Design of trail tracking & Obstacle avoidance system based on model predictive control and artificial potential field

Gongxing Yu

School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China

Abstract. Autopilot is a significant domain in control and computer science in recent years. Generally speaking, auto driving is the mixture of route tracking and obstacle avoidance. In order to realize the automatic tracking of given routes by vehicle models as well as obstacle avoidance of high quality and efficiency, firstly, this paper explains the prevalent tracking method—Model Predictive Control and applies it to a given model. Secondly, an optimized obstacle avoidance method based on Artificial Potential Field using Virtual Destination Point and Weight Allocation is introduced in this approach. Finally, simulations show that fast and stable trajectory tracking and obstacle avoidance can be achieved when the posture and speed of the small parking space meet the requirements.

Keywords: Model Predictive Control, Artificial Potential Field, Trail Tracking.

1. Introduction

The global status report on road safety released by WHO in 2018 pointed out that there are more than 130 million people lose their lives in road accidents, and 90 percent of them are caused by man-made reasons. In order to free people's hands and relief nerve fatigue, driverless technologies came into being, which means computer takes human's seats as a driver. Model predictive control applied on vehicles emerged in Europe in 2019, since then, it has been rated as the most potential vehicle control technology in the World Industrial Symposium, only one place away from the mostly widely used PID control technology. However, due to the limited computing capacity of microchips, MPC has not been applied to autopilot on a larger scale.

This paper tried to combine model predictive control and obstacle avoidance to explore a fast and efficient vehicle control system

2. Solutions

The traditional vehicle control methods mainly include PID control and LQR, both of which have certain defects. Although the integral link of PID control can eliminate the residual in the trajectory tracking process, it will bring a certain lag to make the control effect worse, and a large proportional link will also lead to the full vibration of the travel process and cause danger. The LQR (linear quadratic regulator) algorithm carries out closed-loop control of the linearized vehicle model. The final designed control system has the advantages of high stability and good real-time performance , but in the tracking process there is a large deviation and the LQR algorithm cannot limit the state and control to deal with the constraint problem.

The principal of Model Predictive Control (MPC) is to find out the input output relationship of the MIMO system (the state equation after linearization) and the current state of the system, minimizing the deviation from the reference position to obtain a quadratic planning cost function, so that we can calculate the change of control in the control horizon and carry out rolling optimization. The advantage of MPC is good real-time and accuracy. It can move in advance and adjust the safety and position of the car in reverse time.

Popular obstacle avoidance algorithms are artificial potential field, particle swarm algorithm, A* algorithm and ant colony algorithm. Particle swarm algorithm and A* algorithm can be searched in any direction, but this method has a large amount of calculation. In complex environments, the calculation time is increased and the timeliness is poor. The ant colony algorithm makes reasonable use of pheromone inspiration and expected inspiration, and then changes the fall coefficient. In the

process of path planning, the robot uses a more reasonable dynamic obstacle avoidance strategy, but the path that this algorithm generates is relatively torturous, which is difficult to meet kinematics constraints of the vehicle model.

Artificial Potential Field (APF) method, on the other hand, raised by Stanford's professor Oussama Khatib in 1986, is an efficient approach of obstacle avoidance for unmanned systems. Its principle is to transform the movement of the vehicle in the driving environment into that in an abstract potential field, so that the vehicle can drive to the target point under the action of the compound situation force, and plan the best obstacle avoidance path. This algorithm is suitable for environments with unknown information conditions, with simple calculation and planned paths, it is widely used in the real-time obstacle avoidance of self-driving cars.

3. Vehicle Kinematics Modeling

We have three status variables of a vehicle:x, y, φ

$$\dot{x} = vcos\psi$$
$$\dot{y} = vsin\psi \quad (1)$$

When building the model, give some assumptions:

1 the vehicle will not slip while moving

2 The speed can change instantaneously (regardless of acceleration), but the amount of change should not be too large (from minus 0.25ms to 0.25ms)

3. The maximum steering angle is 30

4. The space is large enough and the vehicle is regarded as a mass point.

5. The speed and angular speed of the previous moment is v_r and ω_r respectively.

According to life experience, people usually use the pedal and steering wheel when controlling the car. These two components correspond to the speed and angular speed (that is, the amount of change of heading angle) of car model respectively, thus, we choose v and ω as control quantities

