Brief Report

Research for Nonlinear Model Predictive Controls to Laterally Control Unmanned Vehicle Trajectory Tracking

Kegang Zhao ^{1,*} , Chengxia Wang ¹, Guoquan Xiao ¹, Haolin Li ¹, Jie Ye ² and Yanwei Liu ³

¹ National Local Engineering Laboratory of Automobile Parts Technology, South China University of Technology, Guangzhou 510641, China; 201821003959@scut.edu.cn (C.W.); megqxiao@scut.edu.cn (G.X.); 201721003640@scut.edu.cn (H.L.)

² School of Mechatronics Engineering, Foshan University, Foshan 528225, China; yejie@fosu.edu.cn

³ School of Electromechanical Engineering, Guangdong University of Technology, Guangzhou 510006, China; ywliu@gdut.edu.cn

* Correspondence: kgzhao@scut.edu.cn

Received: 3 August 2020; Accepted: 28 August 2020; Published: 31 August 2020

Featured Application: This work is used for the unmanned driving of vehicles and other devices.

Abstract: The autonomous driving is rapid developing recently and model predictive controls (MPCs) have been widely used in unmanned vehicle trajectory tracking. MPCs are advantageous because of their predictive modeling, rolling optimization, and feedback correction. In recent years, most studies on unmanned vehicle trajectory tracking have used only linear model predictive controls to solve MPC algorithm shortcomings in real time. Previous studies have not investigated problems under conditions where speeds are too fast or trajectory curvatures change rapidly, because of the poor accuracy of approximate linearization. A nonlinear model predictive control optimization algorithm based on the collocation method is proposed, which can reduce calculation load. The algorithm aims to reduce trajectory tracking errors while ensuring real-time performance. Monte Carlo simulations of the uncertain systems are carried out to analyze the robustness of the algorithm. Hardware-in-the-loop simulation and actual vehicle experiments were also conducted. Experiment results show that under $i7-8700$, the calculation time is less than 100 ms, and the mean square error of the lateral deviation is maintained at 10^{-3} m^2 , which proves the proposed algorithm can meet the requirement of real time and accuracy in some particular situations. The unmanned vehicle trajectory tracking method provided in this article can meet the needs of real-time control.

Keywords: trajectory tracking; nonlinear model predictive controls; collocation method; real-time

1. Introduction

The autonomous driving of unmanned vehicles is in a golden age of development. Trajectory tracking plays a pivotal role in autonomous driving, and tracking controllers are fundamental to the driving performance of unmanned vehicles. In recent years, the number of autonomous controllers has increased. The most widely used of these controllers are proportional-integral-derivative (PID) controllers, linear quadratic regulators (LQR), and model predictive control (MPC) controllers [1–5].

A PID controller is a negative feedback controller based on error compensation. It does not require complex modeling of the control object, and adjust the controller parameters to obtain a good tracking effect [6,7]. However, the parameters of the PID controller need to be determined through a large number of simulations, and have to be adjusted dynamically under different working conditions.

An LQR is based on state feedback, which can reduce steady-state lateral errors during curve driving [8]. Additionally, LQRs can perform real-time computations to handle emergency situations [9]. However, this depends on an accurate control model and requires a curvature continuity trajectory.

An MPC has the basic characteristics of a predictive model, rolling optimization, and feedback correction. It is especially suitable for situations where accurate mathematical models cannot be established and there are multiple constraints [10]. MPCs are more applicable than LQRs, which rely too much on models. In real driving processes, it is necessary to predict driving actions over a future horizon according to the surrounding environment, with complicated road and traffic conditions, even if the vehicle is driving manually. Neither PID nor LQR can realize this. However, MPC could realize the predictions needed for real-time controls with its basic characteristics of predicting several time steps in the future. Linear model predictive controls (LMPCs) was proposed and applied due to its simple linearization model and fast calculation speed [11]. Researchers establish kinematic models for different control object, and use LMPC to obtain good trajectory tracking effects [12,13]. In [14], a multilayer control system that includes three path tracking controllers was proposed based on LMPC, which can actively adjust the longitudinal velocity and use suitable path tracking controller to improve the real-time performance. In [15], a LMPC is designed to linearize the nonlinear vehicle model locally at each sampling point to reduce the computational complexity. In addition, many MPC algorithms for autonomous driving use linear predictive models based on approximate linearization [16]. A linear time-varying predictive model considering cornering characteristics was designed and optimized in [17]. Gallep et al. directly used a simplified vehicle linear kinematics model and verified the algorithm by a single-shift simulation [18]. However, since the control objects are generally nonlinear complex systems, the accuracy of approximate linearization is poor, and more and more system nonlinear factors are added to the consideration of MPCs [19]. Zietkiewicz et al. compared the control effects of nonlinear model predictive controls (NMPCs) and LMPCs, and the result shows that the LMPC algorithm are not satisfactory at high speeds or when target trajectory curvatures change rapidly [20]. Kayacan et al. compared results for LMPCs and NMPCs in simulations, showing that LMPC lateral errors fluctuated greatly at different speeds and that the performance of the LMPCs was lower than that of the NMPCs [21].

