Proceedings of the 2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI), Hangzhou, China, December 18-20, 2020

Trajectory tracking control for autonomous parking using reduced-horizon model predictive control

1st Zhiming Zhang
State Key Laboratory of Industrial Control Technology
Zhejiang University
Hangzhou, China
zhangzhimingzju@zju.edu.cn

2nd Lei Xie
State Key Laboratory of Industrial Control Technology
Zhejiang University
Hangzhou, China
leix@iipc.zju.edu.cn

3rd Hongye Su
State Key Laboratory of Industrial Control Technology
Zhejiang University
Hangzhou, China
hysu69@zju.edu.cn

2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI) | 978-1-7281-8497-5/20/$31.00 ©2020 IEEE | DOI: 10.1109/CVCI51460.2020.9338599

***Abstract*—In this paper, a reduced-horizon model predictive control strategy is proposed for autonomous parking. Given the planned trajectory, the linear time varying model is obtained by discretizing and linearizing along the reference trajectory. Then the linear time varying model is represented in an incremental form. A standard quadratic programming problem is formulated which can be solved online. The joint simulation in Simulink and CarSim in three parking scenarios shows the effectiveness of the proposed method.**

***Index Terms*—Autonomous parking, tracking control, model predictive control, reduced-horizon control.**

I. Introduction

Nowadays, parking assistance systems have become standard for senior automotive. At the same time, autonomous parking has received a lot of attention from the business and academic. Autonomous parking is expected to free human drivers and reduce the accidents caused by human drivers. In general, the autonomous parking system consists of three parts: (i) environment perception, constructing an accurate map including the information of the spot and other vehicles, (ii) path planning, giving a feasible, safe, collision free trajectory or path, (iii) trajectory tracking, controlling the actuator to drive the vehicle along the trajectory as close as possible.

The planned trajectory is usually generated by two means. The first class is path planning and replanning. That is, planning a path from the given initial position and orientation and final position and orientation with some path planning methods such as Dijkstra's algorithm [1], A* algorithm [2], RRT algorithm [3] firstly. The planned path is often unsmooth and more importantly, it is without time information and cannot be called as a trajectory. Then replanning is necessary to design the dynamic of the path. At this stage, considering the vehicle dynamics or kinematics in the replanning problem will be helpful for the trajectory tracking, which is also a trend in the research [2].

The second class is to plan the trajectory directly. The numerical optimization algorithm aims to obtain a trajectory by solving a constrained dynamic optimization problem [4]–[6]. According to the driving mission, an objective function is defined where vehicle kinematics, obstacle avoidance and physical limitations are considered as constraints.

As for trajectory tracking control, the number of literatures on tracking control has been increasing recently. The kinematics model and Kanayama's controller have been used to control the vehicle via the error vector and nonlinear feedback in [7]. The concept of the redesigned virtual trajectory has been proposed in [8] to guarantee practical stability with exponential convergence. Recently, model predictive control (MPC) has been widely studied in the control field for its capacity to handle the constraints systematically and optimize the control on basis of the model [9]. However, the computation is high especially when there is nonlinearity in the optimization problem. In order to address this problem, reference [10] has used the time-state control form to suppress the computational effort. Beyond that, linearizing the vehicle model along the reference trajectory is also an effective method [11], [12].

In the parking mission, the vehicle should stop at the target spot from current position. Due to the narrow spot and surrounding obstacles such as other vehicles, the vehicle needs to switch the gear and goes forwards and backwards during the parking. In this paper, the linear time varying (LTV) model is derivated from the reference trajectory. Then the state space function from the present time instant to the final time instant is deduced and added into the objective function hence the switch point of changing direction is trivial. At each time instant, a reduced-horizon model predictive control problem is converted to a standard quadratic programming (QP) problem which can be solved fast online. A joint simulation in Simulink and CarSim is implemented where the vehicle needs to park in three different scenarios, namely parallel, reverse, diagonal parking. The result shows the proposed method is effective.

