Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5593033, 9 pages
https://doi.org/10.1155/2021/5593033

Research Article

Efficient Path Tracking Control for Autonomous Driving of Tracked Emergency Rescue Robot under 6G Network

Qing Gu,[1,2] Guoxing Bai,[1] Yu Meng,[1] Guodong Wang,[1] Jiazang Zhang,[3] and Lei Zhou[1]

[1]*School of Mechanical Engineering, University of Science and Technology Beijing, Beijing 100083, China*
[2]*Shunde Graduate School, University of Science and Technology Beijing, Foshan 528399, China*
[3]*Zhejiang College of Security Technology, Wenzhou 325016, China*

Correspondence should be addressed to Yu Meng; myu@ustb.edu.cn

Received 23 January 2021; Revised 8 March 2021; Accepted 5 April 2021; Published 16 April 2021

Academic Editor: Li-Xin Li

This paper proposes a path tracking control algorithm of tracked mobile robots based on Preview Linear Model Predictive Control (MPC), which is used to achieve autonomous driving in the unstructured environment under an emergency rescue scenario. It is the future trend to realize the communication and control of rescue equipment with 6G and edge cloud cooperation. In this framework, linear MPC (LMPC) is suitable for the path tracking control of rescue robots due to its advantages of less computing resources and good real-time performance. However, in such a scene, the driving environment is complex and the path curvature changes greatly. Since LMPC can only introduce linearized feedforward information, the tracking accuracy of the path with large curvature changes is low. To overcome this issue, combined with the idea of preview control, preview-linear MPC is designed in this paper. The controller is verified by MATLAB/Simulink simulation and prototype experiment. The results show that the proposed method can improve the tracking accuracy while ensuring real-time performance and has better tracking performance for the path with large curvature variation.

1. Introduction

Due to strong mobility and terrain adaptability, tracked mobile robots are widely used in the rescue environment. In the harsh environment of rescue, it is difficult to build a stable and high-quality communication network. Therefore, it is the future trend to use 6G combined with edge cloud to realize the communication and control of rescue equipment. In this framework, a simple and efficient mobile robot control algorithm is conducive to better accomplish the rescue mission. This is because the short control value computing time is conducive to increasing the information sampling frequency between edge and clouds, which can improve the synergistic ability between edge and clouds. In this case, Linear Model Predictive Control (LMPC) is suitable for the path tracking control of rescue mobile robots. Model Predictive Control (MPC) is a multivariable control strategy, which can obtain the optimal control variables in the prediction time domain through the state equation of the controlled object. Since it does not need the precise model and can deal with the system state constraints, compared with other classical control algorithms (such as Fuzzy, linear quadratic regulator (LQR), and sliding mode control (SMC)), MPC has some advantages [1] and has been widely used in the field of vehicle and robot control [2–6]. Among which, Linear Model Predictive Control (LMPC) has the advantage of faster calculation due to the use of model linearization technology. However, different from general road driving, the rescue scene environment is complex, and the curvature of the target path often changes greatly. Since the feedforward information of the LMPC controller is linearized feedforward information, when the curvature of the reference path changes greatly, the feedforward information obtained through linear prediction is not accurate [5]. Therefore,

when tracking such a reference path, the accuracy of the LMPC controller is poor. Therefore, how to make the tracked rescue mobile robot follow such a path is an urgent problem.

In recent years, the research results of path tracking control based on LMPC have not paid enough attention to this issue [4–17]. However, in the path tracking control based on feedforward-feedback control, the idea of introducing feedforward information through the preview to improve the control accuracy appears. Meng et al. [18] and Xu and Peng [19], respectively, proposed a path tracking control method based on the preview-linear quadratic regulator (LQR). Diao et al. proposed a path tracking control method based on preview-fuzzy control [20]. These studies have proved the effectiveness of introducing feedforward control. However, considering that control methods such as LQR and fuzzy control have no advantages over LMPC in terms of handling system constraints, etc. [21–23], it is still necessary to study the path tracking control method combining preview control and LMPC.

In [24], we discussed the possibility of introducing feedforward information into LMPC through preview control. However, in [24], the strategy of reducing the longitudinal speed during steering has also been introduced. Considering that the longitudinal speed is also an important factor affecting the accuracy of path tracking control [5, 25], whether the preview control can improve the accuracy of LMPC-based path tracking control when the longitudinal speed is constant is still a question to be studied.

2. LMPC Controller

Firstly, the LMPC controller is designed. The prediction model of the LMPC controller can be designed based on the following kinematics model of tracked mobile robots:

$$\begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \\ \dot{\theta} = \omega \end{cases}, \tag{1}$$

where x, y, and θ are the abscissa, ordinate and heading angle of the plane motion center of the mobile robot, respectively. v and ω are longitudinal velocity and yaw rate, respectively.

The specific process of controller design can refer to [25–28]. First, perform Jacobian Linearization on equation (1), that is, perform Taylor expansion on equation (1) at any point $\mathbf{x}_0(x_0, y_0, \theta_0, v_0, \omega_0)$ and retain the first-order term. Then, the discretized prediction model can be obtained through discretization:

$$\tilde{\mathbf{x}}(t+1 \mid t) = \mathbf{A}\tilde{\mathbf{x}}(t \mid t) + \mathbf{B}\Delta\mathbf{u}(t \mid t), \tag{2}$$

where

$$\begin{aligned} \tilde{\mathbf{x}}(t \mid t) &= \begin{bmatrix} x(t \mid t) - x_0 \\ y(t \mid t) - y_0 \\ \theta(t \mid t) - \theta_0 \end{bmatrix}, \\ \Delta\mathbf{u}(t \mid t) &= \begin{bmatrix} v(t \mid t) - v_0 \\ \omega(t \mid t) - \omega_0 \end{bmatrix}, \\ \mathbf{A} &= \begin{bmatrix} 1 & 0 & -Tv_0\sin\theta \\ 0 & 1 & v_0 T\cos\theta \\ 0 & 0 & 1 \end{bmatrix}, \\ \mathbf{B} &= \begin{bmatrix} T\cos\theta & 0 \\ T\sin\theta & 0 \\ 0 & T \end{bmatrix}, \end{aligned} \tag{3}$$

where $\tilde{\mathbf{x}}(t+i \mid t)$ is the ith predictive state at time t, $\mathbf{u}(t+i \mid t)$ is the ith control input at time t, and T is control cycle.

