On the simulation and experimental analysis of imitation learning for preview-based torque vectoring control

VEHICLE SYSTEM DYNAMICS
https://doi.org/10.1080/00423114.2026.2663482

On the simulation and experimental analysis of imitation learning for preview-based torque vectoring control

Davide Lazzarini [a], Antonio Tota[a], Yuki Hosomi[b], Tona Sato[b], Binh-Minh Nguyen[b], Hiroshi Fujimoto[b] and Aldo Sorniotti[a]

[a]Department of Mechanical and Aerospace Engineering (DIMEAS), Politecnico di Torino, Turin, Italy;
[b]Department of Advanced Energy Graduate School of Frontier Sciences, The University of Tokyo, Chiba, Japan

ABSTRACT

Nonlinear model predictive control (NMPC) is a promising technology for chassis control applications, including torque vectoring control (TVC) of electric vehicles with multiple powertrains. Although NMPC can incorporate preview-based information, expected to augment performance of future active safety systems, its practical applications remain limited due to complexity and computational cost. In parallel, a few artificial intelligence (AI) TVC methods have been recently explored, but without considering imitation learning (IL). To cover the gap, this study proposes deep neural network (DNN)-based TVC, where the DNN is trained through IL of an NMPC algorithm including a 7-degree-of-freedom prediction model, and the preview of vehicle trajectory and tyre-road friction level. The application is an in-wheel motor-driven vehicle prototype operating in varying friction conditions. The simulation results highlight: (i) performance comparable to the NMPC, with turnaround time reductions exceeding 150 times; and (ii) TVC robustness to parameter uncertainties, evaluated through Monte Carlo analyses. Moreover, proof-of-concept experimental vehicle tests show that in absence of sideslip angle feedback, which significantly simplifies the estimation requirements, the proposed DNN reduces the yaw rate tracking error by $> 75\%$, compared with a real-time implementable benchmarking TVC system based on a yaw moment observer and a rule-based longitudinal tyre slip controller.

ARTICLE HISTORY
Received 11 November 2025
Revised 21 February 2026
Accepted 11 April 2026

KEYWORDS
Torque vectoring control; imitation learning; artificial intelligence; electric vehicles and electric mobility; road preview; model predictive control

List of symbols

a_f, a_r	Front and rear semi-wheelbases
a_x, a_y	Longitudinal and lateral accelerations
$a_{y,max}$	Maximum lateral acceleration magnitude
b	Front and rear track width
d_x, d_y	Preview position errors in the global reference system
F_c	Corrective factor
F_{drag}	Aerodynamic drag force
$F_{x,ij}$, $F_{y,ij}$	Longitudinal and lateral tyre forces

CONTACT Aldo Sorniotti aldo.sorniotti@polito.it

$F_{z,ij}$	Wheel load
$F_{z,ij,pred}$	Predicted wheel load
$F_{z,pred,\mathrm{RMSE}}$	Mean value of the root mean square errors of the wheel load prediction at each corner
$f(\cdot)$	Prediction model formulation
f_o	Activation function of the deep neural network output layer
$f_{\delta_{sw}}$	Steering input frequency
G_μ	Parameter for the definition of the slip angle constraint
g	Gravitational acceleration
$g(\cdot)$	Prediction model output formulation
h_{CG}	Centre of gravity height
$i = f, r$	Index or subscript referring to the front or rear axles
J	Cost function of the nonlinear model predictive control formulation
J_s	Stage cost of the nonlinear model predictive control formulation
J_t	Terminal cost of the nonlinear model predictive control formulation
J_{WT}	Auxiliary cost function of the 'SurrogateOpt' formulation
J_w	Mass moment of inertia of the wheel about its axis of rotation
J_z	Yaw mass moment of inertia of the vehicle
$j = l, r$	Index or subscript indicating the left or right vehicle sides
k	Generic index
k_x, k_y	Scaling factors of the longitudinal slip stiffness and cornering stiffnesses
k_μ	Scaling coefficient of the tyre-road friction factors
L_2	Regularization term
l_{test}	Generic index related to the 'SurrogateOpt' formulation
$M_{roll,ij}$	Rolling resistance moment
$M_{z,ref}$	Reference direct yaw moment
$M_{z,ref,\mathrm{RMS}}$	Root mean square value of the reference direct yaw moment
m	Total vehicle mass
m_{add}	Additional mass included in the electric vehicle model
m_{batch}	Mini-batch size
N_h	Number of prediction steps
n_{hid}	Number of hidden layers
n_{neu}	Number of neurons per layer
n_{test}	Number of testing scenarios of the nonlinear model predictive controller
p	Online parameter vector of the nonlinear model predictive controller
p_a	Accelerator pedal position
Q, Q_t	Stage and terminal cost function weight matrices associated with the system outputs
Q_{opt}, $Q_{t,opt}$	Optimised stage and terminal cost function weight matrices associated with the system outputs
R	Stage cost weight matrix associated with the control inputs
R_{opt}	Optimised stage cost weight matrix associated with the control inputs

