Original Article

Institution of MECHANICAL ENGINEERS
Journal of MECHANICAL ENGINEERING SCIENCE

Proc IMechE Part C:
J Mechanical Engineering Science
0(0) 1–20

DOI: 10.1177/0954406220945468
journals.sagepub.com/home/pic

Cooperative path-planning and tracking controller evaluation using vehicle models of varying complexities

Husain Kanchwala[1], Icaro Bezerra Viana[2] and Nabil Aouf[3]

Abstract

This paper discusses cooperative path-planning and tracking controller for autonomous vehicles using a distributed model predictive control approach. Mixed-integer quadratic programming approach is used for optimal trajectory generation using a linear model predictive control for path-tracking. Cooperative behaviour is introduced by broadcasting the planned trajectories of two connected automated vehicles. The controller generates steering and torque inputs. The steering and drive motor actuator constraints are incorporated in the control law. Computational simulations are performed to evaluate the controller for vehicle models of varying complexities. A 12-degrees-of-freedom vehicle model is developed and is subsequently linearised to be used as the plant model for the linearised model predictive control-based tracking controller. The model behaviour is compared against the kinematic, bicycle and the sophisticated high-fidelity multi-body dynamics CarSim model of the vehicle. Vehicle trajectories used for tracking are longitudinal and lateral positions, velocities and yaw rate. A cooperative obstacle avoidance manoeuvre is performed at different speeds using a co-simulation between the controller model in Simulink and the high-fidelity vehicle model in CarSim. The simulation results demonstrate the effectiveness of the proposed method.

Keywords

Applied mechanics, automobile, automotive control, control theory, dynamics, dynamic modelling, dynamic systems, electric vehicle, intelligent control, mathematical modelling

Date received: 27 August 2019; accepted: 14 June 2020

Introduction

The main components of an autonomous vehicle are perception, planning and control. This paper discusses the planning and control aspects. For collision avoidance, generally a hierarchical control scheme is used with high level path-planner[1] and low level tracking controller.[2]

Significant research has been done in path-planning strategies using mixed-integer quadratic programming (MIQP),[3] polynomials,[4] B-splines,[5] elastic bands[6] and potential fields.[7] In the research community, the path-planning problem has been widely studied for a single vehicle while in an autonomous driving environment multiple vehicles are present. A challenging research task is to achieve coordination by considering the trajectories of other autonomous vehicles. Schouwenaars et al.[8] used an optimal path-planning approach for multiple vehicles based on MIQP. Frese and Beyerer[9] compared several cooperative path-planning algorithms like tree search, elastic bands, priority-based approach, etc. for their computational times.

The lower level path-tracking controller ensures that a collision-free path is generated while a successful obstacle avoidance manoeuvre is achieved by making the vehicle precisely follow the planned trajectory. Trajectory tracking methods use fuzzy logic,[10] proportional–integral–differential,[11] pure pursuit strategy,[12] linear quadratic regulator,[13] sliding-mode control,[14] robust control[15] and model predictive control.[16] MPC-based controllers easily handle actuator constraints and other uncertainties.[17]

Tracking controller requires vehicle model for mapping the reference trajectories. Researchers have widely used simplified models like kinematic model,[18]

[1]Warwick Manufactruing Group, University of Warwick, Coventry, UK
[2]Centre for Electronic Warfare, Information and Cyber Cranfield University, Shrivenham, UK
[3]Department of Electrical and Electronic Engineering City University of London, London, UK

Corresponding author:
Husain Kanchwala, Warwick Manufacturing Group, University of Warwick, Coventry CV4 7AL, UK.
Email: husain.kanchwala@warwick.ac.uk

bicycle model,[19] planar lateral dynamics full car model[20] and models ignoring effects of roll, pitch and heave motions for developing the tracking controller.[21] Moreover, when the chassis angular motions are considered, a general assumption is made that the vehicle rolls and pitches about the centre of gravity (CG), while in fact the vehicle rolls and pitches about roll and pitch axis respectively.[22] For low-speed manoeuvres, simplified models capture the vehicle dynamics characteristics sufficiently well but at higher vehicle speeds and limits of handling there are significant differences in the predictions of these simplified.[23] These models are routinely linearised about an operating point but they deviate significantly from the nonlinear models at other operating points. To overcome this limitation, some researchers have used linear time-varying bicycle model but effect of out-of-plane motions is not captured.[24] Our contribution in this direction is to develop a linear time-varying model that captures the coupling effects of roll, pitch and heave motions.

Moreover, the limitation of simplified models is that only a limited number of vehicle states can be controlled. A vehicle model of intermediate complexity between a sophisticated commercial multibody dynamics model[25] and these simplified vehicle models is desirable for a real-time application. We developed a four-wheel vehicle model considering the effects of roll, pitch and unsprung mass motions, which can emulate real vehicle dynamic characteristics with low computational expense.

The main contribution of this paper is to develop a cooperative path-planning and tracking algorithm and evaluate it using models of varying complexities.

The cooperative path-planning architecture is based on a distributed MPC approach, where path-planner for each vehicle optimises its optimal trajectory according to the planned trajectories of their neighbouring vehicles that is broadcasted at each time step. In the tracking controller, a future input sequence of steering angle and drive torque is calculated for a predefined prediction horizon. The vehicle trajectories that are tracked are the positions and velocities in longitudinal and lateral directions.

The paper is organised as follows: methodology is described at first followed by the vehicle dynamics model. Four different vehicle models, namely, the kinematic model, bicycle model, a 12-degrees-of freedom (dof) model of intermediate complexity and a high-fidelity CarSim model are discussed in the vehicle model section. A relative comparison study of the vehicle trajectories obtained from various vehicle models for a single lane change manoeuvre is performed. The nonlinear vehicle model is subsequently linearised to be used in the MPC formulation for tracking. Next, the path-planning algorithm developed using model predictive control and MIQP is discussed followed by the formulation of the tracking controller. The results are discussed for a cooperative double lane change manoeuvre and finally conclusions are discussed.

Methodology

The proposed hierarchical scheme is shown in Figure 1. On the upper level, given a set of vehicles $\mathcal{V} \triangleq \{1, 2, \ldots, N_v\}$, a path-planner for each vehicle generates the trajectory $\boldsymbol{X}_p^{(i)}$ or $\mathbf{r}_p^{(i)} \triangleq [x_p^{(i)}\ y_p^{(i)}\ v_{x,p}^{(i)}\ v_{y,p}^{(i)}]^{\mathrm{T}} \in \mathbb{R}^4 \forall i \in \mathcal{V}$ to follow a pre-defined route.

To generate the collision-free reference trajectories, we assume that at every time step k, each vehicle i receives the future plans of the others agents $\boldsymbol{X}_p^{(j)}$ or $\mathbf{r}_p^{(j)} \triangleq [x_p^{(j)}\ y_p^{(j)}\ v_{x,p}^{(j)}\ v_{y,p}^{(j)}]^{\mathrm{T}} \in \mathbb{R}^4$ over a finite horizon, where j denotes the jth surrounding vehicle.

On the lower level, the tracking controller tracks the trajectory (i.e. a set of way-points) $\mathbf{r}_p^{(i)}$ generated by the path-planning layer, which is then used as the reference trajectory $\mathbf{r}_t$. The subscripts p and t are used to represent the trajectories for the path-planner and the controller, respectively. The tracking controller computes the drive torque T and the steering wheel angle δ_{sw} necessary to control the vehicle. Note here that this is different from the road wheel angle δ, which can be obtained by dividing the steering wheel angle with the steering ratio. The assumptions made in this approach are that the route is known beforehand, i.e. a start and a goal position and the current state of the vehicle is estimated based on the measurements from the vehicle sensors and are available for feedback.

MPC-based path-tracking algorithm is presented to follow desired speeds while allowing to temporary deviate from desired paths in Funke et al.[26] The underlying idea of optimisation with both velocity and position states is quite useful.[27] In addition to tracking of the longitudinal and lateral positions and velocities, we also track the yaw rate reference. Therefore, the vehicle control output vector is given by

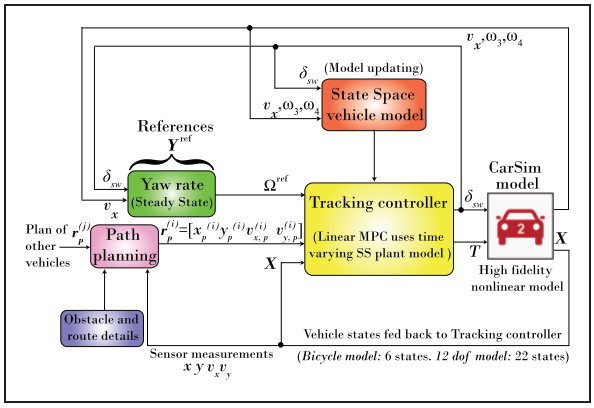

Figure 1. Proposed algorithm of path planning and tracking.

$$\boldsymbol{Y} = \{ x \quad y \quad v_x \quad v_y \quad \omega \}$$

Vehicle model

Four vehicle models are described in the following sections.

Kinematic model

The kinematic vehicle model is shown in Figure 2(a). The inputs to the model are vehicle speed v and steering angle δ. From Figure 2(a), the longitudinal and lateral velocities of the centre of mass G are given by

$$\dot{x} = v\cos(\psi + \beta); \quad \dot{y} = v\sin(\psi + \beta) \tag{1}$$

This model assumes no-slip condition. This implies that the velocities of each wheel points in the same direction of wheel and are equal. As there is no-slip condition, there lies an instantaneous centre of rotation 'O' about which the vehicle rotates and the rotation rate of the vehicle Ω is given by

$$\Omega = \dot{\psi} = \frac{v}{R}; \quad R = \frac{l_r}{\sin\beta} \tag{2}$$

where R is the instantaneous radius of rotation. Eliminating R we obtain the governing equations of motion as

$$\dot{x} = v\cos(\psi + \beta); \quad \dot{y} = v\sin(\psi + \beta); \quad \dot{\psi} = \frac{v}{l_r}\sin\beta$$

From Figure 2(a)

$$\Delta\mathrm{OBG}, \mathrm{OBA}: \ \beta = \tan^{-1}\left(\frac{\mathrm{l_r}}{\mathrm{l}}\tan\delta\right) \tag{3}$$

Bicycle model

The bicycle model is shown in Figure 2(b).

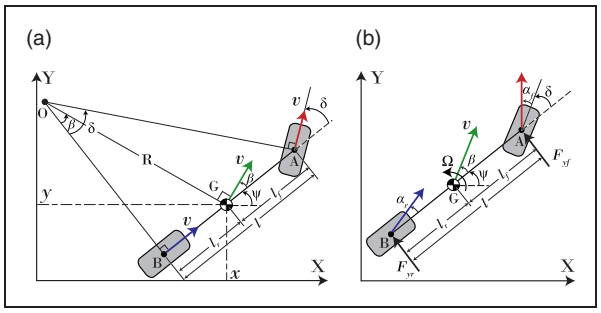

Figure 2. (a) Kinematic model; (b) bicycle model.

Under small steering angle approximation, the front and rear slip angles are given by

$$\alpha_f = \beta + \frac{\Omega l_f}{V} - \delta; \quad \alpha_r = \beta - \frac{\Omega l_r}{V} \tag{4}$$

The lateral acceleration of the CG a_y can be obtained from the fixed and rotating coordinate system representation as

$$a_y = V\dot{\beta} + \dot{V}\beta + \Omega V \tag{5}$$

The lateral force balance equations for the bicycle model (again under small angle approximation) is given by

$$F_{yf} + F_{yr} = ma_y = m(V\dot{\beta} + \dot{V}\beta + \Omega V) \tag{6}$$

F_{yf} and F_{yr} are the front and rear cornering forces given by

$$F_{yf} = -C_{\alpha_f}\alpha_f; \quad F_{yr} = -C_{\alpha_r}\alpha_r \tag{7}$$

From equations (4), (6) and (7) we obtain

$$\begin{aligned} m(V\dot{\beta} + \dot{V}\beta + \Omega V) = &-C_{\alpha_f}\left(\beta + \frac{\Omega l_f}{V} - \delta\right) \\ &- C_{\alpha_r}\left(\beta - \frac{\Omega l_r}{V}\right) \end{aligned} \tag{8}$$

The longitudinal force balance (under small steering angle δ approximation) yields

$$m(\dot{V} - \Omega V\beta) = F_{xr} - F_{yf}\delta - F_d - F_r \tag{9}$$

F_d is the aerodynamic drag and F_r is the rolling resistance given by

$$F_d = \frac{1}{2}\rho C_d A u^2; \quad F_r = C_r W = C_r mg \tag{10}$$

C_d is the drag coefficient, C_r is the coefficient of rolling resistance, A is the car frontal area and W is the vehicle weight. For simplicity we assume no longitudinal slip, which implies that, $F_x = \frac{T}{r_w}$.

Finally, yaw moment balance about the CG 'G' yields

$$\begin{aligned} I_z\dot{\Omega} = &\ l_f F_{yf} - l_r F_{yr} \\ = &-C_{\alpha_f}\left(\beta + \frac{\Omega l_f}{V} - \delta\right)l_f + C_{\alpha_r}\left(\beta - \frac{\Omega l_r}{V}\right)l_r \end{aligned} \tag{11}$$

Equations (8) to (11) represent governing equations of motion.

12-dof model

The 12-dof model is shown in Figure 3.

This model consists of three translational dof corresponding to longitudinal, lateral and vertical motion; three rotational dof corresponding to roll, pitch and yaw; four dof (one for each wheel) for wheel jounce; two dof for the angular rotation of rear left and right wheels. Note that in our present study, we have considered a vehicle with rear wheel drive and so we are interested only in the rear wheel velocities for longitudinal slip calculation as described in the section on the wheel model.

