Obstacle avoidance of autonomous vehicles based on model predictive control

J-M Park[1], **D-W Kim**[1], **Y-S Yoon**[2], **H J Kim**[1]*, and **K-S Yi**[1]

[1]Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul, Republic of Korea

[2]Powertrain, Gasoline System, Continental, Seoul, Republic of Korea

The manuscript was received on 3 February 2009 and was accepted after revision for publication on 18 June 2009.

DOI: 10.1243/09544070JAUTO1149

Abstract: This paper presents an obstacle avoidance scheme for autonomous vehicles as an active safety procedure in unknown environments. Safe trajectories are generated using the non-linear model predictive framework, in which the simplified dynamics of the vehicle are used to predict the state of the vehicle over the look-ahead horizon. To compensate for the slight dissimilarity between the simplified model and the actual vehicle, a separate controller is designed to track the generated trajectory. The longitudinal dynamics of the vehicle are controlled using the inverse dynamics of the vehicle powertrain model, and the lateral dynamics are controlled using a linear quadratic regulator. In the non-linear model predictive framework, to obtain safe trajectories, local obstacle information is incorporated into the performance index using a parallax-based method. Simulation results on a full non-linear vehicle model show that the proposed combination of model-predictive-control-based trajectory generation and tracking controller gives satisfactory online obstacle avoidance performance.

Keywords: autonomous vehicles, obstacle avoidance, non-linear model predictive control (NMPC)

1 INTRODUCTION

There is a recent trend in the automobile industry towards active safety which, as opposed to the concept of passive safety, refers to the active control of the vehicle system using an understanding of the state of the vehicle to avoid or minimize the effect of an accident. Recent studies have shown that vehicle stability and handling is efficiently improved by integrating individual chassis control systems such as electronic stability control, active front steering, and continuous damping control [**1–5**]. Another example of active safety is the adaptive cruise control (ACC) system, which automatically accelerates and decelerates the vehicle to maintain a specified distance from the vehicle in front. Several automakers have already introduced ACC into their top-of-the-line cars, and some are pursuing research to integrate the ACC system with collision warning/avoidance systems to deal with various traffic situations [**6–9**]. A future direction for active safety technologies is the integration of active and passive safety components to construct a system in which components communicate and share data.

Recent studies have reported on active control of steering and braking of vehicles using model predictive controls (MPC) [**10–12**]. In these studies, the steering angle and the tyre slip ratios of the autonomous vehicles are controlled to direct the vehicle along a pre-planned trajectory, which is assumed to be collision-free. In the model predictive framework, a dynamic model of the system is used to predict the outcome of the system, which is then used to obtain an optimal input sequence that minimizes the performance index at each time step. Due to this look-ahead predictive property, MPC can be utilized to generate obstacle-avoiding trajectories by augmenting the performance index with local obstacle information. The look-ahead property also reduces the possibility of falling into local minima, a

*Corresponding author: Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul, Republic of Korea.
email: hjinkim@snu.ac.kr*

well-known problem of potential function methods. In order to avoid local minima problems, modified versions of the potential field method have been proposed. Examples are using harmonic potential fields [**13**, **14**], or navigation functions [**15**]. Also, the wavefront propagation method generates artificial potential fields without local minima [**16**]. However, there are still issues remaining from a practical viewpoint, because it is not simple to integrate system dynamics within these approaches. In addition, they may involve solving partial differential equations, or require discretization of the state space.

The idea of using MPC as an obstacle avoidance algorithm has been applied to collision avoidance problems of unmanned aerial vehicles [**17**, **18**] and autonomous surface vessels [**19**]. In these works, the distances to adjacent obstacles are used to reflect the threat of the obstacles in the performance index. However, wheeled vehicles face more cramped environments when compared with such vehicles, and using only the distance to obstacles to reflect obstacle threat has drawbacks in some cases [**20**].

This paper considers online obstacle avoidance for a full non-linear vehicle model using the model predictive framework. MPC is used to generate safe trajectories and obtain the corresponding input sequences for the vehicle. In order to mitigate the complexity of the full non-linear vehicle model and the computation load in optimization, a simplified model of the vehicle is used in the MPC framework. This makes the simultaneously obtained control input sequence no longer valid to the full non-linear vehicle model. This has motivated the use of the MPC only for generating safe trajectories using simplified dynamics of the vehicle, and to use a separate controller for generating actual control inputs for the full non-linear vehicle model, for tracking the generated trajectory. To guarantee that the actual vehicle can track the generated trajectory, adequate assumptions are made about the actual vehicle to derive the simplified vehicle model. Constraints are introduced to maintain specific state variables and physical parameters in the valid region of the assumptions. The tracking control that generates the actual inputs for the vehicle is composed of a proportional–integral–derivative (PID) controller based on the inverse dynamics of the vehicle powertrain model to control the velocity of the vehicle, and a linear quadratic regulator (LQR) to control the steering angle of the vehicle. In the model predictive framework, the parallax concept is used to reflect obstacle threat which shows improvement over distance-based methods [**20**]. The parallax-based method takes into account, in a simple way, the dimension and the moving direction of the vehicle, which distance-based methods are unable to do. As discussed above, using MPC has benefits over potential-function-based navigation methods in that there is a lower possibility of falling in local minima due to the look-ahead horizon. This is especially advantageous in cramped ground environments where potential functions have a high probability of having local minima.

The paper is structured as follows. Section 2 describes the layout of the overall control structure. Section 3 formulates the MPC framework which is used to generate safe trajectories for the vehicle. Section 4 shows the design procedures of the longitudinal and lateral controllers for tracking the generated trajectory. Section 5 discusses simulation results. Conclusions are presented in section 6.

2 CONTROL STRUCTURE

The purpose of the overall controller is to navigate the vehicle safely to a predesignated goal point, while avoiding collisions with unknown obstacles in an environment. The controller consists of two main layers: one that generates safe trajectories based on MPC, and the other that generates actual control inputs for the vehicle to track the generated trajectory. Figure 1 shows the overall controller layout. In order to reduce the computation load required to the non-linear optimization of the MPC framework, a simplified model of the vehicle is used in the trajectory generation layer. The simplified model is derived so that it has enough accuracy for the purpose of trajectory generation. The body slip angle and yaw rate of the simplified model, which are the significant state variables that determine the trajectory of the vehicle, show similar responses to the actual vehicle.

The tracking controller consists of separate controllers for longitudinal and lateral dynamics. The longitudinal dynamics of the vehicle are controlled using a PID controller, which is based on the inverse dynamics of the vehicle powertrain model. The engine throttle angle and the brake pressure are controlled to accelerate and decelerate the vehicle. The switching between the two is governed by the throttle-brake switching line. The lateral dynamics of the vehicle are controlled using a LQR controller. A bicycle model is used to obtain the feedforward and feedback parts of the steering input.

In simulations throughout the paper, the control inputs obtained by the tracking layer are applied

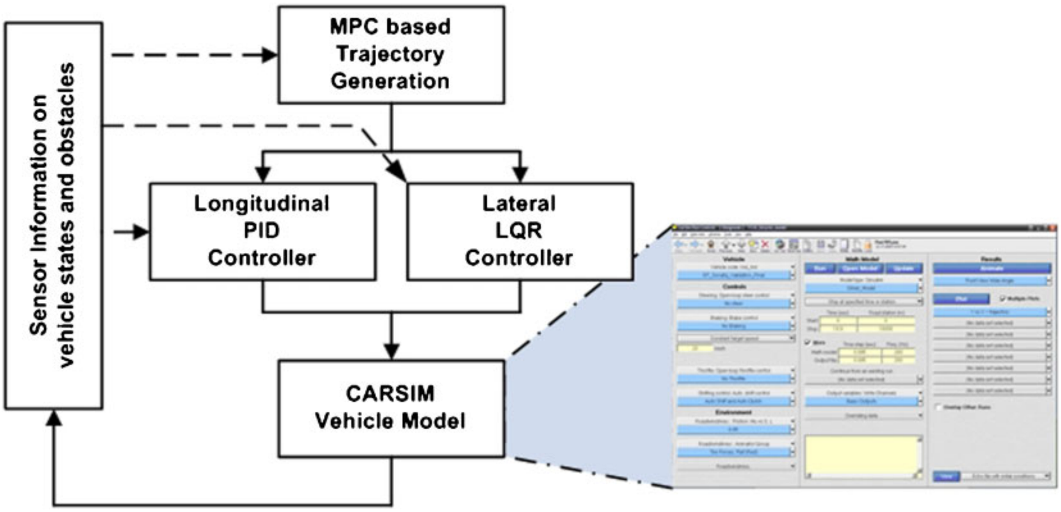
MPC based Trajectory Generation
Sensor Information on vehicle states and obstacles
Longitudinal PID Controller
Lateral LQR Controller
CARSIM Vehicle Model

Fig. 1 Overall control structure and a typical screenshot of the CarSim

to the CarSim vehicle model. CarSim is a software which builds full non-linear models of actual vehicles when parameters such as dimensions of the vehicle platform, specifications of the engine and tyres, and coefficients of the suspensions are given. The CarSim vehicle model has been validated with experimental results of an actual vehicle [**21**]. A typical screenshot of the CarSim is shown in Fig. 1.

3 TRAJECTORY GENERATION

MPC is based on calculating the optimal input sequence with respect to a performance index, over a finite time duration at each time step. The use of a finite horizon starting at each time step is referred to as the receding horizon principle. Using this principle, safe trajectories can be generated by formulating sensor-based obstacle information into the performance index at each time step. Depending on how obstacle information is formulated into the performance index, the resulting trajectories may differ from one another. The most conventional way of formulating obstacle threat is to penalize the proximity to adjacent obstacles. Such distance-based methods are used widely in many path planning algorithms [**17**, **19**]. The method used in this paper is the parallax-based obstacle avoidance term proposed in reference [**20**], which has shown improvements over distance-based methods.

3.1 Modelling

3.1.1 Vehicle model

A simplified non-linear bicycle model (see Fig. 2) is used to describe the dynamics of the vehicle. The following assumptions have been made to the bicycle model:

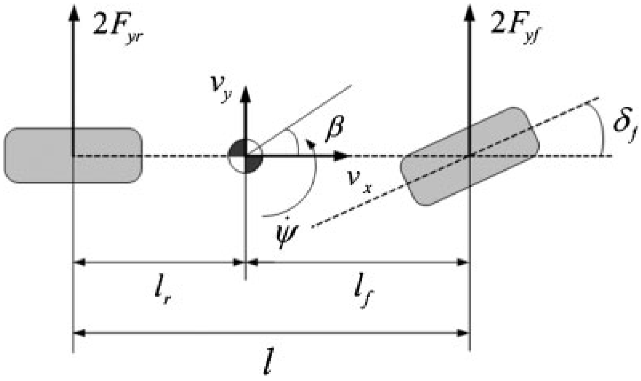

Fig. 2 Simplified bicycle model

(a) longitudinal velocity is constant: $v_x = \text{constant}$, slip ratios ≈ 0;
(b) δ_f is small: $\sin \delta_\text{f} = 0$, $\cos \delta_\text{f} = 1$;
(c) α on both sides are the same: $\alpha_\text{fl} = \alpha_\text{fr} = \alpha_\text{f}$, $\alpha_\text{rl} = \alpha_\text{rr} = \alpha_\text{r}$.