R_w	Wheel radius
$s_{err,max}$	Maximum position preview error
T_1, T_2	Initial and final times considered in the key performance indicator formulations
T_{ij}, $\dot{T}_{ij}$	Reference electric motor torque and its time derivative
$T_{ij,act}$	Actual electric motor torque
$T_{lim,ij}$	Torque saturation limit of the electric machine
T_{max}, T_{rated}	Maximum and rated motor torque
$T_{tot,d}$	Total powertrain torque demand
$T_{tot,lb}$, $T_{tot,ub}$	Lower and upper boundaries on the total powertrain torque
$T_{tot,\mathrm{RMSE}}$	Root mean square value of the difference between the total powertrain torque request and the provided torque
$T_{\delta_{sw,const}}$	Duration of the constant steering phase
t	Generic time instant
t_0, t_1, t_2, t_3	Current time instant and three previous time steps with a 25 ms sampling
t_c	Control action sampling time
t_{ph}	Prediction horizon vector
t_h	Prediction horizon
t_s	Sampling time vector of the control input updates along the prediction horizon
t_δ	Steering wheel ratio
U	Sequence of control inputs along the prediction horizon
u	Control action vector of the nonlinear model predictive controller
u_k	Control inputs at a generic step of the prediction horizon
V, $\dot{V}$	Velocity and acceleration magnitude at the centre of gravity of the vehicle
V_{pred}	Predicted velocity magnitude
$V_{pred,\mathrm{RMSE}}$	Root mean square value of the velocity prediction error
w	Constant weight
$X_{CoG,id,0}$, $Y_{CoG,id,0}$	Actual vehicle centre of gravity position in the global reference system
$X_{CoG,p}$, $Y_{CoG,p}$	Future vehicle centre of gravity position in the global reference system
X_{ij}, Y_{ij}	Wheel corner coordinates in the global reference system
$X_{p,ij}$, $Y_{p,ij}$	Future wheel corner coordinates in the global reference system
$\dot{X}_{ref}$, $\dot{Y}_{ref}$	Velocity components in the global reference system
x	State vector of the nonlinear model predictive controller
$x_{0,\mathrm{DNN}^{\beta}_{\mathrm{prev},4}}$	Input vector of the deep neural network, including the sideslip angle input and four output torque values
x_k	Prediction model states at a generic step of the prediction horizon
x_{pred}	Predicted state vector
z_k, $z_{ref,k}$	Prediction model outputs and their reference values at a generic step of the prediction horizon
α_{ij}	Rear slip angle
$\alpha_{lim,fj}$, $\alpha_{lim,rj}$	Front and rear slip angle limits

