

Article

LTV-MPC Approach for Automated Vehicle Path Following at the Limit of Handling

Ádám Domina * and Viktor Tihanyi

Department of Automotive Technologies, Budapest University of Technology and Economics,
1111 Budapest, Hungary; tihanyi.viktor@kjk.bme.hu

* Correspondence: domina.adam@edu.bme.hu

Abstract: In this paper, a linear time-varying model predictive controller (LTV-MPC) is proposed for automated vehicle path-following applications. In the field of path following, the application of nonlinear MPCs is becoming more common; however, the major disadvantage of this algorithm is the high computational cost. During this research, the authors propose two methods to reduce the nonlinear terms: one is a novel method to define the path-following problem by transforming the path according to the actual state of the vehicle, while the other one is the application of a successive linearization technique to generate the state-space representation of the vehicle used for state prediction by the MPC. Furthermore, the dynamic effect of the steering system is examined as well by modeling the steering dynamics with a first-order lag. Using the proposed method, the necessary segment of the predefined path is transformed, the linearized model of the vehicle is calculated, and the optimal steering control vector is calculated for a finite horizon at every timestep. The longitudinal dynamics of the vehicle are controlled separately from the lateral dynamics by a PI cruise controller. The performance of the controller is evaluated and the effect of the steering model is examined as well.



Citation: Domina, Á.; Tihanyi, V. LTV-MPC Approach for Automated Vehicle Path Following at the Limit of Handling. *Sensors* **2022**, *22*, 5807.
<https://doi.org/10.3390/s22155807>

Academic Editor: Felipe Jiménez

Received: 9 June 2022

Accepted: 27 July 2022

Published: 3 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: automated vehicle; model predictive controller; path following; successive linearization; vehicle dynamics

1. Introduction

Collision avoidance systems, advanced driver-assistance systems, and the many other types of automated vehicle functions are becoming more and more popular and have become the most related topics in the field of automotive research. The advantages of automatization of different vehicle functions include the potential to improve road safety, reduce pollutant emissions and traveling times, and eliminate human errors, which are the primary cause of accidents. An automated vehicle can avoid collisions by using the steering and the braking system conventionally; furthermore, a new alternative solution for accident prevention is to force the vehicle into an unstable state, in which a control software can drive the vehicle at a level as high as a professional driver, for example, in case of a drift maneuver [1–3]. Each of these studies aims to drive a vehicle at the level of a professional human driver. In [1], an LQ controller is used for steady-state drifting, while in [2], an MPC is applied for drifting in varying road surface conditions, and in [3], the drift is realized by a reinforcement learning algorithm.

According to the structure of the automated vehicles, the motion planning and motion control modules are essential to achieve these goals. In this paper, the authors focus on the motion control layer by proposing an LTV-MPC structure for path-following tasks. Based on the existing research and the experimental tests, the authors identify three features to be considered in a path-following controller at higher vehicle speed, the inclusion of the vehicle dynamics in the control law, and the knowledge of the path ahead of the vehicle. Furthermore, the inclusion of the steering dynamics in the control law is identified as the

third required feature, which has a high effect on the performance of the path-following controller. Numerous path-following solutions are summarized in [4–6], e.g., geometric controllers, such as Pure Pursuit and Stanley, linear quadratic regulators, model predictive controllers, neural network based controllers, etc. The authors decided to apply MPC for the path-following tasks, as the identified features can be incorporated into an MPC and the MPC handles constraints well.

Three main types of MPC methods are used in the field of path tracking, the linear parameter-varying MPC (LPV-MPC) [7–9] the linear time-varying MPC (LTV-MPC) [10–13], and the nonlinear MPC (NLMPC) [10,14–16] solutions. The LPV-MPC applies a linear vehicle model for state prediction and the structure of that model does not change over time; however, a few of the variables, e.g., the velocity of the vehicle, can change. The model is calculated at each timestep based on the current states. The main advantage of the LPV-MPC method is the low computational cost; however, this technique reaches the limit of applicability when the controlled system leaves the linear region, where the system model is defined. Thus, the LPV-MPC method is unable to handle nonlinear system dynamics in a wide range, e.g., tire dynamics in the nonlinear region of the tires. The change in given parameters can be determined by measurements or estimated by state estimators.

The LTV-MPC [10–12,17] can handle the nonlinear behavior of the system by linearizing the nonlinear system at every timestep and estimating the states accordingly. The predicted states are still based on a linear system model; however, these are extremely close to the real states of the system, since the controller calculates the predicted states for a short horizon, e.g., a few hundred ms, in which range the prediction is acceptable. In the LTV-MPC structure, the current vehicle state is the basis of the linearization and an additional transformation in the system matrices and the control input is required. The LTV-MPC method claims a bit more computational cost than the LPV-MPC; however, it still can run on an ECU or a rapid prototyping unit in real-time. When using LPV-MPC or LTV-MPC, usually, a quadratic cost function is formulated, the optimum of which is computationally cheaper to find.

Applying the NLMPC, the evolution matrices are calculated based on a nonlinear system model without linearization. A nonlinear cost function is formulated, the optimum of which is computationally expensive to find, and might limit the applicability of the NLMPC, as discussed in [6].

In this paper, the authors propose an LTV-MPC method for automated vehicle path-following scenarios. The aim of this research is to apply a nonlinear vehicle model for state prediction, handled by a successive linearization technique, where the vehicle model is linearized at every timestep by Jacobi linearization. The applied vehicle model is coupled with a Pacejka tire model, the parameters of which are identified based on measurement results. The application of a nonlinear vehicle model coupled with the Pacejka model is expected to be suitable for handling the high sideslip angles of the tires, where the tires are operating in the nonlinear region—which the LPV-MPC is unable to handle—hence, the vehicle should be able to perform the path-following task at the limit of handling.

In the existing research, if the path is defined in a forward-looking way, it is performed by defining the reference path in the global coordinate system by a series of (x,y) points [10,12,18–21]. While this solution is obvious, it incorporates high nonlinearity into the plant model, since the motion of the vehicle needs to be transformed from the vehicle's coordinate system—i.e., the ego frame—to the global frame. A novel method for the definition of the reference path is presented, by which the number of the nonlinear terms involved in the entire control problem is successfully reduced. According to our proposal, the path is transformed from the global coordinate system into the vehicle's coordinate system and the reference states, which need to be realized by the vehicle, are expressed according to the state-space representation of the vehicle plant model used for state prediction by the MPC. In the existing research, the reference path is either not defined in a forward-looking way [7,22], or the forward-looking property is limited to the lateral displacement of the

vehicle [8,12]. In this article, the authors define both the lateral displacement and the heading angle as a reference series for a finite horizon, as presented in Section 3.

Some studies considered the effect of the actuator fault [23] and the robustness of the controlled system [24]; in this article, the effect of the steering system is analyzed. The effect of the consideration of the steering dynamics in the vehicle model applied for state prediction is discussed in this paper as well. The steering system is modeled as a first-order lag. Moreover, the performance of the controller is evaluated with and without the inclusion of the steering dynamics in the plant model by applying the control input to the same vehicle model, which includes the steering dynamics.

The contribution of this paper is twofold, (1) The proposed LTV-MPC method for path following includes a novel reference definition method that effectively reduces the nonlinear terms of the path-following problem by transforming the path from the global frame to the ego frame. By this transformation, the nonlinearity that needs to be managed during the Jacobian linearization is significantly reduced as the state vector has two elements less, since the position of the vehicle in the global coordinate system is not considered in the plant model. This approach removes two equations from the model while retaining the lookahead principle of the reference definition. (2) The analysis of the effect of the steering dynamics on the path-following problem are incorporated in an LTV-MPC.

The structure of this paper is as follows. Section 2 presents the vehicle model applied in the MPC for state prediction, regarding the steering dynamics and the vehicle model used for testing the controller. In Section 3, the generation of the reference path is presented. The structure of the proposed MPC and the derivation of the quadratic programming (QP) cost function is presented in Section 4. Section 5 describes the results, while Section 6 provides an analysis, in which the controller is tested on a sine wave path and the effect of the application of linear and nonlinear vehicle models in the state prediction is compared. Finally, Section 7 discusses the concluding remarks and suggestions for further research directions.

2. Vehicle Modelling

In this section, the applied tire and vehicle models are presented. The authors use different vehicle models for state prediction and for simulation testing. Furthermore, a Pacejka tire model is applied in both models, the parameters of which are identified from measurement data.

2.1. Vehicle Model for Testing

A four-wheel vehicle model is used for testing the path-following controller in a simulation environment. The roll and pitch dynamics are neglected in the vehicle model since they have no significant effect on the path-following problem. The model describes the planar dynamics of the vehicle: the angular acceleration around the vertical axis (1), the longitudinal (2), and the lateral acceleration (3).

$$\dot{r} = \frac{1}{I_z} M_z \quad (1)$$

$$a_x = \frac{1}{m} F_x^V + V_y \dot{r} \quad (2)$$

$$a_y = \frac{1}{m} F_y^V - V_x \dot{r} \quad (3)$$

where r is the yaw rate, I_z is the moment of inertia of the vehicle around the axis z , M_z is the resultant torque, which rotates the vehicle around axis z , a_x is the longitudinal acceleration, m is the mass of the vehicle, F_x^V is the resultant of the longitudinal forces, acting on the vehicle, V_y is the lateral velocity of the vehicle in the ego coordinate system, a_y is the lateral acceleration, F_y^V is the resultant of the lateral forces acting on the vehicle, and V_x is the longitudinal velocity in the ego frame.

The resultant torque and forces are calculated by (4), (5), and (6)

$$M_z = -b(F_{yRL} + F_{yRR}) + s_f(-F_{xFL}\sin(\delta_{FL}) + F_{xFR}\sin(\delta_{FR})) + s_r(-F_{xRL} + F_{xRR}) + a(F_{yFL}\cos(\delta_{FL}) + F_{yFR}\cos(\delta_{FR})) \quad (4)$$

$$F_x^V = F_{xRL} + F_{xRR} + F_{yFL}\sin(\delta_{FL}) + F_{yFR}\sin(\delta_{FR}) + F_{xFL}\cos(\delta_{FL}) + F_{xFR}\cos(\delta_{FR}) \quad (5)$$

$$F_y^V = F_{yRL} + F_{yRR} + F_{xFL}\sin(\delta_{FL}) + F_{xFR}\sin(\delta_{FR}) + F_{yFL}\cos(\delta_{FL}) + F_{yFR}\cos(\delta_{FR}) \quad (6)$$

where a and b are the distances between the center of gravity (C.G.) and the front and the rear axle, respectively, s_f and s_r are the front and rear track of the vehicle, respectively, δ_{FL} and δ_{FR} are the front-left and front-right steering angles of the individual wheels, F_{xij} and F_{yij} are the longitudinal and lateral forces at the individual wheels, where i marks the front and the rear axles, hence, $i = R$ as rear or $i = F$ as front, and j marks the left and the right wheels, hence, $j = L$ as left or $j = R$ as right. In this model, the authors consider a rear-wheel drive vehicle, with equally distributed traction force on the rear wheels, hence, F_{xRL} and F_{xRR} are equal. Furthermore, the authors assume a front-wheel steering vehicle; hence, the steering angle is solely interpreted at the front wheels.

