Constrained Model Predictive Control for Low-cost Autonomous Driving With Stability Guarantees

Yifeng Tang[1,2], Yongsheng Ou[3]*
[1] Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, PR China
[2] University of Chinese Academy of Sciences, Beijing 100049, PR China
[3] Department of Guangdong Provincial Key Lab of Robotics and Intelligent System
Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences

***Abstract*—The cost of autonomous drivings system is currently one of the key factors that will decide whether this technology can be widely deployed. In this paper, the control system design of an autonomous vehicle with low-cost mechanical structure and electrical system is studied. The open-loop responses of the velocity and steering system illustrate that this vehicle is confronted with limited hardware performance. The mean goal of this research is applying high performance algorithm to stabilize the system and guarantee the stability as much as possible. In other words, sacrificing some computation to compensate the limited hardware. Model predictive control (MPC) is a control algorithm based on rolling optimization, although it requires more computation than traditional control algorithms it can taking multi complex constraints into account so that it can tackle various problems in this system. A constrained MPC is proposed to control this low-cost vehicle, it is tested on the simulation and real vehicle. By comparing the system performance in open-loop and close-loop, the algorithm is proved to be effective. Finally, the Lyapunov stability proof is also given out in this paper, which guarantees the theoretical stability of the algorithm.**

***Index Terms*—low-cost autonomous driving, Model Predictive Control (MPC), constrained optimization, Lyapunov stability**

I. Introduction

High accuracy sensors, advanced motors and complex mechanical structures will rapidly increase the cost of the system. High cost will barrier the wide deployment of autonomous driving so that reducing the cost is a proposing research topic in this area. In recent years, the average cost of the computation resources has greatly decreased while that of hardware stays the same. It gives a possible way to reducing the cost, that is, using algorithm to compensate the limited hardware performance. The paper proposes a constrained MPC algorithm with stability guarantees to tackle the trajectory tracking problem of a low-cost autonomous vehicle. There are some control algorithms that are verified in real vehicles, PID [1], sliding mode control (SMC) [2], fuzzy control[3], and some adaptive control etc. PID is a classical model-free control algorithm but it can only solve the Single-Input-Single-Output system while finding a proper parameters configuration for a nonlinear system can be really difficult. An adaptive PID control algorithm is proposed in [3] to solve the trajectory tracking problem of mobile robots with integrated multiple sensors. SMC is a kind of nonlinear control algorithm with stability guarantees since the control law is derived from the Lyapunov function [5]. However, just as it is mentioned in this paper, SMC algorithm cannot reject disturbances. In [7], the authors propose an integrated SMC algorithm with extended Kalman filter, and the results show that this algorithm can reject disturbances and stabilize the system. Fuzzy logic is usually used to design adaptive control algorithms and it is always integrated with others algorithms such as Artificial Neural Network (ANN) and SMC. In [3], a conventional control algorithm combining fuzzy logic and SMC is proposed to solve the tracking problem of an autonomous vehicle.

With the rapid development of the computers, some optimal control algorithms can be realized in real-time platforms. Linear Quadratic Regulator (LQR) [16] is a classical optimal control algorithms based on linear programming theory, solving the Riccati equations derived from the Hamilton-Jacobi-Bellman (HJB) equation can derive the control law. Linear Quadratic Gaussian (LQG) is another optimal control algorithms in which the sensor noise is modeled as Gaussian Process. It is showed that LQG can improve the tracking performance and it can handle noise better that LQR [7,8]. MPC is a kind of finite-horizon optimal control algorithms and it is much more complex than those mentioned before, but it is strongly proved that it handle the tracking problem better. The most typical advantage of MPC is that it can obtain control algorithms suitable for different systems by adding constraints References [11-15] highlight the recent development of MPC by introducing the successful applications in various systems. On the other hand, authors in [17,18] find that the unconstrained MPC cannot reject disturbances and the parameters configuration may greatly influence the performance of the vehicles. It is undeniable that MPC is a very promising control algorithm, and the heavy calculation burden of MPC is no longer a problem for current computers.

The paper is organized as follows. An brief introduction about the system model together with system constrains are introduced in Section II. The derivation of MPC is introduced in detail in Section III and there are also some contents about solving this optimization problem are also included in this

This work was financially supported by National Key Research and Development Program of China under Grant 2018AAA0103001, National Natural Science Foundation of China (Grants No. U1813208, 62173319, 62063006), Guangdong Basic and Applied Basic Research Foundation (2020B1515120054), Shenzhen Fundamental Research Program (JCYJ20200109115610172), Guangdong Provincial Science and Technology Program (2022A0505050057)

*Yongsheng Ou is the corresponding author, Email: ys.ou@siat.ac.cn

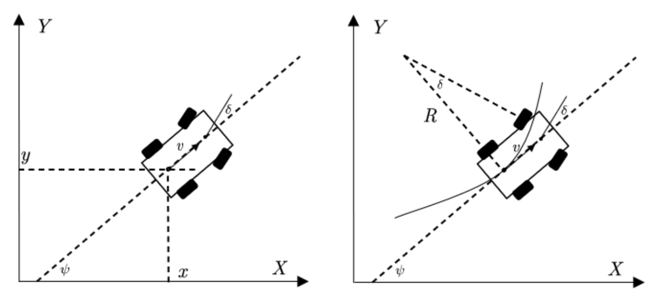
section. Then the contents about simulation experiment and real vehicle experiment is arranged in Section IV. Some test results are introduced in this section which shows the limited hardware performance of this vehicle. Then in Section V, some discussion about stability is given out and the proof of Lyapunov stability is also included. Finally in Section VI, a summary about this work is given out.

