文章编号：0253-374X(2024)S1-0158-07 **DOI**：10.11908/j.issn.0253-374x.24723

基于自适应路径预览的非线性模型预测控制路径跟踪

李俊霆，陳志鏗
（台北理工大学 车辆工程系，台北，中国）

摘要：提出了一种用于自动驾驶汽车路径跟踪的非线性模型预测控制器(NMPC)和自适应调整预览距离的算法。预测模型包括车辆动力学、路径跟随动力学和系统输入动力学。单轨车辆模型考虑了车辆的横向和纵向耦合动力学以及非线性轮胎力。基于曲线坐标推导了跟踪误差动力学。成本函数被设计为最小化路径跟踪误差和控制努力，同时考虑诸如致动器边界和轮胎抓地限制之类的约束。一种利用最佳预览距离矢量来查询相应参考曲率和参考速度的算法。基于车辆速度、航向误差和路径曲率自适应地调整预览路径的长度。在具有自主赛车场景的仿真环境中验证了控制器的性能，结果表明，车辆准确地遵循高度动态的路径，跟踪误差较小。所提出的模型与算法可以预先估计最大预览距离，并指导NMPC的预测范围的选择。

关键词：路径跟踪；曲线坐标；非线性模型预测控制
中图分类号：U461.91 **文献标志码**：A

Path-Following Based on Nonlinear Model Predictive Control with Adaptive Path Preview

Jun-Ting LI, Chih-Keng CHEN
(Department of Vehicle Engineering, National Taipei University of Technology, Taipei, China)

Abstract: This paper presents a Nonlinear Model Predictive Controller (NMPC) for the path following of autonomous vehicles and an algorithm to adaptively adjust the preview distance. The prediction model includes vehicle dynamics, path following dynamics, and system input dynamics. The single-track vehicle model considers the vehicle's coupled lateral and longitudinal dynamics, as well as nonlinear tire forces. The tracking error dynamics are derived based on the curvilinear coordinates. The cost function is designed to minimize path tracking errors and control effort while considering constraints such as actuator bounds and tire grip limits. An algorithm that utilizes the optimal preview distance vector to query the corresponding reference curvature and reference speed. The length of the preview path is adaptively adjusted based on the vehicle speed, heading error, and path curvature. We validate the controller performance in a simulation environment with the autonomous racing scenario. The simulation results show that the vehicle accurately follows the highly dynamic path with small tracking errors. The maximum preview distance can be prior estimated and guidance the selection of the prediction horizon for NMPC.

Keywords: path following; curvilinear coordinates; nonlinear model predictive control

1 Introduction

Model predict control can predict the future behavior of the system, consider the preview trajectory, and utilize these information to optimize the current control commands. In the model prediction path following controller, the information of the path ahead from vehicle is required. Path preview distance significantly affects the path following performance and several papers have investigated this topic. Zhang et al. [1] compares the path-following errors of different predict distances under double lane change maneuver. Wang et al. [2] proposes a variable prediction horizon (VPH), using a particle swarm optimisation (PSO) algorithm to

收稿日期：2023-12-28
基金项目："National Science and Technology Council"(NSTC 111-2221-E-027-088)
第一作者：李俊霆(1999—)，男，博士研究生，主要研究方向为智能汽车控制。E-mail：t111448002@ntut.org.tw
通信作者：陳志鏗(1958—)，男，教授，工学博士，主要研究方向为智能汽车控制。E-mail：ckchen@ntut.edu.tw

optimise the prediction horizon based based on comprehensive performance indexes. Xu and Peng[3] compares the effects of different predict distances on the path-following error with MPC and preview control in both simulations and experiments. The experimental results demonstrate that the preview control achieves smoother steering and better ride comfort compared to the feedback control.

This paper implements vehicle path-following control with a non-linear model prediction controller. By considering the nonlinearity of the vehicle dynamics and path following dy-namics, we can accurately predict the future vehicle traveling distance and use the corresponding path curvature and reference speed as the reference. These approaches can improve the path following performance of a vehicle travelling on a path with arbitrary curvature at different speeds.