The rest of paper is organized as follows. Section II introduces the kinematics model of the vehicle which is suitable in the low speed scenario, the linearization and discretization is carried out along the reference trajectory to get a LTV model. A reduced-horizon model predictive control strategy is presented in Section III and the optimization problem is transformed to a standard QP problem. The whole simulation scheme and results of three parking scenarios are shown in Section IV, which is followed by the conclusion and future work in Section V.

II. Model of the vehicle

The trajectory tracked by a motion controller is generated by solving the following trajectory planning problem: *given the initial position and orientation and final position and orientation, finding a feasible curve that an autonomous vehicle can move along.* Generally, the trajectory can be obtained by some optimal-based trajectory planning methods or re-optimizing the paths of some path planning methods. The goal of MPC-based motion controller is to track the trajectory accurately and to eliminate the disturbances by a sequence of suitable control inputs such as speed (or torque) and steering angle.

A. Kinematics model

The load transfer and the side-slip of tires can be ignored when a vehicle is parking at low speed [13]. The kinematics model of a front-steering vehicle is used in this paper as shown in Fig.1.

$$\begin{cases} \dot{X} = v \cdot \cos(\varphi) \\ \dot{Y} = v \cdot \sin(\varphi) \\ \dot{\varphi} = v \cdot L^{-1} \tan(\delta_f) \end{cases} \triangleq \dot{\xi} = \quad f(\xi, u) \tag{1}$$

where (X, Y) is the position of the center of the rear axle in the global coordinates and φ is the orientation of the vehicle. δ_f is the steering angle of the front wheel. The velocity at the center of the rear axle is represented by v. L is the wheelbase. The left part of Eq.(1) can be denoted in a compact form as the right part, where $\xi = [X, Y, \varphi]^T$ and $u = [v, \delta_f]^T$.

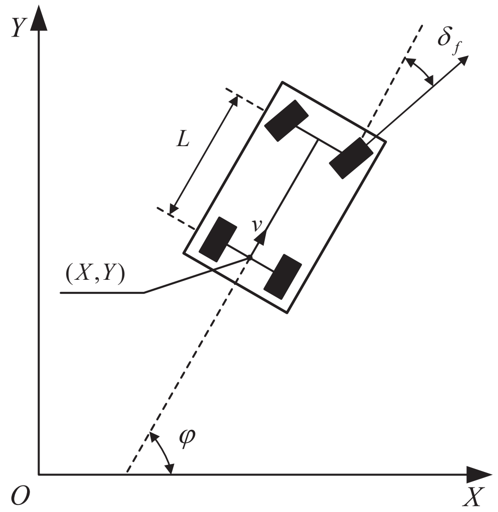

Fig. 1. Notations of the kinematics model.

B. Discretization and Linearization

The vehicle model is nonlinear hence the linearization is necessary to decrease the computation. Given a $(N+1)$-length parking trajectory $\{\xi_r(k), u_r(k)\}_{k=0}^N$ with sample time or control period T_s, the discretization and linearization of Eq.(1) is given by:

$$\tilde{\xi}(k+1) = A(k)\tilde{\xi}(k) + B(k)\tilde{u}(k) \tag{2}$$

where

$$\tilde{\xi}(k) = \begin{bmatrix} \tilde{X}(k) \\ \tilde{Y}(k) \\ \tilde{\varphi}(k) \end{bmatrix} = \begin{bmatrix} X(k) \\ Y(k) \\ \varphi(k) \end{bmatrix} - \begin{bmatrix} X_r(k) \\ Y_r(k) \\ \varphi_r(k) \end{bmatrix},$$

$$\tilde{u}(k) = \begin{bmatrix} \tilde{v}(k) \\ \tilde{\delta}_f(k) \end{bmatrix} = \begin{bmatrix} v(k) \\ \delta_f(k) \end{bmatrix} - \begin{bmatrix} v_r(k) \\ \delta_{f,r}(k) \end{bmatrix}$$

