IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 December 2023, accepted 11 January 2024, date of publication 22 January 2024, date of current version 30 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3356583

RESEARCH ARTICLE

Mecanum Wheel AGV Trajectory Tracking Control Based on Efficient MPC Algorithm

MIN TANG[1], **SHUSEN LIN**[2], **AND YIXUAN LUO**[2]
[1]School of Mechanical Engineering, Zhejiang Sci-Tech University, Hangzhou 310018, China
[2]College of Intelligent Manufacturing, Taizhou University, Taizhou 318000, China

Corresponding author: Shusen Lin (t13587295613@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 51905364.

ABSTRACT In response to the challenge of insufficient trajectory tracking accuracy and low solution efficiency of Mecanum wheel AGV (Automated Guided Vehicle) under complex and constrained working conditions, this paper proposes an efficient Model Predictive Control (MPC) method to achieve superior tracking performance and robustness. Initially, a linear error model of the mobile platform is established based on pose error, serving as the predictive model for the MPC controller. A target function is designed to transform the trajectory tracking control problem into an optimal control problem. To handle inequality constraints, penalty terms are introduced into the objective function, and the resulting constrained problem is subsequently solved to approximate the optimal solution for the original inequalities. To alleviate the computational burden associated with real-time optimization problem-solving, an efficient MPC algorithm. has been developed. To ensure closed-loop stability under the MPC control method, stability constraints are imposed on the new optimization problem. Simulation results demonstrate that, in comparison to traditional MPC methods, the proposed approach reduces the average solution calculation time by 5.1% and the maximum single calculation time by 13.7%, all while maintaining trajectory tracking accuracy. These results validate the algorithm's feasibility, effectively addressing the challenges associated with solving MPC problems.

INDEX TERMS Computational efficiency, Mecanum wheel, model predictive control, stability, trajectory tracking.

I. INTRODUCTION

This document discusses an Automated Guided Vehicle, also referred to as an unmanned transport vehicle–an automated, unmanned, intelligent handling device categorized under mobile robotic systems [1]. Specifically, the Mecanum wheel AGV represents a comprehensive system that integrates various functions, including environmental perception, dynamic decision-making and planning, behavior control, and execution. It finds extensive applications across multiple industries, including manufacturing, logistics, and unmanned warehousing [2], [3]. Renowned for its exceptional maneuverability, capacity for zero-radius turns in any direction, and mobility in confined spaces, the Mecanum wheel AGV is widely utilized in diverse applications [4], [5]. Despite its advantages, practical applications of the Mecanum wheel AGV encounter challenges in trajectory tracking control, a key concern marked by several issues. Factors such as uncertainty, external disturbances, system nonlinearity, and sensor noise present formidable challenges, rending traditional control methods incapable of meeting precise trajectory tracking requirements. Problems like deviation from desired trajectories, control oscillations, response delays, and constraint violations may arise. Traditional control methods often struggle to effectively address tracking control problems under these multiple constraint conditions, causing AGVs to respond inadequately to control commands and resulting in failures in efficient, accurate, and reliable handling tasks.

In addressing the challenging problem of trajectory tracking control, scholars both domestically and internationally have explored various control methods. Fuzzy control [6],

The associate editor coordinating the review of this manuscript and approving it for publication was Ton Duc Do.

PID control [7], neural network control [8], sliding mode control [9], and MPC [10] have all been employed to tackle this complex issue. For instance, in reference [11], a PID control algorithm integrating dynamic lookahead and yaw angle feedback is utilized to achieve lateral trajectory tracking control by effectively decoupling yaw motion and lateral motion. Meanwhile, in reference [12], a single-point preview method is adopted to establish dynamic lateral position and yaw angle error models. It employs a nonsingular terminal sliding mode control strategy to design stable lane-keeping control, ensuring stability through validation using Lyapunov theory and significantly reducing lateral deviation. However, challenges persist in these control methods, as discussed in references [11] and [12], specifically concerning the ineffective constraint on control output. This limitation may result in decreased control performance due to output saturation and even pose a risk of vehicle instability. Moreover, coordinating control between multiple objectives remains a challenge, particularly in multi-objective scenarios. To complement these studies, literature [13] introduces a unique steering strategy based on a lane-line detection model. This method employs both line selection and CNN-based extraction to predict lane markings from images captured by a forward-looking monocular camera. The detected lane markings are then utilized to estimate the vehicle's next destination, with DC servo motor steering control ensuring precise navigation on a golf cart for various tasks. Furthermore, literature [14] explores the application of convolutional neural networks (CNNs) for training and simulating driverless car models. Using three cameras capturing left, right, and center images labeled with steering angles and speed parameters, the CNN is trained to navigate the vehicle, implementing a steering angle adjustment strategy to maintain lane centering. The method is evaluated on the UDACITY simulation platform, yielding impressive accuracy results. In a different approach, literature [15] presents a prototype of a monocular vision autonomous vehicle based on Raspberry Pi deep neural network. This work focuses on developing a model using a deep neural network to directly map the input image to the predicted steering angle. The on-board platform, consisting of 1/10 RC cars, Raspberry Pi 3 model B computer, and a front-facing camera, is used to train CNN model parameters. Experimental road testing in outdoor environments, including autonomous driving on oval and number 8 tracks with traffic signs, validates the model's effectiveness and robustness in lane-keeping tasks.

Comparison with other methods reveals that MPC possesses significant advantages in addressing multi-objective constrained optimization problems. It excels in predicting future trajectory states, rendering it widely applicable in solving problems within the trajectory tracking domain. Reference [16] introduces a path tracking algorithm based on MPC. By employing quadratic programming (QP) optimization, it computes the optimal steering instructions for the trajectory. Combining the characteristics of the steering execution system, it achieves precise and smooth trajectory tracking for vehicles. In a similar vein, reference [17] proposes an MPC method with integral action, which preserves not only the predictive and optimization capabilities of MPC but also mitigates steady-state errors arising during the operation of omnidirectional mobile robots. This enhancement contributes to an overall improvement in the controller's tracking performance. Addressing the steering avoidance challenges for autonomous vehicles, reference [18] presents a hierarchical obstacle avoidance control strategy based on MPC theory. Simulation results demonstrate that the upper-level controller dynamically plans local obstacle avoidance paths, and the lower-level controller consistently tracks these paths, ensuring stable and effective avoidance. While the aforementioned references [16], [17], and [18] meticulously analyze the performance of controllers under simulation conditions, it is noteworthy that a highly complex controlled object model can significantly escalate the online iteration computation load of MPC, consequently diminishing real-time performance and impeding the practical application of MPC controllers [19]. To mitigate the online computation burden of MPC controllers, scholars have proposed methods such as Explicit MPC [20], control block methods [21], and interpolation control methods [22], which find relevant applications in trajectory tracking.

To complement existing studies, we introduce a streamlined MPC approach tailored for the trajectory tracking of Mecanum wheel AGV, with three main contributions:

1) Proposing an efficient MPC scheme that considers constraints of the Mecanum wheel AGV to reduce computational complexity while ensuring trajectory tracking feasibility.

2) Introducing an efficient MPC algorithm to address computational efficiency issues, incorporating penalty terms on the objective function to handle inequality constraints and enhance computational efficiency.

3) Integrating stability constraints within a finite time domain to ensure closed-loop stability in the context of the Mecanum wheel AGV system.

The remainder of this article is organized as follows. Section II focuses on establishing a linear error model for the mobile platform based on the motion characteristics of Mecanum wheels. In Section III, we utilize the established model as the predictive model for the MPC controller. We design a quadratic objective function to transform the trajectory tracking problem into an optimal control problem. In Section IV, we introduce penalty terms to the objective function to relax inequality constraints. We also design an efficient MPC algorithm to enhance computational efficiency while ensuring stability under closed-loop MPC control. Stability constraints are imposed on the new optimization problem. In Section V, we validate the effectiveness of the proposed method through MATLAB simulation experiments. Section VI concludes the paper with a summary of the presented work.

II. AGV MOBILE PLATFORM ERROR MODEL

The motion model of the Mecanum wheel mobile platform is illustrated in Figure 1. Treating the mobile platform as a rigid body model with an unchanged mechanical structure and dimensions, effects such as wheel slipping and mechanical friction are neglected. To accurately represent the platform's motion state, we establish a linear error model of the mobile platform based on pose error.

$X_W O_W Y_W$ is the world coordinate system, (x, y) represents the coordinates of the model center point, and θ indicates the heading angle of the model, ω, V_x, V_y denote the rotational speeds of the mobile platform, and the velocities along the X and Y axes in the platform coordinate system, respectively.