Coordinate system. It is convenient to write the equations of motion in noninertial (moving) reference frame $O(x, y, z)$ instead of the inertial frame $C(X, Y, Z)$ (see Figure 4). The fixed or inertial coordinate system is attached to the ground with X-axis in the forward, Y-axis in the lateral and Z-axis in the vertical direction such that a right-handed Cartesian coordinate system is formed. The noninertial or rotating coordinate system is attached to the vehicle, which is rotating about the z-axis with angular velocity Ω (see Figure 4). The x-axis points forwards and lies both in the ground plane and in the plane normal to the ground that passes through the so-called roll axis. The y-axis points to the left and the z-axis points upwards and passes through the CG at zero roll and pitch.[28]

For deriving all equations of motion, small angle approximation for roll, pitch and wheel steering angle is considered. Eventually, the nonlinear equations of motion are linearised and are represented in the state-space form.

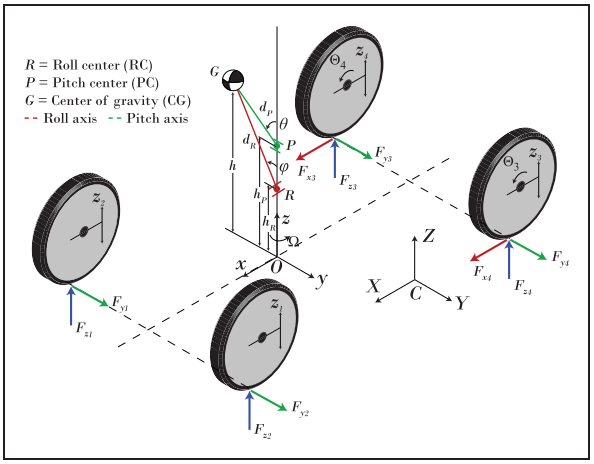

Figure 3. 12-degrees-of-freedom vehicle model.

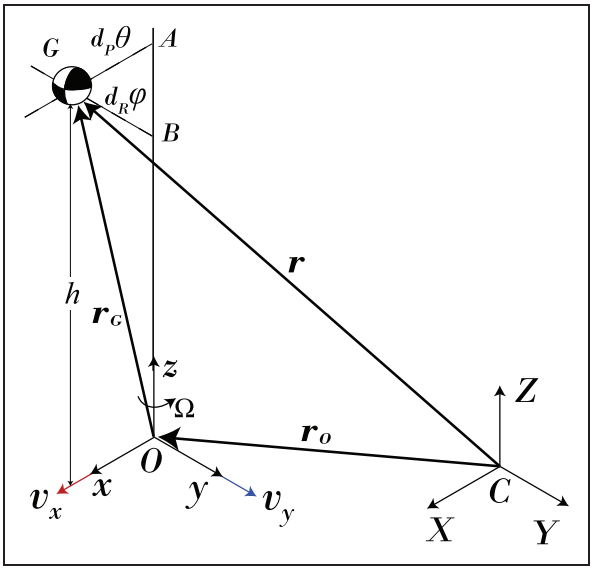

Figure 4. Fixed and rotating coordinate systems.

Planar motions. In this section, we study the planar motions of the chassis in x, y directions and yaw about z-axis.

Longitudinal and lateral dynamics:

The absolute position of the vehicle CG is given by

$$\boldsymbol{r} = \boldsymbol{r}_{\mathrm{O}} + \boldsymbol{r}_{\mathrm{G}} \tag{12}$$

The position vector $\boldsymbol{r}_{\mathrm{G}}$ is given by

$$\boldsymbol{r}_{\mathrm{G}} = d_p\theta\,\hat{i} - d_r\phi\,\hat{j} + h\,\hat{k} \tag{13}$$

Since $O(x, y, z)$ is a rotating coordinate system with angular velocity $\boldsymbol{\Omega} = \Omega\hat{k}$, therefore the absolute velocity of the CG is given by

$$\dot{\boldsymbol{r}} = \boldsymbol{v} = \dot{\boldsymbol{r}}_{\mathrm{O}} + \dot{\boldsymbol{r}}_{\mathrm{G}} + \boldsymbol{\Omega} \times \boldsymbol{r}_{\mathrm{G}} \tag{14}$$

For small pitch and roll angles, above equation reduces to

$$\boldsymbol{v} = (v_x + d_p\dot{\theta} + d_r\phi\Omega)\hat{i} + (v_y - d_r\dot{\phi} + d_p\theta\Omega)\,\hat{j} \tag{15}$$

where v_x and v_y are the velocities of the moving coordinate system. Correspondingly, the acceleration is given by

$$\boldsymbol{a} = \dot{\boldsymbol{v}} + \boldsymbol{\Omega} \times \boldsymbol{v} \tag{16}$$

which yields

$$\begin{aligned}\boldsymbol{a} = {} & (\dot{v}_x + d_p\ddot{\theta} + 2d_r\dot{\phi}\Omega + d_r\phi\dot{\Omega} - v_y\Omega - d_p\theta\Omega^2)\,\hat{i} \\ & + (\dot{v}_y - d_r\ddot{\phi} + 2d_p\dot{\theta}\Omega + d_p\theta\dot{\Omega} + v_x\Omega + d_r\phi\Omega^2)\,\hat{j}\end{aligned} \tag{17}$$

Sum of the external forces is equal to the rate of change of linear momentum $\boldsymbol{P}$. Applying force balance equation $\sum \boldsymbol{F} = \dot{\boldsymbol{P}}$, and taking x and y components (planar motion) yield

$$\sum F_x = ma_x; \quad \sum F_y = ma_y \tag{18}$$

where a_x, a_y are $\hat{i},\hat{j}$ components of the acceleration vector $\boldsymbol{a}$.

For small values of wheel steering angle δ and side-slip angles α_i, the equations can be linearised and the resultant forces in x and y directions are given by (There is a small difference between left and right wheel steering angles for normal handling situations. Using the same steering angle for both inner and outer wheels is an acceptable approximation for handling analyses.[29])

$$\begin{aligned}&\sum F_x = F_{x3} + F_{x4} - (F_{y1} + F_{y2})\delta - F_d - \sum F_{rr,i};\\ &\sum F_y = F_{y1} + F_{y2} + F_{y3} + F_{y4}\end{aligned} \tag{19}$$

It is important to note here that the vehicle used for this study has rear wheel drive and therefore the longitudinal forces are generated by rear tyres only (see Figure 3).

F_d is the aerodynamic drag and $F_{rr,i}$ is the rolling resistance acting on each wheel (i) given by

$$F_d = \frac{1}{2}\rho C_d A v_x^2; \quad F_{rr,i} = C_{rr}F_{zi} \tag{20}$$

where C_d is the drag coefficient, C_{rr} is the coefficient of rolling resistance, A is the car frontal area and F_{zi} is the normal load on the tyre (see 'Suspension dynamics' section) with index $i \in \{1-4\}$ corresponding to front left/right and rear left/right wheel, respectively.

Yaw dynamics:

To study the yaw dynamics, we begin with the calculation of the angular momentum. The angular momentum $\boldsymbol{H_G}$ around CG G of a rigid body can be written as

$$\begin{aligned}\boldsymbol{H_G} = \boldsymbol{I}\times\boldsymbol{\omega} &= (I_{xx}\omega_x - I_{xy}\omega_y - I_{xz}\omega_z)\hat{i}\\ &+ (-I_{yx}\omega_x + I_{yy}\omega_y - I_{yz}\omega_z)\hat{j}\\ &+ (-I_{zx}\omega_x - I_{zy}\omega_y + I_{zz}\omega_z)\hat{k}\end{aligned} \tag{21}$$

The elements I_{xx}, I_{yy} and I_{zz} are the inertia moments and I_{xy}, I_{yz} and I_{zx} are inertia products and are the off-diagonal terms of the inertia tensor. The components of the angular velocity $\boldsymbol{\omega} = [\omega_x\ \omega_y\ \omega_z]$ in x, y and z direction are $\omega_x = \dot{\phi}, \omega_y = \dot{\theta}$ and $\omega_z = \Omega$. Considering sprung mass to be symmetric in x–z plane, $I_{xy} = I_{yz} = 0$ and $I_{zx} = I_{xz}$ in the above equation yields

$$\boldsymbol{H_G} = (I_{xx}\dot{\phi} - I_{xz}\Omega)\hat{i} + I_{yy}\dot{\theta}\hat{j} + (I_{zz}\Omega - I_{zx}\dot{\phi})\hat{k} \tag{22}$$

For calculating the angular momentum $\boldsymbol{H_O}$ about any arbitrary point O, we consider the transfer theorem as

$$\boldsymbol{H_O} = \boldsymbol{H_G} + \boldsymbol{r_G}\times\boldsymbol{P} \tag{23}$$

Finally the moment balance equation is given by

$$\sum M = \dot{\boldsymbol{H}}_{\boldsymbol{O}} \tag{24}$$

Considering the moment balance in z direction about O in Figure 5 we obtain

$$\begin{aligned}&(I_{zz} + m_s((d_r\phi)^2 + (d_p\theta)^2))\dot{\Omega} - I_{xz}\ddot{\phi}\\ &\quad = \sum M_z - m_s(d_r\phi a_x + d_p\theta a_y)\end{aligned} \tag{25}$$

The external moment $\sum M_z$ is given by

$$\begin{aligned}\sum M_z = {}&(F_{y1} + F_{y2})l_f - (F_{y3} + F_{y4})l_r\\ &+ (F_{x4} - F_{x3})\frac{t}{2} + (F_{y1} + F_{y2})\delta\frac{t}{2}\end{aligned} \tag{26}$$

It is important to note that the input given by the driver is the steering wheel angle δ_{sw} and in order to obtain the road wheel angle δ, the steering wheel angle has to be divided by the steering ratio (SR). On substituting the values of $\sum M_z$ from above equation and the values of a_x and a_y from equation (17), simplifying and neglecting small terms we obtain

$$\begin{aligned}&I_{zz}\dot{\Omega} - I_{xz}\ddot{\phi} + m_s[d_r\phi(\dot{v}_x - v_y\Omega) + d_p\theta(\dot{v}_y + v_x\Omega)]\\ &= (F_{y1} + F_{y2})l_f - (F_{y3} + F_{y4})l_r\\ &\quad + (F_{x4} - F_{x3})\frac{t}{2} + (F_{y1} + F_{y2})\delta\frac{t}{2}\end{aligned} \tag{27}$$

Finally, the in-plane motions can be studied using longitudinal and lateral force balance equations in (18) and yaw moment balance equation in (27).

Out-of-plane motions. In this section, out-of-plane motions: roll, pitch, heave and wheel jounce motions are analysed.

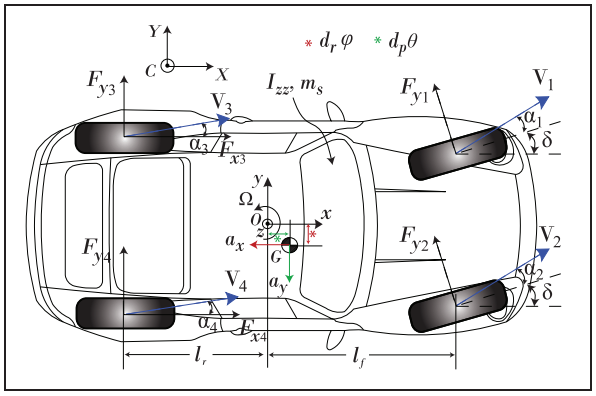

Figure 5. Lateral dynamics model.

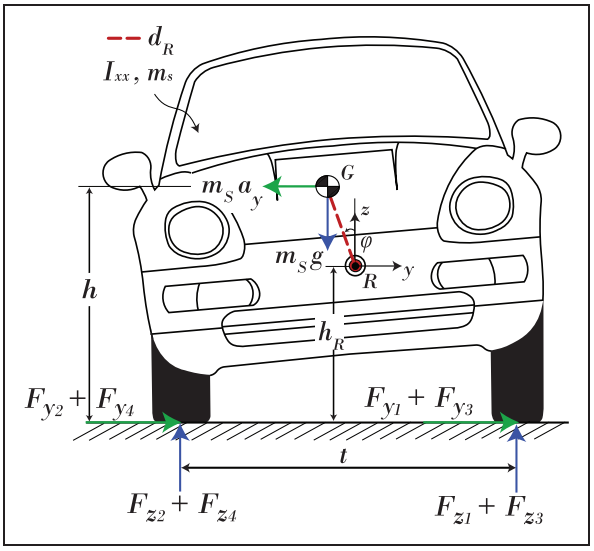

Figure 6. Roll dynamics model.

