Research on control method and evaluation system of unmanned ground vehicle group change

Tianyu Shi*, Yunqing Bao*, Zhichao Wang*, Chonghao Zou*, Sunbo Wang*, Xueyuan Li

1College of Vehicle engineering, Beijing Institute of Technology, Beijing, 100081, China

Corresponding author’s e-mail: lixueyuan@bit.edu.cn

*these authors contributed equally to this work

Abstract. In this paper, a novel formation control system, composed of path planning and tracking, was designed. A master-slave control model, based on the main vehicle, was proposed to solve the potential accumulation, transmission and amplification errors. To be specific, the referenced trajectory was initially generated by integrating a dynamic window and potential field in the process of transforming. Then a path tracking algorithm based on Hermite curve was introduced, which can make the group change process more stable and accurate. The performance of the formation control system was evaluated, in terms of the formation response and capacity of the formation stability in the simulated scenario. As a result, the vehicle control is simulated, the vehicle can quickly and precisely track the variation of the expected position, the maximum error is 0.1 m, and the speed error of the vehicle following can be controlled within 0.2 km / h, which can keep the stability of the vehicle formation better.

1. Introduction

In general, unmanned ground vehicles (UGVs) are driven by electric or hydraulic power. The differential steering provides various wheel speeds on both sides of the wheel, which brings the speed difference and causes the sliding steering on the ground [10]. The steering vehicle features with a differential steering are difficult to describe because of their extremely complex motion and the force between the wheels and the ground [7]. In this study, it has employed a widely used 4x4 UGV as the research object and creates a dynamic model based on a guidance track concept [8]. This reduces the inaccuracy and issues about control parameters.

In this paper, the related motion disturbance in this formation process is evaluated, integrating the dynamic window approach (DWA) [1] to identify the vehicle’s motion performance synthetically and search the optimal group change path in a certain area, which enables the group change safer and more efficient.

Also, regarding the unmanned vehicle path tracking level, the tracking accuracy and stability are highlighted. The stability can be further maintained by the modified Lyapunov function control [12]. At the same time, this modified function has improved stability in the tracking control of industrial robots [3]. The path tracking method for 4x4 UGV deduced in this paper uses the parametric curve approximation method to control the approaching curve through the speed and direction of the vehicle’s two points. This helps the vehicle to align the head direction during the path tracking process.
In addition, the speed and yaw rate are used for input tracking based on the targeted situation in this research.

1.1. This research’s contributions involve:
(1) A modified APF with DWA-based model, which mitigates the disturbance of moving obstacles and enables the formation process to be more adaptable to the vehicle performance. Also, this helps to achieve a safer and more efficient process.
(2) A time-series-based unmanned vehicle path tracking algorithm, which sets the parameters, such as vehicle speed and yaw rate, as parameter equations with time as an independent variable, which embodies the control quantity in discrete time.
(3) A rational equation to evaluate the process performance, which helps modify the parameters.

The rest of paper is organized accordingly. In the next section, the kinematic properties of the differential steering vehicle model are shown. The succeeding section presents the two levels of control algorithm from path planning and tracking. Then simulations are provided to show the effectiveness of the proposed method. An evaluation system is carried out to analyze the results. Finally, the conclusion is cited in the final section.

**Acronyms**

- $L$: is the width between the left and right axles. (It is the width between the left and right wheels.)
- $d$: is the wheel diameter.
- $v_c$: is the speed of the centroid of the vehicle.
- $w$: is the yaw rate for the vehicle.
- $R_v$: turns radius for the vehicle (It is the distance from the turning center to the centroid of the vehicle.)
- $v_l, v_r$: is the left and right wheel speeds.
- $\theta_0$: is the initial turning angle of the vehicle.
- $v_m, v_f$: are the maximum heading acceleration and deceleration of the unmanned vehicle, respectively.
- $w_m, w_i$: are the largest yaw angular acceleration and deceleration of the unmanned vehicle, respectively.
- $U_r$: is the repulsive potential field.
- $U_a$: is the gravitational potential field.
- $\rho_i$: is the radius of the repulsion circle focused on the $i$th obstacle in the detection range.
- $\rho_s$: is the specified radius value selected based on the obstacle and vehicle attributes.
- $v_{\text{max}}$: is the maximum relative speed of the obstacle and the vehicle.
- $v^\prime$: is the current speed of the obstacle pointing to the vehicle.
- $K$: is the gain constant.
- $v$: is the current vehicle speed.
- $v_{\text{goal}}$: is the target moving speed.
- $v_{\text{obs}}$: is the moving obstacle speed, where $q$ is the vehicle position loss.
- $q_{\text{goal}}$: is the target vector.
- $P$: is the distance between the vehicle and the obstacle.
- $s(t)$: is the actual vehicle position.
- $s_{\text{ideal}}$: is the expected vehicle position.
- $a(t)$: is the instantaneous vehicle acceleration.
- $e_x$: is the unit vector of the x-axis.