In Figure 1, the tire sideslip angles and the sideslip angle of the vehicle at C.G. are also shown. The vehicle sideslip angle is calculated by (7).

$$\beta = \tan^{-1}\left(\frac{V_y}{V_x}\right) \quad (7)$$

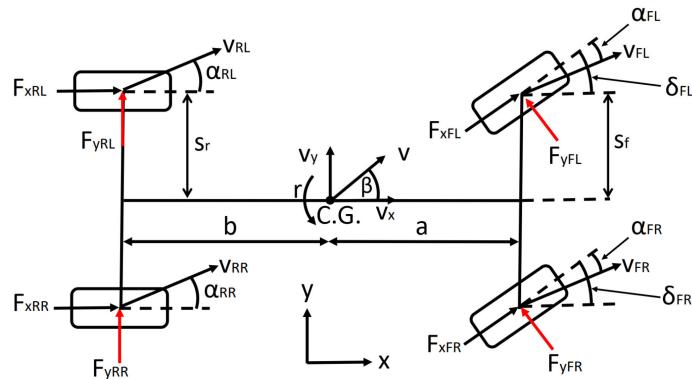


Figure 1. The four-wheel vehicle model applied for controller testing.

The sideslip angle of an individual wheel is calculated using the velocity vector of the given wheel and the steering angle at the front axle. The velocity vector of the wheels is described by (8), the position vector of the wheels is defined by (9), where the origin is the C.G. of the vehicle, and the tire sideslip angles are described by (10) and (11).

$$v_i = \begin{bmatrix} v_{xi} \\ v_{yi} \\ v_{zi} \end{bmatrix} = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ r \end{bmatrix} \times P_i \quad (8)$$

$$P_{FL} = [a \ s_f \ 0]^T, P_{FR} = [a \ -s_f \ 0]^T, P_{RL} = [-b \ s_r \ 0]^T, P_{RR} = [-b \ -s_r \ 0]^T \quad (9)$$

$$\alpha_{FL} = \tan^{-1}\left(\frac{v_{yFL}}{v_{xFL}}\right) - \delta_{FL}, \alpha_{FR} = \tan^{-1}\left(\frac{v_{yFR}}{v_{xFR}}\right) - \delta_{FR} \quad (10)$$

$$\alpha_{RL} = \tan^{-1}\left(\frac{v_{yRL}}{v_{xRL}}\right), \alpha_{RR} = \tan^{-1}\left(\frac{v_{yRR}}{v_{xRR}}\right) \quad (11)$$

where $i = FL \vee FR \vee RL \vee RR$ and α denotes the tire sideslip angle. The horizontal velocity V_z in (8) is considered zero as the model focuses solely on the planar dynamics of the vehicle, as stated previously.

A Pacejka tire model [20] is applied to calculate the lateral forces at each wheel. During this research, the vehicle moves at constant speed; thus, the longitudinal forces are small, especially compared to the lateral forces and, hence, solely a lateral tire model is used in this article. The Pacejka tire model is described by Equations (12) and (13) [25].

$$Y = D \sin(C \tan^{-1}(B\phi)) + S_v \quad (12)$$

$$\phi = (1 - E)(\alpha + S_h) + (E/B)\tan^{-1}(B(\alpha + S_h)) \quad (13)$$

where B is the stiffness factor, C is the shape factor, D is the peak factor, E is the curvature factor, S_v and S_h are the vertical and the horizontal shift, respectively, and the α sideslip angle is defined in degrees. The Pacejka tire model and the meaning of the factor variables are described in more detail in [25].

The entire vehicle model, including the tire model, is fitted to a test vehicle used for automated vehicle function tests, such as automated drift, Moose test, and other path-following tasks. Further details of the test vehicle setup can be found in [1].

To determine the characteristics of the tires, a ramp steer maneuver is taken by the test vehicle and the necessary variables are measured with the data acquisition system. The parameters in the Pacejka tire model are fitted to the measurement results, as shown in Figure 2.

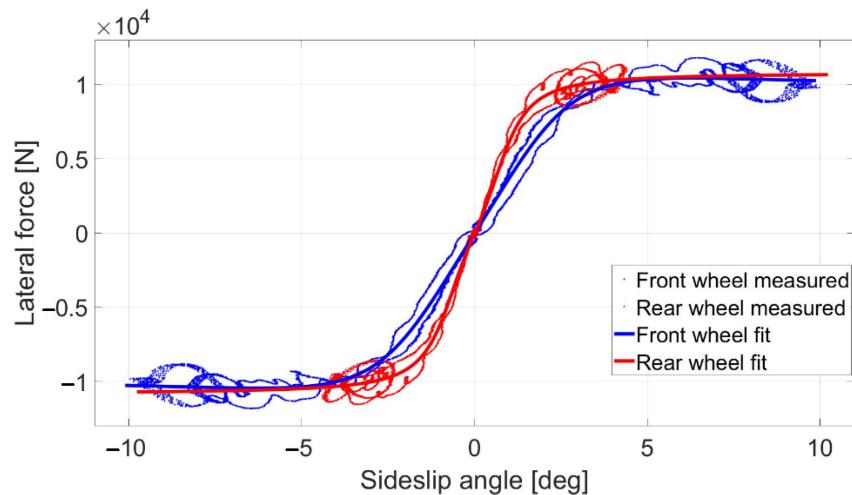


Figure 2. The measured and identified tire characteristics.

The characteristics of the rear tires have a greater slope at small sideslip angles, where the tire model is nearly linear. That means the rear tires have greater cornering stiffness than the front tires. This meets the expectations of the authors, since the front wheels of the vehicle are mounted with 245/35 R19 tires, while 265/35 R19 tires are applied at the rear wheels—using the same sidewall height, the wider tire becomes stiffer. During the identification of the tire characteristics, the vertical load of the tires is assumed to be constant.

The identified tire models describe the characteristics of the tires located on the same axle, not the individual wheels; hence, the results in Figure 2 contain the lateral forces generated by the two front tire pairs and the two rear tire pairs, respectively. Thus, when modeling one wheel, the value of the lateral force needs to be halved.

2.2. Vehicle Model for State Prediction

The states of the vehicle model applied in the MPC for state prediction are advisable to choose in a way to correspond to the control purpose. In this case, the aim is to drive the vehicle along a predefined path as fast as the vehicle is still able to execute the maneuver. The applied vehicle model shown in Figure 3 is a dynamic bicycle model, coupled with different Pacejka tire models at the front and the rear wheels. Since the bicycle model applies one wheel per axle, the identified Pacejka models can be used without halving the values of the lateral forces of the tire characteristics. The applied vehicle model describes the dynamics of the vehicle with four equations, enhanced by a fifth equation for the steering dynamics. The steering dynamics are considered to take into account the dynamic lag of the steering system, in which way the controller considers that the demanded steering angle will only be realized with a time delay. The dynamics of the steering system are modeled by a first-order lag, which has one tuning parameter, the time constant, to identify the measurements made. The time constant is determined by measurements conducted on the test vehicle.

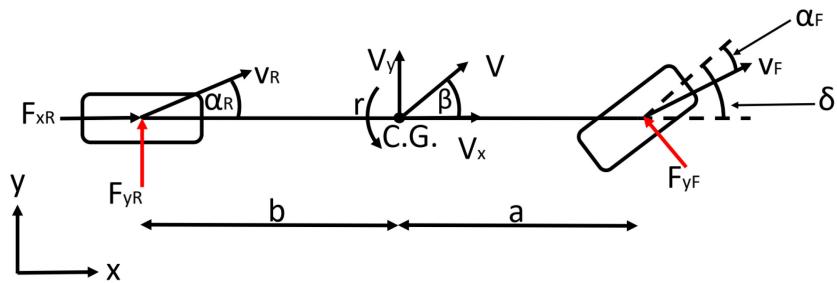


Figure 3. The vehicle model applied to the MPC for state prediction.

The state vector of the system is chosen as $x = [v \ V_y \ \varphi \ r \ \delta_{act}]^T$, where y is the lateral displacement of the vehicle in the ego frame, φ is the heading angle of the vehicle, and δ_{act} is the actual steering angle. The vehicle dynamics are described by (14)–(17), while the steering dynamics by (18).

$$V_y = \dot{y} \quad (14)$$

$$a_y = \dot{V}_y = \left(\frac{F_{yf}\cos(\delta) + F_{yr}}{m} \right) - V_x r \quad (15)$$

$$r = \dot{\varphi} \quad (16)$$

$$\dot{r} = \frac{aF_{yf}\cos(\delta) - bF_{yr}}{I_z} \quad (17)$$

$$\dot{\delta}_{act} = -\frac{1}{T_{st}}\delta_{act} + \frac{1}{T_{st}}\delta_{dem} \quad (18)$$

where δ_{dem} is the steering angle required by the MPC and T_{st} is the time constant of the first-order steering model. To calculate the F_{yf} and F_{yr} lateral forces, the identified Pacejka models are applied by determining the parameters of (12) and (13). The sideslip angles are calculated by (19) and (20).

$$v_{yF} = \dot{v}_y + ar, \ v_{yR} = \dot{v}_y - br \quad (19)$$

$$\alpha_F = \tan^{-1} \left(\frac{v_{yF}}{v_{xF}} \right) - \delta_F, \ \alpha_R = \tan^{-1} \left(\frac{v_{yR}}{v_{xR}} \right) \quad (20)$$

where v_{yF} and v_{yR} are the velocity of the front and the rear wheels and α_F and α_R are the sideslip angles of the front and the rear wheels, respectively. The tests are conducted without the consideration of the steering dynamics, in which case, the state vector is chosen as $x = [v \ V_y \ \varphi \ r]^T$, and (18) is neglected during the state prediction.

The future states of a vehicle using a vehicle model can be predicted for a finite time horizon. In this article, the authors apply successive linearization to continuously generate the state-space representation of the vehicle. The successive linearization allows linearizing the nonlinear vehicle model at the current operating point, calculating the state-space representation accordingly, and making the state prediction based on the linearized system. The linearization is conducted at every timestep; hence, the MPC can use the latest state of the vehicle as a basis of the state prediction.

