

Article

Predictive Path-Tracking Control of an Autonomous Electric Vehicle with Various Multi-Actuation Topologies

Chenhui Lin ^{1,*}, Boyuan Li ^{1,2}, Efstathios Siampis ¹, Stefano Longo ¹ and Efstathios Velenis ¹

¹ Advanced Vehicle Engineering Centre, Cranfield University, Bedford MK43 0AL, UK; liboyuan@zhejianglab.com (B.L.); efstathios.siampis@cranfield.ac.uk (E.S.); s.longo@cranfield.ac.uk (S.L.); e.velenis@cranfield.ac.uk (E.V.)

² Research Center for Intelligent Transportation, Zhejiang Lab, Hangzhou 311121, China

* Correspondence: chenhui.lin@cranfield.ac.uk

Abstract: This paper presents the development of path-tracking control strategies for an over-actuated autonomous electric vehicle. The vehicle platform is equipped with four-wheel steering (4WS) as well as torque vectoring (TV) capabilities, which enable the control of vehicle dynamics to be enhanced. A nonlinear model predictive controller is proposed taking into account the nonlinearities in vehicle dynamics at the limits of handling as well as the crucial actuator constraints. Controllers with different actuation formulations are presented and compared to study the path-tracking performance of the vehicle with different levels of actuation. The controllers are implemented in a high-fidelity simulation environment considering scenarios of vehicle handling limits. According to the simulation results, the vehicle achieves the best overall path-tracking performance with combined 4WS and TV, which illustrates that the over-actuation topology can enhance the path-tracking performance during conditions under the limits of handling. In addition, the performance of the over-actuation controller is further assessed with different sampling times as well as prediction horizons in order to investigate the effect of such parameters on the control performance, and its capability for real-time execution. In the end, the over-actuation control strategy is implemented on a target machine for real-time validation. The control formulation proposed in this paper is proven to be compatible with different levels of actuation, and it is also demonstrated in this work that it is possible to include the particular over-actuation formulation and specific nonlinear vehicle dynamics in real-time operation, with the sampling time and prediction time providing a compromise between path-tracking performance and computational time.

Keywords: autonomous vehicle; path tracking; multi-actuation; predictive control



Citation: Lin, C.; Li, B.; Siampis, E.; Longo, S.; Velenis, E. Predictive Path-Tracking Control of an Autonomous Electric Vehicle with Various Multi-Actuation Topologies. *Sensors* **2024**, *24*, 1566. <https://doi.org/10.3390/s24051566>

Academic Editors: Felipe Jiménez and Yongmin Zhong

Received: 16 December 2023

Revised: 16 February 2024

Accepted: 26 February 2024

Published: 28 February 2024



Copyright: © 2024 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/>).

1. Introduction

Over the past few decades, due to the increasing demands to improve the safety, efficiency and comfort of road vehicles, autonomous vehicles (AVs) have been widely considered as the next generation of road transportation. The control techniques for AVs have been rapidly developed with a great deal of research work carried out for various objectives, including path and motion planning [1–3], path tracking [4,5], obstacle detection and avoidance [6,7] and so forth. Among these topics, the fundamental function of path tracking, and the real-time realization of it, is the main focus of this paper.

Regarding the path-tracking problem, geometry-based control methods have been raised for active front-wheel steering (FWS) vehicles [8,9]. Geometric path-tracking controllers are able to track a path only with the geometry of vehicle kinematics and of the reference path, but they are less suitable for control at the limits of handling due to the lack of knowledge on vehicle dynamics. In terms of the path-tracking controllers involving vehicle dynamics, Roselli et al. proposed a path-tracking controller for lane keeping based on the H-infinity technique [10]. According to experimental tests, the H-infinity controller

was able to achieve an overall smaller lateral error in comparison with a PID controller with feedforward, leading to a smoother action in the corner. In another study, a higher-order sliding mode controller was designed for lateral control, and robustness was validated by the test results [11]. In addition to these, optimal control theory has been introduced to the scope. Lee et al. proposed an optimal path-tracking controller based on linear quadratic Gaussian control, which provided better performance than the geometry-based controllers in terms of tracking deviation [12].

Among various control methods, model predictive control (MPC) has been found to be an outstanding technique for autonomous vehicle control. In [13], various controllers including a geometric controller, linear quadratic regulator and MPC were compared in terms of their path-tracking performance, and the simulation results demonstrated that the best path-tracking performance was achieved by MPC with the minimum control effort. Yakub et al. [14] extended the study by comparing the tracking performance of MPC and linear quadratic control for different speeds, tyre–road friction coefficients and control topologies, including FWS, four-wheel steering (4WS) and FWS with direct yaw moment control (DYC). A similar conclusion was drawn that MPC is more suitable for multi-variable systems. An additional advantage of MPC is that it takes into consideration physical constraints present in the system, such as state ranges, input limitation and road boundaries [3,15]. Therefore, MPC has been extensively applied in research work for the path-tracking control of autonomous vehicles [16–20]. In [16], Raffo et al. presented an MPC controller for path tracking. The controller was based on a linear model, and a constant velocity was assumed in order to neglect the vehicle’s longitudinal dynamics. Another path-tracking controller was developed in [21] with an adaptive preview strategy, where it was demonstrated by the simulation results that the tracking error significantly increased when the vehicle approached the limits of handling condition. In the above studies, linear vehicle dynamics models are applied for control development. This could reduce the complexity of the MPC optimization problem, while on the other hand, when it comes to situations where the vehicle is equipped with multi-actuation, or is operating at extreme conditions, a linear model may no longer provide a good prediction of the vehicle dynamics as the vehicle behaviour becomes highly nonlinear.

In addition to active FWS control, techniques like DYC, torque vectoring (TV) in particular, have been extensively discussed in the literature for vehicle control. Typically, TV refers to the differential technique that varies the torque delivered on each wheel. With the development of electric vehicles (EVs), the application of independent in-wheel motors provides a more straightforward realization of TV. Various path-tracking controllers were compared in [22], some of which applied TV, while the others did not have the functionality. The results showed that the cornering response of the vehicle could be effectively improved with the application of TV, especially at the limits of handling condition, and this was due to the generation of a direct yaw moment helping in the stabilization of the vehicle. Hence, TV has been applied in several studies for the development of driver assist systems, aiming to improve the vehicle performance and guarantee a consistently safe and stable cornering response [23–25]. There is no doubt that AV control can benefit from the multi-actuation formulation, referring to the integration of TV and steering. Chen et al. developed a path-tracking controller for vehicles with four-wheel drive (4WD) as well as 4WS [26]. With a hierarchical structure, the control demand for path-tracking as well as lateral stabilization was obtained in the higher level controller, and tyre force allocation was then carried out at a lower level to achieve that. In [27], another path-tracking controller was proposed based on MPC. The controller was also hierarchical, based on an LTV system, and could be implemented in real time. The disadvantage of such a formulation is that the results may not be optimal as the control inputs are not directly integrated into the high-level strategies. What is more, as the higher levels were based on a linear system, it was hard to guarantee feasibility, especially in extreme conditions. This could hinder the vehicle from operating at the limits of handling. In [28], Acosta et al. developed a multi-actuation controller for autonomous drift control. The controller was based on nonlinear vehicle and

tyre models, and the results showed that the dynamical capability of a vehicle could be exploited by combining FWS and TV. However, a PID controller was used instead of MPC for the path-tracking purpose, which meant that the vehicle might not be able to track a complicated path properly. In addition, the controller might not be able to be implemented in real time. To the best of our knowledge, there have been few studies on multi-actuation control including rear-wheel steering (RWS), which is part of the novelty of our work.