The formula (1) is linearized at any point of the trajectory (x_r, y_r, φ) with first order Taylor expansion:

$$\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{\varphi}\end{pmatrix} = \begin{pmatrix}v_r cos\varphi_r\\ v_r sin\varphi_r\\ w_r\end{pmatrix} + \begin{pmatrix}0 & 0 & v_r sin\varphi_r\\ 0 & 0 & v_r cos\varphi_r\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}x - x_r\\ y - y_r\\ \varphi - \varphi_r\end{pmatrix} + \begin{pmatrix}v_r cos\varphi_r & 0\\ v_r sin\varphi_r & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}v - v_r\\ w - w_r\end{pmatrix}$$

Then we have the change of status $\Delta x = \begin{bmatrix}x - x_r\\ y - y_r\\ \varphi - \varphi_r\end{bmatrix}$ and change of control quantities $\Delta u = \begin{bmatrix}v - v_r\\ \omega - \omega_r\end{bmatrix}$. Notice that the derivatives are still present in the equation and should be discretized for computer processing. According to the first-order forward Eulerian difference method

$$\dot{X} = \frac{\bar{x}(k+1) - \bar{x}(k)}{T} = A * \begin{bmatrix}x(k+1) - x(k)\\ y(k+1) - y(k)\\ \varphi(k+1) - \varphi(k)\end{bmatrix} + B * \begin{bmatrix}v(k+1) - v(k)\\ \omega(k+1) - \omega(k)\end{bmatrix}$$

$$T = 1, \text{Sampling Period}$$

$$\bar{x}(k+1) = (1 + A) * \begin{bmatrix}x(k+1) - x(k)\\ y(k+1) - y(k)\\ \varphi(k+1) - \varphi(k)\end{bmatrix} + B * \begin{bmatrix}v(k+1) - v(k)\\ \omega(k+1) - \omega(k)\end{bmatrix}$$

Then we obtain two time-varying coefficient matrices:

$$A(k) = \begin{bmatrix}1 & 0 & -v_r(k) sin\varphi_r(k)\\ 0 & 1 & v_r(k) cos\varphi_r(k)\\ 0 & 0 & 1\end{bmatrix}, B(k) = \begin{bmatrix}cos\varphi_r(k) & 0\\ sin\varphi_r(k) & 0\\ 0 & 1\end{bmatrix}$$

Which means that we can always describe the status of the vehicle at the next sampling time in this way:

$$X(k+1) = A(k) * X(k) + B(k) * u(k)$$

where $X(k)$ is the state quantity at time k relative to the k-1 time difference, and u(k) is the control quantity at time k relative to the k-1 time difference (i.e., the controller output at time k-1, and both

the A, B matrices need to change as k changes. At this point, the kinematic model of the car is basically completed.

4. Design of MPC controller

4.1 Basic information and preset parameters

Model Predictive Control (MPC) is the sampling of data within the control horizon Nc in the Np time that the system can predict, i.e., assuming that the system status is sampled up to k, the controller can already predict the system state and control quantities up to k+Nc. At each sampling moment the system performance is predicted for some time in the future. The prediction model is corrected by feedback from the measured data of the system's current status and transformed into a real-time online optimization problem, which is solved to obtain a control sequence with the first control quantity applied to the system, and repeated by rolling the predictive horizon forward one step.