The linearization of the vehicle model is solved by Jacobian linearization and leads to the continuous-time $[A_c, B_c, C_c, D_c]$ state-space representation of the system (21).

$$A_c(i, j) = \frac{\partial F_i}{\partial x_j}, B_c(i, j) = \frac{\partial F_i}{\partial u_j}, C_c(i, j) = \frac{\partial z_i}{\partial x_j}, D_c(i, j) = \frac{\partial z_i}{\partial u_j} \quad (21)$$

where F is the system of nonlinear equations (14)–(18), x is the state vector, u is the control input, which, in this case, is the demanded steering angle δ_{dem} , while C_c and D_c are considered constant matrices (22).

$$C_c = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, D_c = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (22)$$

Furthermore, matrix B_c remains constant during the linearization in the following form (23).

$$B_c = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{T_{st}} \end{bmatrix}^T \quad (23)$$

The partial derivatives of the matrices are evaluated at the desired operation point, which is specified by the state vector x_o , the time derivative of the state vector \dot{x}_o , and the control vector u_o and, thereby, the partial derivatives in (21) need to be calculated based on the initial conditions $x = x_o$, $\dot{x} = \dot{x}_o$ and $u = u_o$. The evolution of the continuous-time system can be described by (24)

$$\dot{x}_L = \dot{x}_o + A_c(x_L - x_o) + B_c(u_L - u_o) \quad (24)$$

where x_L and u_L are the linearized state of the system and the control input, respectively. The constant terms of (24) are incorporated into K_c , with which the continuous-time representation of the state-space system becomes (25) and (26).

$$\dot{x}_L = A_c x_L + B_c u_L + K_c, \quad K_c = \dot{x}_o - A_c x_o - B_c u_o \quad (25)$$

$$y_L = C_c x_L + D_c u_L \quad (26)$$

where y_L is the output of the system.

Since the MPC is a discrete-time-control technique, the state-space representation needs to be discretized to get the discrete-time state-space representation of the system $[A_d, B_d, C_d, D_d]$ and K_d (27)–(30).

$$A_d = e^{A_c T_s} \quad (27)$$

$$B_d = A_c^{-1} (e^{A_c T_s} - I) B_c \quad (28)$$

$$K_d = A_c^{-1} (e^{A_c T_s} - I) K_c \quad (29)$$

$$C_d = C_c, \quad D_d = D_c \quad (30)$$

where T_s is the discretization timestep, which is equal to the timestep value of the MPC. Finally, the discrete-time representation of the system (31) can be used for state prediction by the MPC.

$$\begin{aligned} x(k+1) &= A_d x(k) + B_d u(k) + K_d \\ y(k+1) &= C_d x(k+1) + D_d u(k+1) \end{aligned} \quad (31)$$

3. Reference Path Definition

As previously introduced, in numerous studies, the control problem is to minimize the error between the spatial reference path and the position of the vehicle. In this research, the state-space representation of the vehicle contains the position of the vehicle in the global frame, according to the reference definition, where the path is defined by a series of (x, y) points in the global frame. The controlled state is the spatial position of the vehicle—X and Y coordinates in the global frame. The spatial path-following problem is possibly supplemented with the tracking of the heading and the yaw rate. The problem with the formulation when the X and Y coordinates are defined as a reference is that it introduces several nonlinear terms into the vehicle model applied for state prediction by the MPC and increases the dimension of the state vector by two. Therefore, if the control problem is defined to follow spatial points, the vehicle needs to be transformed to the global frame in the state-space representation—this transformation is responsible for higher nonlinearity. Thus, if the increase in the dimension of state vector x by two—displacement in X and Y directions—can be omitted, then the computational requirements of the MPC can be reduced, which increases directly proportionally with the increase in the dimension of state vector x , as presented in [26]. In this article, a novel method is proposed for the reduction in nonlinear terms, which is to transform the path to the vehicle and describe the path-following task in a way that retains the advantageous property of the MPC, which is the knowledge of the path ahead of the vehicle—the knowledge of the reference for a finite horizon. In this article, the lateral displacement and the heading of the vehicle are defined as a reference to be followed, in accordance with the states of the vehicle model used for state prediction. The reference state variables are chosen as they clearly define the relationship between the vehicle and the path, with respect to the lateral terms. To calculate the reference for a finite horizon, firstly, the lateral and angular errors are defined, as shown in Figure 4. The lateral error e is the distance between the C.G. point of the vehicle and the intersection point M on the y axis of the ego frame and the reference path.

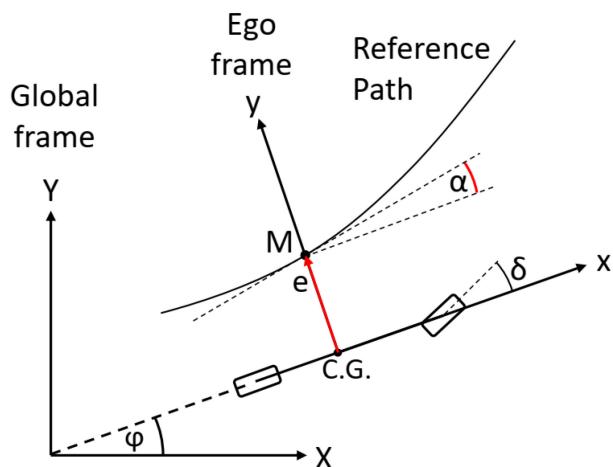


Figure 4. The interpretation of the lateral and angular errors.

The angular error α is the angle between the heading of the vehicle and the tangent of the path at the intersection point M . To generate the reference for the lateral displacement state y and for the heading angle state φ , the vehicle, the reference path, the ego frame, and the global frame need to be considered, as shown in Figure 5. In Figure 5, the lateral errors and the angle errors are shown for a finite N_p horizon, where N_p is the prediction horizon of the MPC. If the lateral error values from the ego coordinate system can be seen, these errors can be interpreted as reference lateral displacements, which corresponds to y , the first state of the state-space representation. In a similar way, the angular errors can be interpreted as φ reference heading angles. By applying this method for reference generation, the

transformation of the vehicle coordinates from the ego frame to the global frame becomes avoidable, resulting in omitting several nonlinear terms from the vehicle model.

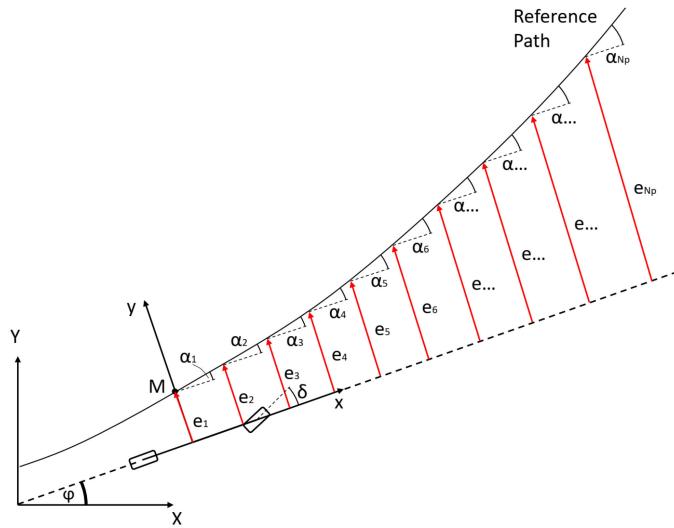


Figure 5. The interpretation of the lateral error reference and orientation angle reference in both ego frame and global frame.

As stated before, if the vehicle model is not transformed into the global frame, the path needs to be transformed into the ego frame of the vehicle. The transformation is defined by Equations (32)–(34),

$$\gamma = \alpha_1 + \varphi \quad (32)$$

$$H = \begin{bmatrix} 0 \\ e_1 \end{bmatrix} \quad (33)$$

$$P_{ref,i} = H + \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) \\ \sin(\gamma) & \cos(\gamma) \end{bmatrix} P_i \quad (34)$$

where γ is the transformation angle, e_1 and α_1 are the lateral and angular errors at the M point, respectively, H is an offset vector, which shifts the entire reference trajectory to the necessary lateral displacement, P_i is the (x,y) coordinate of the i th path point described in the global frame, and $P_{ref,i}$ is the coordinate of the reference path point in the ego frame. After the transformation, the lateral error references and the angular error references can be calculated; the lateral error reference is the y -direction component of $P_{ref,i}$, which determines the series of e_1, e_2, \dots, e_{Np} , while the angular reference can be calculated considering the tangent of the transformed path at every point where the lateral displacements are defined.

After the reference values are calculated, the reference vector can be generated by (35) and (36).

$$t = \begin{bmatrix} e_i \\ \alpha_i \end{bmatrix} \quad (35)$$

$$T = [t_1 \ t_2 \ \dots \ t_{Np}]^T = [e_1 \ \alpha_1 \ e_2 \ \alpha_2 \ \dots \ e_{Np} \ \alpha_{Np}]^T \quad (36)$$

While the original reference path is described by N_p points, the reference lateral displacements and the heading angles need to be calculated for every path point. The result is a stacked matrix T with a dimension of $2 \times N_p$. Using this method of reference definition, several nonlinear terms are omitted from the vehicle model, the reference is coherent to the states of the vehicle model, the reference lateral displacement is coupled with the first state in the state-space model, and the angle reference is coupled with the third state. Furthermore, the presented reference generation method preserves the forward-looking nature of the problem definition, which is necessary for the path-following task.

4. MPC Structure

As stated in Section 2, the authors apply a successive linearization technique, similar to [12,14,17], to handle the nonlinear vehicle model. During the successive linearization, the nonlinear vehicle model is linearized at every timestep based on the actual state vector of the vehicle. The resulting state–space representation is applied to predict the future vehicle states for a finite prediction horizon, the length of which is N_p . Using an MPC controller, the objective is to minimize the difference between the given reference states in the system and the predicted states in the systems by calculating the optimal control input vector as a result of an optimization process, while meeting a set of constraints. The optimization process requires a cost function to minimize. In this section, the derivation of the cost function is presented. The cost function depends on the tracking error and the amount of the control input. The tracking error $e(k)$ in the k -th timestep is defined as $e(k) = y(k) - r(k)$, where $y(k)$ is the current state of the system and $r(k)$ is the given reference. Then, the evolution of the error can be defined for the k -th, $k + 1$ -th, and $k + 2$ -th timesteps, etc., by (37).

$$\begin{aligned} e(k) &= C_d x(k) + D_d u(k) - r(k) \\ e(k+1) &= C_d x(k+1) + D_d u(k+1) - r(k+1) \\ &= C_d A_d x(k) + C_d B_d u(k) + C_d K_d + D_d u(k+1) - r(k+1) \\ e(k+2) &= C_d x(k+2) + D_d u(k+2) - r(k+2) = \\ &C_d A_d^2 x(k) + C_d A_d B_d u(k) + C_d B_d u(k+1) + C_d A_d K_d + C_d K_d + \\ &D_d u(k+2) - r(k+2) \end{aligned} \quad (37)$$