The solution complexity of NMPCs is closely related to the dimensions of the equations. Even if a simple dynamic model is used, multiple time steps need to be predicted, and the computational load will be large [22]. Thus, NMPCs remain challenging to use in real-time operations. But with the development of integrated chips and optimized methods, which have accelerated computing speeds, an increasing number of researchers are considering NMPCs for autonomous trajectory tracking. In [23,24], researchers compared NMPC autonomous steering to manual steering in simulations of lane change and return maneuvers, and verified the advantage of NMPCs. In [25] a method of varying discrete steps is adopted in NMPCs to compatible with path tracking and obstacle avoidance better. In [26], the vehicle model of a NMPC is formulated by approximating the magic tire model with a set of piecewise linear functions to reduce the computational burden in simulations. In [27,28], the optimization problem solution of NMPCs is calculated by the use of sequential quadratic programming (SQP) to improve solution efficiency. In addition to improve the calculation speed, literature [29] applies convex quadratic cone programming optimization numerical calculation method to the nonlinear coupling model, and there are methods such as “warm start” [30] and reduction of optimization process variables by assuming that the control variable is unchanged in a certain period of time [31]. In most of these studies, however, only simulations were performed, and calculation times were largely ignored.

NMPCs have been also widely used in other fields, which may provide some inspirations to our research. Researchers proposed applying a particle swarm optimization to the MPC algorithm for a quadcopter in trajectory tracking which produced fast solutions in [32]. A NMPC approach is applied to a sugar precipitation for Chinese medicine solution which significantly improves precipitation efficiency [33]. A Min–Max selector algorithm and MPC algorithm are compared for aircraft engine

fuel control in [34] and the MPC algorithm provides faster response. A NMPC scheme for the control of brushless direct current motor is presented in [35] and improves energy saving by 19%.

To solve the above problems, an NMPC based on the collocation method is proposed to the lateral control of unmanned vehicle trajectory tracking. The specific objective of this research is to ensure the accuracy of the NMPC algorithm and improve its real-time performance. The Runge-Kutta (RK) formula used in the collocation method in this paper is currently the main method for solving differential equations and has been widely used in various fields. The calculation process can be significantly simplified so that the computing time will be greatly reduced under the conditions where the initial value and derivative of the equation are given. The two-degrees-of-freedom kinematics equations for the vehicle, which are used by the MPC, also fit well with the requirements of the RK formula. And this paper uses Monte Carlo method to analyze the robustness of the algorithm which is a very mature sampling technique and can ensure the reliability of the experiment.

This paper is organized as follows. The second section introduces the theory of the NMPC. In the third section, a second-order RK formula is used to discretize the linear equations transforms the linear equations into a non-linear programming problem with constraints and performs Monte Carlo simulations. The fourth section describes the verification experiment. The results indicate that this method requires a low amount of calculations, with the calculation time for each solution step being within 100 ms. The fifth section presents the conclusions.

2. Nonlinear Model Predictive Control

2.1. Principles of Model Predictive Control

2.1.1. Predictive Model

The vehicle model is the basis of the MPC trajectory tracking controller. The model predicts future states from the current state information provided by the system, plus future steering control inputs. A detailed mathematical model will be introduced in Part B.

2.1.2. Rolling Optimization

At each sampling time, a control sequence that minimizes the objective function is calculated using the trajectory tracking controller. Only the first steering angle of the sequence will be applied to the CAN bus. At the next sampling time, a new sequence will be recalculated.

2.1.3. Feedback Correction

The objective function will include lateral deviations and heading angle deviations, which will be used to correct the steering control sequence and improve the trajectory tracking performance of the controller. This feedback correction provides the MPC controller with a strong ability to resist interference and overcome system uncertainty. The design architecture of the MPC trajectory tracking controller is shown in Figure 1. The controller obtains the initial state firstly, uses the predictive model and the target trajectory to get the objective function secondly, and finally calculate a control sequence to minimize the objective function at each optimization period.

Figure 1. Design architecture of the trajectory tracking controller.

2.2. Mathematical Model of Nonlinear System

2.2.1. System State Equation

A nonlinear system can be expressed using the following equation:

$$\begin{cases} \dot{X}(\tau) = f(X(\tau), u(\tau)) \\ X(0) = X_0 \\ Y(\tau) = h(X(\tau), u(\tau)) \end{cases} \quad (1)$$

where X is the system state, u is the system input, Y is the system output, $f(X, u)$ describes the dynamic system, and $h(X, u)$ defines the system output.