Assuming that the prediction horizon is N_p and the control horizon is N_c, the state variables in the prediction horizon are

$$\tilde{\mathbf{x}}(t+N_p \mid t) = \mathbf{A}^{N_p}\tilde{\mathbf{x}}(t \mid t) + \mathbf{A}^{N_p-1}\mathbf{B}\Delta\mathbf{u}(t \mid t) + \cdots + \mathbf{A}^{N_p-N_c}\mathbf{B}\Delta\mathbf{u}(t+N_c-1 \mid t). \tag{4}$$

The future output of the system can be represented as a matrix

$$\tilde{\mathbf{Y}}(t) = \mathbf{\Psi}\tilde{\mathbf{x}}(t \mid t) + \mathbf{\Theta}\Delta\mathbf{U}(t), \tag{5}$$

where

$$\begin{aligned} \tilde{\mathbf{Y}}(t) &= \begin{bmatrix} \tilde{\mathbf{x}}(t+1 \mid t) \\ \vdots \\ \tilde{\mathbf{x}}(t+N_c \mid t) \\ \vdots \\ \tilde{\mathbf{x}}(t+N_p \mid t) \end{bmatrix}, \\ \Delta\mathbf{U}(t) &= \begin{bmatrix} \Delta\mathbf{u}(t \mid t) \\ \vdots \\ \Delta\mathbf{u}(t+N_c-1 \mid t) \end{bmatrix}, \\ \mathbf{\Psi} &= \begin{bmatrix} \mathbf{A} & \mathbf{A}^2 & \cdots & \mathbf{A}^{N_c} & \cdots & \mathbf{A}^{N_p} \end{bmatrix}^{\mathrm{T}}, \\ \mathbf{\Theta} &= \begin{bmatrix} \mathbf{B} & \cdots & 0 \\ \mathbf{AB} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \mathbf{A}^{N_c-1}\mathbf{B} & \cdots & \mathbf{B} \\ \mathbf{A}^{N_c}\mathbf{B} & \cdots & \mathbf{AB} \\ \vdots & \cdots & \vdots \\ \mathbf{A}^{N_p-1}\mathbf{B} & \cdots & \mathbf{A}^{N_p-N_c}\mathbf{B} \end{bmatrix}, \end{aligned} \tag{6}$$

Equation (5) is the prediction model of the LMPC controller.

The objective function in the rolling optimization process can be designed as

$$J(\tilde{\mathbf{x}}(t), \Delta\mathbf{U}(t)) = \sum_{i=1}^{N_p} \left\| \tilde{\mathbf{x}}(t+i \mid t) - \tilde{\mathbf{x}}_{ref}(t+i \mid t) \right\|_{\mathbf{Q}}^2 + \sum_{i=1}^{N_c} \left\| \Delta\mathbf{u}(t+i \mid t) \right\|_{\mathbf{R}}^2, \tag{7}$$

where $\tilde{\mathbf{x}}_{ref}$ is linearized state information for tracking target points and $\mathbf{Q}$ and $\mathbf{R}$ are the weight matrices of tracking error and control input increment, respectively.

Equation (7) can be abstracted as

$$J(\tilde{\mathbf{x}}(t), \Delta\mathbf{U}(t)) = \left\| \tilde{\mathbf{Y}}(t) - \tilde{\mathbf{Y}}_{ref}(t) \right\|_{\widehat{\mathbf{Q}}}^2 + \left\| \Delta\mathbf{U}(t) \right\|_{\widehat{\mathbf{R}}}^2, \tag{8}$$

where

$$\begin{cases} \widehat{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} & & \\ & \ddots & \\ & & \mathbf{Q} \end{bmatrix}_{N_p \times N_p}, \\ \widehat{\mathbf{R}} = \begin{bmatrix} \mathbf{R} & & \\ & \ddots & \\ & & \mathbf{R} \end{bmatrix}_{N_c \times N_c} \end{cases} \tag{9}$$

because

$$\tilde{\mathbf{x}}_{ref}(t \mid t) = \begin{bmatrix} x_{ref}(t \mid t) - x_0 \\ y_{ref}(t \mid t) - y_0 \\ \theta_{ref}(t \mid t) - \theta_0 \end{bmatrix}, \tag{10}$$

where $x_{ref}(t \mid t)$, $y_{ref}(t \mid t)$, and $\theta_{ref}(t \mid t)$ are the abscissa, ordinate, and heading angle of the tracking target point, respectively.