Roll dynamics:

The vehicle rolls about the roll axis passing through the front and rear suspension roll centres.[30] The distance between the CG and the roll axis (roll centre) is the roll moment arm d_r (see Figure 6). The roll centre height varies with suspension travel but for the sake of simplicity this variation is neglected.

$$\begin{aligned}&(I_{xx} + m_s d_r^2)\ddot{\phi} + (I_{zz} - I_{yy})\dot{\theta}\Omega - I_{xz}\dot{\Omega} \\ &= M_x + m_s a_y d_r + m_s g d_r \phi \\ &\text{where, } M_x = M_{xs} + M_{xu}\end{aligned} \tag{28}$$

M_{xs} and M_{xu} are sprung mass and unsprung mass roll moments given by

$$\begin{aligned}M_{xs} = M_{\phi f} + M_{\phi r} &= -(K_{\phi f} + K_{\phi r})\phi - (C_{\phi f} + C_{\phi r})\dot{\phi}, \\ M_{xu} &= \sum_{i=1}^{4} F_{yi} h_r; \\ K_{\phi f,r}/C_{\phi f,r} &= \frac{t^2}{2} K_{f,r}/C_{f,r}\end{aligned} \tag{29}$$

$K_{\phi f,r}$ and $C_{\phi f,r}$ are front/ rear roll stiffness and roll damping, respectively. From equations (28) through (29) we obtain

$$\begin{aligned}&(I_{xx} + m_s d_r^2)\ddot{\phi} + (I_{zz} - I_{yy})\dot{\theta}\Omega - I_{xz}\dot{\Omega} - m_s d_r(a_y + g\phi) \\ &+ (K_{\phi f} + K_{\phi r})\phi + (C_{\phi f} + C_{\phi r})\dot{\phi} - \sum_{i=1}^{4} F_{yi} h_r = 0\end{aligned} \tag{30}$$

Pitch dynamics:

As similar to the roll dynamics, the vehicle pitches about the pitch centre P in Figure 3. The distance between the CG and the pitch centre is the pitch moment arm d_p (see Figure 7). Considering the pitch moment balance (small pitch angle approximation) about the pitch centre P we obtain

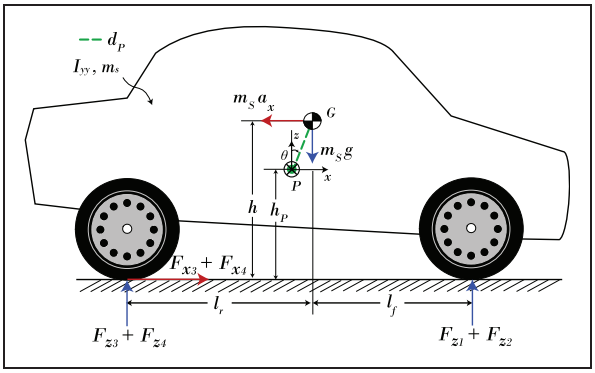

Figure 7. Pitch dynamics model.

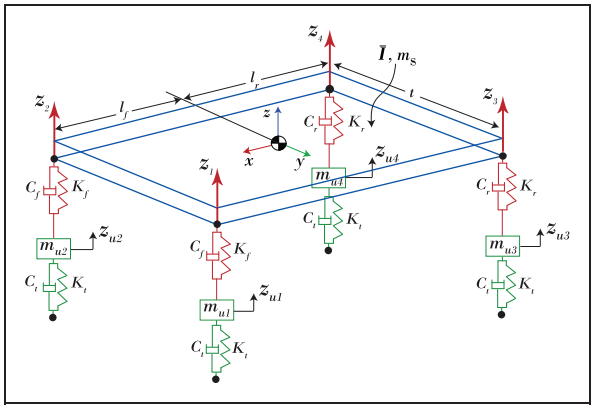

Figure 8. 7-dof ride model. In the present study, the vehicle moves on a flat road so ground deflections are not considered.

$$\begin{aligned}&(I_{yy} + m_s d_p^2)\ddot{\theta} + (I_{xx} - I_{zz})\dot{\phi} - m_s g d_p \theta + m_s a_x d_p \theta\Omega \\ &= (F_{z3} + F_{z4})l_r - (F_{z1} + F_{z2})l_f - (F_{x3} + F_{x4})h_p\end{aligned} \tag{31}$$

Suspension dynamics:

In this part, we will study the heave motions of the chassis and wheel jounce. To study this, we will consider the 7-dof ride model of the car as shown in Figure 8.

Considering the force balance in z direction, the governing equation for the chassis heave motion z is given by

$$\begin{aligned}m_s a_z = \sum F_z &= -\sum_{i=1}^{4} (K_i(z_i - z_{ui}) + C_i(\dot{z}_i - \dot{z}_{ui})), \\ \forall i \in \{1-2\},\ K_i, C_i &= K_f, C_f \text{ else } K_r, C_r\end{aligned} \tag{32}$$

where the corner displacements z_i are given by

$$z_1 = z + \frac{t}{2}\phi - l_f\theta; \quad z_2 = z - \frac{t}{2}\phi - l_f\theta;$$
$$z_3 = z + \frac{t}{2}\phi + l_r\theta; \quad z_4 = z - \frac{t}{2}\phi + l_r\theta$$

Considering the force balance in z direction for each unsprung mass we obtain the wheel jounce z_{ui} as

$$\begin{aligned} m_{u1}\ddot{z}_{u1} &= K_f(z_1 - z_{u1}) + C_f(\dot{z}_1 - \dot{z}_{u1}) - K_t z_{u1} - C_t\dot{z}_{u1};\\ m_{u2}\ddot{z}_{u2} &= K_f(z_2 - z_{u2}) + C_f(\dot{z}_2 - \dot{z}_{u2}) - K_t z_{u2} - C_t\dot{z}_{u2};\\ m_{u3}\ddot{z}_{u3} &= K_r(z_3 - z_{u3}) + C_r(\dot{z}_3 - \dot{z}_{u3}) - K_t z_{u3} - C_t\dot{z}_{u3};\\ m_{u4}\ddot{z}_{u4} &= K_r(z_4 - z_{u4}) + C_r(\dot{z}_4 - \dot{z}_{u4}) - K_t z_{u4} - C_t\dot{z}_{u4} \end{aligned} \tag{33}$$

The vertical forces on the tyres is given by

$$\begin{aligned} F_{zi} &= mg\left(\frac{l - l_i}{l}\right) - K_t z_{ui} - C_t\dot{z}_{ui};\\ \forall i \in \{1-4\}; &\quad l_{1,2} = l_f, \quad l_{3,4} = l_r \end{aligned} \tag{34}$$

Wheel model:

The longitudinal and lateral forces are generated by the wheels. As pointed out earlier, we have considered a rear wheel driven vehicle. As a result, the longitudinal force is generated only by the rear wheels while the lateral forces are generated by all the four wheels.

The first step in calculating the forces generated by a tyre, is to compute the longitudinal slip s_{xi}, and lateral slip s_{yi}. To calculate wheel slip we require wheel corner velocities and wheel angular speeds, which are obtained from Figure 9.

The wheel corner velocities depend on vehicle velocity, yaw rate and geometrical parameters of the chassis as

$$\begin{aligned} v_{x1,3} &= v_x - \Omega\frac{t}{2}; \quad v_{x2,4} = v_x + \Omega\frac{t}{2};\\ v_{y1,2} &= v_y + \Omega l_f; \quad v_{y3,4} = \Omega l_r - v_y \end{aligned} \tag{35}$$

The theoretical slip quantities are defined as[31]

$$s_{xi} = \frac{v_{xi} - \omega_i r_w}{\omega_i r_w}; \quad s_{yi} = \frac{v_{yi}}{\omega_i r_w} = \tan\alpha_i(1 + s_{xi}) \tag{36}$$

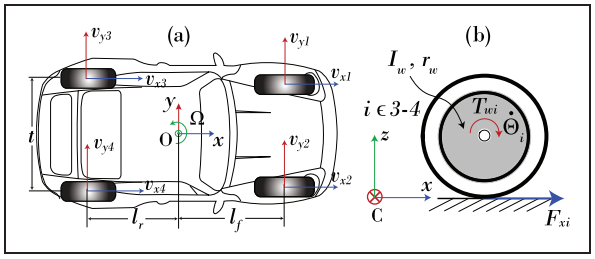

Figure 9. (a) Corner velocities; (b) wheel angular dynamics.

The overall or total slip at each tyre is defined by

$$s_i = \sqrt{s_{xi}^2 + s_{yi}^2} \tag{37}$$

Next the total friction coefficient using Pacejka's 'magic formula (MF)' is given by

$$\mu(s_i) = D\sin(C\tan^{-1}(Bs_i)) \tag{38}$$

Assuming symmetric tyre characteristics, the total friction force for each tyre lies within the friction circle and the longitudinal and lateral tyre forces are given by[32]

$$\begin{aligned} F_{xi} &= \frac{s_{xi}}{s_i}\mu(s_i)F_{zi}; \ \ i \in \{3-4\}\\ F_{yi} &= \frac{s_{yi}}{s_i}\mu(s_i)F_{zi}; \ \ i \in \{1-4\}. \end{aligned} \tag{39}$$

In the above equations, the wheel angular velocity $\omega_i = \dot{\Theta}_i$ is calculated using the driving torque and the longitudinal force generated in the contact between the road and the tyre as shown in Figure 9(b).

$$I_w\dot{\omega}_i = \frac{T}{2} - F_{xi}r_w; \ \ i \in \{3-4\} \tag{40}$$

In our case, the engine torque T is equally distributed to both the rear wheels as the vehicle considered for study is a rear wheel driven car with an open differential of gear ratio 1. As a result, only the longitudinal slips of rear wheels come into picture ($i \in \{3-4\}$) while the lateral forces are generated by all the four wheels ($i \in \{1-4\}$). This brings us to an end of vehicle modelling. These nonlinear vehicle models are finally linearised to be used in the linear MPC based tracking controller. The linearisation procedure is described in the 'Linearisation' section.

In addition to these we have developed a sophisticated multi body dynamics vehicle model in CarSim which represents a real world vehicle and will be eventually used to evaluate the tracking controller.

CarSim model:

The CarSim modeling interface is shown in Figure 10. The vehicle model was built for a prototype Westfield Sports Car.[33] The weight, inertia, CG location of the vehicle, suspension and steering properties were obtained from measurements. The vehicle parameters used in this model are reported in Table 1. The vehicle is equipped with double wishbone suspensions in both front and rear axles with a rigid roll-cage chassis. The values for spring and damper parameters, roll centre height, track width, spin inertias and compliance coefficients were used to build the suspension model. The suspension kinematics and compliance characteristics were fed as look-up tables in CarSim to model the suspension characteristics. An approximate linear relationship between handwheel and

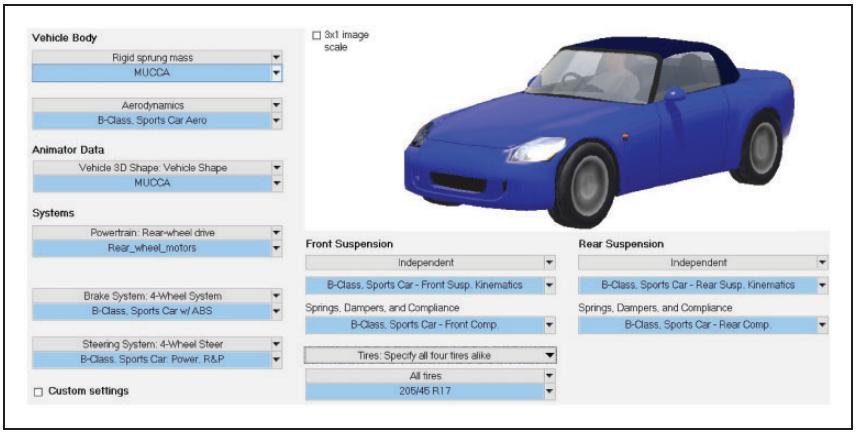

Figure 10. CarSim modeling interface.

Table 1. Vehicle parameters used to develop the mathematical model.

Notation	Description	Value	Notation	Description	Value
m	Vehicle mass	950 kg	l	Wheel base	2350 mm
m_s	Sprung mass	850 kg	l_f	Wheel base front	1350 mm
m_u	Unsprung mass (each)	25 kg	l_r	Wheel base rear	1000 mm
C_d	Drag coefficient	0.3	t	Track width	1500 mm
A	Car frontal area	1.6 m^2	K_f, K_r	Spring stiffness front, rear	45, 60 N/mm
SR	Steering ratio	13	C_f, C_r	Shock damping front, rear	3.5, 4 N s/mm
C_{rr}	Rolling coefficient	0.01	K_t	Tyre stiffness	250 N/mm
I_{xx}	Roll inertia	325 kg m^2	C_t	Tyre damping	0.75 N s/mm
I_{yy}	Pitch inertia	1000 kg m^2	I_w	Wheel inertia	2.1 kg m^2
I_{zz}	Yaw inertia	1200 kg m^2	r_w	Wheel radius	325 mm
I_{xz}	Product of inertia	140 kg m^2	B, C, D	Tyre parameters (MF)	7, 1.6, 1
d_r	Roll moment arm	125 mm	h_r	Roll centre height	200 mm
d_p	Pitch moment arm	150 mm	h_p	Pitch centre height	175 mm
C_{α_f}	Tyre cornering stiffness (front)	4.4 kN/rad	C_{α_r}	Tyre cornering stiffness (rear)	6 kN/rad

roadwheel steer angle is found for the range of $\pm 3\pi$ (three and a half turns). The steering ratio is found to be equal to 13. Also, other parameters like caster, kingpin angle, lateral offset were modelled based on measurements. These details were obtained from the manufacturer and not discussed here for the sake of brevity. A magic formula based tyre model was used and the tyre vertical, longitudinal and lateral characteristics were inputted by means of look-up charts, which were obtained from Westfield Cars. The inputs to the CarSim model are engine troque and steering wheel angle. The default powertrain was replaced by an external drive torque command and the traditional rack and pinion steering system was replaced by external steering command coming from the tracking controller.

The mathematical model so developed has 57 dof, which in turn solves 114 first-order differential equations to calculate the vehicle responses. This model is exported as an S-function and in Simulink and a co-simulation interface is setup to assess the performance of the tracking controller.