A future system state can be predicted using a numerical integration algorithm:

$$X(t+1) = g(X(t), u(t)) \quad (2)$$

where $g(X, u)$ is a numerical integration method.

In this paper, a vehicle kinematic model is employed to describe the system state.

2.2.2. Vehicle Kinematic Model

The research goal of this study is to ensure the accuracy and real-time performance of the vehicle while tracking its desired trajectory. To reduce lateral errors and improve control accuracy, it is necessary to establish a suitably simplified kinematic model. The research focuses on the longitudinal, lateral and yaw movement of the vehicle.

Therefore, the modeling is appropriately simplified shown in Figure 2, and the following assumptions were made as follows: Ignoring the effects of the steering system, using the steering angle of the front wheels as the steering input, ignoring the role of the suspension, ignoring the air resistance, and ignoring the impact of the ground tangential force on the tire. For convenience, the center of the vehicle's rear axis was used as the origin of the vehicle coordinate system.

The simplified model of an intelligent vehicle is as follows:

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = v \tan \delta / L \end{cases} \quad (3)$$

where x is the abscissa of the center of the rear axle of the vehicle, y is the ordinate of the center of the rear axle of the vehicle, θ is the yaw angle of the vehicle, δ is the steering wheel angle, L is the wheelbase of the vehicle, and v is the vehicle speed.

Figure 2. Vehicle kinematic model.

2.2.3. Objective Function

The goal of the MPC algorithm is to calculate a control sequence that can minimize the objective function. The objective function has a similar form to the following equation:

$$J = \sum_{k=1}^N L(X(t+k|t), Y(t+k|t), u(t+k|t)) \quad (4)$$

where $X(t+k|t)$ is a future system state, $Y(t+k|t)$ is a future system output, and $u(t+k|t)$ is a future system input.

The objective function of the trajectory tracking controller should reflect the ability of the controller to smoothly track the desired trajectory. At the same time, the objective function is expected to require a low amount of calculation, to avoid affecting the real-time performance of the algorithm. The objective function is expressed as

$$J(U) = k_1 \sum_{i=1}^n [y_i - T(x_i)]^2 + k_2 \sum_{i=1}^n [\theta_i - \arctan T'(x_i)]^2 + k_3 \sum_{i=1}^n u_i^2 \quad (5)$$

where $T(x)$ is the selected planning target trajectory, n indicates prediction horizon, k_1 , k_2 , and k_3 are the weight coefficients of the lateral deviations, yaw angle deviations, and control inputs, respectively. The first two items reflect the system's ability to track the desired trajectory, and the third item reflects the requirements for stably changing control inputs.

2.2.4. Constraints

In practical applications, there will always be various constraints, including equality and inequality.

The motor power for a real car steering system is limited, the rotation speed of the wheel corner cannot be arbitrarily large, and the wheel angle increment U must be limited for real-world application. At the same time, there are physical limits on the wheel angle of the vehicle, so a wheel angle limit constraint must be added to the prediction model. In summary, the additional constraints are as follows:

$$-\Delta\delta_{max} \leq U \leq \Delta\delta_{max} \quad (6)$$

$$-\delta_{max} - \delta_0 \leq CU \leq \delta_{max} - \delta_0 \quad (7)$$

where

$$\mathbf{U} = \begin{bmatrix} \Delta\delta_1 \\ \Delta\delta_2 \\ \vdots \\ \Delta\delta_{n-1} \\ \Delta\delta_n \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \cdots & 1 & 0 \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix} \quad (8)$$

$\Delta\delta_{max}$ is the maximum wheel angle increment, and δ_{max} is the maximum wheel angle.

3. Optimal Algorithm for NMPC

3.1. Discretization of Kinematic Model Based on the Collocation Method

The RK method is derived from the Taylor formula and uses a slope to express the differential. It is used to estimate the slopes of several points in the integration interval, and then to calculate a weighted average as the basis for the next point. This produces a higher accuracy numerical integration calculation method. In this study, we used a special form of second-order RK, the improved Euler method:

$$\begin{cases} y_{i+1} = y_i + h\left(\frac{1}{2}K_1 + \frac{1}{2}K_2\right) \\ K_1 = f(x_i, y_i) \\ K_2 = f(x_i + h, y_i + hK_1) \end{cases} \quad (9)$$

The vehicle kinematics equation comprises expressions of the derivatives of state parameters, which satisfies the conditions of using a second-order RK. Using Equation (9), the vehicle yaw angle and coordinates in Equation (8) are discretized. Assuming that the discretization period is T and the predicted time domain is n , we obtain

$$\theta_{i+1} = \theta_i + \frac{Tv}{2L}(\tan\delta_i + \tan\delta_{i+1}) \quad (10)$$

$$x_{i+1} = x_i + \frac{Tv}{2}(\cos\theta_i + \cos\theta_{i+1}) \quad (11)$$

$$y_{i+1} = y_i + \frac{Tv}{2}(\sin\theta_i + \sin\theta_{i+1}) \quad (12)$$

where $i = 0, 1, \dots, n-1$.