The system block diagram is shown in Fig.1

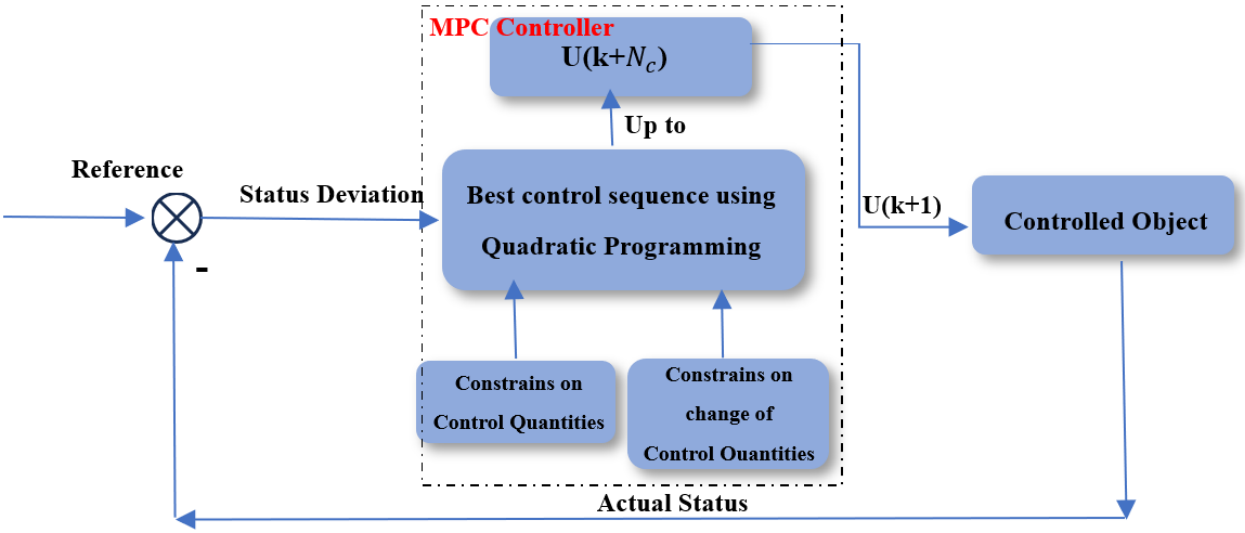


Fig. 1 System Block Diagram

In the vehicle model, predictive horizon N_p=20, control horizon N_c=30.

4.2 Prediction and Constraints

In order to predict the system state at the next moment, the current system state quantity and the output control quantity at the previous moment are combined to form a new variable $\xi(k) = \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix}$ as the state quantity input to form a new state space equation.

Taking $\xi(k+1)$, and switch it to the form of the vehicle kinematic model:

$$\xi(k+1) = \begin{bmatrix} A(k) & B(k) \\ 0_{m*n} & I_m \end{bmatrix} * X(k) + \begin{bmatrix} B(k) \\ I_m \end{bmatrix} * u(k) = \tilde{A} * X(k) + \tilde{B} * u(k)$$

where m is the number of control quantities (2) and n is the number of state quantities (3). To derive the system status after Np from X(k) and u(k), the above equation is expanded.

$$\varepsilon(k+1) = \tilde{A} * X(k) + \tilde{B} * U(k)$$
$$\varepsilon(k+2) = \tilde{A}^2 * X(k) + \tilde{A}\tilde{B} * U(k) + \tilde{B} * U(k+1)$$
$$\varepsilon(k+3) = \tilde{A}^3 * X(k) + \tilde{A}^2\tilde{B} * U(k) + \tilde{A}\tilde{B} * U(k+1) + \tilde{B} * U(k+2)$$
$$\vdots$$
$$\varepsilon(k+N_p) = \tilde{A}^{N_p}\tilde{B} * X(K) + \cdots + \tilde{A}^{N_p - N_C - 1}\tilde{B} * U(k+N_C)$$

Then we have another predictive function:

$$Y = \psi * \xi(k) + \Theta * \Delta u(k)$$

$$Y=\begin{bmatrix}\xi(k+1)\\ \xi(k+2)\\ \vdots\\ \xi(k+N_p)\end{bmatrix},\psi=\begin{bmatrix}\tilde{A}\\ \tilde{A}^2\\ \vdots\\ \tilde{A}^{N_p}\end{bmatrix},\Theta=\begin{bmatrix}\tilde{B} & 0 & \dots & 0\\ \tilde{A}\tilde{B} & \tilde{B} & \ddots & \vdots\\ \vdots & \ddots & \tilde{B} & 0\\ \tilde{A}^{N_p}\tilde{B} & \dots & \tilde{A}^{N_p-N_C-1}\tilde{B} & \tilde{B}\end{bmatrix}$$

If the control and state quantities need to be separated out again, an output matrix $C=\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\end{bmatrix}$ is needed.

The coefficient state matrix can be updated with the help of the C-matrix, or the system state quantities $\eta(k)$ can be taken:

$$\psi = C*\psi, \Theta = \mathrm{C}*\Theta, \eta(k) = C*\xi(k)$$

System's optimization depends on the deviation of the actual trajectory from the reference one, a loss function measuring controller's output accuracy is given below:

$$J=\min\sum||\eta-\eta_r||^2*Q+\sum\Delta u^2*R+\rho\varepsilon^2$$

It can be seen that the optimization direction of the system is to keep the deviation value $|\eta - \eta r|$ as small as possible, and also to make the single output of the controller as small as possible, so as to ensure the smoothness of the vehicle's travel path and prevent the relatively large changes of the car position attitude (especially the starting point is far from the reference trajectory). The Q and R matrices in the equation are both positive definite matrices, giving weights to the error and control magnitude, respectively. I set Q=100* I_{N_p*3} ,R= I_{N_C*2} ,the error weight is 100 times the control magnitude weight, which prioritizes the control quality. ρ ε2is the relaxation factor that ensures that the equation has a best solution.