As a result, when the reference input increment is zero, there are

$$\tilde{\mathbf{Y}}_{ref}(t) = \mathbf{\Psi}\tilde{\mathbf{x}}_{ref}(t \mid t). \tag{11}$$

Further

$$\tilde{\mathbf{Y}}(t) - \tilde{\mathbf{Y}}_{ref}(t) = \mathbf{\Psi}\tilde{\mathbf{x}}(t \mid t) - \mathbf{\Psi}\tilde{\mathbf{x}}_{ref}(t \mid t) = \mathbf{\Psi}\left(\tilde{\mathbf{x}}(t \mid t) - \tilde{\mathbf{x}}_{ref}(t \mid t)\right), \tag{12}$$

Assumption

$$\mathbf{e}(t) = \tilde{\mathbf{x}}(t \mid t) - \tilde{\mathbf{x}}_{ref}(t \mid t). \tag{13}$$

Equation (8) can be simplified into a standard quadratic form

$$J(\tilde{\mathbf{x}}(t), \Delta\mathbf{U}(t)) = \frac{1}{2}\Delta\mathbf{U}(t)^{\mathrm{T}}\mathbf{H}\Delta\mathbf{U}(t) + \mathbf{G}\Delta\mathbf{U}(t) + C, \tag{14}$$

where $\mathbf{H} = \mathbf{\Theta}^{\mathrm{T}}\widehat{\mathbf{Q}}\mathbf{\Theta} + \widehat{\mathbf{R}}$, $\mathbf{G} = 2\mathbf{\Theta}^{\mathrm{T}}\widehat{\mathbf{Q}}\mathbf{\Psi}\mathbf{e}(t)$, and $C = (\mathbf{\Psi}\mathbf{e}(t))^{\mathrm{T}}\widehat{\mathbf{Q}}\mathbf{\Psi}\mathbf{e}(t)$.

Combined with kinematic constraints, the path tracking control problem of mobile robots can be described as a quadratic programming problem

$$\begin{aligned} \min\ & J(\tilde{\mathbf{x}}(t), \Delta\mathbf{U}(t)) = \frac{1}{2}\Delta\mathbf{U}(t)^{\mathrm{T}}\mathbf{H}\Delta\mathbf{U}(t) + \mathbf{G}\Delta\mathbf{U}(t) + C \\ & s.t. -0.01\mathrm{rad/s} \le \omega \le 0.01\mathrm{rad/s}. \end{aligned} \tag{15}$$

Equation (15) can be solved using the "quadprog" function embedded in MATLAB, and the solving algorithm adopts the "active-set" method. Then, the sequence of control input increments in the control horizon can be obtained

$$\Delta\mathbf{U}^* = [\Delta\mathbf{u}^*(t \mid t), \cdots, \Delta\mathbf{u}^*(t + N_c \mid t)]^{\mathrm{T}}. \tag{16}$$

The control input for the next control cycle is

$$\mathbf{u}(t \mid t) = \Delta\mathbf{u}^*(t \mid t) + \mathbf{u}(t-1 \mid t). \tag{17}$$

3. Control Mechanism of Preview

Considering that the curvature of the reference path is limited and the mobile robot will not deviate too far from the reference path, there are the nearest points on the reference path to the plane motion center of the mobile robot. As shown in Figure 1, the point closest to the plane motion center of the mobile robot on the reference path is the intersection point between the normal line of the reference path passing through the P point and the reference path, which can be set as A_0. At the same time, the displacement error and heading error can be defined. P_1P_2 is the tangent line of the reference path at the point A_0, the lateral error e_d is the length of the line segment PP_0, and the heading error e_h is the deviation between the heading of the mobile robot and the heading of the reference path at the point P_0.

In the traditional LMPC controller, the point A_0 is the tracking target point in equation (10).

As shown in Figure 2, if point A_1, which is an arc length in front of the point A_0 on the reference path, is used as the tracking target point, substituting the abscissa, ordinate, and heading angle of point A_1 into equation (10), the preview-LMPC path tracking controller can be obtained. The arc length $\overset{\frown}{A_0A_1}$ is the preview distance.

4. Simulation Verification

The simulation platform is an ordinary desktop computer, the processor is Intel(R) Core(TM) i5-8500 @ 3.00 GHz, the

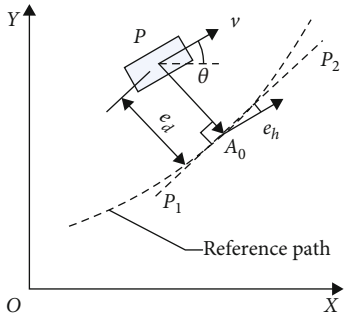

Figure 1: Schematic diagram of the point on the reference path that is closest to the plane motion center of the mobile robot.

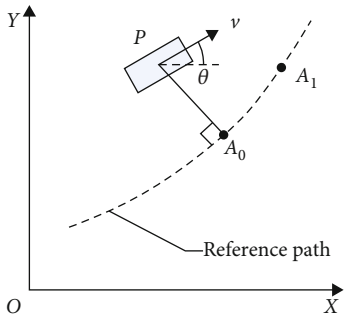

Figure 2: Schematic diagram of preview point.

Table 1: The main parameters of the controller.

Description	Symbol	Value
Prediction horizon	N_p	25
Control horizon	N_c	25
Control cycle	T	0.05 s

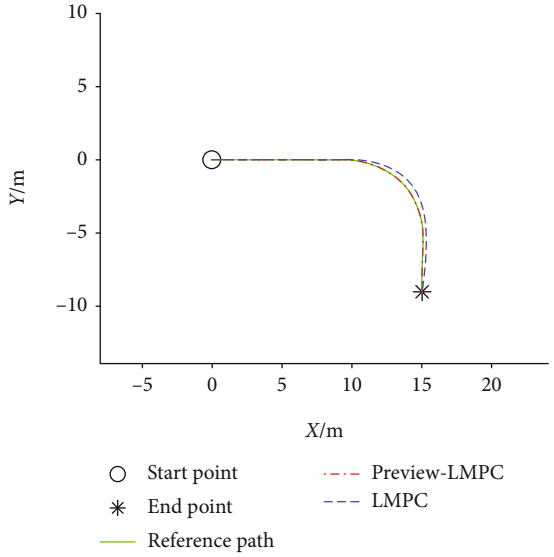

Figure 3: Trajectories of the first set of simulations.