Model comparisons

So far we developed four vehicle models, namely, kinematic model, bicycle model, 12-dof model and the CarSim model. These models differ from each other in terms of complexity. The kinematic model is the simplest with only three states. It does not take into account tyre compliance and is geometry based. The bicycle model is a further correction of the kinematic model with tyre lateral compliance taken into account. The 12-dof model takes suspension compliance, out-of-plane motions and wheel jounce into account and is a model of intermediate complexity as opposed to the sophisticated high-dof CarSim model.

The predictions of these vehicle models differ significantly at high lateral accelerations. To demonstrate this, the vehicle trajectories obtained from various models for a lane change manoeuvre at 80 km/h are compared in Figure 11. A sinusoidal steering wheel angle of 26° is given to achieve a lane

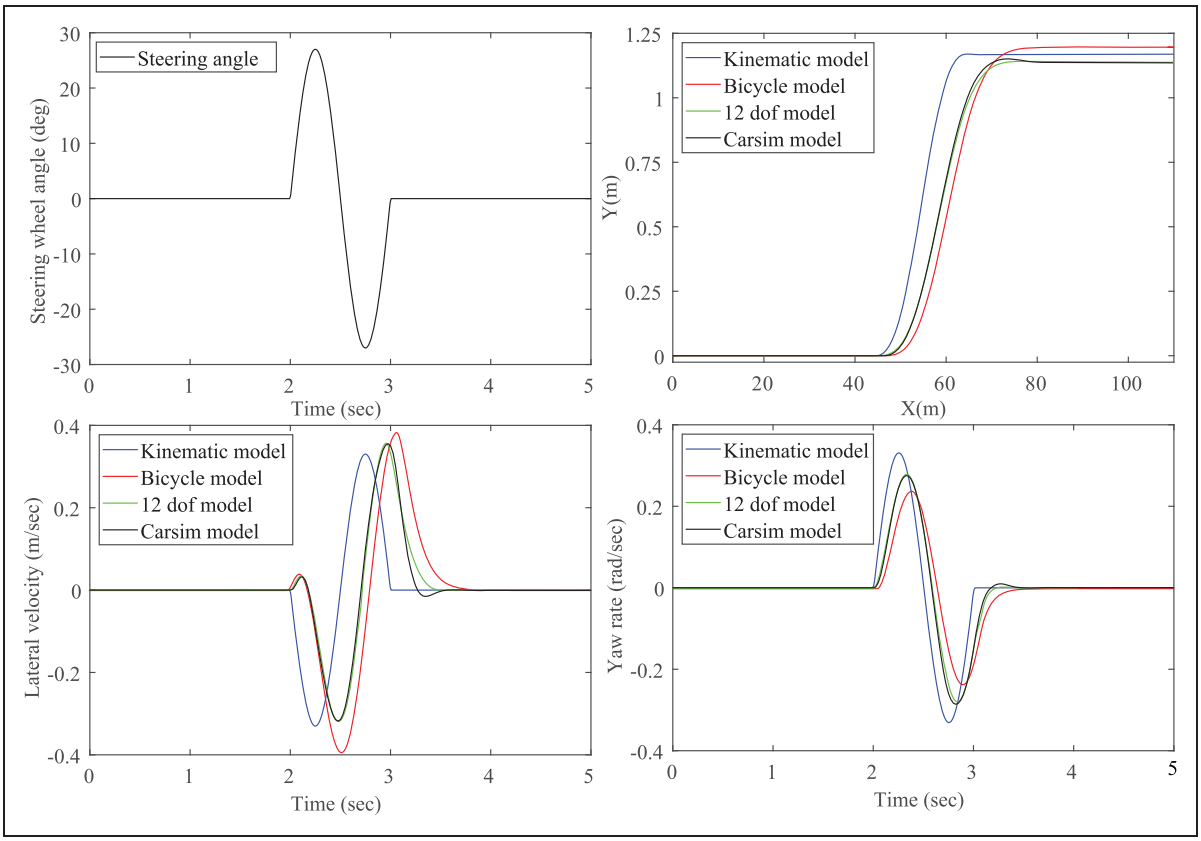

Figure 11. Responses of different models for a sinusoidal steering wheel angle input for a lane change maneuver at 80 km/h.

change manoeuvre. The vehicle path $X - Y$, lateral velocity v_y and yaw rate Ω trajectories are obtained for each model.

From Figure 11, it is evident that the predictions from Kinematic model substantially differ from that of the sophisticated CarSim model. Bicycle model gives a better prediction of the yaw rate response but it fails to give a representative prediction of lateral velocity and correspondingly the lateral position trajectories. The 12-dof proposed model is midway in complexity as opposed to the CarSim model and it fairly captures all the necessary vehicle dynamic handling characteristics.

For subsequent implementation, we have chosen 12-dof model and will compare its response against the bicycle model after integrating path-planning and tracking controller. Now, we discuss the linearisation of these models.

Linearisation

The governing equations of the vehicle dynamics model developed in the earlier section, can be written in first-order form as

$$\dot{\boldsymbol{X}} = f(\boldsymbol{X}, \boldsymbol{U}) \tag{41}$$

where $\boldsymbol{X}$ is the state vector and $\boldsymbol{U}$ is the control vector.

Next, the above nonlinear model can be linearised by expanding around the equilibrium point $(\boldsymbol{X}_e, \boldsymbol{U}_e)$ as

$$\dot{\boldsymbol{X}}_e + \Delta\dot{\boldsymbol{X}} = f(\boldsymbol{X}_e, \boldsymbol{U}_e) + \frac{\partial f}{\partial \boldsymbol{X}}\Delta\boldsymbol{X} + \frac{\partial f}{\partial \boldsymbol{U}}\Delta\boldsymbol{U} \tag{42}$$

The resultant linearised model is given by

$$\dot{\boldsymbol{X}} = A\boldsymbol{X} + B\boldsymbol{U}; \quad \boldsymbol{Y} = C\boldsymbol{X} \tag{43}$$

where the time-variant linear system matrices (state matrix A and input matrix B) are obtained using Jacobian matrix as (To obtain the linearisation point $(\boldsymbol{X}_e, \boldsymbol{U}_e)$, equation (41) is symbolically written in Matlab to obtain Jacobian matrices using Jacobian function.)

$$A \triangleq \frac{\partial f}{\partial \boldsymbol{X}}\bigg|_{\substack{\boldsymbol{X}=\boldsymbol{X}_e \\ \boldsymbol{U}=\boldsymbol{U}_e}} \in \mathbb{R}^{n\times n}; \quad B \triangleq \frac{\partial f}{\partial \boldsymbol{U}}\bigg|_{\substack{\boldsymbol{X}=\boldsymbol{X}_e \\ \boldsymbol{U}=\boldsymbol{U}_e}} \in \mathbb{R}^{n\times m} \tag{44}$$

where n is the number of states and m is the number of control inputs. In equation (43), the state vector $\boldsymbol{X}$, the control vector $\boldsymbol{U}$ and the output vector $\boldsymbol{Y}$ are given by

For bicycle model, $\boldsymbol{X} = \{ x \quad y \quad \Psi \quad v_x \quad v_y \quad \Omega \}^T$.
For 12-dof model

$$\boldsymbol{X} = \begin{Bmatrix} x & y & z & \phi & \theta & \Psi & z_1 & z_2 & z_3 & z_4 & v_x \\ v_y & v_z & \dot{\phi} & \dot{\theta} & \Omega & \omega_3 & \omega_4 & \dot{z}_1 & \dot{z}_2 & \dot{z}_3 & \dot{z}_4 \end{Bmatrix}^T$$

$$\boldsymbol{Y} = \{ x \quad y \quad v_x \quad v_y \quad \Omega \}^T; \quad \boldsymbol{U} = \{ \delta_{\text{sw}} \quad T \}^T$$

Output matrix, $C \in \mathbb{R}^{o \times n}$ ($o = 5$, the number of outputs). To test the path-planning and tracking algorithm, a scenario of vehicle running at $v_x = 60$ km/h is considered. Correspondingly, the linearisation point $(\boldsymbol{X_e}, \boldsymbol{U_e})$ is:

For bicycle model

$$\boldsymbol{X_e} = \{ 0 \quad 0 \quad 0 \quad v_x \quad 0 \quad 0 \}^T; \quad \boldsymbol{U_e} = \{ \delta_{sw} \; 0 \}^T$$

For 12-dof model

$$\boldsymbol{X_e} = \{ [\mathbf{0}]_{1\times 10} \quad v_x \quad [\mathbf{0}]_{1\times 5} \quad \tfrac{v_x}{r_w} \quad \tfrac{v_x}{r_w} \quad [\mathbf{0}]_{1\times 4} \}^T,$$
$$\boldsymbol{U_e} = \{ \delta_{sw} \; 0 \}^T$$

The ground wheel angle δ is obtained by dividing the handwheel steering angle δ_{sw} by the steering ratio (SR).

Finally, the state matrices are discretised to obtain a discrete time state-space model as

$$\begin{aligned} \boldsymbol{X}(k+1) &= \mathbf{A}_{\text{d}} \boldsymbol{X}(k) + \mathbf{B}_{\text{d}} \boldsymbol{U}(k), \\ \boldsymbol{Y}(k) &= \mathbf{C}_{\text{d}} \boldsymbol{X}(k) \end{aligned} \tag{45}$$

For bicycle model: $\mathbf{A}_{\text{d}} \in \mathbb{R}^{6\times 6}$, $\mathbf{B}_{\text{d}} \in \mathbb{R}^{6\times 2}$ and $\mathbf{C}_{\text{d}} \in \mathbb{R}^{5\times 6}$.

For 12-dof model: $\mathbf{A}_{\text{d}} \in \mathbb{R}^{22\times 22}$, $\mathbf{B}_{\text{d}} \in \mathbb{R}^{22\times 2}$, $\mathbf{C}_{\text{d}} \in \mathbb{R}^{5\times 22}$.

These discrete time matrices are obtained using the zero-order hold (ZOH) discretisation method in Matlab, with an integration step of $\Delta T = 0.05$ s.

Path-planning

The path-planner generates optimal trajectories using a mixed-integer quadratic programming approach with collision avoidance constraints in a receding-horizon fashion.[34]

Collision avoidance

Collision avoidance between two autonomous vehicles is ensured by following linear inequality convex constraints

$$|x^{p_1}(k) - x^{p_2}(k)| \geqslant \Delta_{e_x} \text{ AND } |y^{p_1}(k) - y^{p_2}(k)| \geqslant \Delta_{e_y} \tag{46}$$

Big-M method is used to rewrite the collision avoidance constraints in terms of binary variables $b_r(k)$ as

$$\begin{aligned} & x^{p_i}(k) - x^{p_j}(k) \geqslant \Delta_{e_x} - M b_r(k), \quad r \in \{1-2\} \\ & y^{p_i}(k) - y^{p_j}(k) \geqslant \Delta_{e_y} - M b_r(k), \quad r \in \{3-4\} \\ & \sum_{r=1}^{4} b_r(k) \leqslant 3, \quad M \gg 0 \text{ a large number} \end{aligned} \tag{47}$$

The binary variables $b_r(k) \in \{0, 1\}$ are additional decision variables in the optimisation problem added at each time step k. If $b_r = 0$, then the ith constraint in equation (47) is activated. However, if $b_r = 1$, then that constraint is relaxed.

Remark 1. To achieve obstacle avoidance with static objects, we incorporate the same constraints set as described in equation (47) replacing the planned trajectories $\mathbf{r}_p^{(j)} \triangleq [x_p^{(j)} \; y_p^{(j)}]^{\text{T}} \in \mathbb{R}^2$ with the coordinates of obstacles $\mathcal{O} \in \mathbb{R}^2$.

The location of the obstacles is known. This assumption can be satisfied by using cameras, for example light detection and ranging (LiDAR) system attached to the vehicle.[35]

State-space model

The state-space model used in MPC formulation for the path-planning problem is discussed here. The model for the path-planning stage consists of the following double integrator differential equation describing vehicle dynamics as

$$\dot{x} = v_x, \quad \dot{v}_x = a_x, \quad \dot{y} = v_y, \quad \dot{v}_y = a_y \tag{48}$$

Subsequently the linear discrete time-invariant model is

$$\begin{aligned} & \boldsymbol{X}_p(k+1) = \mathbf{A}_{\text{d}} \boldsymbol{X}_p(k) + \mathbf{B}_{\text{d}} \boldsymbol{U_p}(k), \\ & \boldsymbol{Y}_p(k) = \mathbf{C}_{\text{d}} \boldsymbol{X}_p(k), \text{ where} \end{aligned}$$

$$\mathbf{A}_{\text{d}} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{B}_{\text{d}} = \begin{bmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & \frac{T^2}{2} \\ 0 & T \end{bmatrix}, \quad \mathbf{C}_{\text{d}} = I_4 \tag{49}$$

$\mathbf{A}_{\text{d}} \in \mathbb{R}^{4\times 4}$, $\mathbf{B}_{\text{d}} \in \mathbb{R}^{4\times 2}$ and $\mathbf{C}_{\text{d}} \in \mathbb{R}^{4\times 4}$ are obtained using the ZOH discretisation method, with integration step T.

