

Article

Fast Trajectory Tracking Control Algorithm for Autonomous Vehicles Based on the Alternating Direction Multiplier Method (ADMM) to the Receding Optimization of Model Predictive Control (MPC)

Ding Dong ¹, Hongtao Ye ^{1,2,3,*} , Wenguang Luo ^{1,3} , Jiayan Wen ^{1,3} and Dan Huang ⁴

¹ School of Automation, Guangxi University of Science and Technology, Liuzhou 545036, China; 221068297@stdmail.gxust.edu.cn (D.D.); wgluo@gxust.edu.cn (W.L.); wenjyan@gxust.edu.cn (J.W.)

² Guangxi Key Laboratory of Automatic Detecting Technology and Instruments, Guilin University of Electronic Technology, Guilin 541004, China

³ Guangxi Key Laboratory of Automobile Components and Vehicle Technology, Guangxi University of Science and Technology, Liuzhou 545036, China

⁴ School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510641, China; dan78huang@scut.edu.cn

* Correspondence: yehongtao@126.com; Tel.: +86-151-7774-1648

Abstract: In order to improve the real-time performance of the trajectory tracking of autonomous vehicles, this paper applies the alternating direction multiplier method (ADMM) to the receding optimization of model predictive control (MPC), which improves the computational speed of the algorithm. Based on the vehicle dynamics model, the output equation of the autonomous vehicle trajectory tracking control system is constructed, and the auxiliary variable and the dual variable are introduced. The quadratic programming problem transformed from the MPC and the vehicle dynamics constraints are rewritten into the solution of the ADMM form, and a decreasing penalty factor is used during the solution process. The simulation verification is carried out through the joint simulation platform of Simulink and Carsim. The results show that, compared with the active set method (ASM) and the interior point method (IPM), the algorithm proposed in this paper can not only improve the accuracy of trajectory tracking, but also exhibits good real-time performance in different prediction time domains and control time domains. When the prediction time domain increases, the calculation time shows no significant difference. This verifies the effectiveness of the ADMM in improving the real-time performance of MPC.

Keywords: trajectory tracking; model predictive control (MPC); alternately direction multiplier method (ADMM); quadratic programming; real-time performance



Citation: Dong, D.; Ye, H.; Luo, W.; Wen, J.; Huang, D. Fast Trajectory Tracking Control Algorithm for Autonomous Vehicles Based on the Alternating Direction Multiplier Method (ADMM) to the Receding Optimization of Model Predictive Control (MPC). *Sensors* **2023**, *23*, 8391. <https://doi.org/10.3390/s23208391>

Academic Editor: Srikanth Saripalli

Received: 22 August 2023

Revised: 10 October 2023

Accepted: 10 October 2023

Published: 11 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Autonomous driving is the main direction of development in the future automobile field. Its advantages have been demonstrated in many aspects and attracted wide attention and recognition from society. The main content of autonomous driving technology includes perception, decision-making, planning, control and other aspects [1]. Among them, trajectory tracking is the main content of the control level and one of the core technologies of autonomous vehicles, which determines the performance indicators such as the safety and comfort of vehicle operation [2].

Trajectory tracking is a fundamental function of autonomous vehicles that ensures that the vehicle travels along a preset path. Proportional integral derivative (PID) control, preview control theory, sliding mode control, deep learning, model predictive control (MPC), and other methods have provided numerous trajectory tracking strategies for autonomous vehicles [3–8]. MPC is widely used by researchers to address the trajectory tracking control

problems for intelligent vehicles. MPC is characterized by predictive modeling, receding optimization, feedback correction, and the ability to handle constrained optimization problems [9–13]. Aiming at the trajectory tracking problem of distributed drive vehicles, based on the MPC and vehicle dynamics model, a method combining the trajectory tracking preview time control and differential torque control based on the reference angle has been proposed, which improved the trajectory tracking performance [14]. In order to improve the tracking ability of autonomous vehicles, fuzzy control is used in combination with MPC to improve the weight of the cost function in MPC through the fuzzy adaptive algorithm. This not only improves the accuracy of trajectory tracking, but also improves the steering smoothness of the vehicle [15]. Using the Kalman filter algorithm combined with the MPC controller, the online estimation of the vehicle's nonlinear curvature response can reduce the wear and tear of vehicle components while ensuring comfort [16]. Feedforward compensation is integrated with MPC. The error from the expected path is calculated using a pure tracking algorithm as the input for feedback tracking control. The control output is then determined through MPC calculations, resulting in improved accuracy and robustness in trajectory tracking [17]. The weights of MPC are dynamically adjusted using a PSO-BP neural network, which improves the tracking performance of the autonomous vehicle under different speeds and road curvatures [18]. A vehicle model with steering dynamics is proposed. The cascaded MPC structure is used to separate the steering system of the vehicle dynamics model from the trajectory tracking controller. The simulation and test results show better performance than only considering the vehicle dynamics. However, integrating the steering system into the dynamics model can achieve optimal performance, but leads to higher computational requirements [19]. The above methods improve the tracking accuracy, but do not consider the complexity of MPC calculation.

MPC has significant advantages in solving trajectory tracking problems with various constraints. Extensive research has also been conducted to study the accuracy, stability, and robustness of vehicle trajectory tracking based on MPC [20,21]. However, MPC has a large online computation workload and low real-time performance. In addition, if the controlled model is too complex, it will significantly increase the MPC online iterative calculation, which will hinder the application of MPC in practice.

Therefore, improving the real-time performance of MPC has received increasing attention, and many research results have emerged [22,23]. A dynamic optimization toolkit can improve the computational speed and reduce the computation time of MPC [24,25]. A genetic algorithm is used to compute the optimal time domain parameters for real-time vehicle speed and road conditions, which improves the real-time performance of the controller [26]. But, it uses a relatively simple kinematic model. To improve the real-time performance of vehicle longitudinal speed planning, a nonlinear space-domain MPC (SMPC) is proposed to accelerate the nonlinear SMPC computation by generating thermal initialization and subsequently forming SMPC-RTI. However, it considers the longitudinal motion of the vehicle as well as the energy it saves [27]. In [28], the real-time performance and accuracy of vehicle trajectory tracking are improved by reducing the number of control steps or reducing the control frequency domain. Reducing the control frequency domain can meet the real-time requirements but may result in a slightly higher error compared to reducing the number of control steps. Using the linear parameter time-varying MPC and designing a linear quadratic regulator can reduce the computational cost of the dynamic control of the vehicle [29]. Alternating the direction multiplier method (ADMM) is favored by many researchers because of its good scalability. It has been successfully applied to machine learning, distributed computing, and other fields [30–32]. ADMM can divide the optimization variables into two parts and solve them in a separate framework, which can reduce the computational burden caused by the large scale of the system. In [33], a class of ADMM with nonlinear equality constraints is empirically studied and its convergence is analyzed. To enhance the real-time performance of the iterative linear quadratic regulator (iLQR), an optimization is conducted using the ADMM. When employing a logarithmic barrier function, this algorithm can circumvent the feasibility requirements of the initial

trajectory during the first iteration, thereby expediting the optimization process [34]. It shows good real-time performance in different driving scenarios.

In summary, many methods have been proposed to improve the real-time performance of MPC, and some research results have been obtained. However, there are still some issues regarding improving the real-time performance of MPC for autonomous vehicle trajectory tracking. The combination of optimization algorithm and MPC can improve the real-time performance of MPC to a certain extent, but it will take up additional computational resources when solving the optimal parameters of MPC [34]. Simplifying the vehicle model can also improve the real-time performance of the controller, but it may not achieve ideal control effects in complex road conditions [28]. When using optimization algorithms or simplified models in conjunction with MPC controllers, the MPC problem is usually transformed into a quadratic programming (QP) problem [35]. The interior point method (IPM) or the active set method (ASM) are commonly used to solve the QP problem, but the ASM cannot utilize the sparsity of the MPC problem, and has to perform the active set operation at each iteration, which is suitable for the case of a small number of controls and constraints [36]. The IPM can utilize the sparsity of the MPC problem, but each iteration needs to solve the Karush–Kuhn–Tucker (KKT) system, and when the KKT system changes during the iteration, the solution process needs to decompose or inverse the matrix. When the number of constraints is large or the prediction time domain increases, the matrix dimension becomes larger and the computation time becomes longer [37]. This paper focuses on the problem of a long MPC solution time. Starting from the MPC problem-solving process, based on the framework of trajectory tracking control for autonomous vehicles, an auxiliary variable method is introduced to transform the quadratic problems in MPC into a separable structured optimization problem. The ADMM is used to solve it, and a decreasing penalty factor is used to ensure the convergence of the ADMM algorithm during the solution.