II. System Modeling and Constraints

The contents of this section are divided into three sub ones, first is the description of the system model of the system, then is the analysis of the observation equation with sensor noise, and the final one is the analysis about the system constraints.

A. Vehicle Kinematics Model

This paper mainly considers the kinematics model of the vehicle, with hard constraints considering the physical limits of the system. Using $\mathbf{x}=(x,y,\psi)^T$ to represent the position and orientation of the car with the fact that all states variable in this system are considered in the World Coordinates. Specifically, x, y means the position of the center of the rear wheels which are identified with the SLAM system using laser radar. The vehicle body angel ψ is the angle between the body axis across the center of the rear wheels and x-axis. The wheelbase of the vehicle l means the distance between the rear wheels and front wheels. The input of this system $\mathbf{u}=(v,\delta)^T$, also referred as the control value, is the wheel velocity v and the steering angle δ. If there does not exist relative slip between the wheels and vehicle body , v can be regarded as the vehicle's velocity. Figure 1 (a) shows the relationship of all variables and (b) shows the trajectory tracking model of the vehicle in which R represents the turning radius of the vehicle. After that, the kinematics model of the vehicle can be derived which is also the system model to be researched in this paper.



(a) Vehicle Modeling (b) Trajectory Tracking

Fig. 1. System Model of the trajectory tracking problem of an autonomous vehicle.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\psi \\ \sin\psi \\ \tan\delta/l \end{bmatrix} v \tag{1}$$

Obviously, it is a nonlinear system and there is coupling relationship between the inputs v and δ. In order to simplify the controller design, linearization and discretization are taken into account. Equation (1) can be regarded as a first-order continuous differential equation:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \tag{2}$$

The reference trajectory is $(\mathbf{x}_r, \mathbf{u}_r)$ and for the system there is $\dot{\mathbf{x}}_r = f(\mathbf{x}_r, \mathbf{u}_r)$. The system can be approximated in a Taylor series at $(\mathbf{x}_r, \mathbf{u}_r)$:

$$\dot{\mathbf{x}} \approx f(\mathbf{x}_r, \mathbf{u}_r) + \frac{\partial f}{\partial \mathbf{x}}(\mathbf{x}-\mathbf{x}_r) + \frac{\partial f}{\partial \mathbf{u}}(\mathbf{u}-\mathbf{u}_r) \tag{3}$$

After linearization, the system can be written as follows:

$$\begin{gathered} \dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \\ A = \begin{bmatrix} 0 & 0 & -v\sin\psi \\ 0 & 0 & v\cos\psi \\ 0 & 0 & 0 \end{bmatrix} B = \begin{bmatrix} \cos\psi & 0 \\ \sin\psi & 0 \\ \frac{\tan\delta}{l} & \frac{v}{l\cos^2(\delta)} \end{bmatrix} \end{gathered} \tag{4}$$

The goal is controlling the vehicle complete the trajectory tracking, in other words, control the system states converge to reference trajectory. Define $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_r$ and the discretization can be done according to the following equations in which T is the length of the time steps:

$$\begin{cases} \tilde{A} = I_{n\times n} + TA \\ \tilde{B} = TB \end{cases} \tag{5}$$

Then discrete form of the system is written as:

$$\begin{gathered} \tilde{\mathbf{x}}_{k+1} = \tilde{A}\tilde{\mathbf{x}}_k + \tilde{B}\tilde{\mathbf{u}}_k \\ \tilde{\mathbf{x}} = \begin{bmatrix} x-x_r \\ y-y_r \\ \psi-\psi_r \end{bmatrix} \tilde{A} = \begin{bmatrix} 1 & 0 & -v_r\sin\psi_r T \\ 0 & 1 & v_r\cos\psi_r T \\ 0 & 0 & 1 \end{bmatrix} \\ \tilde{B} = \begin{bmatrix} \cos\psi_r T & 0 \\ \sin\psi_r T & 0 \\ \frac{\tan\delta_r T}{l} & \frac{v_r T}{l\cos^2(\delta_r)} \end{bmatrix} \tilde{\mathbf{u}}_k = \begin{bmatrix} v-v_r \\ \delta-\delta_r \end{bmatrix} \end{gathered} \tag{6}$$