Generally, the loss function is converted into a standard quadratic programming equation to calculate the optimal output sequence of the controller to meet the design requirements. The standard quadratic programming equation is given below.

$$\frac{1}{2}X^THX+f^TX$$

Reduce the cost function to the form above:

$$J=(\psi*\eta(k)+\Theta*\Delta u(k))^2*Q+\left||\Delta u^2|\right|*R+\rho\varepsilon^2$$
$$=\Delta u^T(\Theta^T*\mathrm{Q}*\Theta+\mathrm{R})\Delta u+2\psi^TQ\Theta*\Delta\mathrm{u}+\rho\varepsilon^2$$

Set $X=\begin{bmatrix}\Delta u\\ \varepsilon\end{bmatrix}$, then we have

$$J=[\Delta u\ \varepsilon]\begin{bmatrix}\Theta^T\mathrm{Q}\Theta+\mathrm{R} & 0\\ 0 & \rho\end{bmatrix}\begin{bmatrix}\Delta u\\ \varepsilon\end{bmatrix}+[2E^TQ\Theta\quad 0]\begin{bmatrix}\Delta u\\ \varepsilon\end{bmatrix}=\frac{1}{2}X^THX+f^TX, E=\psi*\xi$$

E is the deviation of the actual trajectory from the reference trajectory in the predicted horizon. Next, the variable constraints of the quadratic programming equation are set.

$$\Delta u_{min}\le u(k+1)-u(k)\le\Delta u_{max}$$
$$u_{min}\le u(k)\le u_{max}$$

We convert these two inequations to the form below:

$$A*u<B$$

Which is:

$$\begin{cases}\Delta u(k)\le u_{max}-u(k-1)\\ -\Delta u(k)\le -u_{min}+u(k-1)\end{cases}$$

Then

$$A=\begin{bmatrix}L & 0\\ -L & 0\end{bmatrix},B=\begin{bmatrix}u_{max}-u(k-1)\\ -u_{min}+u(k-1)\end{bmatrix},L=\begin{bmatrix}1&\dots&0\\1&\ddots&0\\1&\dots&1\end{bmatrix}\otimes I_m$$

The lower and upper limits of Δu(k) is Δu_{min} and Δu_{max} respectively

Using QuadProg solver in MATLAB, we have the sequence of control quantities:

$$\Delta U=\begin{bmatrix}\Delta u(k)\\ \Delta u(k+1)\\ \vdots\\ \Delta u(k+N_c)\end{bmatrix}$$

Since updating the system state requires obtaining the controller output after each iteration, it is necessary to calculate the controller output value u(k) at time k+1 by the amount of change in the output.

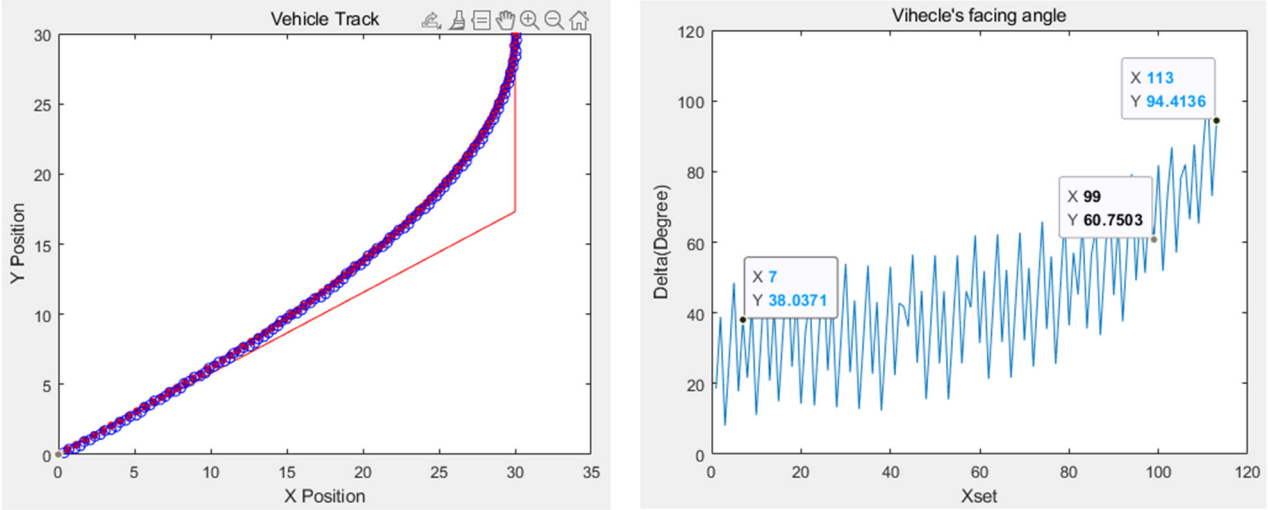
$$\Delta u(k) = \hat{u}(k) - \hat{u}(k-1) = u(k) - u_r(k) + [u(k-1) - u_r(k-1)]$$