The organization of this article is as follows. The vehicle dynamic model and the MPC controller are presented in Section 2. The combination of the ADMM algorithm and MPC is introduced in Section 3. The simulation results of the controller are shown in Section 4. Finally, a brief conclusion is given in Section 5.

2. MPC-Based Trajectory Tracking Control

2.1. Vehicle Dynamics Model

To ensure that the vehicle follows its desired trajectory, the bicycle model in [38] is adopted to represent the vehicle dynamics. In this model, we do not consider the effects of suspension and aerodynamics, the state variables are defined as $\zeta(t) = [\dot{x}, \dot{y}, \psi, \dot{\psi}, Y, X]^T$, and the control input vector is the front wheel angle, defined as $u(t) = \delta_f$. The three-degrees-of-freedom dynamics model of the vehicle is shown in Figure 1. The x axis represents the longitudinal axis in the vehicle coordinate system, the y axis represents the lateral axis in the vehicle coordinate system, the xoy is the body coordinate system, and XOY is the ground coordinate system.

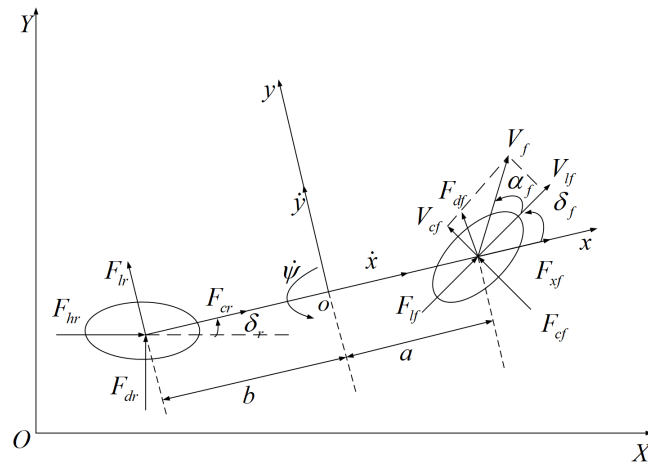


Figure 1. Vehicle dynamics model.

Differential equations for vehicle motion in lateral, longitudinal, and yawing directions are established based on Newton's second law, as given below:

$$\begin{cases} m\ddot{x} = m\dot{y}\dot{\psi} + 2F_{xf} + 2F_{xr} \\ m\ddot{y} = -m\dot{x}\dot{\psi} + 2F_{yf} + 2F_{yr} \\ I_z\ddot{\psi} = 2aF_{yf} - 2bF_{yr} \end{cases} \quad (1)$$

where m is the mass of the vehicle, I_z is the moment of inertia about the z axis, a and b are the distances between the center of mass and front/rear axles of the vehicle, ψ is the yaw angle of the vehicle, F_{xf} and F_{xr} represent the tire force on the longitudinal axis of the vehicle in the body coordinate system, and F_{yf} and F_{yr} represent the tire force on the transverse axis of the vehicle in the body coordinate system, respectively.

When the lateral acceleration is not greater than 0.4 g , the longitudinal force and lateral force exerted on the tire can approximate a linear relationship as given below:

$$\begin{cases} F_{lf} = C_{lf}s_f \\ F_{lr} = C_{lr}s_r \\ F_{cf} = -C_{cf}\alpha_f \\ F_{cr} = -C_{cr}\alpha_r \end{cases} \quad (2)$$

In Equation (2), C_{lf} and C_{lr} represent the longitudinal stiffness of the front and rear tires of the vehicle, C_{cf} and C_{cr} represent the cornering stiffness of the front and rear tires of the vehicle, s_f and s_r represent the slip ratio of the front and rear tires of the vehicle, α_f and α_r represent the cornering angle of the front and rear tires of the vehicle, respectively. Assuming that both front wheel angles of the vehicle are equally small, the lateral acceleration satisfies the small angle assumption, and in this case, the following approximate relationship can be used:

$$\begin{cases} \alpha_f = \frac{\dot{y} + a\dot{\psi}}{\dot{x}} - \delta_f \\ \alpha_r = \frac{\dot{y} - b\dot{\psi}}{\dot{x}} \end{cases} \quad (3)$$

Finally, consider the transformed relationship between the vehicle body coordinate system and the earth inertial coordinate system as follows:

$$\begin{cases} \dot{X} = \dot{x} \sin \psi - \dot{y} \cos \psi \\ \dot{Y} = \dot{x} \cos \psi + \dot{y} \sin \psi \end{cases} \quad (4)$$

From Equations (1)–(4), we can obtain the following nonlinear dynamic model of intelligent vehicles:

$$\begin{cases} m\ddot{x} = m\dot{y}\dot{\psi} + 2\left[C_{lfsf} + C_{cf}\left(\delta_f - \frac{\dot{y}+a\dot{\psi}}{\dot{x}}\right)\delta_f + C_{lrsr}\right] \\ m\ddot{y} = -m\dot{x}\dot{\psi} + 2\left[C_{cf}\left(\delta_f - \frac{\dot{y}+a\dot{\psi}}{\dot{x}}\right) + C_{cr}\frac{b\dot{\psi}-\dot{y}}{\dot{x}}\right] \\ I_z\ddot{\psi} = 2\left[aC_{cf}\left(\delta_f - \frac{\dot{y}+a\dot{\psi}}{\dot{x}}\right) - bC_{cr}\frac{b\dot{\psi}-\dot{y}}{\dot{x}}\right] \\ \dot{Y} = \dot{x}\sin\psi + \dot{y}\cos\psi \\ \dot{X} = \dot{x}\sin\psi - \dot{y}\cos\psi \end{cases} \quad (5)$$

It can be represented by the following differential equation:

$$\dot{\xi} = f(\xi(t), u(t)) \quad (6)$$

The state variables of the system are $\xi(t) = [\dot{x}, \dot{y}, \psi, \dot{\psi}, Y, X]^T$, and the control variable is selected as $u(t) = \delta_f$.

2.2. Trajectory Tracking Based on Model Predictive Control

The nonlinear vehicle dynamics model developed in Equation (6) is complex and takes a long computational time to solve, so it is linearized using Taylor's formula, omitting the higher-order terms except the first order. Assuming the Taylor expansion of Equation (6) at $[\xi_0, u_0]$, the resulting linear time-varying equation is given as follows:

$$\dot{\tilde{\xi}}(t) = A(t)\tilde{\xi}(t) + B(t)\tilde{u}(t) \quad (7)$$

where $A(t) = \frac{\partial f}{\partial \xi} \Big|_{\xi(i, u[i])}$, $B(t) = \frac{\partial f}{\partial u} \Big|_{\xi(i, u[i])}$ are the Jacobi matrices.

Discretizing the above state space equations using the forward Euler's method, the following equation can be obtained:

$$\begin{cases} \xi(k+1) = (TA(t) + I)\xi(k) + TB(t)u(k) + d_{k,t}(k) \\ d_{k,t}(k) = \xi_0(k+1) - (TA(t) + I)\xi_0(k) - TB(t)u_0(k) \end{cases} \quad (8)$$

From the above equations, the discrete linearized system can be shown:

$$\begin{cases} \xi(k+1) = A_{k,t}\xi(k) + B_{k,t}u(k) + d_{k,t}(k) \\ \eta(k) = C\xi(k) \end{cases} \quad (9)$$

where $A_{k,t} = I + TA(t)$, $B_{k,t} = TB(t)$, $C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T$, $d_{k,t}(k)$ is the discrete state error of the system at k -time, and T is the sampling time.