2. Vehicle model formulation

2.1. Unmanned vehicle kinematics model
To simplify the model, only the driving wheels are used as the research objects in this paper, employing them as a two-wheeler model. It is assumed that the wheels of the ground unmanned vehicle move on a flat horizontal ground, where the wheels only have a rolling contact with the ground, with its geometric relations and the vehicle’s kinematics model use the formulas:

$$v = [v_L, v_R] = [v_v \times (R - \frac{L}{2}), v_v \times (R + \frac{L}{2})]$$  \hspace{1cm} (1)

The unmanned vehicle steers based on the speed difference between the left and right wheels. When the unmanned vehicle moves in a straight line, since $R$ tends to be infinite, the four-wheel speeds of the unmanned vehicle are equal, which makes $v_L = v_R$. When the unmanned vehicle carries out a retreating motion, the rotation speed of all the wheels of the vehicle can be reversed. If performing an in-place rotation motion, then $R = 0$ and $v_L = 0$, and $v_L = -v_R$. The vehicle adopts a four-wheel differential steering system and the control method is simple. However, as the wheels cannot rotate freely, it causes the motion freedom of the unmanned vehicle low. So a large steering circle space is necessary during the turning process. This means a threshold is required to limit the steering speed and the yaw rate so the vehicle can be steadily controlled and prevent the mutual interference between the vehicles.

Track deduction is used for the bottom control of the robot. It is based on the unmanned vehicle posture (position and posture) and then calculates the speed and yaw rate of the vehicle, particularly the speeds of the left and right wheels. Conversely, it can also reverse the attitude parameters based on the vehicle speed parameters.

First, based on geometric relations, $\theta_L = \theta_1 = \theta_1$ can be derived. The driving wheel of the vehicle is selected as the research object. With its two existing speeds, the vehicle model consists of a two-degree-of-freedom model. So a time-dependent equation is used to establish a track derivation model. Considering the relative speed relationship between the left and right wheels and the body width $L$, the yaw rate of an unmanned vehicle can be expressed as:

$$\frac{d\theta}{dt} = \frac{(v_R - v_L)}{L}$$  \hspace{1cm} (2)

By integrating, the vehicle corner can be derived using the formula:

$$\theta = \frac{(v_R - v_L)}{L} + \theta_0$$  \hspace{1cm} (3)

Similarly, the moving speed of the vehicle can be calculated as:

$$v_c = \frac{v_R + v_L}{2}$$  \hspace{1cm} (4)
Through the sophisticated consideration of the forward velocity and yaw rate of the vehicle, multiple sets of motion parameters are generated by local path planning, then the trajectory can be deduced by integrating the speed using:

\[
\begin{align*}
x(t_f) &= x_0 + \int_0^{t_f} v(t) \cos \theta(t) \, dt \\
y(t_f) &= y_0 + \int_0^{t_f} v(t) \sin \theta(t) \, dt
\end{align*}
\]  

(5)

2.2. Overall Group Change

For the group change, it assumes that five vehicles consist of a team. The initial formation is longitudinally arranged, which in turn translates into a horizontal formation, a triangular formation, and finally a vertical formation. The process is detailed as follows:

The initial fleet vehicles are numbered 1 to 5, where number 3 is at the center, which is the main vehicle. (Figure 2 (a)) When changing the formation, the other four vehicles’ baselines are attained by shifting the baseline of the main vehicle to the left and right, corresponding to Nos. 1, 2, 4, and 5. Each vehicle seeks an expected path and targets the baseline along with the expected path. At the same time, each adjusts the speed and position to complete the transformation from the longitudinal line to the horizontal line. (Figure 2 (b))

When the formation converts to a triangular, it only needs to change the speeds, No. 1 at low speed, Nos. 2 and 4 at medium speed, and No. 5 at high speed. (Figure 2 (b))

This is the process of group change. The process of restoring the vertical formation is as follows.