With these assumptions, the dynamics of the bicycle model can be described as follows

$$\begin{aligned} m v_x \left(\dot{\beta} + \dot{\psi}\right) &= F_{y,\text{fl}} + F_{y,\text{fr}} + F_{y,\text{rl}} + F_{y,\text{rr}} \\ I_z \ddot{\psi} &= l_\text{f}\left(F_{y,\text{fl}} + F_{y,\text{fr}}\right) - l_\text{r}\left(F_{y,\text{rl}} + F_{y,\text{rr}}\right) \end{aligned} \tag{1}$$

In equation (1), under the assumption that the slip angles are small, linearized lateral tyre forces are used

$$F_{y,\mathrm{fl}} = F_{y,\mathrm{fr}} = C_\mathrm{f}\alpha_\mathrm{f} = C_\mathrm{f}\left[\delta_\mathrm{f} - \left(\beta + l_\mathrm{f}\dot{\psi}\Big/v_x\right)\right]$$
$$F_{y,\mathrm{rl}} = F_{y,\mathrm{rr}} = C_\mathrm{r}\alpha_\mathrm{r} = C_\mathrm{r}\left(-\beta + l_\mathrm{r}\dot{\psi}\Big/v_x\right) \tag{2}$$

where C_f and C_r denote the cornering stiffness of the front and rear tyres, respectively. The motion of the vehicle in the inertial frame is described by

$$\dot{X} = v_x\cos\psi - v_x\tan\beta\sin\psi$$
$$\dot{Y} = v_x\sin\psi + v_x\tan\beta\cos\psi \tag{3}$$

Although the actual input command to the vehicle dynamics is the steering angle of the front wheels, δ_f, the input is redefined to be $u = \dot{\delta}_\mathrm{f}$, and δ_f is set to be a state variable of the system. This is so that the rate of change of δ_f can be simply constrained by placing a limit on u.

Setting the state vector as $\xi = [\beta\ \psi\ \dot{\psi}\ X\ Y\ \delta_\mathrm{f}]$ and using equations (1) to (3), the state space equations for the bicycle model can be written in the following vector form

$$\dot{\xi} = f(\xi, u)$$
$$\boldsymbol{\eta} = h(\xi) \tag{4}$$

The output $\boldsymbol{\eta}$ is set to be the position of the vehicle in the inertial frame so that the trajectories can be generated based on the output variable

$$\boldsymbol{\eta} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}\xi$$
$$= [X \quad Y]^\mathrm{T} \tag{5}$$

To guarantee that the actual vehicle can track the generated trajectory based on this simplified model, the assumptions placed on the actual vehicle must be satisfied. The assumptions made while deriving this model imply that δ_f and α must be small. This leads to a direct constraint on these variables to have small values. If the constraints on the steering angle and tyre slip angles were to be used, complicated derivative terms would appear in the non-linear optimization process. Instead of the two individual constraints, the following constraint on the instantaneous turning radius is introduced, because it has the effect of keeping the values of δ_f, and α small from experience. Also, constraining the turning radius has the effect of maintaining the lateral acceleration at a desirable value, as can be seen in equation (7)

$$\rho \geqslant \rho_{\min}(v_x) \tag{6}$$

$$a_\mathrm{c} = \frac{v_x{}^2}{\rho} \tag{7}$$

where, $\rho_{\min}(v_x)$ denotes the minimum turning radius at the specified velocity, and a_c denotes the lateral acceleration of the vehicle. The instantaneous turning radius can be obtained by the following

$$\rho = \frac{v_x}{\dot{\beta} + \dot{\psi}} \tag{8}$$

The data corresponding to $\rho_{\min}(v_x)$ in equation (6) were obtained by simulating the steady-state trajectories at various velocities using the CarSim vehicle model. Figure 3 shows the data obtained by the simulation. The state space model of equation (4), along with the constraint of equation (6) and simulation data in Fig. 3, are used throughout the MPC framework.

3.1.2 Model validation

The model used for trajectory generation in this study is the bicycle model in which the vehicle is reduced to a mass with inertia moving in a plane. In addition, a linear tyre model is used, in which each

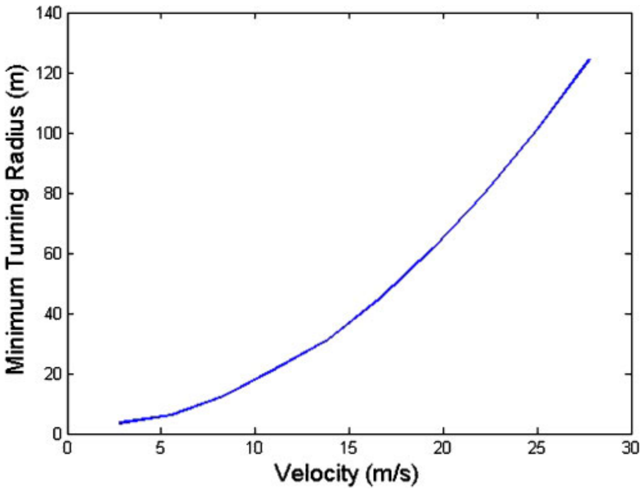

Fig. 3 Minimum turning radius ($\rho_{\min}$) with respect to longitudinal velocity (v_x) obtained from the CarSim vehicle model

tyre force is a linear function of the tyre slip angle and the cornering stiffness. For the exponentially increasing sinusoidal steering manoeuvre of Fig. 4 performed at 70 km/h, Fig. 5 shows a comparison between the bicycle model and the CarSim vehicle model. The body slip angle and yaw rate of the two models start to disagree with each other. This is due to the significantly large steering angle, which induces tyre slip angles larger than 4°. Because the bicycle model uses a linear lateral tyre model that is valid in the range of $-4^\circ \leqslant \alpha \leqslant 4^\circ$, the lateral tyre forces between the two models start to disagree with each other. Consequently, the body slip angle and yaw rate of the two models start to differ from each other.

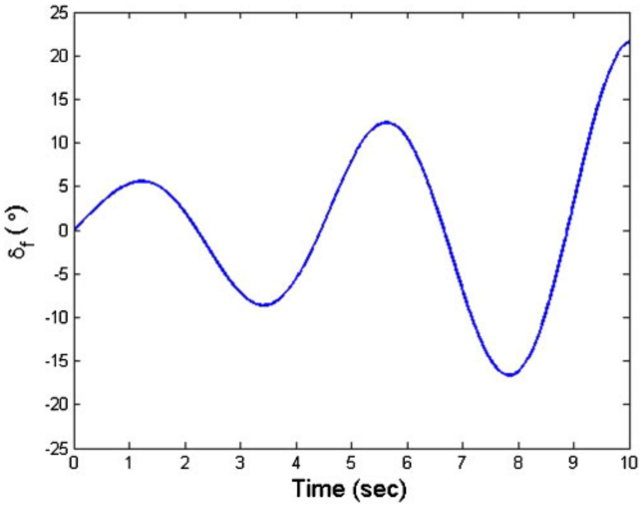

Fig. 4 Steering wheel angle input used for comparison between the bicycle model and the CarSim vehicle model

The two models agree with each other when the magnitude of the lateral acceleration is under 0.5*g*, and magnitudes of the tyre slip angles are under 4°. This can be achieved by constraining the turning radius as in equation (6). Since vehicle behaviour can be well represented by the bicycle model under these driving conditions, the bicycle model can be used for the generation of the vehicle trajectory. Figure 6 shows simulation results at 20 km/h, which is set as the vehicle speed in the simulations shown later. It can be seen that the body slip angle and yaw

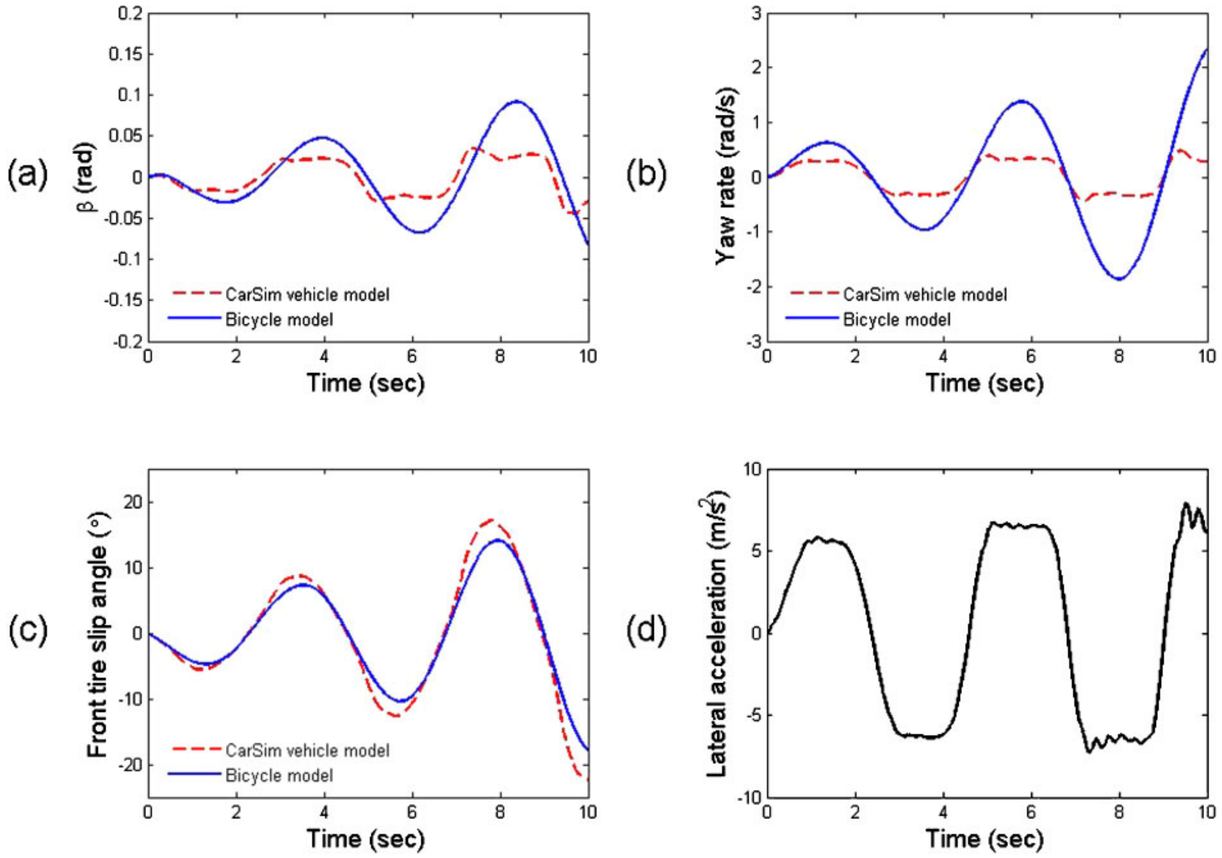

Fig. 5 Comparison between the bicycle model and CarSim vehicle model at 70 km/h: (a) body side slip angles, (b) yaw rates, (c) front tyre slip angles, and (d) lateral acceleration from the CarSim vehicle model

