Proceedings of the 3rd Conference on Fully Actuated System Theory and Applications
May 10-12, 2024, Shenzhen, China

Trajectory Tracking of Differential Driven AGV Based on Kalman Filter and Model Predictive Control

Zhihao Xu[1], Yue Jiang[1], Kai Peng[2], Hongxia Wang[1,*],

1. College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266590, P. R. China
E-mail: 925441069@qq.com; whx1123@163.com; sdustjiangyue@163.com

2. School of Power and Energy, Northwestern Polytechnical University, Xi'an, 710072, P. R. China
E-mail: pengkai@nwpu.edu.cn

Abstract: Trajectory tracking is one of the key technologies of automated guided vehicles. However, the existing relevant research seldom considers the influence of noise on tracking accuracy in the real environment. Considering the automated guided vehicle model with additive noise, a new trajectory tracking algorithm is proposed in this paper. In the algorithm, the initial state and control are first given. Then state estimation and optimal control are implemented alternatively. The former is realized by the Kalman filter and the latter is resorted to model predictive control. The algorithm can effectively reduce the influence of noises on trajectory tracking. Finally, the proposed algorithm is evaluated by simulation experiments.

Key Words: Kalman filter, model predictive control, trajectory tracking, Automatic Guided Vehicle

2024 3rd Conference on Fully Actuated System Theory and Applications (FASTA) | 979-8-3503-7369-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/FASTA61401.2024.10595307

1 Introduction

The demand for modern automation equipment is increasing. As a flexible and intelligent autonomous navigation vehicle, AGV is crucial for optimizing production processes and improving production efficiency [1]. Trajectory tracking is the foundation of AGV, and it plays an important role in the study of AGV. Trajectory tracking not only ensures the stable operation of AGVs in complex environments but also improves the efficiency of their task execution. At present, many control methods have been applied to the trajectory tracking problem of AGV, including PID control, sliding mode control(SMC), and MPC [2]-[7].

The PID algorithm is relatively simple. The algorithm does not need the exact model of the system to realize the trajectory tracking function [8]. However, the parameter adjustment of a traditional PID controller is complicated, and the control parameters cannot be changed timely according to the change of state. This can lead to less-than-ideal tracking [9]. Therefore, researchers often combine PID with other algorithms to improve the control effect. For example, a PID controller based on an improved genetic Algorithm (IGA) was proposed by Yuan T et al. which adopted IGA to optimize PID parameters, thus improving tracking performance [10].

SMC is widely used in AGV systems for trajectory tracking because of its strong robustness and easy implementation [11]. A general rollover prevention control framework for high-speed wheeled mobile robots was proposed by Yan C et al., and an integral sliding mode wheel speed tracking controller was designed, which successfully realized rollover prevention in trajectory tracking and path following [12]. However, it is difficult to design a sliding mode control function and the system driven by SMC is susceptible to high-frequency perturbations [13].

MPC is an algorithm that obtains control inputs by solving constrained optimization problems [14]-[16]. The algorithm first predicts the output of the system, and then adjusts the control strategy according to the designed performance index, to realize the effective control of the system. Therefore, MPC has become one of the important methods to deal with trajectory tracking problems. The controller can guarantee the tracking accuracy and dynamic stability of the vehicle simultaneously. An algorithm combining model predictive control and delayed neural network was designed by Wang D et al. to solve the omnidirectional robot trajectory tracking problem under constraints [17]. A switched MPC framework was given by [18]. By selecting appropriate vehicle models and supervision schemes, the balance between MPC performance and computational cost was achieved, and accurate path tracking was realized.

However, the above trajectory tracking algorithms do not account for high-frequency noise in a real environment which can affect the accuracy of tracking. Therefore, this paper presents a new algorithm for tracking trajectories, taking into account the AGV model with additive Gaussian noise. The algorithm first initializes the state and control. The state and control after initialization are combined with the KF to estimate the state of the next moment. Then the optimal control at the next moment is solved according to the MPC and state estimated by the KF. The alternate execution of state estimation and control optimization reduces noise influence and improves trajectory tracking accuracy.