Taking the increment of the control variable, i.e., $\Delta u(k)$, as the input to the system reduces the effect of the sudden change in the control variable on the system, and therefore the original state vector needs to be augmented. Let the new state vector be $\tilde{\xi}(k) = [\xi(k), u(k-1)]^T$, we can obtain the new state space form as follows:

$$\begin{cases} \tilde{\xi}(k+1|t) = \tilde{A}_{k,t}\tilde{\xi}(k|t) + \tilde{B}_{k,t}\Delta u(k|t) + \tilde{d}(k) \\ \tilde{\eta}(k|t) = \tilde{C}\tilde{\xi}(k|t) \end{cases} \quad (10)$$

where $\tilde{A}_{k,t} = \begin{bmatrix} A_{k,t} & B_{k,t} \\ 0 & I \end{bmatrix}$, $\tilde{B}_{k,t} = \begin{bmatrix} B_{k,t} \\ I \end{bmatrix}$, $\tilde{d}_{k,t} = \begin{bmatrix} d_{k,t} \\ 0 \end{bmatrix}$, $\tilde{C} = [C \ 0]$.

The predicted output expression can be obtained as follows:

$$Y(t) = \Psi(t)\tilde{\xi}(t) + \Phi(t)\Delta U(t) + E(t)D(t) \quad (11)$$

$$\begin{aligned}
\text{where } Y(t) &= \begin{bmatrix} \tilde{\zeta}(t) \\ \tilde{\zeta}(t+1) \\ \vdots \\ \tilde{\zeta}(t+N_p) \end{bmatrix}, \Psi(t) = \begin{bmatrix} \tilde{C}\tilde{A}_{k,t} \\ \tilde{C}\tilde{A}_{k,t}^2 \\ \vdots \\ \tilde{C}\tilde{A}_{k,t}^{N_p} \end{bmatrix}, \Delta U(t) = \begin{bmatrix} u(t) \\ u(t+1) \\ \vdots \\ u(t+N_p) \end{bmatrix}, \\
D(t) &= \begin{bmatrix} \tilde{d}(t) \\ \tilde{d}(t+1) \\ \vdots \\ \tilde{d}(t+N_p-1) \end{bmatrix}, \Phi(t) = \begin{bmatrix} \tilde{C}\tilde{B}_{k,t} & 0 & \cdots & 0 \\ \tilde{C}\tilde{A}_{k,t}\tilde{B}_{k,t} & \tilde{C}\tilde{B}_{k,t} & \vdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{C}\tilde{A}_{k,t}^{N_p-1}\tilde{B}_{k,t} & \tilde{C}\tilde{A}_{k,t}^{N_p-2}\tilde{B}_{k,t} & \cdots & \tilde{C}\tilde{A}_{k,t}^{N_p-N_c-1}\tilde{B}_{k,t} \end{bmatrix}, \\
E(t) &= \begin{bmatrix} \tilde{C} & 0 & \cdots & 0 \\ \tilde{C}\tilde{A}_{k,t} & \tilde{C} & \vdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{C}\tilde{A}_{k,t}^{N_p-1} & \tilde{C}\tilde{A}_{k,t}^{N_p-2} & \cdots & \tilde{C} \end{bmatrix}.
\end{aligned}$$

The primary objective of the MPC controller is to minimize the disparity between the output and the reference value. This ensures precise vehicle tracking along the desired path while maintaining lateral stability. However, due to the intricacies of the vehicle dynamics model and associated constraints, situations may arise where a numerical solution cannot be obtained within a single control cycle. To address this, a relaxation factor, denoted by ϵ , is introduced. This factor guarantees that a viable solution can be obtained in each control cycle. The objective function is expressed as follows:

$$J(\tilde{\zeta}(t), u(t-1), \Delta U(t)) = \sum_{i=1}^{N_p} \left\| \tilde{\eta}(t+i|t) - \eta_{ref}(t+i|t) \right\|_Q^2 + \sum_{i=1}^{N_c-1} \left\| \Delta U(t+i|t) \right\|_R^2 + \rho \epsilon^2 \quad (12)$$

where N_p is the prediction time domain, N_c is the control time domain, ϵ is the relaxation factor, ρ is the weight of the relaxation factor, $\tilde{\eta}(t+i)$ represents the actual output of the system, $\eta_{ref}(t+i)$ represents the reference output, and $\Delta U(t+i)$ represents the increment of the forward turn angle.

According to Equation (12), the total cost of the objective function consists of three main parts. The first term relates to the cost associated with the deviation between the output trajectory and the reference trajectory. A greater deviation leads to a higher cost. The second term considers the cost associated with the forward corner increment. A larger forward corner increment during the tracking process results in a higher cost. Achieving a smoother control process is of the utmost importance when minimizing the control amplitude. The third term encompasses the relaxation factor and weight, ensuring the attainability of an executable solution for the objective function. When the vehicle is performing trajectory tracking, it is necessary to consider both the dynamic constraints of the vehicle itself and the limitations of the actuating mechanism. The constraints of vehicle dynamics are as follows:

$$\begin{cases} \Delta U_{\min} \leq \Delta U_t \leq \Delta U_{\max} \\ \Delta U_{\min} \leq A\Delta U_t + U_t \leq \Delta U_{\max} \\ y_{hc,\min} \leq y_{hc} \leq y_{hc,\max} \\ y_{sc,\min} - \epsilon \leq y_{sc} \leq y_{sc,\max} + \epsilon \\ \epsilon > 0 \end{cases} \quad (13)$$

where y_{hc} is the hard constraint output and y_{sc} is the soft constraint output.

Quadratic programming can solve the optimization problem of the MPC objective function; therefore, Equation (12) can be rewritten to the standard quadratic programming form, and the state variables and control variables constraint are introduced:

$$\min J = \frac{1}{2} \begin{bmatrix} \Delta U(t) \\ \varepsilon \end{bmatrix}^T H_t \begin{bmatrix} \Delta U(t) \\ \varepsilon \end{bmatrix} + G_t \begin{bmatrix} \Delta U(t) \\ \varepsilon \end{bmatrix} + P_t \quad (14)$$

$$s.t. \begin{bmatrix} \Delta U_{\min} \\ \varepsilon \end{bmatrix} \leq \begin{bmatrix} \Delta U \\ \varepsilon \end{bmatrix} \leq \begin{bmatrix} \Delta U_{\max} \\ \varepsilon \end{bmatrix}$$

$$\text{where } H_t = 2 \begin{bmatrix} \Phi_t^T Q \Phi_t & 0 \\ 0 & \rho \end{bmatrix}, G_t = \begin{bmatrix} 2E(t)^T Q \Phi_t & 0 \end{bmatrix},$$

$$Y_{ref} = \begin{bmatrix} \eta_{ref}(t) & \cdots & \eta_{ref}(t+N_p) \end{bmatrix}, M = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} \otimes I,$$

$$E(t) = \Psi_t \zeta(t) + E(t)D(t) - Y_{ref}(t), P_t = E(t)^T Q E(t) + U(t-1)^T R U(t-1) + \rho \varepsilon^2.$$

Solving Equation (14) in each control cycle yields the optimal sequence of control increments:

$$\Delta U_t^* = \begin{bmatrix} \Delta u_t^* & \Delta u_{t+1}^* & \cdots & \Delta u_{t+N_c-1}^* \end{bmatrix}^T \quad (15)$$

If the first item in the control sequence is added to the system, the current control variable is given as follows:

$$u(t) = u(t-1) + \Delta u_t^* \quad (16)$$

After the whole system enters the next cycle, the system repeats the above process and updates the control sequence to complete the trajectory tracking of the intelligent vehicle. MPC obtains the optimal control sequence through iterative optimization search, and the actual application of the algorithm is limited when the iteration process is long. The ADMM algorithm has the factorization of dual ascent and the global convergence of a multiplier method, which has received widespread attention in recent years due to its low computational complexity and simple algorithm structure. In this paper, it is used to solve the optimization problem in the trajectory tracking MPC of autonomous vehicles.