The vehicle's initial yaw angle θ_0 , initial wheel rotation angle δ_0 , initial abscissa x_0 , current ordinate y_0 , and control input set $\{\delta_1, \dots, \delta_{n-1}, \delta_n\}$ are known. Substituting these parameters into Equation (10), the numerical solution sequence for the vehicle yaw angle $\theta(t)$ at each node under the second-order RK method $\{\theta_1, \dots, \theta_{n-1}, \theta_n\}$ can be obtained. Substituting these parameters into Equations (11) and (12), a numerical solution sequence can also be obtained for the vehicle's abscissa $x(t)$ and ordinate $y(t)$.

The Algorithm 1 is designed according to the above calculation steps. The pseudo-code of the algorithm is as follows:

Algorithm 1: Second-order RK Collocation Method for Intelligent Vehicle Prediction Model.

Input: Control sequence $\{\delta_1, \dots, \delta_{n-1}, \delta_n\}$;
 Initial state value: $\{\theta_0, x_0, y_0\}$
 Output: state quantity matrix $\{\theta, X, Y\}$
 1: calculate $\frac{T_v}{2}, \frac{T_v}{(2L)}, \tan \delta_0$;
 2: for $i = 1 : n$
 3: calculate $\tan \delta_i$;
 4: calculate $\theta_i = \theta_{i-1} + \left(\frac{T_v}{(2L)}\right) \times (\tan \delta_{i-1} + \tan \theta_i)$
 5: end for
 6: calculate $\sin \theta_0$ and $\cos \theta_0$
 7: for $i = 1 : n$
 8: calculate $\sin \theta_i$ and $\cos \theta_i$
 9: calculate $x_i = x_{i-1} + \left(\frac{T_v}{2}\right) \times (\cos \theta_{i-1} + \cos \theta_i)$
 10: calculate $y_i = y_{i-1} + \left(\frac{T_v}{2}\right) \times (\sin \theta_{i-1} + \sin \theta_i)$
 11: end for

3.2. Solving Nonlinear Programming Problems

The collocation method is used to discretize the optimal control problems into nonlinear programming problems. When solving nonlinear programming problems, a solver named SNOPT based on a sequential quadratic programming (SQP) algorithm is used to perform successive approximations through a series of quadratic programming, which can efficiently solve large-scale nonlinear optimization problems.

3.3. Monte Carlo Simulations

Monte Carlo method is also called the random simulation method, is a powerful tool for system analysis and design. The principle is to randomly sample the variables that affect the performance index of the control system. The random sampling of each variable will generate a performance index value, and the probability distribution of the performance index can be obtained by repeating n times (such as 100).

Trajectory tracking algorithm always needs a vehicle model, but because the physical parameters, such as mass and tire lateral stiffness, cannot be accurately measured, we can only use a nominal value. However, when the algorithm is applied to a real vehicle, the actual value of vehicle parameters may differ from the nominal value selected in the algorithm, which will inevitably affect the control effect of the algorithm. For the above problems, Simulink blocks provided by Robust Control Toolbox are used to analyze the robustness of the algorithm. A vehicle model is established in Simulink, artificially limited the probability distribution of vehicle parameters. uncertain state space block and Matlab functions are used to generate random parameters conforming to the probability distribution. Finally, the algorithm is applied to the vehicle model. The robustness of the algorithm is analyzed by a large number of repeated simulation experiments. In this experiment, the experimental vehicle was asked to follow a dashed line, and one of the results is as shown in the Figure 3. The red line is the target trajectory and the black line is the actual trajectory. The mean square error of lateral position deviation was used as the evaluation index, and the experiment was repeated 100 times. As can be seen from the Figure 4, although the physical parameters change according to the probability distribution, the mean square deviation remains in a relatively stable state at about $1.0\text{--}1.3 \times 10^{-3} \text{ m}^2$. Therefore, the algorithm is excellent in robustness and has good prospects in practical applications.

Figure 3. Experiment result.

Figure 4. Frequency distribution histogram of mean square error.

4. Verification Experiment Results and Analysis

4.1. Hardware-In-The-Loop Experiment Platform Architecture

The hardware-in-the-loop (HiL) system used the simulation software Prescan and robot operating system (ROS). Prescan running on the host was used to build a simulation environment run in real-time. Prescan is widely used in physical simulation application platforms for automotive advanced assisted driving and autonomous driving systems. The tested machine employs a robot operating system, which has powerful functions, numerous interfaces, and strong portability, and has become a key tool for intelligent vehicle development.

