

An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve[☆]

Baoye Song^a, Zidong Wang^{a,b,*}, Lei Zou^b

^a College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266590, China

^b Department of Computer Science, Brunel University London, Uxbridge, Middlesex, UB8 3PH, United Kingdom

ARTICLE INFO

Article history:

Received 28 October 2018

Received in revised form 11 September 2020

Accepted 25 November 2020

Available online 2 December 2020

Keywords:

Mobile robot

Continuous Bezier curve

Smooth path planning

Adaptive fractional-order velocity

Particle swarm optimization

ABSTRACT

In this paper, a new strategy is developed to plan the smooth path for mobile robots through an improved PSO algorithm in combination with the continuous high-degree Bezier curve. Rather than connecting several low-degree Bezier curve segments, the use of continuous high-degree Bezier curves facilitates the fulfillment of the requirement of high-order continuity such as the continuous curvature derivative, which is critical for the motion control of the mobile robots. On the other hand, the smooth path planning of mobile robots is mathematically an optimization problem that can be dealt with by evolutionary computation algorithms. In this regard, an improved particle swarm optimization (PSO) algorithm is proposed to tackle the local trapping and premature convergence issues. In the improved PSO algorithm, an adaptive fractional-order velocity is introduced to enforce some disturbances on the particle swarm according to its evolutionary state, thereby enhancing its capability of jumping out of the local minima and exploring the searching space more thoroughly. The superiority of the improved PSO algorithm is verified by comparing with several standard and modified PSO algorithms on some benchmark functions, and the advantages of the new strategy is also confirmed by several comprehensive simulation experiments for the smooth path planning of mobile robots.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

The purpose of path planning is to generate a collision-free route from the starting point to the destination, and the corresponding research has been ongoing ever since the inception and the implementation of mobile robots [1,2]. Technically, the problem of path planning can be formulated as an optimization problem that is subject to several constraints and performance criteria (e.g. shortest distance and obstacle-free route), see [3,4]. Up to now, an increasing amount of algorithms have been developed to devise an optimally planned path for mobile robots such as those using rapidly-exploring random tree (RRT) [5], artificial potential field [6], probabilistic roadmap method (PRM) [7], Voronoi-diagram [8], A* [9] and D* [10], etc.

[☆] This work was supported in part by the National Natural Science Foundation of China under Grants 61703242, 61703245, 61873148 and 61933007, the China Postdoctoral Science Foundation under Grant 2018T110702, the Postdoctoral Special Innovation Foundation of Shandong province of China Grant 201701015, and the Alexander von Humboldt Foundation of Germany.

* Corresponding author at: College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266590, China.

E-mail addresses: songbaoye@sdust.edu.cn (B. Song), Zidong.Wang@brunel.ac.uk (Z. Wang), Lei.Zou@brunel.ac.uk (L. Zou).

It is worth pointing out that, for most aforementioned path planning algorithms, the paths generated are usually polygonal lines which might contain undesirable sharp turns [11]. To track these paths, a robot has to frequently switch their motion states among 'stop', 'rotation' and 'restart' [12]. In this case, the switching of motion states will undoubtedly lead to discontinuous velocity and acceleration, which will further result in the occurrence of slippage and over actuation when a mobile robot is moving at a high speed [13]. Therefore, planning a *smooth* yet *optimal* path is an imperative strategy for the mobile robot to satisfy multiple optimization constraints such as low time/energy cost, high-speed movement, and complex service tasks [14,15].

In the context of smooth path planning of mobile robots, several algorithms have been presented in the past few years. For example, an algorithm has been put forward in [16] to plan smooth paths using an improved visibility graph, where the A* algorithm has been employed to search for a collision-avoidance path, which is then smoothed by using an optimized B-spline curve via the particle swarm optimization (PSO) scheme. An efficient continuous-curvature path-smoothing algorithm has been proposed in [17] where the parametric cubic Bezier curves are exploited to smooth the obstacle-avoidance path that is generated by fitting a sequence of way points. A four parameter logistic curve has been proposed and applied for the planning of the smooth robot path in [18], where the complete path can

Nomenclature/Abbreviation

APSO	Adaptive PSO
CPSO	Chaotic Particle Swarm Optimization
EC	Evolutionary Computation
FOPSO	Fractional-Order PSO
GA	Genetic Algorithm
MDPSO	Multimodal Delayed PSO
PRM	Probabilistic Roadmap Method
PSO	Particle Swarm Optimization
PSO-LDIW	PSO with Linearly Decreased Inertia Weight
PSO-TVAC	PSO with Time-Vary Acceleration Coefficients
RRT	Rapidly-exploring Random Tree
SDPSO	Switching Delayed PSO
WMR	Wheeled Mobile Robot

be achieved by connecting the smooth path subsections. In [19], a genetic algorithm (GA) has been applied to plan an optimal obstacle-free path for the mobile robot, where several piecewise cubic Hermite interpolating polynomials are applied for the smoothing of the generated optimal path. In [20], the segments of cubic Bezier curve have been joined together to obtain a smooth robot path with way points and corridor constraints. In [21], a new parallel PSO algorithm has been put forward to devise a linear feasible path for the mobile robot, and then such a path is smoothed by using the B-spline curve. In [22], the η^3 -splines have been employed to smoothly connect the endpoints of the linear path generated by using the MAKLINK Graph and Dijkstra algorithm.

In the literature mentioned above, the planned smooth paths are usually smoothed by using parametric curves (e.g. Bezier curve) after the procedure of linear path planning. As such, the devised paths would lose their optimality, because the procedure of path smoothing is not linked directly to the path optimization. To handle such an optimality loss issue, several methods have been proposed to *directly* design the smooth path of the mobile robot. For example, a Bézier curve-based model has been investigated for path planning, where a novel Chaotic Particle Swarm Optimization (CPSO) algorithm has been proposed to optimize the control points of Bézier curve [23]. However, this approach has not been utilized in the complex environment of mobile robot, where some motion constraints have to be considered in the path planning. The η^3 -splines have been combined with the parallel genetic algorithm in [24] to design a smooth path for the autonomous robot, where only the path length has been considered for the path optimization. In [25], a new genetic algorithm in combination with the Bezier curve has been investigated for the smooth path planning of mobile robot, but the devised optimal path is interrelated to the new grid-based mobile robot workspace, that is, the Bezier curve's control points have to be limited to the center of each grid (rather than arbitrary points of the workspace). To overcome this drawback, a new scheme combining PSO with cubic Bezier curves has been put forward in [26] to produce a smooth optimal path for the mobile robot, but it is very difficult to ensure the high-order continuity for the connected path segments.

In this paper, a new strategy is developed for the planning issue of the smooth robot path by using a continuous high-degree Bezier curve and an improved PSO algorithm. Instead of connecting several low-degree Bezier curve segments (e.g. cubic Bezier curves), we use the continuous high-degree Bezier curve

to fulfill the requirements of the smooth path planning (e.g. continuous curvature). Yet, the Bezier curve's control points should be optimized to ensure the performance of the smooth path, and this is actually an optimization problem that can be dealt with by evolutionary computation (EC) algorithms. In this regard, an improved PSO algorithm is proposed to tackle two typical EC-related issues, that is, local trapping and premature convergence. In the improved PSO algorithm, an adaptive fractional-order velocity is introduced to bring certain disturbances in the searching process, thereby enhancing the possibility for the particles to jump out of the local minima of the searching space and exploring the searching space more thoroughly.

The contributions of the current paper are outlined from the following three aspects. (1) *To satisfy the smoothness requirement on the planned path, a new approach using a continuous high-degree Bezier curve is developed, thereby removing the need to connect several segments of the smooth curves as done in the literature.* (2) *An improved PSO algorithm with adaptive fractional-order velocity is developed to handle two frequently occurred obstacles in the smooth robot path planning, namely, local trapping and premature convergence, and the superiority of the improved PSO algorithm is verified by several comprehensive simulation experiments on some well-known benchmark functions.* (3) *The issue of planning the smooth path for the mobile robot is mathematically formulated as an optimization problem. Then, the problem is solved by the newly developed strategy combining the continuous high-degree Bezier curve with the improved PSO algorithm. Finally, the simulation results demonstrate that the smooth paths produced by the new strategy can outperform the paths derived in our previous studies.*

The organization of the rest of paper is described as follows. The preliminary of the Bezier curve is briefly introduced in Section 2. In Section 3, the modeling of the mobile robot and the workspace are first described in detail. Then, the planning of smooth path of the mobile robot is formulated mathematically as an optimization problem. Furthermore, the improved PSO algorithm with adaptive fractional-order velocity is elaborated and discussed by comparing with several famous PSO algorithms upon some standard benchmark functions. In Section 4, the new strategy combining the continuous high-degree Bezier curve with the improved PSO algorithm is employed to devise the smooth robot path, and the superiority of the new strategy is illustrated by simulation experiments. Finally, Section 5 concludes this paper by pointing out some future works.