3. Implementation of ADMM Algorithm for Trajectory Tracking MPC Problem

3.1. Alternating Direction Method of Multipliers

The alternating direction multiplier method is generally used to solve constraint programming problems with the following equation:

$$\begin{cases} \min & f(z) + g(v) \\ s.t. & Cz + Dv = b \end{cases} \quad (17)$$

where f and g are convex functions, z and v are the variables to be optimized, and $C \in R^{p \times n}$, $D \in R^{p \times m}$, $b \in R^p$, $Cz + Dv = b$ are the linear equality constraints that the problem needs to satisfy.

By introducing the dual variable ω , the augmented Lagrangian function of the above equation is constructed as follows:

$$L_\rho(z, v, \omega) = f(z) + g(v) + \omega^T (Cz + Dv - b) + \frac{\rho}{2} \|Cz + Dv - b\|_2^2 \quad (18)$$

where $\rho > 0$ is the penalty parameter.

The iterative process of the ADMM algorithm includes three parts: iteratively updating the original variable z , iteratively updating the original variable v ; and updating the process of the dual variable ω . The update strategy is given as follows [39]:

$$\begin{cases} z^{k+1} = \arg \min_z L_\rho(z, v^k, \omega^k) \\ v^{k+1} = \arg \min_v L_\rho(z^{k+1}, v, \omega^k) \\ \omega^{k+1} = \omega^k + \rho(Cz^{k+1} + Dv^{k+1} - b) \end{cases} \quad (19)$$

In general, it is convenient to use the scaling dual variable $\mu = \omega/\rho$ to represent the iteration process:

$$\begin{cases} z^{k+1} = \arg \min_z \{f(z) + \frac{\rho}{2} \|Cz + Dv^k - b + \mu^k\|_2^2\} \\ v^{k+1} = \arg \min_v \{g(v) + \frac{\rho}{2} \|Cz^{k+1} + Dv - b + \mu^k\|_2^2\} \\ \mu^{k+1} = \mu^k + Cz^{k+1} + Dv^{k+1} - b \end{cases} \quad (20)$$

One advantage of the ADMM method is that it has only one parameter ρ , and under general conditions, the method can demonstrate convergence to all penalty factors. During the iterative process, the solution for the primal variables z and v are performed alternately, which reduces the scale of the problem and improves computational efficiency.

3.2. Model Predictive Controller Based on ADMM Improvement

Based on the previous section, this section provides a detailed explanation of the combination of the ADMM algorithm and trajectory tracking MPC problem. As shown in Figure 2, the ADMM algorithm is applied to the optimization and solving process in MPC. Based on the dynamic model of intelligent vehicles, the future outputs of the system are predicted using the state information of the vehicle and the control input, and then the feedback correction is performed using the actual output of the detection object. Finally, the ADMM algorithm is used to solve the optimization target online, and the current optimal control input is obtained.

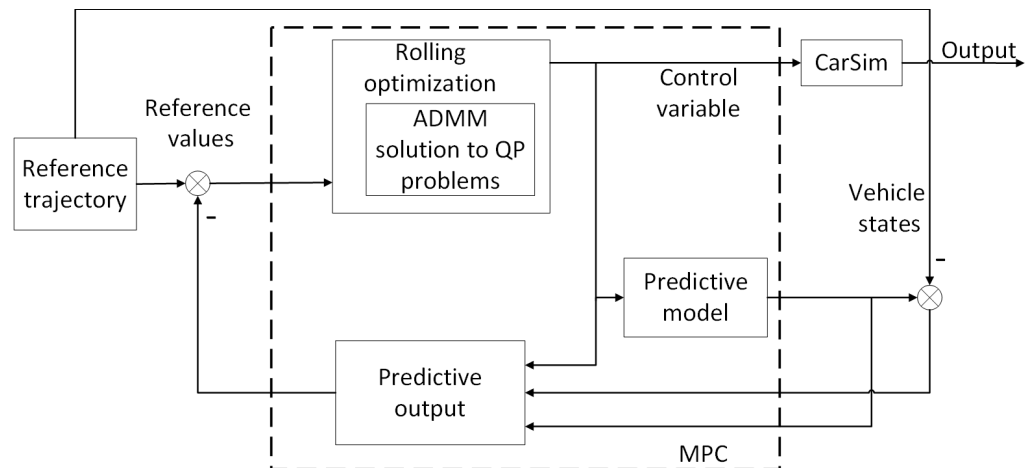


Figure 2. ADMM-MPC structural diagram.

To apply ADMM to the receding optimization process of MPC, the QP problem described by Equation (14) can be abbreviated as follows:

$$\begin{cases} \min J = \frac{1}{2}x^T Hx + f^T x \\ s.t. Ax \leq b \end{cases} \quad (21)$$

$$\text{where } H = 2 \begin{bmatrix} \Phi_t^T Q \Phi_t & 0 \\ 0 & \rho \end{bmatrix}, f = \begin{bmatrix} 2E(t)^T Q \Phi_t & 0 \end{bmatrix}^T, x = \begin{bmatrix} \Delta U \\ \varepsilon \end{bmatrix}, A = \begin{bmatrix} I & 0 \\ -I & 0 \\ \Phi & 0 \\ -\Phi & 0 \\ M & 0 \\ -M & 0 \end{bmatrix},$$

$$b = \begin{bmatrix} \Delta U_{\max} \\ -\Delta U_{\min} \\ Y(t) - \Psi(t)\xi(t) - E(t)D(t) \\ Y(t) + \Psi(t)\xi(t) + E(t)D(t) \\ U(t)_{\max} - U(t) \\ U(t) - U(t)_{\min} \end{bmatrix}.$$

After formulating the quadratic programming in standard form for addressing the trajectory tracking challenge in autonomous vehicles, it is imperative to incorporate auxiliary variables and subsequently reformulate the problem into a structure conducive to the ADMM algorithmic resolution:

$$\begin{cases} \min_{x,z} J = f(x) + g(z) \\ \text{s.t. } Ax - b + z = 0 \end{cases} \quad (22)$$

where $f(x) = \frac{1}{2}x^T Hx + f^T x$, $g(z)$ is the indicator function. $g(z)$ is defined as

$$g(z) = \begin{cases} 0, & \text{if } Az \leq b \\ +\infty, & \text{otherwise} \end{cases}.$$

According to the multiplier method, the augmented Lagrangian function for the optimization problem can be obtained as follows:

$$L_\rho(x, z, y) = f(x) + g(z) + y^T (Ax - b + z) + \frac{\rho}{2} \|Ax - b + z\|_2^2 \quad (23)$$

By introducing the dual scaling variable $\mu = y/\rho$, according to Equation (14), the iterative update process of the ADMM algorithm can be obtained as follows:

$$\begin{cases} x^{k+1} = \arg \min_z \left\{ \frac{1}{2}x^T Hx + f(x) + \frac{\rho}{2} \|Ax - b + z^k + \mu^k\|_2^2 \right\} \\ z^{k+1} = \arg \min_z \left\{ g(z) + \frac{\rho}{2} \|Ax^{k+1} - b + z + \mu^k\|_2^2 \right\} \\ \mu^{k+1} = \frac{1}{\rho} y^{k+1} = \frac{1}{\rho} [y^k + \rho(Ax^{k+1} - b + z^{k+1})] \\ = \mu^k + Ax^{k+1} - b + z^{k+1} \end{cases} \quad (24)$$