The rest of the paper is organized as follows: Section 2 presents the problem and kinematics equations of the system to be solved in this paper. Section 3 describes the detailed algorithm: the control strategy is adjusted by the method of MPC and KF alternating operation, and the trajectory tracking is realized. In section 4, the algorithm is verified by simulation experiments. Finally, the algorithm is summarized in Section 5. The trajectory tracking scheme of this paper is shown in Fig. 1.

This work was supported by the Original Exploratory Program Project of National Natural Science Foundation of China (62250056), the Joint Funds of the National Natural Science Foundation of China (U23A20325), the Major Basic Research of Natural Science Foundation of Shandong Province (ZR2021ZD14), and the High-level Talent Team Project of Qingdao West Coast New Area (RCTD-JC-2019-05).
*Corresponding Author.

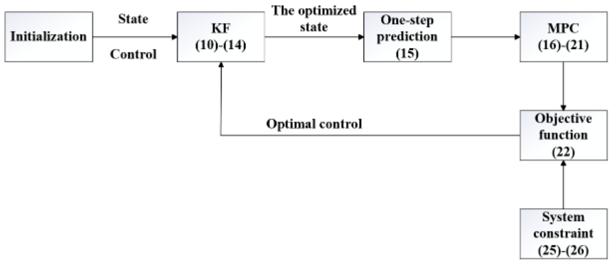

Fig. 1: The scheme of trajectory tracking

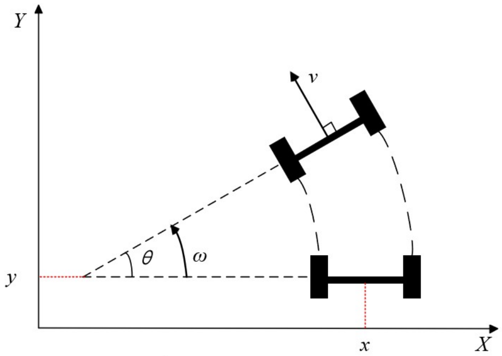

Fig. 2: Kinematic model for differential drive AGV

2 Problem Formulation

When the operation of AGV is disturbed by noise, the tracking will be biased. To deal with this problem, we consider a kinematic model with additive Gaussian noise.

Based on the differential drive AGV shown in Figure 2, the kinematic model established in this paper is as follows

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}cos\theta & 0\\ sin\theta & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix} + W, \tag{1}$$

where x and y represent the abscissa and ordinate of the geometric center of the AGV, θ is the heading angle of the AGV, v is the velocity of the AGV, ω is the angular velocity of the AGV, W denotes the Gaussian noise during AGV operation.

System (1) can be expressed in a concise form as

$$\dot{X} = f(X, u) + W, \tag{2}$$

where state variable $X = \begin{bmatrix}x & y & \theta\end{bmatrix}^T$ and control input $u = \begin{bmatrix}v & \omega\end{bmatrix}^T$.

The state equation of the reference trajectory is set

$$\dot{X}_r = f(X_r, u_r), \tag{3}$$

where state variable $X_r = \begin{bmatrix}x_r & y_r & \theta_r\end{bmatrix}^T$ and control input $u_r = \begin{bmatrix}v_r & \omega_r\end{bmatrix}^T$.

3 A New Trajectory Tracking Method

In order to reduce the influence of noise on trajectory tracking accuracy, an alternate algorithm of MPC and KF is proposed in this paper.

3.1 Linearization of the Error Model

The Taylor series expansion is performed at the reference trajectory point

$$\begin{aligned}\dot{X} =& f(X_r, u_r) + \frac{\partial f(X_r, u_r)}{\partial X}(X - X_r)\\ &+ \frac{\partial f(X_r, u_r)}{\partial u}(u - u_r) + W.\end{aligned} \tag{4}$$

The result of subtracting (3) from (4) is

$$\begin{aligned}\dot{\tilde{X}} &= \dot{X} - \dot{X}_r\\ &= \frac{\partial f(X_r, u_r)}{\partial X}\tilde{X} + \frac{\partial f(X_r, u_r)}{\partial u}\tilde{u} + W,\end{aligned} \tag{5}$$

where $\tilde{X} = X - X_r$, $\tilde{u} = u - u_r$.