In this paper, we present the development of path-tracking control strategies for an over-actuated autonomous EV, which is equipped with 4WS as well as TV functionality on the rear axle. The vehicle is supposed to track a desired path at a specified reference velocity, which is close to the maximum possible velocity according to the turning radius and road–tyre friction coefficient. The control design is based on nonlinear MPC (NMPC) so that the nonlinearities of the vehicle dynamics are taken into account, which is beneficial to the operation at the limits of handling. Four controllers are built and tested in this work, including FWS only, FWS with TV (FWS-TV), 4WS and 4WS with TV (4WS-TV). The four controllers are compared in terms of path-tracking performance to validate the advantage of the over-actuation topology. Further study has also been carried out to implement the proposed over-actuation controller in real time. The control performance of the 4WS-TV controller with different prediction horizons as well as sampling time is investigated and discussed. For the purpose of the real-time implementation of the controller, a compromise has been made between path-tracking performance and computational time, and an appropriate setup of prediction horizon and sampling time is identified. In the end, the control strategy is implemented on a real-time target machine for validation. The major contribution of this paper can be summarized in two parts. First, this paper proposes a control formulation based on NMPC that is compatible with different levels of actuation, and the control performance with those actuation topologies has been compared and studied. The results demonstrate that the over-actuation formulation with both TV and 4WS leads to the best path-tracking performance for an autonomous vehicle at the limits of handling, in comparison with the individual utilization of either TV or 4WS. The second contribution of this paper is the demonstration that the proposed controller including the particular over-actuation formulation as well as nonlinear vehicle dynamics can be implemented in real time. An appropriate prediction horizon and sampling time have been discovered for the controller, which strikes a balance between path-tracking performance and system complexity. Furthermore, the controller is implemented and validated in a real-time target machine to prove that it can be executed in real time. It is worth mentioning that although the work presented in this paper is based on simulation only, some practical testing has been carried out to validate the proposed 4WS-TV control strategy. Both simulation and experimental results for tracking the same reference path are demonstrated and compared in [29], which confirms the effectiveness of the controller as well as its capability of real-time implementation.

This paper is organized as follows. Section 2 introduces the modelling of vehicle dynamics. Then the optimal control problem is formulated in Section 3. Section 4 presents the simulation results of the controllers. In Section 4.1, the path-tracking performances of the four controllers are demonstrated and compared. In Section 4.2, the path-tracking performance of the over-actuated controller with different sampling times and prediction horizons is demonstrated. Finally, we demonstrate the implementation of the controller in a real-time target machine.

2. Vehicle Dynamics Modelling

Figure 1 shows the actuator topology of the specific vehicle for which the controller in this paper is designed. The vehicle in this study is a 4WD EV prototype developed by Delta Cosworth for the AID-CAV project. The powertrain includes three electric motors, responsible for both the acceleration and deceleration of the vehicle. The motor M1 drives the front wheels through an open differential on the front axle, and the rear wheels are, respectively, driven by the other two motors M2 and M3, with which torque vectoring is

performed. In addition, the autonomous EV is equipped with 4WS functionality, realized by two steer-by-wire systems. It should be mentioned that the control strategies we have developed for this specific vehicle are extendable to fit most kinds of multi-actuation configuration of EVs, like an individual hub motor for each wheel.

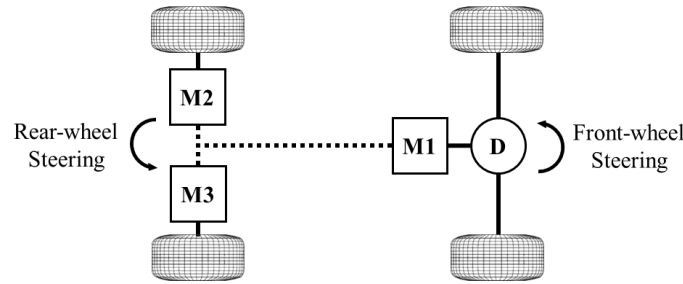


Figure 1. Actuator topology of the case study vehicle.

2.1. Equations of Motion

A two-track vehicle model is applied for the design of the path-tracking controller. The model is formulated around the vehicle's centre of gravity (CoG), and with assumptions that the vehicle travels on a horizontal plane, while the pitch, roll and heave motion are neglected. The model is shown in Figure 2. Focusing on the longitudinal, lateral and yaw motion of the vehicle, the equations of motion can be derived based on Newton–Euler equations using the present tyre forces, and are given as follows:

$$m(\dot{V}_x - V_y r) = (F_{FLx} + F_{FRx}) \cos \delta_F - (F_{FLy} + F_{FRy}) \sin \delta_F + (F_{RLx} + F_{RRx}) \cos \delta_R - (F_{RLy} + F_{RRy}) \sin \delta_R \quad (1)$$

$$m(\dot{V}_y + V_x r) = (F_{FLx} + F_{FRx}) \sin \delta_F - (F_{FLy} + F_{FRy}) \cos \delta_F + (F_{RLx} + F_{RRx}) \sin \delta_R - (F_{RLy} + F_{RRy}) \cos \delta_R \quad (2)$$

$$\begin{aligned} I_z \dot{r} = & l_F \cdot (F_{FLx} + F_{FRx}) \sin \delta_F + l_F \cdot (F_{FLy} + F_{FRy}) \cos \delta_F \\ & - l_R \cdot (F_{RLx} + F_{RRx}) \sin \delta_R - l_R \cdot (F_{RLy} + F_{RRy}) \cos \delta_R \\ & - w_L \cdot (F_{FLx} \cos \delta_F - F_{FLy} \sin \delta_F) \\ & - w_L \cdot (F_{RLx} \cos \delta_R - F_{RLy} \sin \delta_R) \\ & + w_R \cdot (F_{FRx} \cos \delta_F - F_{FRy} \sin \delta_F) \\ & + w_R \cdot (F_{RRx} \cos \delta_R - F_{RRy} \sin \delta_R), \end{aligned} \quad (3)$$

where m and I_z stand for the mass of the vehicle and its moment of inertia about the vertical axis through the CoG. l_F and l_R represent the distances from the CoG to the front and rear axle, while w_L and w_R refer to the left and right portions of the track width divided by the CoG. The longitudinal velocity, lateral velocity and yaw rate of the vehicle are denoted by V_x , V_y and r , respectively. F_{ijk} ($i = F, R$, $j = L, R$, $k = x, y$) stand for the longitudinal and lateral tyre forces, while δ_F and δ_R represent the steering angles of the vehicle on the front and rear wheels, respectively.

For the path-tracking purpose, it is fundamental to include the vehicle's position in the model. In this paper, the vehicle's position is identified in the Cartesian coordinate system, and the derivatives of the vehicle's position coordinates X and Y as well as yaw angle Ψ can be calculated as:

$$\dot{X} = V_x \cos(\Psi) - V_y \sin(\Psi) \quad (4)$$

$$\dot{Y} = V_x \sin(\Psi) + V_y \cos(\Psi) \quad (5)$$

$$\dot{\Psi} = r \quad (6)$$

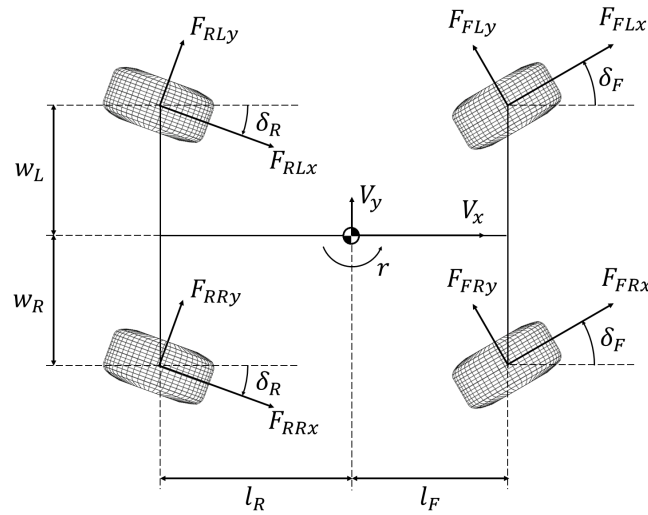


Figure 2. Schematic diagram of the vehicle model.

2.2. Tyre Model

When operating near the adhesive limits, vehicle dynamics become highly nonlinear due to the characteristics of tyre force. Thus, it is important to apply an appropriate tyre model when developing control strategies for such extreme operating conditions. Since control development is focused on planar motion, the pitch and roll of the vehicle as well as the vertical motion of the sprung mass can be neglected. Under these assumptions, the vertical load at each wheel can be calculated as a combination of the static weight on each corner and the transferred load resulting from both longitudinal and lateral acceleration. The total vertical loads at each wheel F_{ijz} are given by:

$$F_{FLz} = F_{FLz0} + \frac{mh}{lw} \cdot (-w_R a_x - l_R a_y) \quad (7)$$

$$F_{FRz} = F_{FRz0} + \frac{mh}{lw} \cdot (-w_L a_x + l_R a_y) \quad (8)$$

$$F_{RLz} = F_{RLz0} + \frac{mh}{lw} \cdot (w_R a_x - l_F a_y) \quad (9)$$

$$F_{RRz} = F_{RRz0} + \frac{mh}{lw} \cdot (w_L a_x - l_F a_y), \quad (10)$$

where a_x and a_y are the longitudinal and lateral acceleration of the vehicle, and h denotes the height of the CoG from the ground. F_{ijz0} are the static vertical forces on each wheel.