Collision avoidance is ensured by incorporating integer constraints, and minimising the objective function given by

$$J(\boldsymbol{X_p}, \boldsymbol{U}) = \sum_{j=1}^{N} [\boldsymbol{X_p}(k+j|k) - \overline{\boldsymbol{X_p}}]^{\mathrm{T}} Q[\boldsymbol{X_p}(k+j|k) - \overline{\boldsymbol{X_p}}] + \sum_{j=1}^{M} [\Delta \boldsymbol{U_p}(k+j-1|k)]^{\mathrm{T}} R[\Delta \boldsymbol{U_p}(k+j-1|k)] \tag{50}$$

Overprediction and control horizon of length N and M. $Q \in \mathbb{R}^{m\times m}$ and $R \in \mathbb{R}^{p\times p}$ are symmetrical weighting matrices, with $Q \geqslant 0$ and $R > 0$. The discrete-time state space model of equation (49) can be rewritten in an incremental form as

$$\begin{aligned} \xi(k) &= \begin{bmatrix} \Delta \boldsymbol{X_p}(k) \\ \boldsymbol{Y_p}(k) \end{bmatrix}; \\ \xi(k+1) &= \tilde{\mathbf{A}}\xi(k) + \tilde{\mathbf{B}}\Delta \mathbf{U}(k); \\ \boldsymbol{Y}(k) &= \tilde{\mathbf{C}}\xi(k) \end{aligned} \tag{51}$$

The incremental state matrices are defined as

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_{\mathrm{d}} & \mathbf{0}_{4\times 4} \\ \mathbf{C}_{\mathrm{d}}\mathbf{A}_{\mathrm{d}} & \mathbf{I}_4 \end{bmatrix} \quad \tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{B}_{\mathrm{d}} \\ \mathbf{C}_{\mathrm{d}}\mathbf{B}_{\mathrm{d}} \end{bmatrix} \quad \tilde{\mathbf{C}} = \begin{bmatrix} \mathbf{0}_{4\times 4} \\ \mathbf{I}_4 \end{bmatrix}^T \tag{52}$$

where $\Delta \boldsymbol{X_p}(k) = \boldsymbol{X_p}(k) - \boldsymbol{X_p}(k-1)$ is incremental state vector, $\Delta \boldsymbol{U} = \boldsymbol{U}(k) - \boldsymbol{U}(k-1)$ is the control input vector, $\mathbf{I}$ and $\mathbf{0}$ are identity and zero matrices of appropriate sizes.

Once the optimal solution is calculated, the set of waypoints serves as the reference trajectory $\bar{\mathbf{r}}_t$ for tracking controller.

Prediction model

The prediction model following the incremental state-space formulation,[36] can be obtained as

$$X_N = \mathcal{G}\Delta \mathcal{U}_M + \mathcal{F} \tag{53}$$

where $X_N \in \mathbb{R}^{4N\times 1}$ stacks the predicted values of the state vector along N, $\Delta \mathcal{U}_M \in \mathbb{R}^{2M\times 1}$ stacks the incremental control inputs along M,

$$\mathcal{G} \triangleq \begin{bmatrix} \tilde{\mathbf{C}}\tilde{\mathbf{B}} & \mathbf{0}_{4\times 2} & \dots & \mathbf{0}_{4\times 2} \\ \tilde{\mathbf{C}}\tilde{\mathbf{A}}\tilde{\mathbf{B}} & \tilde{\mathbf{C}}\tilde{\mathbf{B}} & \dots & \mathbf{0}_{4\times 2} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{C}}\tilde{\mathbf{A}}^{M-1}\tilde{\mathbf{B}} & \tilde{\mathbf{C}}\tilde{\mathbf{A}}^{M-2}\tilde{\mathbf{B}} & \dots & \tilde{\mathbf{C}}\tilde{\mathbf{B}} \\ \vdots & \vdots & & \vdots \\ \tilde{\mathbf{C}}\tilde{\mathbf{A}}^{N-1}\tilde{\mathbf{B}} & \tilde{\mathbf{C}}\tilde{\mathbf{A}}^{N-2}\tilde{\mathbf{B}} & \dots & \tilde{\mathbf{C}}\tilde{\mathbf{A}}^{N-M}\tilde{\mathbf{B}} \end{bmatrix} \in \mathbb{R}^{4N\times 2M}$$

$$\mathcal{F} \triangleq \begin{bmatrix} \tilde{\mathbf{C}}\tilde{\mathbf{A}} \\ \tilde{\mathbf{C}}\tilde{\mathbf{A}}^2 \\ \vdots \\ \tilde{\mathbf{C}}\tilde{\mathbf{A}}^{\mathrm{N}} \end{bmatrix} \xi(k) \in \mathrm{R}^{4N}$$

Road constraints

The road constraint for the vehicle position $\mathbf{r}_p$ is defined using $\mathbf{r}_{\min}$, $\mathbf{r}_{\max} \in \mathrm{R}^2$, which depict the minimum and maximum values of the x–y coordinates of the road via the following linear convex inequality constraint

$$\mathbf{r}_{\min} \leqslant \mathbf{r}_p \leqslant \mathbf{r}_{\max} \tag{54}$$

Repeating equation (54) along the prediction horizon N, yields

$$[\mathbf{r}_{\min}]_N \leqslant X_N \leqslant [\mathbf{r}_{\max}]_N \tag{55}$$

which can be rewritten in terms of $\Delta \mathcal{U}_M$ using equation (53), providing

$$\mathcal{A}\Delta \mathcal{U}_M \leqslant \gamma \tag{56}$$

where

$$\mathcal{A} \triangleq \begin{bmatrix} -\mathcal{G} \\ \mathcal{G} \end{bmatrix} \in \mathbb{R}^{4N\times 2M}; \quad \text{and} \quad \gamma \triangleq \begin{bmatrix} [\mathbf{r}_{\max}]_N - \mathcal{F} \\ \mathcal{F} - [\mathbf{r}_{\min}]_N \end{bmatrix} \in \mathrm{R}^{4N}$$

Note that in this study only straight roads are considered to formulate the road boundary constraints. The model responses are evaluated for vehicle running on a *three-lane* straight road though the same formulation can be used to model more complex scenarios of curved roads too with appropriate modifications for road geometry, curvature, etc.

Collision avoidance constraints

The inequality convex constraints for the distances between the autonomous vehicle and the HDV in matrix form can be written as (recall equation (46)),

$$\mathbf{r}_p - \mathbf{r}^{(j)} \geqslant \Delta \tag{57}$$

The vector $\Delta \in \mathbb{R}^2$ corresponds to the error boundaries Δ_x, Δ_y. Repeating equation (57) along the prediction horizon N yields

$$X_N - [\mathbf{r}^{(j)}]_N \geqslant [\Delta]_N \tag{58}$$

where $[\mathbf{r}^{(j)}]_N \in \mathbb{R}^{2N\times 1}$ stacks the value of the surrounding vehicle position along the prediction horizon N.

Replacing the prediction model of equation (53) into equation (58), and rewriting in terms of the optimisation vector $\mathcal{X} \in \mathbb{R}^{(2M+n_dN)}$, with n_d representing the number of binary variables, one can obtain

$$\mathcal{T}\mathcal{X} \leqslant \varrho \tag{59}$$

where

$$\mathcal{X} \triangleq \begin{bmatrix} \Delta\mathcal{U}_M \\ b_i \end{bmatrix}; \quad \mathcal{T} \triangleq \begin{bmatrix} -\mathcal{G} & -M\mathbf{I}_{\frac{n_d}{2}N} & \mathbf{0}_{\frac{n_d}{2}N\times\frac{n_d}{2}N} \\ \mathcal{G} & \mathbf{0}_{\frac{n_d}{2}N\times\frac{n_d}{2}N} & -M\mathbf{I}_{\frac{n_d}{2}N} \end{bmatrix}$$

with $\mathcal{T} \in \mathbb{R}^{n_d N\times(2M+n_d N)}$ and the vector ϱ is expressed by

$$\varrho \triangleq \begin{bmatrix} -[\Delta]_N + \mathcal{F} - [\mathbf{r}^{(j)}]_N \\ -[\Delta]_N - \mathcal{F} + [\mathbf{r}^{(j)}]_N \end{bmatrix} \in \mathbb{R}^{n_d N}$$

Guaranteeing activation of at least one constraint of (47) by

$$\begin{bmatrix} \mathbf{0}_{N\times 2M} & \left[\mathbf{1}_{n_d N}\right]^{\mathrm{T}} \end{bmatrix} \mathcal{X} \leqslant [4]_N$$

Path-planning controller

In this work, both path-planner and tracking controllers obtain optimal control vector $\mathcal{U}^*(k)$ by minimising cost function of the form in equation (50). Rewritten using notation matrix as

$$\min_{\mathcal{X}} J(k) = \frac{1}{2}\mathcal{X}^{\mathrm{T}}\mathcal{H}\mathcal{X} + \mathcal{M}^T\mathcal{X} \tag{60}$$

where the optimisation vector $\mathcal{X} \triangleq [\Delta\mathcal{U}_M\ b_i]^{\mathrm{T}}$, is augmented by additional binary decision variables b_i.

The matrix $\mathcal{H}$ and the vector ϑ define the terms of the cost function as follows

$$\frac{1}{2}\mathcal{H} = \mathcal{G}^{\mathrm{T}}\mathcal{Q}\mathcal{G} + \mathcal{R}, \mathcal{M}^T = 2(\mathcal{F} - [\bar{\mathbf{r}}_p^{(i)}]_N)^{\mathrm{T}}\mathcal{Q}\mathcal{G} \tag{61}$$

where the controlled output weighting matrix is $\mathcal{Q} = \eta \times \mathbf{I}_{4N}$ and the control input weighting matrix is $\mathcal{R} = \rho \times \mathbf{I}_{2M}$.

The optimal control vector $\mathcal{U}^*(k)$ computed at the discrete-time instant k is given by $\mathcal{U}^*(k) = \Delta\mathcal{U}^*(k) + \mathcal{U}^*(k-1)$, where $\Delta\mathcal{U}^*(k)$ is the first control vector of $\Delta\mathcal{U}_M^*$, which in turn is obtained by solving the optimisation problem in (61) subject to the constraints

$$\mathcal{S}\mathcal{X} \leqslant \vartheta \tag{62}$$

The matrix $\mathcal{S}$ and the vector ϑ define the linear inequality constraint set on the optimisation variables

$$\mathcal{S} \triangleq \begin{bmatrix} \mathcal{A} \\ \mathcal{T} \end{bmatrix} \in \mathbb{R}^{(n_d+4)N\times(2M+(n_d)N)}, \quad \vartheta \triangleq \begin{bmatrix} \gamma \\ \varrho \end{bmatrix} \in \mathbb{R}^{(n_d+4)N}$$

Tracking controller

An adequate mathematical model of the system is an important step in designing a control system. A plant cannot be stabilised if the model description is inadequate.

The tracking controller needs an accurate description of vehicle dynamic characteristics in order to control the vehicle behavior. In the present work, we have used a linear MPC to design the tracking controller. The controller performance is evaluated for two different plant models, namely, the bicycle model and the 12-dof model.

Tracking references

As discussed in previous section, the outputs of both the linear vehicle models are $\boldsymbol{Y} = \begin{Bmatrix} x & y & v_x & v_y & \Omega \end{Bmatrix}^T$. In order to map these output variables a set of reference inputs have to be assigned to the tracking controller $\boldsymbol{Y}^{\mathrm{ref}}$. The first four reference inputs namely the longitudinal and lateral positions and velocities are obtained from the path-planning MPC as shown in Figure 1. The last input is the reference yaw rate Ω^{ref}, which is obtained using a steady-state yaw rate command.[37] Though a dynamic scenario is considered, yet the steady-state (instantaneous) yaw rate is useful in giving an indication of vehicle handling behaviour.[38] It is a function of the longitudinal velocity v_x, vehicle characteristic speed V_{ch}, steering angle δ and wheelbase l and is given by

$$\Omega^{\mathrm{ref}} = \Omega^{\mathrm{ss}} = \frac{v_x\delta}{l\left(1 + \left(\frac{v_x}{V_{ch}}\right)^2\right)}; \quad V_{ch} = \sqrt{\frac{gl}{K_{us}}} \tag{63}$$

In equation (63), K_{us} is the understeering gradient. The value of K_{us} for the vehicle under study is $0.95\,^\circ/\mathrm{g}$.

Cost function

The objective function to be minimised to ensure the vehicle follows desired trajectories is the same as given in equation (50) with the difference that the states here are $\boldsymbol{X} \in \mathbb{R}^6$ and R^{22} for the bicycle and 12-dof model respectively instead of $\boldsymbol{X_p} \in \mathbb{R}^4$ and $\boldsymbol{U} = \{T \quad \delta_{sw}\}^T$ instead of $\boldsymbol{U_p} = \{a_x^p \quad a_y^p\}^T$.

Prediction model

The prediction model remains the same as given in equation (53) with appropriate change in dimensions of system state matrices of vehicle models instead of path-planner.

Control constraints

The control constraints are incorporated in the scheme and are written in terms of $\mathcal{U}$ as

$$\mathcal{U}_{\min} \leqslant \mathcal{U} \leqslant \mathcal{U}_{\max},\ \mathcal{U}_{\min} = [\underline{\delta}_{\mathrm{sw}}\ \underline{T}]^{\mathrm{T}}, \quad \mathcal{U}_{\max} = [\bar{\delta}_{\mathrm{sw}}\ \bar{T}]^{\mathrm{T}} \tag{64}$$

where $\mathcal{U}_{\min}$ and $\mathcal{U}_{\max}$ are the bounds on the control signals. Rewriting the constraint (64) in terms of $\Delta\mathcal{U}$ we obtain

$$\mathcal{U}(k) = [\mathcal{U}(k-1)]_N + \mathcal{T}_N \Delta\mathcal{U}(k) \tag{65}$$

where

$$\mathcal{T}_N = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0}_{2\times2} & \cdots & \mathbf{0}_{2\times2} \\ \mathbf{I}_2 & \mathbf{I}_2 & \cdots & \mathbf{0}_{2\times2} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{I}_2 & \mathbf{I}_2 & \cdots & \mathbf{I}_2 \end{bmatrix} \in \mathbb{R}^{2N\times2M}$$

Table 2. Parameters of the MPC path-planner.

