To appear in *Vehicle System Dynamics*
Vol. 00, No. 00, Month 20XX, 1–17

Unified decision making and control for highway collision avoidance using active front steer and individual wheel torque control

Hongliang Yuan[a], Xuewei Sun[a] and Timothy Gordon[b*]

[a] *Dept. of Control Science and Engineering, Tongji University, Shanghai, China;*
[b] *School of Engineering, University of Lincoln, Lincoln, LN6 7TS, UK.*

(*v4.0 released October 2014*)

Collision avoidance is a crucial function for all ground vehicles, and using integrated chassis systems to support the driver presents a growing opportunity in active safety. With actuators such as in-wheel electric motors, active front steer and individual wheel brake control, there is an opportunity to develop integrated chassis systems that fully support the driver in safety critical situations. Here we consider the scenario of an impending frontal collision with a stationary or slower moving vehicle in the same driving lane. Traditionally, researchers have approached the required collision avoidance maneuver as a hierarchical scheme, which separates the decision making, path planning and path tracking. In this context a key decision is whether to perform straight-line braking, or steer to change lanes, or indeed perform combined braking and steering. This paper approaches the collision avoidance directly from the perspective of constrained dynamic optimization, using a single optimization procedure to cover these aspects within a single online optimization scheme of model predictive control (MPC). While the new approach is demonstrated in the context of a fully autonomous safety system, it is expected that the same approach can incorporate driver inputs as additional constraints, yielding a flexible and coherent driver assistance system.

Keywords: Collision avoidance; model predictive control; active front steering; torque vectoring

1. Introduction

Recently the development of Advanced Driver Assistance System (ADAS) has received much attention due to the continuing high number of accidents registered in road traffic statistics. Over the years, many car companies have successively equipped their vehicles with the most up-to-date driving assistance systems, from the earliest anti-lock braking systems (ABS), to the more recent electronic stability control (ESC) system, to adaptive cruise control (ACC) system, and to the latest collision avoidance and crash mitigation systems. One of the most important applications of ADAS is the collision avoidance system for autonomous road vehicles, for example, Volvo City Safety equipped cars have shown the effectiveness of helping the driver to avoid rear-end collisions at low speeds by automatically braking when a potential collision is detected [1]. Similar functionality can be found in Mercedes-Benzs Pre-Safe Brake and Volkswagens Front Assist and C-ity Emergency Brake [2]. These systems use brakes only to achieve collision avoidance. The availability of multiple actuators, like individual wheel braking, active front wheel steering, active rear wheel steering, etc. enables an agile response from the vehicle, even

[*] Corresponding author. Email: TGordon@lincoln.ac.uk

This work was supported by the High-end Foreign Experts Program of China.

compared to that of the most skilled human driver. In [3], the proposed system helps the driver to steer around the obstacle by overlaying steering torque and braking individual wheels. In [4], an emergency driving support (EDS) algorithm is developed to support the driver to avoid collision using motor driven power steering (MDPS) and electronic stability control (ESC). In [5], the performance of front steer and four-wheel steer vehicles are compared. It is shown that, by adding more actuators, better maneuverability and more agile vehicle response can potentially be achieved.

It is common that in collision avoidance maneuver, a hierarchical control is employed, at the higher level, a collision free path is planned and updated in real-time, at the lower level, a path tracking controller commands the actuators to track the planned path [6, 7]. For examples, in the well-known DARPA challenge, the winning vehicle such as Boss[8] incorporated trajectory generation and trajectory tracking algorithms. In [9], a polynomial path planning algorithm and a tracking control based on model predictive control (MPC) are proposed. In [10], a collision avoidance method with steering control is proposed by generating a trajectory for obstacle avoidance based on application of the velocity potential field. In [3], the optimal collision avoidance path planning was treated as a problem of finding the optimal acceleration as a function of time or path length, an optimal braking and steering control are used for path following. For these two-layer approaches, the trajectory generation often assumed vehicle operation below handling limits, which facilitates using simple planning algorithm. Hence it usually leads to conservative vehicle performance compare to what could be achieved by handling at friction limits. For the tracking control methods in this hierarchical scheme, PID control[11, 12], pure pursuit strategy[13, 14], LQR[15], MPC[16, 17] control have been proposed previously. Several research works have been conducted for vehicle dynamics control with MPC. In [18], a linear time variant MPC approach is presented by using a single track model and steering wheel angle as command signal combined with a solver for quadratic programming. In addition, the longitudinal deceleration is incorporated into the MPC problem for combined braking and steering in [19].

By contrast, in [20], an approach called Modified Hamiltonian Algorithm (MHA) is proposed for direct torque distribution. In this approach, first a global acceleration vector is determined using a particle model; secondly, vehicle stability control is considered by constraining yaw moment to follow a desired yaw moment profile; lastly, control allocation is achieved by minimizing a Hamiltonian function at individual wheel level. Even though this approach avoided using a path planner, it still involves tracking a motion profile from the higher level, hence it can be seen as tracking a velocity profile instead of a position profile when compared to the traditional collision avoidance scheme.

In [21] and [22], path tracking controllers based on MPC are presented to follow desired speeds while allowing to temporary deviate from desired paths to best meet collision avoidance and stabilization criteria. While the bicycle model may pose problems for vehicles handling at friction limits, the underlying idea of optimization with both velocity and position states is quite useful. In this paper, we propose a control scheme that integrates the path planning, and tracking by directly optimize the steering angles and driving/braking torques to achieve collision avoidance. The collision avoidance problem is formulated as an optimal control problem based on full vehicle models by defining a cost function and a collision avoidance constraint on the vehicle position. The underlying idea is to convert the optimal control problem to a convex quadratic programming problem (QP) via MPC control scheme, then the QP problem is solved online to directly yield the steering angle and driving forces [23]. In this way, we can avoid using the path planner and path tracking controller separately, thus it retains the potential to achieve better performance.

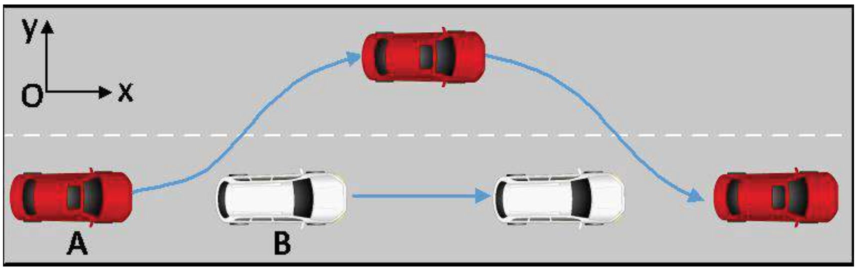

Figure 1. A collision avoidance scenario.

The paper is organized as follows. In Section 2, we define the vehicle model used in the MPC controller, and introduce a corresponding simplified tire model in Section 3. Section 4 presents the formulation of collision avoidance optimization problem and develops the MPC control solutions. In Section 5 we carry out performance evaluation via joint simulations of CarSim and Matlab Simulink, and in Section 6, we draw conclusions of the research.