B. System Contraints

Usually, the reference input of the vehicle $(v_r, \delta_r) = (0,0)$ which means the vehicle finishes the tracking and stopped. In that way, the input $\tilde{\mathbf{u}}_k = (v,\delta)^T$. In a real vehicle system, there are strict steering angle constraints and velocity constraints:

$$\begin{gathered} -5 \le v \le 5 \\ -45^\circ \le \delta \le 45^\circ \end{gathered} \tag{7}$$

$m \cdot s^{-1}$ is used to describe the velocity in the World Coordinate. Besides the state constraints, the increment value of the inputs should also be constrained in order to guarantee the stability of the vehicle's movement.

$$\begin{gathered} -0.5m/s \le \Delta v \le 0.5m/s \\ -2^\circ \le \Delta\delta \le 2^\circ \end{gathered} \tag{8}$$

For system model, the inequalities above can be rewritten as:

$$\begin{gathered} \begin{bmatrix} -5 \\ -45^\circ \end{bmatrix} \le \tilde{\mathbf{u}}_k \le \begin{bmatrix} 5 \\ 45^\circ \end{bmatrix} \\ \begin{bmatrix} -0.5 \\ -2^\circ \end{bmatrix} \le \Delta\tilde{\mathbf{u}}_k \le \begin{bmatrix} 0.5 \\ 2^\circ \end{bmatrix} \end{gathered} \tag{9}$$

All the inequality constraints are rewritten as $h(\tilde{\mathbf{x}}_k, \tilde{\mathbf{u}}_k) \le 0$.

Reference [12] is a very important paper in which a theoretical proof about the stability of MPC is given out. The authors point out that the Lyapunov stability cannot be guaranteed for unconstrained MPC. Adding constraints to different control systems is an effective method to guarantee stability. For an autonomous driving system, the ultimate goal is to make the vehicle state converge to the desired state and the system stay stable around the equilibrium point. Therefore, it is natural to think that adding terminal constraints can guarantee that the optimization converges to the equilibrium point. Here, we redefine a symbol $\tilde{\mathbf{x}}_{k+i|k}\ \ (i=1,2,...N_p)$ to represent the i-th state from $\mathbf{x_k}$ and $\tilde{\mathbf{u}}_{k+i|k}\ \ (i=1,2,...N_p)$ to represent the i-th input from $\mathbf{u_k}$. Then the terminal constraint is:

$$\tilde{\mathbf{x}}_{k+N_p|k}=0 \tag{10}$$

With this constraint, the stability can be guaranteed and the proof about this constrained MPC is given out in Section V. In addition, the experimental results also state this conclusion.

III. MPC Controller

MPC is a control algorithm based on discrete system model prediction. The predictive state of the system can be obtained through the current state and recursion using discrete model. By solving the optimization, the system input u that minimizes the multi-step errors can be obtained.

A. Prediction Equations

One of the important parts of MPC is that the algorithm can predict the system‘s states after multiple time steps according to the discrete model. Then the controller can control the current and future states of the system in multiple steps. Here, N_p is defined as the length of the prediction steps, which means that the controller considers the prediction state after N_pT. There will be:

$$\begin{aligned}
&\tilde{\mathbf{x}}_{k+1|k}=\tilde{A}\tilde{\mathbf{x}}_{k|k}+\tilde{B}\tilde{\mathbf{u}}_{k|k}\\
&\tilde{\mathbf{x}}_{k+2|k}=\tilde{A}\left(\tilde{A}\tilde{\mathbf{x}}_{k|k}+\tilde{B}\tilde{\mathbf{u}}_{k|k}\right)+\tilde{B}\mathbf{u}_{k+1|k}\\
&\tilde{\mathbf{x}}_{k+3|k}=\tilde{A}^3\tilde{\mathbf{x}}_{k|k}+\tilde{A}^2\tilde{B}\tilde{\mathbf{u}}_{k|k}+\tilde{A}\tilde{B}\mathbf{u}_{k+1|k}+\tilde{B}\mathbf{u}_{k+2|k}\\
&...\\
&\tilde{\mathbf{x}}_{k+N_p|k}=\tilde{A}^{N_p}\tilde{\mathbf{x}}_{k|k}+\tilde{A}^{N_p-1}\tilde{B}\tilde{\mathbf{u}}_{k|k}+...\tilde{A}^{N_p-i}\tilde{B}\tilde{\mathbf{u}}_{k-i+1|k}\\
&\quad+\tilde{A}\tilde{B}\tilde{\mathbf{u}}_{k+N_p-2|k}+\tilde{B}\tilde{\mathbf{u}}_{k+N_p-1|k}
\end{aligned} \tag{11}$$