The evolution of the error needs to be calculated for the entire prediction horizon, resulting in the error vector $\bar{e} \in R^{N_p N_o}$, where N_o is the number of outputs in the system, which is equal to the rows of matrix C_d . In this case, $N_o = 2$ corresponds to the lateral displacement and the heading angle references.

$$\bar{e} = \bar{P} x(k) + \bar{H} \bar{u} + \bar{E} K_d - \bar{r} \quad (38)$$

$$\bar{e} = \begin{bmatrix} e(k) \\ e(k+1) \\ e(k+2) \\ \vdots \\ e(k+N_p-1) \end{bmatrix}, \quad \bar{P} = \begin{bmatrix} C_d \\ C_d A_d \\ C_d A_d^2 \\ \vdots \\ C_d A_d^{N_p-1} \end{bmatrix} \quad (39a)$$

$$\bar{H} = \begin{bmatrix} D_d & 0 & 0 & \dots \\ C_d B_d & D_d & 0 & \dots \\ C_d A_d B_d & C_d B_d & D_d & \dots \\ \vdots & \vdots & \vdots & \ddots \\ C_d A_d^{N_p-2} B_d & C_d A_d^{N_p-3} B_d & C_d A_d^{N_p-4} B_d & \dots \end{bmatrix} \quad (39b)$$

$$\bar{u} = \begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \\ \vdots \\ u(k+N_p-1) \end{bmatrix}, \quad \bar{E} = \begin{bmatrix} 0 \\ C_d \\ C_d(I + A_d) \\ \vdots \\ C_d(I + \sum_{i=1}^{N_p-2} A_d^i) \end{bmatrix} \quad (39c)$$

$$\bar{r} = \begin{bmatrix} r(k) \\ r(k+1) \\ r(k+2) \\ \vdots \\ r(N_p - 1) \end{bmatrix} \quad (39d)$$

where $\bar{u} \in R^{N_p N_u}$ is the control input vector, i.e., the result of the optimization process, N_u is the number of the control inputs, which is equal to the columns of matrix B_d —in this case— $N_u = 1$ corresponding to the steering input, $\bar{P} \in R^{N_p N_o \times N_x}$, N_x is the dimension of the state vector, $\bar{H} \in R^{N_p N_o \times N_p N_u}$, $\bar{E} \in R^{N_p N_o \times N_x}$, $\bar{r} \in R^{N_p N_o}$ is the reference matrix, which defines the reference for the N_p horizon. In (39), P , H , and E are the error evolution matrices. The constant terms in (38) can be combined as $\bar{K} = \bar{E}K_d - \bar{r}$, which leads to a more compact form of \bar{e} .

$$\bar{e} = \bar{P}x(k) + \bar{H}\bar{u} + \bar{K} \quad (40)$$

As the evolution of the tracking error is calculated, the cost function can be defined as

$$J(k) = \frac{1}{2} \left(\bar{e}(k)^T \bar{Q} \bar{e}(k) + \bar{u}(k)^T \bar{R} \bar{u}(k) \right) \quad (41)$$

where $\bar{Q} \in R^{N_p N_o \times N_p N_o}$ penalizes the deviation from the reference states and $\bar{R} \in R^{N_p N_u \times N_p N_u}$ penalizes the number of the control input. Both \bar{Q} and \bar{R} are diagonal square matrices, built by using matrix \bar{q} and scalar r , according to (42) and (43), where q_1 and q_2 are the weights of the deviation from the reference lateral distance and the heading angle, respectively, and r is the weight of the control input.

$$\bar{Q} = \begin{bmatrix} \bar{q} & 0 & \dots & 0 \\ 0 & \bar{q} & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & \bar{q} \end{bmatrix}, \quad \bar{R} = \begin{bmatrix} r & 0 & \dots & 0 \\ 0 & r & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & r \end{bmatrix} \quad (42)$$

$$\bar{q} = \begin{bmatrix} q_1 & 0 \\ 0 & q_2 \end{bmatrix}, \quad r = [r] \quad (43)$$

In this article, the authors apply a quadratic cost function (41), the optimum of which is found by the built-in numerical solver in the MATLAB software package, called *quadprog*. Substituting (40) into (41) results in

$$J(k) = \frac{1}{2} \left((\bar{P}x(k) + \bar{H}\bar{u} + \bar{K})^T \bar{Q} (\bar{P}x(k) + \bar{H}\bar{u} + \bar{K}) + \bar{u}(k)^T \bar{R} \bar{u}(k) \right) \quad (44)$$

The following task is to aggregate the terms that do not depend on the control input \bar{u} and aggregate the quadratic and linear terms of (44), which leads to (45).

$$J(k) = \frac{1}{2} \bar{u}(k)^T \underbrace{\left(\bar{H}^T \bar{Q} \bar{H} + \bar{R} \right)}_{\bar{G}} \bar{u}(k) + \underbrace{\left(x(k)^T \bar{P}^T + \bar{K}^T \right)}_{\bar{W}^T} \bar{Q} \bar{H} \bar{u}(k) \quad (45)$$

A more compact form of the cost function is in (46).

$$J(k) = \frac{1}{2} \bar{u}(k)^T \bar{G} \bar{u}(k) + \bar{W}^T \bar{u}(k) \quad (46)$$

When solving optimal control problems, the aim is to minimize the tracking errors and the amount of control input. However, a steady-state error might occur while tracking, since the cost of the steady-state error might be less than the cost of the higher value of the control inputs. To avoid the steady-state error, the cost function is defined by the increments in the current control input, rather than defining the individual control inputs. Applying this solution, the effect of the control input values on the cost function is minimal. The control input vector can be transformed as

$$\begin{aligned} u(k) &= u(k-1) + \rho u_1 \\ u(k+1) &= u(k) + \rho u_2 = u(k-1) + \rho u_1 + \rho u_2 \\ u(k+N_p-1) &= u(k+N_p-2) + \rho u_{N_p} = u(k-1) + \rho u_1 + \rho u_2 + \dots + \rho u_{N_p} \end{aligned} \quad (47)$$

which can be written as

$$\underbrace{\begin{bmatrix} \bar{u} \\ u(k) \\ u(k+1) \\ \vdots \\ u(k+N_p-1) \end{bmatrix}}_{\bar{u}(k-1)} = \underbrace{\begin{bmatrix} \bar{u}(k-1) \\ u(k-1) \\ u(k-1) \\ \vdots \\ u(k-1) \end{bmatrix}}_{\bar{u}(k-1)} + \underbrace{\begin{bmatrix} \bar{D} \\ I & 0 & \dots \\ I & I & \dots \\ \vdots & \vdots & \ddots \\ I & I & \dots \end{bmatrix}}_{\bar{D}} \underbrace{\begin{bmatrix} \rho \bar{u}(k) \\ \rho u_1 \\ \rho u_2 \\ \vdots \\ \rho u_{N_p} \end{bmatrix}}_{\rho \bar{u}(k)} \quad (48)$$

where $\bar{D} \in R^{N_p N_u \times N_p N_u}$ is a lower diagonal matrix built of $I \in R^{N_u \times N_u}$ matrices and zero matrices of the same dimensions. Performing the transformation, the evolution of the tracking error can be written as

$$\bar{e} = \bar{P}x(k) + \bar{H}\left(\bar{u}(k-1) + \bar{D}\rho\bar{u}(k)\right) + \bar{K} \quad (49)$$

where $\bar{u}(k-1)$ is constant for the entire prediction horizon, thus, it can be included in the constant term \bar{K} . The new const function, the solution of which is the optimal series of control input increments, is given in the following form

$$J_D(k) = \frac{1}{2}\rho\bar{u}(k)^T \bar{G}_D \rho\bar{u}(k) + \bar{W}_D^T \rho\bar{u}(k) \quad (50)$$

where the matrices are defined as $\bar{G}_D = D^T \bar{G} D$ and $\bar{W}_D^T = \bar{u}^T(k-1) \bar{G} D$.

5. Results

A reference path is defined for testing the controller shown in Figure 6. The path contains a double-lane-change section, which is considered an evasive maneuver [27], and a U-turn section, which is built by a clothoid segment, marked with A and B in Figure 6. As the vehicle drives along the U-turn section, the curvature of the path is the linear function of the length of the arc. During the simulations, the vehicle is started from the left side of the path, using a 0.2 m offset, which is applied to examine how the vehicle can find the path. The simulations are conducted at 50, 60, and 70 km/h, using vehicle models, excluding and including the steering dynamics; hence, in total, six cases are examined. During the simulations, the identified model of the steering system is applied to the vehicle model in every case; hence, the consideration of the steering dynamics in the prediction model is expected to lead to better results. The value of N_p and N_c is 10 and the sampling time is 0.05 s.

The results are summarized in Table 1, where e is the lateral error and ϕ is the orientation error. As the velocity of the vehicle is increased, the average and maximal errors are also increased; furthermore, the application of the steering dynamics can reduce the errors significantly.

Please note that the figures below sometimes do not show the difference between the reference and the real steering angle, which is due to the accurate reference tracking. As shown in Figures 7 and 8, at 50 km/h, the consideration of the steering dynamics can increase the accuracy of the controller and result in a smoother steering command, which increases the stability of the vehicle and provides better ride comfort to the passengers. Furthermore, smaller steering interventions result in smaller sideslip angles.

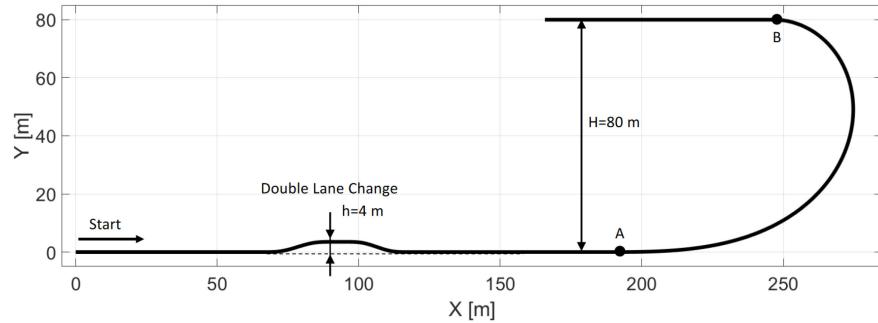


Figure 6. The reference path.

