Academic Journal of Science and Technology
ISSN: 2771-3032 | Vol. 6, No. 2, 2023

Tracking Control of Wheeled Mobile Robot Based on MMPC

Jiabao Xu[1, 2, *], Guoliang Zhang[1, 2]

[1] School of Automation and Information Engineering, Sichuan University of Science & Engineering, Yibin 644000, China
[2] Artificial Intelligence Key Laboratory of Sichuan Province, Yibin 644000, China
[*] Corresponding author: Jiabao Xu (Email: 1248217710@qq.com)

Abstract: In order to solve the problems of low tracking accuracy and large steady-state error in track tracking of wheeled mobile robots, a MPC control scheme combining feedforward and feedback control is designed on the basis of Traditional Model Predictive Control (TMPC). First, when establishing the WMR kinematic error model, the linearization error is retained, and the control input and incremental constraints are considered. Secondly, a feedforward controller is designed for a given desired trajectory of wheeled mobile robot; At the same time, the multi constraint model predictive control (MMPC) strategy with speed optimization is added to design the feedback controller. Then, the stability of the designed MMPC controller is analyzed based on Lyapunov theory. Finally, MMPC control scheme is compared with TMPC control scheme and PID control scheme. The results show that compared with TMPC control scheme, MMPC control scheme reduces the lateral error by 17.87%, the longitudinal error by 17.12%, and the heading angle error by 9.81%; Compared with the PID control scheme, the lateral error is reduced by 20.82%, the longitudinal error is reduced by 23.26%, and the heading angle error is reduced by 5.87%.

Keywords: Multi constraint model predictive control, Wheeled mobile robot, Track tracking, Stability.

1. Introduction

In recent years, the wide application of wheeled mobile robots in various fields has attracted great attention of scholars [1-2]. The WMR control tasks are mainly divided into path planning [3] and trajectory tracking. In order to realize WMR trajectory tracking, many scholars have proposed different control methods. Literature [4] designed a synovial film controller based on the WMR kinematics model, even if the WMR has a large initial error in the control, it can get better control. Literature [5] considered WMR parameters and non-parameters two kinds of uncertainty, combined with adaptive control and neural network methods, to solve the mobile robot center of gravity offset, model uncertainty and other factors. Literature [6] studied the trajectory tracking control of WMR on uneven ground, and designed a double closed-loop strategy; the external disturbance was estimated by the inner loop, and the expected speed was output by the outer loop, which realized the smooth turning of WMR. Literature [7] uses the enhanced variable structure based on the synovium to track the trajectory of WMR, which improves the robot's control ability, reduces the attitude error, and the linear velocity and angular velocity are controllable.

Since WMR is a non-integrity constrained system, model predictive control has a feed-forward-feedback structure, which can handle nonlinear input-output systems with constraints [8-9]. Literature [10] proposes an MPC control scheme for nonholonomic constrained WMR, and solves linear MPC by using quadratic programming for continuous linearization of the WMR error model. Literature [11] developed a robust predictive control based on neural network optimization to stabilize WMR. Experiments under various dynamic conditions proved that the proposed scheme can improve the computational efficiency of MPC. Literature [12] realized the local path tracking of the robot based on the control scheme combining PID control and MPC.

Based on the above analysis, when the traditional model predictive control scheme is applied to WMR trajectory tracking, there will still be problems of low tracking accuracy and large steady-state error. On the one hand, these TMPC-based control schemes do not consider the error caused by the control increment limit and linearization, so the controller is designed through the control increment of WMR, so as to improve the WMR trajectory tracking accuracy. On the other hand, in WMR, the control quantity also has value constraints in the time domain, such as speed constraints, etc. These constraints are unavoidable, so designing a multi-constraint MPC controller is a problem worth studying.

Considering the above problems, this paper proposes a multi-constraint model predictive control scheme with a combination of feedforward and feedback control on the basis of traditional model predictive control. First, the kinematic error model of WMR is linearized and high-order items are retained, while considering the WMR control input and its incremental constraints; second, the feedforward controller is designed by calculating the reference control input through the expected trajectory; then, in the feedback The speed optimization strategy is added to the control, the control incremental state space equation is used as the prediction model, the objective function is designed under multiple constraints, and the constraint optimization problem in the finite time domain is defined to obtain the MMPC trajectory tracking controller; then, the designed MMPC control Finally, the simulation and physical experiments verify that the MMPC control scheme has better performance than the TMPC control scheme and the PID control scheme in trajectory tracking.

2. Problem Description

WMR is a typical underactuated system with nonintegrity constraints. The kinematics model of the wheeled mobile robot is shown in Figure 1.

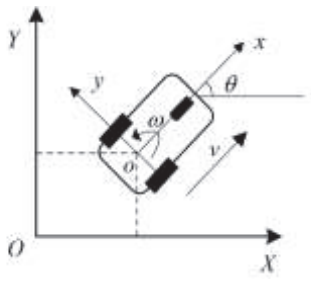


Figure 1. WMR Kinematic Model

In Figure 1, there are two Cartesian coordinate systems, $\{X,O,Y\}$ is the global coordinate system fixed on the ground and $\{x,o,y\}$ is the local coordinate system on the wheeled mobile robot. θ is the angle between the forward direction and X, and $[v,\omega]$ are the translational and rotational velocities of the WMR respectively. It is assumed that the wheel is in a pure rolling motion without sliding in the plane. At the instant of motion, the velocity of the WMR is directed towards the main axis of the robot. The slave wheel acts as a support during the motion and its influence can be neglected[13]. From the non-completeness constraint of the $\dot{x}(t)\sin\theta(t)-\dot{y}(t)\cos\theta(t)=0$, the kinematic equation can be obtained as follows

$$\dot{\chi}(t)=f(\chi(t),u(t))=\begin{bmatrix}\cos\theta(t) & 0\\ \sin\theta(t) & 0\\ 0 & 1\end{bmatrix}u(t) \tag{1}$$

In equation (1), $\boldsymbol{\chi}(t)=[x(t)\ y(t)\ \theta(t)]^T$ represents the position and orientation of the WMR in the global coordinate system $\{X,O,Y\}$ and $\boldsymbol{u}(t)=[v(t)\ \omega(t)]^T$ represents the control input vector where the linear velocity is $v(t)$ velocity is $\omega(t)$。