and

$$A(k) = \begin{bmatrix} 1 & 0 & -v_r(k)T_s \sin\varphi_r(k) \\ 0 & 1 & v_r(k)T_s \cos\varphi_r(k) \\ 0 & 0 & 1 \end{bmatrix},$$

$$B(k) = \begin{bmatrix} T_s \cos\varphi_r(k) & 0 \\ T_s \sin\varphi_r(k) & 0 \\ T_s L^{-1} \tan\delta_{f,r}(k) & T_s v_r(k) L^{-1} \cos^{-2}\delta_{f,r}(k) \end{bmatrix}.$$

Then the discrete state space function can be denoted by

$$\xi(k+1) \quad = \quad A(k)\xi(k) + B(k)u(k) + d(k) \tag{3}$$

where $d(k) = \xi_r(k+1) - A(k)\xi_r(k) - B(k)u_r(k)$.

III. Reduced-horizon model predictive control for autonomous parking

The tracked trajectory is a sequence of positions and orientations with a finite time horizon, i.e., the vehicle should complete the parking mission in settled time. However, the vehicle usually cannot perform well directly in the open loop due to the model mismatch and disturbance. In this section, a reduced-horizon model predictive control method is proposed to eliminate the model mismatch and disturbance.

A. Reduced-horizon model predictive control

Reorganize Eq.(3) in an incremental form as follows

$$\underbrace{\begin{bmatrix} \xi(k+1) \\ u(k) \end{bmatrix}}_{\hat{\xi}(k+1)} = \underbrace{\begin{bmatrix} A(k) & B(k) \\ 0 & I \end{bmatrix}}_{\hat{A}(k)} \underbrace{\begin{bmatrix} \xi(k) \\ u(k-1) \end{bmatrix}}_{\hat{\xi}(k)} + \underbrace{\begin{bmatrix} B(k) \\ I \end{bmatrix}}_{\hat{B}(k)} \Delta u(k) + \underbrace{\begin{bmatrix} d(k) \\ 0 \end{bmatrix}}_{\hat{d}(k)}$$

i.e.,

$$\hat{\xi}(k+1) = \hat{A}(k)\hat{\xi}(k) + \hat{B}(k)\Delta u(k) + \hat{d}(k) \tag{4}$$

At each time instant, the state space function from $k+1$ to N is

$$\mathbf{\Xi}_{k+1} = \mathbf{\Phi}_k \hat{\xi}(k) + \mathbf{\Psi}_{u,k} \Delta\mathbf{u}_k + \mathbf{\Psi}_{d,k} \mathbf{d}_k \tag{5}$$

where

$$\mathbf{\Xi}_{k+1} = \left[\hat{\xi}(k+1)^T \ \hat{\xi}(k+2)^T \ \cdots \ \hat{\xi}(N)^T\right]^T,$$
$$\Delta\mathbf{u}_k = \left[\Delta u(k)^T \ \Delta u(k+1)^T \ \cdots \ \Delta u(N-1)^T\right]^T,$$
$$\mathbf{d}_k = \left[\hat{d}(k)^T \ \hat{d}(k+1)^T \ \cdots \ \hat{d}(N-1)^T\right]^T.$$

The matrices of $\mathbf{\Phi}$, $\mathbf{\Psi_u}$, $\mathbf{\Psi_d}$ and $\mathbf{\Gamma}$ are

$$\mathbf{\Phi}_k = \begin{bmatrix} \hat{A}(k) \\ \hat{A}(k+1)\hat{A}(k) \\ \vdots \\ \prod_{i=k}^{N-1} \hat{A}(i) \end{bmatrix},$$

$$\mathbf{\Psi}_{u,k} = \begin{bmatrix} \hat{B}(k) & \cdots & 0 \\ \hat{A}(k+1)\hat{B}(k) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \prod_{i=k+1}^{N-1} \hat{A}(i)\hat{B}(k) & \cdots & \hat{B}(N-1) \end{bmatrix},$$

$$\mathbf{\Psi}_{d,k} = \begin{bmatrix} I & 0 & \cdots & 0 \\ \hat{A}(k+1) & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \prod_{i=k+1}^{N-1} \hat{A}(i) & \prod_{i=k+2}^{N-1} \hat{A}(i) & \cdots & I \end{bmatrix}$$