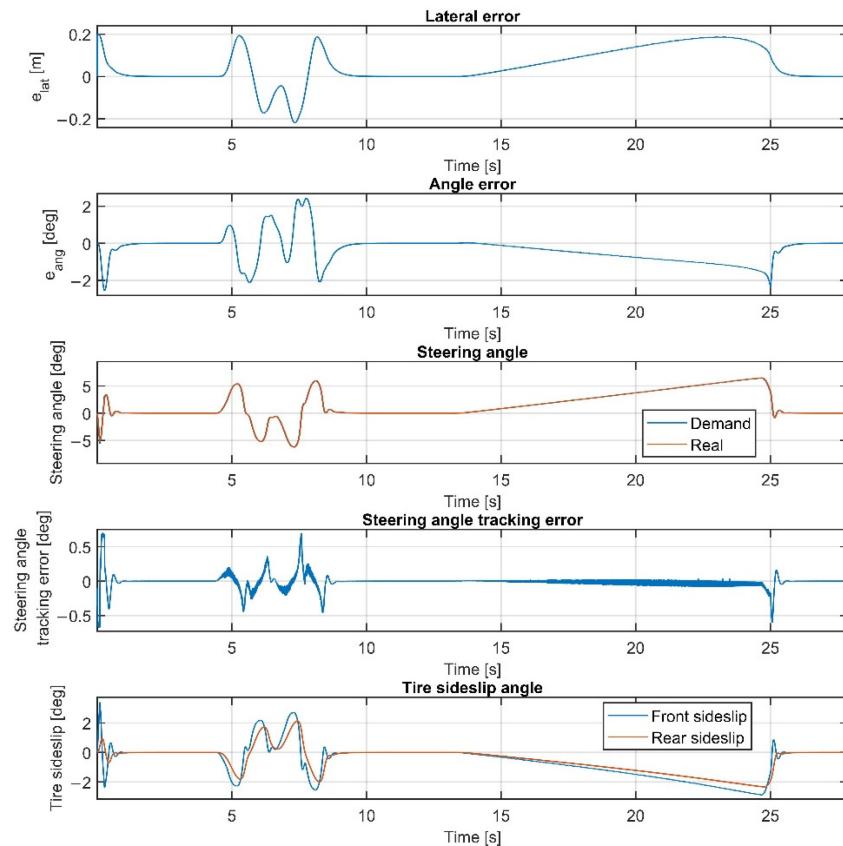


Figure 7. Simulation results at 50 km/h, without steering dynamics.

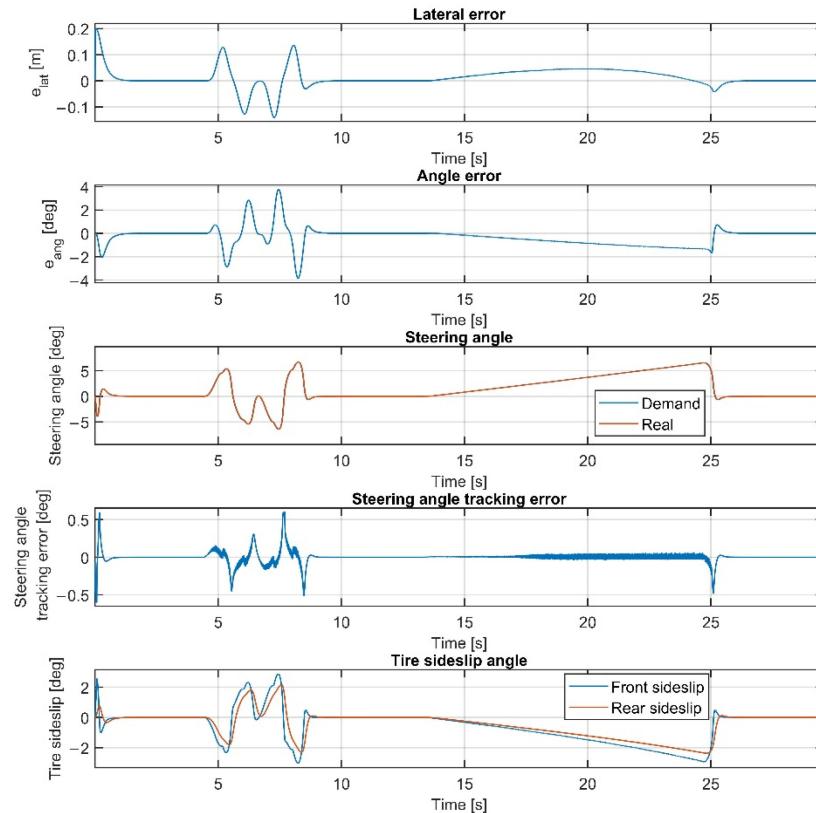
Table 1. Simulation results.

Speed [km/h]	Steering Dynamics	e_{avg} [m]	e_{max} [m]	ϕ_{avg} [deg]	ϕ_{max} [deg]
50	No	0.066	0.220	0.516	2.561
	Yes	0.023	0.200	0.510	3.862
60	No	0.085	0.425	0.407	6.159
	Yes	0.077	0.362	0.378	5.363
70	No	0.544	4.672	1.942	25.336
	Yes	0.522	4.669	1.937	25.869

The result at 60 km/h is shown in Figures 9 and 10. The results show similar behavior to the 50 km/h cases; however, the errors are a little higher, although still in an acceptable range. As shown in Figure 2, the tires become saturated when reaching about 4 deg and 2 deg of sideslip at the front and the rear wheels, respectively.

The tire sideslips reach about 7 deg at the front wheels and 5 deg at the rear wheels, which means the tires operate in a saturated state when the vehicle drives at the ending section of the U turn, resulting in a higher lateral error there.

The results at 70 km/h are shown in Figures 11–14. Figures 12 and 14 show the same results as Figures 11 and 13, respectively, but in a smaller time window to make the results more visible. The vehicle reaches and exceeds the friction limit at the U turn, which leads to high lateral errors—the vehicle is physically not able to follow the path due to its great curvature.

**Figure 8.** Simulation results at 50 km/h, including steering dynamics.

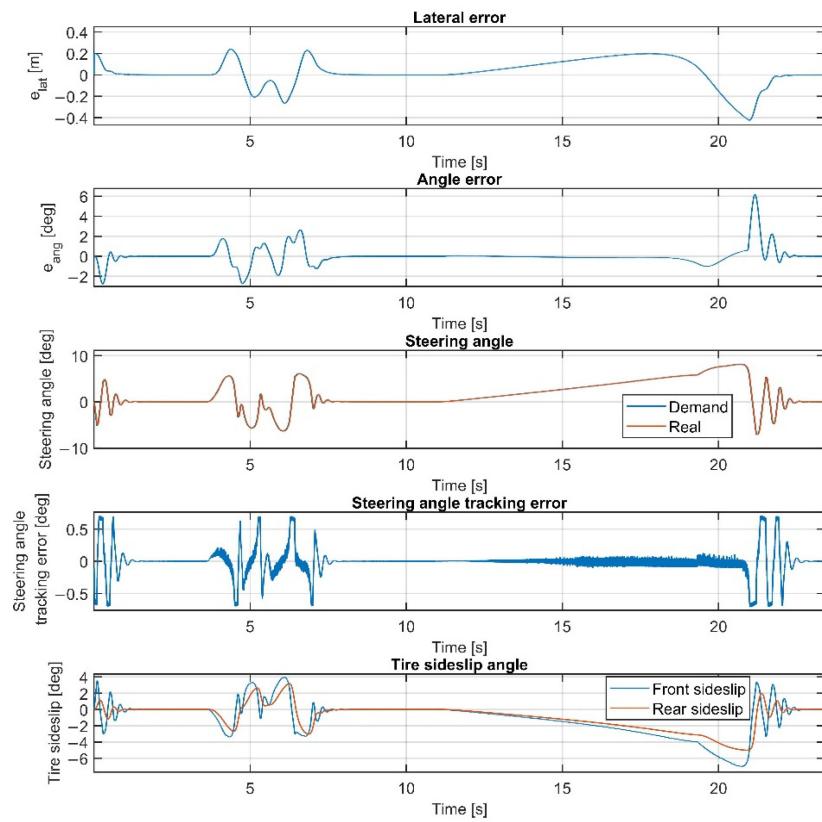


Figure 9. Simulation results at 60 km/h, without steering dynamics.

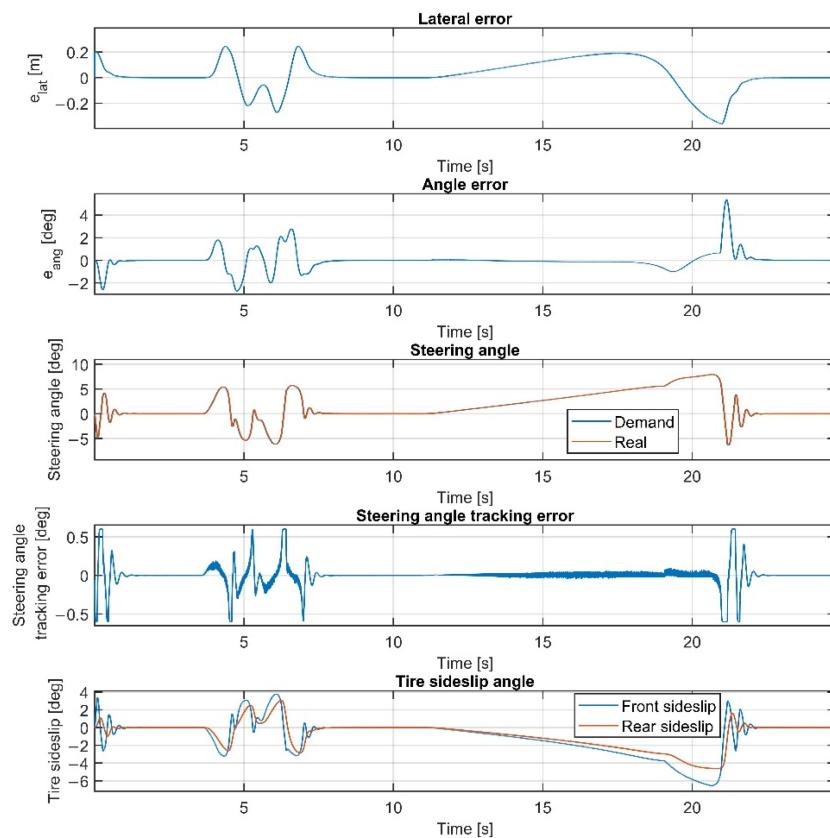


Figure 10. Simulation results at 60 km/h, including steering dynamics.