$\alpha_{lim,rj,mid}$	Parameter for the slip angle constraint definition
$\alpha_{lim,rj,min}$, $\alpha_{lim,rj,max}$	Minimum and maximum threshold values of the slip angle constraints
β, $\dot{\beta}$	Sideslip angle and rate
β_1, β_2, β_3	Sideslip angle at the previous three time steps with a 25 ms sampling
$\beta_{\max}$	Maximum sideslip angle
β_{pred}	Predicted sideslip angle
$\beta_{pred,\mathrm{RMSE}}$	Root mean square value of the sideslip angle prediction error
δ_f	Average front steering angle
$\delta_{f,dyn}$	Dynamic steering wheel angle of the front wheels
$\delta_{f,id}$	Ideal preview vector of the front steering angle
δ_{sw}, $\dot{\delta}_{sw}$	Steering wheel angle and its rate
$\Delta\alpha_r$	Rear slip angle limit violation
$\Delta\alpha_{r,\max}$	Maximum rear slip angle limit violation
$\Delta\alpha_{r,\mathrm{RMS}}$	Root mean square value of the rear slip angle limit violation
Δt	Time shift of the experimental friction profile at the rear corners
$\delta_{sw,max}$	Maximum steering wheel angle during a specific simulation
ϵ_δ	Percentage error applied to the steering angle preview at the last prediction step
ϵ_σ, $\epsilon_{\alpha,f}$, $\epsilon_{\alpha,r}$	Slack variables associated with the constraints on the longitudinal tyre slip ratios and front and rear slip angles
λ	Initial learning rate
μ	Average tyre-road friction factors among the four corners
μ_{ij}	Tyre-road friction factor at the individual corner
$\mu_{rj,ths1}$, $\mu_{rj,ths2}$	Friction factor thresholds
σ_{ij}	Longitudinal tyre slip ratio
$\sigma_{lim,ij}$	Longitudinal tyre slip ratio limit
$\sigma_{r,\max}$	Maximum slip ratio magnitude at the rear wheels
τ_{em}	Electric machine torque time constant
ψ, $\dot{\psi}$, $\ddot{\psi}$	Heading angle, yaw rate and yaw acceleration
$\dot{\psi}_1$, $\dot{\psi}_2$, $\dot{\psi}_3$	Yaw rate at the previous three time steps with a 25 ms sampling time
ψ_p	Future heading angle
$\dot{\psi}_{pred}$	Predicted yaw rate
$\dot{\psi}_{pred,\mathrm{RMSE}}$	Root mean square value of the yaw rate prediction error
$\dot{\psi}_{err,int}$	Integral of the yaw rate error
$\dot{\psi}_{err,\mathrm{ISD}}$	Integral of the squared time derivative of the yaw rate error
$\dot{\psi}_{err,\mathrm{RMS}}$	Root mean square value of the yaw rate error
$\dot{\psi}_{ref}$	Reference yaw rate
ω_{ij}, $\dot{\omega}_{ij}$	Angular wheel speed and acceleration
ω_{max}, ω_{rated}	Maximum and rated motor speed

1. Introduction

Torque vectoring control, i.e. direct yaw moment control through the variation of the motor torque distribution among the wheels, has been widely studied in recent years [1]. TVC can be used to improve vehicle handling and stability, and reduce energy consumption

[2,3]. Such technology reaches its full potential in electric vehicles (EVs) with multiple powertrains, which tend to provide faster wheel torque responses than internal combustion engine-driven vehicles. The benefit is especially evident in EVs equipped with direct drive in-wheel motors (IWMs) because of the absence of a mechanical reducer and a half-shaft between the electric machine and the respective wheel. In addition, TVC can be integrated with additional actuator controls, such as rear wheel steering and active suspension [4], to further improve vehicle performance and stability.

A wide range of TVC architectures, e.g. based on proportional integral derivative (PID) control [5], H_∞ control [6], and sliding mode control (SMC) [7], have been proposed in the literature. Among them, model-based algorithms, such as linear quadratic regulators (LQRs) [8] and model predictive controllers (MPCs) [9], have shown superior performance [2,10]. Moreover, TVC can be augmented by leveraging preview information, e.g. on the future trajectory and tyre-road friction profile. This data can be acquired through on-board exteroceptive sensors, e.g. cameras [11], or vehicle-to-everything (V2X) communication [12]. Being based on the minimisation of a cost function defined along a time horizon, LQR and MPC architectures are especially suitable to account for information on the road scenario ahead [13,14]. In particular, MPCs can include formal consideration of system constraints, and, in the case of NMPC implementations, also nonlinear prediction models [10]. However, a challenge related to the real-time deployment of implicit MPC algorithms for TVC on automotive control hardware is still represented by the high computational effort and turnaround time, in relation to the typically low sampling time [15].