Remark 1: Different from conditional MPC, reduced-horizon model predictive control needs to adjust the prediction horizon to ensure the vehicle stops at the target spot at the terminal time instant. Control horizon and prediction horizon are the same and equal to $N-k+1$. The incremental form Eq.(4) can improve the stability and unbiasedness of the controller.

B. Optimization problem

At every time instant, model predictive control reformulates a new optimization problem on basis of the model (5). A series of control moves are obtained by solving the optimization problem but only the first control move is applied into the controlled object such as a vehicle. In this subsection, the formulation of reduced-horizon model predictive control for autonomous parking is introduced.

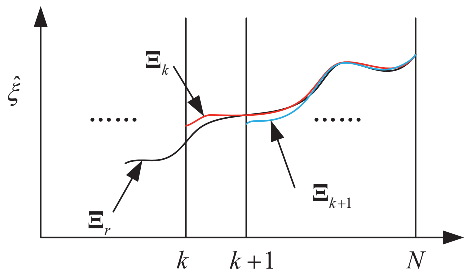

Fig. 2. The diagram of reduced-horizon model predictive control.

As shown in Fig.2, at the present time instant k, the state which is denoted by the following cost function

$$\min_{\Delta u(k),\cdots,\Delta u(N)} \sum_{i=k}^{N} \|\Delta u(i)\|_R^2 + \sum_{i=k+1}^{N+1} \|\hat{C}\hat{\xi}(i) - \xi_r(i)\|_Q^2 \quad (6)$$

s.t.

$$\hat{\xi}(k+1) = \hat{A}(k)\hat{\xi}(k) + \hat{B}(k)\Delta u(k) + \hat{d}(k)$$
$$\Delta u_{min} \leq \Delta u(i) \leq \Delta u_{max}$$

where

$$\hat{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

to extract the position and orientation from $\hat{\xi}(i)$. Eq.(6) is a quadratic programming (QP) problem, which can be represented in a compact form with the definition in Section III-A. The first term can be given by

$$\sum_{i=k}^{N} \|\Delta_u(i)\|_R^2 = \Delta\mathbf{u}_k^T R \Delta\mathbf{u}_k$$

and the second term can be denoted by

$$\begin{aligned} &\sum_{i=k+1}^{N+1} \|C\hat{\xi}(i) - \xi_r(i)\|_Q^2 \\ &= (\mathbf{\Gamma}(\mathbf{\Xi}_{k+1} - \mathbf{\Xi}_{r,k+1}))^T Q\mathbf{\Gamma}(\mathbf{\Xi}_{k+1} - \mathbf{\Xi}_{r,k+1}) \\ &= \mathbf{\Xi}_{k+1}^T \mathbf{\Gamma}^T Q \mathbf{\Gamma} \mathbf{\Xi}_{k+1} - 2\mathbf{\Xi}_{r,k+1}^T \mathbf{\Gamma}^T Q \mathbf{\Gamma} \mathbf{\Xi}_{k+1} \\ &= \Delta\mathbf{u}_k^T \mathbf{\Psi}_{u,k}^T \mathbf{\Gamma}^T Q \mathbf{\Gamma} \mathbf{\Psi}_{u,k} \Delta\mathbf{u}_k \\ &\quad + 2(\hat{\xi}^T(k)\mathbf{\Phi}_k^T + \mathbf{d}_k^T \mathbf{\Psi}_{d,k}^T)\mathbf{\Gamma}^T Q \mathbf{\Gamma} \mathbf{\Psi}_{u,k} \Delta\mathbf{u}_k \\ &\quad - 2\mathbf{\Xi}_{r,k+1}^T \mathbf{\Gamma}^T Q \mathbf{\Gamma} \mathbf{\Psi}_{u,k} \Delta\mathbf{u}_k \end{aligned}$$

where

$$\mathbf{\Gamma} = diag(\underbrace{[\hat{C},\ \hat{C},\ \cdots,\ \hat{C}]}_{N-k+1})$$