A state tracking error vector $\boldsymbol{\chi}_e(t)=[x_e(t)\ y_e(t)\ \theta_e(t)]^T$ is defined in the local coordinate system $\{x,o,y\}$, which is expanded in a Taylor series at the reference trajectory point $(\boldsymbol{\chi}_r(t),\boldsymbol{u}_r(t))$ to obtain a wheeled mobile robot error model, which is represented as follows:

$$\begin{aligned}\dot{\boldsymbol{\chi}}_e(t)=&\begin{bmatrix}0 & 0 & -v_r\sin\theta_r(t)\\ 0 & 0 & v_r\cos\theta_r(t)\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x(t)-x_r(t)\\ y(t)-y_r(t)\\ \theta(t)-\theta_r(t)\end{bmatrix}\\ &+\begin{bmatrix}\cos\theta_r(t) & 0\\ \sin\theta_r(t) & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}v(t)-v_r(t)\\ \omega(t)-\omega_r(t)\end{bmatrix}+\boldsymbol{d}(t)\end{aligned} \tag{2}$$

In equation (2), $\boldsymbol{d}(t)$ is the higher order term of the Taylor expansion of $\dot{\boldsymbol{\chi}}_e(t)$. Considering the operational safety of the wheeled mobile robot, both need to keep v and ω within a certain range, with the following constraints on the controller:

$$\begin{cases} v_{\min}\le v\le v_{\max}\\ \omega_{\min}\le\omega\le\omega_{\max}\\ \Delta v_{\min}\le\Delta v\le\Delta v_{\max}\\ \Delta\omega_{\min}\le\Delta\omega\le\Delta\omega_{\max}\end{cases} \tag{3}$$

The maximum linear and angular velocities are $v_{\max}$ and $\omega_{\max}$, respectively, and the minimum linear and angular velocities are $v_{\min}$ and $\omega_{\min}$ respectively. For each control cycle, there are constraints on the linear velocity increment Δv and the angular velocity increment $\Delta\omega$.

The trajectory tracking problem studied in this paper is described as follows: design the controller to output suitable v and ω, such that the tracking error of the wheeled mobile robot converges asymptotically, which can be expressed as the asymptotic convergence of the modes of $[x_e,y_e,\theta_e]^T$ to zero[14], i.e. $\lim_{t\to\infty}\left\|[x_e,y_e,\theta_e]^T\right\|=0$.

3. Control Scheme Design

3.1. General design of the control scheme

The overall design of the control scheme is shown in Figure 2:

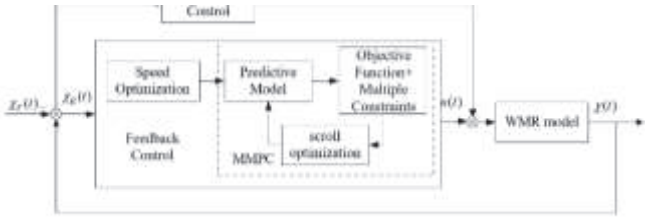


Figure 2. System control block diagram

The control input acting on the WMR system is $\boldsymbol{u}(t)$, and is expressed as

$$\boldsymbol{u}(t)=\boldsymbol{u}_r(t)+\tilde{\boldsymbol{u}}(t) \tag{4}$$

$\boldsymbol{u}_r(t)$ and $\tilde{\boldsymbol{u}}(t)$ are the feed-forward control output and the feedback control output at t respectively.

The feed-forward controller uses the desired trajectory at the current moment to calculate the reference control input. The feedback controller first optimizes the reference speed from the previous moment and then predicts the control input for the current moment via the MMPC controller. Finally the system applies the results of the two controllers together to the WMR, thus completing the trajectory tracking control.

3.2. Feedforward controller design

For a given reference trajectory $(x_r(t),y_r(t))$, the feedforward control input is $\boldsymbol{u}_r(t)=[v_r(t),\omega_r(t)]^T$. When the wheeled mobile robot is tracking the reference trajectory, the linear velocity $v_r(t)$ is calculated as follows:

$$v_r(t)=\pm\frac{\sqrt{(x_r(t)-x_r(t-1))^2+(y_r(t)-y_r(t-1))^2}}{t} \tag{5}$$

The linear velocity is positive when the robot is moving in the positive direction of the trajectory and negative vice versa. The heading angle at each moment on the reference trajectory is:

$$\theta_r(t) = \arctan\left(\frac{y_r(t) - y_r(t-1)}{x_r(t) - x_r(t-1)}\right) + \varepsilon\pi \tag{6}$$

In equation (6), ε is the reference direction, $\varepsilon = 0$ represents the positive direction along the reference trajectory and $\varepsilon = 1$ represents the reverse direction along the reference. The angular velocity is calculated as follows:

$$\omega_r(t) = \frac{\theta_r(t) - \theta_r(t-1)}{t} \tag{7}$$

The feedforward control law $\boldsymbol{u}_r(t)$ can be obtained by deriving the control input $v_r(t), \omega_r(t)$.

3.3. Speed optimization strategy design

Variations in speed during the operation of a wheeled robot affect the lateral error. In order to make the error between the actual trajectory and the reference trajectory during the trajectory tracking of the WMR smaller, specifically in the case of possible large deviations at turns, the speed is optimally adjusted.

The velocity optimization strategy consists of two main aspects, on the one hand, the system calculates the curvature of the WMR trajectory at the next moment based on the current reference trajectory, as shown in Figure 3. On the other hand, is the design of a function between the reference velocity and the curvature of the trajectory.

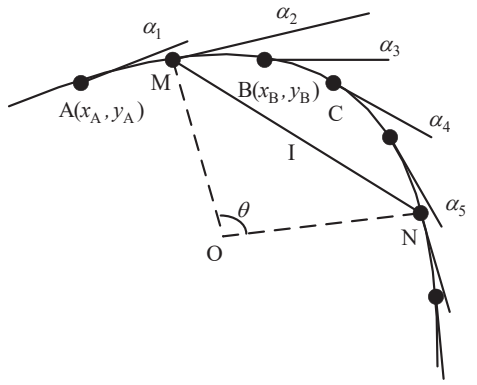


Figure 3. Curve Curvature Diagram

In Figure 3, O is the center point of the curve. Using the point M as an example, the formula for calculating its angle α_2 is as follows:

$$\alpha_2 = \cos^{-1}\left(\frac{(x_M - x_A)(x_B - x_M) + (y_M - y_A)(y_B - y_M)}{\sqrt{(x_M - x_A)^2 + (y_M - y_A)^2} \times \sqrt{(x_B - x_M)^2 + (y_B - y_M)^2}}\right) \tag{8}$$