In this context, AI techniques have been recently applied to the TVC problem. For example, reinforcement learning (RL) approaches are based on learning by trial and error, and are used for adaptive tuning of classical controllers [16] or direct deployment of the resulting agents as TVC algorithms [17–19]. In the latter case, there is a lack of performance comparisons between the RL agents and state-of-the-art MPC implementations. However, for an adaptive cruise control application, in [20], the RL agent reaches a similar performance to a simplified MPC. Nevertheless, the capability of a well-tuned and advanced NMPC algorithm is still difficult to reach. In parallel, deep learning (DL) strategies, which train DNNs from available datasets, have been mainly used in TVC systems for the generation or estimation of relevant variables, e.g. the reference yaw rate of the fuzzy logic direct yaw moment controller in [21] or the future wheel slip profile within the TVC system in [22]. In a few studies, DL has been integrated into MPCs to enhance the fidelity of their internal model, within the so-called neural network MPCs (NNMPCs) [23].

In practical applications, the RL training phase tends to be difficult because of the initial lack of knowledge on the expected agent response and the typically limited datasets. Such challenges can be overcome through IL, which is a DL methodology aimed at cloning expert system behaviours [24]. Only very few vehicle-related studies deal with IL techniques [25,26], which can imitate high-performance and non-real-time implementable controllers, and then replace them with faster DNNs. In [27], Kim et al. use IL to emulate an NMPC-based emergency collision avoidance system with steering and friction brake actuation. In [28], a DNN imitates an NMPC-based autonomous driving controller, while reference [29] applies a similar methodology to autonomous drifting. Finally, in [30], Ajak et al. apply conditional IL to generate the reference steering wheel angle and pedal positions for autonomous driving, based on camera inputs and other sensor information. However,

there is a lack of studies dealing with (i) DNN-based TVCs obtained through IL; and (ii) the assessment of the effect of reducing the required set of variables to operate the DNN-based algorithm. For example, since vehicle sideslip angle estimation – required by current TVC systems – is typically very challenging in variable tyre-road friction conditions [31–33], the DNN generated through IL could be set not to include the sideslip information as an input, thus enabling significant simplification of the state estimation architecture.

To address the identified gaps, this study brings the following contributions:

- A systematic methodology for DNN-based TVC design and implementation, consisting of (i) vehicle model validation; (ii) target controller definition and optimisation; (iii) dataset generation with coverage of driving conditions inducing stable and unstable vehicle responses; (iv) DNN structure definition and training process; (v) simulation-based validation of the IL-generated DNN for TVC; and (vi) robustness verification through Monte Carlo analyses.
- Experimental assessment of a real-time DNN-based TVC system, featuring (i) integrated longitudinal tyre slip control functionality; (ii) road preview augmentation; (iii) feedback independence from the vehicle sideslip angle; and (iv) significantly reduced computational cost, while maintaining comparable performance to the reference high-performance NMPC algorithm.

The remainder is structured as follows: Section 2 discusses the case study vehicle and its model formulations; Section 3 deals with the preview-based NMPC algorithm, and related IL process; Section 4 presents the simulation results comparing the DNN- and NMPC-based TVC performance; finally, Section 5 experimentally assesses the DNN-based TVC, together with a benchmarking controller already present on the same vehicle demonstrator.

2. The case study vehicle and its model

2.1. The case study vehicle

The case study EV prototype, see Figure 1 and its main parameters in Table 1, is equipped with rear direct drive outrunner IWMs that can be independently controlled, and an automated steering setup on the front axle. The latter consists of an electric motor-based servosystem to track the reference steering angle profile, imposed either by a path tracking controller or in open-loop. The total wheel torque demand generation is based on a virtual accelerator pedal position, also in this case generated by the path tracking layer, operating in open-loop in the specific tests of this study. None of the involved manoeuvres implied the intervention of the friction brakes, which are not controllable on the specific EV.