Compute the gradients of the original variable x and the auxiliary variable z , respectively. According to the first-order optimality condition, the iterative process formula is given as follows:

$$\begin{cases} x^{k+1} = (H + \rho A^T A)^{-1} [-f - \rho A^T (z^k + \mu^k - b)] \\ z^{k+1} = \max\{0, -Ax^{k+1} - \mu^k + b\} \\ \mu^{k+1} = \mu^k + Ax^{k+1} - b + z^{k+1} \end{cases} \quad (25)$$

To accelerate convergence, a relaxation factor of $\alpha \in [1, 2]$ is added, and the above iterative process can be obtained as follows:

$$\begin{cases} x^{k+1} = -(H + \rho A^T A)^{-1} [f + \rho A^T (z^k + \mu^k - b)] \\ z^{k+1} = \max\{0, -\alpha(Ax^{k+1} - c) + (1 - \alpha)z^k - \mu^k\} \\ \mu^{k+1} = \mu^k + \alpha(Ax^{k+1} - b + z^{k+1}) + (1 - \alpha)(z^{k+1} - z^k) \end{cases} \quad (26)$$

According to the first-order optimality condition of the QP problem, define the original residuals s_{prim} and dual residuals s_{dual} of (21), so we can obtain Equation (27):

$$\begin{cases} s_{prim}^{k+1} = Ax^k - b + z^k \\ s_{dual}^{k+1} = \rho A^T(z^{k+1} - z^k) \end{cases} \quad (27)$$

The algorithm convergence criterion is given according to the two residuals:

$$\begin{cases} \|s_{prim}^k\|_2 \leq \varepsilon^{prim} \\ \|s_{dual}^k\|_2 \leq \varepsilon^{dual} \end{cases} \quad (28)$$

where

$$\varepsilon^{prim} = \sqrt{n}\varepsilon^{abs} + \varepsilon^{rel} \max\{\|Ax^k\|_2, \|z^k\|_2, \|b\|_2\} \quad (29)$$

$$\varepsilon^{dual} = \sqrt{n}\varepsilon^{abs} + \varepsilon^{rel}\rho\|A^T x^k\|_2 \quad (30)$$

In general, $\varepsilon^{rel} = 10^{-3}$, ε^{abs} can be selected according to the required accuracy.

The update of the ADMM algorithm variables requires the use of the results of the previous moment, and the optimal solution of the previous moment ($d - 1$) is selected as the initial value of the algorithm at the current moment (d). The algorithm flow of the MPC solution control input Δu under trajectory tracking can be obtained.

- (1) Initialize the MPC parameter to obtain the system status information at the d th moment.
- (2) According to the equation of the state variables of the system and the input and output variables, the objective function is converted into a quadratic programming problem in the form of Equation (14).
- (3) Rewrite Equation (14) to form as Equation (22) conforms to the ADMM solution.
- (4) The optimal solution obtained at time $d - 1$ is used as the initial value of the solution to the time problem.
- (5) The variables are updated according to the iterative process of the ADMM algorithm, as shown in Equation (26).
- (6) According to Equations (27)–(30), to determine whether the iteration process meets the termination conditions, if it is met, stop the iteration, send the first term in the calculated optimal solution sequence to the control system as input, and enter step (7); if not, continue to iterate until the maximum number of iterations is reached.
- (7) Go to the next sampling moment $d + 1$, and repeat step (1).

4. Simulation

To verify the effectiveness of the proposed algorithm, a joint simulation platform of Simulink and CarSim was built to simulate and validate the designed controller. The ASM and the IPM are commonly used methods for solving traditional MPC problems, and this paper compares and analyzes the proposed algorithm with ASM and IPM. The processor parameters of the laptop used in the simulation are AMD Ryzen 7 5700U with Radeon Graphics 1.80 GHz. The vehicle parameters used in simulation are shown in Table 1.

4.1. Comparison of Controllers under the Same Prediction and Control Horizon ($N_p = 11$, $N_c = 6$)

In this section, the traditional MPC is solved by ASM and IPM, and the simulation results of the proposed controller were analyzed under the same prediction and control time domains. The simulation results are shown in Figures 4–6.

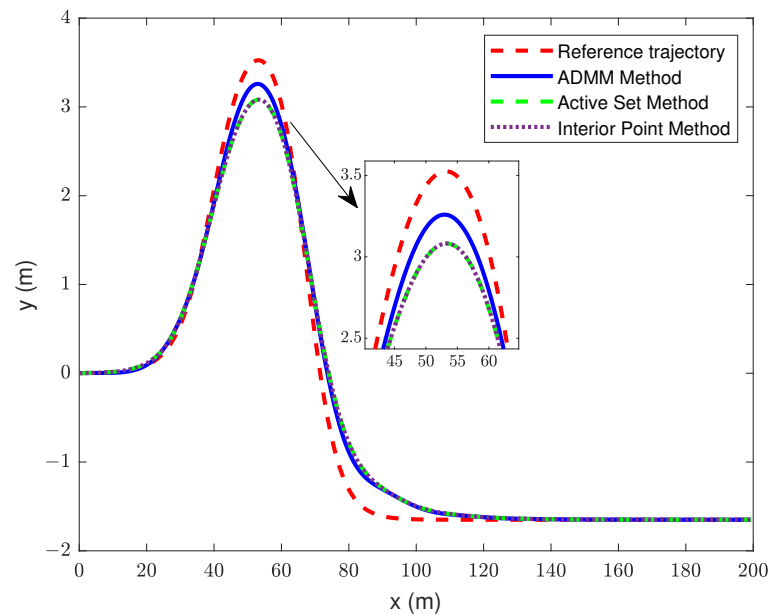


Figure 4. Trajectory tracking curve.

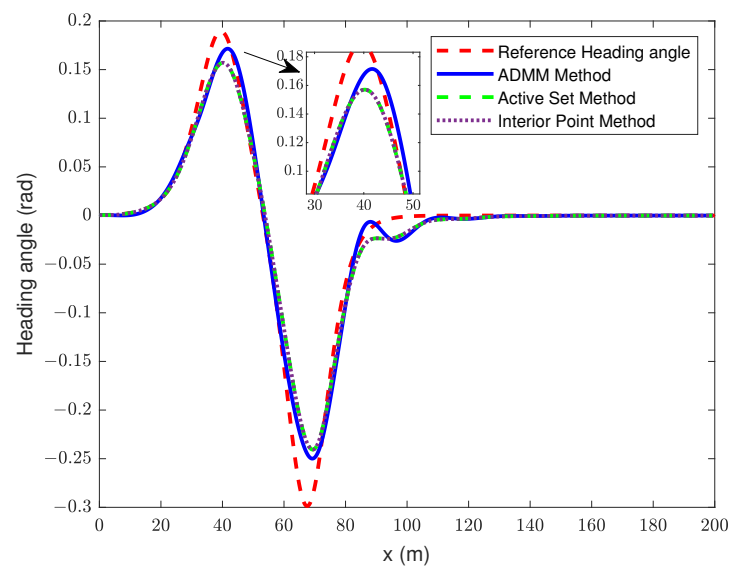


Figure 5. Changed curve of yaw angle.

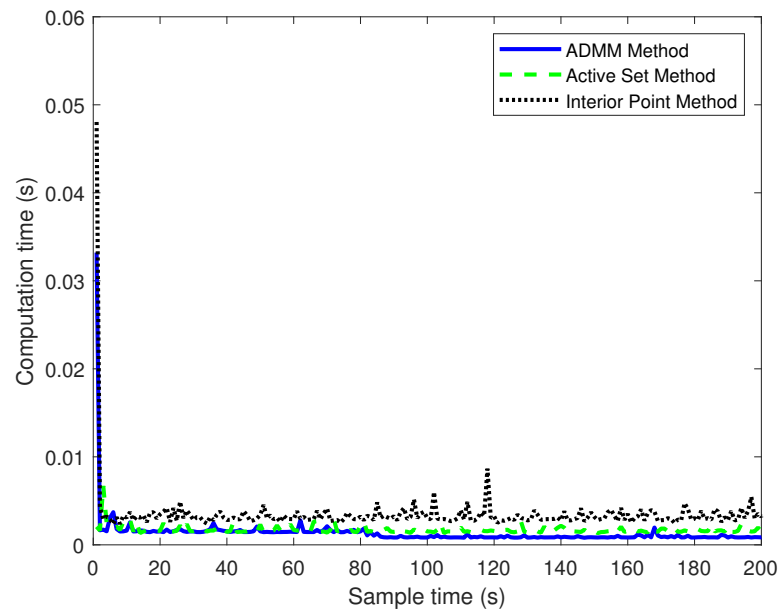


Figure 6. Computation time of different control methods.