If the calculated angle α_2 is greater than the set standard angle α_0, then point A is determined to be the starting point of the curve, and a new path point M is selected as the starting point and again determined according to equation (8) until the calculated angle appears to be less than the standard angle. The radius of curvature of the curve is solved by calculating the coordinates of the start and end points of the curve R_c:

$$R_c = \sqrt{(x_M - x_O)^2 + (y_M - y_O)^2} \tag{9}$$

The chord length of the curve I_c is:

$$I_c = \sqrt{(x_M - x_N)^2 + (y_M - y_N)^2} \tag{10}$$

Corresponding central angle of the curve θ_c:

$$\theta_c = 2\sin^{-1}\left(\frac{R_c}{2I_c}\right) \tag{11}$$

The final calculated curvature of the curve K is:

$$K = \frac{\theta_c}{L_c} \tag{12}$$

In equation (12), L_c is the length of the curved arc, and considering the WMR in a two-dimensional plane, the speed of the mobile robot needs to be limited. Depending on the curvature of the trajectory the WMR runs on, its speed $\tilde{v}(t)$ is set to

$$\tilde{v}(t) = \frac{v_r(t)}{\sqrt{K}} \tag{13}$$

3.4. MMPC controller design

For the feedback controller, the multi-constraint model predictive control strategy is designed by taking the sampling time T, , approximating the discretization of equation (2) using the first-order difference quotient method, and obtaining the expressions for the rewritten discrete state space and output $y(k)$ as follows:

$$\begin{cases} \boldsymbol{\chi}_e(k+1) = A\boldsymbol{\chi}_e(k) + B\tilde{\boldsymbol{u}}(k) + \boldsymbol{d}(k) \\ \boldsymbol{y}(k) = C\boldsymbol{\chi}_e(k) \end{cases} \tag{14}$$

In equation (14), $\boldsymbol{\chi}_e(k)$ is a state variable, $\tilde{\boldsymbol{u}}(k)$ is a control input variable, $\boldsymbol{y}(k)$ is a control output variable, $\boldsymbol{d}(k)$ is a measurable error variable, and

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & -v_r \sin\theta_r T \\ 0 & 1 & v_r \cos\theta_r T \\ 0 & 0 & 1 \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} \cos\theta_r T & 0 \\ \sin\theta_r T & 0 \\ 0 & T \end{bmatrix}, \boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{15}$$

It is assumed that the system state variables are fully measurable and that the system output is fully controllable. To eliminate static errors, the following incremental variables are defined for equation (8):

$$\begin{aligned} \Delta\boldsymbol{\chi}_e(k) &= \boldsymbol{\chi}_e(k) - \boldsymbol{\chi}_e(k-1) \\ \Delta\tilde{\boldsymbol{u}}(k) &= \tilde{\boldsymbol{u}}(k) - \tilde{\boldsymbol{u}}(k-1) \\ \Delta\boldsymbol{d}(k) &= \boldsymbol{d}(k) - \boldsymbol{d}(k-1) \end{aligned} \tag{16}$$

Where $\Delta\boldsymbol{\chi}_e$ is the state increment and $\Delta\tilde{\boldsymbol{u}}$ is the control increment. Bringing equation (16) into equation (14), the incremental model is obtained as

$$\begin{cases}\boldsymbol{\chi}_e(k+1)=(\boldsymbol{A}+\boldsymbol{I})\boldsymbol{\chi}_e(k)-\boldsymbol{A}\boldsymbol{\chi}_e(k-1)+\boldsymbol{B}\Delta\tilde{\boldsymbol{u}}(k)+\Delta\boldsymbol{d}(k)\\ \boldsymbol{y}(k)=\boldsymbol{C}\chi(k)\end{cases} \tag{17}$$

In equation (17), $\boldsymbol{I}$ is the unit matrix. To design the MMPC controller, the future dynamics of the system is first predicted based on model (11) using the latest value of $\boldsymbol{\chi}_e(k)$ as the initial condition. Set the prediction time domain to p, the control time domain to m, and $m\le p$. To calculate the prediction equations for the MMPC, assume that the control quantities in the control time domain are constant. For the current moment k, define the p step predicted control output quantities as follows:

$$\boldsymbol{Y}_p(k+1\,|\,k)=[\boldsymbol{y}(k+1\,|\,k)\ldots\boldsymbol{y}(k+p\,|\,k)]^T_{p\times1} \tag{18}$$

$\Delta\boldsymbol{U}(k)$ is a sequence of control quantity increments as independent variables in a constrained optimization problem defined as

$$\Delta\boldsymbol{U}(k)=[\Delta\tilde{\boldsymbol{u}}(k)\quad\cdots\quad\Delta\tilde{\boldsymbol{u}}(k+m-1)]^T_{m\times1} \tag{19}$$

The number of vectors in the matrix is indicated under the matrix in Eqs. (18) and (19) and is not necessarily the dimension of the matrix. The output predicted for the next p steps of the system can be calculated from the following prediction equation:

$$\boldsymbol{Y}_p(k+1\,|\,k)=\boldsymbol{S}_{x1}x(k)-\boldsymbol{S}_{x2}x(k-1)+\boldsymbol{S}_u\Delta\boldsymbol{U}(k)+\boldsymbol{S}_d\Delta\boldsymbol{d}(k) \tag{20}$$

in equation (20),

$$\boldsymbol{S}_{x1}=\begin{bmatrix}\boldsymbol{C}_c(\boldsymbol{A}+\boldsymbol{I})\\ \boldsymbol{C}_c\sum_{i=0}^{2}\boldsymbol{A}^i\\ \vdots\\ \boldsymbol{C}_c\sum_{i=0}^{p}\boldsymbol{A}^i\end{bmatrix},\boldsymbol{S}_{x2}=\begin{bmatrix}\boldsymbol{C}_c\boldsymbol{A}\\ \boldsymbol{C}_c\sum_{i=1}^{2}\boldsymbol{A}^i\\ \vdots\\ \boldsymbol{C}_c\sum_{i=1}^{p}\boldsymbol{A}^i\end{bmatrix},\boldsymbol{S}_d=\begin{bmatrix}\boldsymbol{C}_c\\ \boldsymbol{C}_c\sum_{i=1}^{2}\boldsymbol{A}^{i-1}\\ \vdots\\ \boldsymbol{C}_c\sum_{i=1}^{p}\boldsymbol{A}^{i-1}\end{bmatrix}$$