The side slip angles on the front and rear tyres can be calculated by the following equations. It is assumed that the side slip angles are the same at the left and right tyres.

$$\alpha_F = \arctan \frac{V_y + l_F \cdot r}{V_x} - \delta_F \quad (11)$$

$$\alpha_R = \arctan \frac{V_y - l_R \cdot r}{V_x} - \delta_R \quad (12)$$

By applying appropriate constraints on the control inputs, an assumption can be made that the tyres do not go beyond their adhesion limit in the longitudinal direction, and hence, the rotational dynamics of the wheels can be neglected [30]. In this case, the longitudinal tyre force is supposed to be proportional to the driving or braking torque applied on the wheels. As mentioned in Section 1, the front wheels are driven by a single motor through an open differential; the torque input on the front axle is evenly distributed on the two front

wheels. With T_F , T_{RL} and T_{RR} representing the three torque inputs from motors M1, M2 and M3, the longitudinal tyre force of the vehicle on each wheel can be calculated as follows:

$$F_{Fjx} = \frac{T_F/2}{R_w} \quad (13)$$

$$F_{Rjx} = \frac{T_{Rj}}{R_w} \quad (14)$$

The maximum tyre force can be calculated as

$$F_{ij,max} = \mu \cdot F_{ijz}, \quad (15)$$

where μ refers to the tyre–road friction coefficient. To maintain the crucial coupling between longitudinal and lateral tyre forces, the maximum available lateral tyre force is determined by the friction circle

$$F_{ijy,max} = \sqrt{F_{ij,max}^2 - F_{ijz}^2} \quad (16)$$

Thus, the lateral tyre force on individual wheels can be calculated by the simplified Pacejka's Magic Formula tyre model [31]

$$F_{ijy} = -F_{ijz} \cdot D \sin(C \arctan(B\alpha_i)) \quad (17)$$

3. Predictive Path-Tracking Controller

The nonlinear continuous-time system can be described as:

$$\dot{x}_t = fc(x_t, u_t), \quad (18)$$

where x_t stands for the state vector $[V_x, V_y, r, X, Y, \Psi]^T$ and u refers to the control input vector. With regards to the actuation configuration, there are four formulations to be investigated in this paper, including FWS, FWS-TV, 4WS and 4WS-TV. For the FWS and FWS-TV formulations, the RWS angle is not included in the controller and thus remains zero. For the FWS and 4WS formulations, it is assumed that the driving torque delivered on each wheel equals the same, which means the torque from the three motors have the relationship:

$$\frac{T_F}{2} = T_{RL} = T_{RR} = T_w, \quad (19)$$

where T_w is the actual control input in the FWS and 4WS formulations. Table 1 shows the actuation configuration of each formulation, and the control input vector u of each of them is shown as follows:

$$\begin{aligned} u_{FWS} &= [\delta_F, T_w]^T \\ u_{FWS-TV} &= [\delta_F, T_F, T_{RL}, T_{RR}]^T \\ u_{4WS} &= [\delta_F, \delta_R, T_w]^T \\ u_{4WS-TV} &= [\delta_F, \delta_R, T_F, T_{RL}, T_{RR}]^T \end{aligned} \quad (20)$$

Table 1. Actuation topologies of the controllers.

Controller	Steering	Driving Torque
FWS	FWS only	Same on each wheel
4WS	FWS + RWS	Same on each wheel
FWS-TV	FWS only	TV applied on rear wheels
4WS-TV	FWS + RWS	TV applied on rear wheels

The controller development is based on NMPC, and is realized in the sampled-data framework by discretizing the nonlinear continuous-time system with the explicit Runge–

Kutta 4th order method. The main purpose of the controller is to follow the reference path at the reference velocity, and the discrete NMPC problem is formulated as

$$\begin{aligned} \min_{x,u} \quad & \sum_{k=0}^{N-1} (x_{k+1} - x_{ref,k+1})^T Q (x_{k+1} - x_{ref,k+1}) + u_k^T R u_k \\ \text{st.} \quad & x_0 = x_{initial} \\ & x_{k+1} = f_d(x_k, u_k), k = 0, \dots, N-1 \\ & x_{min} \leq x_k \leq x_{max}, k = 0, \dots, N-1 \\ & u_{min} \leq u_k \leq u_{max}, k = 0, \dots, N-1, \end{aligned} \quad (21)$$

where N is the prediction horizon steps, x_{ref} is the reference for state vector x , namely $[V_{x,ref}, V_{y,ref}, r_{ref}, X_{ref}, Y_{ref}, \Psi_{ref}]^T$, and Q, R are the weighting matrices of the state and control input vectors, respectively. f_d represents the discrete-time system obtained from f_c . In terms of the constraints on control commands, it is worth mentioning that despite the use of box constraints in this paper, the formulation can easily take a time-varying constraint on the control inputs.

The reference path is parametrized by the arc length S along the path from the origin point, where $S \in [0, L]$ and L is the total length of the path. With this parameterization, the position $X_{ref}(S), Y_{ref}(S)$ of any point on the reference path can be obtained by calculating the third order polynomial for the argument S . In addition, the tangential angle of the path at the point can be obtained as

$$\Psi_{ref}(S) = \arctan \frac{\partial Y_{ref}(S)}{\partial X_{ref}(S)}, \quad (22)$$

and is used as the reference yaw angle of the vehicle. This parameterization takes advantage of the known waypoints on the reference path and provides an accurate enough interpolation within them [20].

For path-tracking purposes, the relative position of the vehicle with regards to the reference path is required. Point $(X_{ref}(S_0), Y_{ref}(S_0))$ is proposed as the projection of the vehicle position on the reference path, and S_0 can be obtained by solving the optimization problem

$$S_0 = \min_S \sqrt{[X - X_{ref}(S)]^2 + [Y - Y_{ref}(S)]^2}. \quad (23)$$

S_0 can be used to denote the progress of the vehicle along the reference path, and the distance between the vehicle and this projection point refers to the lateral deviation of the vehicle from the path. In order to reduce the computational time, the solving of this optimization problem is carried out in a local range, which means only the waypoints within a specific range are included in the problem. The range is identified based on the vehicle's previous position and is proportional to the velocity. This could massively increase the efficiency of localization while maintaining an accurate solution under the assumption that the vehicle does not deviate far from the reference path.

For the discrete objective function, a total of N waypoints are required to generate x_{ref} . The waypoints are supposed to follow the projection point $(X_{ref}(S_0), Y_{ref}(S_0))$, with an interval of ΔS ,

$$\Delta S = S_{k+1} - S_k = V_{ref} \cdot t_s, k = 0, \dots, N-1, \quad (24)$$

where V_{ref} is the reference velocity and t_s is the sampling time of the discrete-time system. Figure 3 shows a diagram of the generated waypoints based on the feedback states of the vehicle. The reference state vector x_{ref} is then evaluated by carrying out third-order spline polynomials based on the argument S .

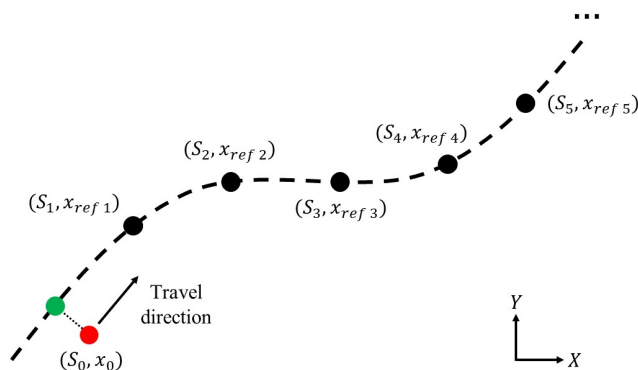


Figure 3. Diagram of the reference waypoints. The red point represents the vehicle’s position, while the green point stands for the projection of the vehicle position on the reference path. The interval of the waypoints $\Delta S = V_{ref} \cdot t_s$, where V_{ref} is the reference velocity and t_s is the sampling time of the controller.

Figure 4 shows the diagram of the complete control algorithm. The parameterization of the desired path is completed offline prior to the simulation. The solvers for the optimization problems are generated with FORCESPRO [32,33], which is an MPC and embedded optimization solution developed by Embotech.