Define a new symbol $\mathbf{X}_k=\left[\tilde{\mathbf{x}}_{k+1|k},\tilde{\mathbf{x}}_{k+2|k},...,\tilde{\mathbf{x}}_{k+N_p|k}\right]^T$ and a new symbol $\mathbf{U}_k=\left[\mathbf{u}_{k|k},\mathbf{u}_{k+1|k},...,\mathbf{u}_{k+N_p|k}\right]^T$.Then the system model can be rewritten into this form:

$$\begin{aligned}
&\mathbf{X}_k=M\tilde{\mathbf{x}}_k+P\mathbf{U}_k\\
&M=\begin{bmatrix} I\\ \tilde{A}\\ \tilde{A}^2\\ ...\\ \tilde{A}^{N_p}\end{bmatrix} P=\begin{bmatrix} 0 & 0 & 0 & ... & 0\\ \tilde{B} & 0 & 0 & ... & 0\\ \tilde{A}\tilde{B} & B & 0 & ... & 0\\ .. & ... & ... & ... & ...\\ \tilde{A}^{N_p-1}\tilde{B} & \tilde{A}^{N_p-2}\tilde{B} & ... & \tilde{A}\tilde{B} & \tilde{B}\end{bmatrix}
\end{aligned} \tag{12}$$

This is the system model with prediction, and the problem turns into designing a control algorithm to stabilize this new system.

B. Optimization with Softened Constraints

The general form of MPC optimization problems is:

$$\begin{aligned}
J\left(\tilde{\mathbf{x}}_k,\tilde{\mathbf{u}}_k\right)&=\min_{\tilde{\mathbf{x}},\tilde{\mathbf{u}}}\sum\mathcal{L}\left(\tilde{\mathbf{x}}_k,\tilde{\mathbf{u}}_k\right)\\
s.t.\tilde{\mathbf{x}}_{k+1}&=f\left(\tilde{\mathbf{x}}_k,\tilde{\mathbf{u}}_k\right)\\
\tilde{\mathbf{x}}_0&=\tilde{\mathbf{x}}_{init}\\
h\left(\tilde{\mathbf{x}}_k,\tilde{\mathbf{u}}_k\right)&\le 0
\end{aligned} \tag{13}$$

where $\mathbf{x}_0$ is the initial state of the system. For this low-cost autonomous driving system, the optimization function should be a linear quadratic function about the tracking errors $\tilde{\mathbf{x}}_k$ and the system inputs $\tilde{\mathbf{u}}_k$. In addition, there must be the corresponding constraints that are derived from the requirement. In this paper, the final optimization function is:

$$\begin{aligned}
J\left(\tilde{\mathbf{x}}_k,\tilde{\mathbf{u}}_k\right)&=\min_{\tilde{\mathbf{x}},\tilde{\mathbf{u}}}\sum_{i=0}^{N_p-1}\left(\tilde{\mathbf{x}}_i^TQ\tilde{\mathbf{x}}_i+\tilde{\mathbf{u}}_i^TR\tilde{\mathbf{u}}_i\right)\\
&\quad+\tilde{\mathbf{x}}_{N_p}^TQ_{N_p}\tilde{\mathbf{x}}_{N_p}\\
s.t.\tilde{\mathbf{x}}_{k+1}&=f\left(\tilde{\mathbf{x}}_k,\tilde{\mathbf{u}}_k\right)\\
\tilde{\mathbf{x}}_0&=\tilde{\mathbf{x}}_{init}\\
\tilde{\mathbf{x}}_{N_p}&=0\\
h\left(\tilde{\mathbf{x}}_k,\tilde{\mathbf{u}}_k\right)&\le 0
\end{aligned} \tag{14}$$

The terminal constraint $\tilde{\mathbf{x}}_{N_p}=0$ can guarantee the algorithm‘s stability which will be proved in section V. This is a typical linear quadratic function, for the parameters matrix, there are the following properties:

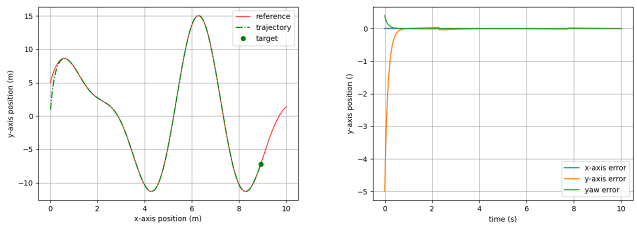
$$Q=Q^T\succ 0, Q_{N_p}=Q_{N_p}^T\succ 0, R=R^T\succ 0 \tag{15}$$

Solving this optimization problem will derive the control outputs $\mathbf{u}$ which is the input of the vehicle system. Linear Quadratic Programming is usually used in solve this kind of optimization problem, but the detailed introduction about optimization problem is not the concentration of this paper. In the experiment part, some optimization software packages are used to solve the problem. The solving algorithms are rolling optimization and linear quadratic programming which means solving the problem iteratively in a limited time.