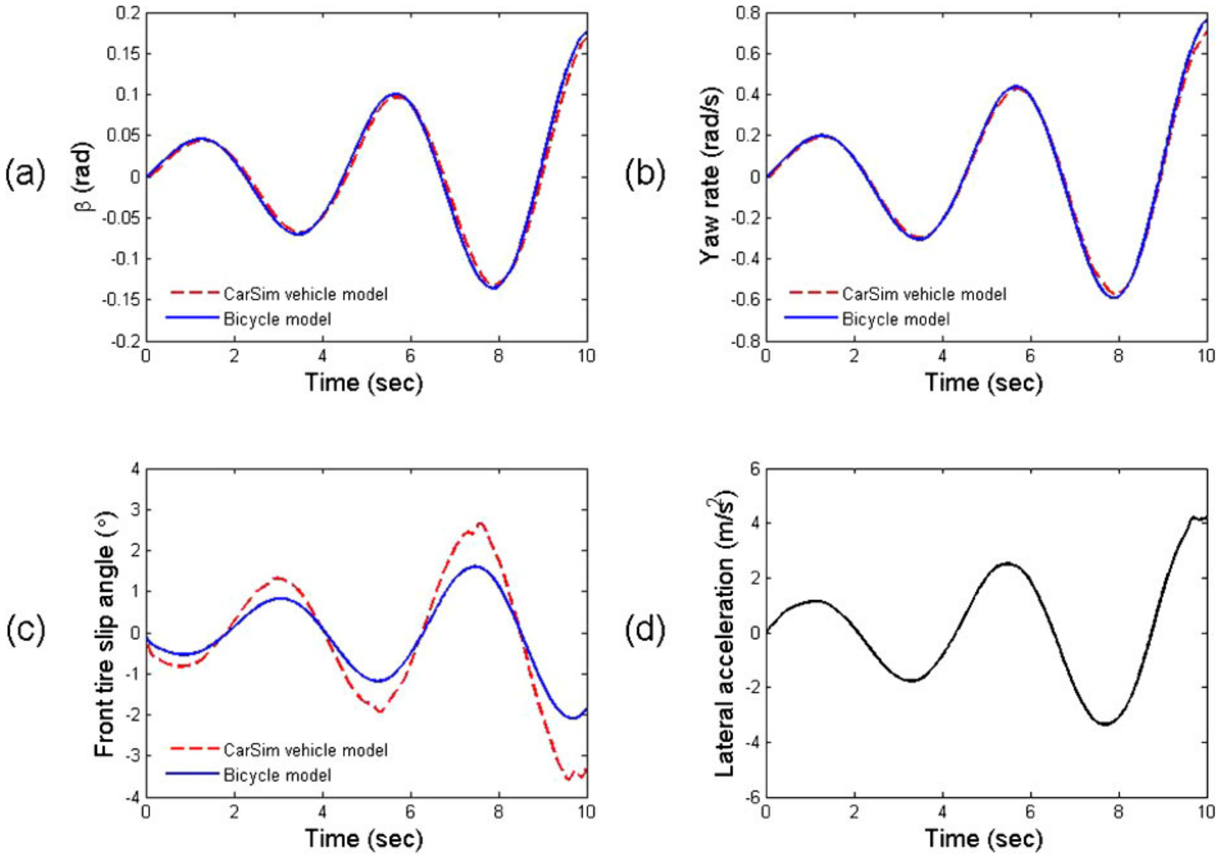

Fig. 6 Comparison between the bicycle model and CarSim vehicle model at 20 km/h: (a) body side slip angles, (b) yaw rates, (c) front tyre slip angles, and (d) lateral acceleration from the CarSim vehicle model

rate, which are the dominant factors of the trajectory of the vehicle, agree with each other.

3.2 MPC

3.2.1 Basic MPC algorithm

For the purpose of discrete time-domain implementation, equation (4) is discretized following the Euler method with time interval ΔT

$$\begin{aligned}\xi_{k+1} &= \xi_k + \Delta T f_k(\xi_k, \boldsymbol{u}_k)\\ &\overset{\Delta}{=} f^{\mathrm{dt}}(\xi_k, \boldsymbol{u}_k)\end{aligned} \qquad (9)$$

Given the current state ξ_0 at the current time step t, the purpose of MPC is to compute the optimal control sequence $[\boldsymbol{u}_k{}^*]_{k=0}^{N-1}$ with the receding horizon principle which solves the following

$$\min \quad J\left([\xi_k]_{k=0}^{N}, [\boldsymbol{u}_k]_{k=0}^{N-1}\right) \qquad (10)$$

subject to

$$\xi_{k+1} - f^{\mathrm{dt}}(\xi_k, \boldsymbol{u}_k) = 0 \qquad (11)$$

$$|u_k| - u_{\mathrm{sat}} \leqslant 0,\ k = 0, \ldots, N-1 \qquad (12)$$

$$|\rho_k| - \rho_{\min}(v_x) \geqslant 0,\ k = 0, \ldots, N \qquad (13)$$

where N is the length of the look-ahead horizon, k is the time step within the look-ahead horizon, and u_{sat} denotes the saturation value of the control input magnitude. Equation (11) is the equality condition related to the dynamics of the vehicle, equation (12) is the constraint on the control input, and equation (13) is the constraint on the instantaneous turning radius discussed in section 3.1.1.

The performance index to be minimized in equation (10) is defined as

$$J = \phi(\tilde{\boldsymbol{\eta}}_N) + \sum_{k=0}^{N-1} L(\tilde{\boldsymbol{\eta}}_k, \boldsymbol{u}_k) \qquad (14)$$

$$\phi = \frac{1}{2}\tilde{\boldsymbol{\eta}}_N^{\mathrm{T}}\mathbf{P}\tilde{\boldsymbol{\eta}}_N \tag{15}$$

$$L = \frac{1}{2}\tilde{\boldsymbol{\eta}}_k^{\mathrm{T}}\mathbf{Q}\tilde{\boldsymbol{\eta}}_k + \frac{1}{2}\boldsymbol{u}_k^{\mathrm{T}}\mathbf{R}\boldsymbol{u}_k \tag{16}$$

where $\tilde{\boldsymbol{\eta}}_k$ represents the deviation from the pre-planned reference trajectory $\boldsymbol{\eta}_{\mathrm{ref},k}$ at time step k, i.e. $\tilde{\boldsymbol{\eta}}_k = \boldsymbol{\eta}_{\mathrm{ref},k} - \boldsymbol{\eta}_k$, and **P**, **Q**, and **R** are constant positive-definite weighting matrices. Equation (15) and the first term of equation (16) penalize the deviation from the reference along the horizon. The second term of equation (16) penalizes the control input in terms of energy consumption. Thus, equation (14) can be seen as a performance index for tracking a predefined reference trajectory while minimizing the input effort. For trajectory generation in unknown environments, additional cost terms must be augmented into the performance index which will be discussed in the next section.

The equality and inequality constraints, equations (11) to (13), are incorporated into the performance index as the following

$$\begin{aligned} J_{\mathrm{a}} = {} & \phi(\tilde{\boldsymbol{\eta}}_N) + \mu_\rho G_\rho(\boldsymbol{\rho}_N) \\ & + \sum_{k=0}^{N-1} \left[L(\tilde{\boldsymbol{\eta}}_k, \boldsymbol{u}_k) + \boldsymbol{\lambda}_{k+1}^{\mathrm{T}} \left(\xi_{k+1} - f^{\mathrm{dt}}(\xi_k, \boldsymbol{u}_k) \right) \right. \\ & \left. + \mu_u G_u(\boldsymbol{u}_k) + \mu_\rho G_\rho(\boldsymbol{\rho}_k) \right] \end{aligned} \tag{17}$$

where $\boldsymbol{\lambda}_k^{\mathrm{T}}$ is a sequence of Lagrange multiplier vectors and μ_u, μ_ρ are weighting parameters for the inequality constraints on the control input and instantaneous turning radius defined as

$$G_u(\boldsymbol{u}_k) = \begin{cases} \frac{1}{2}(|\boldsymbol{u}_k| - u_{\mathrm{sat}})^2, & \text{if } |\boldsymbol{u}_k| - u_{\mathrm{sat}} > 0 \\ 0, & \text{else} \end{cases} \tag{18}$$

$$G_\rho(\boldsymbol{\rho}_k) = \begin{cases} \frac{1}{2}(|\boldsymbol{\rho}_k| - \rho_{\min}(v_x))^2, & \text{if } |\boldsymbol{\rho}_k| - \rho_{\min}(v_x) < 0 \\ 0, & \text{else} \end{cases} \tag{19}$$

3.2.2 *Implementation and modification for obstacle avoidance*

In the problem setting considered in this paper, the vehicle is to traverse an unknown environment to a designated goal point with only local information on obstacles. Because the environment is assumed to be unknown, the reference trajectory is set to be a straight line connecting the start and goal point. It is difficult to define this in the time domain, as it is unpredictable how the vehicle will detour around unknown obstacles. Thus, the reference trajectory is defined as a sequence of points in the space domain. One way of defining $\boldsymbol{\eta}_{\mathrm{ref},k}$ is illustrated in Fig. 7. Here, $\boldsymbol{\eta}_{\mathrm{ref},k}$ is defined as the point on the reference trajectory which has the shortest distance to $\boldsymbol{\eta}_k$.

In order to generate trajectories which avoid obstacles, an additional cost term that reflects local obstacle threat (denoted as the obstacle avoidance term) must be added to the augmented performance index, J_{a} of equation (17). Following from reference [**20**], the obstacle avoidance term is defined based on the parallax angle, i.e. the angle between the straight lines from two different observation points to an object. Using the parallax angle to indicate obstacle threat is inspired from the similar mechanism of humans perceiving distance to objects. The parallax angle from the rear vertices of the vehicle is illustrated in Fig. 8 and the definition follows in equation (20)

$$PF_k^{\mathrm{obs}} = \max_j \theta_j$$

where

$$\begin{aligned} \theta_j &= \pi - \left(\theta_{\mathrm{rl},j} + \theta_{\mathrm{rr},j}\right) \\ \theta_{\mathrm{rl},j} &= \arctan\frac{x_{\mathrm{obs},j} + L_{\mathrm{r}}}{w - y_{\mathrm{obs},j}} \\ \theta_{\mathrm{rr},j} &= \arctan\frac{x_{\mathrm{obs},j} + L_{\mathrm{r}}}{w + y_{\mathrm{obs},j}} \end{aligned} \tag{20}$$

In equation (20), $x_{\mathrm{obs},j}$ and $y_{\mathrm{obs},j}$ denote the position of the jth obstacle in the vehicle frame, that is

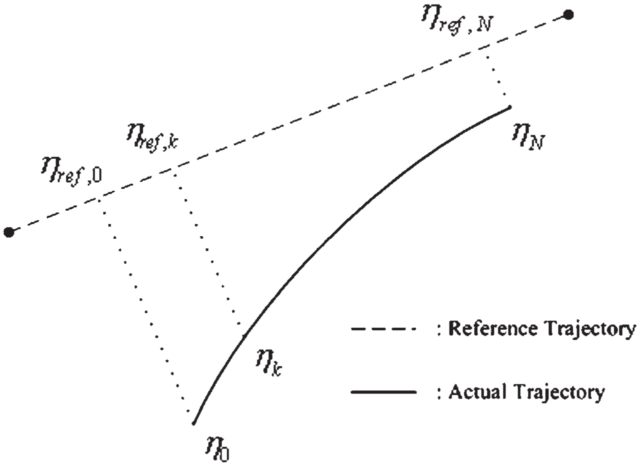

Fig. 7 Defining discrete reference points along the predicting horizon of the MPC framework

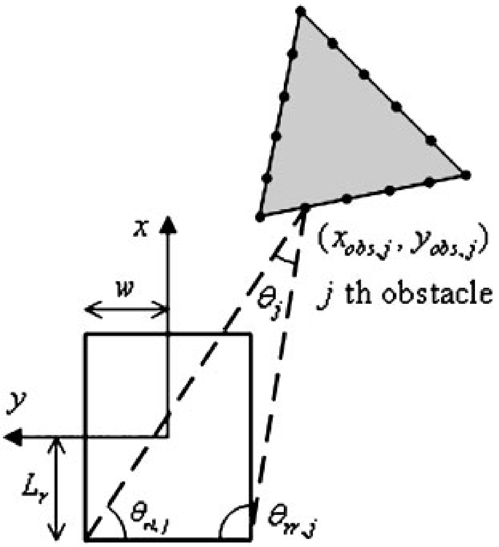

Fig. 8 Definition of the parallax angle (θ_j) of the jth obstacle from the rear vertices of the vehicle

$$\begin{bmatrix} x_{\mathrm{obs},j} \\ y_{\mathrm{obs},j} \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} X_{\mathrm{obs},j} - X \\ Y_{\mathrm{obs},j} - Y \end{bmatrix} \tag{21}$$

where $[X_{\mathrm{obs},j} \; Y_{\mathrm{obs},j}]$ denotes the position of the jth obstacle in the inertial frame. L_{r} denotes the distance from the centre of gravity to the rear end of the vehicle, and w is half the width of the vehicle.