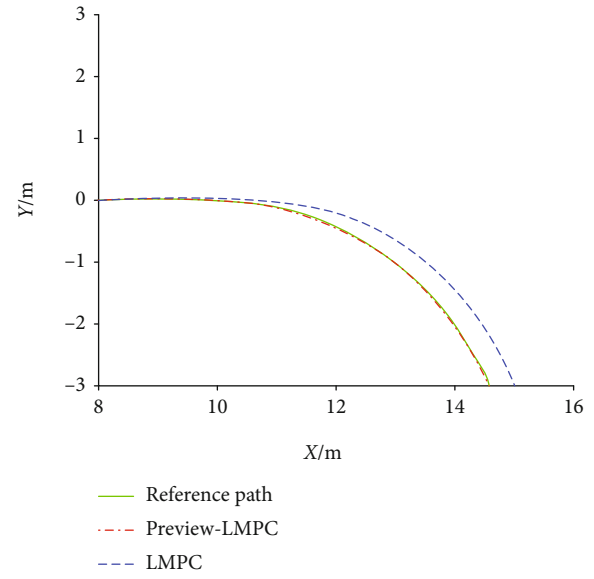

Figure 4: Partially enlarged view trajectories of the first set of simulations.

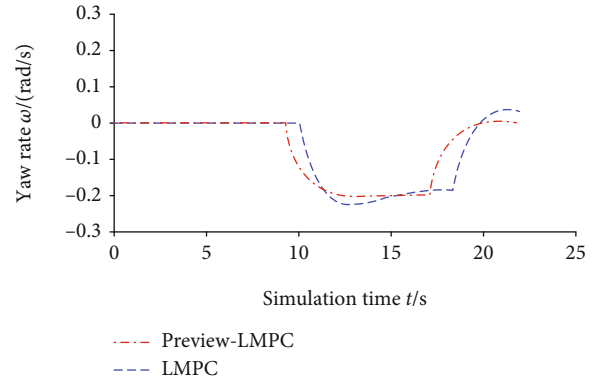

Figure 5: Yaw rate of the first set of simulations.

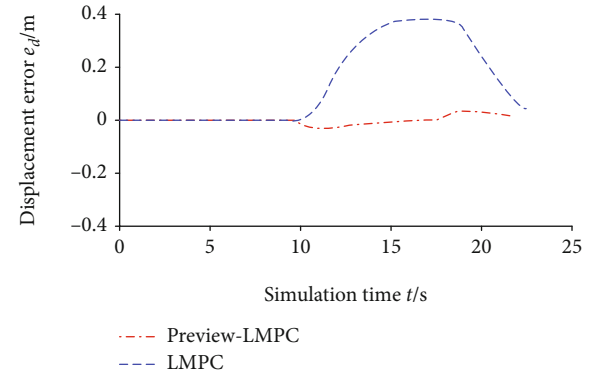

Figure 6: Displacement error of the first set of simulations.

memory is Kingston DDR4 2400 MHz 8 GB, the hard disk is Seagate ST1000DM010-2EP102, and the operating system is Windows 10 Home Chinese Edition. The simulation software uses MATLAB/Simulink R2018b. In this simulation, the main parameters of the controller are shown in Table 1.

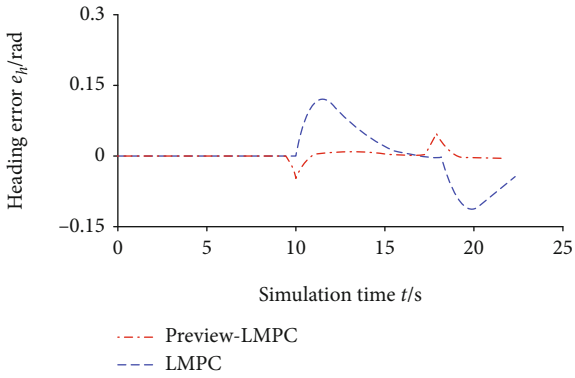

Figure 7: Heading error of the first set of simulations.

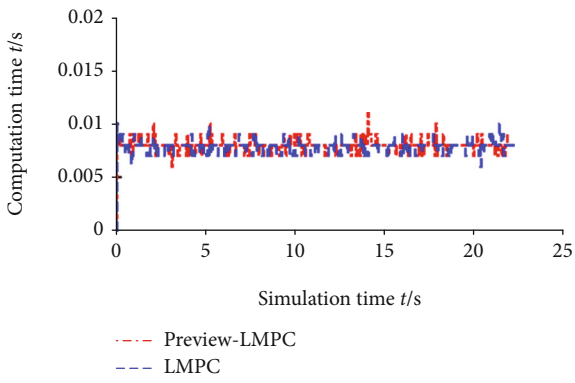

Figure 8: Computation time in each control period of the first set of simulations.