2 VEHICLE MODELING

2.1 Nonlinar Single-Track Vehicle Model

The layout of the single-track model is shown in Fig. 1. To predict the planar vehicle motion about O_B, we use the angular velocity $\dot{\psi}$ (yaw rate), longitudinal velocity V_x and lateral velocity V_y to formulate dynamic equations:

$$\ddot{\psi}=\frac{(\sum l_f[F_{yf}\cos(\delta)+F_{xf}\sin(\delta)]-l_rF_{yr}}{I_z} \quad (1a)$$

$$\dot{V}_x=\frac{\sum -F_{yf}\sin(\delta)+F_{xf}\cos(\delta)+F_{xr}-F_d}{m}+V_y\dot{\psi} \quad (1b)$$

$$\dot{V}_y=\frac{\sum F_{yf}\cos(\delta)+F_{xf}\sin(\delta)+F_{yr}}{m}+V_x\dot{\psi} \quad (1c)$$

where: δ is the front wheel steer angle; l is the wheelbase, l_f and l_r are the distance from CG to the front and rear axles; the lumped tire force F_{ij} with the subscript $\{x, y\}$ denotes the longitudinal or lateral direction and $\{f, r\}$ denotes the front or rear wheel; drag force $F_d=C_dV_x^2$ is the compact aerodynamic coefficient times velocity squared; m is the vehicle mass; I_z is the moment of inertia.

2.2 Lateral Tire Force

The Simplified Magic Formula[4] is used to capture the nonlinear behavior of lateral tire force, the model is a function of the sideslip angle α, the vertical force F_z of a single wheel, and the road friction coefficient μ:

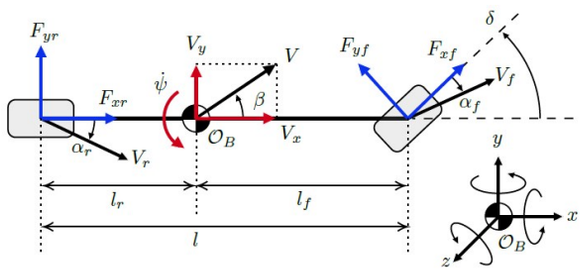

Fig.1 Schematic diagram of the single-track vehicle model. The state variables are in red and the tire forces are in blue, the arrow indicates the positive direction of the corresponding variable

$$F_{y0}(\alpha, F_z, \mu)=\mu D\sin(C\tan^{-1}(B\alpha)) \quad (2a)$$

$$D=d_1F_z+d_2 \quad (2b)$$

where B, C and D are fitting coefficients. The peak factor D is defined by the first-order function of F_z. The lumped sideslip angles at the front and rear axles are described by:

$$\alpha_f=\tan^{-1}\left(\frac{V_y+l_f\dot{\psi}}{V_x}\right)-\delta,\ \alpha_r=\tan^{-1}\left(\frac{V_y+l_r\dot{\psi}}{V_x}\right) \quad (3)$$

Then the combined tire lateral forces F_{yf} and F_{yr} in (4) are available to substitute into single-track model (1).

$$F_{yi}=2F_{y0}\left(\alpha_i, \frac{F_{zi}}{2}, \mu\right),\ i\in\{f, r\} \quad (4)$$

2.3 Longitudinal Tire Force

The longitudinal forces F_{xf} and F_{xr} are composed of rear wheel traction force F_t and all-wheel braking force F_b as following equations:

$$F_{xf}=k_bF_b,\ F_{xr}=F_t+(1-k_b)F_b \quad (5)$$

where: k_b is the braking coefficient that determines the force distribution between the front axle and rear axle, we set $k_b=F_{zf}/(mg)$ to distribute the braking force according to the ratio of axle load. The system inputs are δ, F_t, and F_b, but the large difference in magnitude between these variables may increase numerical instability for an NMPC solver, therefore, the normalized acceleration at and ab are introduced:

$$a_t=F_t/m,\ a_b=F_b/m \quad (6)$$

2.4 Constraints

During racing, we must ensure that the combined tire force is contained within the friction ellipse to avoid large tire slip and loss of grip. Thus

the quadratic non-linear constraints are applied:

$$\frac{F_{xi}^2+F_{yi}^2}{(\mu F_{zi})^2}\leqslant 1,\ i\in\{f,r\} \tag{7}$$

There are upper bounds and lower bounds on the system inputs due to physical limits, these bounds are expressed by inequality constraints as:

$$\begin{gathered}\delta_{\min}\leqslant\delta\leqslant\delta_{\max}\\ 0\leqslant a_t\leqslant\frac{T_{r,\max}}{r_w m},\ \frac{T_{b,\max}}{r_w m}\leqslant a_b\leqslant 0\end{gathered} \tag{8}$$

where: $T_{r,\max}$ is the maximum total traction torque of rear axle; $T_{b,\max}$ is the maximum braking torque of each axle; r_w is the effective rolling radius of wheel. In order to achieve optimal actuation efficiency, an equality constraint is imposed,

$$a_t b_b=0 \tag{9}$$

This equation ensures the traction and braking commands are orthogonal, i. e., they are not active at the same time.

3 Path Modeling

The reference path is fitted by a cubic spline function to create a parametric representation. A curvilinear coordinate system is used to describe the relationship between the vehicle position and the reference waypoints (See Fig. 2).

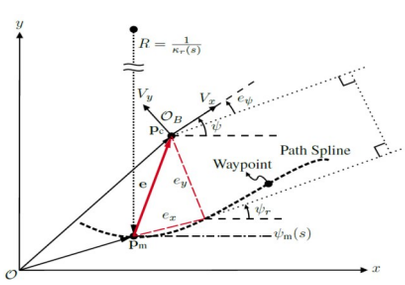

Fig.2 Single-track vehicle model in a curvilinear coordinates

3.1 Parametric Path

The discrete path data set $[\boldsymbol{X},\ \boldsymbol{Y}]$ is composed by path coordinates $\boldsymbol{X}=[x_0,x_1,\cdots,x_n]^{\mathrm{T}}$ and $\boldsymbol{Y}=[y_0,y_1,\cdots,y_n]^{\mathrm{T}}$. Then a spline function $X(s)$ with n equations is used to fit X:

$$\begin{gathered}X(s)=a_n(s-s_{n-1})^3+b_n(s-s_{n-1})^2+c_n(s-s_{n-1}),\\ k=1,2,\cdots,n,\ s\in[s_{(n-1)},s_n]\end{gathered} \tag{10}$$

where: s is the cumulative arc length starting from $s_0=0$:

$$\begin{gathered}s_k=\sum_{i=1}^{k=n}\sqrt{(x_i-x_{i-1})^2+(y_i-y_{i-1})^2},\\ k=1,2,\cdots,n\end{gathered} \tag{11}$$

The higher-order derivative of the spline function with respect to the progressive variable s is easily obtained by:

$$x'(s)=3a_k(s-s_{k-1})^2+2b_k(s-s_{k-1})+c_k \tag{12}$$

$$x''(s)=6a_k(s-s_{k-1})+2b_k \tag{13}$$

$$k=1,2,\cdots,n,\ s\in[s_{k-1},s_k]$$

Following the above procedure, $y(s)$, $y'(s)$ and $y''(s)$ are also derived based on y. The reference heading angle ψ_r and the reference curvature κ_r are calculated by:

$$\psi_r=\arctan 2(y',x') \tag{14}$$

$$\kappa_r=\frac{x'y''-x''y'}{(x'^2+y'^2)^{3/2}} \tag{15}$$

where the arctan 2 is the 2-argument arctangent function.

3.2 Tracking Error for Discrete Path

The vehicle CG position is $\boldsymbol{p}_c=[x_c,y_c]^{\mathrm{T}}$, and the closest waypoint to $\boldsymbol{p}_c$ is defined as "matching point", $\boldsymbol{p}_m=[x_m,y_m]^{\mathrm{T}}$, where m is the index of mathtcing point. Then the tracking error vector in global coordinate is defined as:

$$\boldsymbol{e}=\boldsymbol{p}_c-\boldsymbol{p}_m \tag{16}$$

In order to obtain the longitudinal error e_x and the lateral error e_y, the error vector is projected onto the curvilinear coordinate by rotation matrix:

$$\begin{bmatrix}e_x\\ e_y\end{bmatrix}=\begin{bmatrix}\cos\psi_m & -\sin\psi_m\\ \sin\psi_m & \cos\psi_m\end{bmatrix}\boldsymbol{e} \tag{17}$$