Discretize (5) using the forward Euler method

$$\tilde{X}(k+1) = A(k)\tilde{X}(k) + B(k)\tilde{u}(k) + W(k), \tag{6}$$

$A(k)$ and $B(k)$ are given as

$$A(k) = \begin{bmatrix}1 & 0 & -v_r(k)sin\theta_r(k)T\\ 0 & 1 & -v_r(k)cos\theta_r(k)T\\ 0 & 0 & 1\end{bmatrix}, \tag{7}$$

$$B(k) = \begin{bmatrix}cos\theta_r(k)T & 0\\ sin\theta_r(k)T & 0\\ 0 & T\end{bmatrix}, \tag{8}$$

where $W(k)$ represents Gaussian white noise with a mean of 0 and a covariance matrix of R_w, T is the sampling period.

Then the observation equation of the system is

$$Z(k) = H\tilde{X}(k) + V(k), \tag{9}$$

where $Z(k)$ is the measured value, H is the measurement matrix of the system, and $V(k)$ represents Gaussian white noise with a mean of 0 and a covariance matrix of R_v. In addition, $W(k)$ and $V(k)$ are mutually independent.

3.2 Alternating Operation of MPC and KF

In this algorithm, we first initialize the state and control. The state and control after initialization are $\hat{X}(k-1)$ and $\tilde{u}(k-1)$, respectively. Then we use the Kalman filter to estimate the state at the next moment.

The state equation and observation equation of the system are (6) and (9) respectively.

During the prediction phase of the Kalman filter, the formulas for system state estimation and covariance estimation are given by

$$\bar{X}(k) = A(k-1)\hat{X}(k-1) + B(k-1)\tilde{u}(k-1), \tag{10}$$

$$\bar{P}(k) = A(k)\hat{P}(k-1)A(k)^T + R_w. \tag{11}$$

where $\bar{X}$ represents the prior state estimate, $\hat{X}$ represents the posterior state estimate, $\bar{P}$ represents the prior error covariance, $\hat{P}$ represents the updated error covariance.

Subsequently, during the correction phase of the Kalman filter, the Kalman gain, updated state estimation, and updated error covariance are given by

$$K(k) = \bar{P}(k)H^T[H\bar{P}(k)H^T + R_v]^{-1}, \tag{12}$$

$$\hat{X}(k) = \bar{X}(k) + K(k)[H\bar{X}(k) - Z(k)], \tag{13}$$

$$\hat{P}(k) = [I - K(k)]\bar{P}(k), \tag{14}$$

where $\hat{X}(k)$ is the next moment state estimated by KF. According to the estimated state $\hat{X}(k)$ and MPC, we can solve the optimal control at the next moment.

In order to solve the optimal control using MPC, we need to make a one-step prediction of the estimated state

$$\bar{X}(k+1) = A(k)\hat{X}(k) + B(k)\tilde{u}(k). \tag{15}$$

According to equation (15), we construct a new state vector as follows

$$\xi(k) = \begin{bmatrix} \hat{X}(k) \\ \tilde{u}(k-1) \end{bmatrix}, \tag{16}$$

and we can establish a new state-space representation

$$\xi(k+1) = \tilde{A}(k)\xi(k) + \tilde{B}(k)\Delta u(k), \tag{17}$$

$$\eta(k) = \tilde{C}(k)\xi(k), \tag{18}$$

where

$$\tilde{A}(k) = \begin{bmatrix} A(k) & B(k) \\ 0 & I \end{bmatrix}, \tilde{B}(k) = \begin{bmatrix} B(k) \\ I \end{bmatrix}, \tilde{C}(k) = \begin{bmatrix} I & 0 \end{bmatrix},$$

$\Delta u(k)$ is the control increment at the moment k.And we define $\Delta u(k)$ as

$$\Delta u(k) = u(k) - u(k-1), \tag{19}$$

and we can obtain

$$\begin{aligned} \Delta u(k) &= (u(k) - u_r) - (u(k-1) - u_r) \\ &= \tilde{u}(k) - \tilde{u}(k-1). \end{aligned} \tag{20}$$

State prediction based on (17) yields the following equation, N_p is the forecast time horizon.