2. Vehicle model

Consider the scenario illustrated in Figure 1. In this scenario, vehicle A is moving along a road when it detects vehicle B as an obstacle. The obstacle (Vehicle B) can be static or moving. Vehicle A is supposed to perform an evasive manoeuvre to pass vehicle B, and return to the original lane or stay in the new lane (depends on the objective).

The collision avoidance controller is to make use of the 3-degree-of-freedom (DOF) planar two-track vehicle model shown in Figure 2 [24], where v_x, v_y, r are the longitudinal velocity, lateral velocity and yaw rate in the vehicle body axis system. $F_{xfl}, F_{xfr}, F_{xrl}, F_{xrr}$ are the longitudinal forces of front-left, front-right, rear-left, rear-right tires respectively. Similarly, $F_{yfl}, F_{yfr}, F_{yrl}, F_{yrr}$ are the lateral forces of 4 tires and $\alpha_{fl}, \alpha_{fr}, \alpha_{rl}, \alpha_{rr}$ are tire lateral slip angles. Parameters l_f and l_r are the distance from the vehicle CG to the front and rear axle respectively, and l_t is a half track. δ is the front road wheel steering angle, β is the body slip angle, and r is the vehicle yaw rate. The vehicle total mass and yaw moment of inertia are m and I_{zz} respectively. OXY is the world axis system and (X, Y) is the cartesian coordinates, ψ is the yaw angle with respect to the X axis.

The wheel torque controller is assumed to have full authority of driving and braking torques at each wheel and to have access to appropriate sensors. Using these notations, the motion of the vehicle can be described by the following dynamic and kinematic equations:

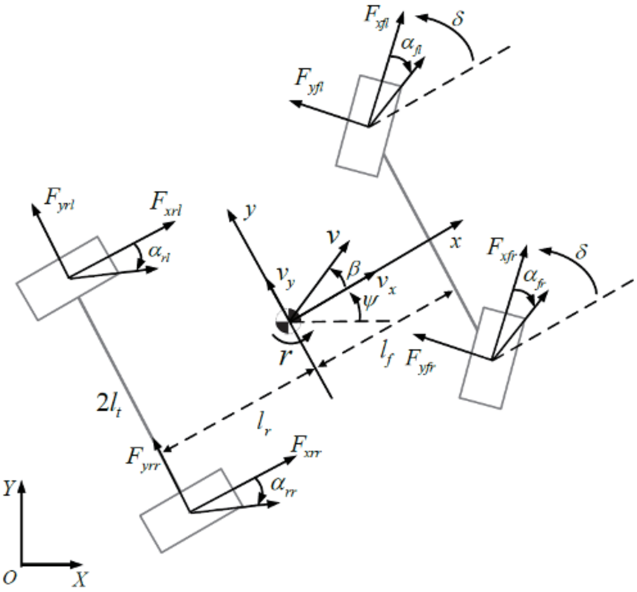

Figure 2. The two-track vehicle model. The arrows show the positive direction of each angle or force [24].

$$
\begin{aligned}
&m(\dot{v}_x - v_y r) = F_{xfl}\cos\delta - F_{yfl}\sin\delta + F_{xfr}\cos\delta - F_{yfr}\sin\delta + F_{xrl} + F_{xrr},\\
&m(\dot{v}_y + v_x r) = F_{xfl}\sin\delta + F_{yfl}\cos\delta + F_{xfr}\sin\delta + F_{yfr}\cos\delta + F_{yrl} + F_{yrr},\\
&I_{zz}\dot{r} = (F_{xfr}\cos\delta - F_{yfr}\sin\delta + F_{xrr} - F_{xfl}\cos\delta + F_{yfl}\sin\delta - F_{xrl})l_t\\
&\qquad + (F_{xfl}\sin\delta + F_{yfl}\cos\delta + F_{xfr}\sin\delta + F_{yfr}\cos\delta)l_f - (F_{yrl} + F_{yrr})l_r, \qquad (1)\\
&\dot{\psi} = r,\\
&\dot{X} = v_x\cos\psi - v_y\sin\psi,\\
&\dot{Y} = v_x\sin\psi + v_y\cos\psi.
\end{aligned}
$$

In the state space model (1), the front axle steering angle δ is treated as a control input and available in real-time. Since the longitudinal and lateral tire forces interact, they cannot both be used as independent control inputs. It is preferred to use longitudinal tire forces as control inputs since they can be directly regulated by driving/braking torques. Hence it is desirable to eliminate the lateral tire forces in the model equations (1), and this can be done by the introduction of a suitable reduced-order tire force model.

3. Tire model

To give an explicit formula for the lateral tire force, a friction ellipse relationship between peak longitudinal and lateral tire forces is assumed, as illustrated in Figure 3 [24]. For each tire, the equation of the friction ellipse is given by:

$$
(\frac{F_{x**}}{F_{x**}^{max}})^2 + (\frac{F_{y**}}{F_{y**}^{max}})^2 = 1, \qquad (2)
$$

Table 1. B, C , D , E at different vertical loads [24]

F_z(N)	1594	3187	4780	6374	7968	9562	11155	12749
B	12.7603	12.5691	12.3569	12.1300	11.8914	11.6432	11.3866	11.1227
C	1.4481	1.4490	1.4499	1.4510	1.4522	1.4535	1.4549	1.4564
D	0.9988	0.9871	0.9741	0.9601	0.9455	0.9302	0.9144	0.8982
E	0.0293e-15	0.0603e-15	0.0387e-15	0.0033e-15	0.1406e-15	0.0347e-15	0.0107e-15	0.0004e-15
LSE(N)	5.0962	10.0173	14.7306	19.2219	23.4742	27.4771	31.2175	34.6881

where the subscript $**$ denotes fl, fr, rl, rr. In the friction ellipse equation (2), the maximum achievable longitudinal force is always $F_{x**}^{max} = \mu F_{z**}$, where μ is the road friction coefficient and F_{z**} is the vertical load of the corresponding tire. The maximum lateral force F_{y**}^{max} is achieved when there is no longitudinal slip, and varies with tire slip angle. It can be expressed via the Pacejka magic formula [25]:

$$F_{y**}(\alpha) = \mu F_{z**} M(\alpha), \tag{3}$$

with

$$M(\alpha) = D \sin\{C \tan^{-1}[B\alpha - E(B\alpha - \tan^{-1} B\alpha)]\},$$

where B, C, D, E are parameters of the magic formula. Substituting equation (3) into equation (2), the lateral forces of combined slip can be expressed by the longitudinal forces:

$$F_{y**} = M(\alpha)\sqrt{(\mu F_{z**})^2 - F_{x**}^2}. \tag{4}$$

Thus equation (4) is used to eliminate the lateral tire forces in model (1). In this study, the magic formula parameters B, C, D, E for the tire model (4) are fitted with the least-squares estimation (LSE) method, using the CarSim tire data from the tire '215/55 R17' [26]. The obtained parameters are shown in Table 1 [24].

