IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 24, 2020, accepted September 2, 2020, date of publication September 10, 2020,
date of current version September 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3023071

A Novel Pure Pursuit Algorithm for Autonomous Vehicles Based on Salp Swarm Algorithm and Velocity Controller

RUI WANG[1,2], YING LI[2,4], JIAHAO FAN[2,4], TAN WANG[3], AND XUETAO CHEN[2,4]
[1]College of Software, Jilin University, Changchun 130012, China
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, Jilin University, Changchun 130012, China
[3]Space Technology (Jilin) Company Ltd., Jilin 132013, China
[4]College of Computer Science and Technology, Jilin University, Changchun 130012, China

Corresponding author: Jiahao Fan (jihanfan@hotmail.com)

This work was supported in part by the Department of Science and Technology of Jilin Province under Grant 20190303135SF, and in part by the Development and Reform Commission of Jilin Province under Grant 2019C053-13.

ABSTRACT Pure pursuit algorithm is one of the most effective ways of path tracking in autonomous vehicles. Nevertheless, the tracking accuracy of the existing pure pursuit algorithm is limited by the look-ahead distance. In this paper, to improve the tracking accuracy of the pure pursuit algorithm, a novel pure pursuit algorithm based on the optimized look-ahead distance named OLDPPA is proposed. Four improvements are presented in OLDPPA. Firstly, to find a better look-ahead distance of pure pursuit algorithm, salp swarm algorithm (SSA) is used in pure pursuit algorithm. Secondly, Brownian motion, a random motion mechanism of particles, is introduced in SSA to enhance its exploitation and exploration capabilities. Thirdly, to accelerate the convergence speed of SSA, a weighted mechanism which uses two different weights in the search process to adjust the salps closer to the food source quickly is assigned. Based on innovations 2 and 3, adaptive Brownian motion salp swarm algorithm (ABMSSA) is proposed and applied to pure pursuit algorithm. Finally, a velocity controller which outputs the speed of the next moment according to the distance and time interval between the look-ahead point and the current vehicle position is designed in OLDPPA, to ensure that the vehicle reaches its destination at a specified time. To verify the effectiveness and efficiency of OLDPPA, OLDPPA is applied in four different paths and the corresponding results are compared with other pure pursuit algorithms that use different look-ahead distances. Experimental results show that the tracking accuracy of OLDPPA is better than other algorithms.

INDEX TERMS Pure pursuit algorithm, autonomous vehicles, salp swarm algorithm, velocity controller, optimized look-ahead distance.

I. INTRODUCTION

With the development of artificial intelligence, autonomous vehicles [1] have received more and more attention [2]. In general, autonomous vehicles are classified into four parts: environment perception and localization [3], decision making [4], path planning [5], and path tracking [6]. Path tracking is the most important part of autonomous vehicles and the goal of path tracking is to control the vehicle to accurately follow the reference path given by path planning [7]. Pure pursuit algorithm is the most popular path tracking algorithm [8].

The associate editor coordinating the review of this manuscript and approving it for publication was Wentao Fan.

Besides pure pursuit algorithm, the path tracking algorithm mainly includes proportional-integral-derivative (PID) algorithm [9], linear quadratic regulator (LQR) algorithm [10], model predictive control (MPC) algorithm [11]. Compared with other path tracking algorithms, pure pursuit algorithm has a more simple implementation principle and better tracking results [12], [13].

Pure pursuit algorithm, a geometric control algorithm [14], determines the look-ahead point according to the look-ahead distance, then obtain the vehicle steering angle control input by the current heading angle and the distance between the current position and the look-ahead point position, so that the vehicle follows an arc trajectory passing through the look-ahead point. At present, a large number of researchers spend their energy on pure pursuit algorithm. Elbanhawi *et al.* [15] proposed a model predictive active yaw control implementation of pure pursuit path tracking to improve tracking performance at high speeds. Shan *et al.* [16] developed a new pursuit method, named CF-Pursuit, which used a clothoid C curve to replace the circle employed in pure pursuit algorithm to decrease tracking errors. However, the tracking performance of the existing pure pursuit algorithm is limited by the look-ahead distance. If the look-ahead distance is too large, it will cause the tracking path to be too smooth and the tracking accuracy will be greatly reduced. If the look-ahead distance is too small, the motion of the vehicle will be unstable. Many researchers adjust the look-ahead distance in a way that the look-ahead distance is directly proportional to the vehicle velocity [17], [18]. However, since the behavior of the vehicle does not change in proportion to the velocity, this method cannot completely solve the problem. Yu *et al.* [19] used the fuzzy rule controller to adjust the look-ahead distance. Serna *et al.* [20] estimated dynamically look-ahead distance based on the vehicle speed and lateral error. Wang *et al.* [21] improved look-ahead distance by a 2-degree polynomial function.

To sum up, although these methods can reduce various errors, they usually specify a fixed look-ahead distance manually through experiment or intuitive experience, which cannot find the best look-ahead distance. However, the best look-ahead distance not only can reduces the tracking error of pure pursuit algorithm but also can improves the stability of vehicle motion greatly. Thus, it is necessary to find the best look-ahead distance.

To solve the above problem, salp swarm algorithm (SSA), a kind of heuristic optimization algorithm that is relatively new and good at present, is used to find the best look-ahead distance. Heuristic algorithm is based on intuition experience or simulating some natural phenomena and mechanical structures to search for the best solution within the acceptable computational cost. In 1995, Kennedy and Eberhart [22] proposed Particle Swarm Optimization (PSO), which is inspired by the fact that a flock of birds always flies toward the area around the bird closest to the food during the foraging process. M. Dorigo *et al.* [23] proposed Ant Colony Optimization (ACO), inspired by the fact that ants can always crawl along the shortest path when looking for food. Yang [24] proposed firefly algorithm (FA) base on the phototaxis between fireflies. Gandomi *et al.* [25] proposed cuckoo search algorithm (CSA) based on the cuckoo's parasitic breeding behavior. Yang and Gandomi [26] proposed bat algorithm (BA) based on the ultrasonic characteristics of bats. SSA is chosen to optimize the look-ahead distance because it has a simpler implementation, lower parameters, and a smaller amount of calculation.

However, the convergence speed of SSA is slow, and it is easy to fall into the local optimum. Yang *et al.* [27] proposed memetic salp swarm algorithm which uses multiple independent salp chains to conduct exploration and development at the same time to search for high-quality optimal solutions quickly. Wang *et al.* [28] proposed an enhanced salp group algorithm based on the simplex method. This simplex method is a random mutation strategy which can expand various populations and improve the local search ability of the algorithm. Sayed *et al.* [29] proposed a novel chaotic salp swarm algorithm which uses ten different chaotic maps to accelerate the convergence speed and improve the accuracy of results. Although these methods have made different changes to accelerate the convergence speed and improve accuracy, they still cannot satisfy the balance of the convergence speed and accuracy in the path tracking of the autonomous vehicles.

Therefore, in order to meet the convergence speed and accuracy requirements of path tracking, and find a suitable solution in the shortest possible time, adaptive brownian motion salp swarm algorithm (ABMSSA) is proposed. Firstly, to improve the accuracy of SSA and meet accuracy requirements of path tracking, Brownian motion is introduced into the leader position update formula of SSA. Brownian motion is a random motion mechanism of particles and the step size of the particles movement is normally distributed. Brownian motion prevents SSA from falling into a local optimum by increasing the jumping ability of the salps. Therefore, the accuracy of SSA is greatly improved. Secondly, to accelerate the convergence speed of SSA and meet the convergence speed requirement of path tracking, two adaptive weights are added to the follower update formula of SSA. The follower salps in SSA follow the adjacent salps blindly, but the blindly following limits the search effect of SSA. To solve this problem, the two adaptive weights which are changed according to the fitness value of the current salp and its adjacent salp is used into SSA. One of the two weights affects the local optimization ability, and the other one affects the global optimization ability. In this way, the convergence speed of SSA is greatly accelerated by the joint adjustment of the global optimization ability and the local optimization ability.

To sum up, to find the optimal look-ahead distance of pure pursuit algorithm to improve the accuracy, a novel pure pursuit algorithm based on the optimized look-ahead distance named OLDPPA is proposed. Firstly, ABMSSA is applied to pure pursuit algorithm. The random look-ahead distance is input into ABMSSA as the initial individual. After many iterations, ABMSSA will output the best look-ahead distance by minimizing the fitness function which composed of path error, vehicle steering angle change speed, and acceleration change speed. In this way, pure pursuit algorithm will accurately choose the best look-ahead distance. The look-ahead distance determines the driving state of the vehicle. An unreasonable look-ahead distance increases the path error. The optimal look-ahead distance reduces the tracking error of the pure pursuit algorithm and improves the stability of vehicle motion. Secondly, a velocity controller is designed in OLDPPA. This controller output the vehicle speed of the next moment according to the distance and time interval between the look-ahead point and the current vehicle position. The real-time updated speed ensures that the vehicle reaches the target point within the required time. In summary, compared with the classic pure pursuit algorithm, OLDPPA not only improves the tracking accuracy but also enhances punctuality.

In the rest of the paper, the related work is introduced in section II which includes pure pursuit algorithm, salp swarm algorithm, brown motion. Then section III describes OLDPPA in detail. After that, section IV provides the experimental results about OLDPPA and the comparison with other related methods. Finally, a conclusion is illustrated in section V.

II. RELATED WORK

A. PURE PURSUIT ALGORITHM

Pure pursuit algorithm is a geometric path tracking method. In this paper, we do not pay attention to the internal dynamic model of the vehicle, but only focus on the overall velocity and heading changes of vehicles, so kinematics model [30] based on the vehicle motion is selected to update the position. Pure pursuit algorithm produces the steering angle required to bring the vehicle back to the reference path. Fig. 1 shows classical pure pursuit algorithm frame. Firstly, because the driver usually looks forward when driving, pure pursuit algorithm calculates and defines the look-ahead point on the reference path by the look-ahead distance and the current vehicle actual position [31]. Then, pure pursuit algorithm outputs the steering angle of the vehicle to follow the input look-ahead point. Finally, autonomous vehicle outputs the vehicle actual position and then continue to calculate the look-ahead point.