So that

$$u(k) = \Delta u(k) + u_r(k) + [u(k-1) - u_r(k-1)]$$

At this point, the MPC controller design is complete, check the vehicle's actual trajectories.

We set the begin point is (0,0) and the initial facing angle is $\frac{\pi}{6}$, the target point's coordinates are (30,30) with 90 facing angle.



(A) Position (B) Facing Angle

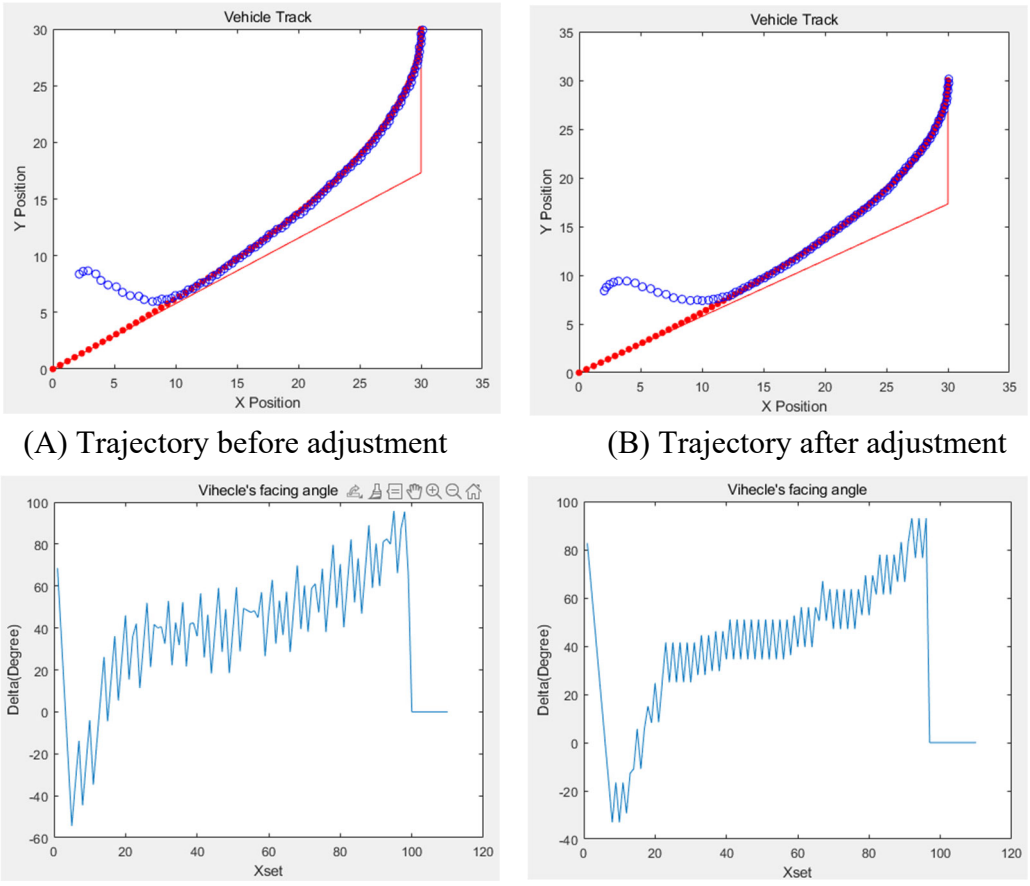
Fig. 2 Actual Trajectory

As shown in Figure 2(A) above, the red dot traces are the reference trajectories and the blue dot ones are the actual trajectories of the car. There are 110 parameter points in the preset trajectory and the same number of tracking trajectories. Thanks to the powerful prediction and speed of the MPC controller, the vehicle can pass the preset reference trajectory with almost zero error in a few seconds.