An interface diagram of the HiL platform is shown in Figure 5. Prescan creates a simulated vehicle and simulates the characteristics of the controlled vehicle. The state variables X , Y , and θ are transmitted to the tested machine through the CAN bus. The software for the intelligent vehicle MPC trajectory tracking controller for is developed in C++. Real-time communication between the host and tested machine was carried out through the ROS bus and CAN bus.

Figure 5. Interface diagram of the hardware-in-the-loop platform.

For the hardware-in-the-loop experiment, an intelligent vehicle in Prescan was selected as the experiment object, and a two-degrees-of-freedom kinematic model was employed in the vehicle configuration. The vehicle kinematic model includes an engine, transmission, chassis, shift logic, and switching between automatic and manual shifting. Vehicle parameters like mass and wheelbase can be modified in the object configuration dialog. In this experiment, we selected only steering,

throttle, and brake press as the inputs and set auto gear shifting to 1. The output of the vehicle kinematic model will be published on the Prescan bus.

4.2. HiLExperiment Configuration

The vehicle mass was $M = 1104$ kg and the vehicle wheelbase was $L = 2.48$ m. One of the difficulties we overcame was poor performance under conditions where curvatures changed rapidly or speeds were high. Therefore, we take the lane change and return maneuvers as an item. As shown in Figure 6, the solid red line is the target trajectory. Considering the adaptability of the algorithm at different speeds, the vehicle should be set to a total of nine speeds between 10–90 km/h.

Figure 6. Experiment schematic diagram.

The central processing unit of the trajectory tracking controller is an Intel Core i7-8700. The operating system is Ubuntu 16.04.4. The compiler uses release mode optimize program running efficiency. The main parameters of the NMPC tracker in the experiment are as follows: Discretization period $T = 0.2$ s, prediction time horizon $n = 25$, lateral deviation weight coefficient $k_1 = 1$, yaw angle deviation weight coefficient $k_2 = 500$, control input weighting coefficient $k_3 = 1000$, rotation wheel angle constraint $\delta_{max} = 0.6$ rad, and rotation wheel angle control increment constraint $\Delta\delta_{max} = 0.04$ rad.

4.3. HiLExperiment Results

The actual trajectory is shown as the black dashed line in Figure 6, for a vehicle tracking the target trajectory at a speed of 30 km/h. The tracking lateral deviation mean square error is 8.7814×10^{-4} m², which shows that the vehicle can track the target trajectory well. The mean calculation time is 6.7 ms. It is obvious from Figure 7 that the computing time for each step was within 100 ms, which illustrates that a real-time solution of the non-linear programming problem was realized under conditions of ensuring accuracy.

Figure 7. Computing time for each step at 30 km/h.

The lateral deviations at speeds of 10, 20, 30, and 60 km/h are shown in Figure 8. As vehicle speeds increase, the amplitude of the lateral deviation fluctuates gradually, and remains in a small range, which illustrates the stability of the algorithm at different speeds. The AB and CD segments had large curvatures relative to the overall path. When the vehicle entered these segments, lateral deviation

increased slightly, but still fell within a low range, verifying the accuracy of the algorithm. Throughout the experiment, a total of nine speeds between 10–90 km/h were selected to complete trajectory tracking. The mean square errors for tests at different speeds were used as the evaluation index for trajectory tracking, as indicated by the solid line in Figure 9. As vehicle speeds increased, the steady-state mean square error values slowly increased. At low speeds between 0 and 30 km/h, the mean squared error values remained at approximately 10^{-4} m^2 . At medium and high speeds, the mean squared error values remained at approximately 10^{-3} m^2 . The steady-state mean square error values for different vehicle speeds reflect the algorithm's good stability and trajectory tracking ability. During actual automatic driving, large deviations may cause accidents. The mean square error cannot be used to effectively evaluate special situations, so we take the max deviation as another evaluation index. It can be seen from Figure 9 that the maximum deviation points are less than 0.6 m which guarantees the driving safety. The maximum deviation points can be divided into two trend lines. This is referred to as an overlap for a nonlinear system, and is a potential subject for future research.

(a) lateral deviations at 10 km/h

(b) lateral deviations at 20 km/h

(c) lateral deviations at 30 km/h

(d) lateral deviations at 60 km/h

Figure 8. Lateral deviations.

Figure 9. Mean square errors and maximum deviations for different speeds.

In order to compare the effects with other algorithms and prove the superiority of the MPC algorithm, we use the LQR algorithm to follow a similar target trajectory, and the effect is shown in Figure 10. The speed is set to be 30 km/h. It is clear that the vehicle cannot follow the target trajectory very well while using LQR algorithm. It is because LQR does not have a predictive model so that there will be a lag during the following process which causes a larger lateral deviation than MPC, as can be seen from Figure 11. Comparing the effects of MPC and LQR, it is convinced that MPC can guarantee the safety to some extent.