$$\boldsymbol{S}_u=\begin{bmatrix}\boldsymbol{C}_c\boldsymbol{B} & 0 & \cdots & 0\\ \boldsymbol{C}_c(\boldsymbol{A}+\boldsymbol{I})\boldsymbol{B} & \boldsymbol{C}_c\boldsymbol{B} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ \boldsymbol{C}_c(\sum_{i=1}^{m-1}\boldsymbol{A}^i)\boldsymbol{B} & \boldsymbol{C}_c(\sum_{i=1}^{m-2}\boldsymbol{A}^i)\boldsymbol{B} & \cdots & \boldsymbol{C}_c\boldsymbol{B}\\ \vdots & \vdots & \cdots & \vdots\\ \boldsymbol{C}_c(\sum_{i=1}^{p-1}\boldsymbol{A}^i)\boldsymbol{B} & \boldsymbol{C}_c(\sum_{i=1}^{p-2}\boldsymbol{A}^i)\boldsymbol{B} & \vdots & \boldsymbol{C}_c(\sum_{i=1}^{p-m}\boldsymbol{A}^i)\boldsymbol{B}\end{bmatrix}$$

Further, the magnitude of the control movements of our wheeled mobile robot must not be too large, for defining the following objective function as

$$\boldsymbol{J}=\left\|\boldsymbol{\Gamma}_y(\boldsymbol{Y}_p(k+1\,|\,k))-\boldsymbol{R}(k+1)\right\|^2+\left\|\boldsymbol{\Gamma}_u\Delta\boldsymbol{U}(k)\right\|^2 \tag{21}$$

In equation (21), $\boldsymbol{\Gamma}_y$ and $\boldsymbol{\Gamma}_u$ are the weighting matrices at the current prediction moment and the reference trajectory sequence input is

$$\boldsymbol{R}(k+1)=\begin{bmatrix}\boldsymbol{r}(k+1) & \boldsymbol{r}(k+2) & \ldots & \boldsymbol{r}(k+p)\end{bmatrix}^T_{p\times1} \tag{22}$$

Due to the presence of constraints, the MMPC optimization problem is solved by numerical methods. Since the objective function is quadratic, the constrained optimization problem is converted to a quadratic programming problem, defined

$$\boldsymbol{E}_p(k+1)=\boldsymbol{R}(k+1)-\boldsymbol{S}_{x1}x(k)+\boldsymbol{S}_{x2}(k-1)-\boldsymbol{S}_d\Delta\boldsymbol{d}(k) \tag{23}$$

Bringing the prediction equation (20) into the objective function (21), the objective function becomes

$$\begin{aligned}\boldsymbol{J}_k&=\left\|\boldsymbol{\Gamma}_y(\boldsymbol{S}_u\Delta\boldsymbol{U}(k))-\boldsymbol{E}(k+1)\right\|^2+\left\|\boldsymbol{\Gamma}_u\Delta\boldsymbol{U}(k)\right\|^2\\ &=\Delta\boldsymbol{U}(k)^T\boldsymbol{S}_u^T\boldsymbol{\Gamma}_y^T\boldsymbol{\Gamma}_y\boldsymbol{S}_u\Delta\boldsymbol{U}(k)+\Delta\boldsymbol{U}(k)^T\boldsymbol{\Gamma}_u^T\boldsymbol{\Gamma}_u\Delta\boldsymbol{U}(k)-\\ &\quad 2\boldsymbol{E}(k+1)^T\boldsymbol{\Gamma}_y^T\boldsymbol{\Gamma}_y\Delta\boldsymbol{U}(k)+\boldsymbol{E}(k+1)^T\boldsymbol{\Gamma}_y^T\boldsymbol{\Gamma}_y\boldsymbol{E}(k+1)\end{aligned} \tag{24}$$

In equation (24), $\boldsymbol{E}(k+1)^T\boldsymbol{\Gamma}_y^T\boldsymbol{\Gamma}_y\boldsymbol{E}(k+1)$ is not related to the control increment sequence $\Delta\boldsymbol{U}(k)$, so the objective function is transformed into the following form

$$\min_{\Delta U}\boldsymbol{J}_k=\Delta\boldsymbol{U}(k)^T\boldsymbol{H}\Delta\boldsymbol{U}(k)-\boldsymbol{G}(k+1)^T\Delta\boldsymbol{U}(k)$$
$$s.t.\begin{cases}\Delta\boldsymbol{U}_{\min}\le\Delta\boldsymbol{U}\le\Delta\boldsymbol{U}_{\max}\\ \boldsymbol{U}(k)_{\min}\le\boldsymbol{A}_I\Delta\boldsymbol{U}(k)+\boldsymbol{U}(k)\le\boldsymbol{U}(k)_{\max}\end{cases} \tag{25}$$

Where $\boldsymbol{H}=\boldsymbol{S}_u^T\boldsymbol{\Gamma}_y^T\boldsymbol{\Gamma}_y\boldsymbol{S}_u+\boldsymbol{\Gamma}_u^T\boldsymbol{\Gamma}_u$, $\boldsymbol{G}(k+1)=2\boldsymbol{S}_u^T\boldsymbol{\Gamma}_y^T\boldsymbol{\Gamma}_y\boldsymbol{E}_p(k+1)$. Since $\boldsymbol{H}\ge0$, , the quadratic programming problem is solvable for any weighting matrices $\boldsymbol{\Gamma}_y$ and $\boldsymbol{\Gamma}_u$. The control input increment for solving is denoted as

$$\Delta\boldsymbol{U}^*(k)=[\Delta\boldsymbol{u}^*(k)\quad\cdots\quad\Delta\boldsymbol{u}^*(k+m-1)]^T_{m\times1} \tag{26}$$

So $\Delta\boldsymbol{U}^*(k)$ is a non-linear function of the state quantity $\boldsymbol{\chi}_e(k)$ in the control time domain and in the prediction time domain. The first element of the resulting control sequence is generally applied to the wheeled mobile robot system. That is

$$\tilde{\boldsymbol{u}}(k)=\tilde{\boldsymbol{u}}(k-1)+\Delta\boldsymbol{u}^*(k) \tag{27}$$

At the next moment, after entering the control cycle, the new state quantities are brought into the multi-constraint optimization problem, i.e. equation (19), and solved again. The closed-loop control law of the MMPC is therefore defined as:

$$\Delta\boldsymbol{u}(k)=[I_{n\times n}\quad0\quad\cdots\quad0]\Delta\boldsymbol{U}^*(k) \tag{28}$$

3.5. MMPC controller stability analysis

For the purpose of discussing the nominal stability of the MMPC closed-loop system, it is assumed that $\boldsymbol{R}(k)=0$, the system states are all measurable, i.e. only the case of state feedback is considered.