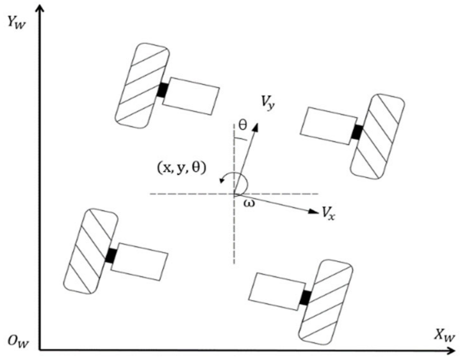

FIGURE 1. Schematic diagram of the Mecanum wheel mobile platform model.

The motion state model of the Mecanum wheel mobile platform is outlined as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} V_x \cos\theta + V_y \sin\theta \\ V_y \cos\theta + V_x \sin\theta \\ \omega \end{bmatrix} \tag{1}$$

Refer to (1), $\left(\dot{x}, \dot{y}, \dot{\theta}\right)$ represents the pose derivative of the mobile platform, and $\boldsymbol{P} = (x, y)^T$ denotes the coordinates of the vehicle's center of mass in the global coordinate system. Here, $\boldsymbol{X} = [x, y, \theta]^T$ and $\boldsymbol{u} = \left[V_x, V_y, \omega\right]^T$ represent the state variables and input control variables of the mobile platform, and the state-space description can be expressed as follows:

$$\dot{\boldsymbol{X}} = f\left(\boldsymbol{X}, \boldsymbol{u}\right) \tag{2}$$

Owing to the nonlinearity in (2), linearization becomes imperative. Assuming that the position of any reference point on the trajectory is denoted as (x_r, y_r, θ_r) and simultaneously satisfies $\dot{\boldsymbol{X}}_{\boldsymbol{r}} = f\left(\boldsymbol{X}_{\boldsymbol{r}}, \boldsymbol{u}_{\boldsymbol{r}}\right)$, where $\boldsymbol{X}_{\boldsymbol{r}} = [x_r, y_r, \theta_r]^T$ and $\boldsymbol{u}_{\boldsymbol{r}} = \left[V_{xr}, V_{yr}, \omega_r\right]^T$. Expanding (2) at the reference point using a Taylor series and neglecting higher-order terms, we obtain the following expression:

$$\dot{\boldsymbol{X}} = f\left(\boldsymbol{X}_{\boldsymbol{r}}, \boldsymbol{u}_{\boldsymbol{r}}\right) + \frac{\partial f\left(\boldsymbol{X}, \boldsymbol{u}\right)}{\partial \boldsymbol{X}}\left(\boldsymbol{X} - \boldsymbol{X}_{\boldsymbol{r}}\right) + \frac{\partial f\left(\boldsymbol{X}, \boldsymbol{u}\right)}{\partial \boldsymbol{u}}\left(\boldsymbol{u} - \boldsymbol{u}_{\boldsymbol{r}}\right) \tag{3}$$

Subtracting the expression (3) from (2) yields the general form of the linear error model as follows:

$$f\left(\boldsymbol{X}, \boldsymbol{u}\right) - f\left(\boldsymbol{X}_{\boldsymbol{r}}, \boldsymbol{u}_{\boldsymbol{r}}\right) = \frac{\partial f\left(\boldsymbol{X}, \boldsymbol{u}\right)}{\partial \boldsymbol{X}}\left(\boldsymbol{X} - \boldsymbol{X}_{\boldsymbol{r}}\right) + \frac{\partial f\left(\boldsymbol{X}, \boldsymbol{u}\right)}{\partial \boldsymbol{u}}\left(\boldsymbol{u} - \boldsymbol{u}_{\boldsymbol{r}}\right) \tag{4}$$

Rewriting the linear error model equation in state-space form results in:

$$\dot{\boldsymbol{X}}_{\boldsymbol{e}} = \boldsymbol{A}\boldsymbol{X}_{\boldsymbol{e}} + \boldsymbol{B}\boldsymbol{u}_{\boldsymbol{e}} \tag{5}$$

where

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & -V_{xr}\sin\theta_r + V_{yr}\cos\theta_r \\ 0 & 0 & -V_{yr}\sin\theta_r - V_{xr}\cos\theta_r \\ 0 & 0 & 0 \end{bmatrix},$$

$$\boldsymbol{B} = \begin{bmatrix} \cos\theta_r & \sin\theta_r & 0 \\ -\sin\theta_r & \cos\theta_r & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

To achieve precise control and trajectory tracking of the mobile platform, a controller needs to be designed to ensure that the platform's state follows the predetermined target trajectory. The control objective is to adjust input control variables to achieve the desired pose state. Specifically, the goal is for the mobile platform to move along the given reference trajectory at the desired speed and heading angle. Two main objectives are sought:

1) Ensure that the system output accurately tracks the desired pose and maintains the pose tracking error within a predetermined steady-state boundary range.

2) Ensure system stability and robustness, with parameters remaining bounded.

III. DESIGNING AN MPC TRAJECTORY TRACKING CONTROLLER

A. ESTABLISHING A PREDICTIVE MODEL FOR THE POSE ERROR OF THE MOBILE PLATFORM

Utilize the linear error model established based on pose error as the predictive model for the MPC controller. Since this linear error model represents a continuous system, it cannot be directly applied to the design of the MPC controller. Hence, discretization of (5) is achieved through the utilization of the Euler method [23]. The forward Euler method is a first-order numerical method for solving ordinary differential equations with given initial values. Assuming T is the sampling period, its discretization is represented as:

$$\boldsymbol{X}_{\boldsymbol{e}}\left(\boldsymbol{k}+1\right) = \boldsymbol{a}\boldsymbol{X}_{\boldsymbol{e}}\left(\boldsymbol{k}\right) + \boldsymbol{b}\boldsymbol{u}_{\boldsymbol{e}}\left(\boldsymbol{k}\right) \tag{6}$$

where

$$\boldsymbol{a} = \begin{bmatrix} 1 & 0 & -TV_{xr}\sin\theta_r + TV_{yr}\cos\theta_r \\ 0 & 1 & -TV_{yr}\sin\theta_r - TV_{xr}\cos\theta_r \\ 0 & 0 & 1 \end{bmatrix},$$

$$\boldsymbol{b} = \begin{bmatrix} T\cos\theta_r & T\sin\theta_r & 0 \\ -T\sin\theta_r & T\cos\theta_r & 0 \\ 0 & 0 & T \end{bmatrix}.$$

Define the output equation as:

$$\boldsymbol{y}(k)=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\boldsymbol{X}_e(k)=\boldsymbol{I}\boldsymbol{X}_e(k) \tag{7}$$

To precisely constrain the control increments and solve for the optimal control, the discrete state variable $\boldsymbol{X}_e(k)$ and control variable $\boldsymbol{u}_e(k-1)$ are combined to construct a new state variable, denoted as $\boldsymbol{\zeta}(k)=\left[\boldsymbol{X}_e(k)\ \boldsymbol{u}_e(k-1)\right]^T$. Consequently, the new state-space equation is expressed as:

$$\begin{cases}\boldsymbol{\zeta}(k+1)=\boldsymbol{A}_k\boldsymbol{\zeta}(k)+\boldsymbol{B}_k\Delta\boldsymbol{u}(k)\\ \boldsymbol{\eta}(k)=\boldsymbol{C}_k\boldsymbol{\zeta}(k)\end{cases} \tag{8}$$

In which, $\boldsymbol{A}_k=\begin{bmatrix}\boldsymbol{a}&\boldsymbol{b}\\0_{N_u\times N_X}&\boldsymbol{I}_{N_u}\end{bmatrix}$, $\boldsymbol{B}_k=\begin{bmatrix}\boldsymbol{b}\\\boldsymbol{I}_{N_u}\end{bmatrix}$, $\boldsymbol{C}_k=\left[\boldsymbol{I}_{N_X}\ 0\right]$, $\boldsymbol{N}_X$ represent the dimensions of the state variables, and $\boldsymbol{N}_u$ represents the dimension of the control variables. At each moment within the prediction time horizon N_p the state variables are given by:

$$\begin{aligned}\boldsymbol{\zeta}(k+1)&=\boldsymbol{A}_k\boldsymbol{\zeta}(k)+\boldsymbol{B}_k\Delta\boldsymbol{u}(k)\\ \boldsymbol{\zeta}(k+2)&=\boldsymbol{A}_k^2\boldsymbol{\zeta}(k)+\boldsymbol{A}_k\boldsymbol{B}_k\Delta\boldsymbol{u}(k)\\&\quad+\boldsymbol{B}_k\Delta\boldsymbol{u}(k+1)\\&\ \vdots\\ \boldsymbol{\zeta}(k+N_c)&=\boldsymbol{A}_k^{N_c}\boldsymbol{\zeta}(k)+\boldsymbol{A}_k^{N_c-1}\boldsymbol{B}_k\Delta\boldsymbol{u}(k)\\&\quad+\cdots+\boldsymbol{A}_k^0\boldsymbol{B}_k\Delta\boldsymbol{u}(k+N_c-1)\\ \boldsymbol{\zeta}\left(k+N_p\right)&=\boldsymbol{A}_k^{N_p}\boldsymbol{\zeta}(k)+\boldsymbol{A}_k^{N_p-1}\boldsymbol{B}_k\Delta\boldsymbol{u}(k)\\&\quad+\cdots+\boldsymbol{A}_k^0\boldsymbol{B}_k\Delta\boldsymbol{u}\left(k+N_p-1\right)\end{aligned} \tag{9}$$