Figure 10. Trajectory tracking effect of LQR.

Figure 11. Lateral deviation of LQR.

4.4. Real Vehicle Experiment

The NMPC algorithm proposed was also applied to an autonomous vehicle, which won third prize in the 2019 Intelligent Connected Vehicle Competition organized by the Guangzhou municipal government. The vehicle was developed by the South China University of Technology and is shown in Figure 12. Figure 13 shows the technical architecture which can be divided into three parts:

Software, hardware, and vehicle. The software architecture significantly impacted control efficacy. The localization module and perception module could send data to the planning module to generate a local target trajectory. Ignoring changes in speed for the time being, lateral controls play a vital role in the control module. The NMPC algorithm proposed above was applied to the lateral control module. Finally, the first steering angle of the solution sequence was sent to the CAN bus.

Figure 12. Autonomous vehicle.

Figure 13. Technical architecture.

Results and Analysis

There is a scene of entering a parking lot through continuous right-angle turns in the competition, as shown in Figure 14a. Figure 14b,c show the reference trajectory and the curvature of the trajectory. It can be seen that the curvature of the trajectory increases rapidly when making a right-angle turn. In order to follow the actual driving rules and guarantee the safety, the turning speed is limited to 8 km/h.

(a) scene of entering a parking lot in the competition

Figure 14. Cont.

(b) tracking effect

(c) curvature of the reference trajectory

Figure 14. Results of the real vehicle experiment.

The result can be seen from Figure 14b that the vehicle can follow the reference trajectory during the whole process. Although there is slight deviation around 0.4 m at the larger turning rate, it is also within an allowable range, which meets the requirements of unmanned vehicle control. This also fully proves the algorithm's ability to follow the trajectory with large curvature.

5. Conclusions

In this paper, a suitable discretization method was used to solve a real-time NMPC problem. Then, a SQP method was used for the solution. Monte Carlo simulations were performed to analyze the robustness, which is rare in autonomous vehicle trajectory tracking. The algorithm was verified with both a HiL simulation platform and a real vehicle, and the calculation accuracy and speed were well balanced. The experimental results showed that the algorithm could effectively solve trajectory tracking problems. The mean square error was maintained at approximately 10^{-3} m^2 and the calculation time was 100 ms. It can be concluded that this method can guarantee the accuracy and real-time performance of an algorithm during a driving process, which is of great significance in practical vehicle applications. However, there are still some aspects which can be improved in this paper. The vehicle model in Monte Carlo simulation can be more complex a small number of parameters were random sampled with the limitations of experimental conditions. And the application research and development are based on the three-degree-of-freedom nonlinear kinematics model. In fact, the vehicle is a high-degree-of-freedom nonlinear system. In the future, it can be extended to a dynamic vehicle model with higher degrees of freedom to explore the performance of algorithmic path tracking combined with vehicle dynamics. The experiment item is the lane change and return

maneuvers, which is classic in a stability test, and we will test more complicated conditions on the basis of this experiment. And in the real vehicle experiment, the speed was kept at about 50 km/h considering the safety for all participants in the competition. More complex conditions and higher speeds will be tested in the future, and the overlap for a nonlinear system is a potential subject. In this paper, a feasible method and some problems are presented which may inspire other researchers and promote the development of unmanned vehicle trajectory tracking.

Author Contributions: Conceptualization, K.Z. and H.L.; methodology, K.Z.; software, G.X.; validation, K.Z., C.W. and G.X.; resources, J.Y.; data curation, H.L.; writing—original draft preparation, C.W.; writing—review and editing, K.Z.; supervision, Y.L.; project administration, K.Z. and Y.L.; funding acquisition, K.Z. and Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Key-Area Research and Development Program of Guangdong Province (No. 2019B090912001) and Natural Science Foundation of Guangdong Province (No. 2019A1515110562).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study, in the collection, analyses, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