IV. Experiment Results

A. Simulation

Before deploying the algorithm on the real vehicle, it is necessary to verify the algorithm in a simulation environment to avoid failure and crash. The simulation experiment was completed on a laptop with Intel i7 cpu and a Nvidia RTX2080 super , and the simulation software was Python. The performance of the computer is similar to that of the on-board IPC, so the calculation time can be used as a benchmark. The reference trajectory is generator with $y=2\sin\left(0.2x\right)+0.8\cos\left(0.5x\right), x\in[0,10]$ and the initial state of the vehicle is $\mathbf{x}=\left(0,0,0.5\right)^T$ with sampling time $dt=0.1s$. As for the parameters in this system, the length of the vehicle $l=1.8$ and the parameters matrix $Q=Q_{Np}=diag(1,1,1.5)$ and $R=diag(1.2,1.5)$.



(a) Simulation (b) Tracking Errors

Fig. 2. The simulation results in which (a) is the trajectory at each point and (b) shows the tracking errors.

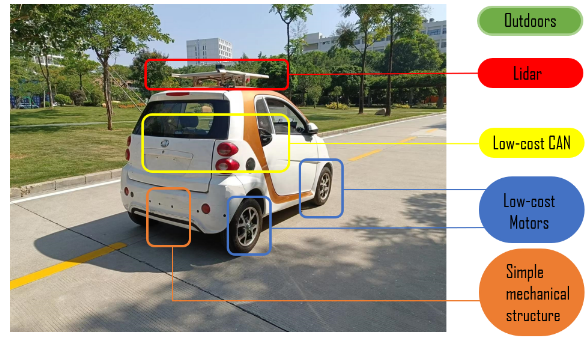


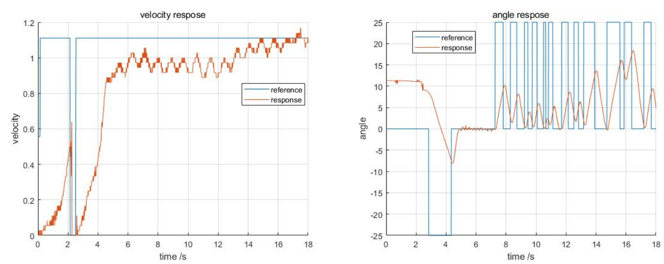
Fig. 3. The real vehicle on the road outside the door. It is equipped with a series of low-cost devices.

It can be seen from the figure above that the algorithm performs well in the simulation environment, and the trajectory strictly converges to the expected trajectory. The reason for choosing Python as the simulation programming language is that the ROS operating system of the real vehicle supports python language. If there is no loophole in the simulation environment, the program can be directly applied to real vehicles. The focus of this paper is not to simulate, but to solve the problem of real vehicles, so the experiment of real vehicles will be the concentration of the next part.

B. Real Vehicle

The real vehicle is a low-cost car with a cost of no more than 100k yuan, while the self driving vehicles on the market are generally more than 200k yuan. The mechanical structure of the vehicle is simple, and the transmission part uses a simplified front wheel steering model. The on-board IPC is a computer with only Intel i7 CPU, equipped with the Ubuntu 18.04 operating system, and the instructions are sent to the on-board communication network through USB2.0. The vehicle communication is low-cost CAN communication, and the data transmission speed is 200-300KB per second, which is not a fast speed. The vehicle attitude data feedback is realized by a low-cost laser radar. GPS and binocular camera are not applied to this system. The low-cost hardware system composition of the vehicle is shown in Figure 3. The reference track of the vehicle is given through the planning module in the front end, and the track planning is completed using Astar algorithm in the cost map. At the same time, in order to obtain the desired attitude of the vehicle, we adopt the local planning algorithm to solve the desired pose of the vehicle ψ with the sampling time of MPC $dt = 20ms$.

First of all, we test the step response of the real system under open-loop condition. Fig.4 is the illustration of the results, it can be seen that the system‘s performance is poor when it is open-loop. The system responses the signals slowly and there is a large steady-state error, and the system cannot converge to the reference state.It can be found from (a) that the velocity of the vehicle cannot reach the reference velocity in a short time, and there can be even stand still in halfway. From (b), we can find that there is an offset at the beginning of the vehicle’s corner. When a signal comes, the process of returning to the center is very slow, and the system cannot response to the subsequent square wave signal quickly, and the steady-state error value will even exceed the current state of the system. We can conclude that low-cost system has these properties: signal noise, serious hysteresis, and large steady-state error.



(a) velocity response (b) angle response

Fig. 4. The simulation results in which (a) is the trajectory at each point and (b) shows the tracking errors.

After that, we test the constraint MPC control algorithm proposed in this paper. MPC has some noise rejection, and the robustness of the algorithm can be analyzed according to the results with noise. In terms of system lag, we solve it by changing the parameters Q and R. Steady state error is a difficult problem to solve. Because the mechanical performance of the vehicle is very poor, we use relaxation factor in optimization to eliminate the steady state error. The tracking result is shown in Fig.5, there are there sub processes. In this experiment, $\rho = 0.5$ and $\varepsilon = 1.5$.The first five coefficients of Q and Q_f are larger than the last five coefficients, and R is the same. The reason for this is that the control algorithm pays more attention to the states in the first several time steps.