The parallax-based obstacle avoidance term has two clear advantages over distance-based methods:

(a) it considers the moving direction of the vehicle;
(b) it takes into account the dimension of the vehicle in an explicit manner.

To see how the parallax-based obstacle avoidance term considers the moving direction of the vehicle, suppose that two obstacles lie at a same distance from the centre of gravity of the vehicle. Also, suppose that one of those two obstacles lies in the moving direction of the vehicle, and the other lies more to the side of the vehicle. This is illustrated in Fig. 9. Clearly, it can be seen that the parallax angle of the obstacle lying ahead of the vehicle is larger than that of the obstacle lying to the side of the vehicle ($\theta_n > \theta_m$), indicating more threat according to equation (20). This agrees with the intuitive judgement that the obstacle in front of the vehicle is potentially more dangerous. This cannot be considered in a distance-based method, because both obstacles will be treated as having the same level of threat to the vehicle. In equation (20), it can be seen that the dimensions of the vehicle, w and L_{r}, appear explicitly in the parallax-based method. On the other hand, distance-based methods will have to undergo additional procedures, such as selecting several points on the edges of the vehicle from which the distances to obstacles are computed, to take into account the dimension of the vehicle.

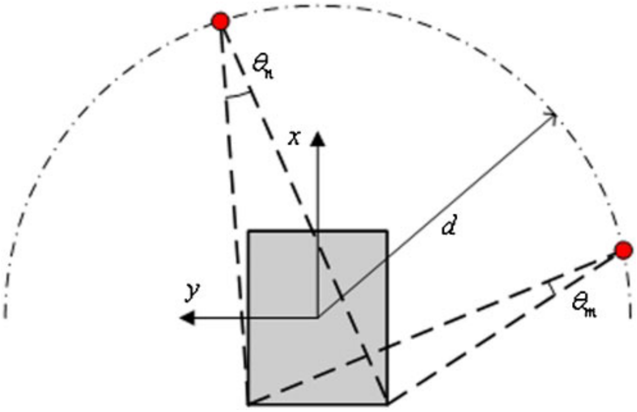

Fig. 9 Comparison of the parallax angle between two obstacles which lie at a same distance from the centre of gravity of the vehicle. The obstacle lying in front of the vehicle has a larger parallax angle than the obstacle lying more to the side of the vehicle (i.e. $\theta_n > \theta_m$)

The parallax angle to obstacles can be defined from the front vertices of the vehicle or from the rear vertices of the vehicle. Using both front and rear parallax angles gives some degree of freedom in weighting the obstacle threat considering its relative position to the vehicle. Using only the parallax angle from the rear vertices does not degrade the overall performance, as it covers all of the front and side of the vehicle, but all obstacles are weighted with the same weighting parameter. This is illustrated in Fig. 10. The resulting augmented cost function with obstacle avoidance is

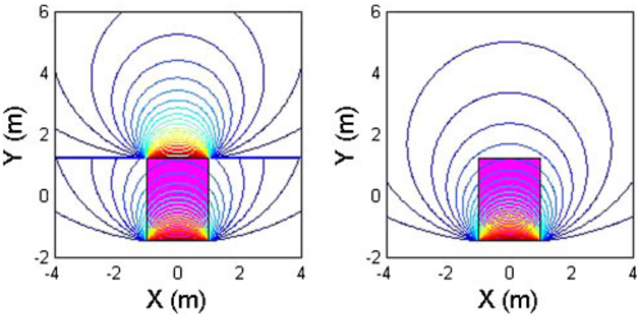

Fig. 10 Comparison of local obstacle threat between two parallax-based methods: (left) using both front and rear parallax terms, (right) using only the rear parallax term. The lines connect the locations which have the same threat in terms of the parallax

$$
\begin{aligned}
J_{\mathrm{a}} = {} & \phi(\tilde{\boldsymbol{\eta}}_N) + \mu_\rho G_\rho(\boldsymbol{\rho}_N) \\
& + \sum_{k=0}^{N-1} \left[L(\tilde{\boldsymbol{\eta}}_k, \boldsymbol{u}_k) + \boldsymbol{\lambda}_{k+1}^{\mathrm{T}} \left(\xi_{k+1} - f^{\mathrm{dt}}(\xi_k, \boldsymbol{u}_k) \right) \right. \\
& \left. + \mu_u G_u(\boldsymbol{u}_k) + \mu_\rho G_\rho(\boldsymbol{\rho}_k) + PF_k^{\mathrm{obs}} \right] \qquad (22)
\end{aligned}
$$

By redefining the first line of equation (22) as $\Phi(\xi_N)$, the following augmented cost function is obtained

$$
\begin{aligned}
J_{\mathrm{a}} = \Phi(\xi_N) + \sum_{k=0}^{N-1} & \left[L(\tilde{\boldsymbol{\eta}}_k, \boldsymbol{u}_k) + \boldsymbol{\lambda}_{k+1}^{\mathrm{T}} \left(\xi_{k+1} - f^{\mathrm{dt}}(\xi_k, \boldsymbol{u}_k) \right) \right. \\
& \left. + \mu_u G_u(\boldsymbol{u}_k) + \mu_\rho G_\rho(\boldsymbol{\rho}_k) + PF_k^{\mathrm{obs}} \right] \qquad (23)
\end{aligned}
$$

3.2.3 *Online optimization*

The gradient search method [22] is used to obtain the optimal input sequence $[\boldsymbol{u}_k{}^*]_{k=0}^{N-1}$. Defining the Hamiltonian as

$$
H_k = L(\tilde{\boldsymbol{\eta}}_k, \boldsymbol{u}_k) + \boldsymbol{\lambda}_{k+1}^{\mathrm{T}} f^{\mathrm{dt}}(\xi_k, \boldsymbol{u}_k) + \mu_u G_u + \mu_\rho G_\rho + PF_k^{\mathrm{obs}} \qquad (24)
$$

for $k = 0, \ldots, N$, the total variation of the cost function can be written as the following

$$
\begin{aligned}
\mathrm{d}J_{\mathrm{a}} = {} & \left(\frac{\partial \Phi}{\partial \xi_N} - \boldsymbol{\lambda}_N^{\mathrm{T}} \right) \mathrm{d}\xi_N + \frac{\partial H_0}{\partial \xi_0} \mathrm{d}\xi_0 + \frac{\partial H_0}{\partial \boldsymbol{u}_0} \mathrm{d}\boldsymbol{u}_0 \\
& + \sum_{k=1}^{N-1} \left[\left(\frac{\partial H_k}{\partial \xi_k} - \boldsymbol{\lambda}_k^{\mathrm{T}} \right) \mathrm{d}\xi_k + \frac{\partial H_k}{\partial \boldsymbol{u}_k} \mathrm{d}\boldsymbol{u}_k \right] \qquad (25)
\end{aligned}
$$

Setting the Lagrange multiplier vectors as

$$
\boldsymbol{\lambda}_N^{\mathrm{T}} = \frac{\partial \Phi}{\partial \xi_N}, \quad \boldsymbol{\lambda}_k^{\mathrm{T}} = \frac{\partial H_k}{\partial \xi_k} \qquad (26)
$$

equation (25) can be rewritten in the following manner

$$
\mathrm{d}J_{\mathrm{a}} = \sum_{k=0}^{N-1} \frac{\partial H_k}{\partial \boldsymbol{u}_k} \mathrm{d}\boldsymbol{u}_k + \boldsymbol{\lambda}_0^{\mathrm{T}} \mathrm{d}\xi_0 \qquad (27)
$$

Using equations (26) and (27), the optimization process shown in Table 1 is executed. Within Table 1, i represents the iteration number at each time step and Δi represents the step size in the gradient descent.

The values of the weighting parameters μ_u and μ_ρ in equation (23) decide how much penalty will be given when variables do not satisfy given constraints. Thus, it is desirable that μ_u and μ_ρ are set to have large values with respect to the other terms in the performance index in order to ensure that the constraints are satisfied. For larger values of μ_u and μ_ρ, the optimization will focus on satisfying the constraints, while on the other hand, for smaller values of μ_u and μ_ρ, the optimization will focus on reference tracking and obstacle avoidance. Also, considerations should be made to tune the respective values of μ_u and μ_ρ with respect to the step size Δ_i, to obtain solutions in adequate time.

Table 1 Optimization process

```
while $|\Delta J = J^{i+1} - J^i| > \varepsilon$
  for $k = 0, \ldots, N$
    compute $\{\xi_k\}_1^N$ using equation (4)
  end
  for $k = 1, \ldots, N$
    compute $\{\boldsymbol{\lambda}_k\}_1^N$ using equation (26)
  end
  for $k = 0, \ldots, N-1$
    compute $\frac{\partial H_k}{\partial \boldsymbol{u}_k}$ using equation (24)
  end
  if $\Delta J < 0$
    for $k = 0, \ldots, N-1$
      $\boldsymbol{u}_k^{i+1} := \boldsymbol{u}_k^i - \Delta_i \frac{\partial H_k}{\partial \boldsymbol{u}_k}$
      $i \leftarrow i+1$
    end
  else
    reduce $\Delta_i$
  end
end
```

4 TRACKING CONTROLLER

4.1 Longitudinal control for velocity tracking

In order to track the desired longitudinal velocity, the desired acceleration of the vehicle is computed. Then, the desired acceleration is tracked by controlling either the engine throttle angle or brake pressure depending on the state of the vehicle. The engine throttle angle is computed from the desired engine torque using an engine map and a torque-converter map. The brake pressure is obtained from the total brake torque, which is computed from the motion of the vehicle. Throttle-brake switching is determined by using a switching line obtained from the data on zero-throttle acceleration of the vehicle.