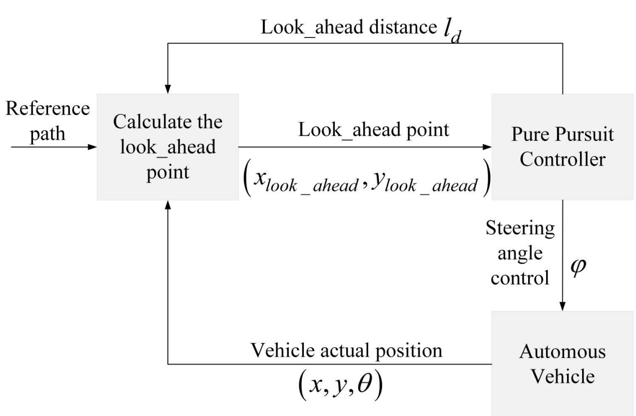

FIGURE 1. Classical pure pursuit algorithm frame.

Fig. 2 shows the schematic of pure pursuit algorithm and each variable is marked. To obtain the steering angle output ϕ, the look-ahead point and the vehicle (rear wheel) position are connected by a straight line. The angle between the line and the vehicle body is set as α, and α which is called look-ahead distance angle [32] is written as (1).

$$\alpha = \left|\theta - \arctan\left(\frac{y_{look\_ahead} - y}{x_{look\_ahead} - x}\right)\right| \tag{1}$$

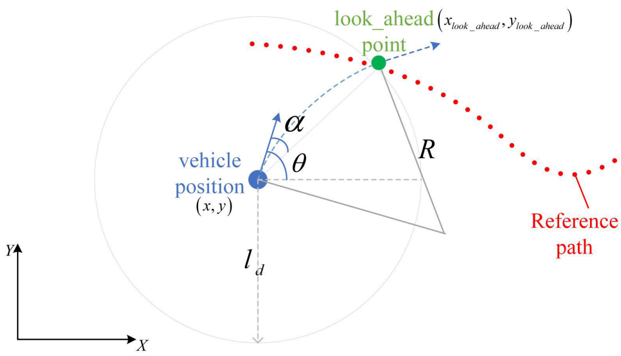

FIGURE 2. Schematic of the pure pursuit algorithm.

where x_{look_ahead} and y_{look_ahead} describe the position of the look-ahead point corresponding to vehicle position, x and y describe the position of the vehicle, θ is the heading of the vehicle. According to geometric knowledge, the radius of curvature R that the vehicle needs to follow is shown in (2).

$$R = \frac{l_d}{2 \sin \alpha} \tag{2}$$

where l_d represents the look-ahead distance followed by the vehicle. According to Ackerman geometric model [33] in Fig. 3, the vehicle steering angle ϕ is shown in (3).

$$\phi = \arctan\left(\frac{L}{R}\right) \tag{3}$$

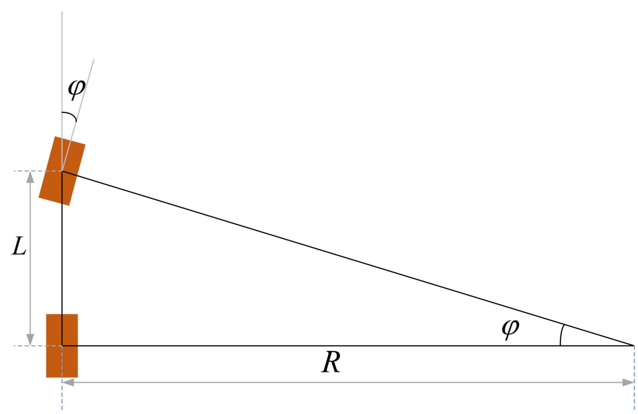

FIGURE 3. Ackerman geometric model.

where L is the wheelbase of the vehicle. By substituting (2) into (3), the vehicle steering angle is written as follows.

$$\phi = \arctan\left(\frac{2L \cdot \sin \alpha}{l_d}\right) \tag{4}$$

B. SALP SWARM ALGORITHM (SSA)

SSA is a kind of Meta-heuristic algorithm recently proposed by Mirjalili *et al.* [34]. According to what Mirjalili has mentioned [34], Salps belong to the family of Salpidae and have a transparent barrel-shaped body. Their tissues and movement are very similar to jellyfish. In deep oceans, salps often form clusters called the salp chain. This swarming behavior helps salps to quickly coordinate changes and find more food. SSA has the advantages of fewer parameters and easy implementation.

SSA divides the population into two groups: leader and followers. The leader is the salp at the front of the chain, and the rest are considered as followers. Leader guides swarm and followers follow the leader [34]. The position of the salps is defined in an n-dimensional search space, where n is the number of variables for a given problem. The positions of all salps are stored in a two-dimensional matrix called x, and the food source which is the population target set to F.

The position of the leader salp is updated as follows.

$$x_j^1 = \begin{cases} F_j + c_1\left(\left(ub_j - lb_j\right) c_2 + lb_j\right), & c_3 \geq 0.5 \\ F_j - c_1\left(\left(ub_j - lb_j\right) c_2 + lb_j\right), & c_3 \leq 0.5 \end{cases} \tag{5}$$

where x_j^1 and F_j represent the positions of leader salp and the source of the food in the jth dimension, ub_j and lb_j are the upper and lower bounds of the jth dimension, respectively. c_2 and c_3 are the random numbers in the range of [0,1]. c_1 is the most important parameter for balancing between the exploitation and exploration and is mathematically defined as follows.

$$c_1 = 2e^{-\left(\frac{4t}{T}\right)^2} \tag{6}$$

where t is the current iteration and T is the maximum number of iterations. The position of the follower salps is updated as shown in (7).

$$x_j^i = \frac{1}{2}\left(x_j^i + x_j^{i-1}\right) \tag{7}$$

where $i \geq 2$ and x_j^i denotes the position vector of the ith follower salp at the jth dimension.

C. BROWNIAN MOTION

Brownian motion [35], a random motion mechanism of tiny particles in liquids, was proposed by British botanist Robert Brown in 1827. In 1918, Wiener defined Brownian motion and a proof of existence. Brownian motion is also called the Wiener process. The study of Brownian motion marks the beginning of the study of stochastic differential equations. The Wiener process shows that the increment of Brownian motion conforms to the normal distribution. For one-dimensional Brownian motion, let W_t be the Brownian real-valued process on the probability space, then ΔW_t which is the increment of W_t is shown in (8) according to what Novikov has mentioned [35].

$$\Delta W_t \sim N(0, h), \quad \forall t > 0,\ h > 0 \tag{8}$$

where h is the step size.

III. METHODOLOGY

OLDPPA has two key steps: using ABMSSA to optimize the look-ahead distance and controlling the vehicle speed in real time by the velocity controller.

A. ABMSSA

ABMSSA, which is based on SSA, uses Brownian motion and an adaptive weight mechanism to improve the exploitation and exploration capabilities and accelerate the convergence speed of SSA. The flowchart of ABMSSA is presented in Fig. 4.

1) BROWNIAN MOTION

Brownian motion is introduced into the leader position update mechanism of SSA to enhance the exploitation and exploration capabilities. In this way, the leader position update formula of SSA is changed from (5) to (9).

$$x_j^1 = \begin{cases} F_j + c_1\left(\left(ub_j - lb_j\right) c_2 + lb_j\right) \cdot W_t, & c_3 \geq 0.5 \\ F_j - c_1\left(\left(ub_j - lb_j\right) c_2 + lb_j\right) \cdot W_t, & c_3 \leq 0.5 \end{cases} \tag{9}$$

This method not only increases the jumping ability of salps to prevent falling into a local optimum but also improves the diversity of SSA.

2) ADAPTIVE WEIGHT

During the search process of SSA, the region of the optimal solution is expected to be determined at an early stage. To this end, two different adaptive weights are designed to accelerate the convergence speed. The update formulas for the two adaptive weights are as follows.

$$w_1 = \frac{2f^2\left(x_j^i\right)}{f^2\left(x_j^{i-1}\right) + f^2\left(x_j^i\right)} \tag{10}$$

$$w_2 = \frac{f^2\left(x_j^{i-1}\right)}{f^2\left(x_j^{i-1}\right) + f^2\left(x_j^i\right)} \tag{11}$$

where $f\left(x_j^i\right)$ and $f\left(x_j^{i-1}\right)$ are the fitness values of the ith follower salp at the jth dimension and the adjacent salp followed by it. And then, in this paper, the follower position update formula of SSA is changed from (7) to (12).

$$x_j^i = \frac{1}{2} w_1 \cdot \left(x_j^i + x_j^{i-1}\right) + w_2 \cdot \left(F_j - x_j^i\right) \tag{12}$$

where F_j is source of the food in the jth dimension. w_1 affects the local optimization ability of SSA, and w_2 affects the global optimization ability. In this way, the joint adjustment of the global optimization ability and the local optimization ability make the salps access to the food source quickly. Therefore, the region of the optimal solution is determined at an early stage and the convergence speed is significantly accelerated.

B. VELOCITY CONTROLLER

Autonomous vehicles need to reach the destination at a specified time in many cases, such as changing lanes, over-taking, etc. Classical pure pursuit algorithm only provides the vehicle with a real-time control input of the steering angle, and does not provide the real-time control input of the vehicle speed.