2. Preliminary on Bezier curve

As a kind of parametric curve, Bezier curve has been widely utilized in practical engineering [20] such as mechanical design and computer graphics [13]. The definition of a Bezier curve of degree n can be expressed by [15]:

$$\mathbf{P}(t) = \sum_{i=0}^n B_{i,n}(t) \mathbf{p}_i, \quad t \in [0, 1], \quad (1)$$

where $\mathbf{P}(t)$ represents the Bezier curve; t denotes the normalized variable of motion time; $\mathbf{p}_i = (x_i, y_i)^T$ indicates a vector of coordinate which consists of the components of the X-Y-coordinate of the i th control point; $B_{i,n}(t)$ ($i = 0, 1, \dots, n$) stands for the Bernstein polynomials of degree n , which actually expresses the Bezier curve's base functions that can be defined as

$$B_{i,n}(t) = C_n^i t^i (1-t)^{n-i} = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} \quad (2)$$

for $i = 0, 1, \dots, n$.

According to the definition, the first derivative of the Bezier curve can be expressed as

$$\dot{\mathbf{P}}(t) = \frac{\mathbf{P}(t)}{dt} = n \sum_{i=0}^{n-1} B_{i,n-1}(t)(\mathbf{p}_{i+1} - \mathbf{p}_i), \quad (3)$$

and its second derivative can be expressed by:

$$\ddot{\mathbf{P}}(t) = n(n-1) \sum_{i=0}^{n-2} B_{i,n-2}(t)(\mathbf{p}_{i+2} - 2\mathbf{p}_{i+1} + \mathbf{p}_i). \quad (4)$$

Hence, the curvature along the Bezier curve in the two-dimensional plane can be computed by:

$$\kappa(t) = \frac{\dot{p}_x(t)\ddot{p}_y(t) - \dot{p}_y(t)\ddot{p}_x(t)}{(\dot{p}_x^2(t) + \dot{p}_y^2(t))^{3/2}}, \quad (5)$$

where $\dot{p}_x(t)$ (respectively, $\dot{p}_y(t)$) and $\ddot{p}_x(t)$ (respectively, $\ddot{p}_y(t)$) indicate the X-Y-coordinate components of the first and second derivatives of the Bezier curve, respectively. Similarly, we can compute other derivatives of the Bezier curve iteratively based on the above expressions.

3. Methodologies of smooth path planning

3.1. Modeling of the mobile robot and the workspace

A differential driving wheeled mobile robot (WMR) is considered in this paper. The velocity model of the WMR has been shown in Fig. 1, where $v_l(t)$ and $v_r(t)$ indicate, respectively, the left and right velocities of the two wheels; l denotes the axial distance of the pair of wheels; and $\omega_R(t)$ and $v_R(t)$ represent, respectively, the angular and linear velocities of the WMR. Thus, the velocity model of the WMR is determined as follows [27,28]:

$$\omega_R(t) = \frac{v_r(t) - v_l(t)}{l/2}, \quad (6)$$

$$v_R(t) = \frac{v_l(t) + v_r(t)}{2}, \quad (7)$$

and the kinematics model of the WMR shown in Fig. 2 can be expressed by:

$$\begin{cases} \dot{\theta}_R(t) = \omega_R(t), \\ \dot{x}_R(t) = v_R(t) \cdot \cos \theta_R(t), \\ \dot{y}_R(t) = v_R(t) \cdot \sin \theta_R(t), \end{cases} \quad (8)$$

where $x_R(t)$ and $y_R(t)$ represent, respectively, the coordinate components of the position of the WMR, and $\theta_R(t)$ represents the orientation of the WMR. Correspondingly, $\dot{x}_R(t)$ and $\dot{y}_R(t)$ indicate the X-Y-coordinate components of the linear velocity, and $\dot{\theta}_R(t)$ denotes the anticlockwise angular velocity.

In this paper, the workspace of the WMR is supposed to be a two-dimensional plane, which is the same environment in [25] and [26] for the convenience comparison of the results. The workspace shown in Fig. 3 is a square plane that has been cut into $2^n \times 2^n$ grids, and each grid is a square of 10×10 units. For each of the grids, it is assigned a grid number and the black grid indicates a square space that is infeasible for the WMR because it is occupied by some obstacles, while the white grid indicates the obstacle-free space for the WMR. Obviously, the environment can be described in more details with a larger parameter n at the cost of increasing the computational load. Therefore, the parameter n should be selected according to the performance specification on the path planning and, in this paper, n is set as 4.

Note that the boundaries of the obstacles are extended in accordance with the robot size. Thus, the mobile robot can be taken as a unit point of the workspace regardless of its real size. To determine whether a unit point is feasible for the mobile robot, the following criterion is defined.

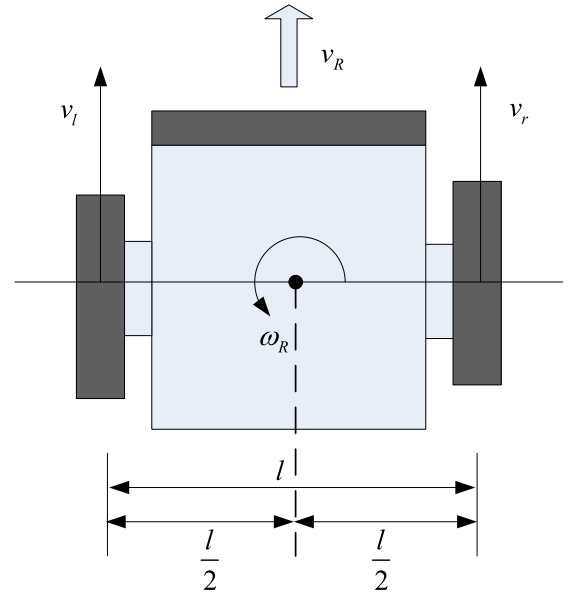


Fig. 1. Velocity model of the WMR.

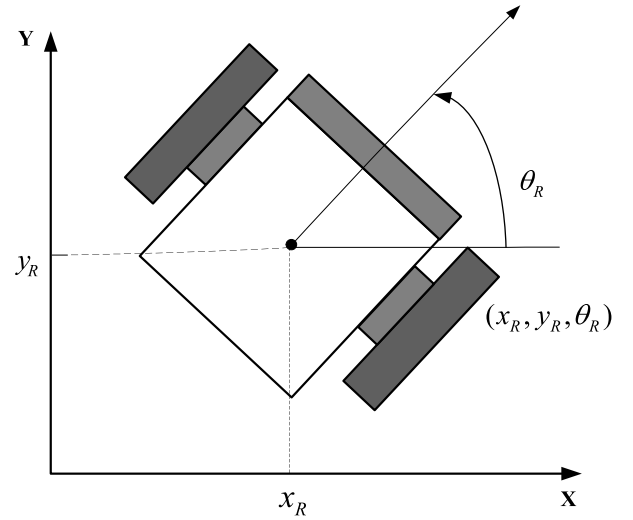


Fig. 2. Kinematics model of the WMR.

Definition 1. A unit point is feasible if and only if it is not in a black square grid, which contains the extended boundaries of the obstacles, i.e. $N_p \notin N_b$, where N_b indicates the number set of the black square grids; N_p indicates a numbered grid that contains the unit point, and the conversion from a unit point to a numbered grid can be expressed by:

$$N_p = \lfloor y(t)/10 \rfloor \times 16 + \lfloor x(t)/10 \rfloor, \quad (9)$$

where $\lfloor \cdot \rfloor$ represents the mathematical operation of rounding down; $x(t)$ and $y(t)$ indicate the X-Y-coordinate components of a unit point in the workspace.

3.2. Problem description

The objective of this paper is to seek a feasible yet optimal smooth path for the mobile robot, where the planned smooth path should fulfill the following criteria and constraints: (1) the smooth path planned in this paper should be a feasible route for the movement of the mobile robot, i.e. an obstacle-avoidance

240	241	242	243		246	247	248	249	250	251	252	253	254	255
224	225	226	227		230	231	232	233	234	235	236	237	238	239
208	209	210	211		214	215	216	217	218	219	220	221	222	223
192	193	194	195		198	199	200	201						
176	177	178	179	180	181	182	183	184	185	186	187	189	190	191
								168	169	170	171	173	174	175
								152	153	154	155	157	158	159
128	129			132	133	134	135	136	137	138	139	141	142	143
112	113			116	117	118	119	120	121	122	123	125	126	127
96	97			100	101	102	103	104	105	106	107	108	109	110
80	81	82	83	84	85	86		89	90	91				
64	65	66	67	68	69	70		73	74	75				
48	49	50	51	52	53	54		57	58	59	60	61	62	63
32	33	34	35					41	42	43	44	45	46	47
16	17	18	19	20	21	22		25	26	27	28	29	30	31
0	1	2	3	4	5	6		9	10	11	12	13	14	15

Fig. 3. Mobile robot workspace.

route; (2) the devised smooth path should satisfy the smoothness requirement of high-order continuity, e.g. a G^3 -path [29]; and (3) the curvature and its derivative of the generated smooth path should be minimized to benefit the path tracking of the mobile robot.