The car formation is first converted from a triangle to a slash and then converted to a vertical form. No. 3, as the main vehicle, is always kept at a straight line along the road. All the paths of the servants are attained based on the path of the mean vehicle while all the following vehicles maintain a certain distance from the main vehicle. This ensures that main vehicle is found at the center. Therefore, at the start of the formation, it is set No. 3 as the main vehicle and used as the basis for the group change.

2.3. Path planning

Path planning is used to deduct the next state parameter \((v_{t+1}, w_{t+1})\) based on the motion parameters \((v_t, w_t)\) of the unmanned vehicle previously used. Employing the earlier created vehicle trajectory estimation formula (5), the trajectory can be deduced from the velocity, taking into account the computational complexity when the sampling on an unmanned vehicle is in motion, including only the sampling multiple sets of the velocity parameters \((v, w)\) in the 2D space. The velocity parameters \((v, w)\) can only be used to extrapolate the vehicle trajectory.
First, the unmanned vehicles are limited by their own maximum and minimum speeds.

\[ V_{\text{en}} = \{ v \in [v_{\text{min}}, v_{\text{max}}], w \in [w_{\text{min}}, w_{\text{max}}] \} \]  

(6)

Because the unmanned vehicle is impacted by its motor performance, along with the limited motor torque, there are both maximum acceleration and deceleration limit. Therefore, the unmanned vehicle has a dynamic range within a certain simulation period, which is the speed range of the vehicle. While \((v, w)\) is the actual speed that unmanned vehicles can achieve:

\[ v_a = \{ (v, w) \mid v \in [v_i - v_a \Delta t, v_i + v_a \Delta t], w \in [w_i - w_a \Delta t, w_i + w_a \Delta t] \} \]  

(7)

To stop before hitting an obstacle, the speed must be set within a certain range under the maximum deceleration condition:

\[
\begin{align*}
\Omega & = \min(2 \text{dist}(v, w) v_a \alpha, (2 \text{dist}(v, w) w_a)^{1/2}, \frac{v_a \tan \alpha}{l}) \\
\end{align*}
\]

(8)

Second, the obstacle avoidance algorithm is designed in path planning. There are two types of obstacles in the motion process of the vehicle. One is the static environmental obstacle and the other is the vehicle in motion state.

Finally, the scope of the vehicle’s velocity is a combination of those three equations:

\[ V = V_{\text{en}} \land V_a \land V_a \]  

(9)

This forms the dynamic window constraint as shown in Figure 3.

The APF method [11], proposed by Khatib, is one of the commonly used methods for local online obstacle avoidance among mobile robots. The principle behind is as follows:

The mobile robot shifts in a virtual force field. The obstacle is surrounded by the repulsive potential field, \( U_r \). The resulting repulsive force increases when the distance between the robot and the obstacle is reduced, and the direction deviates from the obstacle. The target point is enclosed by the gravitational potential field, \( U_g \). The resulting gravitational force declines as the robot approaches while the direction shifts to the target point. Then, based on the total of the artificial potential energy generated by each obstacle and the target, the gradient direction of the potential function is used to achieve a collision-free path planning.

However, because the path planning is only based on the information about the robot, obstacles, and targets, the scope of its application is limited to the static environments. Several modifications to the traditional artificial potential field method can be used not only for tracking and avoiding mobile obstacles but also take advantage of the simplicity and high speed of the control system.