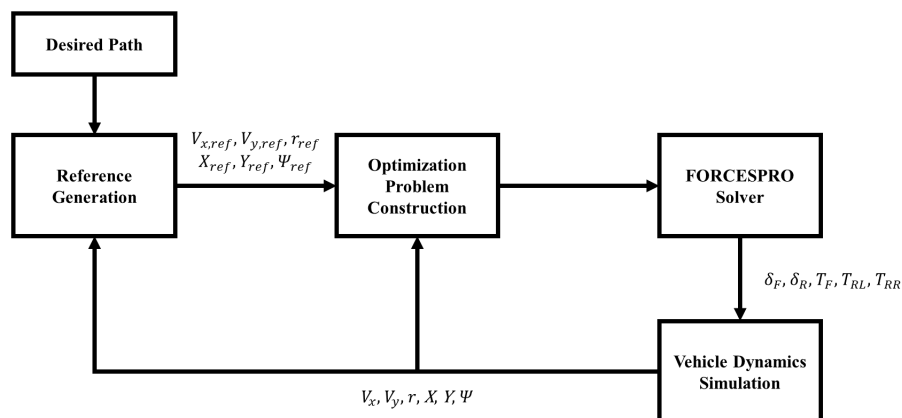


Figure 4. Diagram of the control system.

4. Simulation Results

In this section, we demonstrate the path-tracking performance of the vehicle with different control formulations and configurations according to the simulation results. Except for the real-time implementation, the simulation is carried out in the MATLAB Simulink environment. The generated NMPC solvers are imported into the Simulink model, which is connected to IPG CarMaker to provide a vehicle dynamics simulation with a high-fidelity vehicle model and scenario. All the simulation sessions are run on a workstation laptop (Intel Core i7-8750H CPU at 2.2 GHz with 32 GB RAM).

In this paper, a double U-turn scenario as shown in Figure 5 is used for testing the control performance. The desired path consists of two connected U-turns, with two straights before and after them, and the radius of both the U-turns is 20 m. Instead of the double lane change manoeuvre that is commonly used to examine vehicle stability, the double U-turn scenario is adopted in this work as the fixed curvature helps to better capture the difference in path-tracking performance brought by the different levels of actuation. In addition, the step change in curvature in the middle of the double U-turns requires the vehicle to turn sharply for path-tracking, which is challenging in terms of vehicle response and stability, especially when the vehicle is operating close to the limits of handling. This helps us to understand and compare the vehicle’s flexibility with different controllers by

seeing how close the vehicle can get to this path. In the simulation, the reference velocity of the vehicle is 52 kph, which is close to the handling limits of the vehicle corresponding to the turning radius as well as the maximum tyre–road friction coefficient. This can well demonstrate the control performance at the limits of handling, and can thus better visualize the difference among all actuation topologies. Table 2 shows the parameters of the above vehicle and tyre models. The weighting factors of MPC for each controller can be found in Equations (25) and (26), and they are obtained by scaling down the relative weight of each variable with their respective scaling factor. The scaling factors equal the square of each variable’s reasonable value and bring the different terms in the objective function to a similar level, while the relative weights are used to reflect the priority of each variable. In this work, more penalization was put on position X and Y deviation, as well as the control inputs. In order to compare the performance of different actuation topologies, it is ensured that each variable’s weighting factor is consistent across all topologies.

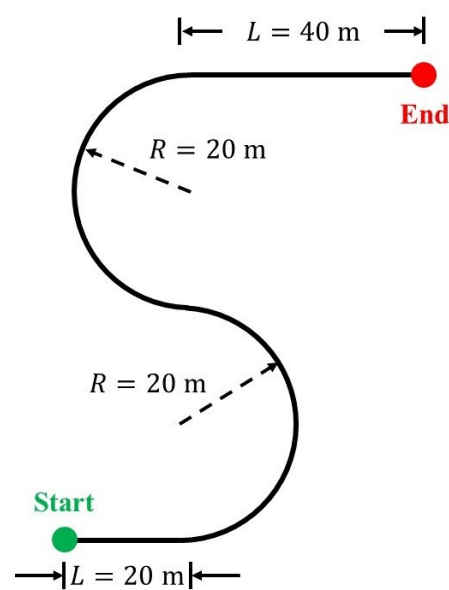


Figure 5. Reference path for the vehicle to follow. Here, L refers to the straight length, and R refers to the turning radius.

Table 2. Parameters of the vehicle model.

Parameter	Value	Parameter	Value
m [kg]	874.5	B	9.50
I_z [kg m ²]	1597.7	C	1.63
l_F [m]	0.815	D	1.16
l_R [m]	1.180	$\delta_{F,lim}$ [deg]	19
w_L [m]	0.765	$\delta_{R,lim}$ [deg]	19
w_R [m]	0.765	$T_{F,lim}$ [Nm]	800
h [m]	0.297	$T_{RL,lim}$ [Nm]	350
R_w [m]	0.315	$T_{RR,lim}$ [Nm]	350

$$Q = \text{diag}([50.0, 50.0, 16.4, 100, 100, 328.3]) \quad (25)$$

$$\begin{aligned} R_{FWS} &= \text{diag}([9848.4, 0.0011]) \\ R_{FWS-TV} &= \text{diag}([9848.4, 0.00031, 0.0011, 0.0011]) \\ R_{4WS} &= \text{diag}([9848.4, 9848.4, 0.0011]) \\ R_{4WS-TV} &= \text{diag}([9848.4, 9848.4, 0.00031, 0.0011, 0.0011]) \end{aligned} \quad (26)$$

4.1. Path-Tracking Performance of Different Actuation Topologies

In this part, the simulation results are presented regarding the path-tracking performance of the vehicle with the four controllers, which have a sampling time of 0.02 s and a prediction horizon of 1 s. The acceleration diagram of the vehicle during the path-tracking process is shown in Figure 6a. It can be seen that the vehicle is quite close to the acceleration limit identified by the tyre friction circle, which indicates that the vehicle is at the limits of handling with all the four controllers at the double U-turns. Figure 6b shows the velocity-tracking performance of the vehicle. A larger velocity deviation takes place during the sharp turn, but with the 4WS-TV controller, the vehicle tends to have the smallest velocity variation during the double U-turns. It can also be noticed that there is some steady-state deviation on the straights, which is due to the presence of penalization on torque inputs in the objective function. If the steady-state deviation is to be further reduced, more torque inputs would be required. This would lead to a larger overall objective and would not be the optimal solution from the controller's perspective. Figure 6c shows the yaw rate of the vehicle with different controllers. It can be noticed that with the FWS and FWS-TV controllers, there is a larger overshoot in the yaw rate in response to the curvature change compared with the 4WS and 4WS-TV controllers. This indicates that the application of 4WS does a better job than TV in terms of reaching the required yaw rate during cornering, while 4WS together with TV can further increase the vehicle's stability when operating at the limits of handling.

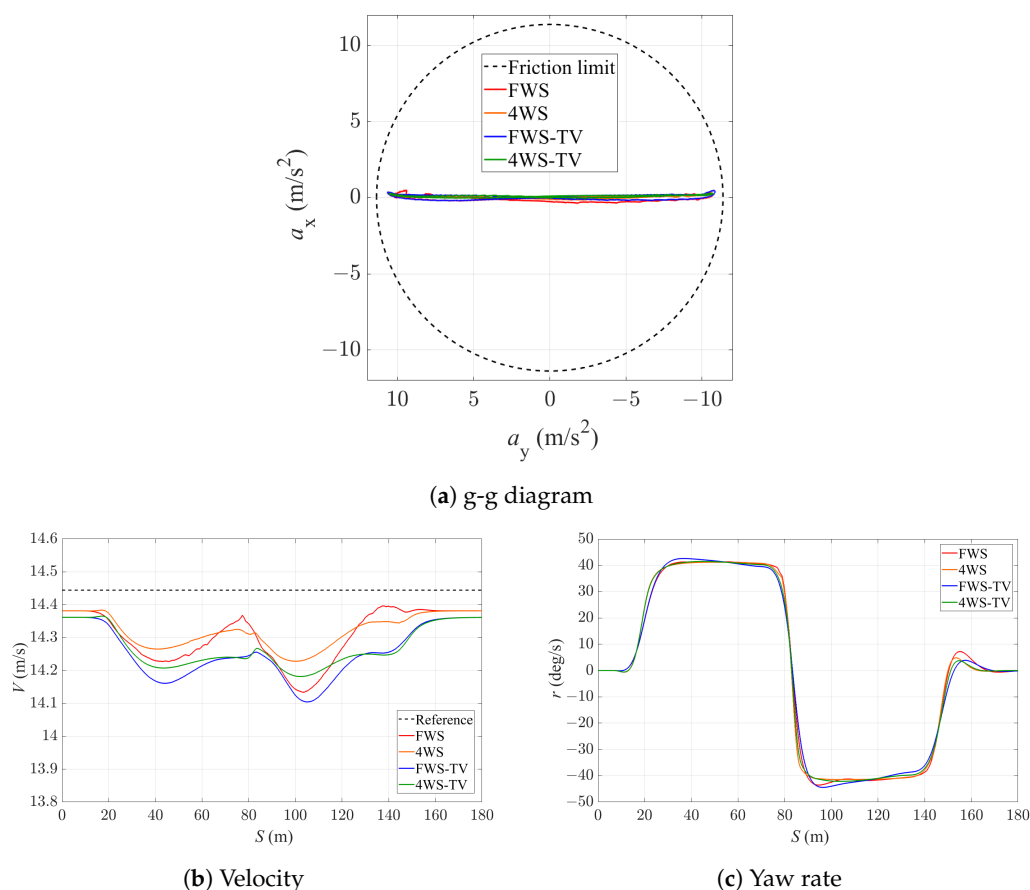


Figure 6. States of the vehicle with the four controllers.