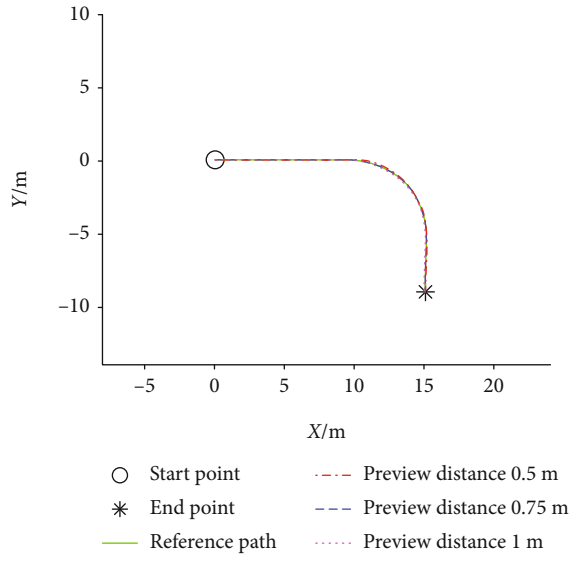

Figure 9: Trajectories of the second set of simulations.

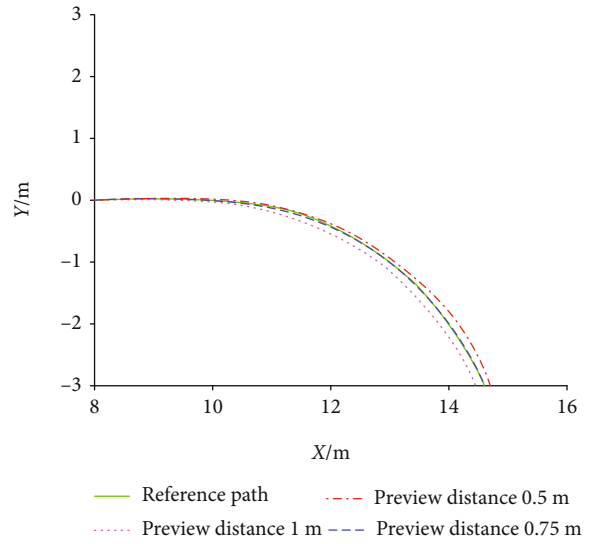

Figure 10: Partially enlarged view trajectories of the second set of simulations.

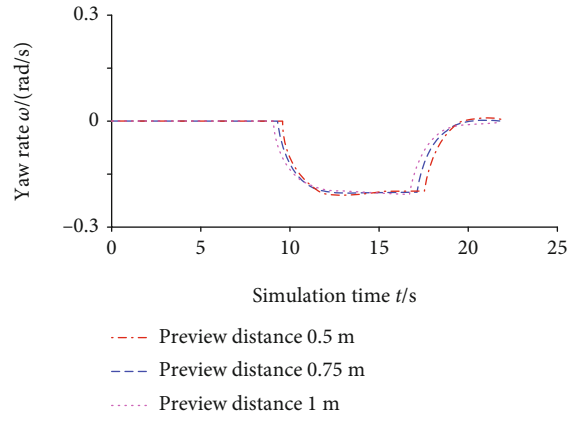

Figure 11: Yaw rate of the second set of simulations.

The weight coefficient matrix is set as

$$\begin{cases} \mathbf{Q} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}. \\ \mathbf{R} = \begin{bmatrix} 1 & \\ & 1 \end{bmatrix} \end{cases} \tag{18}$$

In the simulation, the curvature of the reference path is set to 0.2 m^{-1}. The simulation includes two groups. The former is used to compare the difference between LMPC and preview-LMPC, and the latter is used to test the performance of preview-LMPC when the preview distance is different.

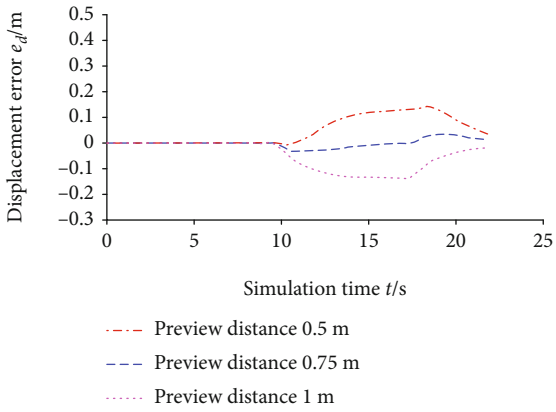

Figure 12: Displacement error of the second set of simulations.

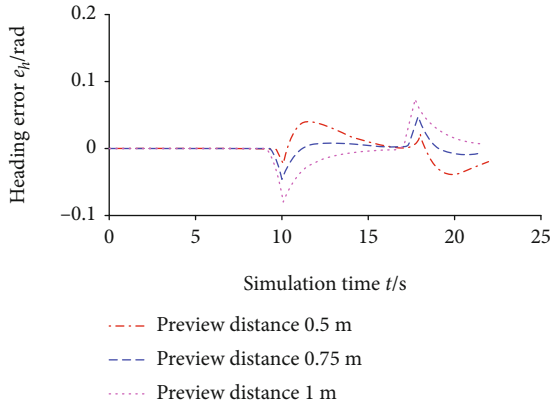

Figure 13: Heading error of the second set of simulations.

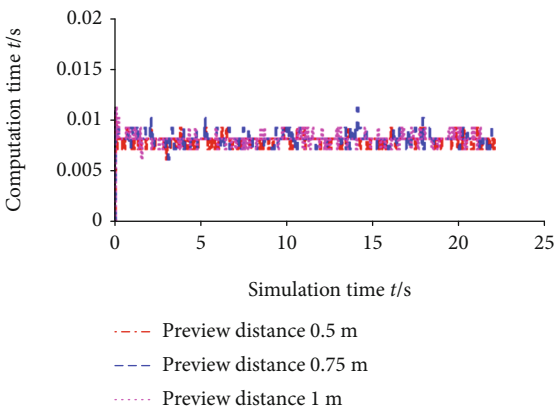

Figure 14: Computation time in each control period of the second set of simulations.