As shown in Fig. B above, the curves represent the change in the orientation of the carriage during the iterations. Due to the discretization of the model, it can be seen that the angle tends to oscillate upward, roughly from 30 degrees to the end state of 90 degrees. During the debugging process, it is found that the oscillation of the angle is always related to the maximum steering angle, so the preset maximum steering angle is adjusted to 30 degrees and the control variation constraint range is narrowed.

Then we set the initial coordinates to be 2,8 with the course angle of pi/3. Observe the change of controller adjustment speed and control equality before and after parameter adjustment.

As we can see in these four figures, after setting the maximum steering angle to 30 degrees, the trajectory of the vehicle is obviously smoother, but the speed of returning to the reference trajectory is also slower as compensation. Before the change, the swinging angles are very large, which is not friendly to the actual control conditions; After that is applied, the swinging angles become significantly smaller, which is more in line with the actual situations



(A) Trajectory before adjustment (B) Trajectory after adjustment

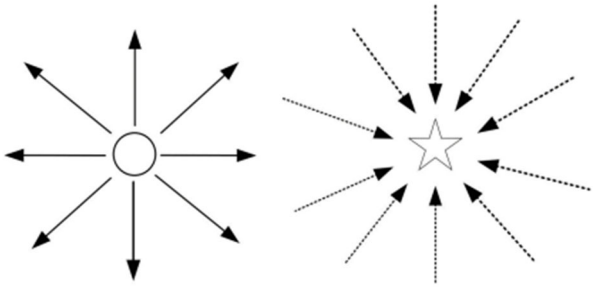
(C) Facing angle before adjustment (D) Facing angle after adjustment

Fig. 3 Control qualities comparison

5. Design of Obstacle Avoidance System

5.1 Artificial Potential Field

The artificial potential field method is a path planning algorithm widely used in the field of robotics and self-driving vehicles. The principle is to transform the motion of a vehicle in the real environment into the motion of an that in an artificially set abstract potential field, which consists of two major potential fields: gravitational force and repulsive force. The superposition of the gravitational potential field and the repulsive potential field is the combined potential field, and the car travels under the action of the combined potential field, and the direction of travel is the direction of potential energy decrease. The force of the car in the abstract potential field is shown in Figure 4 (the arrow represents the direction of force)



(A) Gravitational Field (B) Repulsive Field

Fig. 4 Virtual Gravity and Repulsion

The gravitational potential field is the attraction force of the target point to the vehicle, which is mainly determined by the distance between the car model and the target point. When the vehicle is near the target point, the distance is smaller and the gravitational potential energy is weaker; when it is near the starting point, the distance is larger and the gravitational potential energy is stronger, thus attracting the object to move toward the target position. The gravitational potential field at the target point has the smallest value, which is similar to the bottom of the mountain, and the gravitational potential field is most powerful at the starting point, which is similar to the top of the mountain. The gravitational function can be expressed as

$$U_{att}(q) = \frac{1}{2} k_{att} \rho^2(q, q_g)$$

where k_{att} is a positive constant that represents the gravitational potential field gain factor. $\rho^2(q, q_g)$ is the square of the object's Euclidean distance from the obstacle.

The repulsive potential field reflects the repulsive ability of the obstacle to the vehicle, which is determined by the distance between the obstacle and current position. When the car is near the obstacle, the distance is smaller and the repulsive potential energy is larger to induce it to avoid the obstacle; when the vehicle is far away from the obstacle, the distance is larger and the repulsive potential energy is smaller. When the distance is greater than the maximum influence range of the obstacle, the repulsive potential field will no longer play any role [8]. The expression of the repulsive potential field function is as follows.

$$\begin{cases} \frac{1}{2} k_{rep} \left[\frac{1}{\rho(q, q_g)} - \frac{1}{q_0} \right]^2, \rho(q, q_g) \le q_0 \\ 0, Others \end{cases}$$

where k_{rep} denotes the repulsive potential field gain factor and q_0 represents the artificially determined repulsive influence range.