where: ψ_m is the reference heading angle at $\boldsymbol{p}_m$, the arc length error es can be approximated as longitudinal error e_x in general. Under the assumption that the reference curvature is the same at the matching point and projection point, then the reference heading angle is given by:

$$\psi_r=\psi_m+\kappa_r e_s \tag{18}$$

Consider the vehicle sideslip angle β is regulated in a small value during racing, the course angle can be approximated to heading angle, then the heading error is defined as:

$$e_\psi=\psi-\psi_r \tag{19}$$

3.3 Tracking Error Dynamics

To predict future path tracking error, the dynamics of the heading error, lateral error, and vehicle traveling distance are given by:

$$\dot{e}_\psi = \dot{\psi} - \dot{\psi}_r = \dot{\psi} - \kappa_r \dot{s} \tag{20a}$$

$$\dot{e}_y = V_y \cos e_\psi + V_x \sin e_\psi \tag{20b}$$

$$\dot{s} = \frac{V_x \cos e_\psi - V_y \sin e_\psi}{1 - \kappa_r e_y} \tag{20c}$$

4 Nonlinear Model Predictive Controller

In this section, the path following problem is transformed to nonlinear programming (NLP) problem.

4.1 Control Architecture

The overall control structure of path following is shown in Fig. 3. The tracking error evaluation module outputs the path tracking error e_ψ and ed according to the current position of the vehicle. The preview module looks forward over the preview path and sends the corresponding reference curvature and reference speed to the NMPC. The lower controller distributes the desired acceleration commands a_t, a_b as the rear driving torques or four-wheel braking torques.

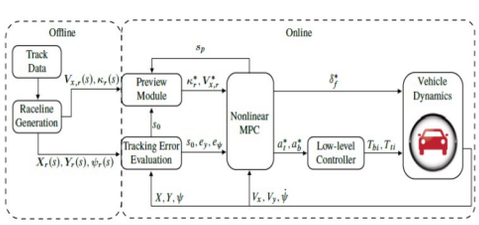

Fig.3 Control architecture of NMPC path-following

4.2 Prediction Model

Consider the vehicle dynamics (1), path following dynamics (20), and extended system input dynamics $\boldsymbol{u} = \left[\dot{\delta}, \dot{a}_t, \dot{a}_b\right]^{\mathrm{T}}$, the prediction model is formulated as (21). Given a prediction horizon N_p, $\boldsymbol{p} = \left[\kappa_r^1, \cdots, \kappa_r^{N_p}\right]^{\mathrm{T}}$ is the parameter trajectory composed by the reference curvature at each prediction stage. The first six equations in (21) can predict future vehicle states, path tracking errors, and driving distance. In the last three equations we assign the slew rate of system inputs as extended states,

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \ddot{\psi} \\ \dot{V}_x \\ \dot{V}_y \\ \dot{e}_\psi \\ \dot{e}_y \\ \dot{s} \\ \dot{\delta} \\ \dot{a}_t \\ \dot{a}_b \end{bmatrix} = \begin{bmatrix} \dfrac{\sum M_z}{I_z} \\ \dfrac{\sum F_x}{m} - V_y \dot{\psi} \\ \dfrac{\sum F_y}{m} - V_x \dot{\psi} \\ \dot{\psi} - \kappa_r \dot{s} \\ V_y \cos e_\psi + V_x \sin e_\psi \\ V_y \cos e_\psi + V_x \sin e_\psi \\ \dfrac{V_x \cos e_\psi - V_y \sin e_\psi}{1 - \kappa_r e_y} \\ \dot{a}_t \\ \dot{a}_b \end{bmatrix} := \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{p}) \tag{21}$$

similar to Ref. [5]. This approach allows to reduce drastic changes of the inputs to obtain smooth command signals.

4.3 Discretization

The multi-step Euler method is used to discrete model in a simple but also precise way:

$$\begin{aligned} &\begin{cases} k_1 = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{p}) \\ \quad \cdots \\ k_n = \boldsymbol{f}(\boldsymbol{x}_k + h k_{n-1}, \boldsymbol{u}_k, \boldsymbol{p}), \end{cases} \\ &\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + h \sum_{i=1}^{n} k_i \\ &:= \boldsymbol{f}_d(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{p}), \quad k = 0, 1, \cdots, N_p - 1 \end{aligned} \tag{22}$$

where $h = t_s/n$ is the step size of Euler method, with the sampling time t_s and the discrete step n. The model is integrated with smaller steps to increase the model prediction accuracy. In this study, h=0.02 s and t_s=0.1 s.