At each moment within the prediction time horizon, the output variables of the system are:

$$\begin{aligned}\boldsymbol{\eta}(k+1)&=\boldsymbol{C}_k\boldsymbol{A}_k\boldsymbol{\zeta}(k)+\boldsymbol{C}_k\boldsymbol{B}_k\Delta\boldsymbol{u}(k)\\ \boldsymbol{\eta}(k+2)&=\boldsymbol{C}_k\boldsymbol{A}_k^2\boldsymbol{\zeta}(k)+\boldsymbol{C}_k\boldsymbol{A}_k\boldsymbol{B}_k\Delta\boldsymbol{u}(k)\\&\quad+\boldsymbol{C}_k\boldsymbol{B}_k\Delta\boldsymbol{u}(k+1)\\&\ \vdots\\ \boldsymbol{\eta}(k+N_c)&=\boldsymbol{C}_k\boldsymbol{A}_k^{N_c}\boldsymbol{\zeta}(k)+\boldsymbol{C}_k\boldsymbol{A}_k^{N_c-1}\boldsymbol{B}_k\Delta\boldsymbol{u}(k)\\&\quad+\cdots+\boldsymbol{C}_k\boldsymbol{A}_k^0\boldsymbol{B}_k\Delta\boldsymbol{u}(k+N_c-1)\\ \boldsymbol{\eta}\left(k+N_p\right)&=\boldsymbol{C}_k\boldsymbol{A}_k^{N_p}\boldsymbol{\zeta}(k)+\boldsymbol{C}_k\boldsymbol{A}_k^{N_p-1}\boldsymbol{B}_k\Delta\boldsymbol{u}(k)\\&\quad+\cdots+\boldsymbol{C}_k\boldsymbol{A}_k^0\boldsymbol{B}_k\Delta\boldsymbol{u}\left(k+N_p-1\right)\end{aligned} \tag{10}$$

The system's output equation can be obtained by deriving it from the expression given in (10) as follows:

$$\boldsymbol{Y}=\boldsymbol{\psi}\boldsymbol{\zeta}(k)+\boldsymbol{\Theta}\Delta\boldsymbol{U} \tag{11}$$

where

$$\boldsymbol{Y}=\begin{bmatrix}\boldsymbol{\eta}(k+1)\\\boldsymbol{\eta}(k+2)\\\cdots\\\boldsymbol{\eta}(k+N_c)\\\cdots\\\boldsymbol{\eta}\left(k+N_p\right)\end{bmatrix},\quad \boldsymbol{\psi}=\begin{bmatrix}\boldsymbol{C}_k\boldsymbol{A}_k\\\boldsymbol{C}_k\boldsymbol{A}_k^2\\\cdots\\\boldsymbol{C}_k\boldsymbol{A}_k^{N_c}\\\cdots\\\boldsymbol{C}_k\boldsymbol{A}_k^{N_p}\end{bmatrix}.$$

$$\boldsymbol{\Theta}=\begin{bmatrix}\boldsymbol{C}_k\boldsymbol{B}_k&0&\cdots&0\\\boldsymbol{C}_k\boldsymbol{A}_k\boldsymbol{B}_k&\boldsymbol{C}_k\boldsymbol{B}_k&\cdots&0\\\cdots&\cdots&\ddots&\cdots\\\boldsymbol{C}_k\boldsymbol{A}_k^{N_c-1}\boldsymbol{B}_k&\boldsymbol{C}_k\boldsymbol{A}_k^{N_c-2}\boldsymbol{B}_k&\cdots&\boldsymbol{C}_k\boldsymbol{A}_k^0\boldsymbol{B}_k\\\cdots&\cdots&\ddots&\cdots\\\boldsymbol{C}_k\boldsymbol{A}_k^{N_p-1}\boldsymbol{B}_k&\boldsymbol{C}_k\boldsymbol{A}_k^{N_p-2}\boldsymbol{B}_k&\cdots&\boldsymbol{C}_k\boldsymbol{A}_k^{N_p-N_c}\boldsymbol{B}_k\end{bmatrix}$$

$$\Delta\boldsymbol{U}=\begin{bmatrix}\Delta\boldsymbol{u}(k)\\\Delta\boldsymbol{u}(k+1)\\\Delta\boldsymbol{u}(k+2)\\\cdots\\\Delta\boldsymbol{u}(k+N_c-1)\end{bmatrix}.$$

Referring to (11), the prediction of the system's output variables within the future horizon N_p can be achieved when the current state variables and control variables within the control horizon N_c are known

B. DESIGNING THE OBJECTIVE FUNCTION

To obtain the optimal control sequence, it is essential to design a rational optimization objective. By computing the objective function, a series of optimal control variables can be obtained. The formulated objective function is as follows:

$$\boldsymbol{J}=(\boldsymbol{Y}-\boldsymbol{Y}_r)^T\boldsymbol{Q}(\boldsymbol{Y}-\boldsymbol{Y}_r)+\Delta\boldsymbol{U}^T\boldsymbol{R}\Delta\boldsymbol{U} \tag{12}$$

In the above expressions, $\boldsymbol{Y}_r$ is the reference output of the system, $\boldsymbol{Q}$ and $\boldsymbol{R}$ are positive definite weight matrices for the system. The first term in the equation represents the accumulation of trajectory deviation during tracking. Increasing the weight matrix $\boldsymbol{Q}$ reduces the tracking error of the mobile platform. The second term signifies the energy consumption during the tracking process; i.e., when the weight matrix $\boldsymbol{R}$ is larger, the output changes of the control variables are smoother, resulting in a more stable motion of the mobile platform.

For ease of computer computation, the objective function is processed according to the methods outlined in reference [24], as follows:

$$\begin{aligned}\boldsymbol{J}&=\Delta\boldsymbol{U}^T\left(\boldsymbol{\Theta}^T\boldsymbol{Q}\boldsymbol{\Theta}+\boldsymbol{R}\right)\Delta\boldsymbol{U}+\boldsymbol{E}^T\boldsymbol{Q}\boldsymbol{E}\\&\quad+2\boldsymbol{E}^T\boldsymbol{Q}\boldsymbol{\Theta}\Delta\boldsymbol{U}-2\boldsymbol{Y}_r\boldsymbol{Q}\boldsymbol{\Theta}\Delta\boldsymbol{U}\\&\quad+\boldsymbol{Y}_r^T\boldsymbol{Q}\boldsymbol{Y}-2\boldsymbol{Y}_r^T\boldsymbol{Q}\boldsymbol{E}\end{aligned} \tag{13}$$

When solving the objective function, the constant term in the above polynomial can be omitted. Simultaneously, to ensure the solvability of the objective function, the introduction of a relaxation factor *row* can facilitate a smooth solving process, avoiding potential failures due to overly strict solving conditions. Further discussion on this matter will be presented in the following text.

$$\boldsymbol{H}=\begin{bmatrix}\boldsymbol{\Theta}^T\boldsymbol{Q}\boldsymbol{\Theta}+\boldsymbol{R}&0_{N_u{*}N_c\times 1}\\0_{1\times N_u{*}N_c}&row\end{bmatrix},\quad \boldsymbol{g}=\left[\boldsymbol{E}^T\boldsymbol{Q}\boldsymbol{\Theta}\quad 0\right],$$

transforming it into the quadratic form of the objective function, the above expression can be rewritten as:

$$\begin{aligned}\min_{\boldsymbol{\Delta U}} \boldsymbol{J} &= 2\left(\frac{1}{2}\boldsymbol{\Delta U}^T\boldsymbol{H}\boldsymbol{\Delta U}+\boldsymbol{g}^T\boldsymbol{\Delta U}\right)\\ &= \frac{1}{2}\boldsymbol{\Delta U}^T\boldsymbol{H}\boldsymbol{\Delta U}+\boldsymbol{g}^T\boldsymbol{\Delta U} \qquad (14)\end{aligned}$$