From Figure 4, it can be seen that, during trajectory tracking, both the MPC with ADMM algorithm improvement and the traditional MPC controller can achieve good tracking effect. The tracking accuracy of the MPC with the ADMM algorithm improvement is higher during the control time. We can also obtain an improved MPC that shows a better tracking accuracy, the proposed controller has a tracking error of 0.266 m at the maximum lateral displacement, while the other two controllers have a tracking error of 0.443 m. During the whole control time, we used the trajectory data driven by the vehicle and the expected trajectory data to obtain the root mean square error (RMSE) of each controller. The RMSE for improved MPC is 0.193, and it is 0.231 for both other controllers. The RMSE formula is shown below:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(Y(i) - Y_{ref}(i) \right)^2} \quad (32)$$

where N represents the total simulation time divided by the sampling time, Y represents the actual trajectory of the vehicle, and Y_{ref} represents the expected trajectory.

Figure 5 reflects the tracking of the three controllers on the yaw angle. It can be seen that all controllers show a good tracking effect on the change of yaw angle. At the longitudinal position of 80 m, the MPC with the ADMM algorithm improvement has some overshoot in tracking the desired yaw angle, and there are some oscillations at the longitudinal position of 90 m. It is worth noting that the driving comfort may be reduced due to the change in the heading angle, but it can quickly track the reference yaw angle afterwards.

Figure 6 reflects the calculation performance of three controllers. The initial iteration of ADMM takes a longer computational time, this is because, in the first iteration, due to the presence of constraints, the computation of the variables needs to solve a large system of linear equations, which will take a long time, and the next step is carried out alternatively to greatly reduce the computational time. The average computation time is shown in Table 2.

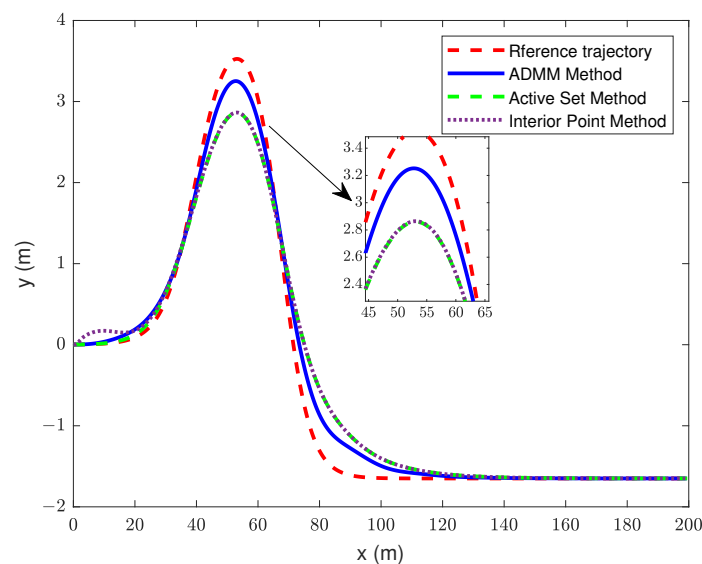
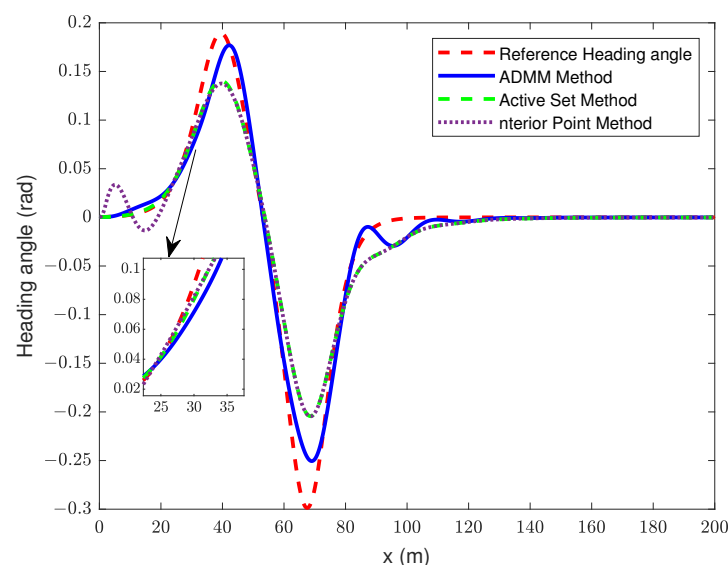
Table 2. Computation time of different control methods.

Method	ADMM	Active Set Method	Interior Point Method
Average computation time (s)	0.0013	0.0020	0.0035

As shown in Table 2, the average computation time of the MPC improved by the ADMM algorithm is 0.0013 s, the average computation time of the ASM is 0.0020 s, and the average computation time of the IPM is 0.0035 s. Compared with the ASM, the average computation time of the MPC improved by the ADMM algorithm is reduced by 35%. The real-time performance of the MPC improved by the ADMM algorithm is higher than that of the IPM, with an average computation time reduction of 62.8%.

4.2. Comparison of Controllers under Different Control Horizons

In this section of the simulation, the control time domain of the MPC based on ADMM improvement is set to 10, and the control time domain of the traditional MPC is 4. The simulation results are shown in Figures 7–9.

**Figure 7.** Trajectory tracking curve.**Figure 8.** Changed curve of yaw angle.

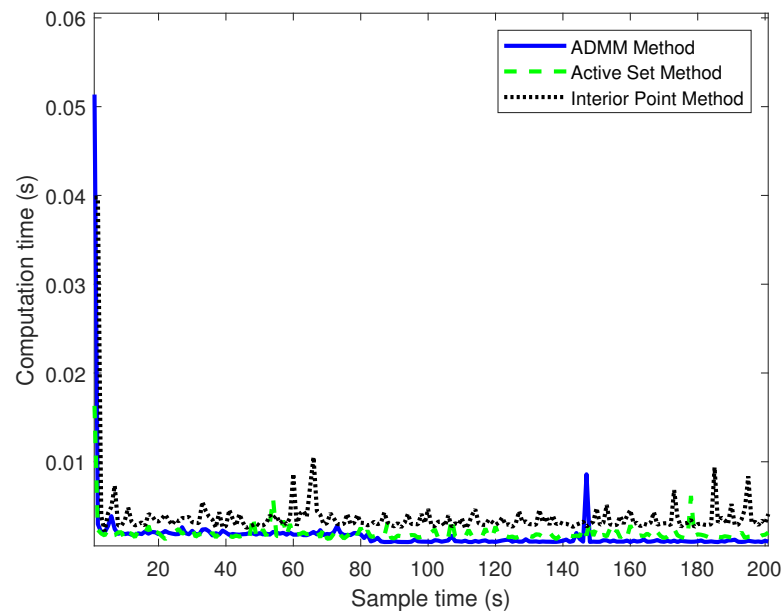


Figure 9. Computation time of different control methods.

By choosing a prediction time domain of $N_p = 14$, it can be seen from Figures 7 and 8 that the trajectory tracking accuracy of the improved MPC is significantly better than that of the IPM and ASM. The error of the improved MPC at the maximum lateral displacement is 0.273 m, while that of the IPM and ASM is 0.665 m. The tracking precision achieved by both the IPM and the ASM falls below the desired level, primarily due to the limited control time domain. Notably, IPM performs less effectively in tracking. The RMSE of improved MPC is 0.95, ASM is 0.317, and IPM is 0.319. The IPM causes the vehicle's front wheel angle to vary more in the 0–20 m range, which affects the stability, whereas the other two controllers show a smoother control performance.