According to Table 1, the parameter E is quite small for all vertical loads, so for simplicity we use the approximation $E = 0$. Further analysis shows that parameters B, C, D have certain linearity with respect the vertical loads as shown in Figure 4.

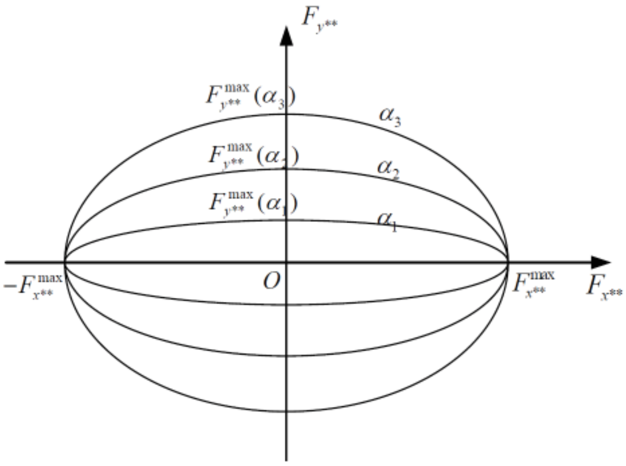

Figure 3. The elliptical relationship between longitudinal and lateral tire forces at different slip angles [24].

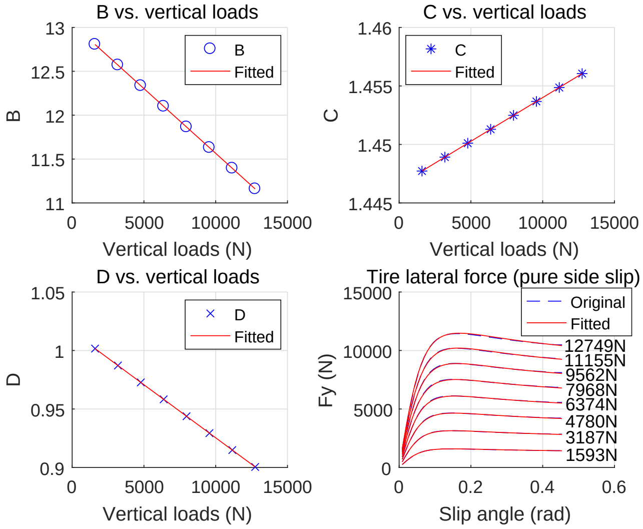

Figure 4. Least-squares estimation fitted results

Therefore, they are further fitted as linear equations:

$$B = -1.4758 \times 10^{-4} F_z + 13.0409,$$
$$C = 7.4666 \times 10^{-7} F_z + 1.4465,$$
$$D = -9.0695 \times 10^{-6} F_z + 1.0161.$$

Figure 4 illustrates the parameter fitting results and the comparison of pure lateral force provided by CarSim tire data and the calculated force from equation (4) at different vertical loads, showing a satisfactory match.

It is also worthwhile testing equation (4) in a combined slip situation, where the longitudinal tire force is nonzero. Here we simulate a double lane change maneuver at 120 km/h on a surface with friction coefficient $\mu = 0.6$. The lateral forces from CarSim are compared with those of the reduced-order model in Figure 5. Again this shows that equation (4) reproduces the lateral forces to a satisfactory level.

The above fitting procedure is equally applicable to an actual vehicle, provided the basic tire characteristics are available, i.e. friction ellipse and pure lateral force curve, both at a representative set of vertical loads.

4. Controller design

4.1. *The optimal control problem for collision avoidance*

In this section, we reformulate the collision avoidance problem into an optimal control problem. For autonomous driving and collision avoidance, suppose we have a reference longitudinal speed v_{xd} and a reference lateral position for the lane position y_d. Moreover, for lateral stability, we want to have the reference lateral speed $v_{yd} = 0$. Then we can establish the following performance index:

$$J = \int_0^t [w_{vx}(v_x - v_{xd})^2 + w_{vy}(v_y - v_{yd})^2 + w_y(Y - y_d)^2]dt,$$

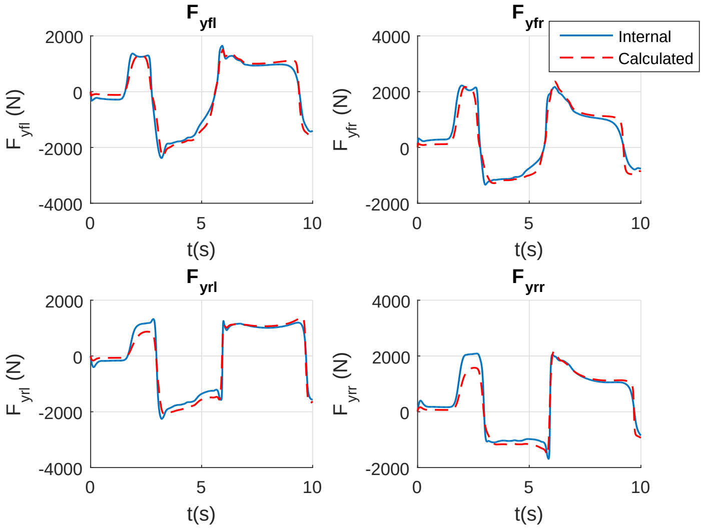

Figure 5. Internal lateral tire force and calculated tire force with equation (4) in a DLC maneuver

where w_{vx}, w_{vy}, w_y are positive weights of tracking errors with respect to v_{xd}, v_{yd}, y_d. Also, we define ρ as a measurement of the distance between the ego vehicle and the obstacle:

$$\rho = (X - x_o)^2/a^2 + (Y - y_o)^2/b^2, \tag{5}$$

where $(x_o\ y_o)$ is the geometric center of obstacles and $(a\ b)$ are the parameters that define an elliptical shape of the obstacle.

Using this performance index J, the optimization problem can be formulated as:

$$\begin{aligned} \min_{|u|\le u_{\max}} \quad & J \\ \text{s.t.} \quad & \dot{\xi} = f(\xi, u), && (6) \\ & \xi(0) = \xi_0, \\ & \rho \ge 1, && (7) \end{aligned}$$

where $\xi = [v_x\ v_y\ r\ \psi\ X\ Y]^T$ is the vehicle state vector, $u = [\delta\ F_{xfl}\ F_{xfr}\ F_{xrl}\ F_{xrr}]^T$ is the input vector, and $u_{\max}$ is the upper limit of the input vector. Equation (6) represents the vehicle model (1), and equation (7) is the geometric constraint for collision avoidance.

4.2. *The MPC formulation*

Theoretically, the constrained nonlinear optimization problem formulated in Section 4.1 can be handled with the Pontryagins minimum principle, which would lead to a two-point-boundary-value-problem (TPBVP). Unless the system is quite simple, only off-line numerical solution can be found, hence it is not very helpful for real-time collision avoidance. On the other hand, the model predictive control (MPC) utilizes the model to be controlled to predict the future evolution of the system over a receding finite-time horizon, and minimizes a predefined cost function with future controls to be solved. It is real-time implementable by seeking a near optimal solution instead of a true one. In what follows, we present the formulation to solve the optimal control problem.