To attain this, the method for calculating the radius of the repulsion circle of the original artificial potential field is integrated with the relative vehicle velocity while the detection range of the vehicle is \( \rho_c \). For the \( i \)th obstacle in the \( \rho_c \) range, the repulsive force is defined as a circle focused on the \( i \)th obstacle and the radius \( \rho_e \), where the radius of the repulsive circle is related to the relative velocity of the \( i \)th obstacle:
Considering the unmanned vehicle 1 as the research object, its own detection radius should be set at $\rho_o$. Through the perception of the surrounding environment, the nearest vehicle $i$ is known as an obstacle and the radius of the repulsive circle is set to $\rho_i$.

$$\rho = (1 + \frac{v}{v_{\text{max}}})\rho_o \quad (10)$$

In the case of $\rho \geq \rho_o$, the vehicle cannot identify the obstacles. However, based on the previous experimental results, it can make the vehicle operationally confusing and may generate *in situ* circling behavior. Therefore, path planning is reconsidered in this case. First, it needs to create simultaneous equations that turn to the target points. The simultaneous unmanned vehicle points to the target point linear equation $L (x, y)$ and other unmanned vehicle’s repulsive force equation $f (x, y)$, which is nearest to the unmanned vehicle.

Discuss the number of roots:
1. If the function $L (x, y)$ and the repulsive force equation $f (x, y)$ have no roots, the vehicle does not do obstacle avoidance actions and goes directly to the target.
2. If the function $L (x, y)$ and the repulsive force equation $f (x, y)$ have two roots, then the vehicle selects the intersection point closer to the robot and create a tangent of the repulsion loop. Then, selecting the direction of the tangent toward the target. This can be used to process moderate preventive obstacle avoidance.
3. If $L (x, y)$ and the repulsive force equation $f (x, y)$ have more than three roots, the vehicle does the tangent line between the robot and each effective repulsion circle. Selecting the direction toward the robot and identify the angle between each tangent line and $L$, and then taking the direction of the tangent with the minimum angle of $L$ as the robot’s motion direction.

Also, in the case of $\rho \leq \rho_o$, using the correction formula combined with the target motion speed of the vehicle. It can be calculated:

Gravitational field potential energy function is:

$$U_{\text{attr}} = \frac{1}{2} K_{\text{aq}} |q - q_{\text{goal}}|^2 + \frac{1}{2} K_{\text{att}} |v - v_{\text{goal}}|^2 \quad (11)$$

The attraction function is:

$$F_{\text{attr}} = -\text{grad}(U_{\text{attr}}) = -K_{\text{aq}} |q - q_{\text{goal}}| - K_{\text{att}} |v - v_{\text{goal}}| \quad (12)$$

The repulsive field potential energy function is:
The repulsive force function is:

\[ F_{\text{rep}} = -\text{grad}(U_{\text{rep}}) \]

\[ F_{\text{rep}} = K_{\text{rep}}[\rho(q, q_{\text{obs}})^{-1} - (\rho)^{-1}]^2 + K_{\text{rep}} |v - v_{\text{obs}}|^2 \]  

\[ (13) \]

This method divides the obstacles to make the obstacle avoidance action more reasonable. At the same time, it launches preventative obstacle avoidance, instead of simple and direct obstacle avoidance treatment, which effectively assists the potential field method to complete obstacle avoidance and tracking task.

### 2.4. Path tracking

Taking the center of the mass of the main vehicle as the original point, the direction of the actual speed of the vehicle is at the y-axis to set up the first coordinate system, \( x_1y_1 \). Identifying the center of mass of the following vehicle as the original point, the direction of the actual speed of the following vehicle is the \( x_1' \) \( y_1' \) axis to set up the second coordinate system, \( x_2' \) \( y_2' \), as shown in Figure 5.

When the team of vehicles is changing the formation, each vehicle travels on its corresponding path’s baseline and the target path between the two-route baselines can be obtained from the cited path planning method:

\[ R_{\text{ideal}}(t) = (x_{\text{ideal}}(t), y_{\text{ideal}}(t)) \]  

\[ (15) \]

Simultaneously, the ideal velocity can be calculated as:

\[ v_{\text{ideal}}(t) = \nabla R_{\text{ideal}}(t) = \begin{bmatrix} x_{\text{ideal}}(t) \\ y_{\text{ideal}}(t) \end{bmatrix} = \begin{bmatrix} v_x(t) \\ v_y(t) \end{bmatrix} \]  

\[ (16) \]