Figure 9 reflects the computation time of the controller. It can be seen that the improved MPC algorithm takes a longer computation time in the first iteration, which is because it has a larger control time domain, which means that more dimensions of the control volume need to be solved in the iteration. Depending on the good performance of the ADMM algorithm, the average computation time of the improved MPC is still smaller than IPM and ASM. As can be seen from Table 3, the average computation time (0.0015 s) of the MPC improved by the ADMM algorithm is slightly lower than that of the ASM (0.0018 s). Compared with the IPM (0.0038 s), the average computation time of the MPC based on the ADMM algorithm is significantly reduced.

Table 3. Computation time of different control methods.

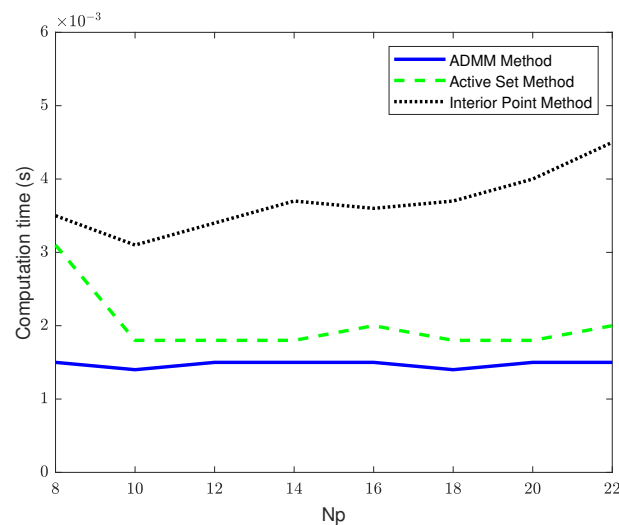
Method	ADMM	Active Set Method	Interior Point Method
Average computation time (s)	0.0015	0.0018	0.0038

4.3. Comparison of Controller Computation Time as the Prediction Horizon Increases

This section of simulation discusses the advantages and disadvantages of the improved MPC based on ADMM and traditional MPC in real-time performance under different prediction time domains. When the prediction time domains are set to 8, 10, 12, 14, 16, 20 and 22, the average computation time of the three controllers is compared as shown in Table 4, and the average solution time of the control sequence by the model based on ADMM improvement is basically unchanged in different prediction time domains, as shown in Figure 10.

Table 4. Comparison of controller computation time in different prediction time domains.

N_p	ADMM Computation Time (s)	Active Set Method Computation Time (s)	Interior Point Method Computation Time (s)
8	0.0015	0.0031	0.0035
10	0.0014	0.0018	0.0031
12	0.0015	0.0018	0.0034
14	0.0015	0.0018	0.0037
16	0.0015	0.0020	0.0036
18	0.0014	0.0018	0.0037
20	0.0015	0.0018	0.0040
22	0.0015	0.0020	0.0045

**Figure 10.** Comparison of the computation time of the three controllers in different prediction time domains.

From Figure 10, it can be seen that as the prediction horizon increases, the computation time of the MPC based on ADMM improvement is basically stable at 0.0015 s. Compared with the active set method, the maximum average computation time is reduced by 51.6%. Compared with the interior point method, as the prediction horizon increases, the computation time of the interior point method also increases. The maximum average computation time of the MPC based on ADMM improvement is reduced by 64.3%.

The simulation results indicate that the MPC improved by the ADMM algorithm can effectively reduce the online solving time of the traditional MPC. Under the same prediction time domain and control time domain, the average computation time is reduced by 35% compared to the ASM, and reduced by 62.8% compared to the IPM. When the same control time domain is used and the prediction time domain is changed, the average computation time of the MPC based on the improved ADMM method is smaller than that of the traditional MPC. Compared with the ASM, the computation time is reduced by up to 51.6%, and compared with the IPM, the computation time is reduced by up to 64.3%. Additionally, the proposed controller has higher tracking accuracy for double lane changes with some improvement in changing lanes.

5. Conclusions

This paper aims to address the problem of slow online solving and the low real-time performance of traditional MPC. A combination of ADMM and MPC is used to improve the real-time performance of trajectory tracking for autonomous vehicles. The ADMM algorithm is used to solve the QP problem transformed by MPC, and the ADMM algorithm is incorporated into the receding optimization of MPC. A relaxation factor α is added to accelerate the convergence.

To validate the effectiveness of the proposed method, we established a joint simulation platform within the Carsim-Simulink environment and conducted system simulations and tests in a double-shifted lane scenario. Regarding tracking accuracy, the algorithm demonstrates significant improvements. Numerical results indicate a reduction in the tracking error by 0.217 m at the maximum transverse displacement under identical prediction and control time domains, and by 0.392 m, at the maximum transverse displacement under varying control time domains. In terms of computation time, the proposed algorithm notably enhances the real-time performance of MPC when compared to previous methods. Numerical results reveal an average computation time reduction of 64.3% compared to IPM and 51.6% compared to ASM under different control time domains. Furthermore, the ADMM algorithm exhibits stability in the computation time compared to the IPM and ASM. As the prediction time domain increases, the computation time for MPC improved by the ADMM algorithm remains nearly constant, while the computation time for the ASM fluctuates significantly and the computation time for the IPM increases.

Future research may consider integrating the proposed algorithm with an optimization toolkit to further enhance its real-time performance. Additionally, this paper does not address longitudinal control for intelligent vehicles. Therefore, the proposed algorithm could potentially be applied to the combined horizontal and longitudinal control of autonomous vehicles in the future.

Author Contributions: Conceptualization, J.W.; formal analysis, H.Y.; data curation, W.L.; writing—original draft preparation, D.D.; writing—review and editing, D.H. and D.D.; supervision, W.L.; funding acquisition, H.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Guangdong Basic and Applied Basic Research Foundation (2021B1515420003), Guangxi Key Laboratory of Automatic Detecting Technology and Instruments (YQ20208), Guangxi Key Laboratory of Automobile Components and Vehicle Technology (2020GK-LACVTZZ01), and the National Natural Science Foundation of China (61963006).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the editor and reviewers for providing valuable review comments.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Chen, H.; Guo, L.; Gong, X.; Gao, B.; Zhang, L. Automotive control in intelligent era. *Acta Autom. Sin.* **2020**, *46*, 1313–1332.
- Yurtsever, E.; Lambert, J.; Carballo, A.; Takeda, K. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access* **2020**, *8*, 58443–58469. [[CrossRef](#)]
- Sun, C.; Zhang, X.; Zhou, Q.; Tian, Y. A Model Predictive Controller With Switched Tracking Error for Autonomous Vehicle Path Tracking. *IEEE Access* **2019**, *7*, 53103–53114. [[CrossRef](#)]
- Scheffe, P.; Henneken, T.M.; Kloock, M.; Alrifaa, B. Sequential Convex Programming Methods for Real-Time Optimal Trajectory Planning in Autonomous Vehicle Racing. *IEEE Trans. Intell. Veh.* **2023**, *8*, 661–672. [[CrossRef](#)]
- Shi, P.; Long, L.; Xuan, N.; Yang, A. Intelligent Vehicle Path Tracking Control Based on Improved MPC and Hybrid PID. *IEEE Access* **2022**, *10*, 94133–94144.
- Li, S.; Wang, G.; Zhang, B.; Yu, Z.; Cui, G. Vehicle Yaw Stability Control at the Handling Limits Based on Model Predictive Control. *Int. J. Automot. Technol.* **2020**, *21*, 361–370. [[CrossRef](#)]
- Qi, L.; Jiao, X.; Wang, Z. Trajectory Tracking Control of Intelligent Vehicle Based on DDPG Method of Reinforcement Learning. *China J. Highw. Transp.* **2021**, *34*, 335–348.
- Ducájú, J.M.S.; Llobregat, J.J.S.; Cuenca, Á.; Tomizuka, M. Autonomous Ground Vehicle Lane-Keeping LPV Model-Based Control: Dual-Rate State Estimation and Comparison of Different Real-Time Control Strategies. *Sensors* **2021**, *21*, 1531. [[CrossRef](#)]
- Kanchwala, H.; Bezerra, V.I.; Aouf, N. Cooperative path-planning and tracking controller evaluation using vehicle models of varying complexities. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2020**, *235*, 2877–2896. [[CrossRef](#)]
- Choi, W.; Nam, H.; Kim, B.; Ahn, C. Model Predictive Control for Evasive Steering of Autonomous Vehicle. *Int. J. Automot. Technol.* **2019**, *20*, 1033–1042.