As shown in equation (6), the vehicle model (1) can be written as:

$$\dot{\xi} = f(\xi, u).$$

Additionally, the output equations are defined as:

$$\eta(t) = c \cdot \xi(t),$$

where,

$$c = \begin{bmatrix} 1\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1 \end{bmatrix},$$

i.e. outputs of the system is $\eta(t) = [v_x \; v_y \; Y]^T$. At sampling instant k, suppose the current measured states, inputs and outputs are: ξ^k, u^k and η^k respectively (in the remaining paper, the superscripts or subscripts k is used to denote kth sampling instant). Then at sampling instant k, the linearized system is:

$$\begin{aligned} \dot{\tilde{\xi}} &= A_k \cdot \tilde{\xi} + B_k \cdot \tilde{u}, \\ \tilde{\eta} &= C_k \cdot \tilde{\xi}, \end{aligned} \tag{8}$$

where

$$\tilde{\xi} = \xi - \xi^k, \;\; \tilde{u} = u - u^k, \;\; \tilde{\eta} = \eta - \eta^k,$$

$$A_k = \frac{\partial f}{\partial \xi}\Big|_{(\xi^k, u^k)} = \begin{bmatrix} 0 & r^k & v_y^k & 0 & 0\,0 \\ -r^k & 0 & -v_x^k & 0 & 0\,0 \\ 0 & 0 & 0 & 0 & 0\,0 \\ 0 & 0 & 1 & 0 & 0\,0 \\ \cos\psi^k & -\sin\psi^k & 0 & -v_x^k \sin\psi^k - v_y^k \cos\psi^k & 0\,0 \\ \sin\psi^k & \cos\psi^k & 0 & v_x^k \cos\psi^k - v_y^k \sin\psi^k & 0\,0 \end{bmatrix},$$

$$C_k = \begin{bmatrix} 1\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1 \end{bmatrix}, \quad B_k = \frac{\partial f}{\partial u}\Big|_{(\xi^k, u^k)} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \text{with}$$

$$
\begin{aligned}
b_{11} &= \frac{\partial f_1}{\partial \delta}\Big|_{(\xi^k,u^k)} = -F^k_{xfl}\sin\delta^k - F^k_{yfl}\cos\delta^k - F^k_{xfr}\sin\delta^k - F^k_{yfr}\cos\delta^k,\\
b_{12} &= \frac{\partial f_1}{\partial F_{xfl}}\Big|_{(\xi^k,u^k)} = \frac{\cos\delta^k}{m} + \frac{M(\alpha_{fl})F^k_{xfl}\sin\delta^k}{m\sqrt{(\mu F^k_{zfl})^2 - (F^k_{xfl})^2}},\\
b_{13} &= \frac{\partial f_1}{\partial F_{xfr}}\Big|_{(\xi^k,u^k)} = \frac{\cos\delta^k}{m} + \frac{M(\alpha_{fr})F^k_{xfr}\sin\delta^k}{m\sqrt{(\mu F^k_{zfr})^2 - (F^k_{xfr})^2}},\\
b_{14} &= \frac{\partial f_1}{\partial F_{xrl}}\Big|_{(\xi^k,u^k)} = \frac{1}{m},\\
b_{15} &= \frac{\partial f_1}{\partial F_{xrr}}\Big|_{(\xi^k,u^k)} = \frac{1}{m}.
\end{aligned}
$$

$$
\begin{aligned}
b_{21} &= \frac{\partial f_2}{\partial \delta}\Big|_{(\xi^k,u^k)} = F^k_{xfl}\cos\delta^k - F^k_{yfl}\sin\delta^k + F^k_{xfr}\cos\delta^k - F^k_{yfr}\sin\delta^k,\\
b_{22} &= \frac{\partial f_2}{\partial F_{xfl}}\Big|_{(\xi^k,u^k)} = \frac{\sin\delta^k}{m} - \frac{M(\alpha_{fl})F^k_{xfl}\cos\delta^k}{m\sqrt{(\mu F^k_{zfl})^2 - (F^k_{xfl})^2}},\\
b_{23} &= \frac{\partial f_2}{\partial F_{xfr}}\Big|_{(\xi^k,u^k)} = \frac{\sin\delta^k}{m} - \frac{M(\alpha_{fr})F^k_{xfr}\cos\delta^k}{m\sqrt{(\mu F^k_{zfr})^2 - (F^k_{xfr})^2}},\\
b_{24} &= \frac{\partial f_2}{\partial F_{xrl}}\Big|_{(\xi^k,u^k)} = -\frac{M(\alpha_{rl})F^k_{xrl}}{m\sqrt{(\mu F^k_{zrl})^2 - (F^k_{xrl})^2}},\\
b_{25} &= \frac{\partial f_2}{\partial F_{xrr}}\Big|_{(\xi^k,u^k)} = -\frac{M(\alpha_{rr})F^k_{xrr}}{m\sqrt{(\mu F^k_{zrr})^2 - (F^k_{xrr})^2}}.
\end{aligned}
$$

$$
\begin{aligned}
b_{31} &= \frac{\partial f_3}{\partial \delta}\Big|_{(\xi^k,u^k)} = (F^k_{xfl}\sin\delta^k + F^k_{yfl}\cos\delta^k - F_{xfr}\sin\delta^k - F_{yfr}\cos\delta^k)l_t\\
&\qquad +(F_{xfl}\cos\delta^k - F_{yfl}\sin\delta^k + F^k_{xfr}\cos\delta^k - F^k_{yfr}\sin\delta^k)l_f,\\
b_{32} &= \frac{\partial f_3}{\partial F_{xfl}}\Big|_{(\xi^k,u^k)} = \frac{l_f\sin\delta^k}{I_{zz}} - \frac{l_f M(\alpha_{fl})F^k_{xfl}\cos\delta^k}{I_{zz}\sqrt{(\mu F^k_{zfl})^2 - (F^k_{xfl})^2}} - \frac{l_t\cos\delta^k}{I_{zz}} - \frac{l_t M(\alpha_{fl})F^k_{xfl}\sin\delta^k}{I_{zz}\sqrt{(\mu F^k_{zfl})^2 - (F^k_{xfl})^2}},\\
b_{33} &= \frac{\partial f_3}{\partial F_{xfr}}\Big|_{(\xi^k,u^k)} = \frac{l_f\sin\delta^k}{I_{zz}} - \frac{l_f M(\alpha_{fr})F^k_{xfr}\cos\delta^k}{I_{zz}\sqrt{(\mu F^k_{zfr})^2 - (F^k_{xfr})^2}} + \frac{l_t\cos\delta^k}{I_{zz}} + \frac{l_t M(\alpha_{fr})F^k_{xfr}\sin\delta^k}{I_{zz}\sqrt{(\mu F^k_{zfr})^2 - (F^k_{xfr})^2}},\\
b_{34} &= \frac{\partial f_3}{\partial F_{xrl}}\Big|_{(\xi^k,u^k)} = -\frac{l_t}{I_{zz}} + \frac{l_r M(\alpha_{rl})F^k_{xrl}}{I_{zz}\sqrt{(\mu F^k_{zrl})^2 - (F^k_{xrl})^2}},\\
b_{35} &= \frac{\partial f_3}{\partial F_{xrr}}\Big|_{(\xi^k,u^k)} = \frac{l_t}{I_{zz}} + \frac{l_r M(\alpha_{rr})F^k_{xrr}}{I_{zz}\sqrt{(\mu F^k_{zrr})^2 - (F^k_{xrr})^2}}.
\end{aligned}
$$