Variables	Values
Prediction horizon:	$N = 40$
Control horizon:	$M = 5$
Sampling time:	$\Delta T = 0.05$ s
Control output weights:	$\eta = [1\ 1\ 1\ 1\ 1]^{\mathsf{T}}$
Control input weights:	$\rho = [20\ 15]^{\mathsf{T}}$
Control constraints:	$\underline{\delta}_{sw} = -630^{\circ}$, $\underline{T} = 0$ N m
	$\bar{\delta}_{sw} = 630^{\circ}$, $\bar{T} = 400$ N m

From equations (65) and (64) we obtain

$$[\mathcal{U}_{\min}]_N \leqslant [\mathcal{U}(k-1)]_N + \mathcal{T}_N \Delta\mathcal{U}(k) \leqslant [\mathcal{U}_{\max}]_N \tag{66}$$

Finally, equation (66) can be written in terms of $\Delta\mathcal{U}_M$ as

$$\Xi\ \Delta\mathcal{U}_M \leqslant \Phi \tag{67}$$

where $\Xi \in \mathbb{R}^{4N\times2M}$ and $\Phi \in \mathbb{R}^{4N}$ are given by

$$\Xi \triangleq \begin{bmatrix} \mathcal{T}_N \\ -\mathcal{T}_N \end{bmatrix}, \qquad \Phi = \begin{bmatrix} [\mathcal{U}_{\max}]_N - [\mathcal{U}(k-1)]_N \\ [\mathcal{U}(k-1)]_N - [\mathcal{U}_{\min}]_N \end{bmatrix}$$

Model predictive controller

Finally, we obtain an optimal control vector $\mathcal{U}^*(k)$ by minimising the cost function in equation (50) subjected to the additional control constraints but without considering mixed-integer collision avoidance constraints

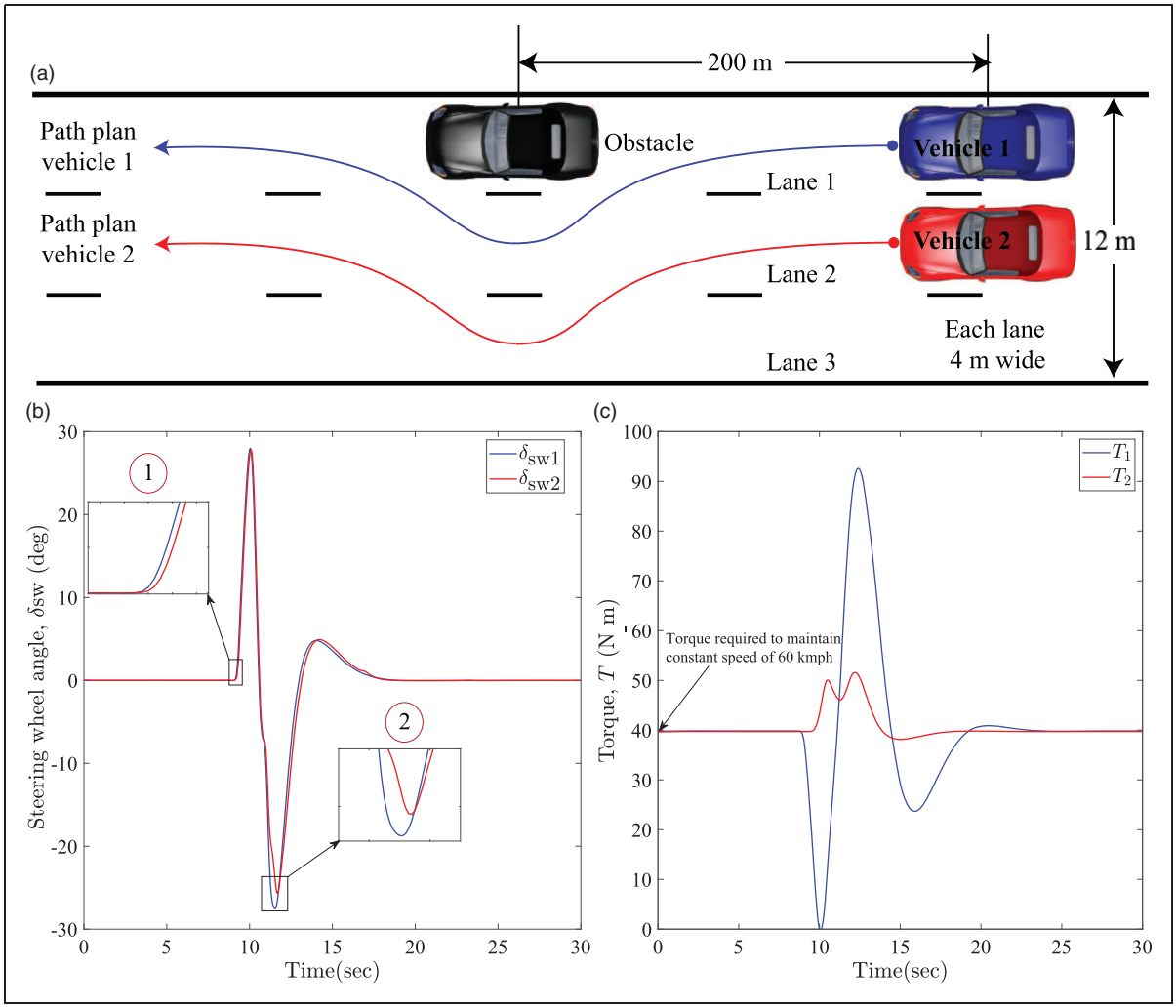

Figure 12. (a) Scenario description: double lane change event; (b) steering wheel angle; (c) engine torque.

$$\min_{\Delta\mathcal{U}} J(k) = \frac{1}{2}\Delta\mathcal{U}^{\mathrm{T}}\mathcal{H}\Delta\mathcal{U} + \mathcal{M}^{T}\Delta\mathcal{U} \tag{68}$$

subject to the control constraints in equation (67).

The objective function of equation (68) is minimised using Matlab's built-in function quadprog to obtain the optimisation vector $\Delta\mathcal{U}_M$ for the tracking controller. For path-planning, the mixed-integer quadratic programing problem needs to be solved in order to take care of the collision avoidance constraints. For this purpose, cplexmiqp solver from IBM[39] is used in Matlab to calculate the optimisation vector.

MPC and control constraint parameters are given in Table 2. The sampling time of 0.05 s is chosen in view of real-time implementation capabilities of the proposed controller. A large prediction horizon N and a small control horizon M was chosen for less aggressive control action. Moreover, small M results in computation of fewer variables in the QP solved at each control interval, thereby promoting faster computations.[40] $M < N$ is essential otherwise some manipulated variable moves might not affect any of outputs and small M promotes an internally stable controller. As a rule of thumb M and N should be chosen as $5 \leqslant M \leqslant 20$ and $N/3 < M < N/2$.[41] The optimisation was performed in MATLAB using the cplex solver from IBM, which can handle binary variables to solve the MIQP problem. For the 12-dof model, the optimisation is computed in 150 ms to 600 ms using an Intel Core i7-6700U CPU clocked at 3.40GHz with 16GB of RAM. The simulation time increases to around 600 ms when the vehicle approaches the obstacle as the controller has to compute steering and torque control commands. Once the vehicle is past the obstacle the simulation time decreases drastically. The code execution time is faster for the bicycle model and it varies from 100 ms to 500 ms. It is evident that the 12-dof model is fast enough for real-time application.

The actuator constraints are set according to the saturation limits. An equal weight is given to all control outputs η in order to track all vehicle dynamics characteristics ($X, Y, V_x, V - y$ and Ω) with an equal precision. A higher value of weight is given to the steering angle as opposed to the vehicle torque in the input control weighing ρ. This is because of the fact that naturally a driver gives more emphasis in steering the vehicle δ_{sw} than braking (torque T) while avoiding the obstacle. The steering wheel turns three and half turns from one end to the other correspondingly the lower and upper bounds of the steering angle are calculated as $\pm 630^{\circ}$. The peak driveline torque is 400 N m.

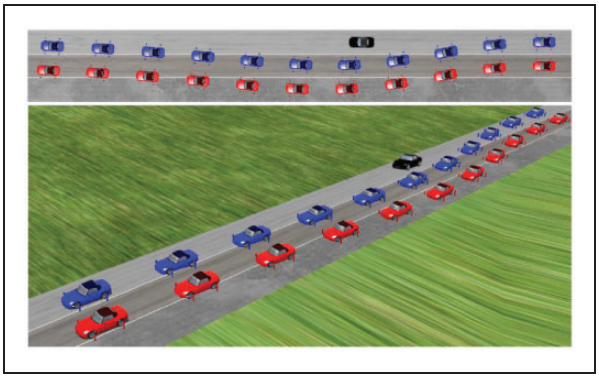

Figure 13. Cooperative double lane change event in CarSim.

Results

The scenario description for the cooperative obstacle avoidance manoeuvre is shown in Figure 12(a). The vehicles are sharing plans with one another. Vehicle 1 is in the first lane ($x = 0, y = 0$) and vehicle 2 is in the second lane ($x = 0, y = 4$). The obstacle is at a distance of 200 m and initial velocity of both vehicles is 60 km/h.

The co-simulation study was performed using the path-planning and tracking controller model and the nonlinear complex vehicle dynamics model built in CarSim. Both the vehicles are able to negotiate the obstacle in a cooperative manner. The results are discussed below.

The control inputs (steering δ_{sw} and torque T) are shown in Figure 12(b) and (c). Vehicle 1 steers to the left as it first encounters the obstacle. Once it starts to steers in lane 2, vehicle 2 starts to steer in the third lane to make way for the first vehicle (see the inset 1 in Figure 12(b)). After vehicle 1 has negotiated the obstacle, it returns back to its lane and requires a larger steering input then vehicle 2. It is because of the fact that vehicle 2 started changing its lane after vehicle 1, so it has to take less corrective action to return back to its original lane (see 2 in Figure 12(b)).

Initially, both vehicles start their motion with the same torque. This is because the vehicle has to overcome the rolling resistance and aerodynamic drag which is roughly equal to 125 N for the vehicle speed of 60 km/h. This force has to be counteracted by an engine torque of 40 N m in order to maintain a constant speed of 60 km/h. Vehicle 1 brakes more aggressively as it sees the obstacle first as a result the torque decreases. Once it negotiates the obstacle by turning towards the left, the torque increases again to avoid the obstacle quickly. Eventually it comes back in the same lane and the torque again becomes constant, i.e. 40 N-m (torque required to maintain the same reference preset speed of 60 km/h). Vehicle 2 on other hand just swivels out on lane 3 and its torque increases to avoid colliding vehicle 1.

The trace of vehicle trajectory obtained from the animation interface of CarSim cosimulation is shown in Figure 13. The tracking controller responses are shown in Figure 14. The model outputs (shown

using solid lines) are compared against the references (shown with broken lines) generated by the path-planner and steady-state yaw rate reference.

The vehicle path $x - y$ trajectories are shown in Figure 14(a). Vehicle 1 starts in lane 1 and vehicle 2 starts in lane 2. Both the vehicles share each other trajectories (x, y coordinates and longitudinal and lateral velocities v_x, v_y). Vehicle 1 can foresee the stationary obstacle in addition to the shared states of vehicle 2. As it encounters the obstacle, it steers to left moves in the second lane and eventually returns back to the same lane. At the same time, vehicle 2 moves in the third lane in order to make way for vehicle 1 and returns back to its lane after the later has negotiated the obstacle. The important thing to note here is that the motion of vehicle 2 is delayed as it takes steering action only when vehicle 1 starts manoeuvring into its lane. Both vehicles are able to track the reference trajectories and a smooth lane change manoeuvre is made. The tracking errors are shown in Figure 15.

The vehicle longitudinal velocity is shown in Figure 14(b). The velocity of vehicle 1 decreases initially because the torque input decreases as it approaches the obstacle. After it negotiates the obstacle, its velocity increases and returns back to the preset reference value of 60 km/h. The velocity variation for vehicle 2 is not substantial. Both the vehicles are able to track the lateral velocity and yaw rate inputs sufficiently well as shown in Figure 14(c) and (d).

The evolution of other remaining states of the 12-dof model for both vehicle 1 (solid-lines) and vehicle 2 (broken-lines) is shown in Figure 16. The CG displacement z shows a minor variation due to load transfer. As the vehicle encounters the obstacle, it swivels out of the lane towards left. Due to the lateral load transfer, the outer left wheels are compressed and the inner right wheels jack up as seen in wheel jounce responses. As the vehicle rolls out of the turn (anti-clockwise direction), the roll angle is positive and a little over 1°. The vehicle pitch also shows a similar trajectory though it value is very small as the vehicle is operating nearly at a constant speed. The rear wheel speeds initially decreases because of reduction in wheel torques as the vehicle approaches obstacle. They regain their initial preset references once the vehicle pasts obstacle.

As suggested by an anonymous reviewer, the lateral acceleration and lateral and vertical wheel forces resulting from load transfers are shown in Figure 17.

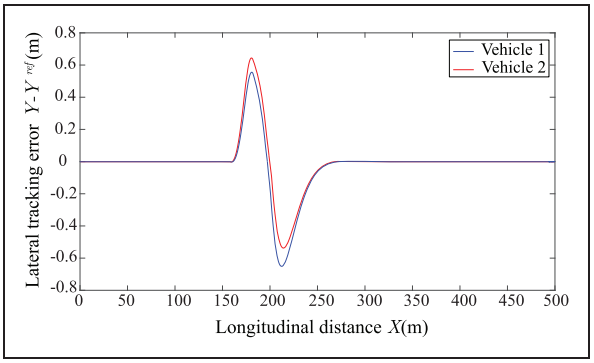

Figure 15. Lateral tracking errors of vehicles 1 and 2.