It can be seen that the system performance is greatly improved with the constrained MPC algorithm. For velocity, after a brake at around 10s, the velocity recover to the reference velocity quickly. As for angle, although there are still some steady state error, the response is significantly improved and the lag is well handled.

In order to analyze the results more accurately, a numerical comparison of the control system performance is given in Table I. The measurement include rise time t_r, peak time t_p, stability time t_s, steady-state error e_{ss} of the system. The time is measured in seconds (s) and the velocity is measured in

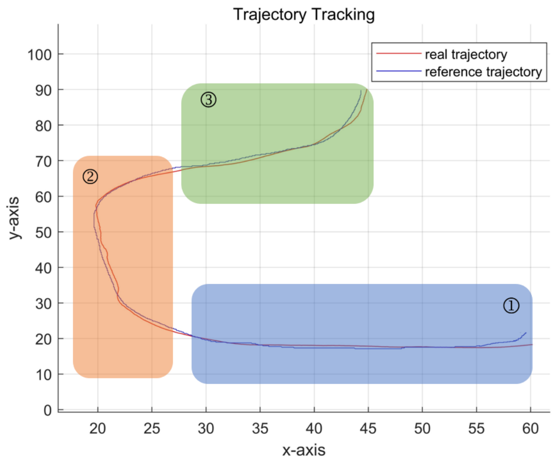


Fig. 5. The trajectory tracking performances of the real vehicle, it can be seen from the lines that the real one almost converge to the reference one. In order to analyze, we divide the tracking process into three sections, which are marked in the figure

meters per second ($m \cdot s^{-1}$) and degree for steering angle. The detailed data in the table is gathered from the numerical results. Then the velocity and angle tracking are as follows: For velocity, it can be found from the comparison that the performance is improved at least 70% with the algorithm especially for ts. Then it can be found that the performance of steering angle control is significantly improved since the original system cannot evenly tracking the squad signals. Meanwhile, it can be pointed out that the terminal constraint can guaranteed the stability of the real vehicle to some extent.

TABLE I
COMPARISON OF ORIGINAL SYSTEM AND CORRECTED SYSTEM

	t_r	t_p	t_s	e_{ss}
original-v	12.6	14.2	16.6	0.0
C-MPC-v	3.3	4.2	5.6	0.1
original-δ	/	/	/	14.0(average)
C-MPC-δ	1.8	2.2	2.8	3.1(average)

V. STABILITY ANALYSIS

The system model in this paper is:

$$\tilde{\mathbf{x}}_{k+1} = f\left(\tilde{\mathbf{x}}_k, \tilde{\mathbf{u}}_k\right) \tag{16}$$

The system input $\mathbf{u}_k$ in the k-th interval is generated by solving the following equation:

$$J_k = \min_{\tilde{\mathbf{x}}_k, \mathbf{u}_k} \sum_{i=1}^{N_p} \mathcal{L}\left(\tilde{\mathbf{x}}_{k+i|k}, \tilde{\mathbf{u}}_{k+i-1|k}\right) \tag{17}$$

According to the detailed introduction in the former part, $\mathcal{L}\left(\tilde{\mathbf{x}}_k, \tilde{\mathbf{u}}_k\right) \geq 0$ and $\mathcal{L}\left(\tilde{\mathbf{x}}_k, \tilde{\mathbf{u}}_k\right) = 0u$ if and only $\left(\tilde{\mathbf{x}}_k, \tilde{\mathbf{u}}_k\right) = (0, 0)$. Obviously, this state satisfies the constraints in the optimization and this is an equilibrium point of this system. In the following contents, we use J_k^e to represent the optimization function around the equilibrium point. Inspired from the Lyapunov Stability Theroy, this function can be treated as the discrete Lyapunov candidate function. Since every element in this function takes the form of linear quadratic, it is positive definite. Which means if its derivative is negative definite the Lyapunov stability is proved. For a discrete function, the derivative is the first order difference between the k-th item and the $(k-1)$-th one. Before introducing the rest proof, there should be an assumption at first.

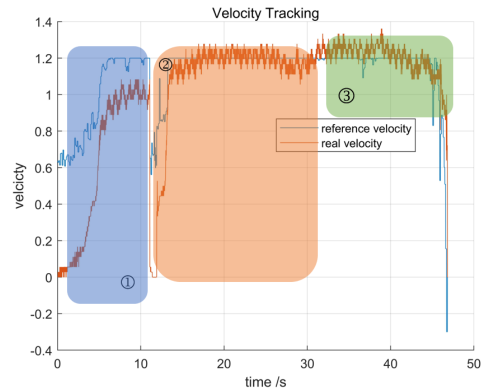
Assumption 1: The noise in the system is not considered, which means that the state $\tilde{\mathbf{x}}_k$ and input $\tilde{\mathbf{u}}_k$ in the system are unbiased with the true value.