4.4 Adaptive Preview Distance

The optimal state vector at the k^{th} prediction step is denoted by

$$\boldsymbol{x}_k^* = \left[V_{x,k}^* \;\; V_{y,k}^* \;\; \dot{\psi}^* \;\; s_k^* \;\; c_{y,k}^* \;\; c_{\psi,k}^* \;\; \delta_{f,k}^* \;\; a_{t,k}^* \;\; a_{b,k}^*\right]^{\mathrm{T}} \tag{23}$$

The predicted vehicle traveling distances over prediction horizon at the current time step are given by

$$\left[s_0 \;\; s_1^* \;\; \cdots \;\; s_{N_p}^*\right] \tag{24}$$

where the first element s_0 is the current vehicle traveling distance. To provide the preview vector for the next time step, we remove s_0 and add a correction term that predicts one more step by multiplying sampling time with last predicted velocity, yielding preview distance vector,

$$\boldsymbol{s}_p = \left[s_1^* \;\; \cdots \;\; s_{N_p}^* \;\; s_{N_p}^* + t_s V_{x,N_p}^*\right] \tag{25}$$

with this approach, the preview distance can be

adaptively adjusted based on the predicted vehicle longitudinal velocity in each prediction step. The required preview distance can be prior estimated by

$$s_p = s^*_{N_p} + t_s V^*_{x,N_p} - s_0 \tag{26}$$

The preview distance vector is used to query the corresponding reference curvature and velocity using linear interpolation,

$$\boldsymbol{V}^*_{x,r} = \boldsymbol{V}_{x,r}(s_p) \tag{27}$$

$$\boldsymbol{\kappa}^*_r = \boldsymbol{\kappa}_r(s_p) \tag{28}$$

4.5 Reference Outputs

To stabilize the lateral dynamics, track the reference velocity, and minimize path tracking error, the output vector of NMPC is $\boldsymbol{y} = [V_x, V_y, e_y]$, and the reference output states at each stage over the prediction horizon N_p are given by

$$\forall k = 1, \cdots, N_p,$$

$$\boldsymbol{y}^{\text{ref}}_k = \begin{bmatrix} V^{\text{ref}}_{x,k} \\ V^{\text{ref}}_{y,k} \\ e^{\text{ref}}_{y,k} \end{bmatrix} = \begin{bmatrix} V_{x,r}(s_{p,k}) \\ \dfrac{l_r}{l}\delta^*_{f,k} V^*_{x,k} \\ 0 \end{bmatrix} \tag{29}$$

where $s_{p,k}$ denotes the kth element in (25). The reference lateral velocity is derived from the steady-state kinematic sideslip angle, which can be found in Ref. [6] as $\beta_s = (l_r/l)\delta_f \approx V_y/V_x$.

4.6 Cost Function

Compose tracking and control costs over the prediction horizon, the cost function J to be minimized is defined as:

$$J = \sum_{k=1}^{N_p} \frac{1}{2} \left\| \boldsymbol{S}_y^{-1}(y_k - y^{\text{ref}}_k) \right\|^2_{\boldsymbol{Q}} + \sum_{k=0}^{N_p-1} \frac{1}{2} \left\| \boldsymbol{S}_u^{-1} \boldsymbol{u}_k \right\|^2_{\boldsymbol{R}} \tag{30}$$

where $\boldsymbol{S}_y$ and $\boldsymbol{S}_u$ are square matrices with diagonal scaling factors. Set these factors to the maximum acceptable values of the variables for normalisation. $\boldsymbol{Q}$ is a positive weighting matrix to penalize the difference between reference states and actual system states, $\boldsymbol{R}$ penalizes the control effort to obtain the smooth input trajectory.