To meet the aforementioned three requirements of the smooth path planning task, a continuous high-degree Bezier curve is employed to produce a satisfactory smooth path by optimizing the Bezier curve's control points. Note that we no longer need to connect several Bezier curve segments (e.g. the cubic Bezier curves) as done in most existing literature, and this makes it possible to link the path smoothing directly to the path optimization. In this case, the planning of smooth path in this paper is essentially an optimization problem that seeks a series of control points, which lead to a feasible and optimal Bezier curve for the mobile robot path. Accordingly, the objective function of such an optimization problem can be expressed by:

$$\begin{aligned} \min \quad & J = w_1 \times \|\mathbf{P}(t)\| + w_2 \times \max |\kappa(t)| + w_3 \times \max |\dot{\kappa}(t)|, \\ & t \in [0, 1], \\ \text{s.t.} \quad & \mathbf{P}(t) \notin N_b, \mathbf{P}(t) \in C^3, \end{aligned} \quad (10)$$

where w_i ($i = 1, 2, 3$) denotes the weighting coefficient of the above cost function; $\|\cdot\|$ indicates the path length; $|\cdot|$ represents the absolute value of the curvature and curvature derivative; and C^3 stands for the set of functions with the continuity until the third derivative. Obviously, the smooth path planning of this paper is a rather complicated optimization problem whose solution is prone to fall into the local minima. To tackle this issue, an improved PSO algorithm is in order and will be discussed later.

Remark 1. It is worth mentioning that the path consisting of a continuous high-degree Bezier curve outperforms the planned path generated by connecting several Bezier curve segments. On the one hand, compared to the path composed of Bezier curve segments, it is more natural for the continuous high-degree Bezier curve path to satisfy the requirement of high-order continuity. On the other hand, with the continuous high-degree Bezier curve path, the computation cost can be much reduced in the process of path optimization because the elements of the

objective function in Eq. (10) (e.g. the curvature derivative of the path) can be expressed in the analytic form and therefore accurately calculated.

3.3. The improved PSO algorithm

PSO is an intelligent optimization algorithm presented in [30] for simulating the swarm behaviors of animals [30], for example, the flocks of fish and birds. Recently, PSO has been widely utilized to search for an optimal solution of various optimization problems in robotics, e.g. the optimal design of controller, autonomous navigation, and path planning of mobile robot, etc.

For PSO, one particle in the swarm acts as a possible optimal solution of an optimization problem. The particle, which represents a candidate solution to the optimization problem, can fly around to explore an optimal solution of a certain searching space. For every particle of the swarm, the velocity and position at the k th iteration will be updated in the next iteration as follows:

$$v_i^{k+1} = wv_i^k + c_1r_1(x_{ib}^k - x_i^k) + c_2r_2(x_{gb}^k - x_i^k), \quad (11)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1}, \quad (12)$$

where x_i^k and v_i^k of the i th particle represent the position and velocity at the k th iteration, respectively; x_{ib}^k and x_{gb}^k indicate, respectively, the i th particle's best position (p_{best}) and the best position of the global swarm (g_{best}) till iteration k ; w is the velocity inertia weight; c_1 and c_2 denote, respectively, the coefficients of the particle acceleration; r_1 and r_2 are two real numbers that are randomly produced and distributed on $[0, 1]$ uniformly.

Up to now, several improved PSO algorithms have been proposed in the literature for the performance promotion of the original PSO. For example, PSO with linearly decreased inertia weight (which is denoted as PSO-LDIW) has been presented by Shi and Eberhart to linearly decrease the inertia weight of the particle velocity [31]. PSO with time-vary acceleration coefficients (which is denoted as PSO-TVAC) has been developed by Ratnaweera and his colleagues [32] to adjust the two acceleration coefficients of the velocity updating function in each iteration. However, the parameters of the above improved PSOs could not be regulated adaptively according to the state of evolution of the particle swarm. To adaptively calculate the acceleration coefficients and inertia weight of the PSO algorithm, an adaptive PSO (which is denoted as APSO) was presented in [33] to switch among several velocity updating functions in accordance with the current evolutionary state that is determined by the calculated evolutionary factor of the particle swarm. Based on the strategies introduced in [33], the switching PSO (which is denoted as SPSO) was proposed in [34] to change the velocity updating function based on the predicted evolutionary state in terms of a Markov chain. Furthermore, a random delay was introduced into the velocity updating function of the switching delayed PSO (which is denoted as SDPSO) in [35] on account of the evolutionary state of the swarm. Moreover, a multimodal delayed PSO (which is denoted as MDPSO) was presented in [26] to improve the performance of convergence and robustness by adaptively adding two novel terms in the velocity updating function. Nevertheless, there are some rather stringent requirements on these modified algorithms, e.g. the computational burden and memory storage.

To overcome the defects of the modified PSO algorithms mentioned above and effectively dispose of the problem of planning a smooth robot path, an improved PSO algorithm motivated by the algorithm of PSO with fractional-order velocity (which is denoted as FOPSO) [36] is developed in this paper and the updating functions of velocity and position are as follows:

$$v_i^{k+1} = \alpha v_i^k + \frac{1}{2}\alpha(1-\alpha)v_i^{k-1} + \frac{1}{6}\alpha(1-\alpha)(2-\alpha)v_i^{k-2}$$

$$+ \frac{1}{24}\alpha(1-\alpha)(2-\alpha)(3-\alpha)v_i^{k-3} + c_1r_1(x_{ib}^k - x_i^k) + c_2r_2(x_{gb}^k - x_i^k), \quad (13)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1}, \quad (14)$$

where c_1 and c_2 are, respectively, calculated as follows:

$$c_1 = (c_{1i} - c_{1f}) \times \frac{k_{\max} - k}{k_{\max}} + c_{1f}, \quad (15)$$

$$c_2 = (c_{2i} - c_{2f}) \times \frac{k_{\max} - k}{k_{\max}} + c_{2f}, \quad (16)$$

where c_{1i} (c_{2i}) and c_{1f} (c_{2f}) denote, respectively, the initial and final values of the acceleration coefficients c_1 and c_2 ; k_{\max} denotes the maximum iteration; and the parameter α indicates the fractional order of the velocity in [36], while in this paper, α is a variable that is linearly and adaptively regulated according to the evolutionary state of the swarm, and it is calculated by

$$\alpha = 0.9 - \frac{1}{1 + e^{-E_f^k}} \times \frac{k}{k_{\max}}, \quad (17)$$

where E_f^k denotes an evolutionary factor that can reflect the state of the swarm evolution at the k th iteration and is calculated by

$$E_f^k = \frac{d_{gb}^k - d_{\min}^k}{d_{\max}^k - d_{\min}^k}, \quad (18)$$

where d_{\max}^k and d_{\min}^k denote the maximum and minimum of an average distance, respectively, from one particle to others in the swarm; d_{gb}^k denotes the average distance of the g_{best} ; while the average distance at the k th iteration (which is denoted as d_i^k) can be calculated for the i th particle by

$$d_i^k = \frac{1}{S-1} \sum_{j=1, j \neq i}^S \sqrt{\sum_{k=1}^D (x_i^k - x_j^k)^2}, \quad (19)$$

where S and D indicate, respectively, the population size of the particle swarm and the dimension of each particle. The algorithm of the improved PSO is completely described in Fig. 4.

Remark 2. In FOPSO, the fractional-order velocities have been introduced into the updating function of traditional PSO algorithm. The additional fractional-order velocity terms are essentially some kind of delays of velocity, which can be regarded as some “disturbances” that will bring some “power” for the particles to jump out of the local minima. Thus, a better optimization result can be obtained owing to the extended search space. It is worth noting that the property of historical memory in the fractional-order velocities can play a similar role as the additional terms in MDPSO, but few memory storage and computational capacity are required in the FOPSO algorithm. Similarly, the velocity of the next iteration is related to not only the velocity of the current iteration but also the velocities of previous three iterations in the improved PSO algorithm. This strategy can bring some “useful” disturbances to the process of convergence, which makes it possible for particles to fly out of the local minima of the search space. Meanwhile, the fractional order of the velocities is linearly and adaptively regulated in terms of the state of the swarm evolution, which makes it possible for the particle to explore and exploit the whole search space more thoroughly so as to obtain an optimum of the global space.