In this way problem (6) is transformed as a standard QP form which can be solve by many existing QP solvers [14].

Remark 2: Here no extra constraints such as collision avoidance constraints are considered in problem (6) since these constraints have been taken into account at the planning stage.

IV. Simulation and Analysis

In this section, simulation results in three common parking scenarios are presented. CarSim is employed to simulate the real vehicle and the reduced-horizon MPC controller is an S-Function in Matlab. All simulations are performed on a Windows 10 OS with an Intel i5 processor clocked at 1.8GHz and the programming language is Matlab with version 2019b.

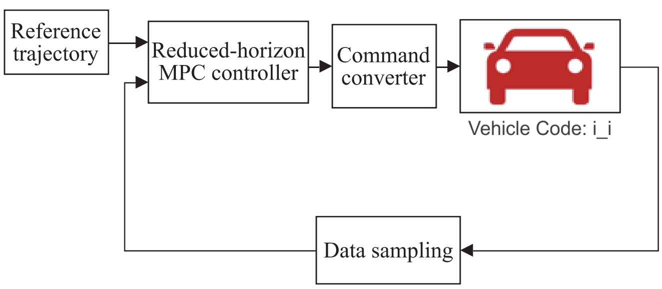

Fig. 3. The control scheme of parking trajectory tracking.

The whole planning and control scheme can be described as Fig.3. The reference trajectory is obtained by some parking trajectory planning methods such as [6], [15]. Then the reduced-horizon MPC controller calculates the speed and steer angle for tracking the trajectory. A command converter is used here to shift gears (forwards or backwards) according to the speed. Without regard to the contour of the vehicle and the collision avoidance, the controller concentrates on tracking the planned trajectory. The only used parameter of the vehicle is the wheelbase, $L = 3.05m$. The detail controller parameter is detailed in the following subsections.

A. Scenario 1: Reverse parking

In the reverse parking scenario, the parking spot is $2.6m$ wide and $5.3m$ long. The vehicle starts at a $6m$ wide road for forwards and backwards maneuvering. In Fig.4, the actual trajectory is shown in blue, which is close to the planned one.

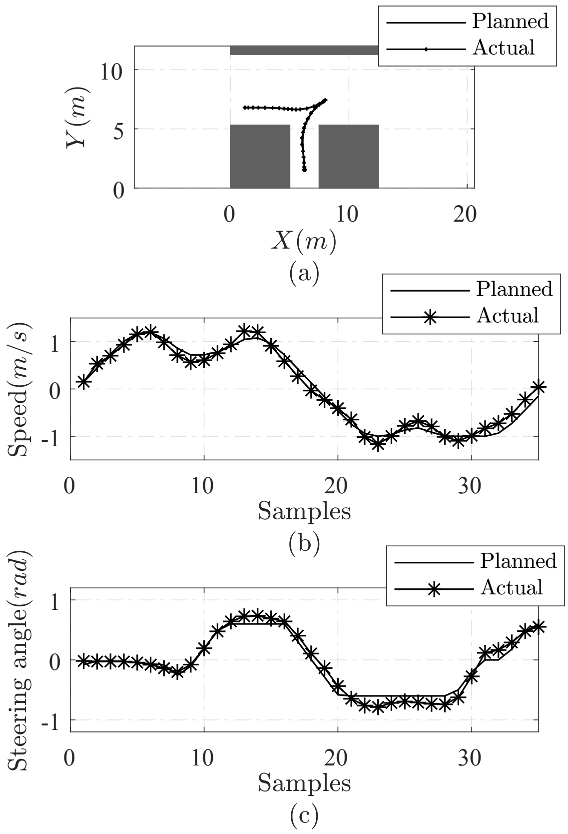

Fig. 4. Scenario 1: Reverse parking. (a) The comparison between the planned trajectory and actual trajectory. (b) The speed command given by reduced-horizon MPC controller. (c) The steering angle command given by reduced-horizon MPC controller.