Currently, the speeds of \( x \) and \( y \) are the first coordinate system speeds. Furthermore, the expected transverse angular velocity can be calculated as:

\[ \omega_{\text{ideal}}(t) = \frac{\Delta \theta_{\text{ideal}}(t)}{\Delta t} = \frac{\theta_{\text{ideal}}(t) - \theta_{\text{ideal}}(t-1)}{\Delta t} \]  

\[ (17) \]

At the same time, the actual speed and the actual yaw angular velocity can be obtained. The yaw angular velocity takes a clockwise movement as positive while counterclockwise is considered negative. There are three states existing when the vehicle is running on the expected path closely:

1. The vehicle is on the left side of the path.
2. The vehicle is on the right side of the path.
3. The vehicle is precisely on the path.

These three states are shown in Fig. 6(a).
Figure 5. Hermite curve path planning for vehicles outside the expected path

Regarding it is difficult to completely coincide with the centroid and the route. Therefore, this paper simply defines a circle whose center is the focus of the centroid and its radius is relatively small. When the path has an intersection with the circle, it is regarded as the vehicle is on the path.

\[
\begin{align*}
\text{If } x_{\text{ideal}}(t) > 0, \text{ the vehicle is on the left side of the path} \\
\text{If } x_{\text{ideal}}(t) < 0, \text{ the vehicle is on the right side of the path} \\
\text{If } x_{\text{ideal}}(t) - r < 0, \text{ the vehicle is on the path}
\end{align*}
\]

(18)

Figure 6. (a) Stated judgment when the vehicle runs on the expected path
(b) The coordinates that establish the relative position between the vehicle and the expected path

When the centroid of the vehicle deviates from the expected trajectory, extend the axis of the vehicle as it intersects with the expected path \(R_{\text{ideal}}(t)\) at the point \(A_{\text{ideal}}\).

The magnitude and direction of the speed and yaw rate at this point can be calculated using the method in 2.1. The position is shown in Figure 5.

At the same time, the vehicle real-time coordinates and the direction of the actual velocity are identified. Considering that the predetermined position must be reached during the running of the vehicle, and the direction of the vehicle is adjusted to facilitate the next driving. Therefore, the planned path should meet the following conditions: (1) smooth transition; (2) starting and arrival points are on this path; (3) initial ending velocity vectors are tangent to this path. With these three conditions, it can be noted that the parameterized curve Hermite curve can meet the condition. The parameters of the Hermite curve are set up according to the constraints of the starting point, the terminal coordinates, and the velocity vector.
\begin{equation}
\begin{aligned}
p(u) &= au^3 + bu^2 + cu + d, \quad 0 \leq u \leq 1 \\
a &= 2p_0 + 2p_i + p_i^w \\
b &= -3p_0 + 3p_i - 2p_i^w - p_i^w \\
c &= p_i^w \\
d &= p_0
\end{aligned}
\end{equation}

In the formula (20), \( p_0, p_i \) is the current point and the coordinate of the point \( A_{\text{ideal}}, p_i^w \), where \( p_i^w \) is the velocity vector of the current point and the point \( A_{\text{ideal}} \).

The curve is taken as a new approach route, and the problem of the vehicle deviating from the original path is translated into path tracking of the vehicle. For the converted problem, the tracking method of the vehicle is described later. When the centroid is on the expected trajectory, a vector tracking method [13] is used to model the vehicle-tracking path.

Calculating the angle between the actual and the expected speeds of the vehicle (Figure 5).

\begin{equation}
\theta(t) = \arccos\left(\frac{v(t) \cdot v_{\text{ideal}}(t)}{|v(t)| \cdot |v_{\text{ideal}}(t)|}\right)
\end{equation}

Calculating the cosine of the angle between the expected direction of velocity and the x-axis of the second coordinate system:

\begin{equation}
\psi(t) = \cos(\phi(t)) = \frac{v_{\text{ideal}}(t)}{|v_{\text{ideal}}(t)|}
\end{equation}

In this formula, \( \phi \) is the unit vector of the x-axis.