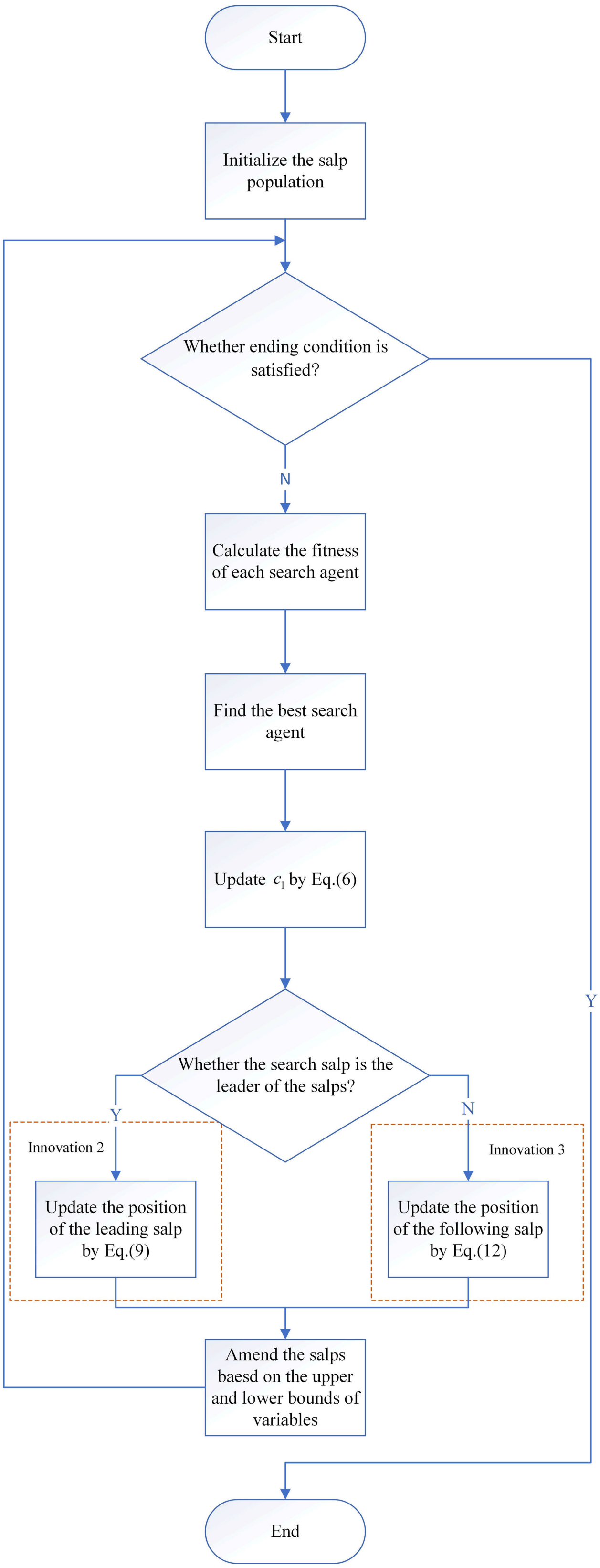

FIGURE 4. The flowchart of the ABMSSA.

This means that classical pure pursuit algorithm cannot meet the punctuality demand of path tracking in these cases. To this end, a velocity controller is proposed, which outputs the speed of the next moment according to the distance and time interval between the look-ahead point and the current vehicle position.

Fig. 5 shows the tracking details and parameters with the velocity controller. Where t_i is the ith time point, x_i and y_i describe the vehicle position at the ith time point, the first green point in Fig. 5 is set as the look-ahead point at the ith time point. This look-ahead point is also the reference path point at the nth time point t_n, then the time interval ΔT is shown in (13).

$$\Delta T = t_n - t_i \tag{13}$$

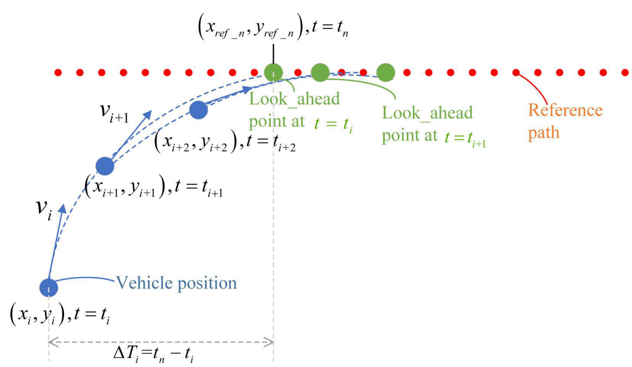

FIGURE 5. Tracking details and parameters of the velocity controller.

In fact, the steering angle control input from pure pursuit algorithm causes the vehicle to follow an arc trajectory, so it is more reasonable to replace the straight line distance with the arc length between the look-ahead point and the vehicle position. According to geometric knowledge, s_i, the length of the arc trajectory that the vehicle plans to pass through the look-ahead point at the ith time point is shown in (14).

$$s_i = 2\alpha_i R_i \tag{14}$$

where α_i is the look-ahead distance angle at the ith time point, which is obtained by (1). R_i is the radius of curvature at the ith time point, which is obtained by (2). Then v_i which is the velocity control output of the velocity controller at the ith time point is defined as (15).

$$v_i = \frac{s_i}{\Delta T} \tag{15}$$

By substituting (2) and (14) into (15), v_i is written as (16).

$$v_i = \frac{\alpha_i l_d}{\Delta T \sin \alpha_i} \tag{16}$$

The design of this velocity controller greatly improves the punctuality of path tracking with pure pursuit algorithm, and also reduces the path tracking error.

C. OLDPPA

The control framework of OLDPPA is shown in Fig. 6. First, in OLDPPA, ABMSSA is used to obtain the optimized look-ahead distance by minimizing the fitness function given by (17). Then the look-ahead point is selected by the look-ahead distance and the vehicle current position. After that, according to the look-ahead point, pure pursuit algorithm outputs the steering angle control input. Meanwhile, the velocity controller output the velocity control input of the controlled vehicle. The steering angle control input and the velocity control input control the vehicle to follow this look-ahead point. Next, the vehicle actual position is output. The next look-ahead point is selected by the vehicle position and the optimized look-ahead distance. OLDPPA will always work until the vehicle tracks the end of the reference path.

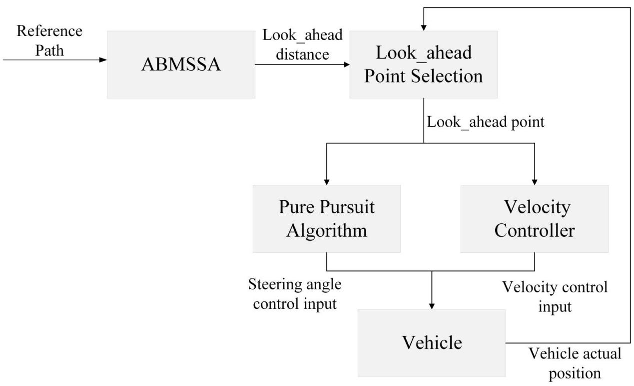

FIGURE 6. OLDPPA control frame.

The fitness function of ABMSSA to optimize the look-ahead distance is as follows.

$$J = w_1 \sum_{i=0}^{n-1} \sqrt{\left(y_i - y_{ref\_i}\right)^2 + \left(x_i - x_{ref\_i}\right)^2} + w_2 \sum_{i=1}^{n-1} \frac{(\theta_i - \theta_{i-1})}{\Delta t} + w_3 \sum_{i=1}^{n-1} (a_i - a_{i-1}) \quad (17)$$

where x_i and y_i describe the vehicle position at the ith $(1 \le i \le k)$ time point, x_{ref_i} and y_{ref_i} describe the reference path position at the ith $(1 \le i \le k)$ time point, θ_i is the heading angle at the ith $(1 \le i \le k)$ time point, a_i is the acceleration value at the ith time point, and k is the number of time points. w_1, w_2, and w_3 are constants. In this paper, w_1, w_2, w_3 are set as 0.5, 0.3, and 0.2. This fitness function consists of three parts: path error, the heading angle variation, and the acceleration variation. The first part, that is, the error between the vehicle position and the reference path point, represents the accuracy of the path tracking. The second part and the third part represent safety and comfort when the vehicle is following the path.

D. THE OVERALL PROCESS OF OLDPPA

The steps of OLDPPA are shown in Fig. 7. First, the reference path and the initial position of the vehicle need to be determined. Second, ABMSSA is used, and the optimal look-ahead distance is obtained by minimizing the fitness function given by (17). Third, the look-ahead point is selected by the optimal look-ahead distance. Fourth, according to the pure pursuit algorithm and the velocity controller which is designed in this paper, the vehicle's steering angle and velocity control input are respectively output. Fifth, the vehicle outputs its own actual position. Sixth, determine whether the current time of the vehicle is less than or equal to the time corresponding to the end of the reference path and if so, return to the third step, otherwise,

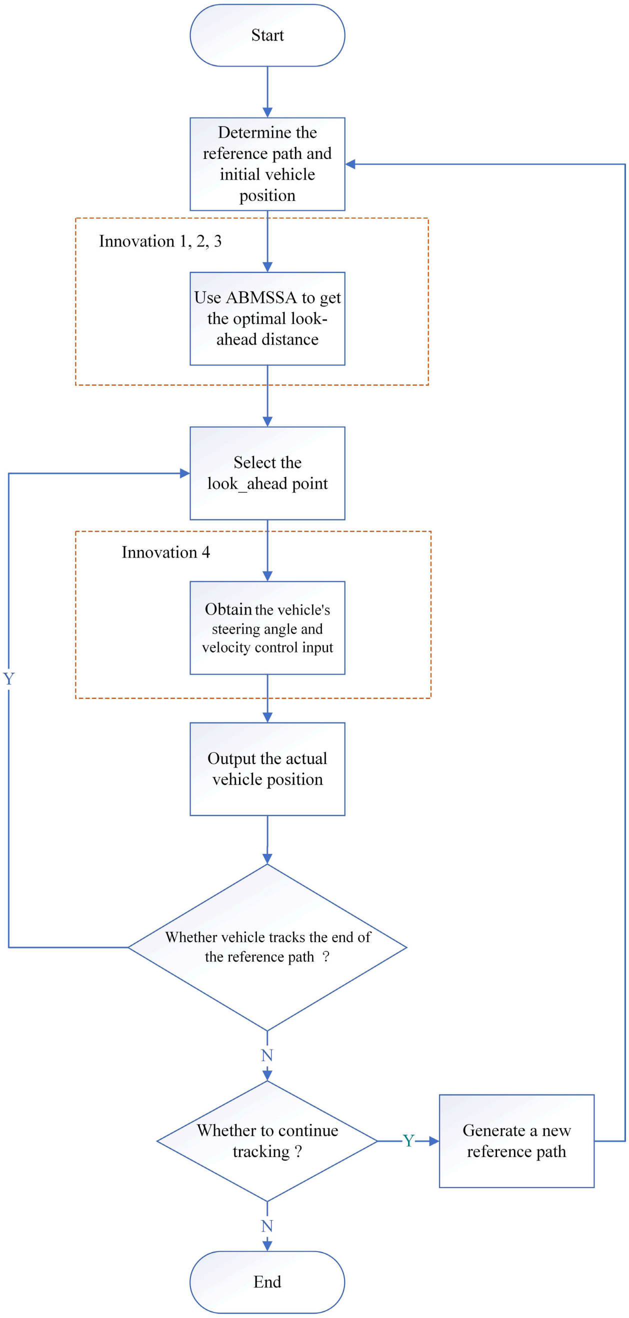

FIGURE 7. Flowchart of OLDPPA.