The LTV-MPC is a discrete-time strategy, so the model used for controller synthesis should be discretised. There are several methods can be used for this purpose such as the Euler or trapezoidal method (in this work the Matlab function 'c2d' was used, and the Euler method was selected). Therefore the linearized model (8) is used in discrete-time form:

$$\begin{aligned}\tilde{\xi}(k+1) &= A_k^d \cdot \tilde{\xi}(k) + B_k^d \cdot \tilde{u}(k),\\ \tilde{\eta}(k) &= C_k^d \cdot \tilde{\xi}(k), \end{aligned} \tag{9}$$

where A_k^d, B_k^d, C_k^d are system matrices of the discrete system.

The fundamental basis of the MPC formulation is to use the prediction model (9) to calculate the future states of the dynamic system over a fixed *prediction horizon* N_p. At a given time k, from the predictions at instants $k, k+1, \cdots, k+N_p$, a cost function is evaluated and minimised under given constraints by solving an optimal control problem with respect to the inputs[27]. At time $k+1$, the same problem is solved, on the receding horizon $k+1, k+2, \cdots, k+N_p+1$ and so on.

Generally, the cost function to be minimised takes the following form [27]:

$$J = \sum_{i=1}^{N_p} ||\tilde{\eta}(k+i) - \tilde{\eta}_{ref}(k+i)||_Q^2 + \sum_{j=0}^{N_c-1} ||\Delta\tilde{u}(k+j)||_R^2, \tag{10}$$

where $\tilde{\eta}(k+i)$ is the (predicted) output for the linearized system and $\tilde{\eta}_{ref}(k+i)$ is the output reference for the linearized system. N_c is the *control horizon*, $N_c \leq N_p$, which defines the dimension of the optimisation problem. The weighting matrices Q and R, respectively represent the weights associated with the tracking errors and the control input energy; usually they are in diagonal form, i.e. $Q = \text{diag}\ [q_{v_x},\ q_{v_y},\ q_r,\ q_\psi,\ q_X,\ q_Y]$ and $R = \text{diag}\ [r_\delta,\ r_{fl},\ r_{fr},\ r_{rl},\ r_{rr}]$. The parameters N_p, N_c, and weighting matrices Q and R together determine the performance of the MPC control system. Since it is based on the linearized system, at sampling instant k the following equations hold:

$$\begin{aligned}\tilde{\eta}_{ref}(k+i) &= [v_{xd}\ v_{yd}\ y_d]^T - \eta^k,\ \ \forall i = 1, 2, \cdots, N_p.\\ \Delta\tilde{u}(k+j) &= \tilde{u}(k+j) - \tilde{u}(k+j-1),\ \ \forall j = 0, 1, \cdots, N_c - 1.\end{aligned}$$

Denoting the optimisation vector to be:

$$\Delta U = \left[\Delta\tilde{u}(k)^T\ \ \Delta\tilde{u}(k+1)^T\ \ \cdots\ \ \Delta\tilde{u}(k+N_c-1)^T\right]^T,$$

the optimal control problem is formulated as follows:

$$\arg\min_{\Delta U}\ J(\tilde{\xi}(k), \Delta U), \tag{11}$$

$$\text{s.t. }\ \tilde{\xi}(k+1) = A_k^d \cdot \tilde{\xi}(k) + B_k^d \cdot \tilde{u}(k), \tag{12}$$

$$\tilde{\eta}(k) = C_k^d \cdot \tilde{\xi}(k), \tag{13}$$

$$u^{\min} \leq u(i) \leq u^{\max},\ \forall i = k, k+1, \cdots, k+N_c-1, \tag{14}$$

$$\rho(i) \geq 1,\ \forall i = k, k+1, \cdots, k+N_c-1, \tag{15}$$

where $u^{\min}$ and $u^{\max}$ are vectors of minimum and maximum longitudinal forces, equivalent to the upper limits on braking torque and driving torque respectively. The vehicle position $(X,\ Y)$ is also to be constrained, with constraint (15).

In this formulation, equation (11) is the performance index to be minimised, and (12) and (13) are the state and output equations respectively. Condition (14) represents the constraint on driving and braking inputs, while (15) is the newly introduced geometric constraint imposed on vehicle position. Constraint (15) is the crucial component for collision avoidance, and in the following it will be shown to act as a further input constraint.

4.3. *Constraints in canonical form*

The MPC solution procedure translates the optimal control problem described by equations (11) – (15) into a quadratic programming (QP) problem w.r.t. the optimisation vector ΔU, from which an optimal control sequence $\Delta\tilde{u}(k+i),\ i = 0, 2, \cdots, N_c - 1$ can be solved. By the receding horizon principle, only the first control is applied at the current step. In the next sampling instant, the whole procedure is repeated. While the QP procedure is quite standard, a key step is to convert the constraints (14) and (15) into the canonical (standard) format used in QP:

$$G \cdot \Delta U \leq \gamma, \tag{16}$$

where G is the coefficient matrix with proper dimensions and γ is a column vector of upper bounds.

4.3.1. Input saturation constraints

Constraint (14) is an input saturation constraint, for which the following matrix equation holds:

$$\begin{bmatrix} \tilde{u}(k) \\ \tilde{u}(k+1) \\ \tilde{u}(k+2) \\ \vdots \\ \tilde{u}(k+N_c-1) \end{bmatrix} = \begin{bmatrix} I_5 \\ I_5 \\ I_5 \\ \vdots \\ I_5 \end{bmatrix} \tilde{u}(k-1) + \begin{bmatrix} I_5 & 0 & 0 & \cdots & 0 \\ I_5 & I_5 & 0 & \cdots & 0 \\ I_5 & I_5 & I_5 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I_5 & I_5 & I_5 & \cdots & I_5 \end{bmatrix} \begin{bmatrix} \Delta\tilde{u}(k) \\ \Delta\tilde{u}(k+1) \\ \Delta\tilde{u}(k+2) \\ \vdots \\ \Delta\tilde{u}(k+N_c-1) \end{bmatrix}, \tag{17}$$

where I_n is the $n \times n$ identity matrix. Denoting

$$N_1 = \begin{bmatrix} I_5 \\ I_5 \\ I_5 \\ \vdots \\ I_5 \end{bmatrix}_{5N_c \times 5}, \quad N_2 = \begin{bmatrix} I_5 & 0 & 0 & \cdots & 0 \\ I_5 & I_5 & 0 & \cdots & 0 \\ I_5 & I_5 & I_5 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I_5 & I_5 & I_5 & \cdots & I_5 \end{bmatrix}_{5N_c \times 5N_c},$$

the constraint becomes:

$$U^{\min} - U^k \leq N_1 \tilde{u}(k-1) + N_2 \Delta U \leq U^{\max} - U^k,$$

where

$$U^{\min} = \begin{bmatrix} u^{\min} \\ u^{\min} \\ u^{\min} \\ \vdots \\ u^{\min} \end{bmatrix}_{5N_c \times 1}, \quad U^{\max} = \begin{bmatrix} u^{\max} \\ u^{\max} \\ u^{\max} \\ \vdots \\ u^{\max} \end{bmatrix}_{5N_c \times 1}, \quad U^k = \begin{bmatrix} u^k \\ u^k \\ u^k \\ \vdots \\ u^k \end{bmatrix}_{5N_c \times 1}.$$