In MMPC, the time k converges to ∞ after each moment of optimization, but when analysis the stability of the MMPC closed-loop system, the value solved for the optimization problem in each state does not guarantee the stability of the closed-loop system.

From the Lyapunov definition of stability [15], the stability of the system is a concept that tends to infinity in time, so the stability of the MMPC is assumed to be in the infinite prediction time domain and infinite control time domain, i.e. the stability of the MMPC is discussed for the case of $p=m=\infty$

Theorem 1: Consider the system model (17), if there exists a positive definite bounded objective function J_k satisfying:

(1) $\boldsymbol{J}_0=0$ and for any $k\neq 0$ exists $\boldsymbol{J}_k>0$,

(2) the Near the equilibrium point $\boldsymbol{\chi}(0)=0$, $\boldsymbol{J}_k$ is continuous,

(3))Along the trajectory of the system, there are

$$\boldsymbol{J}_{k+1}-\boldsymbol{J}_k\leq-\xi\|\boldsymbol{\chi}(k)\| \tag{29}$$

Where ξ is a κ function, the system is progressively stable.

Proof: Suppose that the infinite time domain constrained optimization problem at k has a solution whose value satisfies the control increment constraint and the control quantity constraint such that the optimized value

$$\boldsymbol{J}_k=\sum_{i=0}^{p}\left\|\boldsymbol{Y}_p(k+i\,|\,k)\right\|_Q^2+\sum_{i=0}^{m-1}\left\|\Delta\boldsymbol{U}(k+i)\right\|_R^2 \tag{30}$$

is bounded, in equation (26), $\boldsymbol{Q}$ is a weighting on the control output error and $\boldsymbol{R}$ is a weighting on the control increment, compared to the objective function equation (21) with $\boldsymbol{Q}=\boldsymbol{\Gamma}_y\boldsymbol{\Gamma}_y^T,\boldsymbol{R}=\boldsymbol{\Gamma}_u\boldsymbol{\Gamma}_u^T$.

The sequence of control increments at $k+1$ is $\Delta\boldsymbol{U}(k+1)$, which is the vector corresponding to the infinite time domain interval $[k+1,\infty]$ starting at $k+1$ for the optimal solution at k and the objective function value is calculated as :

$$\begin{aligned}\boldsymbol{J}_{k+1}&=\sum_{i=0}^{\infty}\left\|\boldsymbol{Y}_p(k+1+i\,|\,k+1)\right\|_Q^2+\sum_{i=0}^{\infty}\left\|\Delta\boldsymbol{U}(k+1+i)\right\|_R^2\\&=\boldsymbol{J}_k-\|\boldsymbol{C}\boldsymbol{\chi}(k)\|_Q^2-\left\|\Delta\boldsymbol{U}(k)\right\|_R^2\end{aligned} \tag{31}$$

Similarly $\boldsymbol{J}_{k+1}$ is bounded. The control sequence at $k+1$ is chosen to be a feasible solution to the infinite time domain constrained optimization problem. It is also deduced that:

$$\boldsymbol{J}_{k+1}\leq\boldsymbol{J}_k-\|\boldsymbol{C}\boldsymbol{\chi}(k)\|_Q^2 \tag{32}$$

When is $\boldsymbol{\chi}(0)=0$, $\boldsymbol{u}(k)=0$, then ΔU_k is a feasible solution to the optimization problem, corresponding to $\boldsymbol{J}=0$. From equation (28), we know that for any $k\geq 0$, we have $\boldsymbol{J}_k\geq 0$. Equation (29) shows that $\boldsymbol{J}_k$ is monotonically decreasing and has a minimum at $\boldsymbol{\chi}(0)=0$ and is continuous, which proves Theorem 1.

In summary, the objective function of the MMPC $\boldsymbol{J}_k$ is a Lyapunov function of the WMR system and the WMR system model (11) is nominally asymptotically stable.

4. Experimental Analysis of The System

The WMR was simulated by MATLAB and the MMPC controller module with feedforward control was built in Simulink for simulation experiments. The hardware processor for the simulation experiments is an AMD Ryzen7 4800U and the software platform is Matlab 2018a.

Simulation experiments are carried out under circular trajectory tracking and compared using TMPC and positional PID algorithms. The error maxima and variance values of each index are used as the judging criteria for trajectory tracking, and the trajectory tracking effects of the three control schemes, MMPC, TMPC and PID, are compared. Taking the lateral displacement error as an example, the positional PID controller is shown in equation (33):

$$\beta(k)=K_p(e(k))+K_i\sum_{j=1}^{k}e(j)+K_d(e(k)-e(k-1)) \tag{33}$$

Where $e(k)=x^*(k)-x(k)$ is the transverse error, K_p, K_i and K_d are the proportional, integral and differential parameters respectively. In order to make the system control effect optimal, it is necessary to optimize the multi-constraint model prediction control parameters, after repeated debugging, a set of better parameters of the positional PID control scheme $K_p=0.9$, $K_I=0.01$, $K_d=0.45$, and also obtain the system simulation parameters of MMPC as shown in Table 1:

Table 1. Parameter Comparison Data

Parameters	MMPC	TMPC	PID
Control time domain m	60	60	-
Predicted time domain p	30	30	-
Status quantity $\boldsymbol{\chi}$	3	3	3
Control volume $\boldsymbol{u}$	2	2	2
Relaxation factor weights ρ	10	10	-
Error weights $\boldsymbol{\Gamma}_y$	$10\times eye(3N_p)$	$10\times eye(3N_p)$	-
Control of incremental weights $\boldsymbol{\Gamma}_u$	$eye(2N_c)$	$eye(2N_c)$	-

4.1. Simulation of circular trajectory tracking

A circular trajectory is given as follows:

$$x_r(t)=1.25\sin(0.2t),y_r(t)=1.75-1.25\cos(0.2t) \tag{34}$$

In equation (34), $t\in[0,50]$. The constraints for the linear

velocity v and angular velocity ω are $|v| \le 1m/s$ and $|\omega| \le 0.2rad/s$, MMPC respectively, and the incremental constraints for the MMPC control scheme are $|\Delta v| \le 0.02m/s$ and $|\Delta\omega| \le 0.02rad/s$. 在. During the simulation, the initial starting point of the desired trajectory of the WMR is $(x_r, y_r, \theta_r) = (0, 0.5, 0)$ and the initial error between the actual trajectory starting point and the desired trajectory starting point is set as the initial error, the initial starting point of the actual trajectory is set as $(x, y, \theta) = (-0.5, 0.5, 0)$, and the running time is $T = 50s$, the simulation results are shown in Figure 4.