$$\begin{aligned}
\xi(k+1) =& \tilde{A}(k)\xi(k) + \tilde{B}(k)\Delta u(k) \\
\xi(k+2) =& \tilde{A}(k+1)\tilde{A}(k)\xi(k) \\
&+ \tilde{A}(k+1)\tilde{B}(k)\Delta u(k) \\
&+ \tilde{B}(k+1)\Delta u(k+1) \\
\xi(k+3) =& \tilde{A}(k+2)\tilde{A}(k+1)\tilde{A}(k)\xi(k) \\
&+ \tilde{A}(k+2)\tilde{A}(k+1)\tilde{B}(k)\Delta u(k) \\
&+ \tilde{A}(k+2)\tilde{B}(k+1)\Delta u(k+1) \\
&+ \tilde{B}(k+2)\Delta u(k+2) \\
&\vdots \\
\xi(k+N_p) =& \prod_{i=N_p-1}^{0} \tilde{A}(k+i)\xi(k) \\
&+ [\sum_{i=1}^{N_p-1} (\prod_{N_p-1}^{i} \tilde{A}(k+i))\tilde{B}(k+i-1) \\
&\times \Delta u(k+i-1)] + \tilde{B}(k+N_p-1) \\
&\times \Delta u(k+N_p-1).
\end{aligned}$$

According to (18), the new system output can be calculated as follows

$$\begin{aligned}
\eta(k+1) =& \tilde{C}(k)\xi(k+1) \\
=& \tilde{C}(k)\tilde{A}(k)\xi(k) + \tilde{C}(k)\tilde{B}(k)\Delta u(k) \\
\eta(k+2) =& \tilde{C}(k)\xi(k+2) \\
=& \tilde{C}(k)\tilde{A}(k+1)\tilde{A}(k)\xi(k) \\
&+ \tilde{C}(k)\tilde{A}(k+1)\tilde{B}(k)\Delta u(k) \\
&+ \tilde{C}(k)\tilde{B}(k)\Delta u(k+1) \\
\eta(k+3) =& \tilde{C}(k)\xi(k+3) \\
=& \tilde{C}(k)\tilde{A}(k+2)\tilde{A}(k+1)\tilde{A}(k)\xi(k) \\
&+ \tilde{C}(k)\tilde{A}(k+2)\tilde{A}(k+1)\tilde{B}(k)\Delta u(k) \\
&+ \tilde{C}(k)\tilde{A}(k+2)\tilde{B}(k+1)\Delta u(k+1) \\
&+ \tilde{C}(k)\tilde{B}(k+2)\Delta u(k+2) \\
&\vdots \\
\eta(k+N_p) =& \tilde{C}(k)\xi(k+N_p) \\
=& \prod_{i=N_p-1}^{0} \tilde{C}(k)\tilde{A}(k+i)\xi(k) \\
&+ [\sum_{i=1}^{N_p-1} (\prod_{N_p-1}^{i} \tilde{C}(k)\tilde{A}(k+i))\tilde{B}(k+i-1) \\
&\times \Delta u(k+i-1)] + \tilde{C}(k)\tilde{B}(k+N_p-1) \\
&\times \Delta u(k+N_p-1).
\end{aligned}$$

To make the relationship more explicit, we express the output of the system at future moments in the following form

$$Y(k) = \Psi(k)\xi(k) + \Theta(k)\Delta U(k), \tag{21}$$

where we set $A_1(k) = (\prod_{N_p-1}^{1} \tilde{A}(k+i))$, $A_2(k) = (\prod_{N_p-1}^{2} \tilde{A}(k+i))$,

$$\Theta(k) = \tilde{C}(k) \begin{bmatrix} \tilde{B}(k) & 0 & \cdots & 0 \\ \tilde{A}(k)\tilde{B}(k) & \tilde{B}(k+1) & \cdots & 0 \\ \cdots & \cdots & \ddots & \cdots \\ A_1(k)\tilde{B}(k) & A_2(k)\tilde{B}(k+1) & \cdots & \tilde{B}(k+N_p-1) \end{bmatrix},$$

$$\Delta U(k) = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \cdots \\ \Delta u(k+N_c-1) \end{bmatrix}, Y(k) = \begin{bmatrix} \eta(k+1) \\ \eta(k+2) \\ \cdots \\ \eta(k+N_p) \end{bmatrix},$$

$$\Psi(k) = \begin{bmatrix} \tilde{C}(k)\tilde{A}(k) \\ \tilde{C}(k)\tilde{A}(k+1)\tilde{A}(k) \\ \cdots \\ \prod_{i=N_p-1}^{0} \tilde{C}(k)\tilde{A}(k+i) \end{bmatrix}.$$