To test the improved PSO algorithm, several simulation experiments are implemented upon some frequently used benchmark functions in this paper. The benchmark functions are formulated as Eqs. (20)–(23), all of them are some typical evaluation functions whose solutions are difficult to obtain. The configuration of

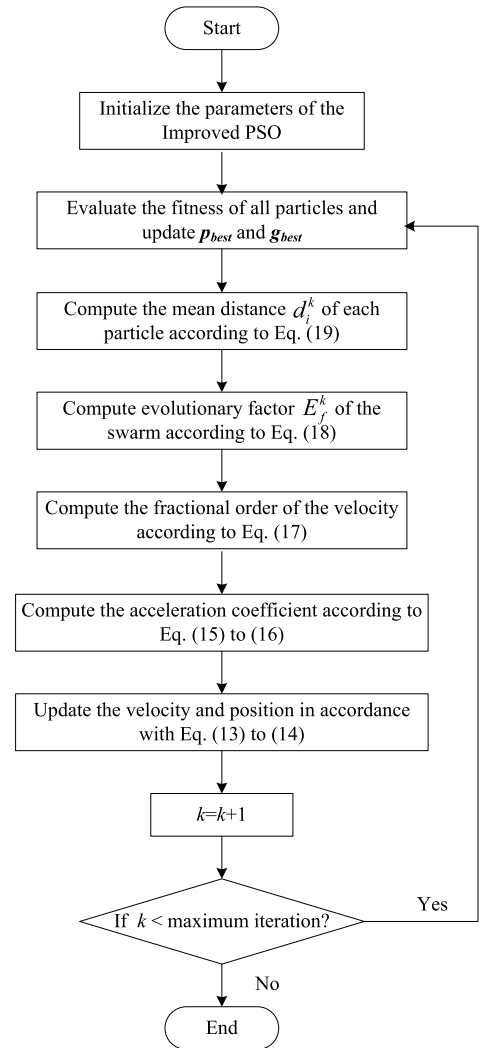


Fig. 4. Flowchart of the improved PSO algorithm.

these functions is demonstrated as Table 1, in which the last three columns list, respectively, the searching range of each dimension, the threshold of a successful optimal solution, and the theoretical optimum value of each benchmark function.

$$\text{Sphere : } f_1(x) = \sum_{i=1}^D x_i^2, \quad (20)$$

$$\text{Rosenbrock : } f_2(x) = \sum_{i=1}^{D-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2, \quad (21)$$

$$\begin{aligned} \text{Penalized I : } f_3(x) = & \frac{\pi}{D} \{ 10 \sin^2(\pi y_1) \\ & + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] \\ & + (y_D - 1)^2 \} + \sum_{i=1}^D u(x_i), \end{aligned} \quad (22)$$

$$y_i = 1 + 1/4(x_i + 1),$$

$$u(x_i) = \begin{cases} 100(-x_i - 10)^4, & x_i < -10, \\ 0, & |x_i| \leq 10, \\ 100(x_i - 10)^4, & x_i > 10. \end{cases}$$

Table 1
Configuration of some standard benchmark functions.

Function	Name	Dimension	Searching range	Threshold	Optimum
$f_1(x)$	Sphere	20	$[-100, 100]^D$	0.01	0
$f_2(x)$	Rosenbrock	20	$[-30, 30]^D$	100	0
$f_3(x)$	Penalized 1	20	$[-50, 50]^D$	0.01	0
$f_4(x)$	Schwefel 2.21	20	$[-100, 100]^D$	0.01	0

$$\text{Schwefel 2.21: } f_4(x) = \max\{|x_i|\}, \quad 1 \leq i \leq D \quad (23)$$

To compare the improved PSO algorithm with some well-known PSOs, the parameters of the simulations are set as follows. The size of the swarm S equals to 20, the dimension of the particle D equals to 20, the maximum iteration N equals to 20000 in the experiments, and the number of repetition T equals to 50 in each experiment. For each of the benchmark functions, the performance of the improved PSO is evaluated by comparing with several standard or modified PSOs, including the PSO-LDIW [31], PSO-TVAC [32], SDPSO [35], MDPSO [26] and FOPSO [36].

The logarithmic mean values of the fitness functions are shown in Figs. 5–8, in which the variation of the logarithmic mean fitness with the increasing of iteration has been clearly demonstrated by using diverse mark and line styles. Meanwhile, the statistics of the optimization solutions have also been listed in Table 2 to further compare the performance of the PSO algorithms, including the achievable optima, the mean values and standard deviations of the optima as well as the success rates of the PSO algorithms for every benchmark functions.

From Figs. 5–8, it can be clearly observed that the improved PSO outperforms all other standard and modified PSOs. For instance, the Sphere function is a typical unimodal benchmark function to test the rate of convergence of the intelligent optimization algorithms. Obviously, the mean fitness of the improved PSO algorithm converges much faster than the algorithms of PSO-LDIW and FOPSO as shown in Fig. 5, though the latter ones can achieve slightly better final optimization results. For the Rosenbrock function, which is usually treated as a multimodal benchmark function hard to achieve an optimum, the improved PSO algorithm performs perfectly comparing with other PSO algorithms on both the convergence rate and the optimum result. For the Penalized 1 function, several PSO algorithms, i.e. PSO-TVAC, MDPSO, FOPSO and the improved PSO, have similar performance on the achievable optimization results. However, the convergence rate of the improved PSO is faster than MDPSO and FOPSO and a bit slower than its ancestor PSO-TVAC. This performance is benefited from the linearly and adaptively regulated fractional-order velocity, which can not only maintain the convergence rate of standard PSO algorithm but also make the exploitation and exploration of the searching space more thoroughly. The above-mentioned performance of the improved PSO can also be verified by the optimization results of the Schwefel 2.21 function, where the improved PSO outperforms all the other PSO algorithms on both the convergence rate and the achievable optimum result.

The superior performance of the improved PSO can also be confirmed by the statistics of the optimization results in Table 2. For example, the improved PSO can obtain an optimum much better than the other PSOs except PSO-LDIW. For the Rosenbrock function, the improved PSO is the winner on the items of the achievable optimum, the mean value and standard deviation of the optimization results, and the success rate. The similar statistics results on the Penalized 1 function can also affirm the relationship between the improved PSO and the other ones. Actually, the improved PSO is not only the descendant of PSO-TVAC and FOPSO but also similar to the other PSOs on the updating scheme. Moreover, it is clear that the superiority of the improved PSO can also be demonstrated based on the statistics results on the Schwefel 2.21 function. Thus, the improved PSO

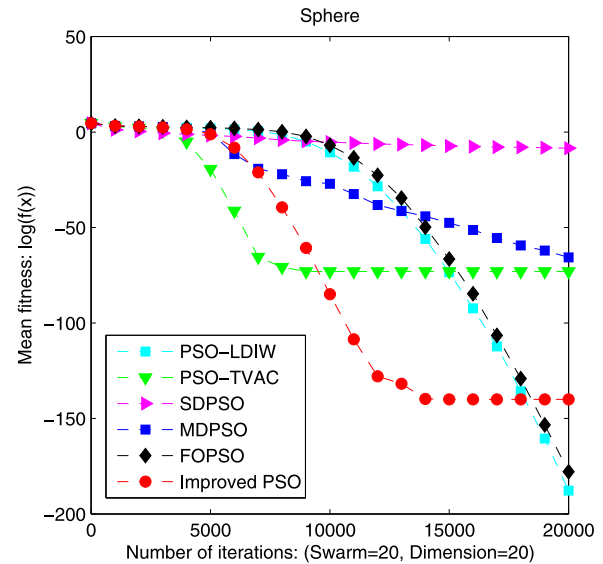


Fig. 5. Mean fitness of function Sphere.

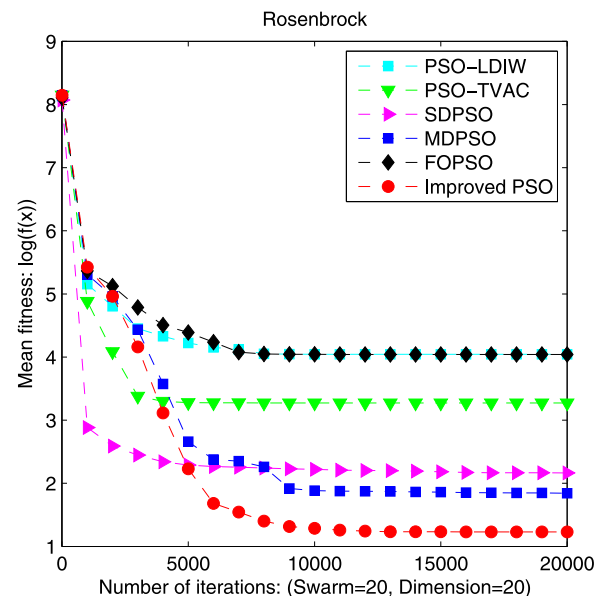


Fig. 6. Mean fitness of function Rosenbrock.

can outperform the other PSOs in Table 2 from the view of comprehensive evaluation on the statistic results. Additionally, it is worth pointing out that both of the improved PSO and MDPSO have a better success rate than the other PSOs. But, the improved PSO requires less memory space comparing with the MDPSO, which will randomly select the p_{best} and g_{best} of the previous iterations in the process of optimization.