An outstanding attribute of MPC lies in its capacity to manage numerous constraints. In practical engineering control, it is crucial to set sensible boundaries on control variables and increments. This entails designing constraint conditions for both the control variables and increments of the mobile platform. The connection between control variables and increments can be articulated as follows:

$$\begin{aligned}\boldsymbol{U} = \begin{bmatrix}\boldsymbol{u_e}(\boldsymbol{k})\\ \boldsymbol{u_e}(\boldsymbol{k}+1)\\ \cdots\\ \boldsymbol{u_e}(\boldsymbol{k}+\boldsymbol{N_c}-1)\end{bmatrix} &= \begin{bmatrix}\boldsymbol{u_e}(\boldsymbol{k}-1)\\ \boldsymbol{u_e}(\boldsymbol{k}-1)\\ \cdots\\ \boldsymbol{u_e}(\boldsymbol{k}-1)\end{bmatrix}\\ &+ \begin{bmatrix}\boldsymbol{I}_{\boldsymbol{N_c}} & 0 & \cdots & 0\\ \boldsymbol{I}_{\boldsymbol{N_c}} & \boldsymbol{I}_{\boldsymbol{N_c}} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ \boldsymbol{I}_{\boldsymbol{N_c}} & \boldsymbol{I}_{\boldsymbol{N_c}} & \cdots & \boldsymbol{I}_{\boldsymbol{N_c}}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Delta u}(\boldsymbol{k})\\ \boldsymbol{\Delta u}(\boldsymbol{k}+1)\\ \cdots\\ \boldsymbol{\Delta u}(\boldsymbol{k}+\boldsymbol{N_c}-1)\end{bmatrix} \qquad (15)\end{aligned}$$

By combining the expressions from (7) and (8), we obtain the following result:

$$\begin{bmatrix}\boldsymbol{u}_{\boldsymbol{e}_{min}}\\ \boldsymbol{u}_{\boldsymbol{e}_{min}}\\ \vdots\\ \boldsymbol{u}_{\boldsymbol{e}_{min}}\end{bmatrix} \le \begin{bmatrix}\boldsymbol{u_e}(\boldsymbol{k})\\ \boldsymbol{u_e}(\boldsymbol{k}+1)\\ \vdots\\ \boldsymbol{u_e}(\boldsymbol{k}+\boldsymbol{N_c}-1)\end{bmatrix} \le \begin{bmatrix}\boldsymbol{u}_{\boldsymbol{e}_{max}}\\ \boldsymbol{u}_{\boldsymbol{e}_{max}}\\ \vdots\\ \boldsymbol{u}_{\boldsymbol{e}_{max}}\end{bmatrix} \qquad (16)$$

Hence, the objective function can be streamlined into a standard quadratic form along with the associated constraint conditions. This transformation turns the trajectory tracking control problem into an optimal control problem:

$$\min_{\boldsymbol{\Delta U}} \boldsymbol{J} = \frac{1}{2}\boldsymbol{\Delta U}^T\boldsymbol{H}\boldsymbol{\Delta U}+\boldsymbol{g}^T\boldsymbol{\Delta U} \qquad (17)$$

$$s.t.\ \begin{cases}\boldsymbol{U_{min}}+\boldsymbol{U_t} \le \boldsymbol{A_I}\boldsymbol{\Delta U_t} \le \boldsymbol{U_{max}}-\boldsymbol{U_t}\\ \boldsymbol{U_{min}} \le \boldsymbol{U_t}+\boldsymbol{A_I}\boldsymbol{\Delta U_t} \le \boldsymbol{U_{max}}\\ \boldsymbol{\Delta U_{min}} \le \boldsymbol{\Delta U_t} \le \boldsymbol{\Delta U_{max}}\\ \boldsymbol{c_{in}}(\boldsymbol{X_i},\boldsymbol{u_i}) \le 0\end{cases} \qquad (18)$$

where $\boldsymbol{U_t}$ represents the actual control variable at the previous time step;$\boldsymbol{U_{min}}$ and $\boldsymbol{U_{max}}$ denote the minimum and maximum values of the control variable, respectively;$\boldsymbol{\Delta U_{min}}$ and $\boldsymbol{\Delta U_{max}}$ represent the minimum and maximum values of the control increment, respectively. $\boldsymbol{c_{in}}(\boldsymbol{X_i},\boldsymbol{u_i}) \le 0$ denotes a vector-valued inequality constraint.

Following the aforementioned solution within each control cycle, only the first element of the control sequence is implemented on the system. In the next time step, the process is iteratively solved, enabling continuous control of the system. From this point, the designed AGV trajectory tracking system is primarily controlled and determined by the MPC (as shown in Figure 2), enabling the AGV to precisely follow the desired trajectory.

IV. EFFICIENT MPC OPTIMIZATION CONTROL

Regarding the AGV trajectory tracking control problem mentioned in the previous section, it is addressed by transforming it into an optimal control problem and employing the MPC method for resolution. However, in the process of MPC solving, the optimization problem involves a nonlinear problem with inequality constraints, which typically cannot be obtained through conventional methods. To address the challenges posed by inequality constraints in the MPC solving process, a method is proposed—namely, relaxing the inequality constraints in the objective function by introducing so-called soft constraints. Penalty terms are incorporated into the objective function to streamline the optimization problem-solving process. Specifically, to alleviate the complexity of the problem, a relaxation factor is introduced by easing the inequality constraints in the objective function (18). Subsequently, an efficient MPC algorithm is proposed to ensure that the control system can complete the optimization solution within finite computation time, facilitating real-time trajectory tracking control for stability.

A. CONSTRUCTING CONSTRAINT RELAXATION FACTORS

To address the constraints imposed by the inequality constraints, especially those denoted as $\boldsymbol{c_{in}}(\boldsymbol{X_i},\boldsymbol{u_i}) \le 0$ in (18), a constraint penalty term is introduced to address the inequality constraint approximately within the objective function. The penalty term associated with the inequality constraint is defined based on the switching mode of the constraint as follows:

$$\zeta\left(\boldsymbol{c_j}(\boldsymbol{X_i},\boldsymbol{u_i})\right) = \begin{cases}1, & \text{for } \boldsymbol{c_j}(\boldsymbol{X_i},\boldsymbol{u_i}) > 0\\ 0, & \text{for } \boldsymbol{c_j}(\boldsymbol{X_i},\boldsymbol{u_i}) \le 0\end{cases} \qquad (19)$$

In the equation, the $j-th$ row of $\boldsymbol{c_{in}}(\cdot,\cdot)$ represents the dimensions of the vector-valued inequality constraints $\boldsymbol{c_j}(\boldsymbol{X_i},\boldsymbol{u_i}), j = 1, 2, \cdots, n$, and n.

However, $\boldsymbol{c_j}(\boldsymbol{X_i},\boldsymbol{u_i}) = 0$ introduces non-differentiability. To address this issue, the problem can be resolved by approximating relaxation using the following function (19):

$$\begin{aligned}\zeta\left(\boldsymbol{c_j}(\boldsymbol{X_i},\boldsymbol{u_i})\right) &\approx sig\left(\boldsymbol{c_j}(\cdot,\cdot),\alpha\right)\\ &= \frac{1}{1+e^{-\boldsymbol{c_j}(\boldsymbol{X_i},\boldsymbol{u_i})\alpha}} \qquad (20)\end{aligned}$$

In this context, $sig(\cdot,\cdot)$ is the *sigmoid* function, with α determining the sharpness of the switching behavior. It is noteworthy that the *sigmoid* function can be effectively approximated by using a suitably large α (20). Subsequently, the updated formula for the new objective function with penalty terms is as follows:

$$\min_{\boldsymbol{\Delta U}} \boldsymbol{J} = \frac{1}{2}\boldsymbol{\Delta U}^T\boldsymbol{H}\boldsymbol{\Delta U}+\boldsymbol{g}^T\boldsymbol{\Delta U}+\beta\cdot\chi(\boldsymbol{X_i},\boldsymbol{u_i}) \qquad (21)$$

where $\chi(\boldsymbol{X_i},\boldsymbol{u_i}) = \sum_{j=1}^{n}\zeta\left(\boldsymbol{c_j}(\boldsymbol{X_i},\boldsymbol{u_i})\right)$. It is worth noting that when the upper limit of inequality