When the desired acceleration, a_{des}, is greater than the switching line, i.e. in the throttle-control region, the desired turbine torque of the transmission, $T_{\mathrm{t,des}}$, is computed from the equations of motion of the vehicle as the following

$$
\begin{aligned}
T_{\mathrm{t,des}} &= R_{\mathrm{g}} T_{\mathrm{s,des}} \\
&= R_{\mathrm{g}} r \left[m \left(a_{\mathrm{des}} + K_{\mathrm{p}} e_{\mathrm{a}} + K_{\mathrm{i}} \int e_{\mathrm{a}} \mathrm{d}t \right) + F_{\mathrm{L}} \right] \\
e_{\mathrm{a}} &= a_{\mathrm{des}} - a
\end{aligned} \qquad (28)
$$

where R_{g} denotes the gear ratio from the turbine of the torque converter to the wheels, $T_{\mathrm{s,des}}$ the desired shaft torque of the driving axle shaft, r the effective tyre radius, K_{p} and K_{i} gains, a the actual acceleration of the vehicle, and F_{L} the driving resistance load. It is assumed that the turbine of the torque converter in the vehicle automatic transmission is directly connected to the engine output shaft.

The desired throttle angle, α_{des}, for the desired engine speed, $\omega_{\mathrm{e,des}}$ is computed from the desired net engine torque, $T_{\mathrm{net,des}}$, using the engine map as follows

$$\alpha_{\mathrm{des}} = EM^{-1}\left(\omega_{\mathrm{e}}, T_{\mathrm{net,des}}\right) \tag{29}$$

$$T_{\mathrm{net,des}} = T_{\mathrm{p}}(\omega_{\mathrm{e}}, \omega_{\mathrm{t}}) + K_{\mathrm{e}}\left(\omega_{\mathrm{e,des}} - \omega_{\mathrm{e}}\right) \tag{30}$$

where EM^{-1} is the inverse engine map, T_{p} is the pump torque, and ω_{e}, ω_{t} are the speeds of the engine and turbine, respectively. K_{e} denotes the gain. Typically, the engine map is provided by the engine manufacturer as a lookup table.

The brake torque is applied only when the engine braking is not sufficient to follow the desired acceleration profile. When the desired acceleration for the given vehicle velocity is smaller than the switching line, i.e. in the brake control region, the desired brake torque, $T_{\mathrm{b,des}}$ is computed from the equations of motion of the vehicle as follows

$$T_{\mathrm{b,des}} = -r(ma_{\mathrm{des}} + F_{\mathrm{L}}) + T_{\mathrm{s}} \tag{31}$$

Since the total brake torque is proportional to the brake pressure, the desired brake pressure, $p_{\mathrm{w,des}}$ can be obtained by the following equation

$$p_{\mathrm{w,des}} = \frac{1}{K_{\mathrm{b}}} T_{\mathrm{b,des}} \tag{32}$$

where K_{b} is the lumped gain for the entire brake system. K_{b} lumps all the uncertainties in the brake model from the brake pressure to the brake torque. The parameter K_{b} can be obtained from experimental data. The value of $K_{\mathrm{b}} = 850\,\mathrm{Nm/Pa}$ was used, which provides a good fit to one set of the experimental results. Because the brake actuator system dynamics are not negligible and the brake control system shows non-linear characteristics, a feedforward plus PID control law is used to control the brake pressure

$$\begin{aligned} u_{\mathrm{b}} &= g^{-1}(p_{\mathrm{w,des}}) + K_{\mathrm{P}}e_{\mathrm{p}} + K_{\mathrm{I}}\int e_{\mathrm{p}}\mathrm{d}t + K_{\mathrm{D}}e_{\mathrm{p}} \\ e_{\mathrm{p}} &= p_{\mathrm{w,des}} - p_{\mathrm{w}} \end{aligned} \tag{33}$$

where u_{b} is the applied duty input to the brakes solenoid valve, p_{w} the wheel caliper pressure, K_{P}, K_{I}, and K_{D} the gains. $g(\cdot)$ is the function representing the relationship between u_{b} and the steady-state values of the brake pressure, that is

$$p_{\mathrm{w}} = g(u_{\mathrm{b}}) \tag{34}$$

Figure 11 shows the overall block diagram of the longitudinal controller.

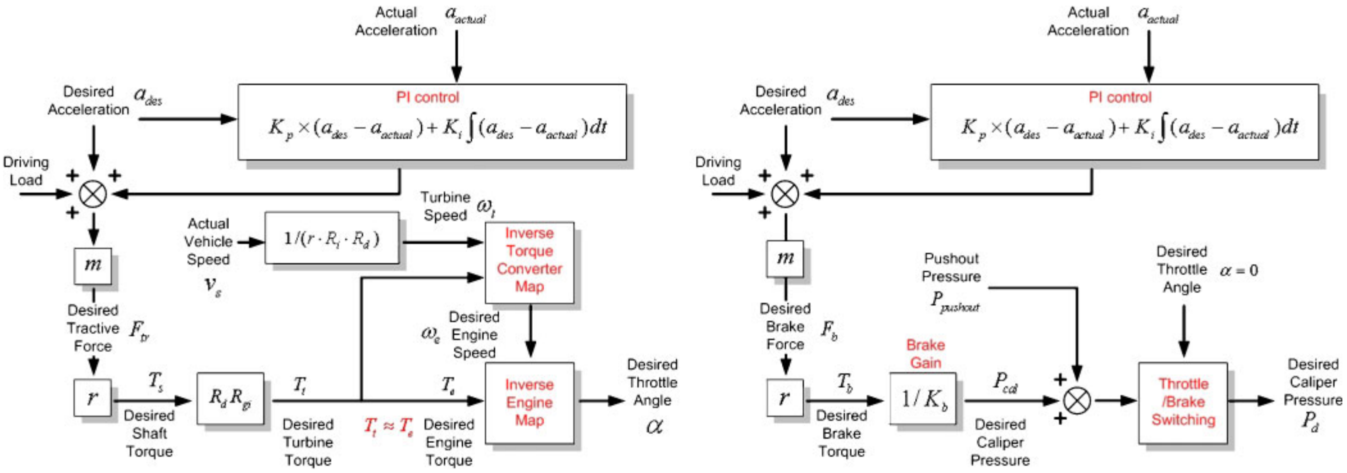
Actual Acceleration a_{actual}
Desired Acceleration a_{des}
PI control
$K_p \times (a_{des} - a_{actual}) + K_i \int (a_{des} - a_{actual})dt$
Driving Load
+
+
+
m
Desired Tractive Force F_{tr}
r
T_s
Desired Shaft Torque
$R_d R_{gi}$
T_t
Desired Turbine Torque
Actual Vehicle Speed v_s
$1/(r \cdot R_i \cdot R_d)$
Turbine Speed ω_t
Inverse Torque Converter Map
ω_e Desired Engine Speed
T_e
$T_t \approx T_e$
Desired Engine Torque
Inverse Engine Map
Desired Throttle Angle α
Actual Acceleration a_{actual}
Desired Acceleration a_{des}
PI control
$K_p \times (a_{des} - a_{actual}) + K_i \int (a_{des} - a_{actual})dt$
Driving Load
+
+
+
m
Desired Brake Force F_b
r
T_b
Desired Brake Torque
Brake Gain
$1/K_b$
P_{cal}
Desired Caliper Pressure
Pushout Pressure $P_{pushout}$
+
+
Desired Throttle Angle $\alpha = 0$
Throttle /Brake Switching
Desired Caliper Pressure P_d

Fig. 11 The left diagram illustrates the procedure for obtaining the throttle angle of the engine, and the right diagram illustrates the procedure for obtaining the brake pressure. The switching between the two controllers is governed by the throttle-brake switching line, which is obtained from data on zero-throttle acceleration of the vehicle

4.2 Lateral control for path tracking

4.2.1 *Vehicle model*

To obtain the feedforward and feedback gains of the LQR controller, the diagram in Fig. 12 is considered, which involves the reference path and the vehicle. In Fig. 12, C is the centre of gravity of the vehicle and R is the intersection point of the lateral axis of the vehicle and the reference trajectory. L_p is the observation distance of the vehicle. The lateral error of the vehicle y_r is defined as the distance between C and R. The yaw angle error is defined as the difference between the heading angle of the vehicle (ε) and the direction of the tangential vector of the reference trajectory at point R (ε_d). The derivatives of the lateral error y_r and the angle of the tangential vector ε_d can be approximated by the following equations

$$\begin{aligned}\dot{y}_\text{r} &= v_y + v_y(\varepsilon - \varepsilon_\text{d})\\ \dot{\varepsilon}_\text{d} &= \frac{v_x}{\rho}\end{aligned} \tag{35}$$

For the vehicle model, a bicycle model similar to the one used in the MPC framework is assumed. The derivation of the model is analogous to the procedure in section 3.1.1, with the substitution of ψ with ε. Also, using the error dynamics defined in equation (35), the following state space equation is obtained

$$\begin{aligned}\dot{\boldsymbol{\chi}} &= \mathbf{A}\boldsymbol{\chi} + \mathbf{B}\delta_\text{f} + \mathbf{F}_\text{d}\boldsymbol{w}_\text{d}\\ \boldsymbol{\chi} &= [\,y_\text{r} \quad \dot{y}_\text{r} \quad \varepsilon - \varepsilon_\text{d} \quad \dot{\varepsilon} - \dot{\varepsilon}_\text{d}\,]\end{aligned} \tag{36}$$

where

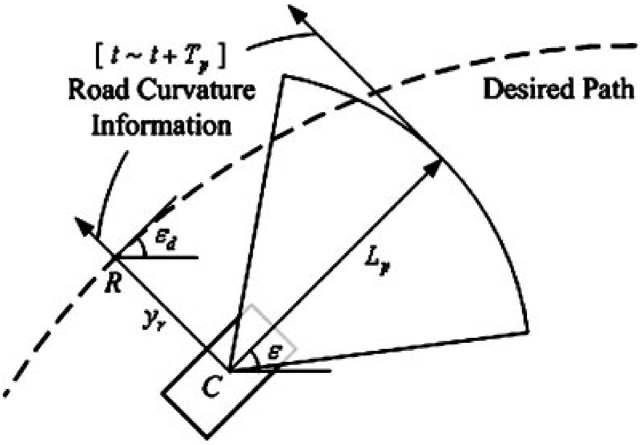

Fig. 12 Reference trajectory and vehicle model used in the lateral controller

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0\\ 0 & \frac{A_1}{v_x} & -A_1 & \frac{A_2}{v_x}\\ 0 & 0 & 0 & 1\\ 0 & \frac{A_3}{v_x} & -A_3 & \frac{A_4}{v_x}\end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0\\ B_1\\ 0\\ B_2\end{bmatrix},$$

$$\mathbf{F}_\text{d} = \begin{bmatrix} 0 & 0\\ 1 & 0\\ 0 & 0\\ 0 & 1\end{bmatrix}, \quad \boldsymbol{w}_\text{d} = \begin{bmatrix} d_1\\ d_2\end{bmatrix}$$

$$A_1 = -2(C_\text{f} + C_\text{r})/m, \quad B_1 = 2C_\text{f}/m$$

$$A_2 = -2(-l_\text{f}C_\text{f} + l_\text{r}C_\text{r})/m, \quad B_2 = 2C_\text{f}l_\text{f}/I_z$$

$$A_3 = -2(-l_\text{f}C_\text{f} + l_\text{r}C_\text{r})/I_z, \quad d_1 = -v_x^2/\rho + A_2\dot{\varepsilon}_\text{d}/v_x$$

$$A_4 = -2\left(l_\text{f}^2C_\text{f} + l_\text{r}^2C_\text{r}\right)/I_z, \quad d_2 = A_4\dot{\varepsilon}_\text{d}/v_x - \ddot{\varepsilon}_\text{d}$$