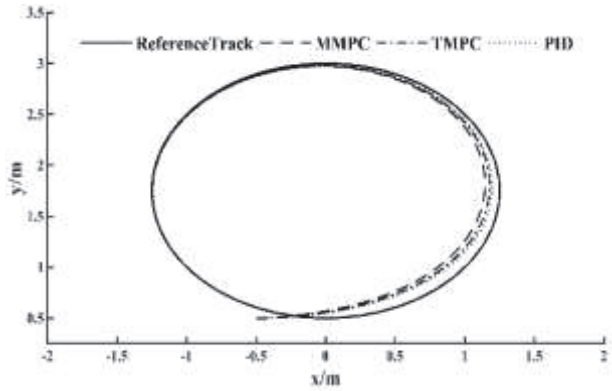
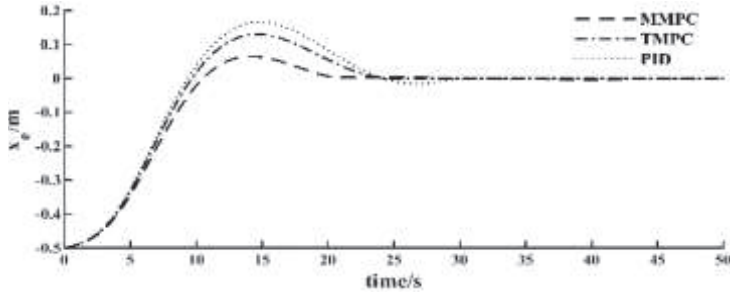
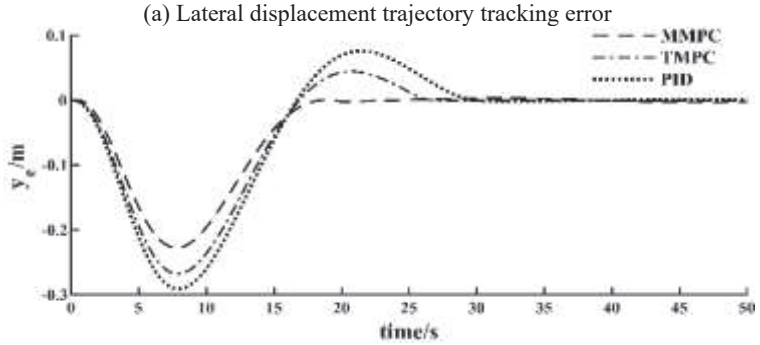


Figure 4. Simulation Diagram of Circular Trajectory Tracking

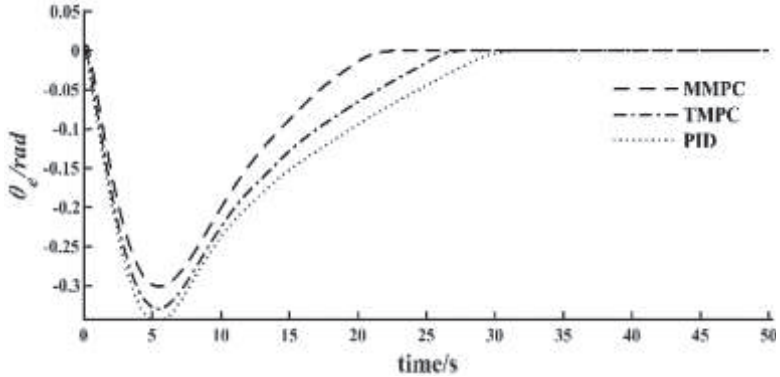
It can be seen in Figure 4 that the MMPC control scheme, the TMPC control scheme and the PID control scheme are all able to achieve trajectory tracking. It can also be seen from Figure 4 that the MMPC has the best tracking effect with the presence of initial position error, the TMPC has the second best effect and the PID has a relatively large tracking error. The variation of tracking error for the three control schemes is shown in Figure 5(a-c).



(a) Lateral displacement trajectory tracking error



(b) Longitudinal displacement trajectory tracking error



(c) Heading angle trajectory tracking error

Figure 5. Simulation Diagram of Circular Trajectory Tracking Error

Figures 5(a-c) show the tracking error simulation, where WMR converges to zero faster than TMPC and PID under the MMPC control scheme. The regulation time to track to the desired trajectory is $t = 20s$, in the MMPC and $t = 26s$ in the TMPC control scheme, while WMR has the longest regulation time using the PID control scheme at $t = 31s$. This is due to the MMPC control scheme with feed-forward control, which responds faster to the steering process when there is an initial error, improving the trajectory tracking accuracy of the WMR. After $31s$, the WMR maintains good tracking performance for all three control schemes.

Table 2. Comparison of circular track simulation

Performance indicators		MMPC	TMPC	PID
Lateral error	Maximum values	0.065	0.13	0.167
	Variance values	0.15	0.16	0.163
Longitudinal error	Maximum values	0.22	0.271	0.29
	Variance values	0.07	0.089	0.102
Heading angle error	Maximum values	0.29	0.33	0.349
	Variance values	0.097	0.106	0.111

Table 2 shows comparative performance data for the circular trajectory simulation tracking. It can be seen that MMPC has the smallest of all three performance metrics compared to the three control schemes. In terms of lateral error, the maximum error value for MMPC is 0.065m, while the maximum error for TMPC and PID is 0.13m and 0.167m respectively.

The calculation shows that the MMPC control scheme reduces lateral error by 17.96%, longitudinal error by 6.53% and heading angle error by 4.13% compared to the TMPC scheme. Compared to the PID control scheme, the lateral error is reduced by 28.18%, the longitudinal error by 8.94% and the heading angle error by 6.09%. This indicates that the MMPC control scheme is superior to the TMPC and PID control schemes.

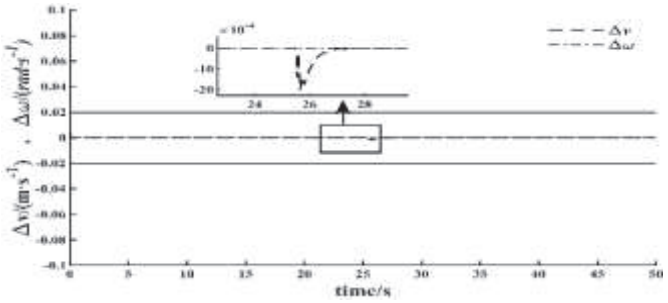


Figure 6. Control Incremental Change of MMPC

Figure 6 shows the MMPC control increment variation and it can be seen that both the linear velocity increment Δv and the angular velocity increment $\Delta\omega$ satisfy the input constraints. With the velocity optimisation strategy, the velocity of the WMR stably stays within the constraint. The simulation results verify the effectiveness of the velocity optimisation strategy.