In Fig.4(b) and (c) the control commands calculated by the MPC controller are compared with the reference command. The speed command especially the steering angle command is a little different from the reference ones. Apparently, Due to the model mismatch and disturbance, there will be a large deviation if the vehicle drives according to the reference command directly. The length of the reference trajectory is $N = 35$ and the control period is $0.5s$ hence the vehicle finished the parking task in $17.5s$. The weight matrices used in the controller are $R = diag([100, 200])$ and $Q = diag([500, 500, 500])$. The upper and lower limits of $\Delta u(i)$ is $[-1, 1]m/s$ and $[-0.5, 0.5]rad$.

B. Scenario 2: Parallel parking

In the parallel parking (see Fig.5), the length of the reference trajectory is $N = 42$ and the control period is $0.97s$ which is reasonable since the parallel parking is usually more challenging than the reverse parking. The size of the spot in this scenario is $2.5m$ wide and $6m$ long. Since the wheelbase of the vehicle is long, it is worth noting that, at the final stage, the vehicle goes forward and turns right to align with the spot, just like a human driver does. This action is necessary both in the human driver parking and the autonomous parking. The constraints and weight matrices used in this scenario are the same with the reverse parking.

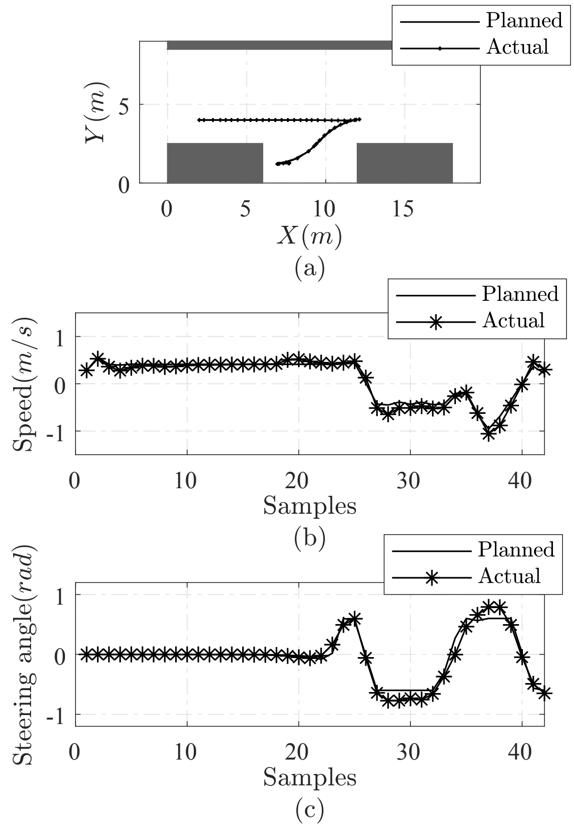

Fig. 5. Scenario 2: Parallel parking. (a) The comparison between the planned trajectory and actual trajectory. (b) The speed command given by reduced-horizon MPC controller. (c) The steering angle command given by reduced-horizon MPC controller.

C. Scenario 3: Diagonal parking

Diagonal parking is not very common in daily life. The difficulty of diagonal parking is between that of reverse parking and parallel parking. In this scenario, the spot is $2.5m$ wide and $6m$ long and faces at 45 degrees. The length of the reference trajectory is $N = 46$ and control period is $0.5s$. Fig.6 shows that the controller is still effective and gives a very small tracking error. The constraints and weight matrices used in diagonal parking scenario are the same with the other two scenarios.