Note that the system is linearized at every sampling instant, so we always have:

$$\tilde{u}(k-1) = [0\ 0\ 0\ 0\ 0]^T,$$

It follows that the constraint (14) in canonical form can be imposed as:

$$\begin{bmatrix} -N_2 \\ N_2 \end{bmatrix} \Delta U \leq \begin{bmatrix} -(U^{\min} - U^k) \\ U^{\max} - U^k \end{bmatrix}.$$

4.3.2. The collision avoidance constraint

At sampling instant k, using the distance measurement ρ defined in equation (5), the collision avoidance constraint (15) can be linearized as:

$$\begin{aligned} \rho &= \rho^k + \frac{\partial \rho}{\partial X}\Big|_{(\xi^k, u^k)} (X - X^k) + \frac{\partial \rho}{\partial Y}\Big|_{(\xi^k, u^k)} (Y - Y^k), \\ &\triangleq \rho^k + p_1 \cdot \tilde{X} + p_2 \cdot \tilde{Y}, \end{aligned}$$

where

$$p_1 = \frac{\partial \rho}{\partial X}\Big|_{(\xi^k, u^k)}, \ \ p_2 = \frac{\partial \rho}{\partial Y}\Big|_{(\xi^k, u^k)}.$$

Therefore, collision avoidance constraint (15) becomes:

$$p_1 \cdot \tilde{X}(k+i) + p_2 \cdot \tilde{Y}(k+i) \geq 1 - \rho^k, \ \ \forall i = 1, 2, \ \cdots, \ N_p.$$

Hence, denoting $P = [0\ 0\ 0\ 0\ p_1\ p_2]$, the constraint equation (15) becomes:

$$P \cdot \tilde{\xi}(k+i) \geq 1 - \rho^k, \ \ \forall i = 1, 2, \ \cdots, \ N_p.$$

This constraint should hold in the whole prediction horizon for each sampling instant. From the discrete state equation, we can obtain the following future states:

$$\begin{aligned} \tilde{\xi}(k+1) &= A_k^d \tilde{\xi}(k) + B_k^d \tilde{u}(k), \\ \tilde{\xi}(k+2) &= (A_k^d)^2 \tilde{\xi}(k) + A_k^d B_k^d \tilde{u}(k) + B_k^d \tilde{u}(k+1), \\ &\vdots \\ \tilde{\xi}(k+N_p) &= (A_k^d)^{N_p} \tilde{\xi}(k) + (A_k^d)^{N_p - 1} B_k^d \tilde{u}(k) + (A_k^d)^{N_p - 2} B_k^d \tilde{u}(k+1) + \cdots \\ &\quad + (A_k^d)^{N_p - N_c} B_k^d \tilde{u}(k + N_c - 1). \end{aligned}$$

Since the linearization takes place in at each sampling instant, we have $\tilde{\xi}(k) = 0$, hence the above equations can be reorganized into a matrix form:

$$\begin{bmatrix} \tilde{\xi}(k+1) \\ \tilde{\xi}(k+2) \\ \vdots \\ \tilde{\xi}(k+N_p) \end{bmatrix} = W \begin{bmatrix} \tilde{u}(k) \\ \tilde{u}(k+1) \\ \vdots \\ \tilde{u}(k+N_c-1) \end{bmatrix},$$

where,

$$W = \begin{bmatrix} B_k^d & 0 & 0 & \cdots & 0 \\ A_k^d B_k^d & B_k^d & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (A_k^d)^{N_p-1} B_k^d & (A_k^d)^{N_p-2} B_k^d & (A_k^d)^{N_p-3} B_k^d & \cdots & (A_k^d)^{N_p-N_c} B_k^d \end{bmatrix}_{6N_p \times 4N_c}.$$

Then, by using the relationship in equation (17), the collision avoidance constraint can be further convert to:

$$I_{N_p} \otimes PW[N_1 \cdot \tilde{u}(k-1) + N_2 \cdot \Delta U] \geq N_3,$$

where $\otimes$ denotes the Kronecker product, and

$$N_3 = \begin{bmatrix} 1-\rho^k \\ 1-\rho^k \\ \vdots \\ 1-\rho^k \end{bmatrix}_{N_p \times 1}.$$

It follows that the collision avoidance constraint can be given in canonical form as:

$$-I_{N_p} \otimes PWN_2 \cdot \Delta U \leq -N_3.$$

5. Performance evaluation

To evaluate the performance of the proposed collision avoidance scheme based on MPC, the algorithm was implemented and tested using a joint platform of Matlab-Simulink in co-simulation with the commercial vehicle simulation package CarSim 8.1 (Mechanical Simulation Corporation 2011). The vehicle used in the simulation is a D-class sedan passenger car, with the parameters described in Table 2. The joint simulation platform of CarSim and Simulink is illustrated in Figure 6. Note that in the simulation we use wheel torque input T_{**} instead of driving/braking forces as it can be directly generated by CarSim, and since the wheel spin dynamics are much faster than the chassis motion dynamics, we use $T_{**} = F_{x**} \cdot R_t$ to approximate the driving/braking torque.

The vehicle starts at $(0, 0)$, and moves along the horizontal axis. The obstacle is placed at $(25, 0)$, with radius 2 meters and is static. The control parameters are listed in Table 3.