The vehicle includes a dSPACE AutoBox–DS1103 rapid control prototyping unit, with a PC 750GX control board running at 933 MHz. The controlled and measured variables are monitored through the dSPACE Control Desk software. 12-bit resolvers measure the wheel speeds, while a Kalman Filter, previously tuned and validated through an optical sensor (Correvit by Corrsys-Datron), is used for accurate real-time sideslip angle estimation [34]. A dedicated inertial measurement unit (IMU) provides the yaw rate measurement; a VBOX unit measures the vehicle speed; and a KGM-10G GPS receiver provides vehicle position and heading angle.

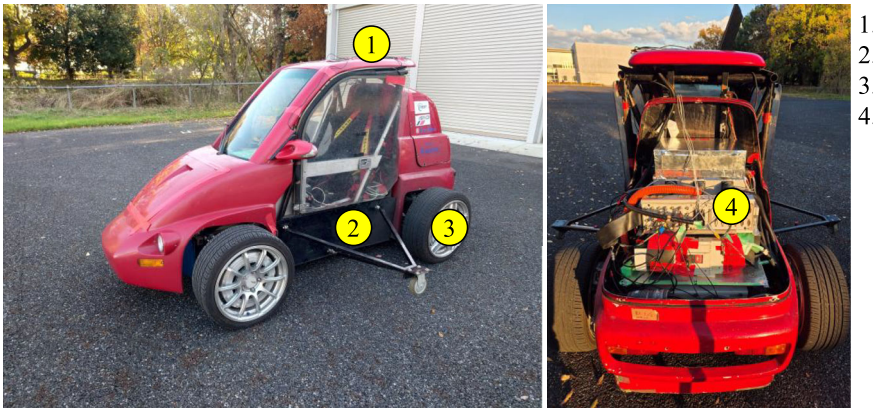


Figure 1. Views of the University of Tokyo vehicle prototype, with the indication of the actuators, sensors and a rapid control prototyping unit.

Table 1. Main vehicle parameters.

Description	Symbol	Value	Unit
Vehicle			
Total vehicle mass	m	925	[kg]
Front semi-wheelbase	a_f	0.988	[m]
Rear semi-wheelbase	a_r	0.712	[m]
Front and rear track width	b	1.3	[m]
Centre of gravity (CoG) height	h_{CG}	0.46	[m]
Yaw mass moment of inertia	J_z	617	[kgm^2]
Wheel radius	R_w	0.302	[m]
Wheel mass moment of inertia	J_w	1.24	[kgm^2]
Steering wheel ratio	t_δ	0.06	[–]
In-wheel motor			
Maximum motor torque	T_{max}	530	[Nm]
Rated motor torque	T_{rated}	127	[Nm]
Maximum motor speed	ω_{max}	1200	[rpm]
Rated motor speed	ω_{rated}	450	[rpm]
Electric machine torque time constant	τ_{em}	6	[ms]

2.2. The vehicle dynamics model

A vehicle dynamics simulation model was implemented in the Matlab-Simulink environment and was used to support the generation of the IL training datasets, as well as for the simulation-based assessment of the resulting DNNs. The dedicated Matlab-Simulink implementation of the model, rather than the adoption of a specialised vehicle dynamics simulation software, enables: (i) model generation with a limited vehicle dataset, e.g. without knowledge of the kinematic and compliance characteristics of the suspension system, which is the case of this implementation, and also rather common in typical applications and (ii) significantly faster data generation for IL agent training. The effectiveness of the methodology will be confirmed by the experimental test results.