TABLE 1. Benchmark functions.

Function	Dim	Range	Min
$F_1(x)=\sum_{i=1}^{n} x_i^2$	30	[-100,100]	0
$F_2(x)=\sum_{i=1}^{n} \lvert x_i \rvert+\prod_{i=1}^{n} x_i$	30	[-10,10]	0
$F_3(x)=\sum_{i=1}^{n}\left(\sum_{j-1}^{i} x_j\right)^2$	30	[-100,100]	0
$F_4(x)=\max_i \lvert x_i \rvert, 1\le i\le n$	30	[-100,100]	0
$F_5(x)=\sum_{i=1}^{n-1}\left[100\left(x_{i+1}+x_i^2\right)^2+(x_i-1)^2\right]$	30	[-30,30]	0
$F_6(x)=\sum_{i=1}^{n}([x_i+0.5])^2$	30	[-100,100]	0
$F_7(x)=\sum_{i=1}^{n} ix_i^4+random[0,1)$	30	[-1.28,1.28]	0
$F_8(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)+10\right]$	30	[-5.12,5.12]	0
$F_9(x)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i{}^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi i)\right)+20+e$	30	[-32,32]	0
$F_{10}(x)=\frac{1}{4000}\sum_{i=1}^{n} x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30	[-600,600]	0
$F_{11}(x)=\frac{\pi}{n}\left\{10\sin(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2\left[1+10\sin^2(\pi y_{i+1})\right]+(y_n-1)^2\right\}+\sum_{i=1}^{n} u(x_i,10,100,4)$ $y_i=1+\frac{x_i+1}{4}$ $u(x_i,a,k,m)=\begin{cases} k(x_i-a)^m\ x_i>a \\ 0-a<x_i<a \\ k(-x_i-a)^m\ x_i<-a \end{cases}$	30	[-50,50]	0
$F_{12}(x)=0.1\left\{\sin^2(3\pi x_1)+\sum_{i=1}^{n}(x_i-1)^2\left[1+\sin^2(3\pi x_i+1)\right]+(x_n-1)^2\left[1+\sin^2(2\pi x_n)\right]\right\}+\sum_{i=1}^{n} u(x_i,5100,4)$	30	[-50,50]	0

return to the first step after generating a new reference path.

In OLDPPA, an improved algorithm ABMSSA based on SSA is proposed to optimize the look-ahead distance and a new velocity controller is designed to improve the punctuality of vehicle path tracking.

IV. EXPERIMENTS AND ANALYSIS

A. BENCHMARK FUNCTIONS

In this section, the presented ABMSSA algorithm is verified with twelve benchmark functions [36] which have been shown in Table 1. The ABMSSA algorithm is compared with SSA [34], GWO [37], WOA [38], ESSA [39] and MSSA [40] by solving the twelve benchmark functions, as Table 2 shows. To get a fair comparison, the population number is set as 30, the total iterations are set as 500, and the executed runs are set as 30 times to compute the average and standard deviation of best solutions as shown in Tables 3. All of the codes are implemented in MATLAB R2016a and run on a Window 10 PC with Intel Core i5-4200M 2.50 GHz CPU and 8 GB RAM.

TABLE 2. Other algorithm for comparison.

Algorithm	Parameters setting	Reference
SSA	$c_2:0\sim1, c_3:0\sim1$	[34]
GWO	$r_1:0\sim1, r_2:0\sim1$	[37]
WOA	$r_1:0\sim1, r_2:0\sim1, b=1, p:0\sim1$	[38]
ESSA	$c_2:0\sim1, c_3:0\sim1, r:0\sim1$	[39]
MSSA	$c_2:0\sim1, c_3:0\sim1$	[40]

TABLE 3. The results of algorithms for optimizing benchmark function.

Functions		ABMSSA	SSA	GWO	WOA	ESSA	MSSA
F1	Mean	**0**	1.25E-07	9.58E-28	2.66E-74	8.90E-15	1.41E-07
	Std	**0**	7.79E-15	1.30E-54	5.30E-147	3.01E-28	3.85E-14
F2	Mean	**0**	0.01229	4.18E-33	2.00E-52	1.42E-08	4.01E-05
	Std	**0**	0.00435	3.29E-65	7.13E-103	4.51E-16	2.66E-08
F3	Mean	**0**	5.27E-07	9.53E-24	1.35E+02	2.12E-14	3.26E-06
	Std	**0**	1.28E-12	2.33E-45	7.54E+04	1.71E-17	2.83E-10
F4	Mean	**0**	2.28E-05	4.79E-18	5.37176	3.47E-08	2.21E-05
	Std	**0**	1.24E-10	8.19E-35	93.18333	2.24E-15	3.19E-11
F5	Mean	8.19958	1.35E+02	**6.47118**	6.92794	7.82142	9.77371
	Std	**0.03938**	1.34E+05	0.39376	0.37560	0.04202	1.42E+02
F6	Mean	**6.49E-11**	9.14E-10	3.14E-06	0.01017	1.06E-09	1.20E-09
	Std	**9.66E-21**	9.32E-20	1.57E-12	0.00199	8.86E-20	1.88E-19
F7	Mean	6.04E-04	0.01460	6.15E-04	0.00256	**8.82E-05**	0.00339
	Std	1.19E-06	1.21E-04	1.83E-07	7.77E-06	**5.45E-09**	1.01E-05
F8	Mean	**0**	18.4067	0.56078	0.98723	**0**	0.53064
	Std	**0**	59.31363	1.80716	21.01374	**0**	2.42426
F9	Mean	**8.88E-16**	1.03414	7.88E-15	3.61E-15	1.22E-08	7.88E-04
	Std	**0**	0.81698	3.77E-30	4.78E-30	1.49E-16	1.74E-05
F10	Mean	**0**	0.23037	0.26335	0.07244	2.40E-14	0.12176
	Std	**0**	0.01220	5.69E-04	0.01637	5.13E-27	0.01250
F11	Mean	**3.83E-13**	0.64695	0.00306	0.15398	5.19E-11	0.04592
	Std	**2.10E-25**	0.51588	7.16E-05	0.32930	2.41E-21	0.00966
F12	Mean	**0.00220**	0.00290	0.01064	0.02888	0.00360	0.00476
	Std	**1.93E-05**	3.05E-05	9.98E-04	0.00158	4.07E-05	2.96E-05

In Table 3, the best results are highlighted in bold. Meanwhile, the convergence curves of the six algorithms are shown in Fig. 8. According to Table 3, in addition to the benchmark functions F5 and F7, it can be seen that the average and

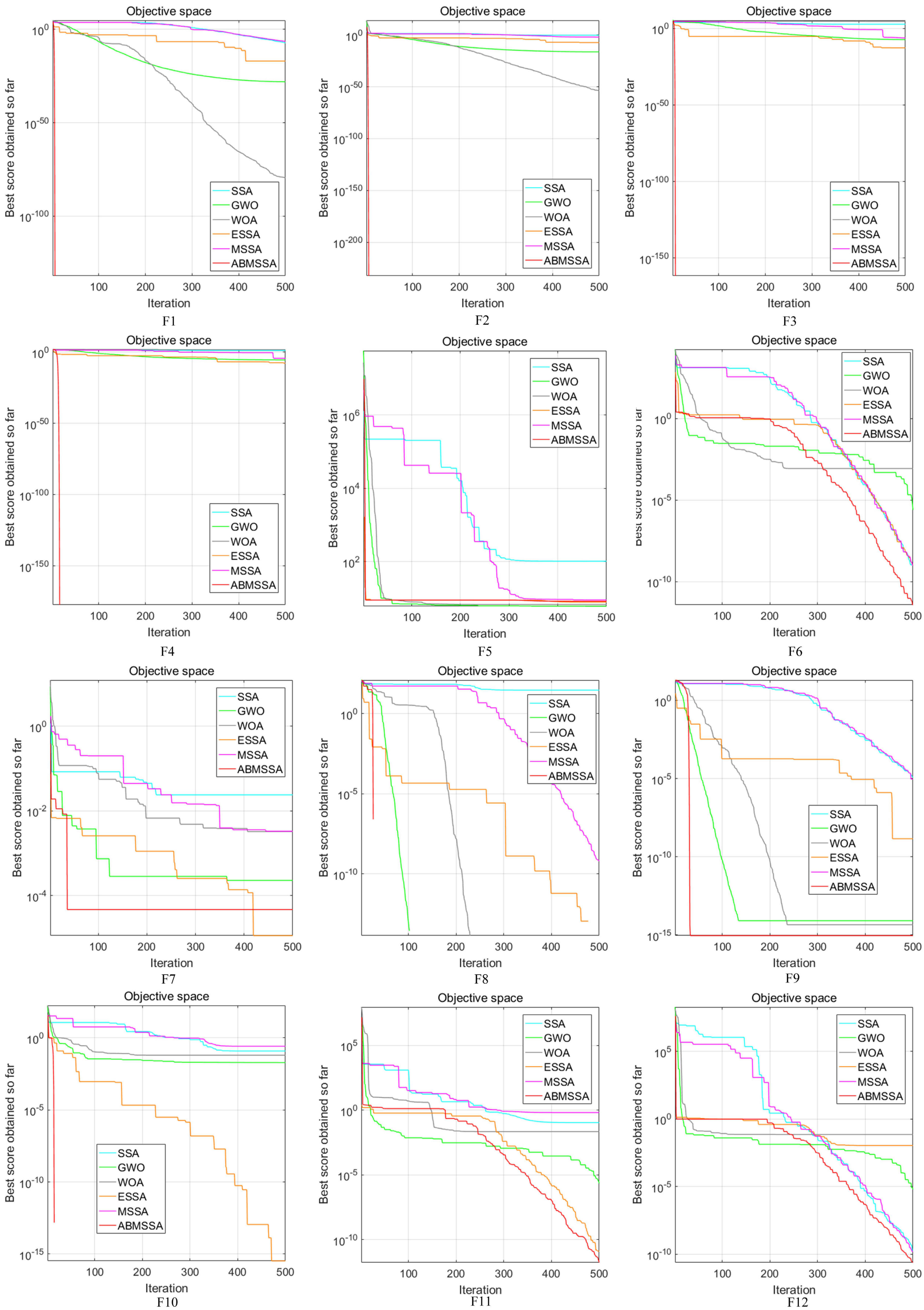

FIGURE 8. The convergence curves of algorithms for optimizing benchmark function.