Setting the angle threshold \( \theta_{\text{threshold}} \),

\begin{equation}
\begin{aligned}
\psi(t) \geq 0, & \text{ the following vehicle needs to turn right} \\
\psi(t) < 0, & \text{ the following vehicle needs to turn left}
\end{aligned}
\end{equation}

Then, the Steering determination formula is:

\begin{equation}
\phi(t) = \max(\theta(t) - \theta_{\text{threshold}}, 0) \psi(t)
\end{equation}

Setting the speed threshold \( v_{\text{threshold}} \), where the judgment condition of speed adjustment is:

\begin{equation}
f(t) = \max\left| |v_{\text{ideal}}(t)| - |v(t)| - \theta_{\text{threshold}} \right| \\
\begin{cases}
\begin{aligned}
f(t) > 0, & \text{ the following vehicle needs to accelerate} \\
f(t) = 0, & \text{ the following vehicle needs to maintain the speed} \\
f(t) < 0, & \text{ the following vehicle needs to decelerate}
\end{aligned}
\end{cases}
\end{equation}

The adjustment model is:

\begin{equation}
\begin{aligned}
v(t_{i+1}) &= v(t_i) + \text{sign}(f(t_i)) \frac{|v_{\text{ideal}}(t_i)| - |v(t_i)|}{\tau} \\
w(t_{i+1}) &= w(t_i) + \text{sign}(f(t_i)) \frac{\theta_{\text{ideal}}(t_i) - \theta_{\text{threshold}}}{\tau}
\end{aligned}
\end{equation}

In the process of turning the vehicle adjustment, the change of speed and yaw rate should be uniform and stable to avoid instability. In turn, the vehicles can roll over and so on.

3. Experiment and Performance Evaluation

3.1. The structure of the formation system
The formation system consists of both the hardware and software parts. The hardware component involves the vital part of the main and following vehicles, industrial computers, STM32 microcontrollers, and a series of vehicle sensors on board. The software component includes the path planning and path tracking parts by integrating the dynamic window approach and potential energy field. In the running method, the main and following cars communicate via Wi-Fi. The master car transmits the baseline equation of its own path to the corresponding slave cars by offsetting the transformation. Upon receiving the signal of the baseline equation from the vehicle, the path planning method is used to achieve an optimal path to the baseline. At the same time, the vehicle speed and yaw rate collected by the on board sensors are filtered out and used as inputs, then employing the path tracking method. The program is burned into the micro-controller unit (MCU) through inter-process communication (IPC) to achieve the cited group change.

3.2. Simulation in the Robot Operating System

The Robot Operating System (ROS) in the following section [21] is an open source meta-operating system suitable for the robots. Based on the ROS, this paper simulates the formation control of the five unmanned vehicles and the findings can be used to validate the implementation ability of the controller [22].

As seen in the formation process, the simulation process resolution is set to 0.02, which is calculated every 0.02 s. First, the initial state is the formation A, which is along the longitudinal line. The vehicle formation transforms to the middle state B, which is askew formation, by issuing a command to convert the formation. The transmitting time is 11.6 s and the relative positions can be regulated by the adjustment of the relative velocity so that the skew formation can be converted into a horizontal line while the transmitting time is 3 s. Then, it follows the same steps to adjust the relative positions to convert the formation to a triangular one.

![Figure 6. The test of group change](image)

3.3. Result Analysis

There are three main aspects to evaluate the control system performance: (1) under the control of the system, (2) proximity of the vehicle’s real-time position to the theoretical position, which is the approaching degree of the actual vehicle speed to the expected vehicle speed, and (3) fluctuation of the vehicle speed, that is, the magnitude of the acceleration. Based on these three criteria, proper weight is set for each and then creates the control index function $W$ to analyze the control performance of the control system, where the functional relation is as follows:

$$W = \int_0^t \left[ (s(t) - s_{ideal}(t))^2 + 9 \times \left( \frac{V(t) - V_{ideal}(t)}{3.6} \right)^2 + 25 \times (a(t))^2 \right] dt$$  \hspace{1cm} (28)

The smaller the $W$ value is, the better function that the control system has. Based on the control index function $W$, it can be found the best control strategy to track the target path under the condition of a slight acceleration fluctuation. In the design simulation experiment, a planned lane change path is initially achieved:

$$R_{ideal}(t) = (X_{ideal}(t), Y_{ideal}(t))$$  \hspace{1cm} (29)

The path is derived from time and the variation of line speed. In this case, the speed change is the ideal speed curve while ideal speed direction is the tangent of the ideal path at the corresponding position.
When adopting the cited control strategy, the results of the simulation are as follows:

**Figure 7.** The path and speed transformation diagrams that correspond to the path

For the analysis of the cited figures, the relative position in Fig. 7 (a) is the difference between the actual vehicle position and the expected vehicle position. Under the control system, the vehicle can track the change of the expected position easily. The maximum error is about 0.1 m in the entire process. This embodies high-tracking accuracy.