Based on this assumption, there is $\tilde{\mathbf{x}}_{k+i|k} = \tilde{\mathbf{x}}_{k+i}, \tilde{\mathbf{u}}_{k+i|k} = \tilde{\mathbf{u}}_{k+i}$. Then it can be found that:

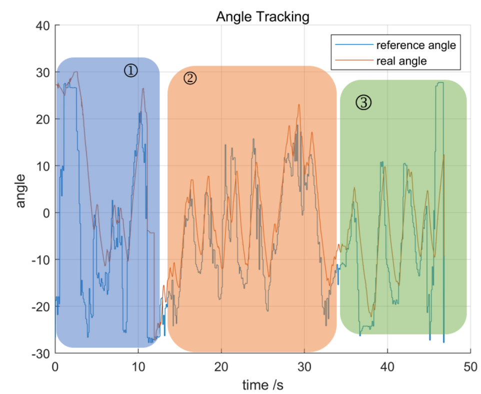
$$\begin{aligned}
J_{k+1}^e &= \min_{\tilde{\mathbf{x}}_k, \tilde{\mathbf{u}}_k} \sum_{i=1}^{N_p} \mathcal{L}\left(\tilde{\mathbf{x}}_{k+i+1}, \tilde{\mathbf{u}}_{k+i}\right) \\
&= \min_{\tilde{\mathbf{x}}_k, \mathbf{u}_k} \left\{ \begin{array}{c} \sum_{i=1}^{N} \mathcal{L}\left(\tilde{\mathbf{x}}_{k+i}, \tilde{\mathbf{u}}_{k+i-1}\right) + \\ \mathcal{L}\left(\tilde{\mathbf{x}}_{k+1+N_p}, \tilde{\mathbf{u}}_{k+N_p}\right) - \mathcal{L}\left(\tilde{\mathbf{x}}_{k+1}, \tilde{\mathbf{u}}_k\right) \end{array} \right\} \\
&= -\mathcal{L}\left(\tilde{\mathbf{x}}_{k+1}, \tilde{\mathbf{u}}_k\right) + J_k^e + \min_{\tilde{\mathbf{x}}_k, \mathbf{u}_k} \mathcal{L}\left(\tilde{\mathbf{x}}_{k+1+N_p}, \tilde{\mathbf{u}}_{k+N_p}\right)
\end{aligned} \tag{18}$$

Because of the terminal constraint (10), there is

$$\min_{\tilde{\mathbf{x}}_k, \tilde{\mathbf{u}}_k} \mathcal{L}\left(\tilde{\mathbf{x}}_{k+1+N_p}, \tilde{\mathbf{u}}_{k+N_p}\right) = 0 \tag{19}$$



(a) velocity tracking



(b) angle tracking

Fig. 6. The tracking results of velocity and angle, and it is worth noting that there is a brake between 1 and 2 so the velocity rapidly reduces to 0.

Then there is

$$J_{k+1}^e - J_k^e \leq 0 \tag{20}$$

which means the Lyapunov function J is positive definite with whose derivative is negative definite. Then the stability of the constrained MPC is proved. In fact, in order to accelerate the convergence of the algorithm, there is $\tilde{\mathbf{x}}_{N_p} = \varepsilon$ in which ε is a constant close to 0 and in real vehicle experiment $\varepsilon = 10^{-3}$.

VI. Conclusion

The main purpose of this paper is to design a control algorithm for a low-cost autonomous vehicle. The open-loop responses show that the hardware performances of the vehicle is limited. Using algorithms to compensate for hardware deficiencies is a promising solution so that MPC is chosen in this paper. The experimental results show that for this the algorithm can tackle the trajectory tracking problem and accelerates the response speed of the system. Since the limited hardware performances can influence the vehicle's movement in real world, so a terminal constraint is taken into account to guarantee the system's stability. This is also realize in the real vehicle experiment and the tracking result shows that the vehicle can track the reference trajectory stably using this algorithm. Additionally, the proof about the algorithm's Lyapunov stability is also given in this paper, which can guarantee the algorithm's stability theoretically. In sum, simulation experiments show that the algorithm has no theoretical defects, and experiments on real vehicles show that the algorithm is effective for low-cost vehicles.

The are still many problems in low-cost autonomous driving to be solved, which requires more effort. This research makes an attempt to compensate the limited hardware performance with algorithms. Although the MPC algorithm has achieved remarkable results, the computational burden is an inevitable problem when applying it. Reducing the computational burden can further reduce the cost. Therefore, in the future work, more simplified algorithms will be studied to solve the control problem of the system.

References

[1] B. Varma, N. Swamy and S. Mukherjee, "Trajectory Tracking of Autonomous Vehicles using Different Control Techniques(PID vs LQR vs MPC)," 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), pp. 84-89, 2020.

[2] C. L. Hwang, C. C. Yang and J. Y. Hung, "Path Tracking of an Autonomous Ground Vehicle With Different Payloads by Hierarchical Improved Fuzzy Dynamic Sliding-Mode Control," in IEEE Transactions on Fuzzy Systems, vol. 26, no. 2, pp. 899-914, 2018.