11. Khan, S.; Guivant, J.; Li, X. Design and experimental validation of a robust model predictive control for the optimal trajectory tracking of a small-scale autonomous bulldozer. *Robot. Auton. Syst.* **2022**, *147*, 103903. [\[CrossRef\]](#)
12. Jiang, C.; Zhai, J.; Tian, H.; Wei, C.; Hu, J. Approximated Long Horizon MPC with Hindsight for Autonomous Vehicles Path Tracking. In Proceedings of the 2020 3rd International Conference on Unmanned Systems (ICUS), Harbin, China, 27–28 November 2020; pp. 696–701.
13. Mata, S.; Zubizarreta, A.; Pinto, C. Robust tube-based model predictive control for lateral path tracking. *IEEE Trans. Intell. Veh.* **2019**, *4*, 569–577. [\[CrossRef\]](#)
14. Tang, Z.; Xu, X.; Wang, F. Coordinated control for path following of two-wheel independently actuated autonomous ground vehicle. *IET Intell. Transp. Syst.* **2019**, *13*, 628–635. [\[CrossRef\]](#)
15. Wang, H.; Liu, B.; Ping, X.; An, Q. Path Tracking Control for Autonomous Vehicles Based on an Improved MPC. *IEEE Access* **2019**, *7*, 161064–161073. [\[CrossRef\]](#)
16. Pereira, G.C.; Wahlberg, B.; Pettersson, H.; Mårtensson, J. Adaptive reference aware MPC for lateral control of autonomous vehicles. *Cont. Eng. Pract.* **2023**, *132*, 105403. [\[CrossRef\]](#)
17. Nan, J.; Shang, B.; Deng, W.; Ren, B.; Liu, Y. MPC-based Path Tracking Control with Forward Compensation for Autonomous Driving. *IFAC-PapersOnLine* **2021**, *54*, 443–448. [\[CrossRef\]](#)
18. Tang, X.; Shi, L.; Wang, B.; Cheng, A. Weight Adaptive Path Tracking Control for Autonomous Vehicles Based on PSO-BP Neural Network. *Sensors* **2023**, *23*, 412. [\[CrossRef\]](#)
19. Domina, Á.; Tihanyi, V. Model Predictive Controller Approach for Automated Vehicle's Path Tracking. *Sensors* **2023**, *23*, 6862. [\[CrossRef\]](#)
20. Tan, Q.; Dai, P.; Zhang, Z.; Katupitiya, J. MPC and PSO Based Control Methodology for Path Tracking of 4WS4WD Vehicles. *Appl. Sci.* **2018**, *8*, 1000. [\[CrossRef\]](#)
21. Xu, F.; Zhang, J.; Hu, Y.; Qu, T.; Qu, Y.; Liu, Q. Lateral and longitudinal coupling real-time predictive controller for intelligent vehicle path tracking. *J. Jilin Univ. (Eng. Technol. Ed.)* **2021**, *51*, 2287–2294.
22. Shekhar, R.C.; Manzie, C. Optimal move blocking strategies for model predictive control. *Automatica* **2015**, *61*, 27–34. [\[CrossRef\]](#)
23. Leng, Y.; Zhao, S. Explicit Model Predictive Control for Intelligent Vehicle Lateral Trajectory Tracking. *J. Syst. Simul.* **2021**, *33*, 1177–1187.
24. Kayacan, E.; Kayacan, E.; Ramon, H.; Saeys, W. Learning in Centralized Nonlinear Model Predictive Control: Application to an Autonomous Tractor-Trailer System. *IEEE Trans. Control Syst. Technol.* **2014**, *23*, 197–205. [\[CrossRef\]](#)
25. Pas, P.; Schuurmans, M.; Patrinos, P. Alpaqa: A matrix-free solver for nonlinear MPC and large-scale nonconvex optimization. In Proceedings of the 2022 European Control Conference (ECC), London, UK, 12–15 July 2022; pp. 417–422.
26. Zhou, B.; Su, X.; Yu, H.; Guo, W.; Zhang, Q. Research on Path Tracking of Articulated Steering Tractor Based on Modified Model Predictive Control. *Agriculture* **2023**, *13*, 871. [\[CrossRef\]](#)
27. Ju, F.; Zong, Y.; Zhuang, W.; Wang, Q.; Wang, L. Real-Time NMPC for Speed Planning of Connected Hybrid Electric Vehicles. *Machines* **2022**, *10*, 1129. [\[CrossRef\]](#)
28. Bai, G.; Liu, L.; Meng, Y.; Liu, S. Real-time path tracking of mobile robot based on nonlinear model predictive control. *J. Agric. Mach.* **2020**, *51*, 47–52+60.
29. Alcalá, E.; Puig, V.; Quevedo, J. LPV-MPC Control for Autonomous Vehicles. *IFAC-PapersOnLine* **2019**, *52*, 106–113. [\[CrossRef\]](#)
30. Cheng, Z.; Ma, J.; Zhang, X.; Lee, T. Semi-Proximal ADMM for Model Predictive Control Problem with Application to a UAV System. In Proceedings of the 2020 20th International Conference on Control, Automation and Systems (ICCAS), Busan, Republic of Korea, 13–16 October 2020; pp. 82–87.
31. Li, X.; Tyagi, A. Block-Active ADMM to Minimize NMF with Bregman Divergences. *Sensors* **2023**, *23*, 7229. [\[CrossRef\]](#)
32. Toyoda, M.; Tanaka, M. An analysis of hot-started ADMM for linear MPC. *IET Control. Theory Appl.* **2021**, *15*, 1999–2016. [\[CrossRef\]](#)
33. Hong, M.; Lou, Z.; Razaviyayn, M. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, 19–24 April 2015; pp. 3836–3840.
34. Ma, J.; Cheng, X.; Zhang, X.; Tomizuka, M.; Lee, T. Alternating Direction Method of Multipliers for Constrained Iterative LQR in Autonomous Driving. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 23031–23042. [\[CrossRef\]](#)
35. Vroemen, B.; Essen, V.; Steenhoven, V.A.; Kok, J.J. Nonlinear model predictive control of a laboratory gas turbine installation. *J. Eng. Gas Turbine Power* **1999**, *121*, 629–634. [\[CrossRef\]](#)
36. Lau, M.S.K.; Yue, S.; Ling, K.; Maciejowski, J.M. A comparison of interior point and active set methods for FPGA implementation of model predictive control. In Proceedings of the European Control Conference (ECC), Budapest, Hungary, 23–26 August 2009; pp. 156–161.
37. Shahzad, A.; Kerrigan, E.C.; Constantinides, G.A. A warm-start interior-point method for predictive control. In Proceedings of the UKACC International Conference on Control 2010, Coventry, UK, 7–10 September 2010; pp. 1–6.

38. Lin, X.; Tang, Y.; Zhou, B. Improved Model Predictive Control Path Tracking Strategy Based an Online Updating Algorithm with Cosine Similarity and a Horizon Factor. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 12429–12438. [[CrossRef](#)]
39. Shan, R.; Li, Q.; He, F.; Feng, M.; Guan, T. Model predictive control based on ADMM for aero-engine. *J. Beijing Univ. Aeronaut. Astronaut.* **2019**, *6*, 8.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.