4.2. Circular trajectory tracking experiment

Figure 7. EAIBOT Mobile Platform

Figure 7 shows the EAIBOT mobile platform. In order to verify the robustness and feasibility of the control scheme designed in this paper in a real environment, physical experiments are carried out using this mobile platform. To record the trajectory tracking data, a model was built using Simulink in conjunction with ROS as shown in Figure 8.

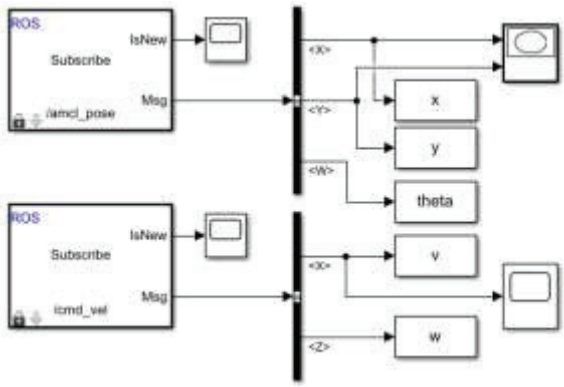


Figure 8. Simulink and ROS Joint Simulation

The MPC control scheme and the PID control scheme are still chosen for comparison. The algorithm parameters are the same as in Table 1.

The constraints for the linear velocity v and ω the angular velocity are $|v| \le 1m/s$ and $|\omega| \le 0.2rad/s$ respectively. The control increment constraints are $|\Delta v| \le 0.1m/s$ and $|\Delta\omega| \le 0.1rad/s$ .Without affecting the experimental results, the initial starting point of the desired trajectory of the WMR was set to $(x_r, y_r, \theta_r) = (0.85, 0.8, 0)$ to track a circular trajectory and compare the performance of the three control schemes.

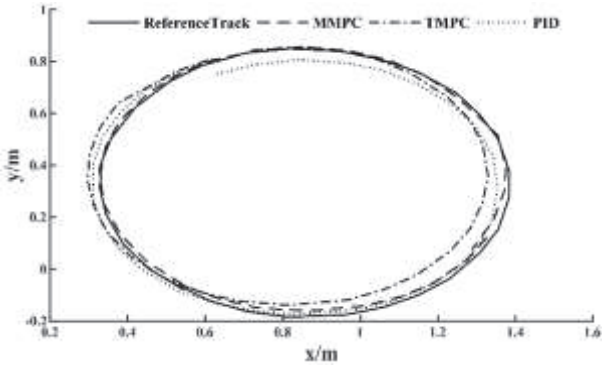
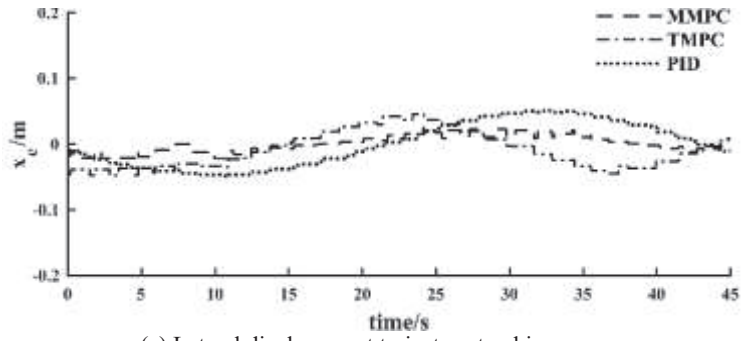


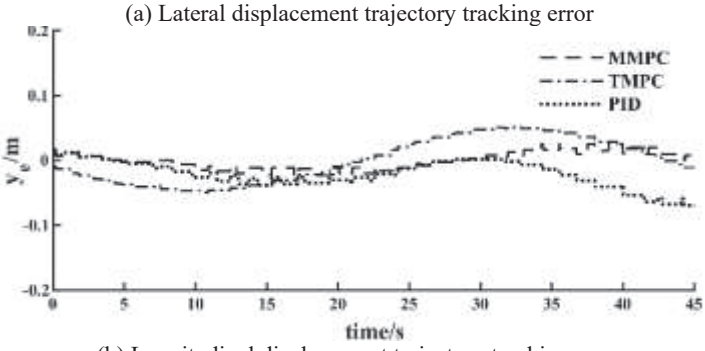
Figure 9. Effect Picture of Circular Trajectory Tracking

Figure 9 shows the effect of circular trajectory tracking. Under the physical experiment, the MMPC control scheme can complete the trajectory tracking experiment better and the tracking performance of the system is good, while the MPC

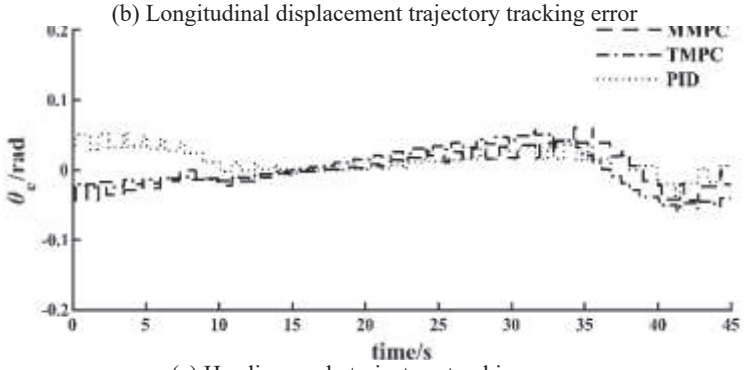
control scheme and the PID control scheme both show different degrees of deviation during the tracking trajectory.



(a) Lateral displacement trajectory tracking error



(b) Longitudinal displacement trajectory tracking error



(c) Heading angle trajectory tracking error

Figure 10. Rendering of Circular Track Tracking Error

Figure 10(a-c) shows the circular trajectory tracking error diagram. During the trajectory tracking process, its tracking effect has a certain bias, taking into account two main factors:

(1) Factors of the experimental site, such as the coefficient of friction and flatness of the ground causing the robot tyres to slip sideways or create a gap with the ground.