standard deviation of ABMSSA obtained by solving other benchmark functions are better than the other five algorithms. A better average indicates that the average performance of ABMSSA is better than the other five algorithms, and a smaller standard deviation indicates that ABMSSA is more stable. Although the average obtained by ABMSSA is not the best in solving the benchmark function F5, the standard deviation of ABMSSA is the best. Thus ABMSSA has great advantages when the stability of the optimization algorithm is required. Although the average and standard deviation obtained by ABMSSA are not the best in solving the benchmark function F7, it can be seen from Fig. 8 that the convergence speed of ABMSSA is faster than other algorithms, so when the convergence speed of the optimization algorithm is required, ABMSSA has great advantages. Observing the convergence curve in Figure 8, it can be found that ABMSSA achieves good results in the early convergence stage, and the convergence speed is faster. In summary, the performance of ABMSSA is higher than the other five algorithms, which not only can obtain better optimal values, but also has a faster convergence speed.

B. SIMULATION AND PERFORMANCE EVALUATION OF OLDPPA

To verify the effectiveness of the proposed OLDPPA controller, CarSim-Matlab/simulink co-simulation and analysis are performed. Four different paths are constructed in this section: a straight path, a sinusoidal path, a parabolic path, and a lane changing path. These four paths represent the common driving states of vehicles, including going straight, turning continuously, avoiding obstacles, and changing lanes. It is reasonable that the four paths are selected to demonstrate the effectiveness of OLDPPA. On these four paths, the performance comparison between OLDPPA controller and classic pure pursuit controller is introduced. In OLDPPA, the look-ahead distance of the pure pursuit controller is optimized and a real-time velocity controller is proposed. To highlight the comparison results, we have compared OLDPPA with two classical pure pursuit controllers which have different look-ahead distances, and the look-ahead distance of one classic pure pursuit controller is set to 4.0m (CPP-4m) and the other is set to 8.0m (CPP-8m). In this experiment, the parameters of the simulation vehicle is shown in Table 4.

TABLE 4. **The parameters of the simulation vehicle.**

Parameter	Symbol	Value	Units
Vehicle wheelbase	L	2.9	m
Time interval	Δt	0.5	s
Initial position	(x_0, y_0)	$(0, -5)$	m
Initial heading	θ_0	0	rad

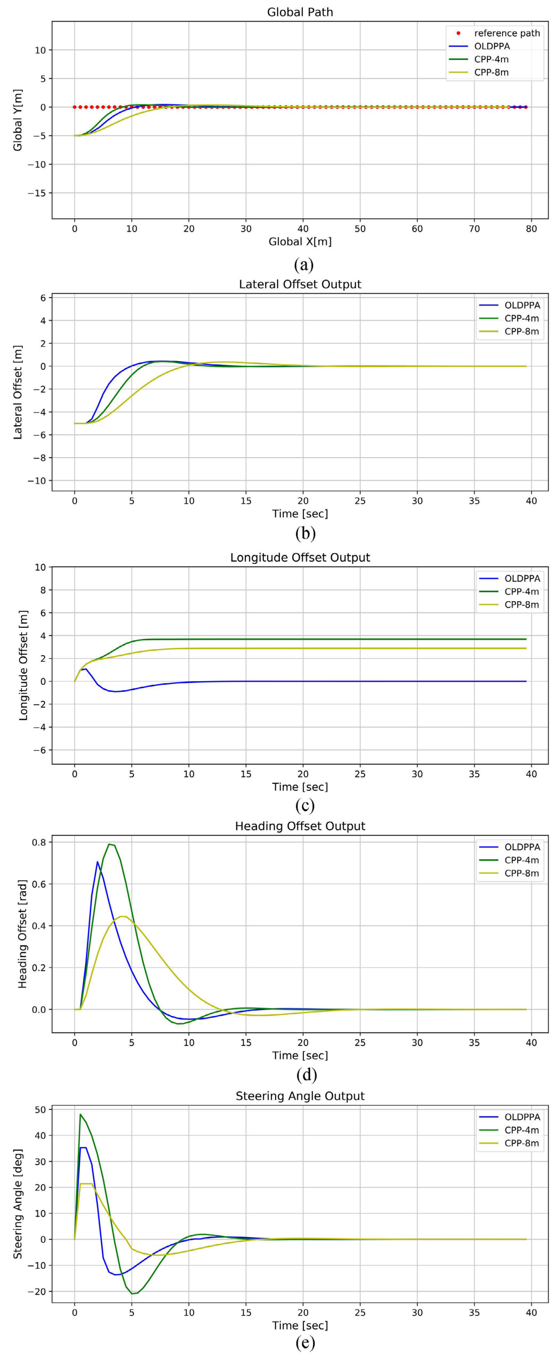

FIGURE 9. **Tracking performance experiment results with OLDPPA, CPP-4m and CPP-8m for the straight path. (a) Global path, (b) Lateral offset, (c)Longitude offset, (d) Heading offset, (e) Steering wheel angle.**

1) STRAIGHT PATH

We choose 80 path points to form a straight path. The velocity of classic pure pursuit controller is set to 2.0m/s. Fig. 9 compares the tracking performance of the three controllers based on the experimental results of tracking path, lateral offset, longitudinal offset, heading offset and steering angle on a straight path.

It can be seen from Fig. 9a that the tracking paths of the three controllers are not much different. But from Fig. 9b,

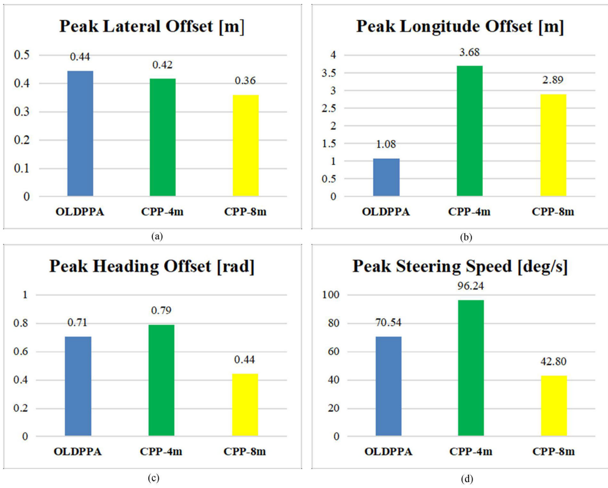

FIGURE 10. **Comparison of experiment results with OLDPPA, CPP-4m and CPP-8m for the straight path. (a) Peak lateral offset, (b) Peak longitude offset, (c) Peak heading offset, (d) Peak steering speed.**

it can be found that the lateral offset of OLDPPA is reduced to 0 more quickly which is obtained by the reference path point and the vehicle position according to time. From Fig. 9c, it can be seen that the longitude offset of OLDPPA has always been smaller than CPP-4m and CPP-8m and has been close 0 which is also obtained by the reference path point and the vehicle position according to time. It is illustrated that OLDPPA makes path tracking more accurate and more timely according to Fig. 9b and Fig. 9c. Looking at Fig. 9d, although the peak heading offset of CPP-8m is smaller than OLDPPA, OLDPPA reduces the heading error to 0 faster. Fig. 9e represents the stability of the vehicle during path tracking, although CPP-8m makes the vehicle more stable, under the reasonable steering angle, OLDPPA returns to the reference path faster. In summary, compared to the other two controllers on the straight path, OLDPPA has higher accuracy and better punctuality.

To quantitatively evaluate the performance of the three controllers, the peak of the lateral offset, the peak longitudinal offset, the peak heading offset, and the peak steering speed on this straight path are compared, as shown in Fig. 10. The peak lateral offset of the tracking path of the three controllers is not much different, but the peak longitude offset of OLDPPA is about 2.5m smaller than CPP-4m and 1.8m smaller than CPP-8m. The peak heading offset of OLDPPA is close to that of CPP-4m, which is about 0.1rad smaller than that of CPP-4m. The peak steering speed of OLDPPA is about 25 deg/s smaller than that of CPP-4m. Although CPP-8m has a better performance in the peak heading offset and the peak steering speed, CPP-8m cannot return to the reference path quickly, and the path tracking accuracy is much lower than OLDPPA.

2) SINUSOIDAL PATH

We choose 80 path points to form a sinusoidal path. We set the velocity of classic pure pursuit controllers to 5.0m/s.