In the process of AGV trajectory tracking, we obtain the control sequence by solving the objective function, thus ensuring that the AGV tracks the reference trajectory as accurately as possible. We define the objective function as

$$J(k) = Y^T Q Y + \Delta U(k)^T R \Delta U(k). \tag{22}$$

Substituting (21) into (22) yields

$$J(k) = \Delta U(k)^T(\Theta(k)^T Q\Theta(k) + R)\Delta U(k) + 2(\Psi(k)\xi(k))^T Q\Theta(k)\Delta U(k) + E, \tag{23}$$

$$E = (\Psi(k)\xi(k))^T Q(\Psi(k)\xi(k)), \tag{24}$$

where Q and R are weighting matrices, E is a constant that can be neglected during the simplification process.

In the actual control system, we consider the constraints of control quantity and control increment to satisfy the requirements of AGV safety and stability. The constrained targets are given as

$$\begin{gathered} u_{min} \leq u(k+t) \leq u_{max}, \\ t = 0, 1, \cdots, N_c - 1, \end{gathered} \tag{25}$$

$$\begin{gathered} \Delta u_{min} \leq \Delta u(k+t) \leq \Delta u_{max}, \\ t = 0, 1, \cdots, N_c - 1, \end{gathered} \tag{26}$$

where u_{min} is the minimum control quantity, u_{max} is the maximum control quantity, Δu_{min} is the minimum control increment and Δu_{max} is the maximum control increment.

Since the solution of the objective function is the control increment, the constraint is written in the form of a control increment. In addition, we know the control quantity at the last moment. Therefore, the control quantity at the current moment can be expressed in the following form

$$u(k + t) = u(k + t - 1) + \Delta u(k + t). \tag{27}$$

And we set

$$U(k - 1) = 1_{N_c} \otimes u(k - 1), \tag{28}$$

$$\Upsilon = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \ddots & \cdots \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix}_{N_c \times N_c} \otimes I, \tag{29}$$

where 1_{N_c} is an N_c-dimensional column vector with all elements equal to 1, $\otimes$ is Kronecker Product.

Therefore, (25) and (26) can be rewritten as

$$U_{min} \leq \Upsilon\Delta U(k) + U(k - 1) \leq U_{max}, \tag{30}$$

$$\Delta U_{min} \leq \Delta U(k) \leq \Delta U_{max}, \tag{31}$$

where

$$U_{min} = 1_{N_c} \otimes u_{min}, \tag{32}$$

$$U_{max} = 1_{N_c} \otimes u_{max}, \tag{33}$$

$$\Delta U_{min} = 1_{N_c} \otimes \Delta u_{min}, \tag{34}$$

$$\Delta U_{max} = 1_{N_c} \otimes \Delta u_{max}. \tag{35}$$

We transform the objective function (22) into a QP problem and solve it in conjunction with constraints,

$$J = \Delta U(k)^T H(k)\Delta U(k) + g(k)\Delta U(k), \tag{36}$$

$$\begin{gathered} s.t. U_{min} \leq \Upsilon\Delta U(k) + U(k - 1) \leq U_{max}, \\ \Delta U_{min} \leq \Delta U(k) \leq \Delta U_{max}, \end{gathered}$$

with

$$H(k) = \Theta(k)^T Q\Theta(k) + R, \tag{37}$$

$$g(k) = 2(\Psi(k)\xi(k))^T Q\Theta(k). \tag{38}$$

By solving (36), a series of control increments in the control horizon can be obtained

$$\Delta U^*(k) = \begin{bmatrix}\Delta u^*(k) & \cdots & \Delta u^*(k + N_p - 1)\end{bmatrix}^T. \tag{39}$$

Apply the first element $\Delta u^*(k)$ in (38) to the system. We can get the optimal control at k moment, that is

$$u^*(k) = u(k - 1) + \Delta u^*(k). \tag{40}$$

In addition, we can also obtain the optimal $\tilde{u}^*$ at the moment k, that is

$$\tilde{u}^*(k) = \tilde{u}(k - 1) + \Delta u^*(k). \tag{41}$$

and we can use $\tilde{u}^*(k)$ and the Kalman filter to estimate the state at the next moment.