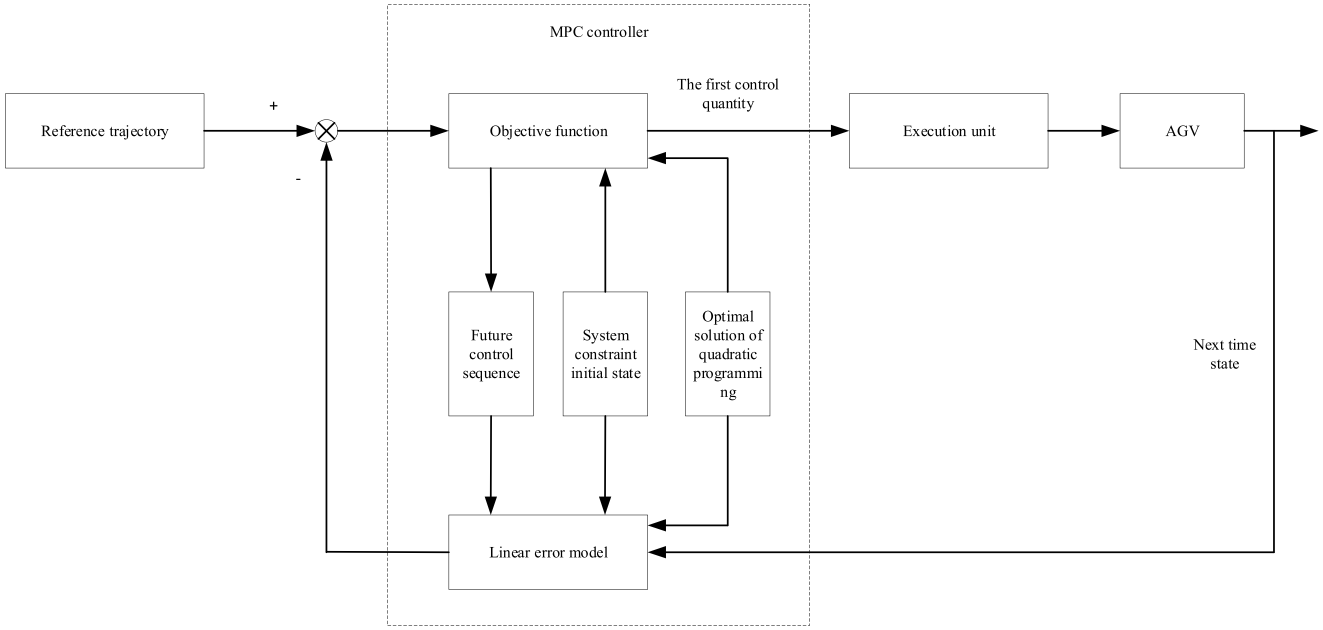

FIGURE 2. **AGV trajectory tracking control system diagram.**

$\boldsymbol{c_{in}}\left(\boldsymbol{X}_i, \boldsymbol{u_i}\right) \leq 0$ is reached, this inequality constraint can ensure that the constraints remain within the specified range, ensuring the existence of a solution during the calculation process. Additionally, different values for parameter β can also have an impact on the numerical convergence characteristics.

The above method for handling inequality constraints avoids situations where the inequality requirements are violated during the computation. Even in cases of slight violation of the inequality constraints, the new optimal control problem can still be successfully solved. This makes the method particularly suitable for robot applications with state constraints. Furthermore, it eliminates the need for additional optimization variables, thus maintaining a relatively low level in terms of problem dimensions and computational workload.

B. ESTABLISHING EFFICIENT MPC STABILITY CONSTRAINTS

In the preceding section, the primary emphasis was on resolving the matter of inequality constraints, without explicitly delving into the closed-loop stability of the MPC control method. To theoretically ensure this crucial property, stability constraints are introduced to constrain the new MPC optimization problem. By incorporating these stability constraints into the optimization problem, it is possible to ensure system stability while maintaining control performance. The formula for stability constraints is expressed as follows:

$$\frac{\partial V}{\partial \boldsymbol{X}} f\left(\boldsymbol{X}\left(\boldsymbol{t}\right), \boldsymbol{u}\left(\boldsymbol{t}\right)\right) \leq \frac{\partial V}{\partial \boldsymbol{X}} f\left(\boldsymbol{X}\left(\boldsymbol{t}\right), \boldsymbol{h}\left(\boldsymbol{X}\left(\boldsymbol{t}\right)\right)\right) \tag{22}$$

In the above expression, $V(\cdot)$ is the Lyapunov function and $\boldsymbol{h}(\cdot)$ is an auxiliary control law based on Lyapunov. As only the initial element of the optimal control sequence $\boldsymbol{U}^*$ is transmitted to the mobile robot, we solely account for stability constraints within the first prediction time $[t, t+\delta]$.

From the above, it can be concluded that the efficient MPC approach incorporates stability through auxiliary control laws. It ensures flexibility within specified ranges without compromising the stability of tracking control. In other words, when employing a cost-effective processor for the control algorithm, the prediction horizon can be tailored to judiciously diminish the size of real-time MPC optimization. By providing real-time feedback on the computation time t_p for each iteration, we apply such adjustments to Algorithm 1. This optimization measure enables the control system to flexibly adjust the prediction horizon based on the actual computational resource constraints, improving computational efficiency and meeting the requirements of real-time control. Here, ΔN represents the prediction horizon increment (or decrement), $\omega \in (0, 1)$ is the weighting factor, and t_p is the computation time for each iteration. This algorithm further reduces computational complexity while enhancing control performance.

C. STABILITY ANALYSIS

Diverging from conventional MPC, efficient MPC guarantees stability when the auxiliary control law $\boldsymbol{h}(\boldsymbol{X})$ is a viable solution, provided that certain conditions $\left\|\boldsymbol{h}(\boldsymbol{X})\right\|_{\infty} \leq u_{\max}$ can be satisfied. The formulation of the auxiliary control law $\boldsymbol{h}(\boldsymbol{X})$ is grounded in the principles of the Lyapunov stability theorem.

To analyze the stability of efficient MPC, the following Lyapunov function is designed:

$$V = \tfrac{1}{2}z_{\boldsymbol{1}}^{\boldsymbol{T}}z_{\boldsymbol{1}} + \tfrac{1}{2}z_{\boldsymbol{2}}^{\boldsymbol{T}}z_{\boldsymbol{2}} \tag{23}$$

Algorithm 1 Efficient MPC Algorithm

```
Initialization: Set simulation time $t_{sim}$, initial state $\boldsymbol{X}(t_0)$, set $k=0$, $t_0=0$, $\Delta \boldsymbol{X}=0$, $\omega$, $N$, $\Delta N$;
Convert (14) to (21);
Use the initial state $\boldsymbol{X}(0)$ to solve (21) and generate the optimal control sequence $\boldsymbol{U}^*$ at time $t=t_0$;
Implement the calculated control actions $\boldsymbol{u}(t)=\Delta\left(\boldsymbol{U}_0^*\right)$;
$k=k+1$, $t=k\delta$;
while $t \leq t_{sim}$ do
    Capture the value of $t_p$;
    If $t_p > \delta$ then set $N=N-\Delta N$;
    else if $t_p < \omega\delta$ then set $N=N+\Delta N$;
    else $N=N$;
    Sample the system state $\boldsymbol{X}(t)$;
    Calculate the state difference $\Delta \boldsymbol{X}=\boldsymbol{X}(k\delta)-\boldsymbol{X}((k-1)\delta)$;
    Generate $U_k^*$ by solving the updated $N$ through (21);
    Implement the calculated control action $\boldsymbol{u}(t)=\Delta\left(\boldsymbol{U}_k^*\right)$;
    $k=k+1$, $t=k\delta$;
end while
```

where $\boldsymbol{z_1}=\boldsymbol{X_r}-\boldsymbol{X}$ represents the position error, and $\boldsymbol{z_2}=\dot{\boldsymbol{X}}_{\boldsymbol{r}}-\boldsymbol{Bu}+\alpha_1\boldsymbol{z_1}$ represents the velocity error. Then, using the backstepping control (BC) technique, auxiliary control laws can be designed as follows:

$$\boldsymbol{h}(\boldsymbol{X})=\boldsymbol{G}^{-1}\boldsymbol{B}^{\boldsymbol{T}}(\boldsymbol{\theta})\boldsymbol{\mu}(\boldsymbol{X}) \tag{24}$$

where

$$\boldsymbol{\mu}(\boldsymbol{X})=\ddot{\boldsymbol{X}}_{\boldsymbol{r}}-\dot{\boldsymbol{B}}(\boldsymbol{\theta})\boldsymbol{u}+(\alpha_1+\alpha_2)\boldsymbol{z_2}+(1-\alpha_1^2)\boldsymbol{z_1} \tag{25}$$

where $\alpha_1, \alpha_2 > 0$ are the parameters of the controller, and for the convenience of the following proof, parameter $\boldsymbol{B}$ is is replaced with $\boldsymbol{B}(\boldsymbol{\theta})$. It is essential to highlight that when $\left\|\boldsymbol{h}(\boldsymbol{X})\right\|_\infty \leq u_{\max}$ is met, the stability of efficient MPC is assured. Consequently, the objective is to identify a feasible set of controller parameters.