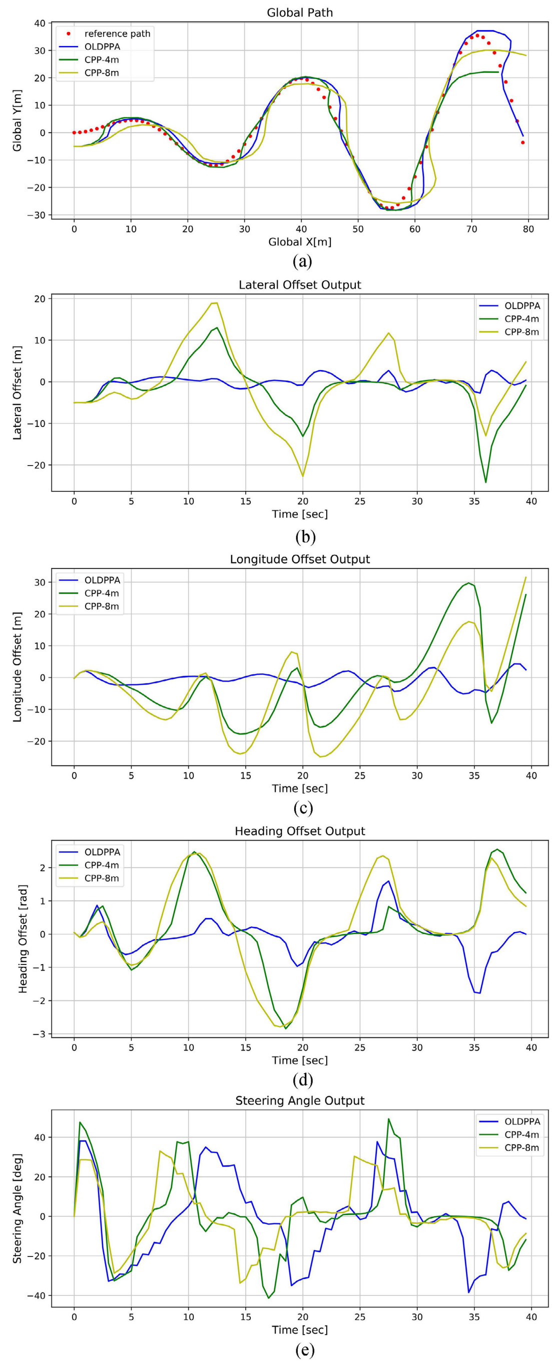

FIGURE 11. **Tracking performance experiment results with OLDPPA, CPP-4m, and CPP-8m for the sinusoidal path. (a) Global path, (b) Lateral offset, (c) Longitude offset, (d) Heading offset, (e) Steering wheel angle.**

Fig. 11 compares the tracking performance of the three controllers based on the experimental results of tracking path, lateral offset, longitudinal offset, heading offset and steering angle on this sinusoidal path.

It can be seen from Fig. 11a that the tracking path of OLDPPA fits the reference path best. From Fig. 11b and Fig. 11c, it can be seen that OLDPPA performs best, and the lateral and longitude offsets are always close to 0.

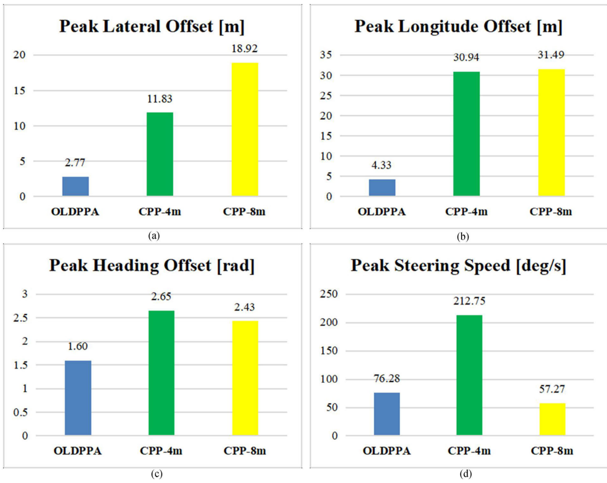

FIGURE 12. Comparison of experiment results with OLDPPA, CPP-4m, and CPP-8m for the sinusoidal path. (a) Peak lateral offset, (b) Peak longitude offset, (c) Peak heading offset, (d) Peak steering speed.

It means that OLDPPA tracks very accurately on this sinusoidal path. Observing Fig. 11d, the heading offset fluctuation of OLDPPA has always been less than CPP-4m and CPP-8m and has been floating within a small range of 0. From Fig. 11e, it can be found that the steering angle of OLDPPA is more stable than CPP-4m and CPP-8m, and CPP-4m is so instability that it has even caused a shocking phenomenon. To summarize, compared with the other two controllers on this sinusoidal path, OLDPPA has higher accuracy, better punctuality, and better stability.

Similarly, quantitative comparison and evaluation results of the three controllers for this sinusoidal path are shown in Fig. 12. OLDPPA has the best performance in the peak lateral offset, peak longitude offset, and peak heading offset. In the peak lateral offset, OLDPPA is about 9.1m smaller than CPP-4m and about 16.2m smaller than CPP-8m. In the peak longitude offset, OLDPPA is about 26.6m smaller than CPP-4m and about 27m smaller than CPP-8m. The peak heading offset peak is about 1.0rad smaller than CPP-4m and about 0.5rad smaller than CPP-8m. In the peak steering speed, OLDPPA is about 136.5deg/s less than CPP-4m. Although the peak steering speed of OLDPPA is about 20deg/s than CPP-8m, its steering speed is reasonable and has an excellent lateral offset, longitudinal offset and heading offset. This indicates that the path tracking accuracy of OLDPPA is much higher than CPP-8m.

3) PARABOLIC PATH

We choose 80 path points to form a parabolic path which is seen as the path when the vehicle avoids obstacles. We set the speed of two classic pure pursuit controllers to 2.0m/s. The experiment results of the three controllers are compared in Fig. 13.

It can be seen from Fig. 13a that the tracking path of OLDPPA fits the reference path best, and CPP-4m and CPP-8m do not reach the end of the reference path within a specific time. From Fig. 13b and Fig. 13c, the lateral and longitude offset of OLDPPA are smaller than CPP-4m and CPP-8m and are always close to 0, which shows that OLDPPA is also very accurate when tracking on this parabolic path. From Fig. 13d, the heading offset fluctuation has always been less than CPP-4m and CPP-8m and has been floating within a small range of 0. From Fig. 13e, it can be found that the steering angle of OLDPPA is more stable than that of CPP-4m, and a little bigger than CPP-8m, it is

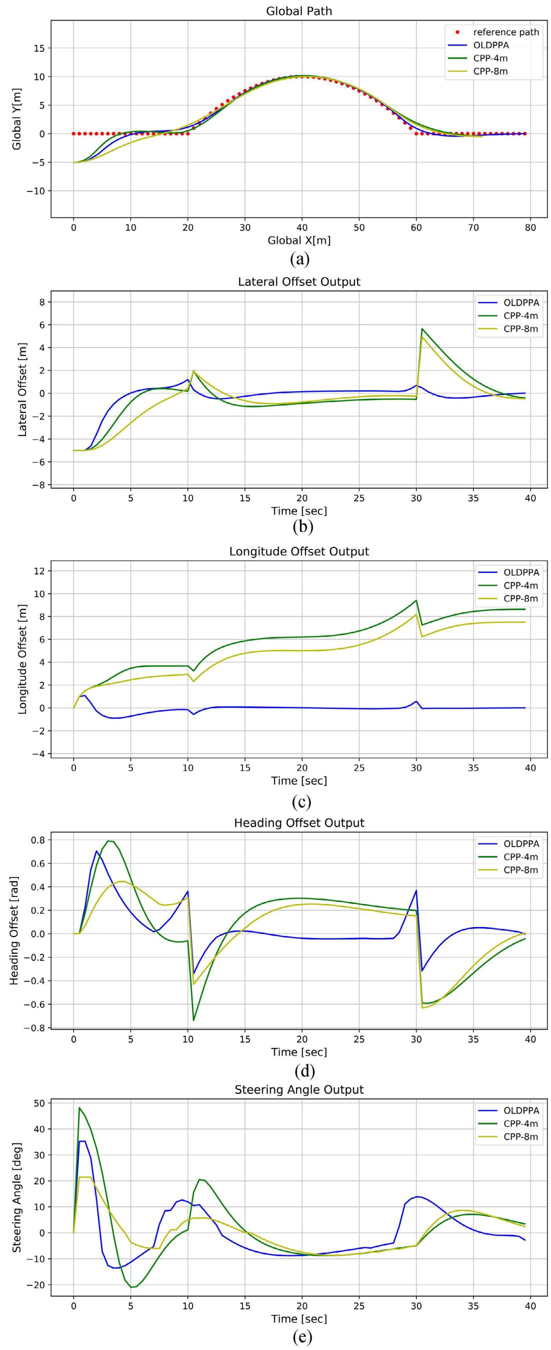

FIGURE 13. Tracking performance experiment results with OLDPPA, CPP-4m and CPP-8m for the parabolic path. (a) Global path, (b) Lateral offset, (c)Longitude offset, (d) Heading offset, (e) Steering wheel angle.

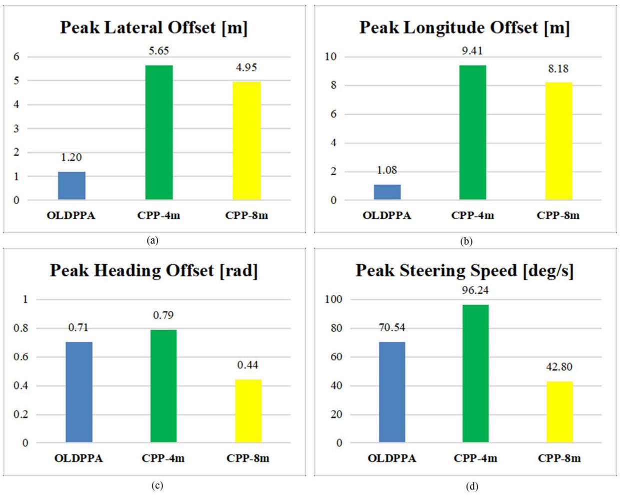

FIGURE 14. **Comparison of experiment results with OLDPPA, CPP-4m, and CPP-8m for the parabolic path. (a) Peak lateral offset, (b) Peak longitude offset, (c) Peak heading offset, (d) Peak steering speed.**

because that the tracking path of CPP-8m is relatively smooth. However CPP-8m is unable to track reference path points at the corner of the path, so the tracking effect is not as good as OLDPPA. In summary, compared to the other two controllers on this parabolic path, OLDPPA has higher accuracy and better punctuality.