Finally we collect cost function, prediction model, and constraints to formulate path-following optimization problem:

$$\min_{\substack{x_0,\cdots,x_{N_p} \\ u_0,\cdots,u_{N_p-1}}} \quad J \tag{31a}$$

$$\text{s. t. } x_0 = \hat{\boldsymbol{x}}(t) \tag{31b}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_d(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{p}) \tag{31c}$$

$$\text{constraints}(7), (8), (9) \tag{31d}$$

where $\hat{\boldsymbol{x}}(t)$ is the estimated or measured states at the current time. All the programs are deployed on a desktop computer with Intel i5-12500 @3.0 GHz processor, and we show that the problem of each step can be solved within milliseconds.

5 Simulation Results

5.1 Simulation Results

The optimal reference raceline is generated by the minimum curvature algorithm and the corresponding reference speed is generated by the quasi-steady-state lap time simulation tool all these programs are open source and available in Ref. [7]. The reference curvature and heading angle of the race line are obtained by following the process in Section III-A.

5.2 Vehicle Configuration

The CarSim built-in B-class sports car was used as the test vehicle in this study, which is a neutral-steering vehicle. The main vehicle parameters are: m =1 209 kg; I_z=1 020 kg·m^2; l_f=1.165 m; l_r= 1.165 m; h_c=0.35 m.

5.3 Path tracking performance

Fig. 4 presents the overall path tracking response. Since the lateral tracking error is very small we omit the reference path in the plot for clarity. The path displays the position of the vehicle's CG along with its longitudinal velocity indicated by a color gradient. The corresponding values for colors are displayed in a colorbar located on the right side. The average longitudinal velocity of the vehicle was 87.22 km/h, and the lap time was 92.07 s.

Fig. 5 shows the path tracking errors and primary vehicle states. The root mean square (RMS) and maximum lateral error are 0.430 m and 0.094 m, the RMS and maximum heading error are 1.09° and 1.92°. The velocity profile shows that the vehicle decelerated properly before the apex, accelerated during the exit, and maintained high speeds on straight lines. During the turn, the yaw rate was used to track the curvature profile thus they have similar trends. The vehicle's sideslip angle is kept below 2°

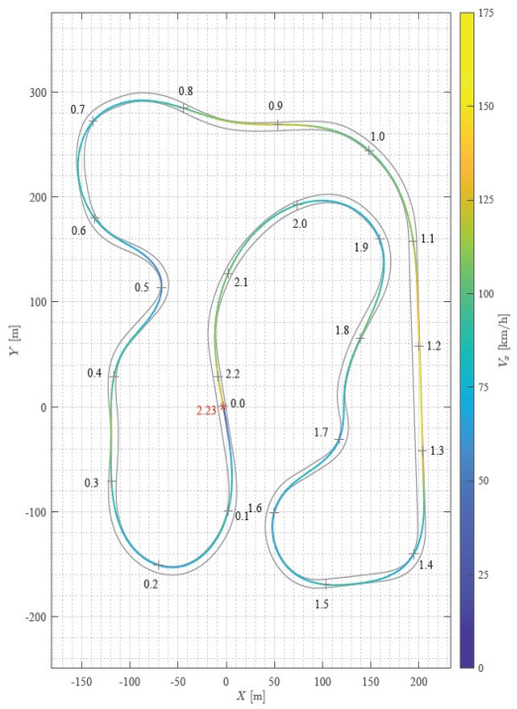

Fig.4　Path tracking result in Cartesian coordinates

during racing, which means that the vehicle maintains excellent lateral stability in the high-speed turns.

Fig. 6 shows the maximum preview distance during racing. With the preview distance vector that takes into account the path following dynamics, the length of preview path can be adaptively adjusted based on the vehicle speed, heading error, and path curvature. This approach allows us to evaluate the required preview distance in advance. As the prediction time of the NMPC gets longer, the preview distance will increase. We need to consider the practical application conditions (e. g., limitations of sensing technology, visibility of the path ahead) to set a prediction time with feasible preview distance.