Table 2
Statistics of the optimization results.

		PSO-LDIW	PSO-TVAC	SDPSO	MDPSO	FOPSO	Improved PSO
$f_1(x)$	Optimum	3.40×10^{-202}	5.23×10^{-138}	1.31×10^{-4}	4.98×10^{-105}	1.69×10^{-193}	1.74×10^{-201}
	Mean	1.67×10^{-188}	9.49×10^{-74}	3.29×10^{-9}	2.52×10^{-66}	1.27×10^{-178}	9.01×10^{-141}
	Std. Dev.	0.00	6.70×10^{-73}	9.97×10^{-9}	1.44×10^{-65}	0.00	6.37×10^{-140}
	Rate	100%	100%	100%	100%	100%	100%
$f_2(x)$	Optimum	1.32×10^{-4}	1.97×10^{-2}	1.00×10^{-1}	3.46×10^{-4}	1.22×10^{-3}	3.86×10^{-11}
	Mean	1.10×10^4	1.87×10^3	1.46×10^2	6.97×10^1	1.09×10^4	1.69×10^1
	Std. Dev.	2.94×10^4	1.27×10^4	5.95×10^2	4.26×10^2	2.94×10^4	7.88×10^1
	Rate	76%	94%	94%	98%	80%	98%
$f_3(x)$	Optimum	2.35×10^{-32}	2.35×10^{-32}	1.34×10^{-14}	2.35×10^{-32}	2.35×10^{-32}	2.35×10^{-32}
	Mean	3.11×10^{-3}	2.35×10^{-32}	6.31×10^{-9}	2.35×10^{-32}	2.36×10^{-32}	2.36×10^{-32}
	Std. Dev.	2.19×10^{-2}	8.29×10^{-48}	2.02×10^{-8}	8.29×10^{-48}	3.83×10^{-34}	3.83×10^{-34}
	Rate	98%	100%	100%	100%	100%	100%
$f_4(x)$	Optimum	9.52×10^{-16}	2.12×10^{-12}	1.15×10^{-2}	1.74×10^{-14}	2.68×10^{-14}	3.63×10^{-23}
	Mean	4.33×10^{-13}	1.93×10^{-8}	7.42×10^{-2}	2.74×10^{-11}	2.05×10^{-12}	1.07×10^{-17}
	Std. Dev.	9.17×10^{-13}	4.56×10^{-8}	5.29×10^{-2}	4.97×10^{-11}	3.60×10^{-12}	4.86×10^{-17}
	Rate	100%	100%	0%	100%	100%	100%

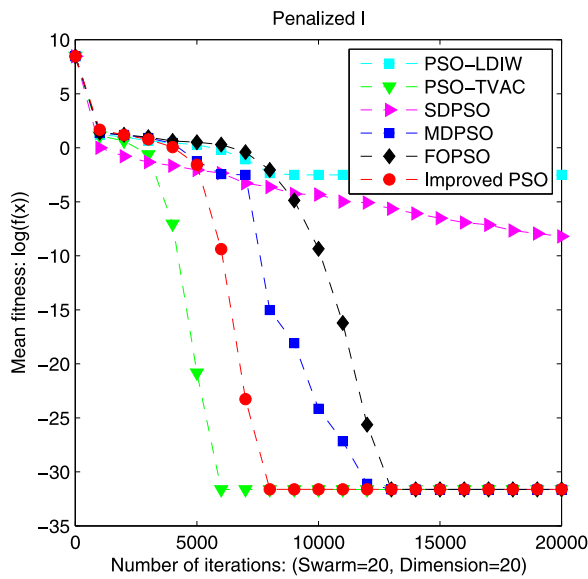


Fig. 7. Mean fitness of function Penalized I.

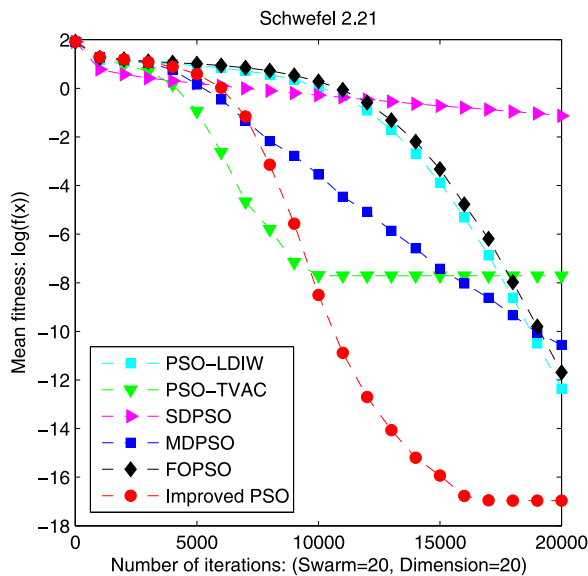


Fig. 8. Mean fitness of function Schwefel 2.21.

4. Simulation results

The new strategy combining the continuous high-degree Bezier curve with the improved PSO algorithm has been employed in this section to handle the issue of planning smooth path for the mobile robot in the workspace of Fig. 3. The parameters for the simulation experiments are depicted by: the number of control points is taken as 7 for the continuous high-degree Bezier curve, the size of population is set as 50 for the particle swarm, the maximum value is set as 100 for the iteration, the ratio of penalty is taken as 30 for the path points violating the obstacle-free constraint, and the weighting coefficients w_i ($i = 1, 2, 3$) are empirically taken as 1, 10000 and 1000 in this paper, respectively.

To demonstrate the advantages of the new strategy, several simulation experiments have been implemented to produce an optimal smooth robot path in the workspace of Fig. 3, and two of the experiment results have been illustrated in Figs. 9 and 10. In the two cases of the simulation experiments, the starts and destinations are set, respectively, from Grid 0 to Grid 255 and Grid 15 to Grid 240. The specific coordinate values of the grids are taken as (5, 5) for Grid 0, (155, 155) for Grid 255, (155, 5) for Grid 15, and (5, 155) for Grid 240. In Figs. 9 and 10, the Bezier curve's control points are denoted by the blue hollow circles; the convex hull of the high-degree continuous Bezier curve is composed of the blue solid lines; and the devised optimal smooth path is indicated by the red solid curve. In order to contrast the performance of the generated smooth paths, the counterpart results of the strategy combining the MDPSO with the cubic Bezier curve segments [26] have been illustrated in Figs. 11 and 12. Though the smooth paths seem to be similar with each other in Fig. 9 (respectively, Fig. 10) and Fig. 11 (respectively, Fig. 12), there are essential differences on these smooth paths. Take Fig. 9 and its counterpart Fig. 11 for an example. The smooth path in Fig. 9 is a continuous high-order Bezier curve, while the smooth path in Fig. 11 is composed of several cubic Bezier curve segments. The smooth path consisted of curve segments can be included in the convex hull composed of a feasible linear path, which itself is a feasible path from the start to the destination. Nevertheless, it can be observed clearly from Fig. 9 that the convex hull of the continuous high-degree Bezier curve is no longer a feasible linear path, i.e. it cannot be segmented and then smoothed by the curves like cubic Bezier curves. It is no doubt that the optimization problem is more difficult for the smooth path planning using continuous high-order Bezier curve. But, the continuous curve has more advantages on the continuity of the smooth path than the segment curves.