Figure 7 shows the lateral deviation ε_y of the vehicle from the reference path with all the four controllers, and Table 3 provides a summary of the average and maximum lateral deviation with the controllers. In general, the increase in actuation leads to a smaller lateral tracking error. The vehicle obtained the largest lateral deviation up to 1.337 m with the FWS controller in the double U-turn scenario. By comparing the FWS, 4WS and

FWS-TV controllers, it can be seen that the application of either RWS or TV can significantly enhance the cornering response and reduce the lateral deviation value, while the utilization of TV seems to provide a better performance than 4WS, with the ability to reduce lateral deviation more quickly. By applying both 4WS and TV, the lateral deviation is reduced most significantly compared to the FWS case. The average tracking error is 73% smaller, and the error is around 0.4 m even at the sharp change in curvature. This proves the advantage of RWS in addition to TV for autonomous vehicle path tracking, particularly when close to the limits of handling. RWS improves the vehicle's flexibility and potential to deal with emergency scenarios that require sharp turning.

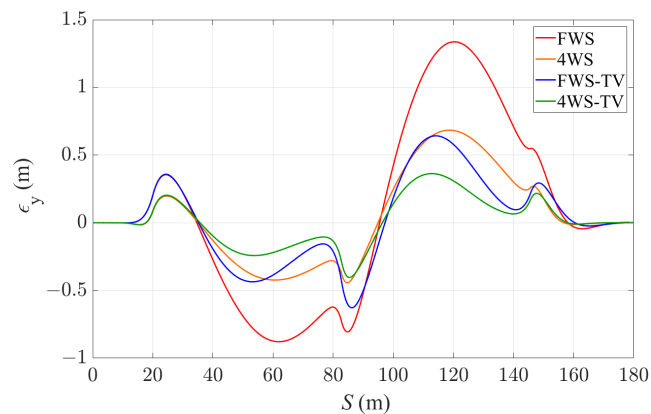


Figure 7. Lateral deviation of the vehicle with the four controllers.

Table 3. Average and maximum lateral deviation of the vehicle with the four controllers.

t_s [s]	Average $ \epsilon_y $ [m]	Maximum $ \epsilon_y $ [m]
FWS	0.531	1.337
4WS	0.256	0.684
FWS-TV	0.248	0.642
4WS-TV	0.141	0.404

Figure 8 shows the driving torques of the three motors with the four controllers. It can be observed that there is some oscillation in the driving torque commands, and this is due to the controller making efforts to track the reference velocity at the limits of handling condition. In general, it can be seen that there is less oscillation in the torque commands of the FWS-TV and 4WS-TV controllers, which indicates that with the application of TV, it tends to be easier to maintain the vehicle's handling stability. Furthermore, it can be seen from Figure 8a that the FWS-TV and 4WS-TV controllers require less driving torque on the front axle as a result of TV application. As shown in Figure 8c, the two controllers with TV generate a yaw moment on the rear axle during the turning, which helps to satisfy the required yaw rate, to enhance vehicle stability and thus to improve the path-tracking performance. In addition, due to the application of 4WS, the 4WS-TV controller requires less driving torque on the rear wheels compared with the FWS-TV controller, which requires more torque to compensate for the absence of RWS.

Figure 9 shows the steering angle commands from the four controllers to the vehicle. Figure 9a compares the front steering commands by the FWS and FWS-TV controllers, and it is shown that with the application of TV, the FWS-TV controller requires less front steering angle input than the FWS controller. In Figure 9b, both the 4WS and 4WS-TV controllers tend to have front and rear steering angle in opposite directions in response to the change in curvature, which could generate a larger yaw moment than the FWS and FWS-TV controllers to deal with the harsh change in turning direction. With the utilization of TV, the 4WS-TV controller requires less steering angles on the wheels than the 4WS controller, which is an additional advantage brought by TV.

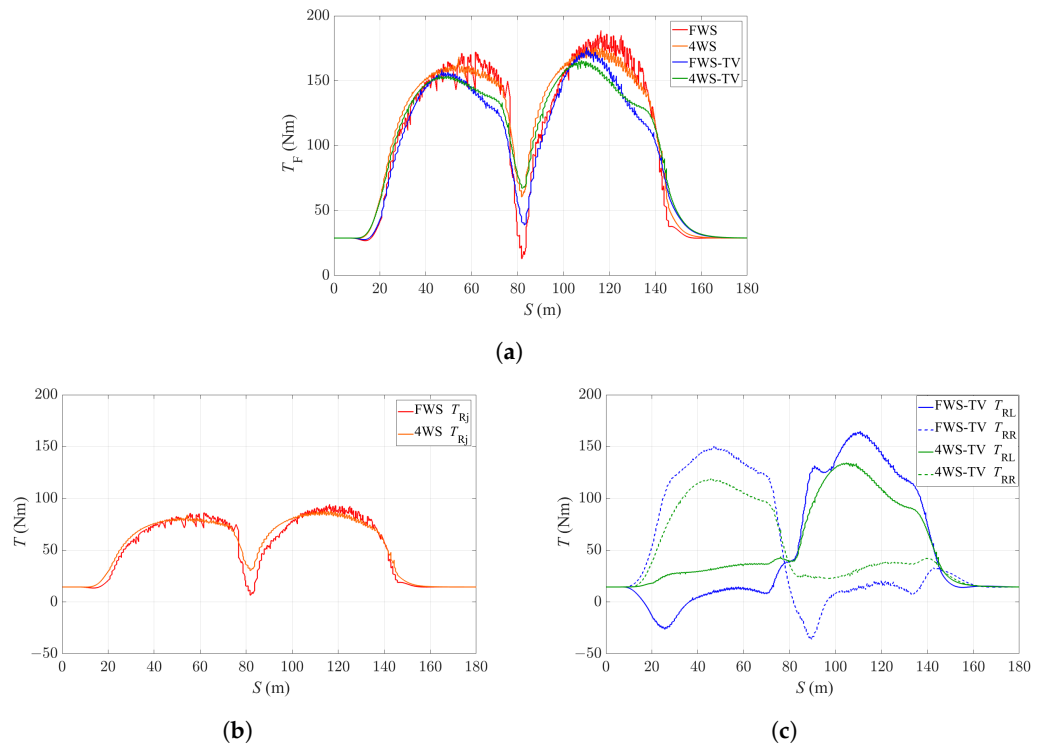


Figure 8. Driving torque commands of the four controllers. (a) Front motor torque commands. (b) Rear motor torque commands of the FWS and 4WS controllers. (c) Rear motor torque commands of the FWS-TV and 4WS-TV controllers.

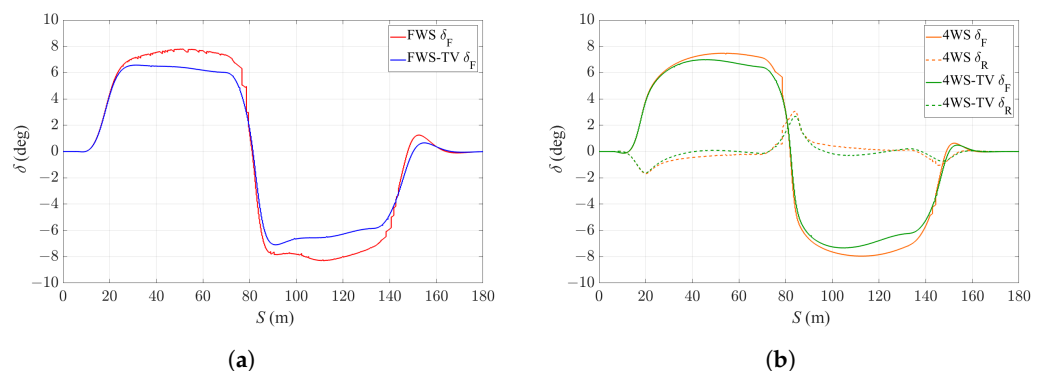


Figure 9. Steering angle commands of the four controllers. (a) Steering angle commands of FWS and FWS-TV controllers. (b) Steering commands of 4WS and 4WS-TV controllers.