When programming, the synthesis of forces requires the use of the synthesis extraction rule for vectors. A vector is divided by its norm to obtain a unit vector representing the direction.

$$\overrightarrow{F^i} = \frac{\vec{F}}{|\vec{F}|}$$

5.2 Algorithm optimization and verification

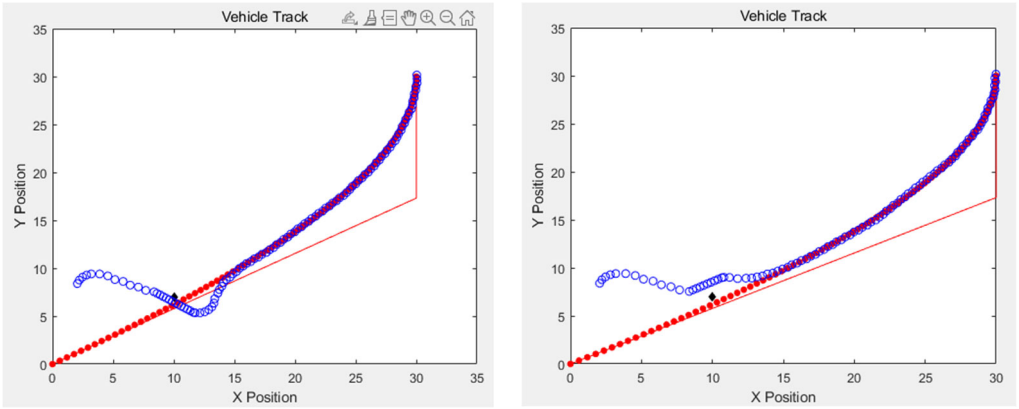
Although the APF algorithm for obstacle avoidance is simple to implement and easy to compute, if there are multiple obstacles in space, there is still a possibility of local minima and failure to reach the target point under certain circumstances [10], because the abstract gravitational and repulsive models may cancel each other out. This problem can still occur when there is only a single obstacle and the obstacle, the starting point and the cart are in the same line.

To solve this problem, a new kind of variable virtual target point is introduced in this paper. Assuming that the coordinates of the obstacle are (x_O, y_O), the coordinates of the vehicle are (x,y), and the coordinates of the virtual target point are $((x_D, y_D)$. The algorithm is as follows.

$$\begin{cases} (x_D, y_D) = (x_O + 30, y_O + 40) & \arctan\left(\frac{x_O - x}{y_O - y}\right) \le \varphi \\ (x_D, y_D) = (x_O + 30, y_O - 40) & \arctan\left(\frac{x_O - x}{y_O - y}\right) > \varphi \end{cases}$$

where φ is the heading angle of the vehicle at the current moment. The essence of the algorithm is to select different obstacle avoidance routes for different relative positions of the obstacles and the vehicle, and the inclusion of a changing gravitational source can avoid the vehicle from falling into the local optimum

comparisons can be shown in Fig.5:



(A) Before (B) After

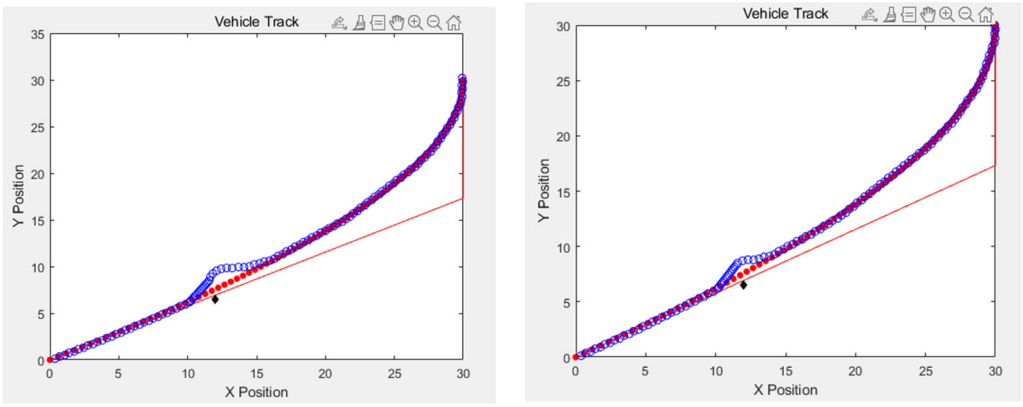
Fig. 5 Variable virtual target point algorithm

It can be seen that before applying the virtual target point algorithm, because there is only one direction, the trajectory seems to appear too close to the obstacle point when the angle is not perfect. And after optimization, the steering of obstacle avoidance is adjusted to reduce the chance of collision with the obstacle.