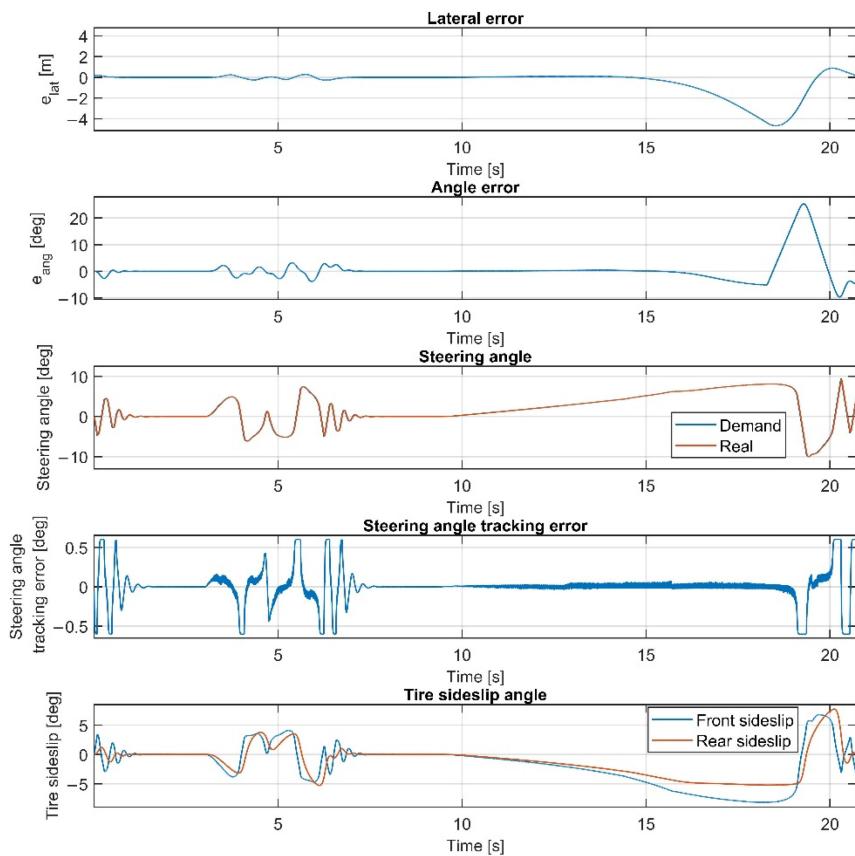


Figure 11. Simulation results at 70 km/h, without steering dynamics.

The importance of the 70 km/h case is to show whether the controller can handle the vehicle at the friction limit or increase the steering angle to a high value, applying unnecessarily large sideslip angles, which can not decrease the path-following errors. According to the simulation results, the controller can handle the nonlinear characteristics of the tires and operates stably, even when the sideslip angles reach high values. As shown in Figures 11 and 13, the sideslip angles reach a maximal 8 deg and 5 deg at the front and the rear wheels, respectively. The further increase in the sideslip would not generate a larger lateral force. As the cost function penalizes the value of the steering intervention, the steering angle does not increase. In this research, the authors do not apply constraints on the sideslip angle; however, the controller does not generate too-large values due to the inclusion of the nonlinear Pacejka tire model in the plant model.

In the double-lane-change section, the errors do not show a significant increase compared to the lower-speed scenarios; the controller could drive the vehicle faster in this section, but the higher speed would result in a greater error at the U-turn. Please consider that in Table 1, the high values of the average errors at 70 km/h are caused by the large errors at the U-turn, which has a serious effect when calculating the average errors.

Overall, the inclusion of the steering dynamics in the plant model leads to a more accurate path following, with a smoother steering intervention, while the complexity of the model is not seriously increased. Furthermore, as shown by the fourth subgraphs in Figures 7–14, the inclusion of the steering dynamics in the plant model results in a more accurate steering demand tracking, where the tracking error is decreased in every case.

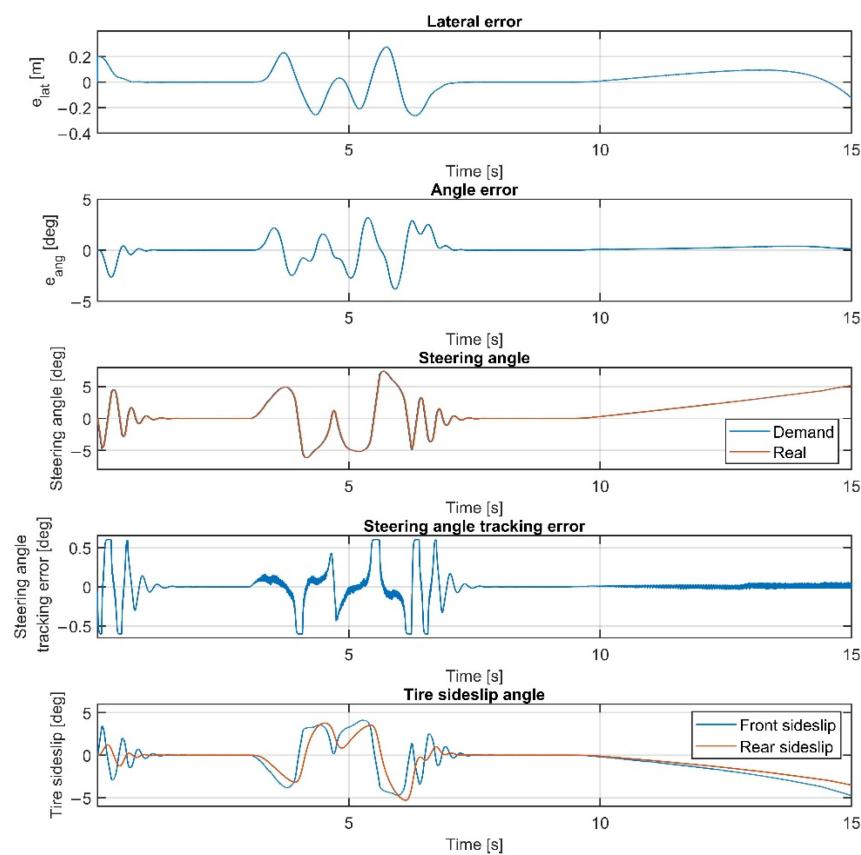


Figure 12. Simulation results at 70 km/h, without steering dynamics in 0–15 sec interval.

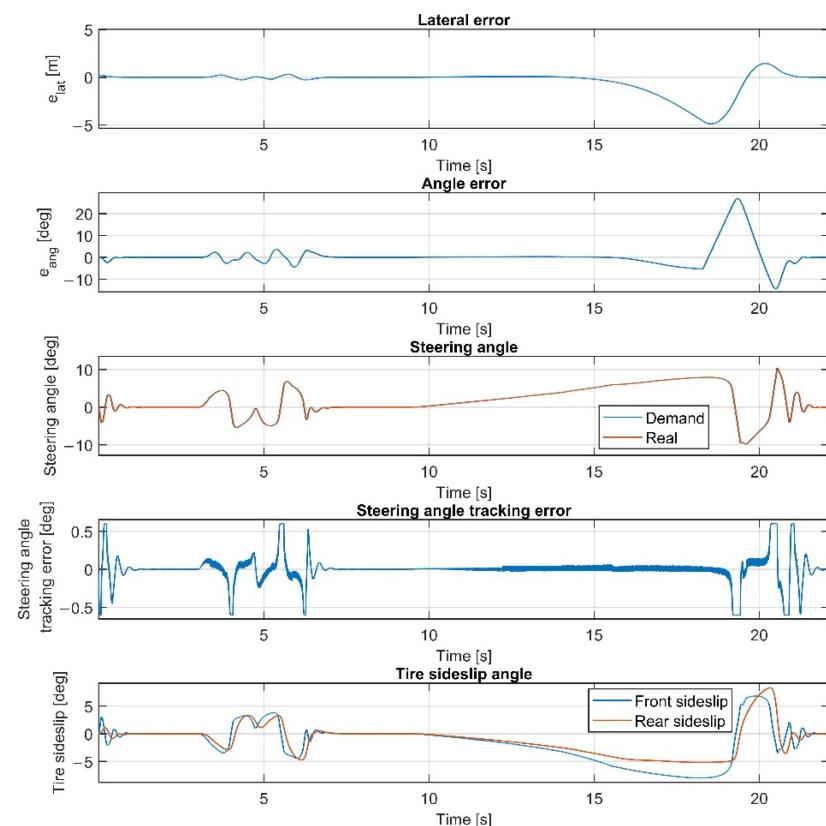


Figure 13. Simulation results at 70 km/h, including steering dynamics.

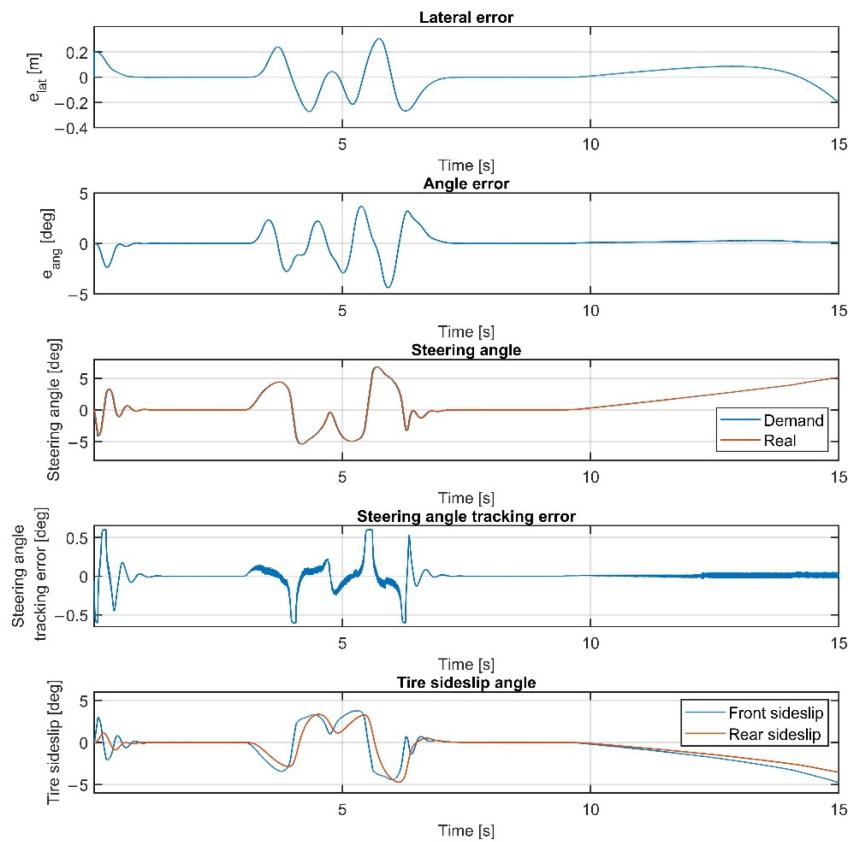


Figure 14. Simulation results at 70 km/h, including steering dynamics in 0–15 sec interval.

6. Evaluation of the Effectiveness of the Controller and Comparison of the Linear/Nonlinear Plant Model

In this section, the effectiveness of the controller is evaluated by testing on a sine wave path. Furthermore, the effect of the vehicle plant model is also evaluated by comparing the results of a linear vehicle model and a nonlinear vehicle model. The reference path is shown in Figure 15. The distance between the cones is 30 m, which means 60 m wavelength, and the lateral peak value of the sine wave is 2.5 m in both directions. Each test scenario presented in this section is conducted using the four-wheel vehicle model presented in Section 2.1 and the plant model contains the steering dynamics in each test case.