Fig. 14 shows the quantitative comparison results of the three controllers for this parabolic path. In the peak lateral offset, OLDPPA is about 4.4m smaller than CPP-4m and about 3.8m smaller than CPP-8m. In the peak longitude offset, OLDPPA is about 8.4m smaller than CPP-4m and about 7.1m smaller than CPP-8m. The peak heading offset of OLDPPA is about 0.1rad smaller than that of CPP-4m, and the peak steering speed of OLDPPA steering angular velocity is about 25.7 deg/s smaller than that of CPP-4m. Although the peak heading offset and the peak steering speed offset of OLDPPA are smaller than OLDPPA, CPP-8m's tracking accuracy is indeed much lower than OLDPPA.

4) LANE CHANGING PATH

We choose 80 path points to form a lane changing path. The speeds of both classical pure pursuit controllers are set to 2.0m/s. The experiment results of the three controllers are compared in Fig. 15.

It can be seen from Fig. 15a that the tracking effect of OLDPPA is the best. Many reference path points of CPP-4m and CPP-8m have not even been tracked, and it is obvious that CPP-4m and CPP-8m have not reached the end of the reference path within a specific time. From Fig. 15b and Fig. 15c, the lateral and longitude offsets of OLDPPA are significantly smaller than CPP-4m and CPP-8m and are always close to 0, which indicates that OLDPPA has highly tracking accuracy on this lane change path. In Fig. 15d, the heading error of OLDPPA is also smaller than CPP-4m and CPP-8m, and it always floats within a small range of 0. Fig. 15e, it can be found that the steering angle change of OLDPPA is

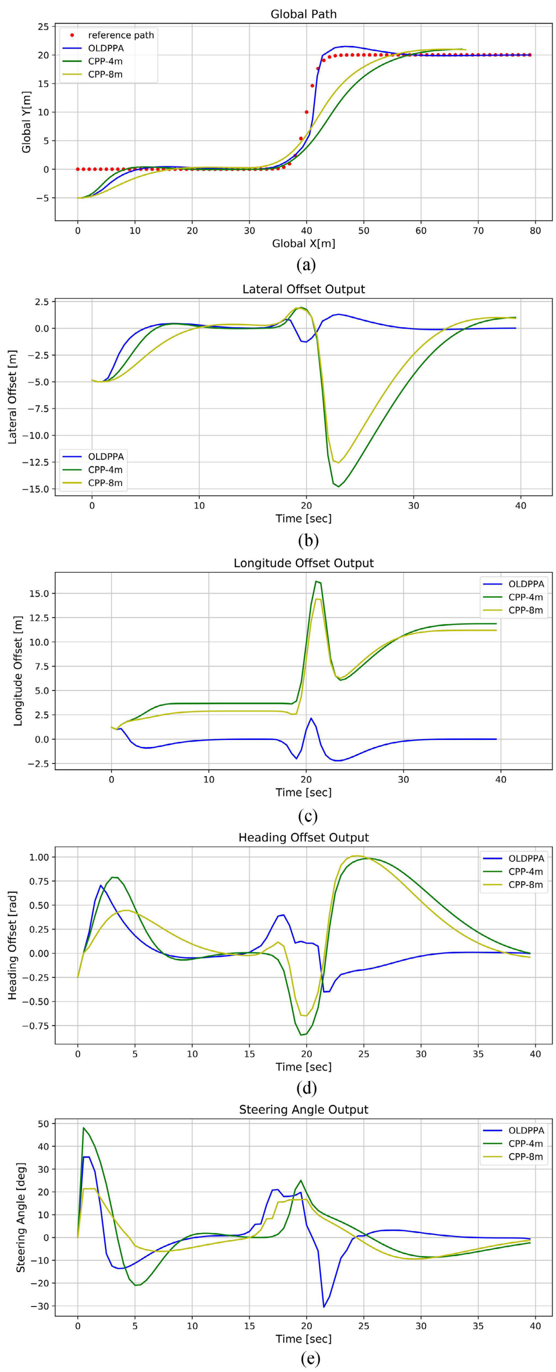

FIGURE 15. **Tracking performance experiment results with OLDPPA, CPP-4m and CPP-8m for the lane changing path. (a) Global path, (b) Lateral offset, (c) Longitude offset, (d) Heading offset, (e) Steering wheel angle.**

more stable than CPP-4m. Because CPP-8m's tracking path is relatively smooth, CPP-8m is unable to track reference path points at the corner of the path, so even though CPP-8m's steering angle change is more stable than OLDPPA, the tracking performance is far worse than OLDPPA. To summarize, OLDPPA has a better tracking effect on this lane changing path.

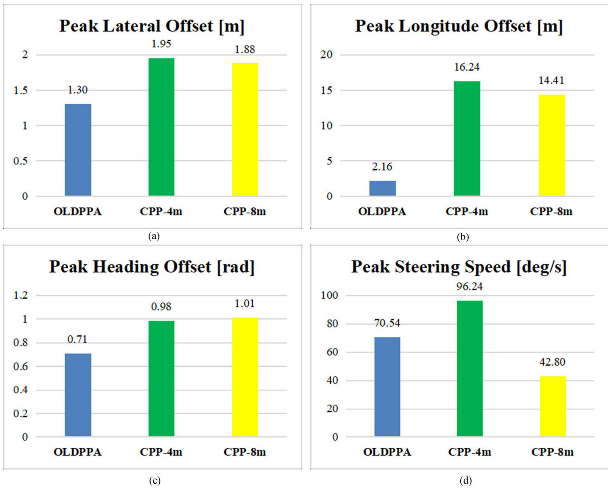

FIGURE 16. Comparison of experiment results with OLDPPA, CPP-4m, and CPP-8m for the lane changing path. (a) Peak lateral offset, (b) Peak longitude offset, (c) Peak heading offset, (d) Peak steering speed.

The quantitative comparison results of the three controllers for this parabolic path are shown in Fig. 16. OLDPPA performs best on the peak lateral offset, peak longitude offset and peak heading offset. In the peak lateral offset, OLDPPA is about 0.65m smaller than CPP-4m and about 0.57m smaller than CPP-8m. In the peak longitudinal offset, OLDPPA is about 14.1m smaller than CPP-4m and about 12.3m smaller than CPP-8m. The peak heading offset of OLDPPA is about 0.28rad less than CPP-4m and about 0.30rad less than CPP-8m. The peak steering speed of OLDPPA is about 19.7 deg/s smaller than that of CPP-4m. Although the peak steering speed of OLDPPA is about 27.8 deg/s higher than CPP-8m, OLDPPA's tracking accuracy is higher than CPP-8m.

C. DISCUSSION

According to the above observations, it can be concluded that ABMSSA has better performance than other algorithms by solving the other ten benchmark functions in addition to F5 and F7. Although the average obtained by ABMSSA is not the best in solving the benchmark function F5, the standard deviation of ABMSSA is the best. Thus ABMSSA has great advantages when the stability of the optimization algorithm is required. Although the average and standard deviation obtained by ABMSSA are not the best in solving the benchmark function F7, it can be seen from Fig. 8 that the convergence speed of ABMSSA is faster than other algorithms, so when the convergence speed of the optimization algorithm is required, ABMSSA has great advantages.

It also can be seen that the tracking accuracy of OLDPPA is higher than the other two algorithms. In terms of tracking stability, OLDPPA is not as stable as CPP-8m, but the stability of OLDPPA is acceptable and can make the vehicle return to the reference path faster. Although OLDPPA has good tracking performance, it still has some limitations. The computational cost in OLDPPA is relatively high, and the optimized look-ahead distance may not be the best result because the number of optimization iterations is limited when ABMSSA is used to optimize the look-ahead distance.

V. CONCLUSION

This study presents OLDPPA based on the optimized look-ahead distance. First, Brownian motion is introduced in SSA to improve the exploitation and exploration capabilities. Second, two adaptive weight is added to the position update mechanism of SSA to accelerate the convergence speed. Based on the two points, ABMSSA is proposed. Third, ABMSSA is used to optimize the look-ahead distance of pure pursuit algorithm. Finally, a new velocity controller is designed in pure pursuit algorithm. OLDPPA solves the problem that the existing pure pursuit algorithms cannot find the best look-ahead distance. OLDPPA not only greatly improves the tracking accuracy of pure pursuit algorithm, but also effectively enhances the tracking punctuality.

To verified the effectiveness of OLDPPA, we have followed four different kinds reference paths, including straight path, sinusoidal path, parabolic path, and lane changing path. Compared with the other two pure pursuit algorithms without optimized look-ahead distances, the lateral error, longitudinal error, and heading angle error of OLDPPA are all obvious smaller when tracking the four paths, and the steering angle is relatively stable. It indicates that the OLDPPA is more suitable to be used in path tracking.

It is worth concerned that, compared with deep learning algorithm, OLDPPA has better interpretability and lower computational cost in improving the accuracy of path tracking of autonomous vehicles. But when the data set is large, deep learning algorithm have greater advantages. In the future, we can try to combine deep learning and path tracking of autonomous vehicles to improve the accuracy of path tracking in the face of large data sets. In addition, the stability and convergence for OLDPPA are an open problem.

REFERENCES

[1] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations," *Transp. Res. A, Policy Pract.*, vol. 77, pp. 167–181, Jul. 2015, doi: 10.1016/j.tra.2015.04.003.

[2] Y. Huang, H. Ding, Y. Zhang, H. Wang, D. Cao, N. Xu, and C. Hu, "A motion planning and tracking framework for autonomous vehicles based on artificial potential field elaborated resistance network approach," *IEEE Trans. Ind. Electron.*, vol. 67, no. 2, pp. 1376–1386, Feb. 2020, doi: 10.1109/TIE.2019.2898599.

[3] C. Ji, Y. Li, J. Fan, and S. Lan, "A novel simplification method for 3D geometric point cloud based on the importance of point," *IEEE Access*, vol. 7, pp. 129029–129042, 2019, doi: 10.1109/ACCESS.2019.2939684.

[4] J. Nie, J. Zhang, W. Ding, X. Wan, X. Chen, and B. Ran, "Decentralized cooperative lane-changing decision-making for connected autonomous vehicles," *IEEE Access*, vol. 4, pp. 9413–9420, 2016, doi: 10.1109/ACCESS.2017.2649567.