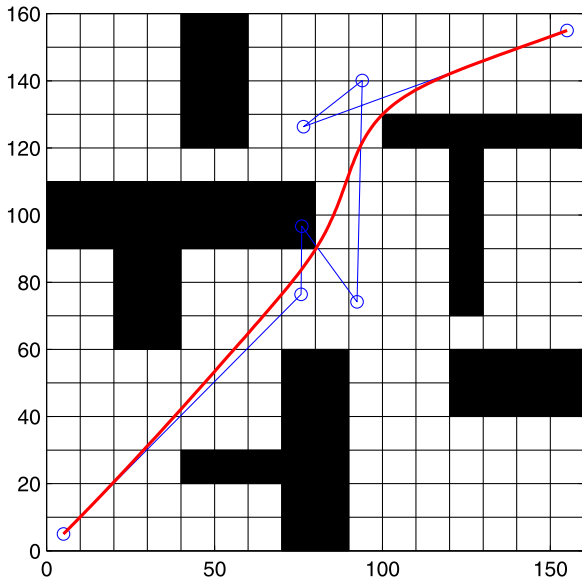


Fig. 9. Smooth path I (Continuous high-degree Bezier curve + Improved PSO)

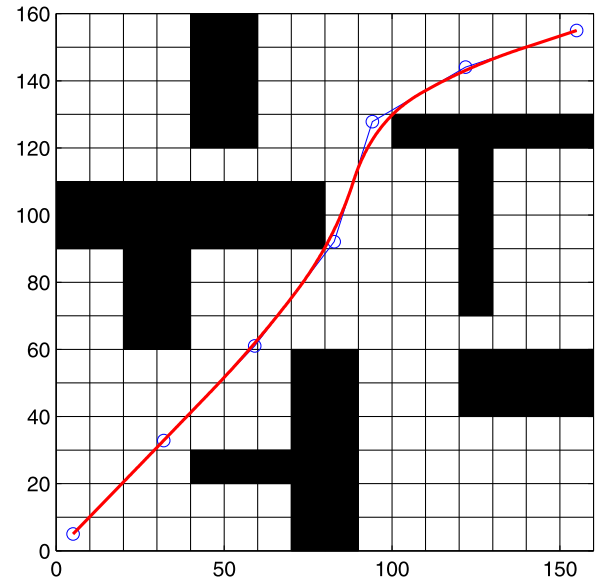


Fig. 11. Smooth path I (Cubic Bezier curve segments + MDPSO)

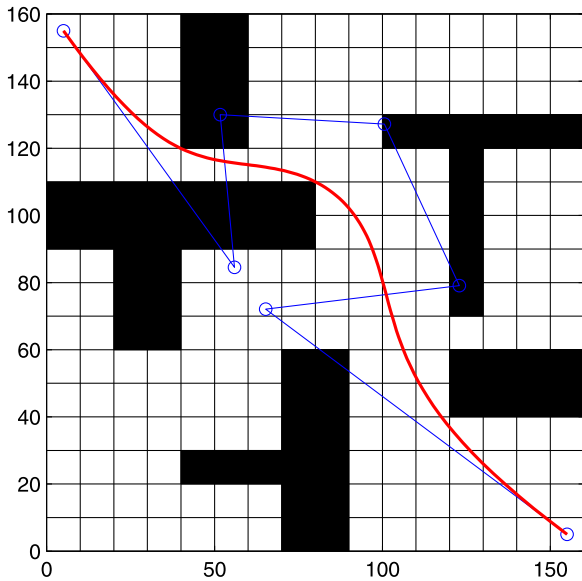


Fig. 10. Smooth path II (Continuous high-degree Bezier curve + Improved PSO)

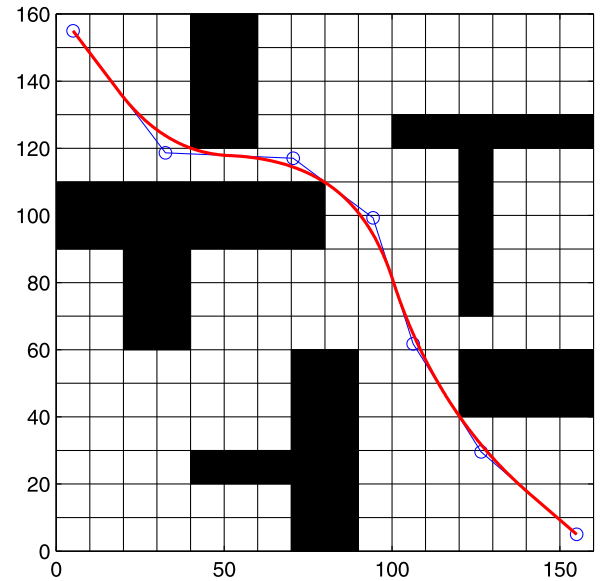


Fig. 12. Smooth path II (Cubic Bezier curve segments + MDPSO)

Meanwhile, the corresponding curvature curves of these smooth paths have been depicted in Figs. 13 and 14, respectively. Obviously, the curvature of the smooth paths produced via the strategy of this paper is continuous, which ensures the continuous movement of the mobile robot and makes the tracking of the smooth path much easier. However, there are several curvature steps at some of the joints of the counterpart smooth path segments, which will inevitably lead to several frequent switches of the motion state of mobile robot. This will undoubtedly result in discontinuous velocity and acceleration of the mobile robot, which can cause the slippage and over actuation in the high-speed moving cases. Besides, it is really a complicated task to achieve the smooth connection of the path segments, especially for the high-order continuity of the smooth robot path. In contrast, the continuous high-order Bezier curve can guarantee the high-order continuity of the path naturally avoiding the complicated work to connect the smooth path segments, not to mention the criteria on high-order continuity can be linked directly to the

smooth path optimization problem. The aforementioned observations can be further confirmed by the curvature derivatives of the smooth paths as shown in Figs. 15 and 16. Apparently, the smooth curvature derivatives imply the continuous curvatures of the smooth paths generated using the proposed strategy. Meanwhile, the steps on the curvature derivatives of the counterpart smooth path segments would lead to discontinuous curvatures, thereby causing the adverse motion of the mobile robot.

Remark 3. It is worth noting that the planning of smooth path in this paper is essentially a challenging optimization problem for the optimizer to seek an optimum smooth path for the WMR in the cases of Figs. 9 and 10. It is very difficult and complex to accomplish this task because of the following reasons at least. Firstly, a feasible optimum smooth robot path has to lie in a narrow valley full of obstacles nearby, thus it is very likely to be trapped into a local optimum solution that actually is not a feasible smooth path. Secondly, the convex hull of the continuous

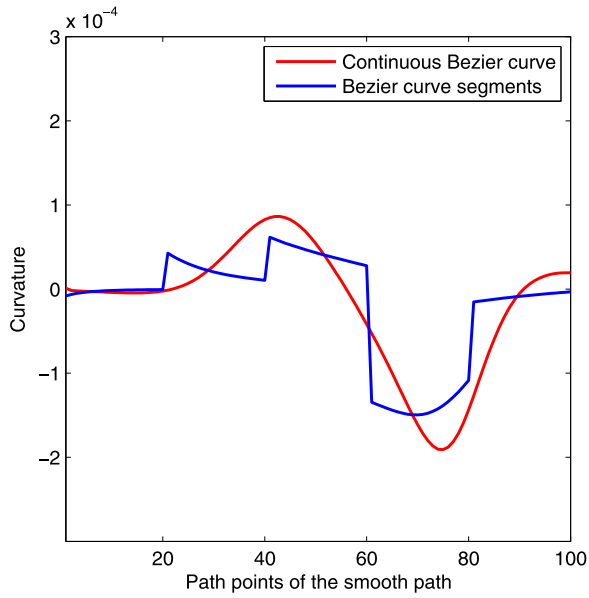


Fig. 13. Curvature of smooth path I.

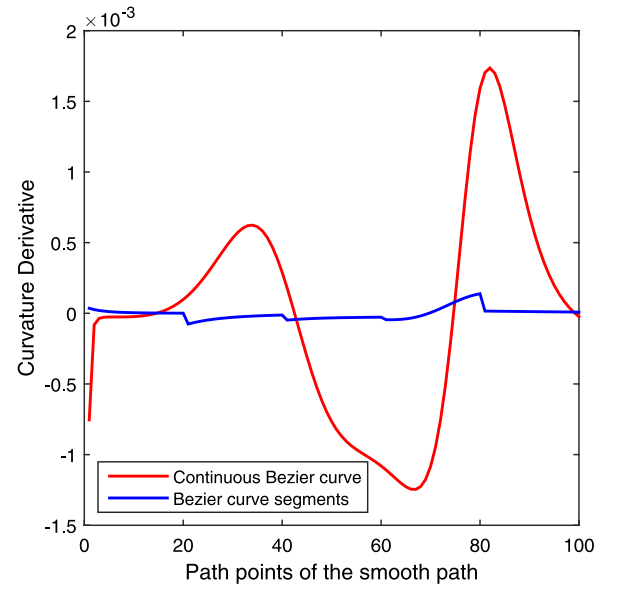


Fig. 15. Curvature derivative of smooth path I.