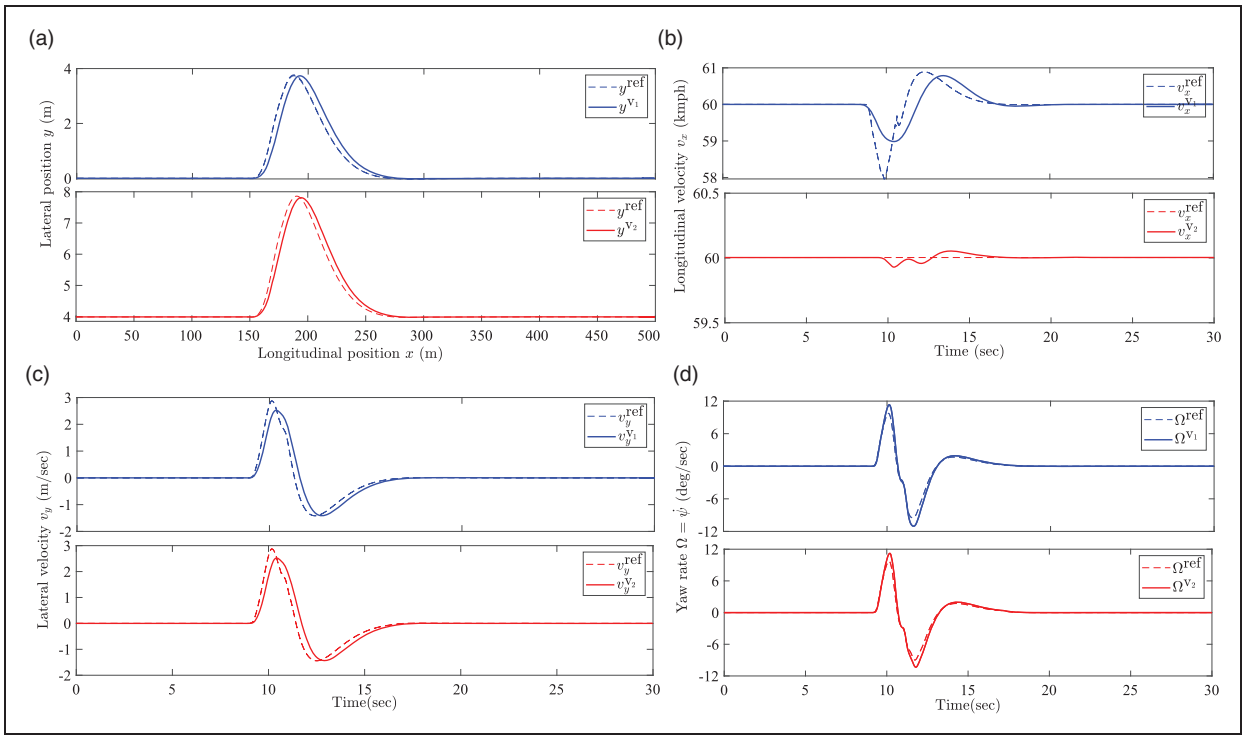

Figure 14. Comparison of model outputs against the reference inputs for the cooperative path planning and tracking controller: (a) Longitudinal versus lateral position; (b) Longitudinal velocity; (c) Lateral velocity; (d) Yaw rate.

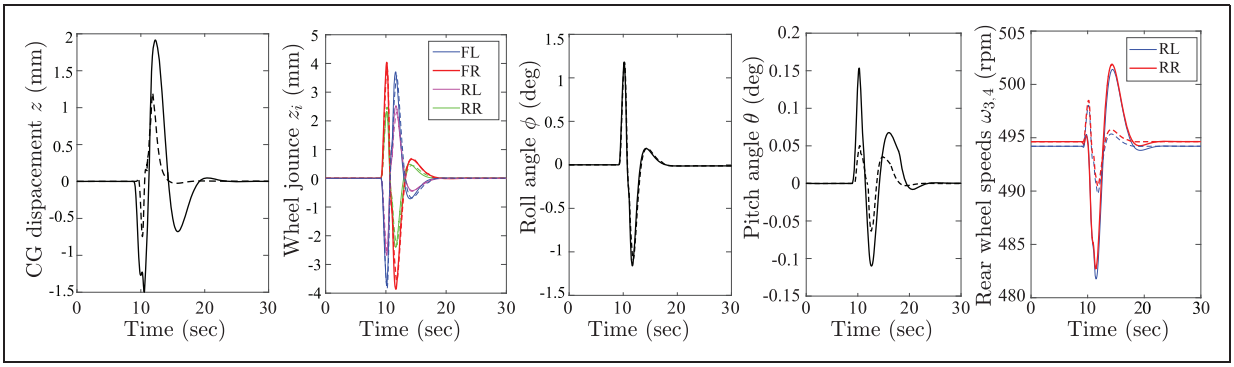

Figure 16. Evolution of the remaining vehicle states for the cooperative path planning and tracking controller using 12-dof model.

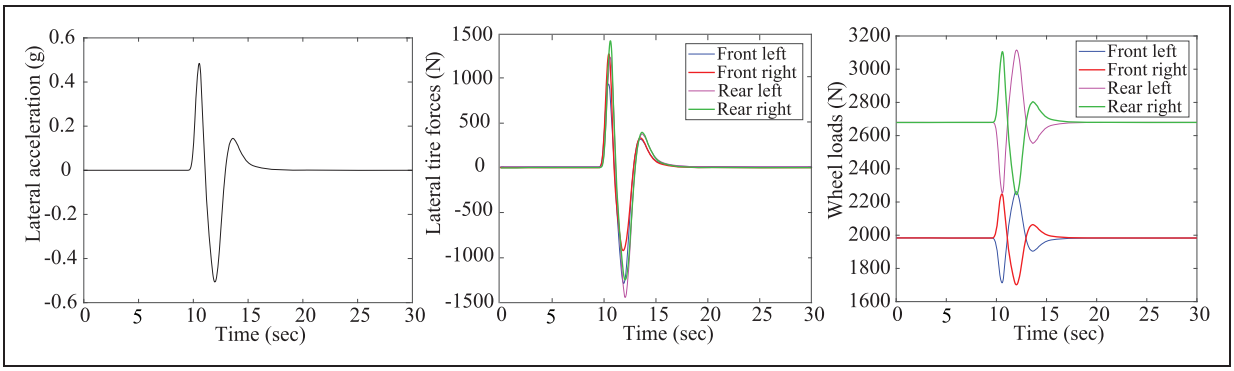

Figure 17. Lateral acceleration, lateral and vertical tyre forces for vehicle 1 undergoing the obstacle avoidance manoeuvre.

It gives the reader an idea of the magnitude of forces encountered in a normal on-road driving scenario far away from the boundaries of the dynamic envelope.

Model evaluations

In this section we investigate the performance of our model at different speeds and compare the responses obtained from our model against the bicycle model predictions.

Performance evaluation at different speeds. To evaluate the performance of the 12-dof model for a range of operating speeds we have compared the tracking controller results at 40, 60 and 80 km/h in Figure 18(a), (b) and (c), respectively. For this we used different initial conditions of vehicle velocity v_x for linearing the 12-dof model as $\boldsymbol{X_e} = \left\{ [\mathbf{0}]_{1\times 10} \quad v_x \quad [\mathbf{0}]_{1\times 5} \quad \frac{v_x}{r_w} \quad \frac{v_x}{r_w} \quad [\mathbf{0}]_{1\times 4} \right\}^T$. A brief discussion of performance evaluation results is given below.

1. Vehicle longitudinal and lateral positions $X - Y$.
2. The vehicle is able to track reference path trajectory sufficiently well in all three cases. A sufficiently smooth double lane change manoeuvre is performed. The tracking is better at low speed and inferior at high speeds but still the match is pretty good. This is obvious because of lag in following the reference path at high speeds because of the second-order actuator dynamics.
3. Lateral velocity v_y.
4. The vehicle is able to track the lateral velocity quite well in all the three cases. The lateral velocity decreases as speed increases.
5. Yaw rate Ω.
6. The vehicle roughly maps the yaw rate input. The reason for the in all three cases mismatch is because a steady-state yaw rate measure is used as a tracking reference which is slightly less than dynamic yaw rate.

Comparison against the bicycle model. The vehicle dynamic responses ($\delta_{sw}, x, y, v_x, v_y$ and Ω) of the 12-dof model are compared against the bicycle model for vehicle 1 for the above mentioned obstacle avoidance scenario in Figure 19.

The steering angle input computed by the tracking controller using bicycle model is on the higher side as compared with the 12-dof model (see Figure 19(a)). We have earlier seen in Model comparisons section that for the same steering angle input the dynamic behaviour of the vehicle obtained from the 12-dof model matches closely with the sophisticated CarSim model. This implies that if higher values of steering wheel angle are used in a real vehicle according to bicycle model predictions it will result in high values of other dynamic characteristics, which will eventually result in inaccurate tracking of the reference trajectories.

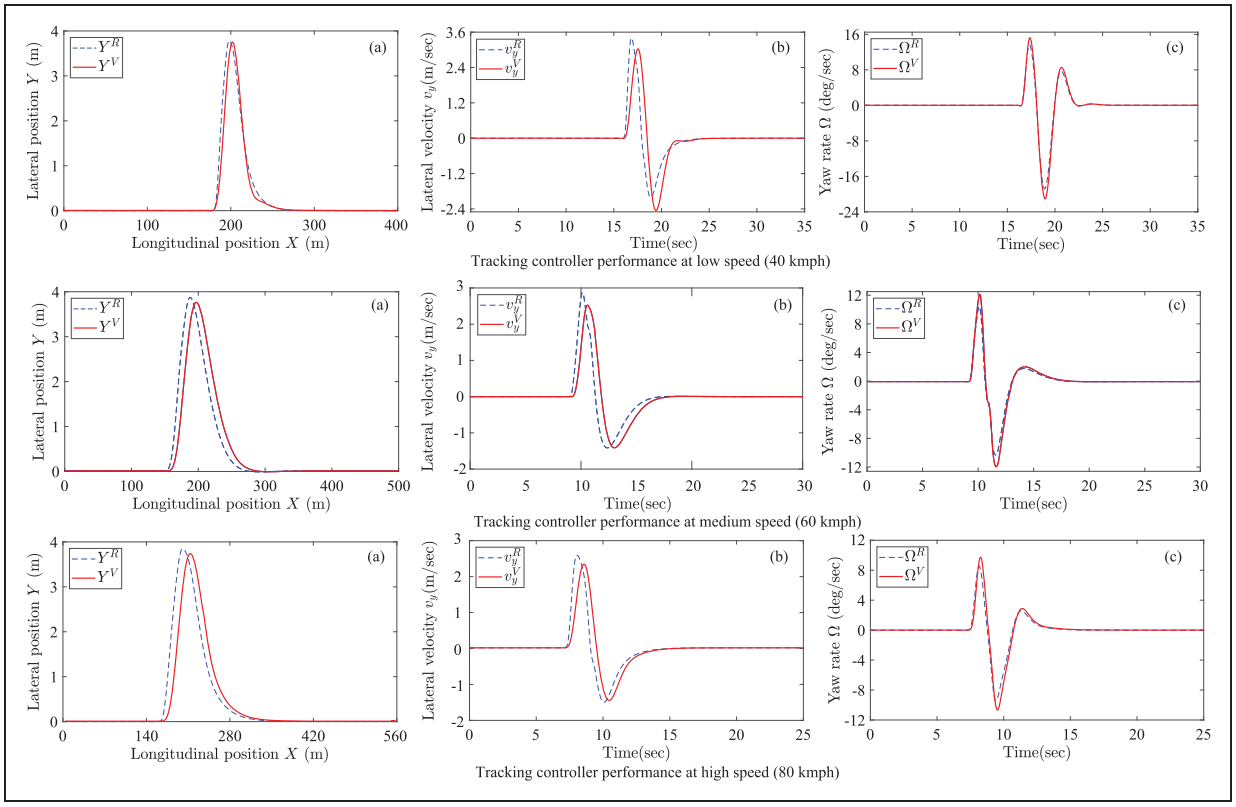

Figure 18. Tracking controller responses at different speeds: (a) Longitudinal versus lateral position; (b) Lateral velocity; (c) Yaw rate.