Figure 10 shows the computational time of each controller. Here, the solve time refers to the time that the FORCESPRO solver takes to solve the MPC optimization problem. The FWS controller has an average solve time around 0.01 s and a maximum solve time lower than 0.03 s. In comparison, the 4WS-TV controller has an average solve time longer than 0.03 s and a maximum solve time up to 0.07 s. In general, with an increasing level of actuation, the controller takes a longer computational time due to the system complexity. The average solve times of 4WS, FWS-TV and 4WS-TV are 16%, 115% and 216% higher than that of the FWS controller. The black dashed line in Figure 10 marks the sampling time of the controllers, which is 0.02 s. This determines the maximum solve time allowance for real-time execution, and it can be seen that none of the four proposed controllers can be directly implemented in real time. The FWS and 4WS controllers have the potential for real-time operation if the maximum solve time is capped, while the FWS-TV and 4WS-TV cannot be run in real time with the current parameters.

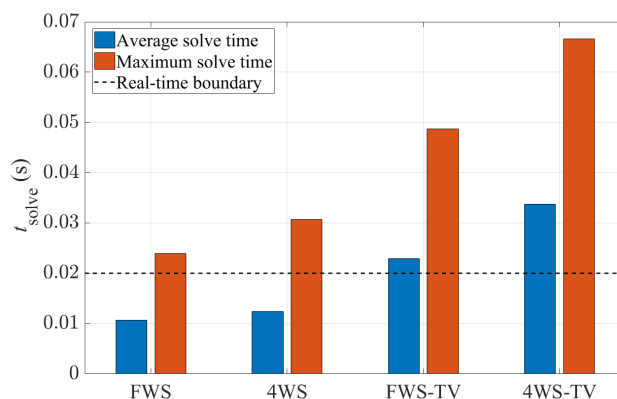


Figure 10. Computational time of the four controllers.

By comparing the simulation results of the four proposed controllers, it is obvious that both 4WS and TV are able to improve the path-tracking performance of the vehicle. TV improves the vehicle's response and stability in turning by generating a yaw moment directly. In addition to that, 4WS is able to manipulate the vehicle's motion with the direction of FWS and RWS angles; thus, the vehicle's flexibility is increased. The 4WS-TV controller provides the best path-tracking performance among the four formulations, but at the same time, it requires the most computational time among the four controllers and cannot be directly implemented in real time. Hence, further investigation is required on the simplification of the system complexity of the 4WS-TV controller for the purpose of real-time operation.

4.2. Path-Tracking Simulation Results with Different Time Parameters

In Section 4.1, the proposed 4WS-TV controller has shown its great advantage for path tracking. On the other hand, the system complexity is preventing the controller from operation in real time. In order to implement the 4WS-TV controller in real time, its system complexity needs to be reduced. One approach to do this while retaining the control formulation is to limit the prediction steps of the MPC formulation. In this part, the control performance of the 4WS-TV controller is further studied with different prediction horizons and sampling times, in order to explore the possibility of the real-time operation of the controller.

First, the effects of prediction horizon on control performance is studied. The sampling time t_s is fixed as 0.02 s, and five different prediction horizon times t of 0.6 s, 0.8 s, 1.0 s, 1.2 s and 1.4 s are used in the simulation. The lateral deviation of the vehicle is shown in Figure 11 to demonstrate the path-tracking performance, and Table 4 summarizes the average and maximum lateral deviation values with different prediction horizons. As expected, the vehicle achieves a relatively smaller lateral deviation throughout the reference path with a longer prediction horizon, with the average lateral deviation being reduced by 81% with a prediction horizon of 1.4 s, compared to a 0.6 s prediction horizon. This is because a longer prediction horizon allows the controller to plan for a further look-ahead distance, and improves the feasibility to identify the optimal control actions according to the cost function. On the other hand, it can be seen from Table 4 that the maximum lateral deviation value does not change monotonically with the prediction horizon. After the prediction horizon exceeds 1.0 s, the maximum $|\epsilon_y|$ value starts increasing with longer prediction horizons. This is because with a longer prediction horizon, the controller carries out control planning in a longer timescale, which potentially leads to a less responsive action in terms of reference tracking.

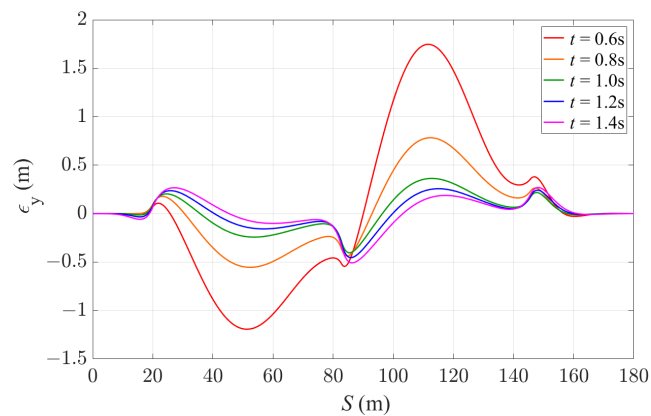


Figure 11. Lateral deviation of the vehicle with 4WS-TV controller with a fixed sampling time of 0.02 s and different prediction horizon times.

Table 4. Average and maximum lateral deviation of the vehicle with 4WS-TV controller with a fixed sampling time of 0.02 s and different prediction horizon times.

t [s]	Average $ \epsilon_y $ [m]	Maximum $ \epsilon_y $ [m]
0.6	0.613	1.747
0.8	0.280	0.783
1.0	0.141	0.404
1.2	0.122	0.454
1.4	0.117	0.508

Figure 12 shows the solve time information of the 4WS-TV controller with different prediction horizons. Despite the better path-tracking performance, the controller has a longer solve time with a longer prediction horizon. With a 0.6 s prediction horizon, the controller has the best potential to run in real time, while with a prediction horizon of 1.4 s, the average solve time is 213% higher, and is more than twice of the sampling time. According to Table 4 and Figure 12, it can be inferred that a prediction horizon time of 1.0 s achieves a relatively good balance between control performance and computational time, and thus is probably suitable for the real-time implementation of the 4WS-TV controller. However, the solve time is still higher than the limitation given by the sampling time, so next, the performance is studied with different sampling times.

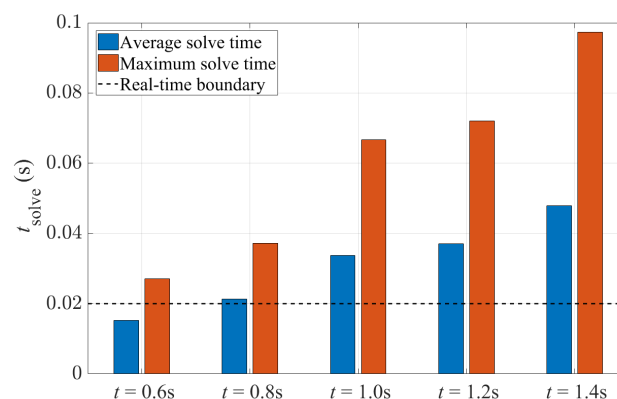


Figure 12. Solve times of the 4WS-TV controller with a fixed sampling time of 0.02 s and different prediction horizon times.

Next, the prediction horizon time t is fixed as 1.0 s, and the sampling time t_s is gradually reduced, giving increasing prediction steps of MPC. Four different sampling time are applied here in the simulation, including 0.05 s, 0.04 s, 0.025 s and 0.02 s. As shown

in Figure 13 and Table 5, a better prediction of the system dynamics could be achieved with a shorter sampling time; the controller with a 0.02 s sampling time achieves the smallest lateral deviation for path tracking.

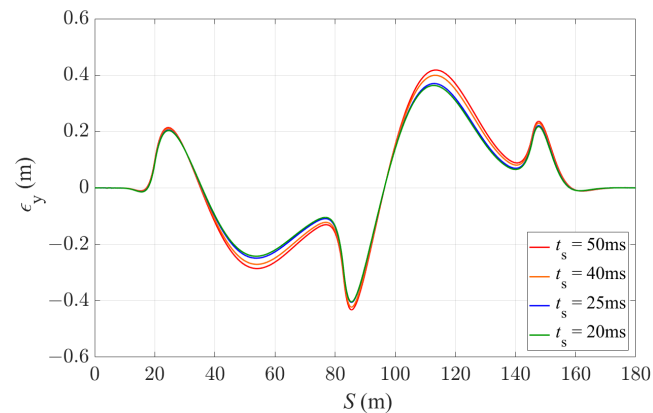


Figure 13. Lateral deviation of the vehicle with 4WS-TV controller with a fixed prediction horizon of 1.0 s and different sampling times.