Algorithm 1 Alternating Operation of MPC and KF

1: Define N as the maximum number of iterations
2: Initialize $\hat{X}(k-1)$, $\tilde{u}(k-1)$
3: **while** $k \leq N$ **do**
4: Estimate $\hat{X}(k)$ based on KF
5: Solving MPC based on $\hat{X}(k)$ yields $\Delta u^*(k)$
6: $u^*(k) \leftarrow u^*(k-1) + \Delta u^*(k)$
7: $k \leftarrow k + 1$
8: **end while**

4 Simulation

To demonstrate the effectiveness of the algorithm, we consider tracking different trajectories and compare the algorithm with the noise-disturbed MPC algorithm. We use MATLAB to conduct simulation experiments.

4.1 Tracking a Circular Trajectory

In the experiment, we set the sampling period T = 0.1, reference line velocity $v_r = 2$, reference angular velocity $\omega_r = 0.033\pi$, the initial state $X = \begin{bmatrix}-0.2 & 1 & 0.016\pi\end{bmatrix}^T$. We consider tracking the circular trajectory in Fig.3.

Fig.3 shows the tracking results of the two algorithms for the circular trajectory. It can be seen from the figure that under the influence of noise, the algorithm designed in this paper can successfully track the circular trajectory and is closer to the reference trajectory than the MPC algorithm. Fig.4 and Fig.5 show the circular trajectory tracking errors of the two algorithms under the influence of noise. It can be seen that the trajectory tracking error of this algorithm is always kept in a small range, and the error of this algorithm is smaller than that of the MPC algorithm.

4.2 Tracking a Straight Trajectory

In the experiment, we set the sampling period T = 0.1, reference line velocity $v_r = 2.5$, reference angular velocity $\omega_r = 0$, the initial state $X = \begin{bmatrix}0 & 0.5 & 0.016\pi\end{bmatrix}^T$.We consider tracking the straight trajectory in Fig.6.