Simulation results of the MPC control are illustrated in Figure 7 to Figure 10. Figure 7 shows the graphic outputs of CarSim, the red vehicle is implemented with the proposed

Table 2. Vehicle parameters

Description	Symbol	Value
Sprung mass	m_s	1370(kg)
Unsprung mass	m_u	80 × 2(kg)
Yaw moment of inertia	I_{zz}	2315.3(Nm)
Wheelbase	L	2.78(m)
Distance to CG from front axle	l_f	1.11(m)
Distance to CG from rear axle	l_r	1.67(m)
Half track	l_t	0.775(m)
Tire effective rolling radius	R_t	0.325(m)
Road friction coefficients	μ	0.85

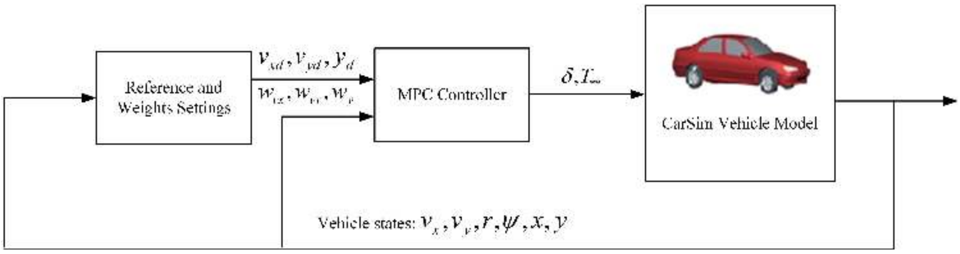

Figure 6. Simulink-CarSim joint simulation platform

Table 3. Control parameters

Description	Symbol	Value
Desired longitudinal velocity	v_{xd}	72(km/h)
Desired lateral velocity	v_{yd}	0(km/h)
Desired lateral offset	y_d	0(m)
Longitudinal velocity weight	w_{vx}	1
Lateral velocity weight	w_{vy}	1
Lateral offset weight	w_y	10
Prediction horizon	N_p	20
Control horizon	N_c	15
Sampling period	T_s	0.05(sec)

collision avoidance strategy, the green vehicle is the obstacle. It shows the red vehicle successfully steers around the obstacle and returns to its original lane as expected. Figure 8 illustrates the vehicle states, it shows the longitudinal speed has small fluctuations around the reference speed during the maneuver, and the lateral velocity and yaw rate converge to 0 after a transient response. This shows that the vehicle eventually enters a stable state after the collision avoidance maneuver. Figure 9 and Figure 10 illustrates the control inputs, i.e. driving/braking forces and front wheel steering angles. It is clear to see that the wheel steering angles in Figure 10 do not change continuously due to the discrete nature of the MPC control. It is the same with the driving/braking forces in Figure 9, but not very obvious. This is because the incremental change of driving forces are quite small compare to the scale of existing forces. Figure 9 also shows the driving forces are stabilized in the end and the forces of front and rear tires have different signs, this could be due to the insufficient information in the cost function that only considers position and velocity, without specifying forces or energy used. Hence, the MPC optimizer could not distinguish a better solution when it finds many. Overall, the steering behavior of MPC to avoid the obstacle is quite reasonable, it turns left a bit to go around the obstacle then turns right to return to the original lane (determined by y_d).

It is worth mention that even though the optimization framework is successful in the relatively simple scenario, it is also found that for more complex scenario with varying speed and dynamic obstacles, the MPC optimizer with static weighting parameters usually yields unsatisfactory results. Intuitively, there could be some relationship between these parameters and various types of risks. However, an adaptive mechanism is needed

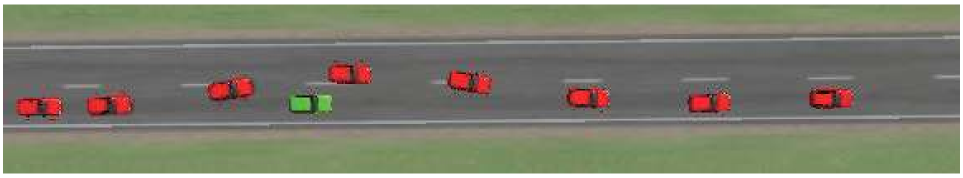

Figure 7. Collision avoidance maneuver of MPC control

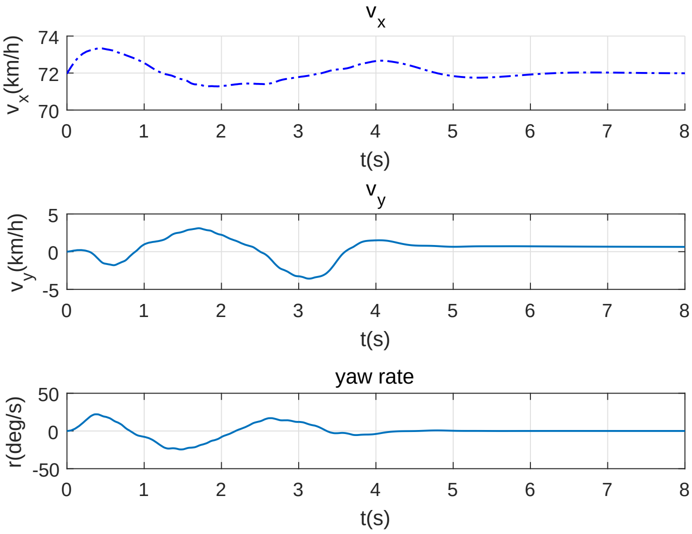

Figure 8. Vehicle states

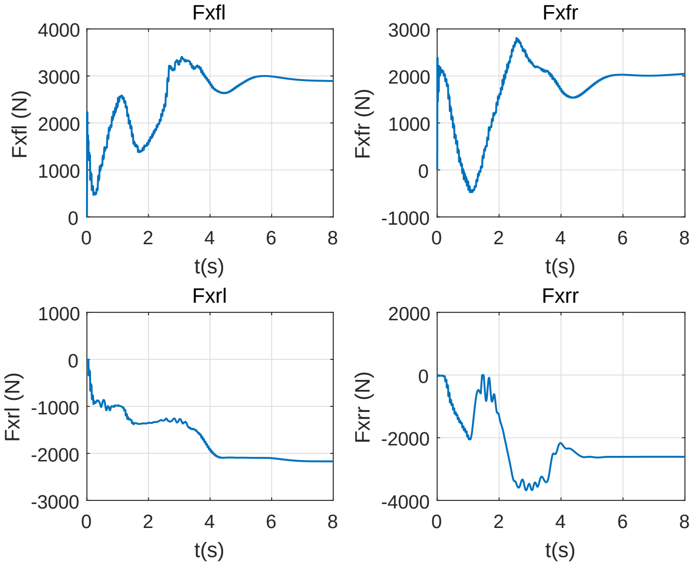

Figure 9. Driving/braking forces

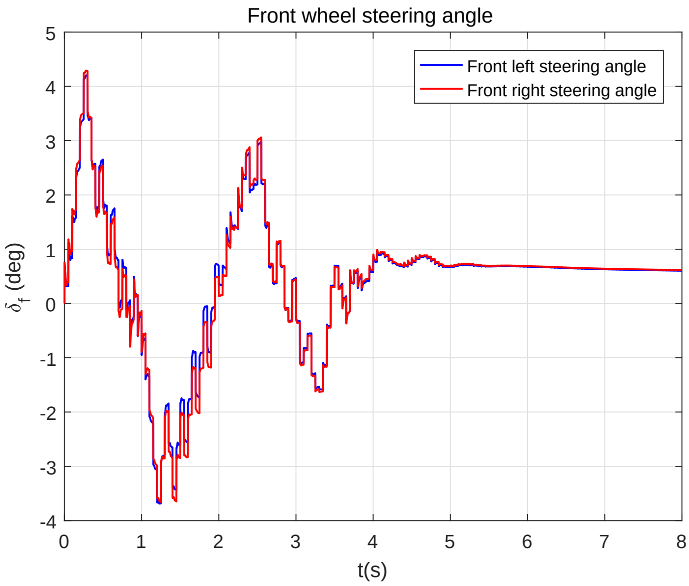

Figure 10. Front steering angles

to establish such a relationship. Currently, an adaptive mechanism based on evolutionary algorithm that performs online simulations and searching is under study.