From the mechanical viewpoint, the model has 7 degrees of freedom (DOFs), i.e. three of them associated with the vehicle body motions (longitudinal, lateral and yaw dynamics), and one per corner to consider each wheel rotation. The electric machine (EM)

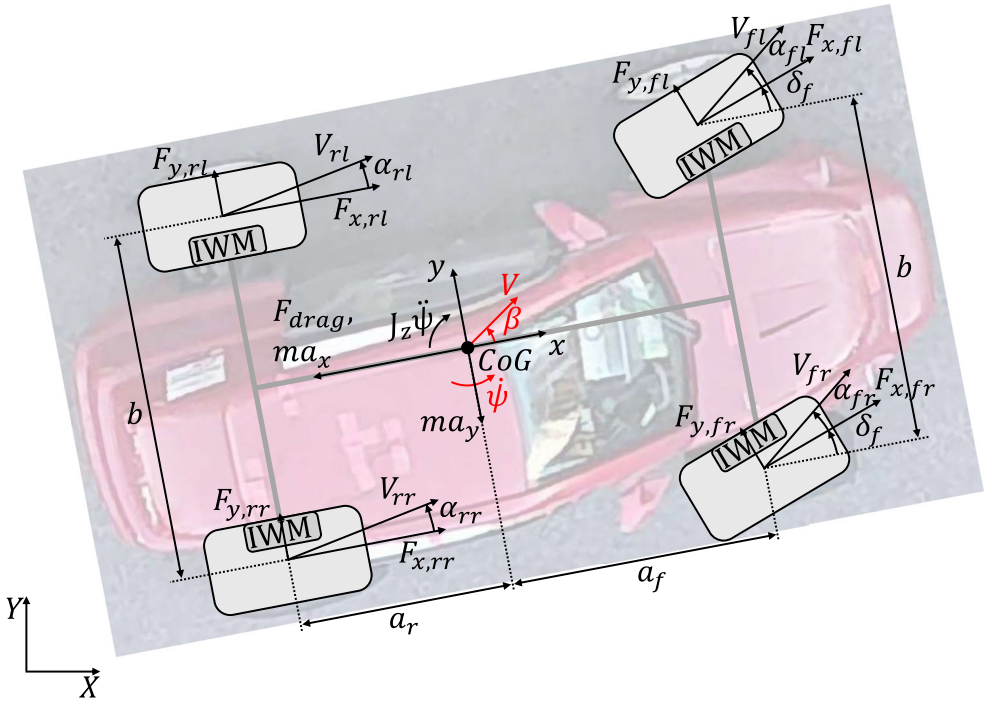


Figure 2. Simplified schematic of the vehicle top view, with the main variables and parameters involved in the simulation model.

torque dynamics are also accounted for through a first-order model. Figure 2 shows the vehicle schematic with the main parameters and variables, and the corresponding sign conventions.

The model is described by the following equations:

- Force balance along the vehicle motion direction

$$\dot{V} = \frac{1}{m}\left\{\cos(\beta)\left\{\sum_{j=l,r}[F_{x,fj}\cos(\delta_f) - F_{y,fj}\sin(\delta_f) + F_{x,rj}] - F_{drag}\right\}\right.$$
$$\left. + \sin(\beta)\sum_{j=l,r}[F_{x,fj}\sin(\delta_f) + F_{y,fj}\cos(\delta_f) + F_{y,rj}]\right\} \quad (1)$$

where V is the velocity magnitude at the vehicle centre of gravity; β is the sideslip angle; m is the total vehicle mass; δ_f is the average front steering angle; F_{drag} is the aerodynamic drag force; and $F_{x,ij}$ and $F_{y,ij}$ are the longitudinal and lateral tyre forces at the corner ij, where here and in the remainder the subscript $i = f, r$ refers to the front or rear corners, while the subscript $j = l, r$ refers to the left and right corners.