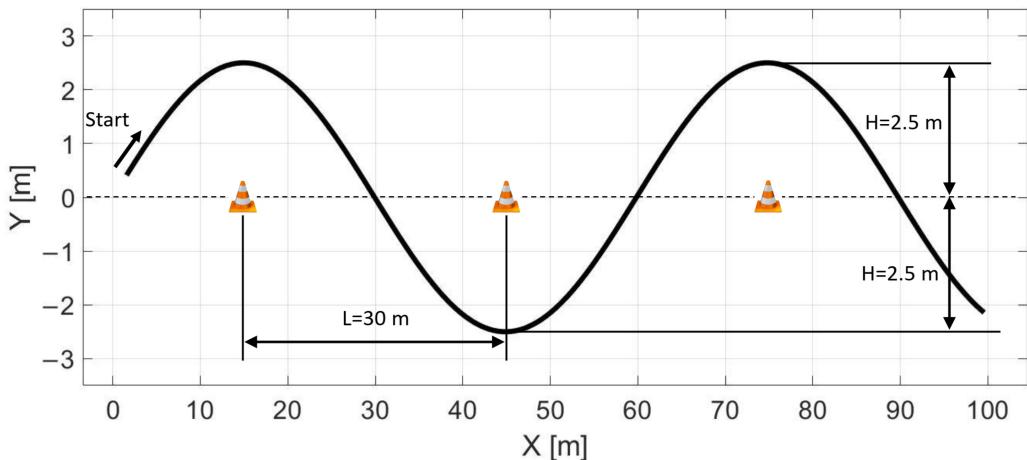


Figure 15. Sine wave reference path.

The results of the scenarios conducted on the sine wave path are summarized in Table 2. Please note that when the linear plant model is applied in the 70 km/h scenario, the results are calculated solely for the time interval of 0–4.2 s, since at 4.2 s, the vehicle left the path. In the first scenario, the controller presented in Section 4 is tested, using the vehicle model presented in Section 2.2 for state prediction. The results at 70 km/h vehicle velocity are shown in Figure 16. The controller drives the vehicle accurately and stably, the maximum sideslip angles are 4 deg, and the small values in the lateral and the angular errors also demonstrate the accuracy of the controller.

Table 2. Simulation results on the sine wave path.

	Speed [km/h]	e_{avg} [m]	e_{max} [m]	ϕ_{avg} [deg]	ϕ_{max} [deg]
Nonlinear plant model	70	0.098	0.192	0.689	2.414
Linear plant model	60	0.390	0.687	2.442	4.259
	70	1.225	2.903	15.307	71.385

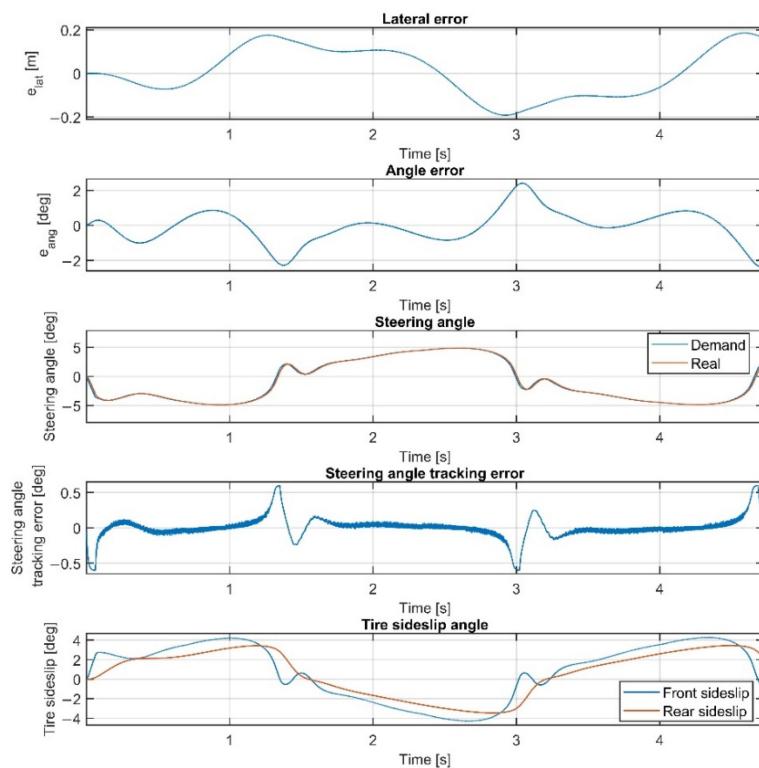


Figure 16. Simulation results on the sine wave path at 70 km/h, including steering dynamics, using the nonlinear vehicle model for state prediction.

In the next scenario, the nonlinear vehicle plant model is replaced by a linear one, in which small angle assumptions are applied during the linearization of the model, and the vehicle model is coupled with a linear tire model, while the model of the steering dynamics remains unchanged. The resulting linear vehicle model is described by Equations (51)–(55),

$$V_y = \dot{y} \quad (51)$$

$$a_y = \dot{V}_y = \frac{-c_f v_y - c_f l_f r}{m V_x} + \frac{c_f \delta}{m} + \frac{-c_r v_y + c_r l_r r}{m V_x} - V_x r \quad (52)$$

$$r = \dot{\varphi} \quad (53)$$

$$\dot{r} = \frac{-l_f c_f v_y - l_f^2 c_f r}{I_z V_x} + \frac{l_f c_f \delta}{I_z} + \frac{l_r c_r v_y - l_r^2 c_r r}{I_z V_x} \quad (54)$$

$$\dot{\delta}_{act} = -\frac{1}{T_{st}} \delta_{act} + \frac{1}{T_{st}} \delta_{dem} \quad (55)$$

where the lateral force is a linear function of the sideslip angle (56).

$$F_{yf} = -c_{af} \alpha_f, \quad F_{yr} = -c_{ar} \alpha_r \quad (56)$$

In (56), c_{af} and c_{ar} are the cornering stiffness of the front and rear tires, respectively, which are identified from the measurement results (Figure 2), and the state vector remains unchanged, $x = [v \ V_y \ \varphi \ r \ \delta_{act}]^T$.

The first test using the linear plant model is conducted at 60 km/h and the results are shown in Figure 17. Both the lateral and angular errors are large, while the controller is unable to drive the vehicle along the path when the velocity is increased to 70 km/h, as shown in Figure 18.

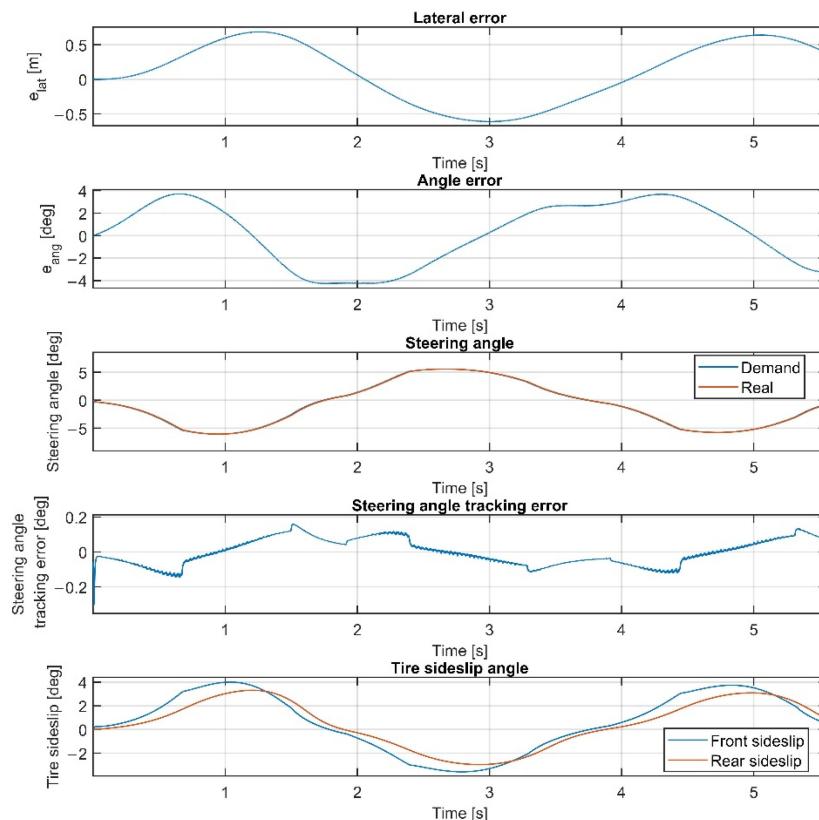


Figure 17. Simulation results on the sine wave path at 60 km/h, including steering dynamics, using the linear vehicle model for state prediction.

In the 70 km/h scenario, the vehicle is unable to follow the path anymore, resulting in leaving the path. The reason for the poor performance of the linear-model-based control is the neglection of the nonlinearities of the controlled vehicle in the state prediction. Both the nonlinear terms in the vehicle model and the tire model are neglected, which results in a less accurate state prediction; hence, in the 60 km/h scenario, performance degradation of the controller is realized, and at 70 km/h, the controller is unable to drive the vehicle along the path.

The results clearly show that the application of the nonlinear vehicle model coupled with the nonlinear tire model for state prediction has a significant advantage over the linear vehicle model and linear tire model couple. Using the nonlinear model, the controller is able to drive the vehicle stable at higher speeds and more accurately than with the linear

model. Furthermore, the effectiveness of the proposed LTV-MPC method is also confirmed in this section by testing the controller in the sine wave path.

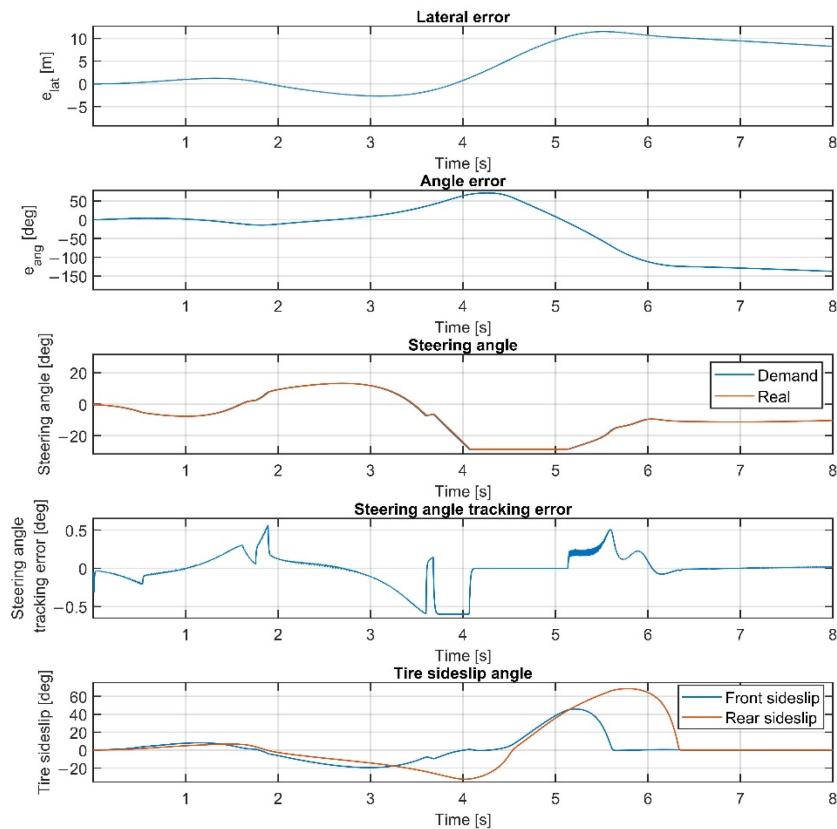


Figure 18. Simulation results on the sine wave path at 70 km/h, including steering dynamics, using linear the vehicle model for state prediction.