5. First Set of Simulations

Figures 3–7 show the results. In this simulation, the preview distance of preview-LMPC is 0.75 m. Figure 3 shows the travel trajectory of the mobile robot, and Figure 4 is a partially enlarged view of Figure 3. It can be seen that under the control of preview-LMPC, the trajectory of the mobile robot is the closest to the reference path.

Figure 15: Model prototype of tracked mobile robot.

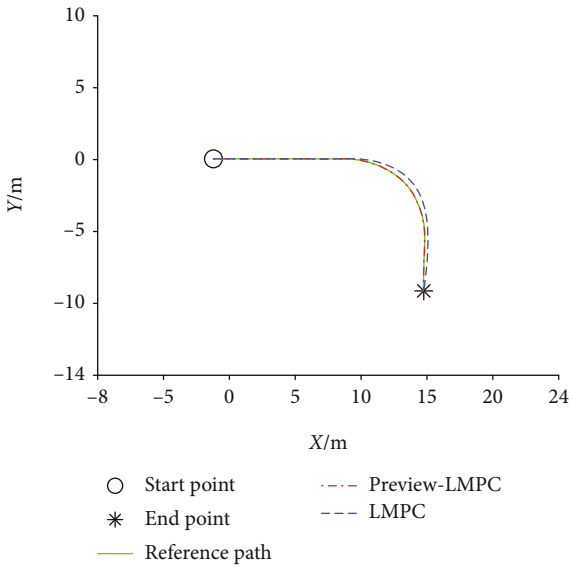

Figure 16: Trajectories of the experiment.

Figure 5 shows the yaw rate. Figures 6 and 7 show the displacement error and heading error. The displacement error is the lateral error between the robot and the reference path, and the heading error is the error between the robot heading and the tangent of the reference path. The maximum absolute value of displacement error of LMPC is 0.3767 m, and the maximum absolute value of heading error is 0.1185 rad. The above variables of preview-LMPC are 0.0333 m and 0.0486 rad, respectively. Compared with LMPC, the maximum absolute value of displacement error is reduced by 91.16%, and the maximum absolute value of heading error is reduced by 58.99%.

Figure 8 shows the calculation time of each controller in each control cycle. The results show that preview control has no effect on the real-time performance of the controller, and the real-time performance of the preview-LMPC controller can satisfy the control requirements.

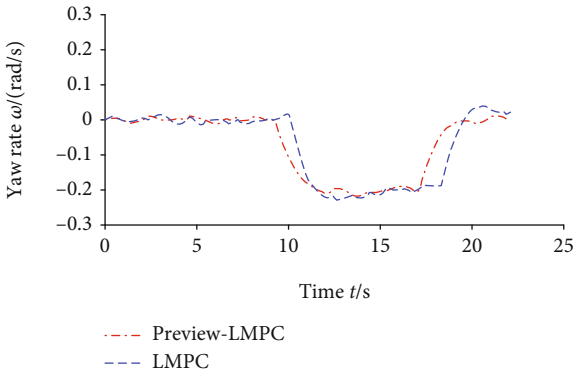

Figure 17: Yaw rate of the experiment.

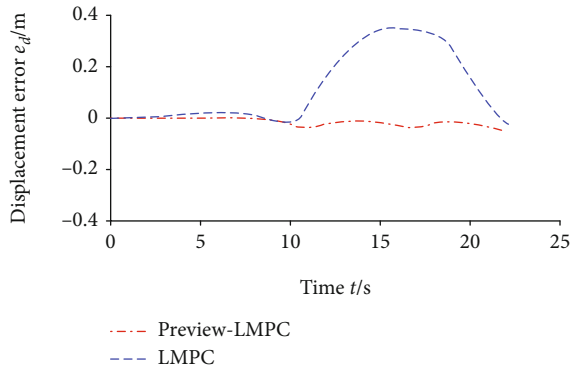

Figure 18: Displacement error of the experiment.

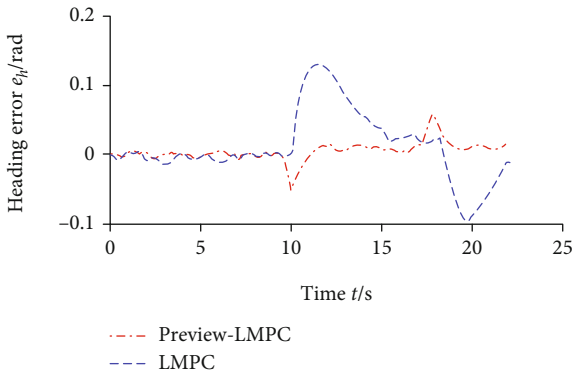

Figure 19: Heading error of the experiment.

6. The Second Set of Simulations

Figures 9–14 show the simulation results of the second group. The simulation tests the performance of the preview-LMPC path tracking controller when the preview distance is 0.5 m, 0.75 m, and 1 m.

Figure 9 shows the trajectory of the mobile robot in the simulation. Figure 10 is a partially enlarged view of Figure 9. The simulation results show that the trajectory under preview-LMPC control is the closest to the reference path. Figure 11 shows the yaw rate.

Figures 12 and 13 show the displacement error and heading error. When the preview distance is 0.75 m, the maximum absolute values of displacement error and heading error are small, while when the preview distance is 0.5 m and 1 m, the maximum absolute values of error are relatively large. Therefore, when the speed and reference path are constant, the preview-LMPC path tracking controller has the optimal preview distance. Figure 14 shows the calculation time of each controller in each control cycle. It can be seen from the figure that the preview distance has no effect on the real-time performance of the controller, and the real-time performance of the preview-LMPC controller can satisfy the control requirements.