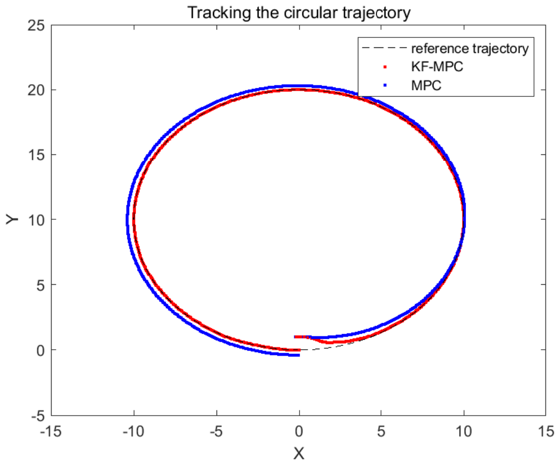

Fig. 3: Circular trajectory tracking

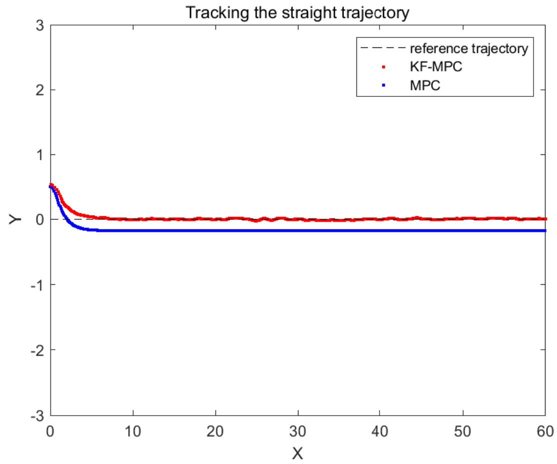

Fig. 6: Straight trajectory tracking

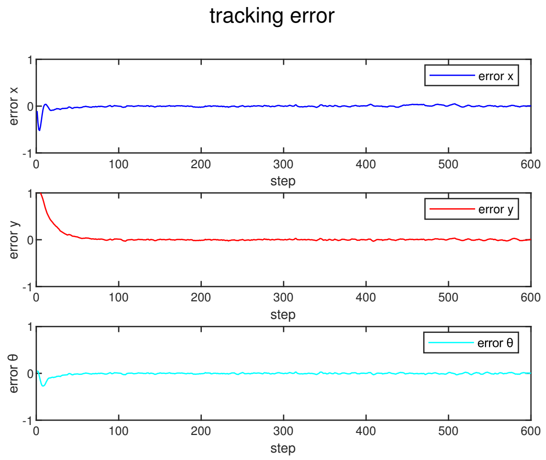

Fig. 4: Circular trajectory tracking error based on KF-MPC

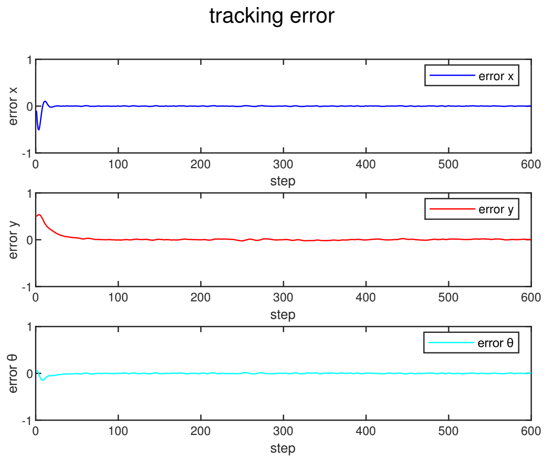

Fig. 7: Straight trajectory tracking error based on KF-MPC

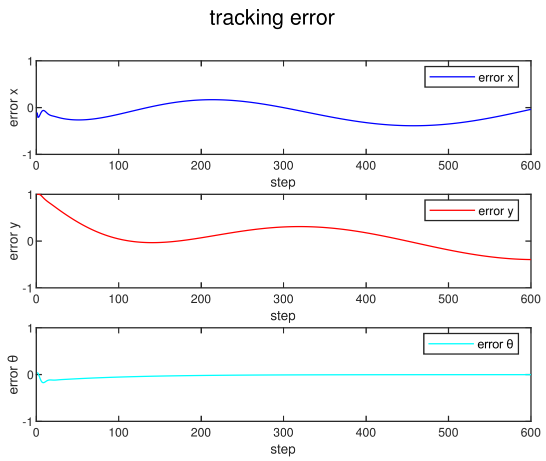

Fig. 5: Circular trajectory tracking error based on MPC

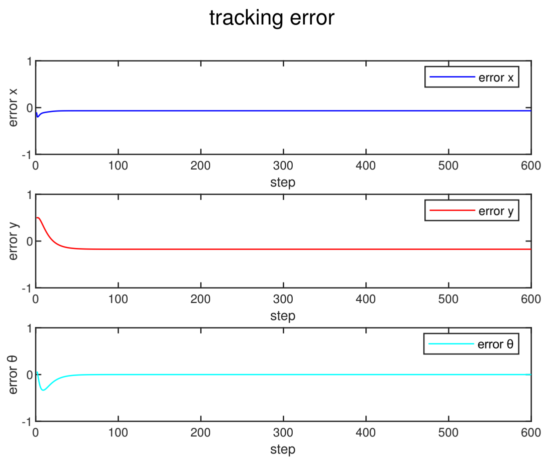

Fig. 8: Straight trajectory tracking error based on MPC

As can be seen from Fig.6, due to the influence of noise, only MPC can not accurately track the linear trajectory. However, the algorithm designed in this paper can effectively reduce the influence of noise, so that AGV can complete the tracking more accurately. Fig.7 and Fig.8 show the tracking errors of the two algorithms. It can be seen that the accuracy of the proposed algorithm is higher.

In summary, it can be seen from the above two sets of experiments that the algorithm can reduce the influence of noise on AGV, enable it to accurately track the reference trajectory, and control the error within a small range. Simulation results further prove the effectiveness of the proposed algorithm.