Assumption 1: The desired position trajectory $\boldsymbol{p}(\boldsymbol{t})$ and its derivatives are both smooth and bounded, adhering to: $|x_r(t)| \leq \bar{x}$, $|y_r(t)| \leq \bar{y}$, $|\dot{x}_r(t)| \leq \bar{x}_1$, $|\dot{y}_r(t)| \leq \bar{y}_1$, $|\ddot{x}_r(t)| \leq \bar{x}_2$, $|\ddot{y}_r(t)| \leq \bar{y}_2$.

Lemma 1: The reference states of the system, denoted by $\boldsymbol{X_r(t)}$ and its first and second derivatives $\dot{\boldsymbol{X}}_{\boldsymbol{r}}\boldsymbol{(t)}, \ddot{\boldsymbol{X}}_{\boldsymbol{r}}\boldsymbol{(t)}$, are bounded. In other words, there exist positive constants $\bar{X}$, $\bar{X}_1$, and $\bar{X}_2$ such that for some $\left\|\boldsymbol{X_r(t)}\right\|_\infty \leq \bar{X}$, $\left\|\dot{\boldsymbol{X}}_{\boldsymbol{r}}\boldsymbol{(t)}\right\|_\infty \leq \bar{X}_1$, $\left\|\ddot{\boldsymbol{X}}_{\boldsymbol{r}}\boldsymbol{(t)}\right\|_\infty \leq \bar{X}_2$.

Note 1: Assuming a continuously smooth curve $\boldsymbol{p(t)}$, the fulfillment of Assumption 1 is straightforward. With Assumption 1, the proof of state boundedness concerning the reference can be established.

Theorem 1: Assume that the necessary conditions are met to validate Assumption 1. Given $\bar{g} = \left\|\boldsymbol{G^{-1}}\right\|_\infty$, that is, $\bar{g} = \max\left\{a_g^{-1}, b_g^{-1}, c_g^{-1}\right\}$. If the following conditions can be met:

$$\sqrt{2}\cdot\bar{g}(\bar{X}_2+2\sqrt{2}\cdot l^2+w) \leq u_{\max} \tag{26}$$

where $l = \bar{X}_1 + \|\boldsymbol{z_2}(t_0)\|_2 + \alpha_1\|\boldsymbol{z_1}(t_0)\|_2$ and $w = (\alpha_1+\alpha_2)\|\boldsymbol{z_2}(t_0)\|_2+\left(1-\alpha_1^2\right)\|\boldsymbol{z_1}(t_0)\|_2$, t_0 is the initial time, the efficient MPC scheme is consistently assured to possess a feasible solution.

Proof: Taking the infinity norm on both sides of (24), we derive the following inequality:

$$\left\|\boldsymbol{h}(\boldsymbol{X})\right\|_\infty \leq \left\|\boldsymbol{G^{-1}}\right\|_\infty \left\|\boldsymbol{B^T}(\boldsymbol{\theta})\right\|_\infty \|\boldsymbol{\mu}\|_\infty \tag{27}$$

$$\left\|\boldsymbol{B^T}(\theta)\right\|_\infty = \max\left\{|\sin(\theta)|+\cos(\theta), 1\right\} \leq \sqrt{2} \tag{28}$$

Likewise, a similar inequality is applicable to (25):

$$\|\boldsymbol{\mu}\|_\infty \leq \bar{X}_2 + \|\boldsymbol{\Omega}\|_\infty\|\boldsymbol{u}\|_\infty + (1-\alpha_1^2)\|\boldsymbol{z_1}\|_\infty + (\alpha_1+\alpha_2)\|\boldsymbol{z_2}\|_\infty \tag{29}$$

where

$$\boldsymbol{\Omega} = -\dot{\boldsymbol{B}}(\theta) = \begin{bmatrix} \sin(\theta)r & \cos(\theta)r & 0 \\ -\cos(\theta)r & \sin(\theta)r & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

In fact, $\|\boldsymbol{\Omega}\|_\infty \leq \sqrt{2}\|\boldsymbol{u}\|_\infty$.

According to (2), we can obtain:

$$\|\boldsymbol{u}\|_\infty = \left\|\boldsymbol{B^T}(\boldsymbol{\theta})\dot{\boldsymbol{X}}\right\|_\infty \leq \sqrt{2}\left\|\dot{\boldsymbol{X}}\right\|_\infty \tag{30}$$

Based on $\boldsymbol{z_1} = \boldsymbol{X_r} - \boldsymbol{X}$ and $\dot{\boldsymbol{z}}_{\boldsymbol{1}} = \boldsymbol{z_2} - \alpha_1\boldsymbol{z_1}$, we can conclude:

$$\left\|\dot{\boldsymbol{X}}\right\|_\infty = \left\|\dot{\boldsymbol{X}}_{\boldsymbol{d}} - \dot{\boldsymbol{z}}_{\boldsymbol{1}}\right\| \leq \bar{X}_1 + \|\dot{\boldsymbol{z}}_{\boldsymbol{1}}\|_\infty \leq \bar{X}_1 + \|\boldsymbol{z_2}\|_\infty + \alpha_1\|\boldsymbol{z_1}\|_\infty \tag{31}$$

As $\dot{V} \leq 0$, $\|\boldsymbol{z_1}(t)\|_\infty \leq \|\boldsymbol{z_1}(t)\|_2 \leq \|\boldsymbol{z_1}(t_0)\|_2$, and $\|\boldsymbol{z_2}(t)\|_\infty \leq \|\boldsymbol{z_2}(t)\|_2 \leq \|\boldsymbol{z_2}(t_0)\|_2$ are typically functions of $t_0 = 0$. Therefore, according to (31), we can obtain:

$$\|\mu\|_\infty \leq \bar{X}_2 + 2\sqrt{2}\cdot l^2 + w \tag{32}$$

where $l = \bar{X}_1 + \|\boldsymbol{z_2}(t_0)\|_2 + \alpha_1\|\boldsymbol{z_1}(t_0)\|_2$, $w = (\alpha_1 + \alpha_2)\|\boldsymbol{z_2}(t_0)\|_2 + (1-\alpha_1^2)\|\boldsymbol{z_1}(t_0)\|_2\circ$

Substituting (28), (30), and (32) into (27), we get:

$$\left\|\boldsymbol{h}(\boldsymbol{X})\right\|_\infty \leq \sqrt{2}\cdot\bar{f}(\bar{s}\cdot l + \bar{X}_2 + 2\sqrt{2}\cdot l^2 + w) \tag{33}$$

Certainly, the satisfaction of condition (26) implies the fulfillment of $\left\|\boldsymbol{h}(\boldsymbol{X})\right\|_\infty \leq u_{\max}$, thereby ensuring that the efficient MPC scheme, as defined in (22), consistently provides feasible solutions adhering to the constraints.

Note 2: Theorem 1 provides criteria for ensuring $\left\|\boldsymbol{h}(\boldsymbol{X})\right\|_\infty \leq u_{\max}$, offering guidelines for an appropriate choice of α_1 and α_2. To facilitate obtaining possible optimal solutions, it is desired to minimize α_1 and α_2 as much as possible to maximize the allowable operational region. While smaller values of α_1 and α_2 may result in slower convergence, the use of efficient MPC ensures optimal control performance relative to a certain performance metric.

It is important to note that stability constraints are introduced to guarantee closed-loop stability when the controller parameters satisfy condition (26). Following this, the stability of the proposed efficient MPC is established through Lyapunov's direct method.

Theorem 2: Assuming the problem possessed a singular optimal solution and optimal cost, both of which exhibit bounded solutions. Assume that Assumption 1 and the inequality (26) hold. With the auxiliary control law defined in (24), Algorithm 1 ensures asymptotic convergence to the desired trajectory.