(2) Insufficient accuracy of the EAIBOT moving platform sensors, resulting in deviations in the calculation of both linear and angular velocities at the previous moment.

As a result of these two main factors, the tracking error of the robot does not tend to be zero, but the overall trend is as expected, i.e. the results are similar to those of the simulation. It can be seen that the MMPC control scheme has improved tracking accuracy and reduced steady-state error compared to the TMPC and PID control schemes.

Table 2. Comparison of circular track performance

Performance indicators		MMPC	TMPC	PID
Lateral error	Maximum values	0.025	0.047	0.051
	Variance values	0.013	0.027	0.033
Longitudinal error	Maximum values	0.026	0.051	0.069
	Variance values	0.014	0.02	0.033
Heading angle error	Maximum values	0.043	0.058	0.052
	Variance values	0.024	0.027	0.029

Table 3 shows the comparative data of the circular trajectory tracking performance in the physical experimental environment. In terms of longitudinal error, the MMPC has a maximum error value of 0.026 m, while the TMPC and PID have a maximum error of 0.051 m and 0.069 m respectively. The calculation shows that the MMPC control scheme reduces the lateral error by 17.87%, the longitudinal error by 17.12% and the heading angle error by 9.81% compared to

the TMPC scheme. Compared to the PID control scheme, the lateral error is reduced by 20.82%, the longitudinal 9 error is reduced by 23.26% and the heading angle error is reduced by 5.87%.

Figure 11 shows the variation of the MMPC control increments. Even in the real scenario, with the velocity optimization strategy, both the linear velocity increments Δv and the angular velocity increments $\Delta\omega$ satisfy the input constraints. In summary, both the simulation and physical experiments validate the effectiveness and robustness of the MMPC control scheme.

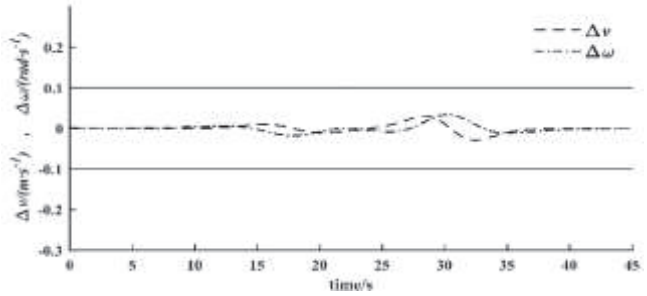


Figure 11. Control Incremental Change of MMPC

5. Conclusion

To address the problems of low WMR tracking accuracy and large steady-state error, this paper designs a MMPC control scheme combining feedforward control and feedback control. Firstly, the WMR kinematic model is established considering the control increment constraint and the error caused by the retained linearization. Then, the MMPC feedback controller with feedforward controller is designed with the addition of MMPC with speed optimization strategy, and the stability of the MMPC controller is demonstrated using Lyapunov theory. Finally, the feasibility of the control scheme designed in this paper is verified through simulation experiments and physical experiments. The results show that the MMPC control scheme based on the combination of feedforward control and feedback control is better than the TMPC control scheme and the PID control scheme in terms of tracking accuracy and tracking error. In future research, we will consider improving the real-time performance of the algorithm, using the WMR dynamics model, etc. On this basis, we will further improve the control scheme proposed in this paper in order to improve the trajectory tracking control performance.

References

[1] Li Yichun, WANG Junwei,YU Xinhai,WEN Rong,et al. Research on trajectory tracking control in mobile robot modeling and unmodeled disturbance environment [J]. Machine Tool & Hydraulics, 2021, 49(9): 21-27.

[2] Lin Yifan, Chen Yangjie, He Bingwei, et al. Non-collision checking RRT* algorithm for mobile robot motion planning. [J] Chinese Journal of Scientific Instrument. 2020, 41(10): 257-267.

[3] Xu Yuqiong, Lou Ke, Li Tingting, et al. Path planning of mobile robot based on improved adaptive ant colony algorithm.[J] Journal of electronic measurement and instrumentation, 2019, 33(10): 89-95.

[4] Chwa D. Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates[J]. IEEE Transactions on Control Systems Technology, 2004, 12 (4): 637-644.

[5] Tinh N, Linh L E. Neural network-based adaptive tracking control for a nonholonomic wheeled mobile robot with unknown wheel slips, model uncertainties, and unknown bounded disturbances[J]. Turkish Journal of Electrical Engineering and Computer Sciences, 2018, 26(1):378-392.

[6] Yang H, Wang S, Zuo Z, et al. Trajectory tracking for a wheeled mobile robot with an omnidirectional wheel on uneven ground[J]. IET Control Theory and Applications, 2020, 14(7): 921-929.

[7] Hua C, Singh B K. Nonholonomic Wheeled Mobile Robot Trajectory Tracking Control Based on Improved Sliding Mode Variable Structure[J]. Wireless Communications and Mobile Computing, 2021, 2021(10):1-9.

[8] Yue M, An C, Ding L, et al. A MPC motion planning-based sliding mode control for underactuated WPS vehicle via Olfati transformation[J]. IET Control Theory & Applications, 2017, 12(4):495-503.

[9] Sun Z, Dai L, Xia Y, et al. Event-based model predictive tracking control of nonholonomic systems with coupled input constraint and bounded disturbances[J]. IEEE Transactions on Automatic Control, 2018(63):608-615..

[10] Liu H, Zhang G M, Ouyang H M. Trajectory Tracking Algorithm Based on Model Predictive Control[J]. Control Engineering of China, 2016(S1):61-65.

[11] Xiao H, Li Z, Yang C, et al. Robust stabilization of a wheeled mobile robot using model predictive control based on neurodynamics optimization [J]. IEEE Transactions on Industrial Electronics, 2016, 64(1): 505-516.

[12] Pacheco L, Luo N. Testing PID and MPC performance for mobile robot local path-following[J]. International Journal of Advanced Robotic Systems, 2015, 12(11): 155-168.

[13] Pepy R, Lambert A, Mounier H. Reducing navigation errors by planning with realistic vehicle model[C]. IEEE Intelligent Vehicles Symposium, 2006:300-307.

[14] Sun Z T, Bai J J, Chen B X, et al. Adaptive Trajectory Tracking Control for Wheeled Mobile Robots[J]. Control Engineering of China, 2021,28(12):2420-2425.

[15] Ma C, Zhang X K, Yang G P. Improved nonlinear control for ship course-keeping based on Lyapunov stability [J]. Chinese Journal of Ship Research, 2019, 14(1)150-155, 161.