As for real time, the time consumption in the three scenario is pretty small (see Fig.7). In addition, the trends of the time consumption in the three figures are downward since the horizon at each control time instant is diminishing.

V. Conclusion

In this paper, a reduced-horizon model predictive control method is proposed for autonomous parking. The kinematic model of the vehicle is linearized along the given reference trajectory and a linear time varying model is obtained. To handle the action of shifting gear, the state space function from the present time instant to the terminal time instant is presented to keeping the consistency of the control action.

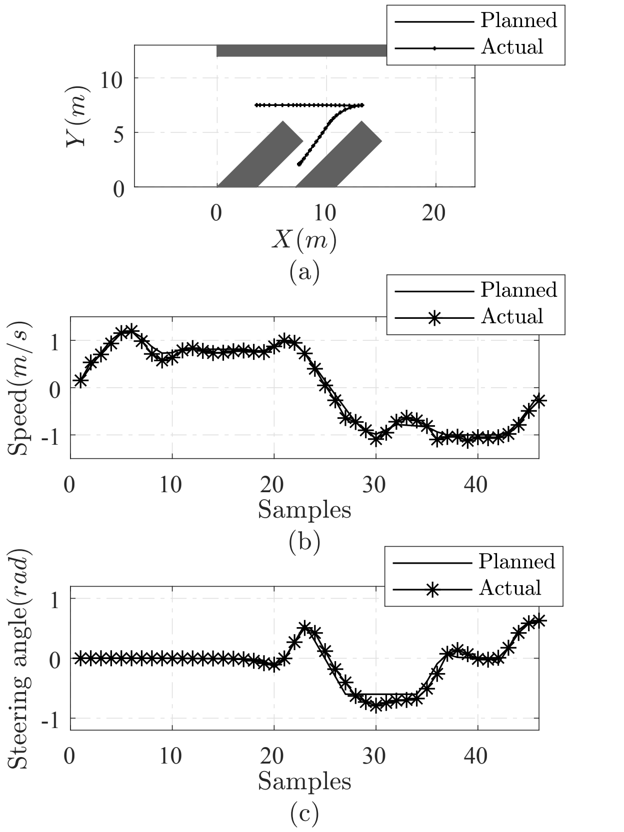

Fig. 6. Scenario 3: Diagonal parking. (a) The comparison between the planned trajectory and actual trajectory. (b) The speed command given by reduced-horizon MPC controller. (c) The steering angle command given by reduced-horizon MPC controller.

Then the optimization problem is reformulated as a standard quadratic programming problem, which can be solved by many existing QP solvers. Simulation results with Simulink and CarSim illustrate the effectiveness of the proposed method. The vehicle has tracked the planned trajectory in three parking scenarios with tiny tracking errors. The time consumption of the method is also practical.

In the future work, the implementation on a real car robot is necessary to verify the practicality.