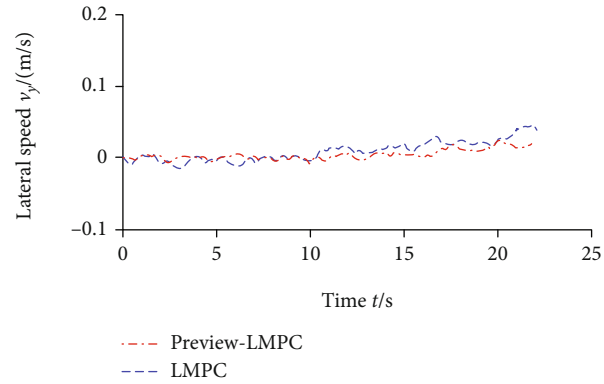

Figure 20: Lateral speed of the experiment.

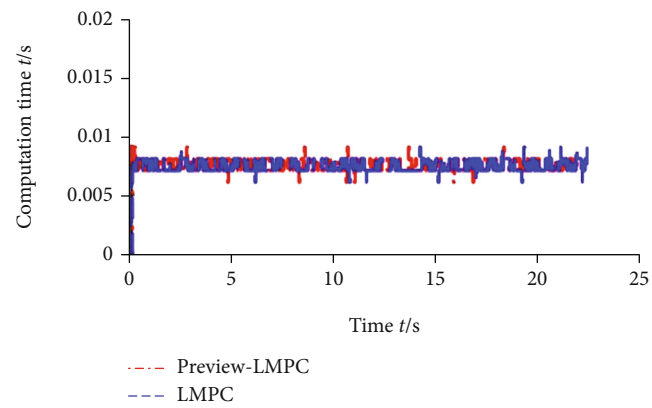

Figure 21: Computation time in each control period of the experiment.

7. Experimental Verification

The experiment uses the same experimental platform as in [29], as shown in Figure 15. The prototype is a tracked mobile robot produced by Guoxing Intelligent. The main controller uses Advantech ARK-3500 industrial computer, the processor type is Intel Core i5-3610ME, and the frequency is 2.7 GHz. The robot is localized through differential GPS (DGPS). In this experiment verification, the parameters of the controller are consistent with the simulation verification, as shown in Table 1.

Due to the limitation of the experimental site, the speed is set at 1 m/s, and the reference path curvature is set at 0.2 m^{-1}. The preview distance of preview-LMPC is set to 0.75 m. Figures 16–21 show the simulation results.

Figure 16 shows the trajectory of the tracked mobile robot. The results show that the robot under the control of preview-LMPC is the closest to the reference path. Figure 17 shows the yaw rate.

Figures 18 and 19 show the displacement error and heading error. The maximum absolute value of displacement error of LMPC is 0.3471 m, and the maximum absolute value of heading error is 0.1317 rad. The above state variables of preview-LMPC are 0.0477 m and 0.0636 rad, respectively. Therefore, in the experimental results, compared with LMPC, the preview-LMPC reduces the maximum absolute value of displacement error by 86.26% and reduces the maximum absolute value of heading error by 51.71%.

Figure 20 shows the lateral speed. Due to the low speed and centrifugal acceleration, the lateral speed of tracked mobile robot controlled by each controller is small, that is, no severe sideslip occurs. Figure 21 shows the calculation time of the controller in each control cycle. It can be seen from the figure that the preview distance has no effect on the real-time performance of the controller, and the real-time performance of the preview-LMPC controller can satisfy the control requirements.

8. Conclusions

(1) According to the simulation and experimental results, the introduction of preview control can significantly improve the accuracy of the LMPC path tracking controller. In the simulation, the preview-LMPC can reduce the maximum absolute value of displacement error and heading error by 91.16% and 58.99% compared with LMPC. In the experiment, it can be reduced by 86.26% and 51.71%, respectively

(2) According to the second group of simulations, when the speed and the reference path are constant, the preview-LMPC based path tracking controller has the optimal preview distance. Simulation and experiment show that the accuracy of path tracking control is the best when the preview distance is 0.75 m. In addition, the different preview distance has no effect on the real-time performance of the path tracking controller, and the real-time performance of the preview-LMPC controller can satisfy the control requirements

Data Availability

The simulation and experimental parameters data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the National Key Research and Development Program of China (No. 2019YFC0605300), Guangdong Basic and Applied Basic Research Foundation (No. 2019A1515111015), Fundamental Research Funds for the Central Universities (FRF-MP-20-07), and Science and Technology Plan Project of China Nonferrous Metal Mining Group (2018KJJH01).

References

[1] A. Rupp and M. Stolz, "Survey on control schemes for automated driving on highways," in *Automated Driving*, D. Watzenig and M. Horn, Eds., pp. 43–69, Springer, Cham, 2017.

[2] P. Falcone, M. Tufo, F. Borrelli, J. Asgari, and H. E. Tseng, "A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems," in *2007 46th IEEE Conference on Decision and Control*, pp. 2980–2985, New Orleans, LA, USA, 2007.

[3] H. Guo, D. Cao, H. Chen, Z. Sun, and Y. Hu, "Model predictive path following control for autonomous cars considering a measurable disturbance: implementation, testing, and verification," *Mechanical Systems and Signal Processing*, vol. 118, pp. 41–60, 2019.

[4] G. Bai, L. Liu, Y. Meng, W. Luo, Q. Gu, and J. Wang, "Path tracking of wheeled mobile robots based on dynamic prediction model," *IEEE Access*, vol. 7, pp. 39690–39701, 2019.