References

1. Guan, Z.; Ding, J.; Du, F. *Simulation Research on Trajectory Tracking Controller Based on MPC Algorithm*; IEEE: Piscataway, NJ, USA, 2017; pp. 212–216.
2. Liu, P.; Wang, Y.; Cong, W.; Lei, W. Grouping-Sorting-Optimized Model Predictive Control for Modular Multilevel Converter With Reduced Computational Load. *IEEE Trans. Power Electron.* **2016**, *31*, 1896–1907. [\[CrossRef\]](#)
3. Tang, L.; Yan, F.; Zou, B.; Wang, K.; Lv, C. An Improved Kinematic Model Predictive Control for High-Speed Path Tracking of Autonomous Vehicles. *IEEE Access* **2020**, *8*, 51400–51413. [\[CrossRef\]](#)
4. Guo, P.; He, Z.; Yue, Y.; Xu, Q.; Huang, X.; Chen, Y.; Luo, A. A Novel Two-Stage Model Predictive Control for Modular Multilevel Converter With Reduced Computation. *IEEE Trans. Ind. Electron.* **2019**, *66*, 2410–2422. [\[CrossRef\]](#)
5. Kayacan, E.; Ramon, H.; Saeys, W. Robust Trajectory Tracking Error Model-Based Predictive Control for Unmanned Ground Vehicles. *IEEE-Asme Trans. Mechatron.* **2016**, *21*, 806–814. [\[CrossRef\]](#)
6. Moreno-Valenzuela, J.; Perez-Alcocer, R.; Guerrero-Medina, M.; Dzul, A. Nonlinear PID-Type Controller for Quadrotor Trajectory Tracking. *IEEE-Asme Trans. Mechatron.* **2018**, *23*, 2436–2447. [\[CrossRef\]](#)
7. Shan, Y.; Yang, W.; Chen, C.; Zhou, J.; Zheng, L.; Li, B. CF-Pursuit: A Pursuit Method with a Clothoid Fitting and a Fuzzy Controller for Autonomous Vehicles. *Int. J. Adv. Robot. Syst.* **2015**, *12*, 134. [\[CrossRef\]](#)
8. Morales, S.; Magallanes, J.; Delgado, C.; Canahuire, R. LQR Trajectory Tracking Control of an Omnidirectional Wheeled Mobile Robot. In Proceedings of the 2018 IEEE 2nd Colombian Conference on Robotics and Automation (CCRA), Barranquilla, Colombia, 1–3 November 2018.
9. Jianyu, C.; Wei, Z.; Tomizuka, M. Autonomous driving motion planning with constrained iterative LQR. *IEEE Trans. Intell. Veh.* **2019**, *4*, 244–254. [\[CrossRef\]](#)
10. Shen, C.; Shi, Y.; Buckham, B. Trajectory Tracking Control of an Autonomous Underwater Vehicle Using Lyapunov-Based Model Predictive Control. *IEEE Trans. Ind. Electron.* **2018**, *65*, 5796–5805. [\[CrossRef\]](#)
11. Falcone, P.; Borrelli, F.; Asgari, J.; Tseng, H.E.; Hrovat, D. Predictive active steering control for autonomous vehicle systems. *IEEE Trans. Control Syst. Technol.* **2007**, *15*, 566–580. [\[CrossRef\]](#)
12. Chang, C.-W.; Shiao, J.-K. Quadrotor Formation Strategies Based on Distributed Consensus and Model Predictive Controls. *Appl. Sci. Basel* **2018**, *8*, 2246. [\[CrossRef\]](#)
13. Wang, C.; Liu, X.; Yang, X.; Hu, F.; Jiang, A.; Yang, C. Trajectory Tracking of an Omni-Directional Wheeled Mobile Robot Using a Model Predictive Control Strategy. *Appl. Sci. Basel* **2018**, *8*, 231. [\[CrossRef\]](#)
14. Bai, G.; Meng, Y.; Liu, L.; Luo, W.; Gu, Q.; Li, K. A New Path Tracking Method Based on Multilayer Model Predictive Control. *Appl. Sci. Basel* **2019**, *9*, 2649. [\[CrossRef\]](#)
15. Wu, H.; Si, Z.; Li, Z. Trajectory Tracking Control for Four-Wheel Independent Drive Intelligent Vehicle Based on Model Predictive Control. *IEEE Access* **2020**, *8*, 73071–73081. [\[CrossRef\]](#)
16. Weiskircher, T.; Wang, Q.; Ayalew, B. Predictive Guidance and Control Framework for (Semi-) Autonomous Vehicles in Public Traffic. *IEEE Trans. Control Syst. Technol.* **2017**, *25*, 2034–2046. [\[CrossRef\]](#)