4.2.2 *LQR controller design*

To obtain an optimal input within a finite preview window, the performance index is defined as

$$J = \int_0^{t_f} \frac{1}{2}\left(\boldsymbol{\chi}^\text{T}\mathbf{Q}\boldsymbol{\chi} + \boldsymbol{u}^\text{T}\mathbf{R}\boldsymbol{u}\right)\text{d}t \tag{37}$$

Introducing Lagrange multipliers, the augmented performance index is

$$J_\text{a} = \int_0^{t_f} \frac{1}{2}\left[\boldsymbol{\chi}^\text{T}\mathbf{Q}\boldsymbol{\chi} + \boldsymbol{u}^\text{T}\mathbf{R}\boldsymbol{u} + \lambda^\text{T}(\mathbf{A}\boldsymbol{\chi} + \mathbf{B}\boldsymbol{u} + \mathbf{F}_\text{d}\boldsymbol{w}_\text{d} - \dot{\boldsymbol{\chi}})\right]\text{d}t \tag{38}$$

Through the Hamiltonian approach, the following necessary conditions for optimality are obtained

$$\begin{aligned}\dot{\lambda} &= -\mathbf{A}^\text{T}\lambda(t) - \mathbf{Q}\boldsymbol{\chi}(t), \quad \lambda(t_\text{f}) = 0\\ \boldsymbol{u}(t) &= -\mathbf{R}^{-1}\mathbf{B}^\text{T}\lambda(t)\\ \dot{\boldsymbol{\chi}} &= \mathbf{A}\boldsymbol{\chi} + \mathbf{B}\boldsymbol{u} + \mathbf{F}_\text{d}\boldsymbol{w}_\text{d}\end{aligned} \tag{39}$$

Assuming that the input has feedforward and feedback parts, the Lagrangian can be set as

$$\lambda(t) = \mathbf{P}(t)\boldsymbol{\chi}(t) + \mathbf{H}(t) \tag{40}$$

Differentiating equation (40) and using equation (39), the following equation is obtained

$$\begin{aligned}&(\dot{\mathbf{P}}+\mathbf{PA}+\mathbf{A}^{\mathrm{T}}\mathbf{P}+\mathbf{Q}-\mathbf{PBR}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{P})\boldsymbol{\chi}(t)\\&\quad+(\dot{\mathbf{H}}+\mathbf{A}^{\mathrm{T}}\mathbf{H}-\mathbf{PBR}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{H}+\mathbf{P}(t)\mathbf{F}_{\mathrm{d}}\boldsymbol{w}_{\mathrm{d}})=0\end{aligned} \tag{41}$$

This gives two matrix differential equations

$$\begin{aligned}&\dot{\mathbf{P}}(t)+\mathbf{P}(t)\mathbf{A}+\mathbf{A}^{\mathrm{T}}\mathbf{P}(t)+\mathbf{Q}-\mathbf{P}(t)\mathbf{BR}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{P}(t)=0,\\&\quad\mathbf{P}(t_{\mathrm{f}})=0\end{aligned} \tag{42}$$

$$\begin{aligned}\dot{\mathbf{H}}(t)&=-\left[\mathbf{A}-\mathbf{BR}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{P}\right]^{\mathrm{T}}\mathbf{H}(t)-\mathbf{PF}_{\mathrm{d}}\boldsymbol{w}_{\mathrm{d}}(t)\\&=-\mathbf{A}_{\mathrm{c}}^{\mathrm{T}}\mathbf{H}(t)-\mathbf{PF}_{\mathrm{d}}\boldsymbol{w}_{\mathrm{d}}(t)\\\mathbf{H}(t_{\mathrm{f}})&=0,\quad \mathbf{A}_{\mathrm{c}}=\mathbf{A}-\mathbf{BR}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{P}\end{aligned} \tag{43}$$

Equation (42) can be changed into an algebraic Riccati equation, assuming steady-state gain P_{ss}

$$\mathbf{P}_{\mathrm{ss}}\mathbf{A}+\mathbf{A}^{\mathrm{T}}\mathbf{P}_{\mathrm{ss}}+\mathbf{Q}-\mathbf{P}_{\mathrm{ss}}\mathbf{BR}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{P}_{\mathrm{ss}}=0 \tag{44}$$

Solving for P_{ss} in equation (44), the feedback gain is obtained. The solution of equation (43) can be obtained using the convolution integral

$$H(t)=\int_{t}^{t+T_{\mathrm{p}}}\mathrm{e}^{-\mathbf{A}_{\mathrm{c}}^{\mathrm{T}}(t-\tau)}\mathbf{P}_{\mathrm{ss}}\mathbf{F}_{\mathrm{d}}w_{\mathrm{d}}(\tau)\mathrm{d}\tau \tag{45}$$

where

$$\mathbf{A}_{\mathrm{c}}=\mathbf{A}-\mathbf{BR}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{P}_{\mathrm{ss}}$$

Changing the integration interval from $\tau\in[t,t+T_{\mathrm{p}}]$ to $\zeta\in[0,T_{\mathrm{p}}]$ by setting $\zeta=\tau-t$, the following equation is obtained

$$H(t)=\int_{0}^{T_{\mathrm{p}}}\mathrm{e}^{\mathbf{A}_{\mathrm{c}}^{\mathrm{T}}\zeta}\mathbf{P}_{\mathrm{ss}}\mathbf{F}_{\mathrm{d}}\boldsymbol{w}_{\mathrm{d}}(t+\zeta)\mathrm{d}\zeta \tag{46}$$

In equation (46), T_{p} is the preview time which is obtained by the observation distance, L_{p}, by the following equation

$$T_{\mathrm{p}}=\frac{L_{\mathrm{p}}}{v_x} \tag{47}$$

Finally, the optimal input is obtained as

$$\delta_{\mathrm{f}}(t)=-\mathbf{G}\boldsymbol{\chi}(t)+\mathbf{M}(t) \tag{48}$$

where

$$\begin{aligned}&\mathbf{G}=\mathbf{R}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{P}_{\mathrm{ss}}\\&\mathbf{M}(t)=\mathbf{R}^{-1}\mathbf{B}^{\mathrm{T}}\int_{0}^{T_{\mathrm{p}}}\mathbf{F}_{1}(\zeta)\mathbf{w}(t+\zeta)\mathrm{d}\zeta\\&\mathbf{F}_{1}(t)=\mathrm{e}^{\mathbf{A}_{\mathrm{c}}^{\mathrm{T}}t}\mathbf{P}_{\mathrm{ss}}\mathbf{F}_{\mathrm{d}}\end{aligned}$$

Figure 13 shows the block diagram of the lateral control structure. In Fig. 13, u_{ff} and u_{fb} denote the feedforward and feedback parts of the input, respectively, and w_{d} denotes the disturbance to the plant defined from the road information.

5 SIMULATION RESULTS

Three scenarios are considered in the simulation. In the first scenario, three circular obstacles with a 4 m diameter are placed slightly off the middle of a 15 m wide road. The vehicle is set to travel at a constant speed of 20 km/h (5.5 m/s). The vehicle is assumed to have been equipped with a sensor with a range of 20 m. The goal is to arrive at the final destination point at [95 m 55 m], passing through waypoints at [40 m 17 m], and [58 m 43 m]. The second scenario considered is a dynamic environment where two vehicle-like obstacles advance towards the vehicle with constant velocity of 24 km/h and 8 km/h, respectively. The waypoints and goal point are the same as the first scenario. In the last scenario, a velocity-varying model of the vehicle is considered. In contrast to the first two scenarios where the

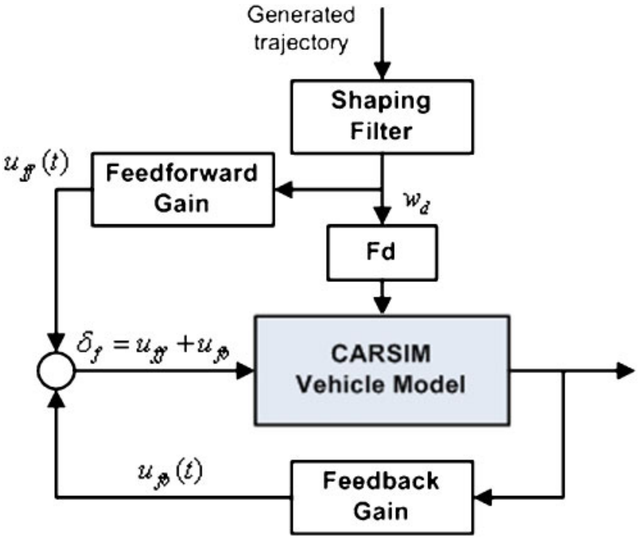

Fig. 13 Lateral control structure consisting of feedforward (u_{ff}) and feedback (u_{fb}) control inputs, designed using LQR

Table 2 Parameters used in the static-environment simulation

Parameter	Value
Horizon length(N)	40
Sampling time (Δt)	0.05 s
Control command limit	$-10^{\circ}\,\mathrm{s}^{-1} \leqslant u \leqslant 10^{\circ}\,\mathrm{s}^{-1}$
Minimum turning radius	15 m
Vehicle specification	$m = 1723.8\,\mathrm{kg}$, $I_z = 4175\,\mathrm{kg{\cdot}m^2}$
Vehicle dimension	(L) 4 m × (W) 1.988 m
Axle length	$l_f = 1.232\,\mathrm{m}$, $l_r = 1.468\,\mathrm{m}$
Cornering stiffness	$C_f = 66\,900\,\mathrm{N/rad}$, $C_r = 62\,700\,\mathrm{N/rad}$

longitudinal velocity was held at a constant value, the longitudinal velocity of the vehicle is set free. The same environment as in the first scenario is used. Trajectories are generated using the MPC at 20 Hz and the tracking controller is set at 200 Hz in all simulations.

5.1 Static environment

Table 2 shows the parameters used in the static-environment simulation. Figure 14 reports the si-