7. Conclusions

In this paper, an LTV-MPC structure is proposed for path-following scenarios. The controller uses a nonlinear vehicle model coupled with a Pacejka tire model for state prediction. Furthermore, the dynamics of the steering system are also incorporated into the prediction model. The reference definition is embedded into the LTV-MPC, corresponding to the states in the plant model, which results in a simplified plant model, removing two equations from the model, which are responsible for the coordinate transformation. The presented reference definition method can reduce the nonlinear terms in the path-following problem definition by transforming the path into the ego frame, resulting in a simplified description of the vehicle dynamics, the nonlinearities of which are easier to handle. Using this method, the transformation from the ego frame to the global frame is practically conducted separately from the vehicle model.

During the simulation tests, the inclusion of the steering dynamics results in a more accurate path-following performance with smaller errors and smoother steering command demanded by the controller. Furthermore, the proposed reference definition is proven to be effective, while the nonlinear terms in the plant model are reduced. The proposed controller structure coupled with the plant model can handle the vehicle at sharp corners when the tires operate in the nonlinear region. The vehicle drives stably during the tests, while the errors remain low. According to the results, the controller is applicable for even emergency scenarios, e.g., for a double-lane change. The effectiveness of the proposed controller is tested in a sine wave path, where the controller is proven to be as accurate as on the double-lane-change path. Furthermore, the effect of the linear and nonlinear plant

model is also analyzed and the advantage of applying the nonlinear plant model is clearly proven on the sine wave path.

Regarding future research opportunities, the vehicle plant model and the model of the steering system, as an actuator, could be further detailed, which could result in more accurate state prediction. The dynamics of the steering system could be modeled as a function of the vehicle velocity, the yaw rate, and the lateral acceleration, and the self-aligning torque of the tires could be incorporated into the plant model as well. Furthermore, the proposed reference definition may reduce the computational requirements of an NLMPG, as the plant model has fewer nonlinear terms.

Author Contributions: Conceptualization, Á.D. and V.T.; Formal analysis, Á.D. and V.T.; Investigation, Á.D.; Methodology, Á.D.; Project administration, V.T.; Resources, V.T.; Software, Á.D.; Supervision, V.T.; Validation, Á.D.; Writing—review & editing, Á.D. and V.T. All authors have read and agreed to the published version of the manuscript.

Funding: The research reported in this paper and carried out at the Budapest University of Technology and Economics was supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bárdos, Á.; Domina, Á.; Tihanyi, V.; Szalay, Z.; Palkovics, L. Implementation and experimental evaluation of a MIMO drifting controller on a test vehicle. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; pp. 1472–1478. [[CrossRef](#)]
2. Czibere, S.; Domina, Á.; Bárdos, Á.; Szalay, Z. Model predictive controller design for vehicle motion control at handling limits in multiple equilibria on varying road surfaces. *Energies* **2021**, *14*, 6667. [[CrossRef](#)]
3. Orgován, L.; Bécsi, T.; Aradi, S. Autonomous drifting using reinforcement learning. *Period. Polytech. Transp. Eng.* **2021**, *49*, 292–300. [[CrossRef](#)]
4. Lu, Z.; Shyrokau, B.; Boulkroune, B.; van Aalst, S.; Happee, R. Performance benchmark of state-of-the-art lateral path-following controllers. In Proceedings of the 2018 IEEE 15th International Workshop on Advanced Motion Control (AMC), Tokyo, Japan, 9–11 March 2018; pp. 541–546. [[CrossRef](#)]
5. Viana, I.B.; Kanchwala, H.; Ahiska, K.; Aouf, N. A comparison of trajectory planning and control frameworks for cooperative autonomous driving. *J. Dyn. Syst. Meas. Control* **2021**, *143*, 071002. [[CrossRef](#)]
6. Viana, I.B.; Kanchwala, H.; Aouf, N. Cooperative trajectory planning for autonomous driving using nonlinear model predictive control. In Proceedings of the 2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE), Graz, Austria, 4–8 November 2019; pp. 1–6. [[CrossRef](#)]
7. Kianfar, R.; Falcone, P.; Frederikson, J. A distributed model predictive control approach to active steering control of string stable cooperative vehicle platoon. *IFAC Proc. Vol.* **2013**, *46*, 750–755. [[CrossRef](#)]
8. Wu, N.; Huang, W.; Song, Z.; Wu, X.; Zhang, Q.; Yao, S. Adaptive dynamic preview control for autonomous vehicle trajectory following with DDP based path planner. In Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, Korea, 28 June–1 July 2015; pp. 1012–1017.
9. Alcalá, E.; Puig, V.; Quevedo, J.; Rosolia, U. Autonomous racing using Linear Parameter Varying-Model Predictive Control (LPV-MPC). *Control Eng. Pract.* **2020**, *95*, 104270. [[CrossRef](#)]
10. Falcone, P.; Borelli, F.; Asgari, J.; Tseng, H.E.; Hrovat, D. Predictive active steering control for autonomous vehicle systems. *IEEE Trans. Control Syst. Technol.* **2007**, *15*, 566–580. [[CrossRef](#)]
11. Katriniok, A.; Maschuw, J.P.; Christen, F.; Eckstein, L.; Abel, D. Optimal vehicle dynamics control for combined longitudinal and lateral autonomous vehicle guidance. In Proceedings of the 2013 European Control Conference (ECC), Zurich, Switzerland, 17–19 July 2013; pp. 974–979.
12. Xu, Y.; Tang, W.; Chen, B.; Qiu, L.; Yang, R. A model predictive control with preview-follower theory algorithm for trajectory tracking control in autonomous vehicles. *Symmetry* **2021**, *13*, 381. [[CrossRef](#)]

13. Wang, Z.; Bai, Y.; Wang, J.; Wang, X. Vehicle path tracking LTV-MPC controller parameter selection considering CPU computational load. *J. Dyn. Syst. Meas. Control* **2018**, *141*, 051004. [[CrossRef](#)]
14. Borrelli, F.; Falcone, P.; Keviczky, T.; Asgari, J.; Hrovat, D. MPC-based approach to active steering for autonomous vehicle systems. *Int. J. Veh. Auton. Syst.* **2005**, *3*, 265–291. [[CrossRef](#)]
15. Yu, C.; Zheng, Y.; Shyrokau, B.; Ivanov, V. MPC-based path following design for automated vehicles with rear wheel steering. In Proceedings of the IEEE International Conference on Mechatronics, ICM 2021, Piscataway, NJ, USA, 7–9 March 2021. [[CrossRef](#)]
16. Chowdhri, N.; Ferranti, L.; Iribarren, F.S.; Shyrokau, B. Integrated nonlinear model predictive control for automated driving. *Control Eng. Pract.* **2021**, *106*, 104654. [[CrossRef](#)]
17. Falcone, P.; Borrelli, F.; Tseng, H.E.; Asgari, J.; Hrovat, D. Linear timevarying model predictive control and its application to active steering systems: Stability analysis and experimental validation. *Int. J. Robust Nonlinear Control* **2007**, *18*, 862–875. [[CrossRef](#)]
18. Falcone, P.; Tseng, H.E.; Borrelli, F.; Asgari, J.; Hrovat, D. MPC-based yaw and lateral stabilisation via active front steering and braking. *Veh. Syst. Dyn.* **2008**, *46*, 611–628. [[CrossRef](#)]
19. Ducajú, J.M.S.; Llobregat, J.J.S.; Cuenca, Á.; Tomizuka, M. Autonomous ground vehicle lane-keeping LPV model-based control: Dual-rate state estimation and comparison of different RealTime control strategies. *Sensors* **2021**, *21*, 1531. [[CrossRef](#)] [[PubMed](#)]
20. Zhang, W.; Wang, Z.; Drugge, L.; Nybacka, M. Evaluating model predictive path following and yaw stability controllers for over actuated autonomous electric vehicles. *IEEE Trans. Veh. Technol.* **2020**, *69*, 12807–12821. [[CrossRef](#)]
21. Wei, S.; Zou, Y.; Zhang, X.; Zhang, T.; Li, X. An integrated longitudinal and lateral vehicle following control system with radar and vehicle-to-vehicle communication. *IEEE Trans. Veh. Technol.* **2019**, *68*, 1116–1127. [[CrossRef](#)]
22. Ni, L.; Gupta, A.; Falcone, P.; Johannesson, L. Vehicle lateral motion control with performance and safety guarantees. *IFAC-PapersOnLine* **2016**, *49*, 285–290. [[CrossRef](#)]
23. Moradi, M.; Fekih, A. A stability guaranteed robust fault tolerant control design for vehicle suspension systems subject to actuator faults and disturbances. *IEEE Trans. Control Syst. Technol.* **2015**, *23*, 1164–1171. [[CrossRef](#)]
24. Jin, X.; Wang, J.; Yan, Z.; Xu, L.; Yin, G.; Chen, N. Robust Vibration Control for Active Suspension System of In-Wheel-Motor-Driven Electric Vehicle Via μ -Synthesis Methodology. *J. Dyn. Syst. Meas. Control* **2022**, *144*, 051007. [[CrossRef](#)]
25. Pacejka, H.B.; Bakker, E.; Nyborg, L. *Tyre Modelling for Use in Vehicle Dynamics Studies*; SAE Technical Paper Series; SAE International: Warrendale, PA, USA, 1987; p. 870421.
26. Schwenzer, M.; Ay, M.; Bergs, T.; Abel, D. Review on model predictive control: An engineering perspective. *Int. J. Adv. Manuf. Technol.* **2021**, *117*, 1327–1349. [[CrossRef](#)]
27. Rákos, O.; Aradi, S.; Bécsi, T. Lane Change Prediction Using Gaussian Classification, Support Vector Classification and Neural Network Classifiers. *Period. Polytech. Transp. Eng.* **2020**, *48*, 327–333. [[CrossRef](#)]