17. Cao, J.; Song, C.; Peng, S.; Song, S.; Zhang, X.; Xiao, F. Trajectory Tracking Control Algorithm for Autonomous Vehicle Considering Cornering Characteristics. *IEEE Access* **2020**, *8*, 59470–59484. [\[CrossRef\]](#)
18. Gallego, J.; Govender, V.; Müller, S. Model Predictive Lateral Vehicle Guidance Using a Position Controlled EPS System. *IFAC-PapersOnLine* **2017**, *50*, 265–270. [\[CrossRef\]](#)
19. Paden, B.; Cap, M.; Sze Zheng, Y.; Yershov, D.; Frazzoli, E. A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles. *IEEE Trans. Intell. Veh.* **2016**, *1*, 33–55. [\[CrossRef\]](#)
20. Zietkiewicz, J.; Owczarkowski, A. Direct Nonlinear Model Predictive Control and Predictive Control with Feedback Linearization. A Comparison of the Approaches. In Proceedings of the 2017 18th International Carpathian Control Conference (ICCC), Sinaia, Romania, 28–31 May 2017; pp. 451–455.
21. Kayacan, E.; Saeys, W.; Ramon, H.; Belta, C.; Peschel, J.M. Experimental Validation of Linear and Nonlinear MPC on an Articulated Unmanned Ground Vehicle. *IEEE-Asme Trans. Mechatron.* **2018**, *23*, 2023–2030. [\[CrossRef\]](#)
22. Kayacan, E.; Kayacan, E.; Ramon, H.; Saeys, W. Robust Tube-Based Decentralized Nonlinear Model Predictive Control of an Autonomous Tractor-Trailer System. *IEEE-Asme Trans. Mechatron.* **2015**, *20*, 447–456. [\[CrossRef\]](#)
23. Rafaila, R.C.; Livint, G. Nonlinear Model Predictive Control of Autonomous Vehicle Steering. In Proceedings of the 2015 19th International Conference on System Theory, Control and Computing (ICSTCC), Cheile Gradistei, Romania, 14–16 October 2015; pp. 466–471.
24. Rafaila, R.C.; Livint, G.; IEEE. Predictive Control of Autonomous Steering for Ground Vehicles. In Proceedings of the 2015 9th International Symposium on Advanced Topics in Electrical Engineering (ATEE), Bucharest, Romania, 7–9 May 2015; pp. 543–547.
25. Li, S.; Li, Z.; Yu, Z.; Zhang, B.; Zhang, N. Dynamic Trajectory Planning and Tracking for Autonomous Vehicle With Obstacle Avoidance Based on Model Predictive Control. *IEEE Access* **2019**, *7*, 132074–132086. [\[CrossRef\]](#)
26. Chen, J.; Lin, C.; Liang, S. Mixed logical dynamical model-based MPC for yaw stability control of distributed drive electric vehicles. In Proceedings of the 2018 10th International Conference on Applied Energy (ICAE), Hong Kong, China, 22–25 August 2018; pp. 2518–2523.
27. Tan, Q.; Dai, P.; Zhang, Z.; Katupitiya, J. MPC and PSO Based Control Methodology for Path Tracking of 4WS4WD Vehicles. *Appl. Sci. Basel* **2018**, *8*, 1000. [\[CrossRef\]](#)
28. Ul Haq, A.A.; Cholette, M.E.; Djurdjanovic, D. A Dual-Mode Model Predictive Control Algorithm Trajectory Tracking in Discrete-Time Nonlinear Dynamic Systems. *J. Dyn. Syst. Meas. Control.* **2017**, *139*, 044501. [\[CrossRef\]](#)
29. Johansson, B.; Gafvert, M. Untripped SUV Rollover Detection and Prevention. In Proceedings of the 2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No. 04CH37601), Nassau, Bahamas, 14–17 December 2004.
30. Yildirim, E.A.; Wright, S.J. Warm-start Strategies in Interior-point Methods for Linear Programming. *Siam J. Optim.* **2002**, *12*, 782–810. [\[CrossRef\]](#)
31. Velenis, E.; Frazzoli, E.; Tsiotras, P. Steady-state Cornering Equilibria and Stabilisation for a Vehicle During Extreme Operating Conditions. *Int. J. Veh. Autonom. Syst.* **2010**, *8*, 217–241. [\[CrossRef\]](#)
32. Merabti, H.; Bouchachi, I.; Belarbi, K. Nonlinear Model Predictive Control of Quadcopter. In Proceedings of the 2015 16th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Monastir, Tunisia, 21–23 December 2015; pp. 208–211.
33. Duan, H.; Li, Q. Sugar Precipitation Control of Chinese Medicine Solution Based on Nonlinear Model Predictive. In Proceedings of the 2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics, Hangzhou, China, 26–27 August 2015; pp. 66–69.
34. Montazeri-Gh, M.; Rasti, A.; Imani, A. Comparison of Model Predictive Controller and Min-max Approach for Aircraft Engine Fuel Control. In Proceedings of the 2017 5th International Conference on Control, Instrumentation, and Automation (ICCIA), Shiraz, Iran, 21–23 November 2017; pp. 331–336.
35. Ubare, P.; Ingole, D.; Sonawane, D. Energy-efficient Nonlinear Model Predictive Control of BLDC Motor in Electric Vehicles. In Proceedings of the 2019 Sixth Indian Control Conference (ICC), IEEE, Hyderabad, India, 18–20 December 2019; pp. 194–199.