In Fig. 8 (b), the dotted line is the expected vehicle speed while the solid line is the actual vehicle speed. When the expected speed changes rapidly, the self-vehicle speed can respond accordingly and the overshoot is relatively inconspicuous. If the expected vehicle speed is well-maintained, vehicles can run at a stable speed without generating the oscillation, showing that the system is relatively stable.

In Fig. 9 (a), the relative speed is the difference between the actual and expected vehicle speeds. The simulation shows that the speed error can be controlled within 0.2 km/h. Within 0 s to 5 s, the vehicle simply enters the path tracking state, where the speed drastically fluctuates and rate fluctuation is very obvious. Within 5 s to 25 s, the vehicle is in a stable path tracking state, and the vehicle speed has relatively high stability. At 25 s to 30 s, the lane change is completed, as the speed slightly fluctuates. In Fig. 7 (b), the vehicle has high stability at low speed while the speed fluctuates when the vehicle is driving at a high speed.
In Fig. 9 (b), the maximum acceleration $a_{\text{max}} = 0.07 \text{ g}$, indicating that under the control system, the speed of the vehicle varies smoothly. There is no risky condition in the rapid acceleration and deceleration, which is advantageous to enhance driving safety.

4. Summary and Conclusion
This paper presents a innovative method of the dynamic planning path and path tracking of the unmanned vehicles. This study recommends a collaborative formation controller design for multiple UGV designs from the two components, path planning and following. Finally, the simulation results are demonstrated and analyzed.

With regard to path planning, the DWA algorithm can effectively avoid the collision with static obstacles. When the vehicles encounter the dynamic obstacles, they move along with the obstacles, which will lead to a great disturbance to the path planning. Therefore, this paper adopts the obstacle avoidance algorithm based on DWA and potential energy field method, which effectively reduces the possibility of interference caused by other platform movements during the process of changing teams. Also, it enables the vehicles to track the movement of the target points more quickly and precisely. Meanwhile, through a series of constraints based on the vehicle’s own performance, the vehicle can avoid rapid acceleration or deceleration, where this new approach has better maneuverability.

In path tracking level, this method simplified the motion control sets to the speed of the vehicle and the yaw rate. And then the baseline equation, which is calculated by the path planning algorithm, is transmitted from the main vehicle to the following vehicle. Then, the state decision and adjustment functions are set. The former controls the steering direction, acceleration, and deceleration of the vehicle while the latter regulates the speed based on the time. When the vehicle deviates from the expected path, the algorithm extends the axis of the vehicle and gets into an intersection with the path. Based on the intersection and its expected speed, its current position and speed, planning a smooth Hermite curve ensure that the vehicle can approach the original and well-planned path smoothly and fix the direction of the vehicle in the process.

Finally, the control performance of the path tracking control system is verified using the simulation experiments applied in this paper. Throughout the simulation process, the vehicle can track the variation of the expected position fleetly and precisely, with a maximum error of 0.1 m. Meanwhile, the vehicle can follow the expected speed rapidly with an inconspicuous overshoot while the speed error can be controlled within 0.2 km/h. Therefore, the path tracking control system guarantees the formation stability of several unmanned vehicles.

Acknowledgments
The authors are very grateful for the reviewers for their valuable suggestions and the final support from Cool High Tech (Beijing) Co, Ltd.