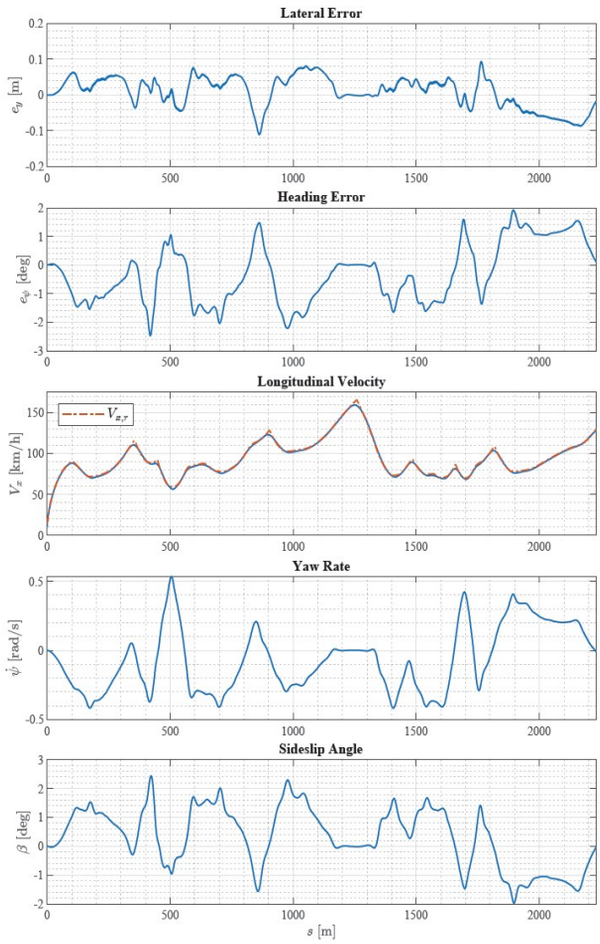

Fig.5　Path tracking errors and vehicle states

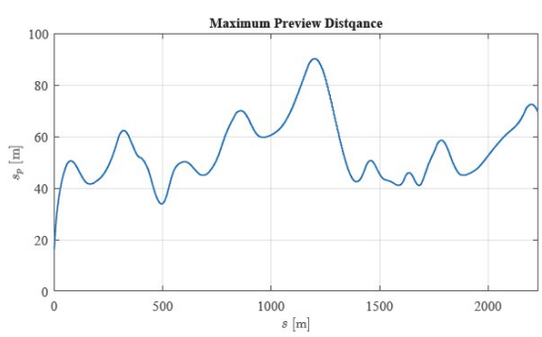

Fig.6　Maximum preview distance

6　Conclusions

This paper presents a NMPC for the path following of autonomous vehicles and an algorithm to adaptively adjust the preview distance. The proposed controller scheme coordinates the vehicle's lateral and longitudinal dynamics to follow the racing line at high speed. By combining path dynamics and vehicle dynamics to build a prediction model for NMPC, path following performance and driving stability are significantly improved. The simulation results show that the vehicle accurately follows the highly dynamic path with small tracking errors.

References:

[1] ZHANG C Y, CHU D F, LIU S D, *et al*. Trajectory planning and tracking for autonomous vehicle based on state lattice and model predictive control [J]. IEEE Intelligent Transportation Systems Magazine, 2019, 11(2): 29.

[2] WANG H R, WANG Q D, CHEN W W, *et al*. Path tracking based on model predictive control with variable predictive

horizon[J]. Transactions of the Institute of Measurement and Control, 2021, 43(12): 2676. https://doi.org/10.1177/01423312211003809.

[3] XU S B, PENG H. Design, analysis, and experiments of preview path tracking control for autonomous vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 21(1): 48.

[4] METZLER M, TAVERNINI D, GRUBER P, *et al*. On prediction model fidelity in explicit nonlinear model predictive vehicle stability control[J]. IEEE Transactions on Control Systems Technology, 2020, 29(5): 1964.

[5] LOT R, BIRAL F. A curvilinear abscissa approach for the lap time optimization of racing vehicles[J]. IFAC Proceedings Volumes (19th IFAC World Congress), 2014, 47(3): 7559. https://www.sciencedirect.com/science/article/pii/S1474667016428041.

[6] Abe M. Vehicle handling dynamics: theory and application[M]. Oxford, UK: Butterworth-Heinemann, 2015.

[7] Heilmeier A. Quasi-steady-state lap time simulation[DB/OL]. Github Repository, https://github.com/TUMFTM/laptimesimulation.