Considering that the artificial potential field method sometimes has a large impact on the traveling trajectory of the cart [12], if only the algorithm is used when encountering obstacles it will lead to a large steering angle of the vehicle, causing the control effect to become poor. The method of joint control of MPC and artificial potential field is introduced, and different weights are given to both controllers in different situations.

$$\begin{cases} U(k) = 0.3 * U_{MPC}(k) + U_{APF}(k), & \rho(q, q_g) \le q_0 \\ U(k) = U_{MPC}(k), & \rho(q, q_g) > q_0 \end{cases}$$

Comparisons can be shown in Fig.6, where initial coordinates are (0,0) with the course angle of $\frac{\pi}{6}$.



(A) Before (B) After

Fig. 6 Joint Control Algorithm

In the above figure (A), not only the adjustment deviation is obviously larger than that in figure (B), but also the articulation of MPC controller at the end of obstacle avoidance is obviously poor and there is a large steering acceleration. In Fig. (B), the regulation is small and can quickly and smoothly turn to the established track. The comparison shows that the weighting method has significantly improved the adjustment speed of attitude and the quality of obstacle avoidance.

6. Conclusions

This thesis makes comprehensive use of the advancement and rapidity of MPC controller and the efficiency of improved APF algorithm, the results design a multi-functional tracking + obstacle avoidance intelligent vehicle control system that can be applied in practice, which can ensure the accurate tracking of the route and the control of vehicle position. The obstacle avoidance system enables the vehicle to make reasonable avoidance actions for different obstacles, and the control can be returned to the MPC controller to continue tracking the path after the avoidance is finished.

References

[1] World Health Organization. Global status report on road safety 2018: Summary [R]. Switzerland: World Health Organization, 2018.

[2] Cui Yuding, Xiong Haojie, He Manchuan, Xue Renqiang. Autonomous active obstacle avoidance strategy based on Model Prediction Control[J/OL]. Agriculture Equipment & Vehicle Engineering,2023

[3] Yu Hui,Guo Baohe, Qin Zhichang. Dynamic obstacle avoidance algorithm for intelligent vehicles based on improved artificial potential fields [J]. Journal of Shandong University of Science and Technology,2021,35(4)

[4] J. Berberich, J. Köhler, M. A. Müller and F. Allgöwer, "Data-Driven Model Predictive Control With Stability and Robustness Guarantees," in IEEE Transactions on Automatic Control, vol. 66, no. 4, pp. 1702-1717, April 2021, doi: 10.1109/TAC.2020.3000182.

[5] Oussama Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots [J]. The International Journal of Robotics Research,1986, Vol.5, Issue 1

[6] Yu Yang, Xu Changzhao, Wang Zhao. Trajectory optimization study of driverless cars [J].Journal of Practical Automotive Technology,2023,48(11):49-56.DOI:10.16638/j.cnki.1671-7988.2023.011.009.

[7] N. Guo, B. Lenzo, X. Zhang, Y. Zou, R. Zhai and T. Zhang, "A Real-Time Nonlinear Model Predictive Controller for Yaw Motion Optimization of Distributed Drive Electric Vehicles," in IEEE Transactions on Vehicular Technology, vol. 69, no. 5, pp. 4935-4946, May 2020, doi: 10.1109/TVT.2020.2980169.

[8] Yu Zheng. Overtaking trajectory planning strategy for intelligent cars based on improved artificial potential field method[J]. Agriculture Equipment & Vehicle Engineering, 2021, 59(9): 64-68.

[9] Chao Shen, Yang Shi, Distributed implementation of nonlinear model predictive control for AUV trajectory tracking, Automatica, Volume 115, 2020

[10] Yan, X.; Jiang, D.; Miao, R.; Li, Y. Formation Control and Obstacle Avoidance Algorithm of a Multi-USV System Based on Virtual Structure and Artificial Potential Field. J. Mar. Sci. Eng. 2021, 9, 161. https://doi.org/10.3390/jmse9020161

[11] T. Dong, X. Chen, Y. Song, W. Ying and J. Fan, "Dynamic Artificial Potential Fields for Multi-User Redirected Walking," 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Atlanta, GA, USA, 2020, pp. 146-154, doi: 10.1109/VR46266.2020.00033.

[12] N. He, Y. Su, j. Guo, X. Fan, Z. Liu and B. Wang, "Dynamic path planning of mobile robot based on artificial potential field," 2020 International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI), Sanya, China, 2020, pp. 259-264, doi: 10.1109/ICHCI51889.2020.00063.