Proof: For the Lyapunov candidate $V(X)$ in (23), which is continuously differentiable and radially unbounded, functions κ_∞ and $\beta_i(\cdot)$, $i = 1, 2, 3$ exist, satisfying the following inequalities according to the inverse Lyapunov theorem:

$$\beta_1(\|\boldsymbol{x}\|) \leq \boldsymbol{V}(\boldsymbol{X}) \leq \beta_2(\|\boldsymbol{x}\|) \tag{34}$$

$$\frac{\partial \boldsymbol{V}(\boldsymbol{X})}{\partial \boldsymbol{X}} f(\boldsymbol{X}, \boldsymbol{h}(\boldsymbol{X})) \leq -\beta_3(\|\boldsymbol{x}\|) \tag{35}$$

Taking into account the stability constraint (22) and the control $\boldsymbol{U}(\boldsymbol{t}) = \boldsymbol{U}^*(s)$, $s \in [t, t+\delta]$ applied at each sampling instant in the efficient MPC control, the Lyapunov candidate $\boldsymbol{V}(\boldsymbol{X})$ in (35) adheres to the following inequality:

$$\frac{\partial \boldsymbol{V}(\boldsymbol{X})}{\partial \boldsymbol{X}} f(\boldsymbol{X}, \boldsymbol{u}) \leq \frac{\partial \boldsymbol{V}(\boldsymbol{X})}{\partial \boldsymbol{X}} f(\boldsymbol{X}, \boldsymbol{h}(\boldsymbol{X})) \leq -\beta_3(\|\boldsymbol{x}\|) \tag{36}$$

Hence, following the principles of the Lyapunov stability theorem, the demonstration of the proposed efficient MPC algorithm in this paper is concluded.

V. SIMULATION EXPERIMENT VERIFICATION AND ANALYSIS

In this section, we conducted a simulation analysis on the MATLAB platform to assess the precision and computational speed of the proposed efficient MPC algorithm for AGV trajectory tracking. A comparative analysis between the efficient MPC algorithm and the traditional MPC algorithm was carried out to showcase the effectiveness of the proposed approach. Tables 1 and 2 present the fundamental parameters of the AGV and the controller, respectively.

In the experiment, trigonometric functions were employed as the reference trajectory for the simulation analysis. The traditional MPC and efficient MPC algorithms were utilized to track the reference trajectory, and the corresponding deviation values, along with the computation time incurred during tracking, were obtained to assess the algorithm's performance. Figure 3 illustrates the trajectory tracking plot, while Figures 4 and 5 depict the deviations along the X and Y axes during the tracking process.

From Figure 3, it can be observed that there is a significant error fluctuation between 7-10 seconds, while the efficient MPC shows relatively stable control without drastic error fluctuations during the entire process. The tracking performance of the efficient MPC is notably superior to that of traditional MPC. To accurately compare the performance of each controller, the mean deviation and variance in the X and Y directions are calculated as metrics to evaluate the controller's tracking effectiveness. A higher mean deviation or variance indicates a greater deviation or dispersion of the trajectory, resulting in poorer tracking performance. Conversely, lower values suggest better tracking performance. The data for both controllers are presented in Table 3.

The table shows that the mean deviation of the X-axis and Y-axis for the efficient MPC controller is 74.7%

TABLE 1. Basic parameters of AGV.

Name	Numerical value
Vehicle weight/kg	3.3
Size/mm	260×250×187
Wheelbase/mm	176
Wheel radius/mm	50
Light load speed m/s	1.6
Battery voltage/v	12.6

TABLE 2. Basic parameters of controller.

Parameter	Numerical value
Prediction horizon N_p	4
Sampling period t / s	0.2
Weight matrix Q	[0.01,0.01,1e-5,1e-5, 1e-5,1e-5]
Weight matrix R	[1e-5,1e-5,1e-5, 1e-5,0.8,0.4]
Relaxation factor	0.05

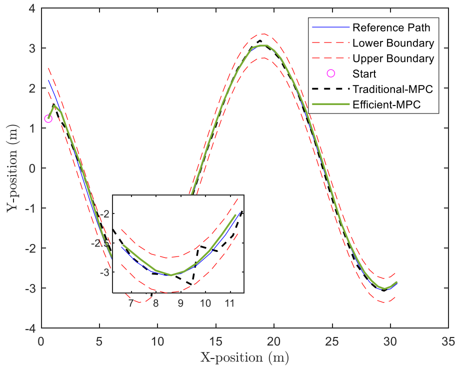

FIGURE 3. Trajectory tracking curve.

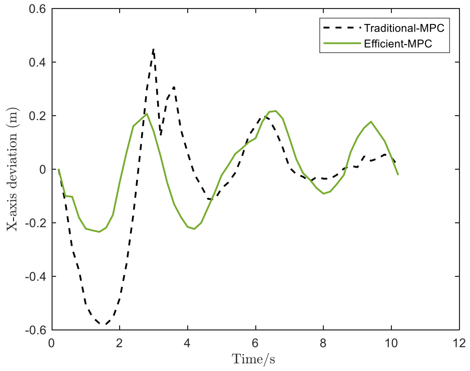

FIGURE 4. X-axis deviation curve.

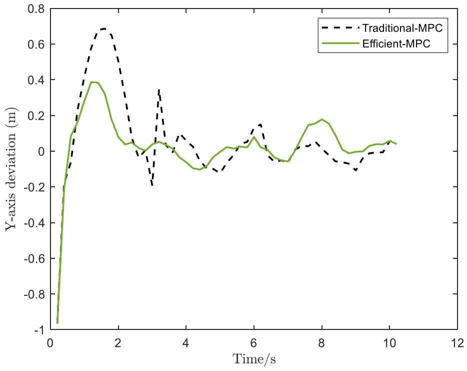

FIGURE 5. Y-axis deviation curve.

TABLE 3. Comparison of controller performance.

Controller	Mean deviation X/m	Mean deviation Y/m	Variance X/m^2	Variance Y/m^2
Traditional MPC	0.1597	0.1529	0.0308	0.0462
Efficient MPC	0.1193	0.0999	0.0049	0.0237

and 65.3% of those for the traditional MPC controller, respectively. The variances are 16% and 51.3% of those for the traditional MPC controller. Simulation results demonstrate that the efficient MPC controller exhibits better control performance compared to the traditional MPC controller.

Figures 6-8 represent the acceleration control chart, steering angle chart, and controller computation time chart, respectively. From Figure 6, it can be observed that the accelerations of both controllers meet the requirements under constraints. However, Figure 7 shows that the efficient MPC controller initially violates the constraint conditions, but the presence of constraints ensures that it always stays within the specified range. In contrast, the traditional MPC controller exhibits multiple instances of constraint violation. Compared to the traditional MPC controller, the efficient MPC controller demonstrates better stability.

From Figure 8, it is evident that the computation time used by the efficient MPC controller is significantly less than the time used by the traditional MPC controller. As shown in Table 4, the average computation time for the efficient MPC controller is 0.9762s, with a maximum single computation time of 1.4314s. This represents an improvement compared to the traditional MPC controller, which has an average computation time of 1.0208s and a maximum single computation time of 1.6593s.

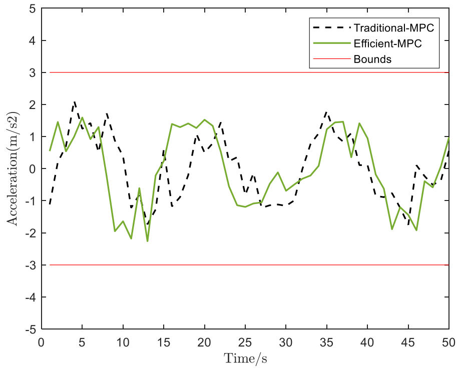

FIGURE 6. Acceleration control diagram.

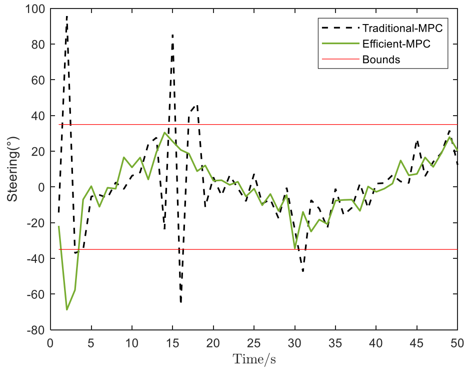

FIGURE 7. Steering angle diagram.

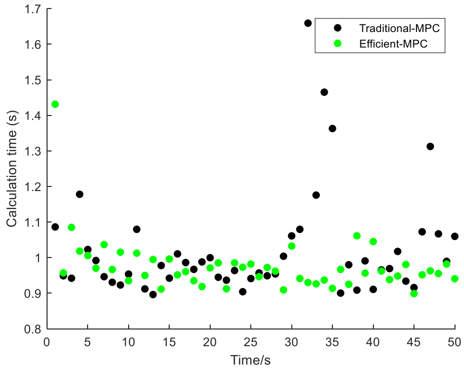

FIGURE 8. Calculation time diagram.