Table 5. Average and maximum lateral deviation of the vehicle with 4WS-TV controller with a fixed prediction horizon of 1.0 s and different sampling times.

t_s [s]	Average $ \epsilon_y $ [m]	Maximum $ \epsilon_y $ [m]
0.05	0.164	0.432
0.04	0.151	0.423
0.025	0.145	0.406
0.02	0.141	0.404

However, the difference in path-tracking performance with various sampling times is not significant. The relative difference between the maximum and minimum value of average lateral deviation is 14%, while as shown in Figure 14, there is a large difference regarding the solve time. With a sampling time of 0.02 s, it takes 2.4 times longer on average to solve the NMPC problem, in comparison with the sampling time of 0.05 s. With sampling times of 0.05 s and 0.04 s, the 4WS-TV controller is able to guarantee operation in real time as the solve time is maintained below the boundary. In comparison, the controller cannot run in real time with a sampling time of 0.025 s or 0.02 s due to the solve time exceeding the limit. In summary, the combination of the 0.04 s sampling time and 1.0 s prediction horizon is proven to be able to obtain a good balance between control performance and computational time, and thus is suggested as being suitable for real-time implementation of the proposed 4WS-TV controller.

Finally, the 4WS-TV formulation with a 0.04 s sampling time and 1.0 s prediction horizon is implemented on a Speedgoat mobile real-time target machine (Intel Core i7-3555LE CPU at 2.5 GHz with 4 GB RAM and 4MB L2 cache). A 7DoF vehicle model based on [34] with an extension of RWS functionality is used for the real-time simulation. The vehicle dynamics simulation runs at a frequency of 1 kHz. Figure 15a and Figure 15b, respectively, show the velocity-tracking and path-tracking performance of the vehicle in the real-time simulation. The velocity-tracking error is less than 0.2 m/s, while the lateral deviation of the vehicle from the reference path is up to 0.8 m, which is similar to the former simulation results. Figure 16 shows the task execution time (TET) of the control algorithm running in Speedgoat. The capability of the proposed control strategy for real-time execution is validated as the TET remains below the sampling time of the controller. The real-time simulation results prove that the proposed 4WS-TV controller is real-time implementable, and the control performance corresponds with the simulation results in CM.

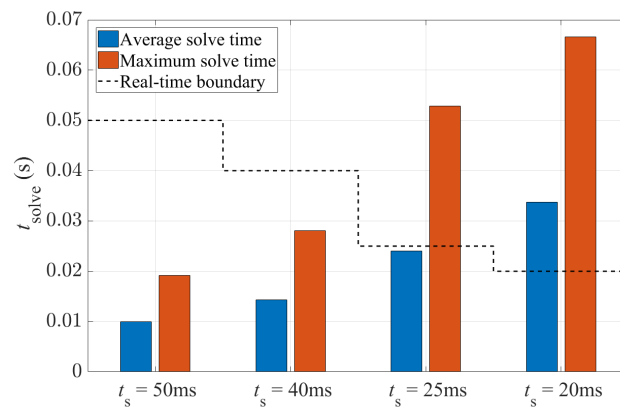
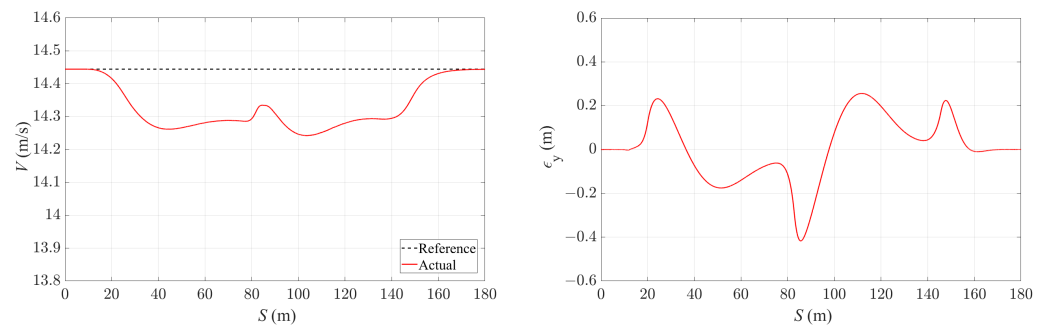


Figure 14. Solve time of the 4WS-TV controller with a fixed prediction horizon of 1.0 s and different sampling times.



(a) Velocity.

(b) Lateral tracking error.

Figure 15. Velocity and lateral tracking error of the vehicle with 4WS-TV controller under real-time implementation with 0.04 s sampling time and 1.0 s prediction horizon.

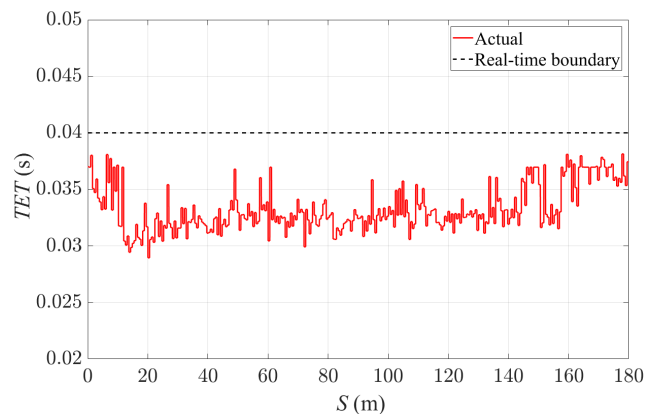


Figure 16. Task execution time of 4WS-TV controller under real-time implementation with 0.04 s sampling time and 1.0 s prediction horizon.

5. Conclusions

In this paper, we have presented a path-tracking controller for multi-actuated autonomous EVs equipped with 4WS as well as TV functionality. The controller design is based on NMPC, and the control formulation has the ability to adapt to different levels of actuation. Four controllers including FWS only, FWS with TV, 4WS and 4WS with TV are constructed and compared in the aspect of path-tracking performance. According to the simulation results, the 4WS-TV controller is able to achieve the best path-tracking performance, especially at the limits of handling condition, by exploiting the multi-actuation capability. When carrying out the same manoeuvre, the maximum lateral deviation of the vehicle from the reference path is around 0.4 m, which is significantly smaller than that with

the other three controllers. In addition, with the 4WS-TV controller, there is less oscillation in the vehicle's yaw rate in transient conditions, which indicates that the application of over-actuation formulation can enhance the vehicle's stability as well as flexibility at the limits of handling in corners.

Furthermore, it has been proven that the particular level of over-actuation can be implemented in real time with NMPC control formulation, which is another contribution of this paper. In order to reduce system complexity, the impact of prediction horizon and sampling time on the control performance has also been studied. As expected, a longer prediction horizon and shorter sampling time can, in general, have a relatively better path-tracking performance with smaller lateral deviation, at the cost of a higher computational time. Among the different combinations of prediction horizons and sampling times, a compromise has been made between path-tracking performance and computational efficiency, with the combination of a 0.04 s sampling time and 1.0 s prediction horizon time. Finally, the control strategy is implemented in a Speedgoat target machine for real-time simulation, and the simulation results prove that the proposed 4WS-TV controller including the specific nonlinearities of vehicle dynamics is able to run in real time with the particular time parameters while achieving the expected control performance.

In future work, further experimental testing is to be carried out to validate the performance with different levels of actuation. Predictive filtering will also be introduced and investigated as to how that will affect the control performance in response to signal noise and model uncertainty, which is critical in practical implementation. In addition, online path planning will be integrated into the control strategy taking into account the vehicle dynamical capability.

Author Contributions: Conceptualization, C.L. and E.V.; methodology, C.L., E.S. and E.V.; software, C.L., B.L. and E.S.; investigation, C.L.; formal analysis, C.L.; writing—original draft preparation, C.L.; writing—review and editing, B.L., S.L., E.S. and E.V.; supervision, E.S., S.L. and E.V.; project administration, E.V.; funding acquisition, E.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Innovate UK project reference 104277.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting this study are included within the article.