[5] G. Bai, L. Liu, Y. Meng, W. Luo, Q. Gu, and B. Ma, "Path tracking of mining vehicles based on nonlinear model predictive control," *Applied Sciences*, vol. 9, no. 7, p. 1372, 2019.

[6] M. M. G. Plessen and A. Bemporad, "Reference trajectory planning under constraints and path tracking using linear time-varying model predictive control for agricultural machines," *Biosystems Engineering*, vol. 153, pp. 28–41, 2017.

[7] W. Zhang, W. Bai, Z. Lü, Z. Liu, and C. Huang, "Linear time-varying model predictive controller improving precision of navigation path automatic tracking for agricultural vehicle," *Transactions of the CSAE*, vol. 33, no. 13, pp. 104–111, 2017.

[8] A. Eskandarpour and I. Sharf, "A constrained error-based MPC for path following of quadrotor with stability analysis," *Nonlinear Dynamics*, vol. 99, no. 2, pp. 899–918, 2020.

[9] M. Ataei, A. Khajepour, and S. Jeon, "Model predictive control for integrated lateral stability, traction/braking control, and rollover prevention of electric vehicles," *Vehicle System Dynamics*, vol. 58, no. 1, pp. 49–73, 2020.

[10] 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, 2019.

[11] F. Dou, Y. Huang, L. Liu, H. Wang, Y. Meng, and L. Zhao, "Path planning and tracking for autonomous mining articulated vehicles," *International Journal of Heavy Vehicle Systems*, vol. 26, no. 3-4, pp. 315–333, 2019.

[12] B. Zhang, C. Zong, G. Chen, and B. Zhang, "Electrical vehicle path tracking based model predictive control with a laguerre function and exponential weight," *IEEE Access*, vol. 7, pp. 17082–17097, 2019.

[13] C. Sun, X. Zhang, Q. Zhou, and Y. Tian, "A model predictive controller with switched tracking error for autonomous vehicle path tracking," *IEEE Access*, vol. 7, pp. 53103–53114, 2019.

[14] Y. Bo, H. Guo, C. Shen, and J. Liu, "MPC-based path tracking controller design for intelligent driving vehicle," in *2019 Chinese Control Conference (CCC)*, pp. 3006–3011, Guangzhou, China, 2019.

[15] H. Xu and J. Zhu, "Interval trajectory tracking for AGV based on MPC," in *2019 Chinese Control Conference (CCC).*, pp. 2835–2839, Guangzhou, China, 2019.

[16] H. Kanchwala, Í. B. Viana, M. Ceccoti, and N. Aouf, "Model predictive tracking controller for a high fidelity vehicle dynamics model," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 1496–1503, Auckland, New Zealand, 2019.

[17] M. Li, H. Cao, X. Song, Y. Huang, J. Wang, and Z. Huang, "Shared control driver assistance system based on driving intention and situation assessment," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4982–4994, 2018.

[18] Y. Meng, Y. Wang, Q. Gu, and G. Bai, "LQR-GA path tracking control of articulated vehicle based on predictive information," *Nongye Jixie Xuebao/Transactions of the Chinese Society for Agricultural Machinery*, vol. 49, no. 6, pp. 375–384, 2018.

[19] S. Xu and H. Peng, "Design, analysis, and experiments of preview path tracking control for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 48–58, 2019.

[20] Q. Diao, Y. Zhang, and L. Zhu, "A lateral fuzzy controller of intelligent vehicles with double look-ahead points for large curvature roads," *China Mechanical Engineering*, vol. 30, no. 12, pp. 1445–1452, 2019.

[21] D. Meola, "A comparison between LTV-MPC and LQR yaw rate-side slip controller," *IFAC Proceedings Volumes*, vol. 42, no. 26, pp. 154–159, 2009.

[22] F. Yakub and Y. Mori, "Comparative study of autonomous path-following vehicle control via model predictive control and linear quadratic control," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 229, no. 12, pp. 1695–1714, 2015.

[23] T. Nayl, G. Nikolakopoulos, and T. Gustafsson, "A full error dynamics switching modeling and control scheme for an articulated vehicle," *International Journal of Control, Automation and Systems*, vol. 13, no. 5, pp. 1221–1232, 2015.

[24] Y. Meng, X. Gan, and G. Bai, "Path following control of underground mining articulated vehicle based on the preview control method," *Chinese Journal of Engineering*, vol. 41, no. 5, p. 671, 2019.

[25] G. Bai, Y. Meng, L. Liu, W. Luo, Q. Gu, and K. Li, "A new path tracking method based on multilayer model predictive control," *Applied Sciences*, vol. 9, no. 13, article 2649, 2019.

[26] J. Gong, W. Xu, Y. Jiang, K. Liu, H. Guo, and Y. Sun, "Multi-constrained model predictive control for autonomous ground vehicle trajectory tracking," *Journal of Beijing Institute of Technology (English Edition)*, vol. 24, no. 4, pp. 441–443, 2015.

[27] J. Gong, J. Yan, and X. Wei, *Model predictive control for self-driving vehicles*, Beijing Institute of Technology Press, Beijing, 2014.

[28] Y. Cai, J. Li, X. Sun et al., "Research on hybrid control strategy for intelligent vehicle path tracking," *China Mechanical Engineering*, vol. 31, no. 3, pp. 289–298, 2020.

[29] G. Bai, L. Liu, Y. Meng, S. Liu, L. Liu, and W. Luo, "Real-time path tracking of mobile robot based on nonlinear model predictive control," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 51, no. 9, pp. 47–52, 2020.