TABLE 4. Comparison of controller computation time.

	Traditional MPC	Efficient MPC
Average calculation time (s)	1.0208	0.9762
Maximum computation time (s)	1.6593	1.4314

The simulation results above highlight that the efficient MPC algorithm effectively minimizes the solution time of the model predictive controller. The efficient MPC algorithm ensures, under the premise of trajectory tracking accuracy, a 5.1% reduction in average solution computation time and a 13.7% reduction in the maximum single computation time compared to the traditional MPC algorithm. This validates the feasibility of the algorithm.

VI. CONCLUSION

In conclusion, we propose an efficient MPC algorithm tailored for Mecanum wheel AGV trajectory tracking, achieving effective and stable control. A linear error model based on pose error is established, and an optimization objective function is designed. Comparative analysis in MATLAB demonstrates the algorithm's superiority over traditional MPC, reducing trajectory tracking error and enhancing accuracy by 5.1%. Stability under constraints is notably improved, and average computation time is reduced. This ensures ideal trajectory tracking, meeting real-time requirements.

Our investigation into multi-vehicle coordinated control for AGV systems sets the stage for future research. We plan to extend the efficient MPC algorithm for coordinated motion among multiple AGVs, enhancing system efficiency. Additionally, integrating emerging technologies like deep learning and reinforcement learning with the efficient MPC algorithm holds promise for elevating system intelligence and adaptability, addressing dynamic changes and diverse task requirements. By delineating these future directions, we emphasize the foresight of our research and anticipate robust interest from academic and industrial communities.

REFERENCES

[1] J. J. Hao, "Automatic warehousing design and the application foreground," Univ. Sci. Technol. China, 2015.

[2] Q. Jia, M. L. Wang, and S. Q. Liu, "A review of omni-directional mobile robots," *Manuf. Automat.*, vol. 37, no. 7, pp. 131–134, 2015.

[3] L. Masar, "Design and control of a quasi-omnidirectional mobile robot FAAK," in *Proc. IASTED Int. Conf. Robot. Appl.*, 2009, pp. 391–397.

[4] X. S. Wang, "Mecanum wheel omni-directional mobile robot principles and applications," Nanjing Southeast Univ. Press, Tech. Rep. 201806. 194, 2018.

[5] Y. Li, G. Jiang, and J. F. Yang, "Research on modeling method of Mecanum wheel structure characteristic parameter," *Mech. Des. Manuf.*, vol. 11, pp. 245–248&252, Jan. 2018, doi: 10.19356/j.cnki.1001-3997.2018.11.063.

[6] P. Zhang, Z. J. Niu, and L. N. Wang, "Research on fuzzy PI control of omnidirectional mobile robot," *Comput. Simul.*, vol. 38, no. 10, pp. 353–360, 2021.

[7] P. Zhang and C. Y. Li, "Research on omnidirectional handling robot path tracking control based on improved fuzzy PID," *Food Machinery*, vol. 37, no. 6, pp. 114–119&190, 2021.

[8] X. Lu, X. Zhang, G. Zhang, J. Fan, and S. Jia, "Neural network adaptive sliding mode control for omnidirectional vehicle with uncertainties," *ISA Trans.*, vol. 86, pp. 201–214, Mar. 2019.

[9] M. L. Jiang, D. Q. Zhao, and Y. F. Sun, "Adaptive sliding mode controller design for Mecanum-wheeled mobile vehicle," *Manuf. Automat.*, vol. 44, no. 1, pp. 164–169, 2022.

[10] Y. Han and Q. Zhu, "Robust optimal control of omni-directional mobile robot using model predictive control method," in *Proc. Chin. Control Conf. (CCC)*, 2019, pp. 4679–4684.

[11] C.-L. Hwang, C.-C. Yang, and J. Y. Hung, "Path tracking of an autonomous ground vehicle with different payloads by hierarchical improved fuzzy dynamic sliding-mode control," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 2, pp. 899–914, Apr. 2018.

[12] D. B. Ren, S. M. Cui, and H. Z. Wu, "Preview control for lane keeping and its steady-state error analysis," *Automot. Eng.*, vol. 16, no. 38, pp. 192–199, 2016, doi: 10.19562/j.chinasae.qcgc.2016.02.010.

[13] T.-D. Do, M.-T. Duong, Q.-V. Dang, and M.-H. Le, "Real-time self-driving car navigation using deep neural network," in *Proc. 4th Int. Conf. Green Technol. Sustain. Develop. (GTSD)*, Nov. 2018, pp. 7–12.

[14] M.-T. Duong, T.-D. Do, and M.-H. Le, "Navigating self-driving vehicles using convolutional neural network," in *Proc. 4th Int. Conf. Green Technol. Sustain. Develop. (GTSD)*, Nov. 2018, pp. 607–610.

[15] T.-D. Phan, T.-T.-N. Nguyen, M.-T. Duong, C.-T. Nguyen, H.-A. Le, and M.-H. Le, "A steering strategy for self-driving automobile systems based on lane-line detection," in *Proc. 6th Int. Conf. Green Technol. Sustain. Develop. (GTSD)*, Jul. 2022, pp. 724–730.

[16] E. Kim, J. Kim, and M. Sunwoo, "Model predictive control strategy for smooth path tracking of autonomous vehicles with steering actuator dynamics," *Int. J. Automot. Technol.*, vol. 15, no. 7, pp. 1155–1164, Dec. 2014.

[17] J. Q. Yan, C. D. Lu, and Y. J. Cai, "Research on trajectory tracking of omnidirectional mobile robot based on integral model predictive control," *High Technol. Lett.*, vol. 31, no. 10, pp. 1081–1089, 2021.

[18] Y. Ren, L. Zheng, and W. Zhang, "A study on active collision avoidance control of autonomous vehicles based on model predictive control," *Automot. Eng.*, vol. 41, no. 4, pp. 404–410, 2019, doi: 10.19562/j.chinasae.qcgc.2019.04.007.

[19] B. Kouvaritakis, M. Cannon, and J. A. Rossiter, "Who needs QP for linear MPC anyway?" *Automatica*, vol. 38, no. 5, pp. 879–884, May 2002.

[20] G. Naus, R. van den Bleek, J. Ploeg, B. Scheepers, R. van de Molengraft, and M. Steinbuch, "Explicit MPC design and performance evaluation of an ACC stop-&-go," in *Proc. Amer. Control Conf.*, Jun. 2008, pp. 224–229.

[21] S. E. Li, Z. Jia, K. Li, and B. Cheng, "Fast online computation of a model predictive controller and its application to fuel economy-oriented adaptive cruise control," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1199–1209, Jun. 2015.

[22] A. Tuchner and J. Haddad, "Vehicle platoon formation using interpolating control: A laboratory experimental analysis," *Transp. Res. C, Emerg. Technol.*, vol. 84, pp. 21–47, Nov. 2017.

[23] Z. Q. Sun, "Study on tracking control of wheeled mobile robots: Model predictive control approach," Beijing Inst. Technol., 2021, doi: 10.26948/d.cnki.gbjlu.2018.000073.

[24] X. Y. Huang, "Research on Mecanum wheels AGV path planning and trajectory tracking algorithm," Zhejiang Univ. Sci. Technol., 2022, doi: 10.27840/d.cnki.gzjkj.2022.000227.

MIN TANG was born in China, in 1997. He received the bachelor's degree in mechanical engineering from Wenzhou University, in 2021. He is currently pursuing the master's degree in mechanical engineering with the Zhejiang Sci-Tech University, specializes in research and control systems. His research interests include the planning and tracking of mobile robot path.

SHUSEN LIN received the B.E. and Ph.D. degrees in vehicle engineering from the Nanjing University of Science and Technology, Nanjing, China, in 2008 and 2013, respectively. From 2013 to 2015, he was with the Department of Engineering College, Zhejiang Normal University, Jinhua, China. Since 2016, he has been with the Department of Mechanical Engineering, Taizhou University, Taizhou, China. His research interests include mechatronics, vehicle transmissions, and motion control.

YIXUAN LUO was born in China, in 2000. He received the bachelor's degree in automation from the School of Information Engineering, Xidian University, in 2023. He is currently pursuing the master's degree in intelligent manufacturing technology with Taizhou University, specializing in research and control systems. His research interests include embedded and motor control.