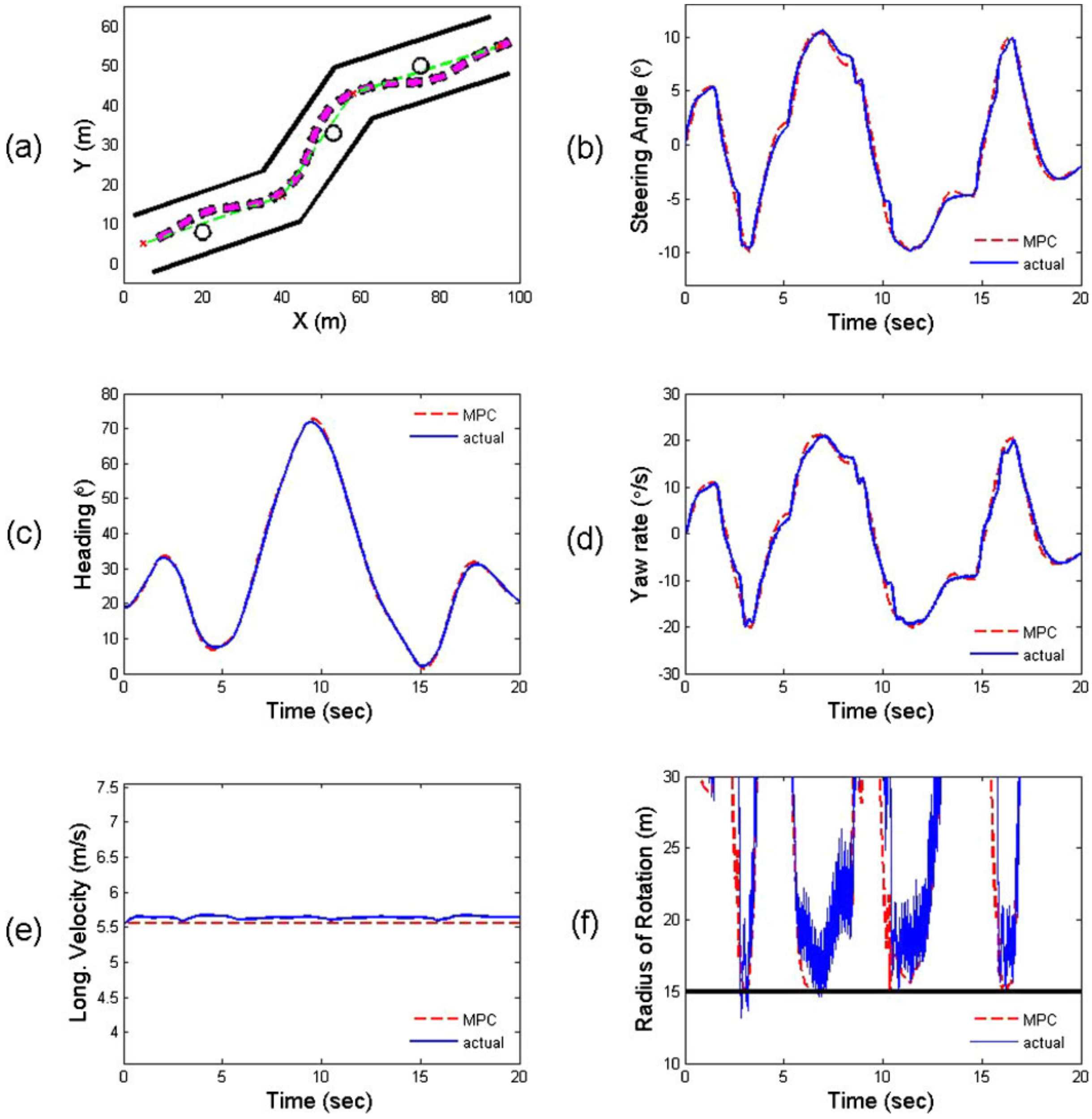

Fig. 14 Simulation results in a static environment at a constant velocity of 20 km/h: (a) resulting trajectory, (b) input history, (c) heading history, (d) yaw rate history, (e) longitudinal velocity history, and (f) instantaneous turning radius along the trajectory

mulation results. Figure 14(a) shows the resulting trajectory, where the dashed line represents the predefined reference trajectory and the rectangle represents the vehicle. Figures 14(b) to (e) show the history of the states of the vehicle, with the dashed line representing states predicted in the MPC and the solid line representing the actual states of the full non-linear vehicle. It can be seen that the states and input predicted by the MPC closely match the actual states and inputs, implying that the generated trajectories are indeed compatible with the actual non-linear vehicle dynamics. Figure 14(f) shows the predicted (dashed line) and actual (solid line) turning radius of the vehicle. The horizontal solid line represents the minimum turning radius, which is constant as the vehicle is moving at a constant velocity.

5.2 Dynamic environment

Table 3 shows the parameters used in the dynamic-environment simulation. A larger control command

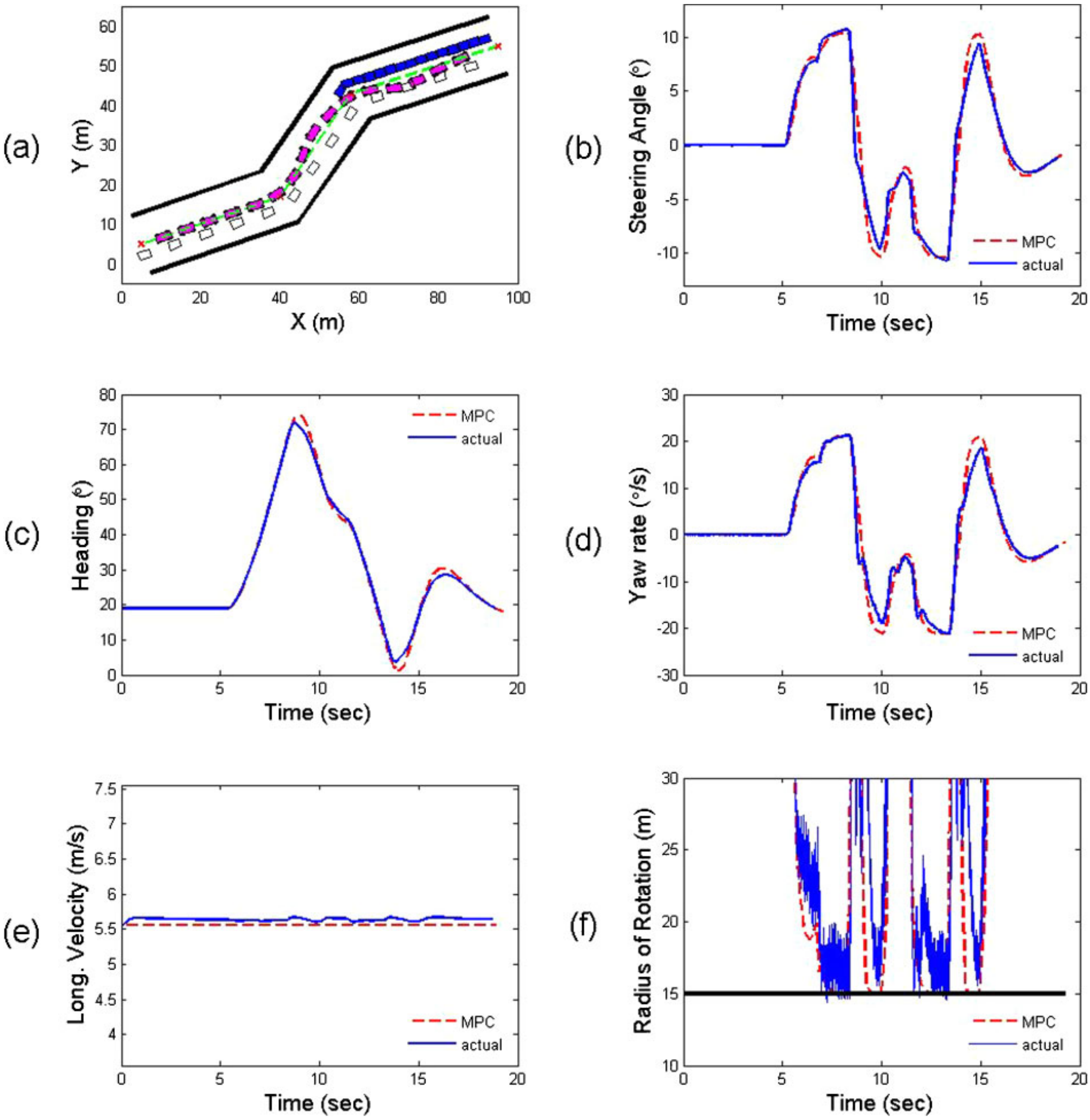

Fig. 15 Simulation results in a dynamic environment at a constant velocity of 20 km/h: (a) resulting trajectory, (b) input history, (c) heading history, (d) yaw rate history, (e) longitudinal velocity history, and (f) instantaneous turning radius along the trajectory

Table 3 Parameters used in the dynamic-environment simulation

Parameter	Value
Horizon length(N)	40
Sampling time (Δt)	0.05 s
Control command limit	$-20^\circ\,s^{-1} \leqslant u \leqslant 20^\circ\,s^{-1}$
Minimum turning radius	15 m
Vehicle specification	$m = 1723.8$ kg, $I_z = 4175$ kg·m^2
Vehicle dimension	(L) 4 m × (W) 1.988 m
Axle length	$l_f = 1.232$ m, $l_r = 1.468$ m
Cornering stiffness	$C_f = 66\,900$ N/rad, $C_r = 62\,700$ N/rad

limit is used in the dynamic environment simulation, due to the need of faster response to avoid collision with moving obstacles. Also, a larger sensing range of 30 m is used for the same reason. In Fig. 15(a), the white rectangle represents the faster moving vehicle, and the grey rectangle represents the slower moving vehicle. Both vehicles move in the south-west direction starting from the upper-right corner of the figure. It can be seen that collision with oncoming obstacles is avoided around [55 m 30 m] and [70 m 45 m]. Figures 15(b) to (f) show the history of the vehicle states during the simulation. Similar to the results from the static-environment simulation, it can be seen that the states and input obtained from the MPC closely match the actual states and input.

5.3 Static environment using a velocity-varying model

In this simulation, the proposed control scheme was tested using a velocity-varying model in the model predictive framework. The inputs of the velocity-varying model are the longitudinal and yaw acceleration, the outputs are the longitudinal velocity, heading angle, yaw rate, and the position of the vehicle in the inertial frame. The constraints on the lateral acceleration, yaw acceleration, and instantaneous turning radius are considered with the model. Compared with the constant-velocity bicycle model, the velocity-varying model has an additional degree of freedom in the longitudinal direction of the vehicle. To make this additional degree of freedom effective, a potential-like cost term penalizing the distance to waypoints was added into the cost function. This potential-like cost term acts as a pulling mechanism to attract the vehicle to the

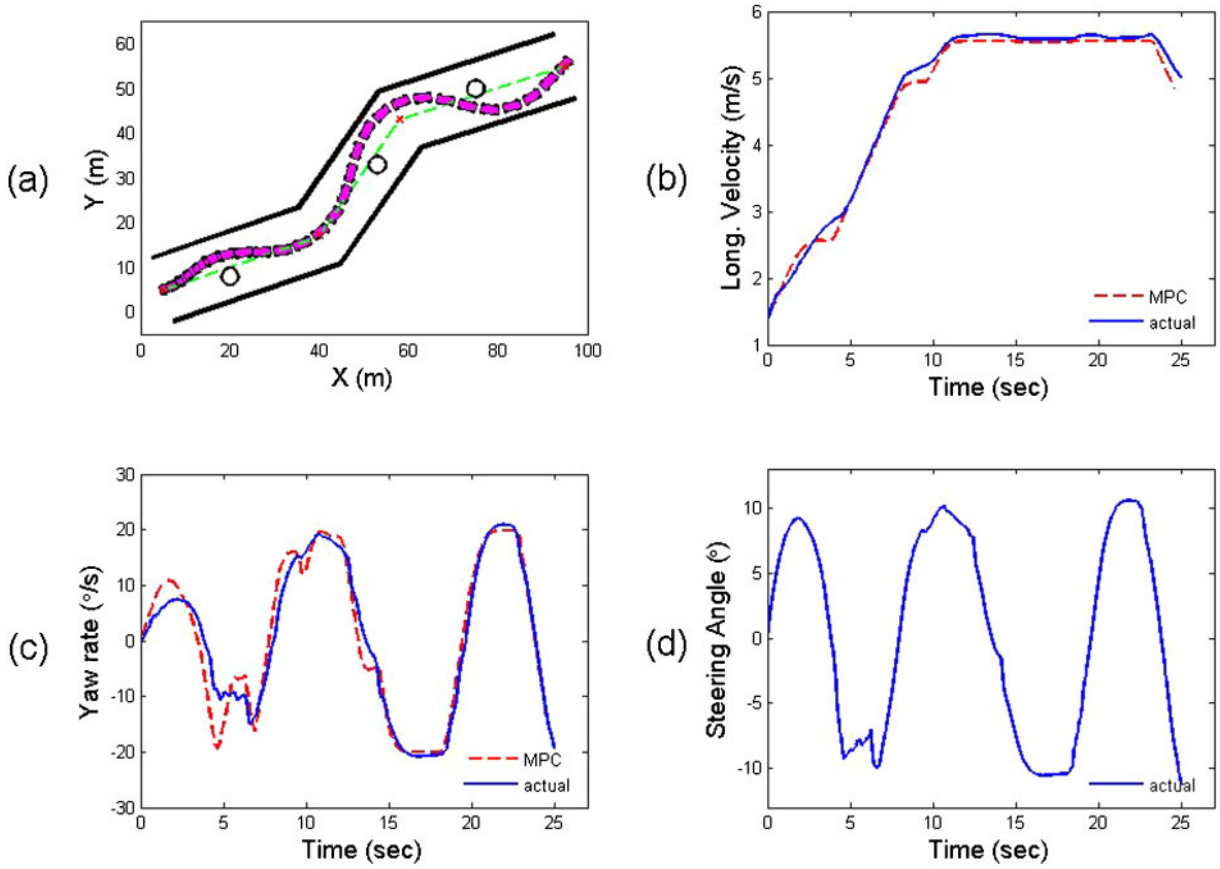

Fig. 16 Simulation results using a velocity-varying model: (a) resulting trajectory, (b) longitudinal velocity history, (c) yaw rate history, and (d) steering input history