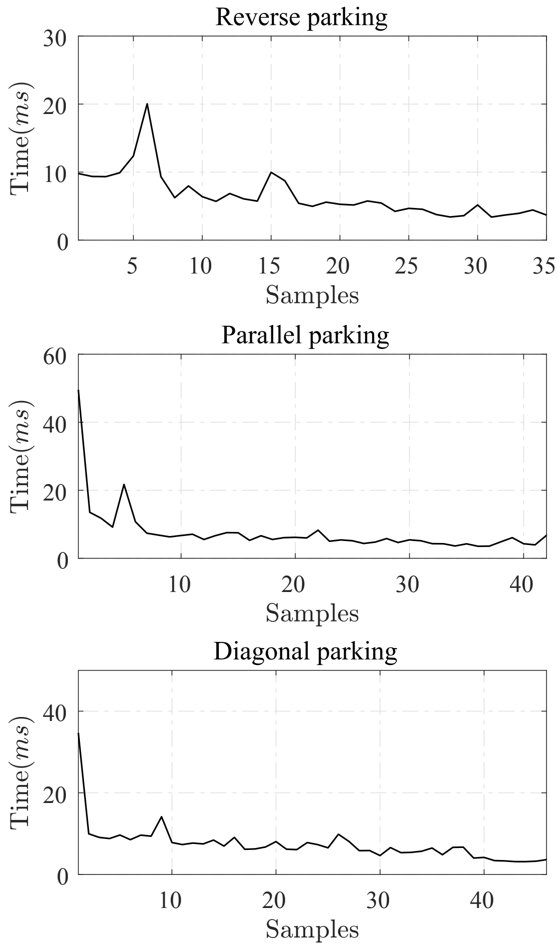

Fig. 7. Time consumption of reduced-horizon MPC controller in three scenarios.

ACKNOWLEDGEMENT

This work is supported by National Key Research and Development Program of China (NO. 2016YFB0303404).

REFERENCES

[1] H. Wang, Y. Yu, and Q. Yuan, "Application of dijkstra algorithm in robot path-planning," in *2011 second international conference on mechanic automation and control engineering*. IEEE, 2011, pp. 1067–1069.

[2] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 485–501, 2010.

[3] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The international journal of robotics research*, vol. 20, no. 5, pp. 378–400, 2001.

[4] T. Gu and J. M. Dolan, "On-road motion planning for autonomous vehicles," in *International Conference on Intelligent Robotics and Applications*. Springer, 2012, pp. 588–597.

[5] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for berthaa local, continuous method," in *2014 IEEE intelligent vehicles symposium proceedings*. IEEE, 2014, pp. 450–457.

[6] Z. Zhang, S. Lu, L. Xie, H. Su, D. Li, Q. Wang, and W. Xu, "A guaranteed collision-free trajectory planning method for autonomous parking," *IET Intelligent Transport Systems*, 2020.

[7] K. Lee, D. Kim, W. Chung, H. W. Chang, and P. Yoon, "Car parking control using a trajectory tracking controller," in *2006 SICE-ICASE International Joint Conference*. IEEE, 2006, pp. 2058–2063.

[8] T.-C. Lee, C.-Y. Tsai, and K.-T. Song, "Fast parking control of mobile robots: a motion planning approach with experimental validation," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 5, pp. 661–676, 2004.

[9] Z. Zhang, J. A. Rossiter, L. Xie, and H. Su, "Predictive functional control for integrator systems," *Journal of the Franklin Institute*, 2020.

[10] K. Oyama and K. Nonaka, "Model predictive parking control for non-holonomic vehicles using time-state control form," in *2013 European Control Conference (ECC)*. IEEE, 2013, pp. 458–465.

[11] D. Gu and H. Hu, "Receding horizon tracking control of wheeled mobile robots," *IEEE Transactions on control systems technology*, vol. 14, no. 4, pp. 743–749, 2006.

[12] H. Ye, H. Jiang, S. Ma, B. Tang, and L. Wahab, "Linear model predictive control of automatic parking path tracking with soft constraints," *International Journal of Advanced Robotic Systems*, vol. 16, no. 3, p. 1729881419852201, 2019.

[13] T.-h. Hsu, J.-F. Liu, P.-N. Yu, W.-S. Lee, and J.-S. Hsu, "Development of an automatic parking system for vehicle," in *2008 IEEE Vehicle Power and Propulsion Conference*. IEEE, 2008, pp. 1–6.

[14] P. Patrinos and A. Bemporad, "An accelerated dual gradient-projection algorithm for embedded linear model predictive control," *IEEE Transactions on Automatic Control*, vol. 59, no. 1, pp. 18–33, 2013.

[15] X. Zhang, A. Liniger, A. Sakai, and F. Borrelli, "Autonomous parking using optimization-based collision avoidance," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 4327–4332.