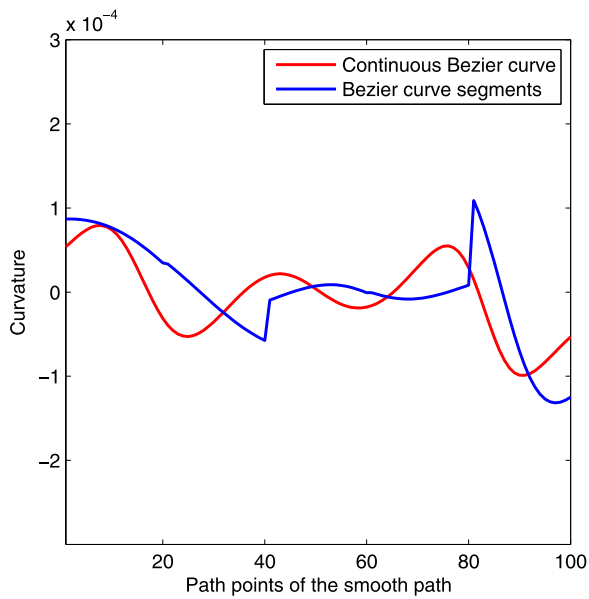


Fig. 14. Curvature of smooth path II.

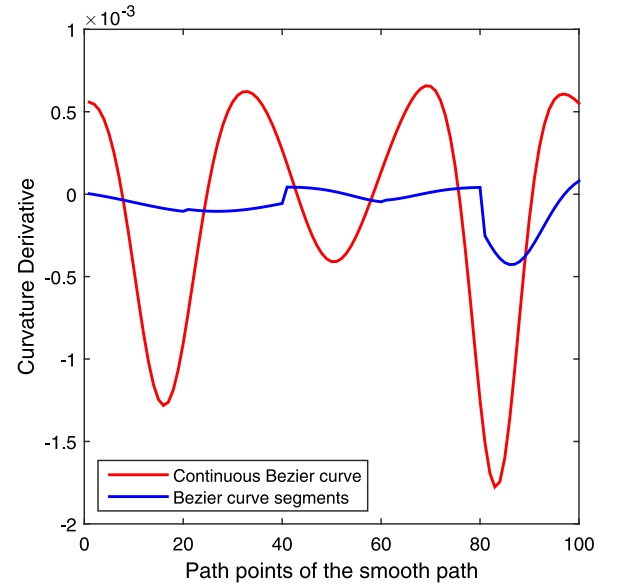


Fig. 16. Curvature derivative of smooth path II.

high-degree Bezier curve usually will be no longer a feasible linear path in contrast with the cubic Bezier curve segments of Figs. 11 and 12, which will make the strategy of smoothing the linear path not be workable. Thirdly, the criteria of minimum curvature and curvature derivative is linked to the optimization of the smooth path, which makes the issue of planning smooth path a multimodal optimization problem that is prone to trap into the local optimum solution. Fortunately, the challenging task of this paper can be successfully accomplished by using the strategy combining the continuous high-degree Bezier curve with the improved PSO algorithm proposed in this paper, which can achieve advanced performance in comparison with the approaches presented in previous papers [25,26].

5. Conclusions

A new strategy combining the continuous high-degree Bezier curve with an improved PSO algorithm has been proposed in this paper for the planning of the smooth robot path. In comparison with the smooth path planning approaches in the literature, the proposed new strategy can improve the smooth path planning of mobile robot from the following aspects at least. On the one hand, the smooth paths produced via the continuous high-degree Bezier curve are more “smoother” than those composed of several low-degree Bezier curve segments. That is because the high-order continuity (e.g. continuous curvature derivative) can be naturally satisfied along the smooth path made up of continuous high-degree Bezier curve. On the other hand, the objective function in Eq. (10) can be calculated accurately with a low computation cost, since its elements (e.g. curvature derivative of the path) can be expressed in analytic form. Moreover, an optimal smooth

path can be obtained based on the new strategy, because the requirements of continuity are linked directly to the optimization criteria rather than connecting several smooth curve segments as done in the literature. Hence, the proposed new strategy will undoubtedly benefit the motion control of mobile robot.

To deal with the hard optimization problem of the continuous high-degree Bezier curve, an improved PSO is proposed to overcome the frequently encountered problems of local trapping and premature convergence. By introducing an adaptive fractional-order velocity into the velocity updating function according to the state of the particle swarm evolution, some “disturbances” will be brought to the particle swarm. As such, some “power” will make it more likely to jump out of the local minima, so as to explore and exploit the searching space more thoroughly. Thus, the problems of local trapping and premature convergence can be greatly improved. The superiority of the improved PSO algorithm can be verified by comparing with several standard and modified PSO algorithms on several well-known standard benchmark functions. Furthermore, the advantage of the new strategy upon the criteria of the formulated optimization problem has also been confirmed by the simulation results of smooth path planning.

In future research, we will focus on some interesting topics including (1) the determination of the weight coefficients of the fractional-order velocities in a parameter domain that could guarantee the convergence of the algorithm, as well as the combination of some new strategies that can further promote the performance of the improved PSO, e.g. the switching delay [37], the randomly distributed delay [38], the sigmoid-function-based adaptive weight [39], and the Markovian state jumping [40]; (2) the combination of multiobjective optimization algorithms with the developed approaches to further improve the performance, see e.g. [41,42] for novel intelligent algorithms capable of finding the global optimum of high-dimensional problem efficiently; and (3) the application of the developed approaches to some more complicate situations such as state estimation [43–45], path planning of multi-robot [46,47], dynamic motion planning [48,49], and three-dimensional trajectory generation [50–52], etc.