References
[1] Wells, P., Deguire D. (2005) Talon: a universal unmanned ground vehicle platform, enabling the mission to be the focus. In: Proc. SPIE, Unmanned Ground Vehicle Technology VII. Florida. 5804.
[2] Burgard, W., Moors, M., Stachniss, C., Schneider, F.E. (2005) Coordinated multi-robot exploration. IEEE T. Robot., 21: 376-386.
[3] Olson, E., Strom, J., Morton, R., Richardson, A., Ranganathan, P., & Goeddel, R., et al. (2012) Progress toward multi-robot reconnaissance and the magic 2010 competition. J.Field Robot., 29:762-792.
[4] Yanco, H. A., Drury, J. L. (2007) Rescuing interfaces: a multi-year study of human-robot interaction at the AAAI robot rescue competition. Auton. Robot., 22: 333-352.
[5] Cong, M., Fang, B. (2008) Multisensor fusion and navigation for robot mower. IEEE International Conference on Robotics and Biomimetics, pp. 417-422.
[6] Nikitenko, A., Grundspenkis, J., Liekna, A., Ekmanis, M., Kulikovskis, G., Andersone, I. (2014) Multi-robot System for Vacuum Cleaning Domain. PAAMS International Conference on Practical Applications of Agents and Multi-Agent Systems. pp. 363-366.

[7] Allard, J., Barrett, D. S., Filippov, M., Pack, R. T., & Svendsen, S. (2009) Systems and methods for control of an unmanned ground vehicle. US patent. US 7499776 B2.

[8] Borhaug, E., Pettersen, K. Y., & Pavlov, A. (2006) An optimal guidance scheme for cross-track control of underactuated underwater vehicles. Mediterranean Conference on Control and Automation. Ancona. pp.1-5.

[9] Shi, T., Wang, P., Chan, C. Y., & Zou, C. (2019). A Data Driven Method of Optimizing Feedforward Compensator for Autonomous Vehicle. arXiv preprint arXiv:1901.11212.

[10] Li, X., Xiao, J. (2005) Robot formation control in leader-follower motion using the direct Lyapunov method. Int. J. Intell. Control Syst., 10: 244-250.

[11] Min, C. L., & Min, G. P. (2003). Artificial potential field based path planning for mobile robots using a virtual obstacle concept.

[12] Jeff Wit, Carl D. Crane III, & David Armstrong. (2010). Autonomous ground vehicle path tracking. Journal of Field Robotics, 21(8), 439–449.

[13] Iii, C. D. C., Ii, D. G. A., Touchton, R., Galluzzo, T., Solanki, S., & Lee, J., et al. (2010). Team cimar's navigator: an unmanned ground vehicle for the 2005 Darpa grand challenge. Journal of Field Robotics, 23(8), 599-623.

[14] Heredia, G. (1995) Stability analysis of mobile robot path tracking. IROS '95 Proceedings of the International Conference on Intelligent Robots and Systems. Washington. pp.461-466.

[15] Koh, K.C., Cho, H.S. (1999) A smooth path tracking algorithm for wheeled mobile robots with dynamic constraints. J. Intell. Robot. Syst., 24: 367-385.

[16] Kanayama, Y., Kimura, Y., Miyazaki, F., Noguchi, T. (1991) A stable tracking control method for an autonomous mobile robot. IEEE International Conference on Robotics and Automation. Cincinnati. pp.384-389.

[17] Wang, Yan Qing, Ye, Yan Hui, & Zhang, Hua. (2011). A stable tracking control method for an autonomous welding mobile robot. Applied Mechanics & Materials, 79, 264-269.

[18] University, S., & Ji, N. (2007). Shape interpolating geometric g−1 Hermite curves. Journal of Computer-Aided Design & Computer Graphics.

[19] Wit J., Crane III C.D., Armstrong D. (2010) Autonomous ground vehicle path tracking. J. Field Robot., 21: 439–449.

[20] Piserchia, G.T., Wilson, E.F. (1994). Vehicle guidance track system.US patent, US5297484.

[21] Kim, J., Jung, S. (2002) A dynamic window-based approximate shortest path re-computation method for digital road map databases in mobile environments. Eurasian Conference on Information and Communication Technology. Shiraz. pp.711-720.

[22] Quigley, M., Gerkey B., Conley K., Faust J., Foote T., Leibs J. et al. (2009) ROS: an open-source robot operating system. Proc. IEEE ICRA Workshop on Open Source Robotics. pp. 1-6.