6. Conclusions

This study has considered the problem of obstacle avoidance of road vehicles via an MPC control scheme. The problem is treated as a optimal control problem by establishing a cost function and a collision avoidance constraint in the MPC formulation and the optimization problem is solved using quadratic programming algorithm for MPC control. Unlike many collision avoidance schemes that require a path planner to plan a collision free path and a controller to tracking this path, the proposed MPC control do not require a path planner as it directly find the optimal driving/braking forces that meet the collision avoidance constraints. Obviously, the form of a cost function is crucial in determining the behavior of the vehicle. In this article, the cost function is in a quadratic form, and the weighting parameters are variable. Intuitively, we could presume that there might be a relationship between these parameters and potential risks. To tackle more complex scenario of varying speeds and dynamic obstacles, it is essential to establish an adaptive optimal mechanism that allows some control parameters change with the environments, which is a challenging task. Our future study will try to establish such an adaptive optimal mechanism by incorporating an evolutionary algorithm that performs online simulations and searching of optimal parameters for the MPC optimizer.

References

[1] Distner M, Bengtsson M. City safety C A system addressing rear-end collisions at low speeds. 21st Enhanced Safety Vehicles Conference; June 15-18, 2009; Stuttgart, German.

[2] Galvani M, Biral F, Nguyen M, Fujimoto H. Four-wheel optimal autonomous steering for improving safety in emergency collision avoidance manoeuvres. IEEE 13th International workshop on advanced Motion Control; Mar. 14-16, 2014; Yokohama, Japan.

[3] Shah J, Zegelaar P, Best M. Performance of emergency steer assist in front wheel and rear wheel steering vehicle. 12th International Symposium on Advanced Vehicle Control (AVEC '14); Sept. 22-26, 2014; Tokyo, Japan.

[4] Choi J, Kim K, Yi K. Emergency driving support algorithm with steering torque overlay and differential braking. 14th IEEE International Conference on Intelligent Transportation Systems (ITSC 2011); Oct. 05-07, 2011; Washington, DC, USA.

[5] Dingle P, Guzzella L. Optimal emergency manoeuvres on highways for passenger vehicles with two-and four-wheel active steering. American Control Conference (ACC); June 30-July 2, 2010; Baltimore, Maryland, USA.

[6] Kelly A, Nagy B. Reactive nonholonomic trajectory generation via parametric optimal control. The International Journal of Robotics Research. 2003; 22(7-8):583–601.

[7] Hassanzadeh M, Lidberg M, Keshavarz M, Bjelkeflo L. Path and speed control of a heavy vehicle for collision avoidance manoeuvres. IEEE Intelligent Vehicle Symposium; June 3-7, 2012; Alcala de Henares, Spain.

[8] Urmson C, Anhalt J, Bagnell D, Baker C, et al. Autonomous Driving in Urban Environ ments: Boss and the Urban Challenge. Journal of Field Robotics. 2008; 25(8):425–466.

[9] Shim T, Adireddy G, Yuan H. Autonomous vehicle collision avoidance system using path planning and model-predictive-control-based active front steering and wheel torque control. Proc IMechE Part D: J. Auto. Eng. 2012; 226(6):767–778.

[10] Shibata N, Seiji S, Takahiro W. Collision avoidance control with steering using velocity potential field. IEEE Intelligent Vehicle Symposium; June 8-11, 2014; Ypsilanti, MI, USA.

[11] Leonard J, How J, Teller S, Berger M, et al. A perception-driven autonomous urban vehicle. The DARPA Urban Challenge. Springer Tracts in Advanced Robotics. Springer, Berlin, Heidelberg, 2009.

[12] Bacha A, Bauman C, Faruque R, Fleming M, et al. Odin: team VictorTango's entry in the DARPA urban challenge. Journal of Field Robotics. 2008; 25(8):125–162.

[13] Valois J, Herman H, Bares J, Rice D. Remote operation of the Black Knight unmanned ground combat vehicle. SPIE Defense and Security Symposium; Mar 16-20, 2008; Orlando, Florida, USA.

[14] Urmson C, Anhalt J, Bartz D, Clark M, et al. A robust approach to high-speed navigation for unrehearsed desert terrain. Journal of Field Robotics. 2006; 23(8):467–508.

[15] Levinson J, Askeland J, Becker J, Dolson J, et al. Towards fully autonomous driving: Systems and algorithms. IEEE Intelligent Vehicle Symposium; June 5-9, 2011; Baden-Baden, Germany.

[16] Yakub F, Mori Y. Comparative study of autonomous path-following vehicle control via model predictive control and linear quadratic control. Proc IMechE Part D: J. Auto. Eng. 2015; 229(12):1695–1714.

[17] Choi M, Choi S. MPC for vehicle lateral stability via differential braking and active front steering considering practical aspects. Proc IMechE Part D: J. Auto. Eng. 2016; 230(4):459–469.

[18] Falcone P, Borrelli F, Asgari J, Tseng H, et al. Predictive active steering control for autonomous vehicle systems. IEEE Trans. on Control System Technology. 2007; 15(3):566–580.

[19] Katriniok A. Optimal vehicle dynamics control and state estimation for a low cost GNSS-based collision avoidance system. Dusseldorf: VDI-Verlag, 2013. IEEE Trans. on Control System Technology. 2007; 15(3):566–580.

[20] Gao Y, Lidberg M, Gordon T. Modified hamiltonian algorithm for optimal lane change with application to collision avoidance. MM Science Journal. Mar. 2014; p. 1411–1419.

[21] Funke J, Brown M, Erlien S, Gerdes J. Collision avoidance and stabilization for autonomous vehicles in emergency scenarios. IEEE Trans. on Control System Technology. 2017; 25(4):1204–1216.

[22] Brown M, Funke J, Erlien S, Gerdes J. Safe driving envelopes for pathtracking in autonomous vehicles. Control Engineering Practice. 2017; 61(4):307–316.

[23] Hongliang Yuan, Dexiang Lin, Timothy Gordon. Unified decision making and control for highway collision avoidance using active front steer and individual wheel torque control. 25th International Symposium on Dynamics of Vehicles on Roads and Tracks; Aug. 14-18, 2017; Rockhampton, Queensland, Australia.

[24] Hongliang Yuan, Yangyan Gao, Timothy J Gordon. Vehicle optimal road departure prevention via model predictive control. Proc IMechE Part D: J. Auto. Eng. 2017; 231(7):952–962.

[25] Pacejka HB. Tyre and vehicle dynamics. 2nd ed. Oxford: Butterworth-Heinemann, 2005.

[26] Mechanical Simulation Corporation. CarSim user reference manual, Ver 8.1. Aug. 2011.

[27] Wang L. Model predictive control system design and implementation using MATLAB. London: Springer-Verlag, 2009.