5 Conclusions

In this paper, an alternate algorithm of Kalman filter and model predictive control is proposed to reduce the influence of noise and improve the accuracy of trajectory tracking for differential drive AGV systems with additive Gaussian noise. The Kalman filter part of the algorithm can estimate the state and reduce the interference of noise. In the model predictive control part of the algorithm, we can get the optimal control at the current moment. In addition, the effectiveness of the proposed algorithm can be evaluated by the experimental results of tracking different trajectories. In conclusion, the algorithm achieves good results in trajectory tracking and provides an effective control strategy for practical application.

References

[1] H. Rashidi, F. Matinfar, and F. A. Parand. Automated guided vehicles - a review on applications, problem modeling and solutions. *Tarrahan Parseh Transportation Research Institute*, (3), 2021.

[2] H. Khodadadi and H. Ghadiri. Self-tuning pid controller design using fuzzy logic for half car active suspension system. *International Journal of Dynamics and Control*, 2016.

[3] S. T. Kao, W. J. Chiou, and M. T. Ho. Integral sliding mode control for trajectory tracking control of an omnidirectional mobile robot. *IEEE*, 2011.

[4] B. Moudoud, H. Aissaoui, and M. Diany. Fuzzy adaptive sliding mode controller for electrically driven wheeled mobile robot for trajectory tracking task. *Journal of Control and Decision*, pages 1–9, 2021.

[5] D. Chu, H. Li, C. Zhao, and T. Zhou. Trajectory tracking of autonomous vehicle based on model predictive control with pid feedback. *IEEE Transactions on Intelligent Transportation Systems*, 24(2):2239–2250, 2023.

[6] Z. Zuo, X. Yang, Z. Li, Y. Wang, Q. Han, L. Wang, and X. Luo. Mpc-based cooperative control strategy of path planning and trajectory tracking for intelligent vehicles. *IEEE Transactions on Intelligent Vehicles*, 6(3):513–522, 2021.

[7] H. Pang, N. Liu, C. Hu, and Z. Xu. A practical trajectory tracking control of autonomous vehicles using linear time-varying mpc method. *Proceedings of the Institution of Mechanical Engineers, Part D. Journal of Automobile Engineering*, (4):236, 2022.

[8] R. Marino, S. Scalzi, and M. Netto. Nested pid steering control for lane keeping in autonomous vehicles. *Control Engineering Practice*, 19(12):1459–1467, 2011.

[9] W. Farag. Complex trajectory tracking using pid control for autonomous driving. *International Journal of Intelligent Transportation Systems Research*, 18(2), 2019.

[10] T. Yuan, G. Guo, B. Du, Z. Zhao, and W. Xu. The adaptive sliding mode control using improved genetic algorithm tuning pid controller for the planetary rover. *Aircraft Engineering and Aerospace Technology*, 2021.

[11] T. Truong, A. T. Vo, and H. Kang. A backstepping global fast terminal sliding mode control for trajectory tracking control of industrial robotic manipulators. *IEEE Access*, 9:31921–31931, 2021.

[12] C. Yan, K. Shao, X. Wang, J. Zheng, and B. Liang. Reference governor-based control for active rollover avoidance of mobile robots. *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 429–435, 2021.

[13] X. Zhou and L. He. Sliding mode control for trajectory tracking of agv based on improved fast stationary power approach law. *Journal of Physics: Conference Series*, 2417(1):12–32, 2022.

[14] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[15] K. Zhang, Q. Sun, and Y. Shi. Trajectory tracking control of autonomous ground vehicles using adaptive learning mpc. *IEEE Transactions on Neural Networks and Learning Systems*, 32(12):5554–5564, 2021.

[16] Z. Zhao, H. Liu, H. Chen, J. Hu, and H. Guo. Kinematics-aware model predictive control for autonomous high-speed tracked vehicles under the off-road conditions. *Mechanical Systems and Signal Processing*, 123:333–350, 2019.

[17] D. Wang, W. Wei, Y. Yeboah, Y. Li, and Y. Gao. A robust model predictive control strategy for trajectory tracking of omni-directional mobile robots. *Journal of Intelligent and Robotic Systems*, 98:439–453, 2020.

[18] M. Rokonuzzaman, N. Mohajer, and S. Nahavandi. Effective adoption of vehicle models for autonomous vehicle path tracking: a switched mpc approach. *Vehicle system dynamics*, 61(5):1236–1259, 2023.