CRedit authorship contribution statement

Baoye Song: Conceptualization, Methodology, Software, Writing - original draft. **Zidong Wang:** Supervision, Conceptualization, Methodology, Writing - review & editing. **Lei Zou:** Methodology, Software, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] M. Elbanhawi, M. Simic, Sampling-based robot motion planning: a review, *IEEE Access* 2 (1) (2014) 56–77.
- [2] D. Gonzalez, J. Perez, V. Milanés, F. Nashashibi, A review of motion planning techniques for automated vehicles, *IEEE Trans. Intell. Transp. Syst.* 17 (4) (2016) 1135–1145.
- [3] A. Atiyabi, D.M.W. Powers, Review of classical and heuristic-based navigation and path planning approaches, *Int. J. Adv. Comput. Technol.* 5 (2013) 1–14.
- [4] D. Lin, B. Shen, Y. Liu, F.E. Alsaadi, A. Alssadi, Genetic algorithm-based compliant path planning: an improved bi-RRT-based initialization method, *Assem. Autom.* 37 (3) (2017) 261–270.
- [5] S.M. LaValle, Rapidly-exploring random-tree: a new tool for path planning, *Res. Rep.* (1998) 293–308.
- [6] M.H. Kim, J.H. Heo, Y. Wei, M.C. Lee, A path planning algorithm using artificial potential field based on probability map, in: *Proceedings of IEEE 8th International Conference on Ubiquitous Robots and Ambient Intelligence*, 2011, pp. 41–43.
- [7] L.E. Kavraki, P. Svestka, J.-C. Latombe, M.H. Overmars, Probabilistic roadmaps for path planning in high-dimensional configuration spaces, *IEEE Trans. Robot. Autom.* 12 (1996) 566–580.
- [8] P. Bhattacharya, M.L. Gavrilova, Roadmap-based path planning using the voronoi diagram for a clearance-based shortest path, *IEEE Robot. Autom. Mag.* 15 (2) (2008) 58–66.
- [9] X. Liu, D. Gong, A comparative study of A-star algorithms for search and rescue in perfect maze, in: *Proceedings of International Conference on Electric Information and Control Engineering*, 2011, pp. 24–27.
- [10] A. Stentz, Optimal and efficient path planning for unknown and dynamic environments, *Int. J. Rob. Autom.* 10 (3) (1993) 89–100.
- [11] B. Song, G. Tian, F. Zhou, A comparison study on path smoothing algorithms for laser robot navigated mobile robot path planning in intelligent space, *J. Inform. Comput. Sci.* 7 (14) (2010) 2943–2950.
- [12] T. Berglund, A. Brokni, H. Jonsson, M. Staffanson, I. Soderkvist, Planning smooth and obstacle avoiding B-spline paths for autonomous mining vehicles, *IEEE Trans. Autom. Sci. Eng.* 7 (1) (2010) 167–172.
- [13] M. Elhoseny, A. Tharwat, A.E. Hassanien, Bezier curve based path planning in a dynamic field using modified genetic algorithm, *J. Comput. Sci.* 25 (2018) 339–350.
- [14] L. Huang, H. Qu, P. Ji, X. Liu, Z. Fan, A novel coordinated path planning method using k-degree smoothing for multi-UAVs, *Appl. Soft Comput.* 48 (2016) 182–192.
- [15] F. Zhou, B. Song, G. Tian, Bezier curve based smooth path planning for mobile robot, *J. Inform. Comput. Sci.* 8 (12) (2011) 2441–2450.
- [16] T. Lv, M. Feng, A smooth local path planning algorithm based on modified visibility graph, *Modern Phys. Lett. B*, 31 (19–21) 1740091.
- [17] K. Yang, S. Sukkarieh, An analytical continuous-curvature path-smoothing algorithm, *IEEE Trans. Robot.* 26 (3) (2010) 561–568.
- [18] S. Upadhyay, A. Ratnoo, Continuous-curvature path planning with obstacle avoidance using four parameter logistic curves, *IEEE Robot. Autom. Lett.* 1 (2) (2016) 609–616.
- [19] A. Bakdi, A. Hentout, H. Boutami, A. Maoudj, O. Hachour, B. Bouzoula, Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control, *Robot. Auton. Syst.* 89 (2017) 95–109.
- [20] J. Choi, R. Curry, G. Elkaim, Path planning based on Bezier curve for autonomous ground vehicles, in: *Proceedings of IEEE IAENG Special Edition of the World Congress on Engineering and Computer Science*, 2008, pp. 158–166.
- [21] H.-C. Huang, FPGA-based parallel metaheuristic PSO algorithm and its application to global path planning for autonomous robot navigation, *J. Intell. Robot. Syst.* 76 (2014) 475–488.
- [22] S. Zhang, L. Sun, Z. Chen, X. Lu, J. Liu, Smooth path planning for a home service robot using η^3 -splines, in: *Proceedings of IEEE International Conference on Robotics and Biomimetics*, 2014, pp. 1910–1915.
- [23] A. Tharwat, M. Elhoseny, A.E. Hassanien, T. Gabel, A. Kumar, Intelligent Bézier curve-based path planning model using chaotic particle swarm optimization algorithm, *Cluster Comput.* 22 (2019) 4745–4766.
- [24] H. Chang, J. Liu, High-quality path planning for autonomous mobile robots with η^3 -splines and parallel genetic algorithms, in: *Proceedings of IEEE International Conference on Robotics and Biomimetics*, 2009, pp. 1671–1677.
- [25] B. Song, Z. Wang, L. Sheng, A new genetic algorithm approach to smooth path planning for mobile robots, *Assem. Autom.* 36 (2) (2016) 138–145.
- [26] B. Song, Z. Wang, L. Zou, On global smooth path planning for mobile robots using a novel multimodal delayed PSO algorithm, *Cogn. Comput.* 9 (1) (2017) 5–17.
- [27] S. Han, B. Choi, J. Lee, A precise curved motion planning for a differential driving mobile robot, *Mechatronics* 18 (2008) 486–494.
- [28] Y. Liu, W. Xu, A.M. Dobaie, Y. Zhuang, Autonomous road detection and modeling for UGVs using vision-laser data fusion, *Neurocomputing* 275 (2018) 2752–2761.
- [29] A. Piazzi, L.B.C. Guarino, M. Romano, η^3 -Splines for the smooth path generation of wheeled mobile robots, *IEEE Trans. Robot.* 23 (5) (2007) 1089–1095.
- [30] J. Kennedy, R. Eberhart, Particle swarm optimization, in: *Proceedings of IEEE International Conference on Neural Network*, 1995, pp. 1942–1948.
- [31] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: *Proceedings of IEEE International Conference on Evolutionary Computation*, 1998, pp. 69–73.
- [32] A. Ratnaweera, S.K. Halgamure, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, *IEEE Trans. Evol. Comput.* 8 (2004) 240–255.
- [33] Z. Zhan, J. Zhang, Y. Li, H. Chung, Adaptive particle swarm optimization, *IEEE Trans. Syst. Man Cybern. Part B* 39 (6) (2009) 1362–1381.

- [34] Y. Tang, Z. Wang, J. Fang, Parameters identification of unknown delayed genetic regulatory networks by a switching particle swarm optimization algorithm, *Expert Syst Appl.* 38 (2011) 2523–2535.
- [35] N. Zeng, Z. Wang, H. Zhang, F.E. Alsaadi, A novel switching delayed PSO algorithm for estimating unknown parameters of lateral flow immunoassay, *Cogn. Comput.* 8 (2016) 143–152.
- [36] E.J.S. Pires, J.A.T. Machado, P.B.M. Oliveira, J.B. Cunha, L. Mendes, Particle swarm optimization with fractional-order velocity, *Nonlinear Dynam.* 61 (2010) 295–301.
- [37] N. Zeng, H. Qiu, Zidong Wang, W. Liu, H. Zhang, Y. Li, A new switching-delayed-PSO-based optimized svm algorithm for diagnosis of Alzheimer's disease, *Neurocomputing* 320 (2018) 195–202.
- [38] W. Liu, Z. Wang, X. Liu, N. Zeng, D. Bell, A novel particle swarm optimization approach for patient clustering from emergency departments, *IEEE Trans. Evol. Comput.* 23 (4) (2019) 632–644.
- [39] W. Liu, Z. Wang, Y. Yuan, N. Zeng, K. Hone, X. Liu, A novel sigmoid-function-based adaptive weighted particle swarm optimizer, *IEEE Trans. Cybern.* (2019) <http://dx.doi.org/10.1109/TCYB.2019.2925015>, in press.
- [40] I.U. Rahman, Z. Wang, W. Liu, B. Ye, M. Zakarya, X. Liu, An N-state Markovian jumping particle swarm optimization algorithm, *IEEE Trans. Syst. Man Cybern. - Syst.* (2019) <http://dx.doi.org/10.1109/TSMC.2019.2958550>, in press.
- [41] Y. Liu, Q. Cheng, Y. Gan, Y. Wang, Z. Li, J. Zhao, Multi-objective optimization of energy consumption in crude oil pipeline transportation system operation based on exergy loss analysis, *Neurocomputing* 332 (2019) 100–110.
- [42] Y. Liu, S. Chen, B. Guan, P. Xu, Layout optimization of large-scale oil-gas gathering system based on combined optimization strategy, *Neurocomputing* 332 (2019) 159–183.
- [43] S. Liu, Z. Wang, Y. Chen, G. Wei, Protocol-based unscented Kalman filtering in the presence of stochastic uncertainties, *IEEE Trans. Automat. Control* 65 (3) (2019) 1303–1309.
- [44] B. Shen, Z. Wang, D. Wang, J. Luo, H. Pu, Y. Peng, Finite-horizon filtering for a class of nonlinear time-delayed systems with an energy harvesting sensor, *Automatica* 100 (2) (2019) 144–152.
- [45] B. Shen, Z. Wang, D. Wang, Q. Li, State-saturated recursive filter design for stochastic time-varying nonlinear complex networks under deception attacks, *IEEE Trans. Neural Netw. Learn. Syst.* 31 (10) (2020) 3788–3800.
- [46] B. Shen, Z. Wang, D. Wang, H. Liu, Distributed state-saturated recursive filtering over sensor networks under round-robin protocol, *IEEE Trans. Cybern.* 50 (8) (2020) 3605–3615.
- [47] Y. Shen, Z. Wang, B. Shen, F.E. Alsaadi, F.E. Alsaadi, Fusion estimation for multi-rate linear repetitive processes under weighted Try-Once-Discard protocol, *Inf. Fusion* 55 (2020) 281–291.
- [48] D. Zhao, Z. Wang, Y. Chen, G. Wei, Proportional-integral observer design for multi-delayed sensor-saturated recurrent neural networks: A dynamic event-triggered protocol, *IEEE Trans. Cybern.* (2020) <http://dx.doi.org/10.1109/TCYB.2020.2969377>, in press.
- [49] D. Zhao, Z. Wang, D.W.C. Ho, G. Wei, Observer-based PID security control for discrete time-delay systems under cyber-attacks, *IEEE Trans. Syst. Man Cybern.-Syst.* (2019) <http://dx.doi.org/10.1109/TSMC.2019.2952539>, in press.
- [50] Z. Zhu, G. Zhang, M. Li, X. Liu, Evolutionary multi-objective workflow scheduling in cloud, *IEEE Trans. Parallel Distrib. Syst.* 27 (5) (2016) 1344–1357.
- [51] L. Zou, Z. Wang, J. Hu, D.H. Zhou, Moving horizon estimation with unknown inputs under dynamic quantization effects, *IEEE Trans. Automat. Control* 65 (12) (2020) 5368–5375.
- [52] L. Zou, Z. Wang, Q.-L. Han, D.H. Zhou, Moving horizon estimation of networked nonlinear systems with random access protocol, *IEEE Trans. Syst. Man Cybern.-Syst.* (2019) <http://dx.doi.org/10.1109/TSMC.2019.2918002>, in press.