[3] P. Sarhadi, A. R. Noei, A. Khosravi, "Model reference adaptive PID control with anti-windup compensator for an autonomous underwater vehicle," Robotics and Autonomous Systems, vol. 83, pp. 87-93,2016.

[4] E. Alcala, V. Puig, J. Quevedo, T. Escobet, R. Comasolivas, "Autonomous vehicle control using a kinematic Lyapunov-based technique with LQR-LMI tuning," Control Engineering Practice, vol. 73, pp. 1-12, 2018.

[5] J. Yang, R. Ma, Y. R. Zhang, C. Z. Zhao, Sliding Mode Control for Trajectory Tracking of Intelligent Vehicle, Physics Procedia, vol. 33, pp.1160-1167, 2012.

[6] C. Hu et al., "MME-EKF-Based Path-Tracking Control of Autonomous Vehicles Considering Input Saturation," IEEE Transactions on Vehicular Technology, vol. 68, no. 6, pp. 5246-5259, 2019.

[7] T. M. Vu, A. Nitin, "Robust Model Predictive Control for Input Saturated and Softened State Constraints," Asian J. Control. vol. 7, 2005, pp. 319–325.

[8] A. Reda,A. Bouzid, J. Vásárhelyi, "Model Predictive Control for Automated Vehicle Steering." Acta Polytech. Hung. vol. 17, pp. 163–182, 2020.

[9] S. Chen, H. Chen, D. Negrut. "Implementation of MPC-Based Trajectory Tracking Considering Different Fidelity Vehicle Models." Journal of Beijing Institute of Technology, vol 29, no. 3, pp. 303-316, 2020.

[10] K. Lee, S. Jeon, H. Kim and D. Kum, "Optimal Path Tracking Control of Autonomous Vehicle: Adaptive Full-State Linear Quadratic Gaussian (LQG) Control," in IEEE Access, vol. 7, pp. 109120-109133, 2019.

[11] M. Marcano, S. Díaz, J. Pérez and E. Irigoyen, "A Review of Shared Control for Automated Vehicles: Theory and Applications," in IEEE Transactions on Human-Machine Systems, vol. 50, no. 6, pp. 475-491, 2020.

[12] D.Q. Mayne, J.B. Rawlings, C.V. Rao, P.O.M. Scokaert, Constrained model predictive control: Stability and optimality, Automatica, vol 36, issue 6, pp 789-814, 2000.

[13] Bakarác, Peter et al. "Comparison of inverted pendulum stabilization with PID, LQ, and MPC control." Cybernetics & Informatics (K&I) pp. 1-6, 2018

[14] Jiangfeng Nan, Bingxu Shang, Weiwen Deng, Bingtao Ren, Yang Liu, MPC-based Path Tracking Control with Forward Compensation for Autonomous Driving, IFAC-PapersOnLine, vol 54, issue 10, pp. 443-448, 2021

[15] K, Geng, S, Liy, "Robust Path Tracking Control for Autonomous Vehicle Based on a Novel Fault Tolerant Adaptive Model Predictive Control Algorithm." Appl. Sci. 2020, vol. 10, 6249, 2020.

[16] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," Automatica, vol. 38, no. 1, pp. 3–20, Jan. 2002.

[17] P. Tøndel, T. A. Johansen, and A. Bemporad, "An algorithm for multi-parametric quadratic programming and explicit MFC solutions," in Proc. IEEE Conf. Dec. Control, 2001, pp. 1199–1204.

[18] M. Vu, "Tracking Setpoint Robust Model Predictive Control for Input Saturated and Softened State Constraints. " International Journal of Control Automation and Systems. vol .9, pp. 958-965, 2011.

[19] B. Yang, J. Huang, X. Chen, C. Xiong and Y, Hasegawa, "Supernumerary Robotic Limbs: A Review and Future Outlook." vol. 3, no. 3, pp.623-639, 2021

[20] Y. Chen and J. Wang, "Trajectory Tracking Control for Autonomous Vehicles in Different Cut-in Scenarios," 2019 American Control Conference (ACC), pp. 4878-4883, 2019.

[21] H. Wang, B. Liu, X. Ping, Q. An, "Path Tracking Control for Autonomous Vehicles Based on an Improved MPC." IEEE Access. PP. 1-1. 10.1109, 2019.

[22] W. Zhang, A robust lateral tracking control strategy for autonomous driving vehicles, Mechanical Systems and Signal Processing, vol 150, pp. 107238, 2021.

[23] X. Wang, X. Yu and W. Sun, "A Path Planning and Tracking Control for Autonomous Vehicle With Obstacle Avoidance," 2020 Chinese Automation Congress (CAC), 2020, pp. 2973-2978

[24] J. Kong, M. Pfeiffer, G. Schildbach and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," 2015 IEEE Intelligent Vehicles Symposium (IV), 2015, pp. 1094-1099