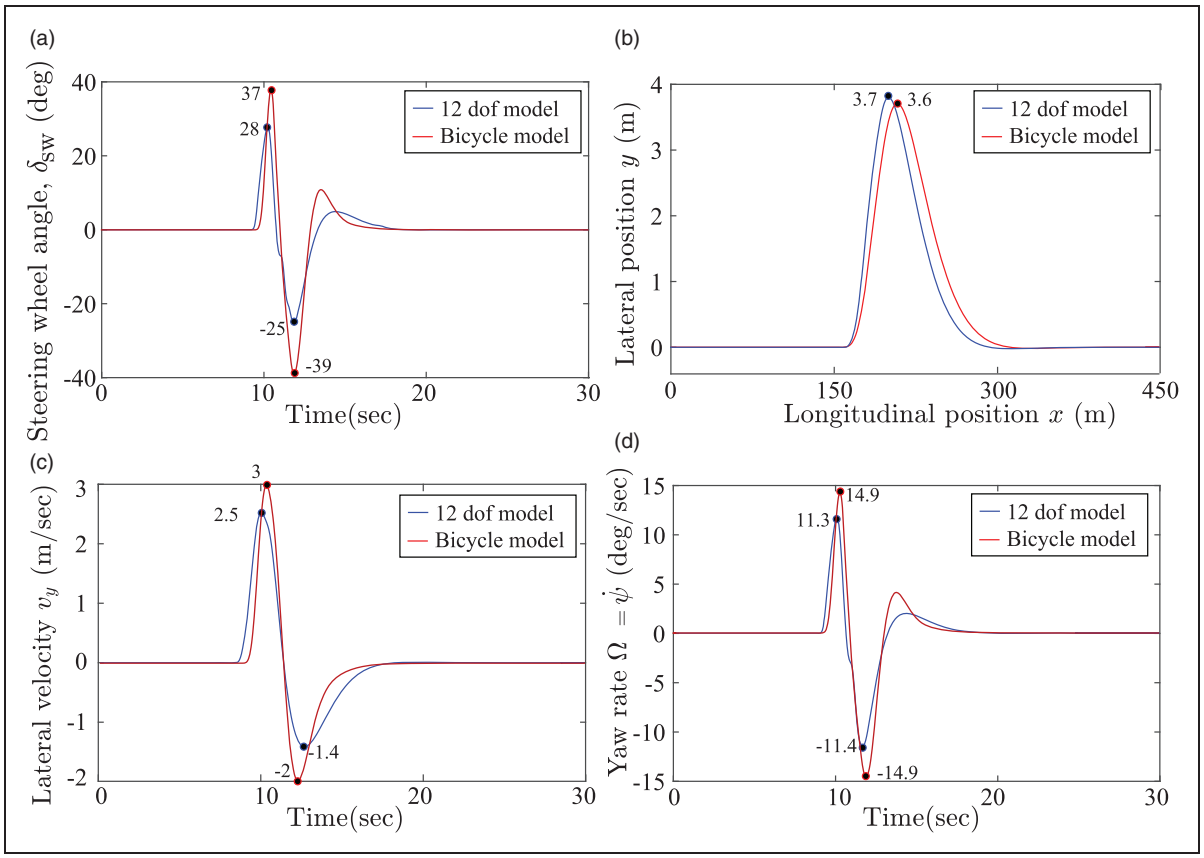

Figure 19. Comparison of model responses of 12 dof model and bicycle model: (a) Steering wheel angle; (b) Lateral versus longitudinal position; (c) Lateral velocity; (d) Yaw rate.

Regarding path tracking, the bicycle model underpredicts the lateral position as compared to the high fidelity model as shown in Figure 19(b) even though it uses a high value of vehicle steering input. This justifies the need of a model of increased complexity for tracking the vehicle trajectory.

There is a substantial amount of difference in the lateral velocity v_y and yaw rate Ω responses between the two models (see Figure 19(c) and (d)). The reason for this mismatch is the unmodelled lateral dynamics of the bicycle model as opposed to the 12-dof model. For instance, refer to equation (5) and $\hat{f}$ component of $\boldsymbol{a}$ in equation (17).

Bicycle model, $a_y = V\dot{\beta} + \dot{V}\beta + \Omega V$.

$$12-\text{dof}: a_y = \dot{v}_y - d_r\ddot{\phi} + 2d_p\dot{\theta}\Omega + d_p\theta\dot{\Omega} + v_x\Omega + d_r\phi\Omega^2$$

The effect of pitch θ and roll ϕ angle is evident on the value of lateral acceleration a_y, which is missing in the bicycle model but captured in the 12-dof model. This shows the superiority of the 12-dof model in tracking the reference trajectories generated by the path-planning MPC.

Conclusion and future work

This work gives a useful insight by evaluating the controller performance for different models. The high fidelity nonlinear vehicle model of intermediate complexity inclusive of roll, pitch and suspension jounce motions is developed. In addition to this, two simplified models namely the kinematic model and the bicycle model were also developed. The models have been compared for a single lane change manoeuvre and it has been established that the 12-dof model emulates the real vehicle dynamic characteristics fairly well. This nonlinear model is subsequently linearised to take an account of different operating conditions.

In the next part, the paper addressed the problem of path-planning and tracking controller for an autonomous vehicle considering the trajectories of other autonomous vehicle. The problem is solved using a hierarchical MPC design with MIQP at the planning layer. Integer constraints are incorporated to ensure collision avoidance between the autonomous vehicles and stationary obstacle. Control actuation constraints have also been incorporated in the proposed approach. To evaluate the effectiveness of the method, numerical simulations are performed for a cooperative double lane change manoeuvre, which demonstrate the effectiveness of the proposed method. The tracking controller is able to follow the references and the method is quite appropriate to attain the on-road autonomous driving environment. This is because the linear tracking controller uses only 22 state variables to control the high fidelity nonlinear vehicle model in CarSim.

In future, the current method with linear MPC will be compared with nonlinear MPC and other alternatives will be evaluated to account for probabilistic trajectories and soft constraints. An initial investigation of these concepts has been evaluated by the Viana et al.[42] Moreover the goal is to demonstrate cooperative autonomy in an urban environment where the obstacles are moving human-driven vehicles as opposed to stationary obstacles. This work will be extended to incorporate moving obstacles with probabilistic trajectories coming from the human driver model.[43]

Finally, this algorithm is planned to be implemented in a real-time vehicle operation and that can be taken up as the future developmental work.

Acknowledgements

This project comprises a consortium with six partners, namely, Applus IDIADA, Cosworth Electronics, Connected Places Catapult, Westfield Sportscars, SBD Automotive and Cranfield University. This work is planned to be implemented in real vehicles by the end of the project.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work has been supported by Innovate UK within the scope of the project grant MuCCA 'Multi-Car Collision Avoidance'.

References

1. Wang D, Hu M, Wang Y, et al. Model predictive control-based cooperative lane change strategy for improving traffic flow. *Adv Mech Eng* 2016; 8: 1687814016632992.
2. Hassanzadeh M, Lidberg M, Keshavarz M, et al. Path and speed control of a heavy vehicle for collision avoidance manoeuvres. In: *2012 IEEE intelligent vehicles symposium*, 2012, pp.129–134. New York: IEEE.
3. Qian X, Altché F, Bender P, et al. Optimal trajectory planning for autonomous driving integrating logical constraints: an MIQP perspective. In: *2016 IEEE 19th international conference on intelligent transportation systems (ITSC)*, 2016, pp.205–210. New York: IEEE.
4. Shim T, Adireddy G and 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 Automobile Engineering* 2012; 226: 767–778.
5. Suzuki T, Usami R and Maekawa T. Automatic two-lane path generation for autonomous vehicles using quartic b-spline curves. *IEEE Trans Intell Veh* 2018; 3: 547–558.
6. Khatib M, Jaouni H, Chatila R, et al. Dynamic path modification for car-like nonholonomic mobile robots.

In: *Proceedings of international conference on robotics and automation*, vol. 4, 1997, pp.2920–2925. New York: IEEE.
7. Ji J, Khajepour A, Melek W, et al. Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints. *IEEE Trans Veh Technol* 2017; 66: 952–964.
8. Schouwenaars T, De Moor B, Feron E, et al. Mixed integer programming for multi-vehicle path planning. In: *2001 European control conference (ECC)*, 2001, pp.2603–2608. New York: IEEE.
9. Frese C and Beyerer J. A comparison of motion planning algorithms for cooperative collision avoidance of multiple cognitive automobiles. In: *2011 IEEE intelligent vehicles symposium (IV)*, 2011, pp.1156–1162. New York: IEEE.
10. Rovira-Más F and Zhang Q. Fuzzy logic control of an electrohydraulic valve for auto-steering off-road vehicles. *Proc IMechE, Part D: J Automobile Engineering* 2008; 222: 917–934.
11. Leonard J, How J, Teller S, et al. A perception-driven autonomous urban vehicle. *J Field Robot* 2008; 25: 727–774.
12. Urmson C, Ragusa C, Ray D, et al. A robust approach to high-speed navigation for unrehearsed desert terrain. *J Field Robot* 2006; 23: 467–508.
13. Levinson J, Askeland J, Becker J, et al. Towards fully autonomous driving: Systems and algorithms. In: *2011 IEEE intelligent vehicles symposium (IV)*, 2011, pp.163–168. New York: IEEE.
14. Tabatabaei Oreh S, Kazemi R and Azadi S. A sliding-mode controller for directional control of articulated heavy vehicles. *Proc IMechE, Part D: J Automobile Engineering* 2014; 228: 245–262.
15. Nam K, Oh S, Fujimoto H, et al. Robust yaw stability control for electric vehicles based on active front steering control through a steer-by-wire system. *Int J Autom Technol* 2012; 13: 1169–1176.
16. Kanchwala H and Bordons C. Improving handling performance of an electric vehicle using model predictive control. SAE Technical Paper 2015-01-0082, 2015.
17. Kanchwala H, Viana ÍB, Ceccoti M, et al. Model predictive tracking controller for a high fidelity vehicle dynamics model. In: *2019 IEEE intelligent transportation systems conference (ITSC)*, 2019, pp.1496–1503. New York: IEEE.
18. Kong J, Pfeiffer M, Schildbach G, et al. Kinematic and dynamic vehicle models for autonomous driving control design. In: *2015 IEEE intelligent vehicles symposium (IV)*, 2015, pp.1094–1099. New York: IEEE.
19. Tomas-Gabarron J, Egea-Lopez E and Garcia-Haro J. Vehicular trajectory optimization for cooperative collision avoidance at high speeds. *IEEE Trans Intell Transp Syst* 2013; 14: 1930–1941.
20. Anwar S. Generalized predictive control of yaw dynamics of a hybrid brake-by-wire equipped vehicle. *Mechatronics* 2005; 15: 1089–1108.
21. Zhao C, Xiang W and Richardson P. Vehicle lateral control and yaw stability control through differential braking. In: *2006 IEEE international symposium on industrial electronics*, vol. 1, pp.384–389. New York: IEEE.
22. Kanchwala H and Wideberg J. Pitch reduction and traction enhancement of an EV by real-time brake biasing and in-wheel motor torque control. *Int J Veh Syst Model Test* 2016; 11: 165–192.
23. Polack P, Altché F, d'Andréa Novel B, et al. The kinematic bicycle model: a consistent model for planning feasible trajectories for autonomous vehicles? In: *2017 IEEE intelligent vehicles symposium (IV)*, 2017, pp.812–818. New York: IEEE.
24. Gutjahr B, Gröll L and Werling M. Lateral vehicle trajectory optimization using constrained linear time-varying MPC. *IEEE Trans Intell Transp Syst* 2016; 18: 1586–1595.
25. Kanchwala H and Ogai H. Development of an intelligent transport system for EV. *SAE Int J Passen Cars-Electron Electric Syst* 2016; 9: 9–21.
26. Funke J, Brown M, Erlien S, et al. Collision avoidance and stabilization for autonomous vehicles in emergency scenarios. *IEEE Trans Control Syst Technol* 2017; 25: 1204–1216.
27. Yuan H, Sun X and Gordon T. Unified decision-making and control for highway collision avoidance using active front steer and individual wheel torque control. *Veh Syst Dyn* 2019; 57: 1188–1205.
28. Pacejka H. *Tire and vehicle dynamics*. New York: Elsevier, 2005.
29. Abe M. Vehicle active motion controls. In: Abe M (ed.) *Vehicle handling dynamics*. 2nd ed. Oxford: Butterworth-Heinemann, 2015, pp.201–229.
30. Kanchwala H and Trigell A. Vehicle handling control of an electric vehicle using active torque distribution and rear wheel steering. *Int J Veh Des* 2017; 74: 319–345.
31. Bakker E, Nyborg L and Pacejka H. Tyre modelling for use in vehicle dynamics studies. Technical report, SAE Technical Paper, 1987.
32. Velenis E, Frazzoli E and Tsiotras P. Steady-state cornering equilibria and stabilisation for a vehicle during extreme operating conditions. *Int J Veh Autonom Syst* 2010; 8: 217–241.
33. Westfield. Sports cars. https://westfield-sportscars.co.uk/ (accessed 22 July 2020).
34. Williams H and Brailsford S. *Computational logic and integer programming*. Oxford: Oxford University Press, Inc., 1996.
35. Eckelmann S, Trautmann T, Ußler H, et al. V2v-communication, lidar system and positioning sensors for future fusion algorithms in connected vehicles. *Transp Res Procedia* 2017; 27: 69–76.
36. Maciejowski J. *Predictive control: with constraints*. London: Pearson Education, 2002.
37. Wong J. *Theory of ground vehicles*. New York: John Wiley & Sons, 2008.
38. Smith E, Tavernini D, Claret C, et al. Optimal yaw-rate target for electric vehicle torque vectoring system. In: *The dynamics of vehicles on roads and tracks: proceedings of the 24th symposium of the international association for vehicle system dynamics (IAVSD 2015)*, Graz, Austria, 17–21 August 2015. Boca Raton, FL: CRC Press.
39. IBM. *IBM ILOG CPLEX V12.1*. User manual for CPLEX, 2009.
40. Zhao S, Maxim A, Liu S, et al. Effect of control horizon in model predictive control for steam/water loop in large-scale ships. *Processes* 2018; 6: 265.
41. Seborg D, Mellichamp D, Edgar T, et al. *Process dynamics and control*. New York: John Wiley & Sons, 2010.

42. Viana I, Kanchwala H and Aouf N. Cooperative trajectory planning for autonomous driving using nonlinear model predictive control. In: *2019 IEEE international conference on connected vehicles and expo (ICCVE) (IEEE ICCVE 2019)*, Graz, Austria, 2019.

43. Kanchwala H. Path planning and tracking of an autonomous car with high fidelity vehicle dynamics model and human driver trajectories. In: *2019 IEEE 10th international conference on mechanical and aerospace engineering (ICMAE)*, 2019, pp.306–313. New York: IEEE.