[5] A. Artunedo, J. Godoy, and J. Villagra, "A primitive comparison for traffic-free path planning," *IEEE Access*, vol. 6, pp. 28801–28817, 2018, doi: 10.1109/ACCESS.2018.2839884.

[6] C. Sun, X. Zhang, Q. Zhou, and Y. Tian, "A model predictive controller with switched tracking error for autonomous vehicle path tracking," *IEEE Access*, vol. 7, pp. 53103–53114, 2019, doi: 10.1109/ACCESS.2019.2912094.

[7] B. Zhang, C. Zong, G. Chen, and B. Zhang, "Electrical vehicle path tracking based model predictive control with a Laguerre function and exponential weight," *IEEE Access*, vol. 7, pp. 17082–17097, 2019, doi: 10.1109/ACCESS.2019.2892746.

[8] J. Morales, J. L. Martínez, M. A. Martínez, and A. Mandow, "Pure-pursuit reactive path tracking for nonholonomic mobile robots with a 2D laser scanner," *EURASIP J. Adv. Signal Process.*, vol. 2009, no. 1, Dec. 2009, Art. no. 935237, doi: 10.1155/2009/935237.

[9] R. Marino, S. Scalzi, and M. Netto, "Nested PID steering control for lane keeping in autonomous vehicles," *Control Eng. Pract.*, vol. 19, no. 12, pp. 1459–1467, Dec. 2011, doi: 10.1016/j.conengprac.2011.08.005.

[10] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, Jan. 2002, doi: 10.1016/S0005-1098(01)00174-1.

[11] G. V. Raffo, G. K. Gomes, J. E. Normey-Rico, C. R. Kelber, and L. B. Becker, "A predictive controller for autonomous vehicle path tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 92–102, Mar. 2009, doi: 10.1109/TITS.2008.2011697.

[12] Z. Wang, Y. Bai, J. Wang, and X. Wang, "Vehicle path-tracking linear-time-varying model predictive control controller parameter selection considering central process unit computational load," *J. Dyn. Syst., Meas., Control*, vol. 141, no. 5, pp. 1–12, May 2019, doi: 10.1115/1.4042196.

[13] Q. Q. Yao and Y. Tian, "A model predictive controller with longitudinal speed compensation for autonomous vehicle path tracking," *Appl. Sci.*, vol. 9, no. 22, p. 4739, Nov. 2019, doi: 10.3390/app9224739.

[14] H. Andersen, Z. J. Chong, Y. H. Eng, S. Pendleton, and M. H. Ang, "Geometric path tracking algorithm for autonomous driving in pedestrian environment," in *Proc. IEEE Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2016, pp. 1669–1674.

[15] M. Elbanhawi, M. Simic, and R. Jazar, "Receding horizon lateral vehicle control for pure pursuit path tracking," *J. Vib. Control*, vol. 24, no. 3, pp. 619–642, Feb. 2018, doi: 10.1177/1077546316646906.

[16] Y. Shan, W. Yang, C. Chen, J. Zhou, L. Zheng, and B. Li, "CF-pursuit: A pursuit method with a clothoid fitting and a fuzzy controller for autonomous vehicles," *Int. J. Adv. Robot. Syst.*, vol. 12, no. 9, p. 134, Sep. 2015, doi: 10.5772/61391.

[17] H. Ohta, N. Akai, E. Takeuchi, S. Kato, and M. Edahiro, "Pure pursuit revisited: Field testing of autonomous vehicles in urban areas," in *Proc. IEEE 4th Int. Conf. Cyber-Phys. Syst., Netw., Appl. (CPSNA)*, Nagoya, Japan, Oct. 2016, pp. 7–12.

[18] M.-W. Park, S.-W. Lee, and W.-Y. Han, "Development of lateral control system for autonomous vehicle based on adaptive pure pursuit algorithm," in *Proc. 14th Int. Conf. Control, Automat. Syst. (ICCAS)*, Seoul, South Korea, Oct. 2014, pp. 1443–1447.

[19] L. Yu, X. Yan, Z. Kuang, B. Chen, and Y. Zhao, "Driverless bus path tracking based on fuzzy pure pursuit control with a front axle reference," *Appl. Sci.*, vol. 10, no. 1, p. 230, Dec. 2019, doi: 10.3390/app10010230.

[20] C. G. Serna, A. Lombard, Y. Ruichek, and A. Abbas-Turki, "GPS-based curve estimation for an adaptive pure pursuit algorithm," in *Proc. MICAI*, Cancun, Mexico, 2017, pp. 497–511.

[21] W.-J. Wang, T.-M. Hsu, and T.-S. Wu, "The improved pure pursuit algorithm for autonomous driving advanced system," in *Proc. IEEE 10th Int. Workshop Comput. Intell. Appl. (IWCIA)*, Hiroshima, Japan, Nov. 2017, pp. 33–38.

[22] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*. San Francisco, CA, USA: Morgan Kaufmann, 2001.

[23] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006, doi: 10.1109/MCI.2006.329691.

[24] X. S. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic Algorithms: Foundations and Applications*. Berlin, Germany: Springer, 2009, pp. 169–178, doi: 10.1007/978-3-642-04944-6_14.

[25] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems," *Eng. Comput.*, vol. 29, no. 1, pp. 17–35, Jan. 2013.

[26] X. Yang and A. H. Gandomi, "Bat algorithm: A novel approach for global engineering optimization," *Eng. Comput.*, vol. 29, no. 5, pp. 464–483, Jul. 2012, doi: 10.1108/02644401211235834.

[27] B. Yang, L. Zhong, X. Zhang, H. Shu, T. Yu, H. Li, L. Jiang, and L. Sun, "Novel bio-inspired memetic salp swarm algorithm and application to MPPT for PV systems considering partial shading condition," *J. Cleaner Prod.*, vol. 215, pp. 1203–1222, Apr. 2019, doi: 10.1016/j.jclepro.2019.01.150.

[28] D. Wang, Y. Zhou, S. Jiang, and X. Liu, "A simplex method-based salp swarm algorithm for numerical and engineering optimization," in *Proc. Int. Conf. Intell. Inf. Process.* Berlin, Germany: Springer, 2018, pp. 150–159.

[29] G. I. Sayed, G. Khoriba, and M. H. Haggag, "A novel chaotic salp swarm algorithm for global optimization and feature selection," *Appl. Intell.*, vol. 48, pp. 3462–3481, Mar. 2018, doi: 10.1007/s10489-018-1158-6.

[30] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Seoul, South Korea, Jun. 2015, pp. 1094–1099.

[31] H.-G. Park, K.-K. Ahn, M.-K. Park, and S.-H. Lee, "Study on robust lateral controller for differential GPS-based autonomous vehicles," *Int. J. Precis. Eng. Manuf.*, vol. 19, no. 3, pp. 367–376, Mar. 2018, doi: 10.1007/s12541-018-0044-9.

[32] K. Lee, S. Jeon, H. Kim, and D. Kum, "Optimal path tracking control of autonomous vehicle: Adaptive full-state linear quadratic Gaussian (LQG) control," *IEEE Access*, vol. 7, pp. 109120–109133, 2019, doi: 10.1109/ACCESS.2019.2933895.

[33] Y. Liu, D. Iacomini, B. Powell, J. Gafford, J. Ball, J. Shi, and C. Davenport, "Development, testing, and assessment of a kinematic path-following model for towing vehicle systems," *SAE Int. J. Vehicle Dyn., Stability*, vol. 3, no. 1, pp. 57–70, Jan. 2019, doi: 10.4271/10-03-01-0005.

[34] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017, doi: 10.1016/j.advengsoft.2017.07.002.

[35] A. Novikov, D. Kuzmin, and O. Ahmadi, "Random walk methods for Monte Carlo simulations of Brownian diffusion on a sphere," *Appl. Math. Comput.*, vol. 364, Jan. 2020, Art. no. 124670, doi: 10.1016/j.amc.2019.124670.

[36] S. Mirjalili, "The ant lion optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, May 2015, doi: 10.1016/j.advengsoft.2015.01.010.

[37] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[38] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[39] M. H. Qais, H. M. Hasanien, and S. Alghuwainem, "Enhanced salp swarm algorithm: Application to variable speed wind generators," *Eng. Appl. Artif. Intell.*, vol. 80, pp. 82–96, Apr. 2019, doi: 10.1016/j.engappai.2019.01.011.

[40] K. Gholami and M. H. Parvaneh, "A mutated salp swarm algorithm for optimum allocation of active and reactive power sources in radial distribution systems," *Appl. Soft Comput.*, vol. 85, Dec. 2019, Art. no. 105833, doi: 10.1016/j.asoc.2019.105833.

RUI WANG received the B.S. degree from the College of Software, Jilin University, Changchun, China, in 2018, where she is currently pursuing the master's degree. Her research interests include path tracking of autonomous vehicles and swarm intelligence algorithm.

YING LI received the B.S., M.S., and Ph.D. degrees from Jilin University. From 2000 to 2006, she was an Associate Professor with the Department of Space Information Processing, Jilin University, where she has been a Professor in computer application technology since 2006. She has published over 60 articles in journals and international conference. Her research interests include big data, 3D visual modeling, 3D image processing, machine vision, and machine learning. She is a Fellow of the China Computer Federation.

JIAHAO FAN received the B.S. degree from the Computer Science and Technology College, Jilin University, Changchun, China, in 2015, and the master's degree from Jilin University, in 2017, where he is currently pursuing the Ph.D. degree. His research interests include swarm intelligence algorithm, machine learning, image processing, data mining, and 3D data processing.

TAN WANG received the B.S. degree in journalism and communication from the Jilin University of Finance and Economics, Changchun, China, in 2017, and the M.S. degree from Northeast Normal University, Changchun, in 2020. She works in part-time with Space Technology (Jilin) Company Ltd. Her research interests include media technology ethics, network ecological, and feature selection.

XUETAO CHEN received the B.S. degree from the Computer Science and Technology College, Jilin University, Changchun, China, in 2018, where he is currently pursuing the master's degree. His research interests include deep learning, machine learning, image processing, and 3D data processing.