. 2018, 317(1): 12-56.

[18] Jia Lihong. Research bidirectional search path of GPS-Based. Anhui: *Anhui University of Science and Technology*, 2017.

[19] Hu X, Chen L, Tang B, et al. Dynamic path planning for

autonomous driving on various roads with avoidance of static and moving obstacles. *Mechanical Systems and Signal Processing*, 2018, 100: 482-500.

[20] Wang Mingqiang, Wang Zhenpo, Zhang Lei. Local Path Planning for Intelligent Vehicles Based on Collision Risk Evaluation. *Journal of Mechanical Engineering*, 2021, 57(10): 28-41.

[21] Lyu Taizhi, Zhou Wu, Zhao Chunxia. Improved visibility graph method using particleswarm optimization and B-spline curve for path planning. *Journal of Huaqiao University: Natural Science*, 2018, 39(01): 103-108.

[22] Zhang Y, Chen H, Waslander S L, et al. Hybrid trajectory planning for autonomous driving in highly constrained environments. *IEEE Access*, 2018(6): 32800-32819.

[23] Gao Song, Zhang Jinwei, Rong Hui, et al. Application of unmanned vehicle local obstacle avoidance method based on Bezier curve. *Modern Electronics Technique* ,2019, 42(09): 163-166.

[24] Chen Cheng, He Yuqing, Bu Chunguang, et al. Feasible Trajectory Generation for Autonomous Vehicles Based on Quartic Bezier Curve. *Acta Automatica Sinica* ,2015, 41(3): 486-496.

[25] Cui Genqun, Hu Kerun, Tang Fengmin. Intelligent vehicle path planning based on genetic algorithm and Bézier curve. *Modern Electronics Technique*, 2021, 44(01): 144-148.

[26] Yu Lingli, Long Ziwei, Zhou Kaijun. Non-time trajectory tracking method based on Bezier curve for robot. *Chinese journal of scientific instrument*, 2016, 37(7): 1564-1572.

[27] Cimurs R, Hwang J, Suh I H. Bezier curve-based smoothing for path planner with curvature constraint. *International Conference on Robotic Computing (IRC)*. 2017: 241-248.

[28] Li Hai, JIang Tao, Su Xiaojie, et al. Fast and Safe Autonomous Trajectory Generation in Unknown Environments. *Control Engineering of China*, 2021, 28(11): 2153-2157.

[29] Zhang L, Li T, Zhang J, et al. Optimization on the Crosswind Stability of Trains Using Neural Network Surrogate Model. *Chinese Journal of Mechanical Engineering*, 2021, 34(1): 1-17.

[30] Li Weishuo, Sun Jian, Chen Wei. Real-time obstacle avoidance algorithm for robots based on BP neural network. *Chinese journal of scientific instrument*, 2019, 40(11): 204-211.

Biographical notes

Shu-Yong Duan, born in 1984, is currently an associate professor at *State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, China*. She received her PhD degree from *Hunan University, China*, in 2016. Her research interests include robot reliability, computational inverse techniques, intelligent robotics.
E-mail: duanshuyong@hebut.edu.cn

Lin-Xin Zhang, born in 1997, is currently a master candidate at *State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, China*.
E-mail: zlinxin@126.com

Xu Han, born in 1968, is currently a professor and a PhD candidate supervisor at *State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, China*. His main research interests include computational inverse techniques, optimization theory and algorithm.
E-mail: xhan@hebut.edu.cn

Yu-Le Li, born in 1997, is currently a master candidate at *State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, China*.
E-mail: 15735717718@163.com

Fang Wang, born in 1965, is currently a professor at *Hebei University of Technology, China*. She received her PhD degree from *Nanyang Technological University, Singapore*, in 1998. Her research interests include protective structure, high performance computing.
E-mail: mmmwangf@yahoo.com.sg

G.R. Liu, born in 1958, is currently a professor and a PhD candidate supervisor at *Department of Aerospace Engineering and Engineering Mechanics, University of Cincinnati, USA*. His main research interests include computational inverse techniques, intelligent perception of robots, intelligent algorithm, optimization theory and algorithm.
E-mail: liugr@uc.edu