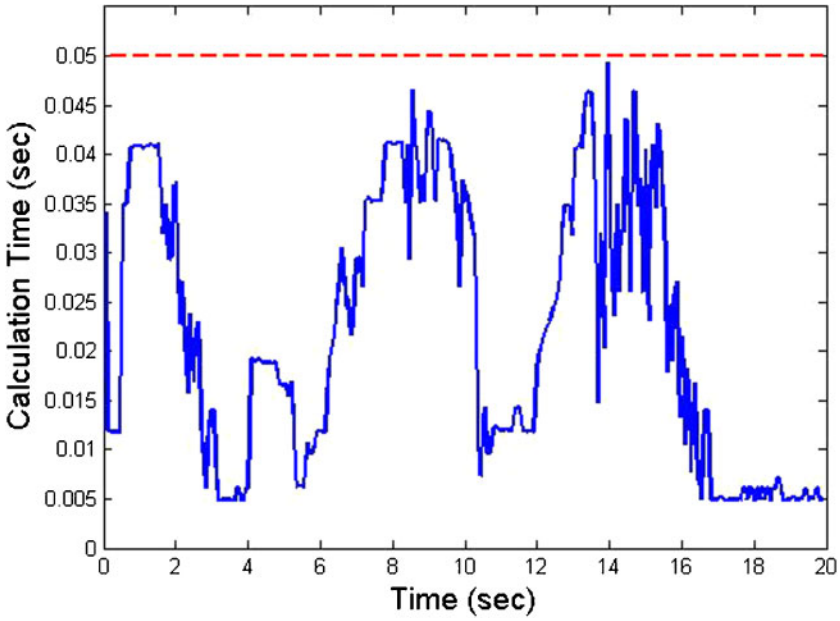

Fig. 17 Calculation time of the MPC-based trajectory generation at each time step in a typical simulation. The dashed line represents the sampling time, which was set to be 0.05 s. It is interesting to observe that the calculation time is relatively high when the rate of change in the control input is large

destination. This was not needed in the bicycle model, because the vehicle always moved with a constant velocity. The velocity reference generated was passed on to the longitudinal controller to track. Figure 16 shows the resulting trajectory and state histories of the vehicle. It should be noted that there is no predicted steering input of the MPC in Fig. 16(d), in contrast to the two previous simulations, because the steering input is not considered in the velocity-varying model.

5.4 Comment on calculation time of the MPC

Figure 17 shows the calculation time of the MPC-based trajectory generation at each time step in a typical simulation. The dashed line represents the sampling time, which was set to be 0.05 s for all simulations. All calculations were carried out within the sampling time, showing that the overall controller is applicable online.

6 CONCLUSIONS

An obstacle avoidance algorithm for ground vehicles based on MPC was presented. The MPC-based trajectory generation was separated from the tracking controller due to the difficulties in calculation of optimal solutions using complex non-linear models. In the MPC framework, a local obstacle threat was incorporated using the parallax-based method. The trajectories were generated so that they were compatible with the vehicle dynamics. The obstacle avoiding trajectories were calculated using the steepest gradient descent method. Control inputs for tracking were obtained using the longitudinal PID controller, which used inverse dynamics of the vehicle powertrain model and the lateral LQR controller which used a similar bicycle model used in the MPC framework. The actual plant used in the simulations was the CarSim vehicle model. Simulation results show that the generated trajectories are compatible with the full non-linear vehicle dynamics. Furthermore, satisfactory performance in terms of collision avoidance was shown in both static and dynamic environments.

ACKNOWLEDGEMENTS

This research was supported by the Basic Science Research Program for Engineering Research Centre (ERC), through the National Research Foundation (NRF) of Korea, funded by the Ministry of Education, Science and Technology, and partially supported by the Institute of Advanced Aerospace Technology at Seoul National University, Republic of Korea.

REFERENCES

1 **Kou, Y. S., Peng, H.,** and **Jung, D. H.** Development of an integrated chassis control system for worst-case studies. In Proceedings of the Eighth International Symposium on *Advanced vehicle control*, Taipei, Republic of China, 20–24 August 2006, pp. 47–52 (JSAE, Tokyo, Japan).

2 **Cho, W. K., Yoon, J. Y., Kim, J. T., Hur, J. W.,** and **Yi, K. S.** An investigation into unified chassis control scheme for optimized vehicle stability and maneuverability. *Veh. System Dyn.*, 2008, **46**(1), 87–105.

3 **Zheng, S., Tang, H., Han, Z.,** and **Zhang, Y.** Controller design for vehicle stability enhancement. *Control Engng Pract.*, 2009, **14**(12), 1413–1421.

4 **Rieth, P. E.** and **Schwarz, R.** ESC II - ESC with active steering intervention. SAE technical paper 2004-01-0260, 2004.

5 **Chung, T.** and **Yi, K. S.** An investigation into differential braking strategies on a bank road for vehicle stability control. *Proc. IMechE, Part D: J. Automobile Engineering*, 2007, **221**, 443–455. DOI: 10.1243/09544070JAUTO77.

6 **Moon, S. W., Moon, I. K.,** and **Yi, K. S.** Design, tuning and evaluation of a full-range adaptive cruise control system with collision avoidance. *Control Engng Pract.*, 2009, **17**(4), 442–455.

7 **Vahidi, A.** and **Eskandarian, A.** Research advances in intelligent collision avoidance and adaptive cruise control. *IEEE Trans. Intell. Trans. System*, 2003, **4**(3), 143–153.

8 **Fancher, P., Bareket, Z.,** and **Erivin, R.** Human-centered design of an ACC-with-braking and forward-crash-warning system. In Proceedings of the Fifth International Symposium on *Advanced vehicle control*, Ann Arbor, Michigan, USA, 22–24 August 2000 (JSAE, Tokyo, Japan).

9 **Wang, J.** and **Rajamani, R.** Should adaptive cruise control systems be designed to maintain a constant time gap between vehicles. *IEEE Trans. Veh. Technol.*, 2004, **53**(5), 1480–1490.

10 **Borelli, F., Falcone, P., Keviczky, T., Asgari, J.,** and **Hrovat, D.** MPC-based approach to active steering for autonomous vehicle systems. *Int. J. Veh. Auton. System*, 2005, **3**, 265–291.

11 **Falcone, P., Borrelli, F., Asgari, J., Tseng, H. E.,** and **Hrovat, D.** A real-time model predictive control approach for autonomous active steering. In Proceedings of the IFAC Workshop on *NMPC for fast systems*, Grenoble, France, 2006, pp. 59–64 (IFAC, Laxenburg, Austria).

12 **Falcone, P., Borrelli, F., Asgari, J., Tseng, H. E.,** and **Hrovat, D.** A model predictive control approach for combined braking and steering in autonomous vehicles. In Proceedings of the 15th Mediterranean Conference on Control and Automation, 2007.

13 **Connolly, C. I., Burns, J. B.,** and **Weiss, R.** Path planning using Laplace's equation. In Proceedings of the IEEE International Conference on *Robotics and automation*, Cincinnati, Ohio, 1990, pp. 2102–2106 (IEEE, Piscataway, New Jersey).

14 **Kim, J. O.** and **Khosla, P.** Real-time obstacle avoidance using harmonic potential functions. In Proceedings of the IEEE International Conference on *Robotics and automation*, Sacramento, California, 1991, pp. 790–796 (IEEE, Piscataway, New Jersey).

15 **Sundar, S.** and **Shiller, Z.** Near-time optimal path planning using potential functions. In Proceedings of the IEEE/RSJ International Workshop on *Intelligent robotics and systems (IROS)*, Osaka, Japan, 1991, pp. 1269–1274 (IEEE, Piscataway, New Jersey, USA).

16 **Al-Jumily, A.** and **Leoung, C.** Wavefront propagation and fuzzy-based autonomous navigation. *Int. J. Adv. Robot. Systems*, 2005, **2**(2), 93–102.

17 **Kim, H. J., Shim, D. H.,** and **Sastry, S.** Nonlinear model predictive control for rotorcraft-base unmanned aerial vehicles. In Proceedings of the *American control conference*, Anchorage, Alaska, 2002, pp. 3576–3581 (IEEE, Piscataway, New Jersey, USA).

18 **Kim, H. J.** and **Shim, D. H.** A flight control system for aerial robots: algorithms and experiments. *Control Engng Pract.*, 2003, **11**(12), 1389–1400.

19 **Fahimi, F.** Non-linear model predicitve formation control for groups of autonomous surface vessels. *Int. J. Control*, 2007, **80**(8), 1248–1259.

20 **Yoon, Y. S., Choe, T. S., Park, Y. W.,** and **Kim, H. J.** Obstacle avoidance for wheeled robots in unknown environments using model predictive control. In Proceedings of the IFAC World Congress, Seoul, Korea, 2008.

21 **Kang, J. Y.** *Development of the human driver model based on the human factor*. MS Thesis, Seoul National University, 2007.

22 **Bertsekas, D.** *Nonlinear programming*, 1999 (Athena Scientific, Nashua, New Hampshire, USA).

APPENDIX

Notation

F	force acting on the vehicle
g	gravitational constant
I_z	moment of inertia
l_f, l_r	distance of front, rear wheels from the centre of gravity
m	vehicle mass
x_{obs}, y_{obs}	position of obstacles in the vehicle frame
X_{obs}, Y_{obs}	position of obstacles in the inertial frame
X, Y	position of vehicle in the inertial frame
α	slip angle of tyres
β	side slip angle of vehicle

δ_f	steering angle of the front tyres
v	vehicle velocity
ρ	turning radius of vehicle
ψ	heading angle of vehicle
$(\cdot)_{\mathrm{f}\cdot}$, $(\cdot)_{\mathrm{r}\cdot}$, $(\cdot)_{\cdot\mathrm{r}}$, $(\cdot)_{\cdot\mathrm{l}}$	front, rear, right, left wheel
$(\cdot)_x$, $(\cdot)_y$	longitudinal, lateral direction in the vehicle frame