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

References

1. Katrakazas, C.; Quddus, M.; Chen, W.; Deka, L. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transp. Res. Part C Emerg. Technol.* **2015**, *60*, 416–442. [[CrossRef](#)]
2. Schwesinger, U.; Rufli, M.; Furgale, P.; Siegwart, R. A sampling-based partial motion planning framework for system-compliant navigation along a reference path. In Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast, Australia, 23 June 2013; pp. 391–396.
3. Bonab, S.A.; Emadi, A. Optimization-based Path Planning for an Autonomous Vehicle in a Racing Track. In Proceedings of the IECON 2019—45th Annual Conference of the IEEE Industrial Electronics Society, Lisbon, Portugal, 14–17 October 2019; Volume 1, pp. 3823–3828.
4. Faulwasser, T.; Kern, B.; Findeisen, R. Model predictive path-following for constrained nonlinear systems. In Proceedings of the 48th IEEE Conference on Decision and Control (CDC) Held Jointly with 2009 28th Chinese Control Conference, Shanghai, China, 15–18 December 2009; pp. 8642–8647.
5. Laurence, V.A.; Goh, J.Y.; Gerdes, J.C. Path-tracking for autonomous vehicles at the limit of friction. In Proceedings of the 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017; pp. 5586–5591.
6. Madãs, D.; Nosratinia, M.; Keshavarz, M.; Sundström, P.; Philippsen, R.; Eidehall, A.; Dahlén, K. On path planning methods for automotive collision avoidance. In Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast, Australia, 23–26 June 2013; pp. 931–937.
7. Funke, J.; Brown, M.; Erlien, S.M.; Gerdes, J.C. Collision Avoidance and Stabilization for Autonomous Vehicles in Emergency Scenarios. *IEEE Trans. Control Syst. Technol.* **2017**, *25*, 1204–1216. [[CrossRef](#)]

8. Coulter, R.C. *Implementation of the Pure Pursuit Path Tracking Algorithm*; Technical Report CMU-RI-TR-92-01; Carnegie Mellon University: Pittsburgh, PA, USA, 1991.
9. Hoffmann, G.M.; Tomlin, C.J.; Montemerlo, M.; Thrun, S. Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing. In Proceedings of the 2007 American Control Conference, New York, NY, USA, 9–13 July 2007; pp. 2296–2301.
10. Roselli, F.; Corno, M.; Savaresi, S.M.; Giorelli, M.; Azzolini, D.; Irilli, A.; Panzani, G. H_∞ control with look-ahead for lane keeping in autonomous vehicles. In Proceedings of the 2017 IEEE Conference on Control Technology and Applications (CCTA), Kohala Coast, HI, USA, 27–30 August 2017; pp. 2220–2225.
11. Tagne, G.; Talj, R.; Charara, A. Higher-order sliding mode control for lateral dynamics of autonomous vehicles, with experimental validation. In Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast, Australia, 23–26 June 2013; pp. 678–683.
12. Lee, K.; Jeon, S.; Kim, H.; Kum, D. Optimal Path Tracking Control of Autonomous Vehicle: Adaptive Full-State Linear Quadratic Gaussian (LQG) Control. *IEEE Access* **2019**, *7*, 109120–109133. [[CrossRef](#)]
13. K, V.; Sheta, M.A.; Gumtapure, V. A Comparative Study of Stanley, LQR and MPC Controllers for Path Tracking Application (ADAS/AD). In Proceedings of the 2019 IEEE International Conference on Intelligent Systems and Green Technology (ICISGT), Visakhapatnam, India, 29–30 June 2019; pp. 67–71.
14. Yakub, F.; Mori, Y. Comparative study of autonomous path-following vehicle control via model predictive control and linear quadratic control. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2015**, *229*, 1695–1714. [[CrossRef](#)]
15. Beal, C.E.; Gerdes, J.C. Model Predictive Control for Vehicle Stabilization at the Limits of Handling. *IEEE Trans. Control Syst. Technol.* **2013**, *21*, 1258–1269. [[CrossRef](#)]
16. Raffo, G.V.; Gomes, G.K.; Normey-Rico, J.E.; Kelber, C.R.; Becker, L.B. A Predictive Controller for Autonomous Vehicle Path Tracking. *IEEE Trans. Intell. Transp. Syst.* **2009**, *10*, 92–102. [[CrossRef](#)]
17. Alessandretti, A.; Aguiar, A.P.; Jones, C.N. Trajectory-tracking and path-following controllers for constrained underactuated vehicles using Model Predictive Control. In Proceedings of the 2013 European Control Conference (ECC), Zurich, Switzerland, 17–19 July 2013; pp. 1371–1376.
18. Zhang, H.; Heng, B.; Zhao, W. Path Tracking Control for Active Rear Steering Vehicles Considering Driver Steering Characteristics. *IEEE Access* **2020**, *8*, 98009–98017. [[CrossRef](#)]
19. Kim, W.; Kim, D.; Yi, K.; Kim, H.J. Development of a path-tracking control system based on model predictive control using infrastructure sensors. *Veh. Syst. Dyn.* **2012**, *50*, 1001–1023. [[CrossRef](#)]
20. Liniger, A.; Domahidi, A.; Morari, M. Optimization-based autonomous racing of 1:43 scale RC cars. *Optim. Control Appl. Methods* **2015**, *36*, 628–647. [[CrossRef](#)]
21. Dai, C.; Zong, C.; Chen, G. Path Tracking Control Based on Model Predictive Control with Adaptive Preview Characteristics and Speed-Assisted Constraint. *IEEE Access* **2020**, *8*, 184697–184709. [[CrossRef](#)]
22. Chatzikomis, C.; Sorniotti, A.; Gruber, P.; Zanchetta, M.; Willans, D.; Balcombe, B. Comparison of Path Tracking and Torque-Vectoring Controllers for Autonomous Electric Vehicles. *IEEE Trans. Intell. Veh.* **2018**, *3*, 559–570. [[CrossRef](#)]
23. Siampis, E.; Velenis, E.; Longo, S. Model Predictive torque vectoring control for electric vehicles near the limits of handling. In Proceedings of the 2015 European Control Conference (ECC), Linz, Austria, 15–17 July 2015; pp. 2553–2558.
24. Vasiljevic, G.; Bogdan, S. Model predictive control based torque vectoring algorithm for electric car with independent drives. In Proceedings of the 2016 24th Mediterranean Conference on Control and Automation (MED), Athens, Greece, 24 June 2016; pp. 316–321.
25. Mikuláš, E.; Gulán, M.; Takács, G. Model Predictive Torque Vectoring Control for a Formula Student Electric Racing Car. In Proceedings of the 2018 European Control Conference (ECC), Limassol, Cyprus, 12–15 June 2018; pp. 581–588.
26. Chen, T.; Chen, L.; Xu, X.; Cai, Y.; Sun, X. Simultaneous path following and lateral stability control of 4WD-4WS autonomous electric vehicles with actuator saturation. *Adv. Eng. Softw.* **2019**, *128*, 46–54. [[CrossRef](#)]
27. Guo, J.; Luo, Y.; Li, K.; Dai, Y. Coordinated path-following and direct yaw-moment control of autonomous electric vehicles with sideslip angle estimation. *Mech. Syst. Signal Process.* **2018**, *105*, 183–199. [[CrossRef](#)]
28. Acosta, M.; Kanarachos, S.; Fitzpatrick, M.E. On full MAGV lateral dynamics exploitation: Autonomous drift control. In Proceedings of the 2018 IEEE 15th International Workshop on Advanced Motion Control (AMC), Tokyo, Japan, 9–11 March 2018; pp. 529–534.
29. Lin, C.; Siampis, E.; Velenis, E. Real-Time Path-tracking MPC for an Over-Actuated Autonomous Electric Vehicle. In Proceedings of the 2022 American Control Conference (ACC), Atlanta, GA, USA, 8–10 June 2022; pp. 2006–2011.
30. Zarkadis, K.; Velenis, E.; Siampis, E.; Longo, S. Predictive Torque Vectoring Control with Active Trail-Braking. In Proceedings of the 2018 European Control Conference (ECC), Limassol, Cyprus, 12–15 June 2018; pp. 569–574.
31. Pacejka, H.B.; Bakker, E. The Magic Formula Tyre Model. *Veh. Syst. Dyn.* **1992**, *21*, 1–18. [[CrossRef](#)]
32. Domahidi, A.; Jerez, J. FORCES Professional. Embotech AG. 2014–2019. Available online: <https://embotech.com/FORCES-Pro> (accessed on 15 December 2023).

33. Zanelli, A.; Domahidi, A.; Jerez, J.; Morari, M. FORCES NLP: An efficient implementation of interior-point methods for multistage nonlinear nonconvex programs. *Int. J. Control* **2017**, *93*, 13–29. [[CrossRef](#)]
34. Velenis, E.; Katzourakis, D.; Frazzoli, E.; Tsiotras, P.; Happee, R. Steady-state drifting stabilization of RWD vehicles. *Control Eng. Pract.* **2011**, *19*, 1363–1376. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.