- Force balance along a perpendicular direction to the motion direction

$$\dot{\beta} = \frac{1}{mV}\left\{\cos(\beta)\sum_{j=l,r}[F_{x,fj}\sin(\delta_f) + F_{y,fj}\cos(\delta_f) + F_{y,rj}] - \sin(\beta)\left\{\sum_{j=l,r}[F_{x,fj}\cos(\delta_f) - F_{y,fj}\sin(\delta_f) + F_{x,rj}] - F_{drag}\right\}\right\} - \dot{\psi} \tag{2}$$

  where $\dot{\psi}$ is the yaw rate.
- Yaw moment balance

$$\ddot{\psi} = \frac{1}{J_z}\left\{\sum_{j=l,r} a_f[F_{x,fj}\sin(\delta_f) + F_{y,fj}\cos(\delta_f)] - a_r F_{y,rj} + \frac{b}{2}[F_{x,fr}\cos(\delta_f) - F_{y,fr}\sin(\delta_f) - F_{x,fl}\cos(\delta_f) + F_{y,fl}\sin(\delta_f) + F_{x,rr} - F_{x,rl}]\right\} \tag{3}$$

  where J_z is the yaw mass moment of inertia; a_f and a_r are the front and rear semi-wheelbases; and b is the track width, which is approximately the same on the two axles.
- Moment balance of the wheel ij about its axis of rotation

$$\dot{\omega}_{ij} = \frac{1}{J_w}[T_{ij,act} - F_{x,ij}R_w - M_{roll,ij}] \tag{4}$$

  where ω_{ij} is the angular wheel speed; J_w is the wheel mass moment of inertia – including all the relevant components rotating with the wheel – about the axis of rotation; $T_{ij,act}$ is the actual EM torque; R_w is the wheel radius; and $M_{roll,ij}$ is the rolling resistance moment.
- EM torque dynamics

$$\dot{T}_{ij} = \frac{T_{ij} - T_{ij,act}}{\tau_{em}} \tag{5}$$

  where T_{ij} is the reference motor torque, after the application of a saturation limit based on ω_{ij}, according to the torque characteristic curve of the machine; and τ_{em} is the EM torque time constant.

The tyre model is based on the simplified version of the magic formula in [2], including consideration of the interaction between longitudinal and lateral tyre forces. The tyre model provides vertical tyre load values that consider the longitudinal and lateral load transfer effects, under the assumption – already adopted in [14] – of neglecting the suspension dynamics, but considering the roll stiffness distribution effect on the lateral load transfer. For completeness, the vehicle model was implemented for both the experimentally implemented rear-wheel-drive (RWD) EV layout and the corresponding all-wheel-drive (AWD) version, the latter employed in the simulation-based assessment of the TVC algorithm.

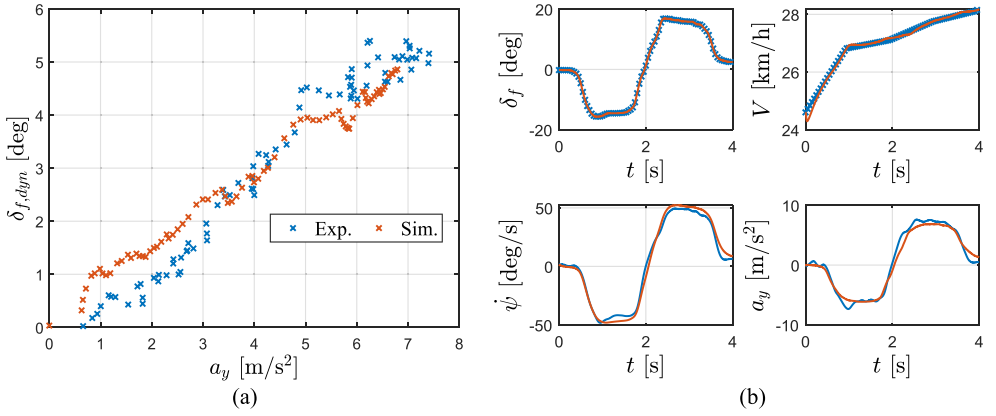


Figure 3. Vehicle model validation results along (a) ramp steer and (b) double step steer manoeuvres.

2.3. Model validation

To experimentally validate the vehicle model, multiple manoeuvres covering quasi-steady-state and transient conditions were carried out with the EV demonstrator, on high and variable friction surfaces. For example, Figure 3(a) shows the results during a ramp steer manoeuvre at 25 km/h, in terms of understeer characteristic, i.e. the plot of the dynamic steering angle at the front wheels, $\delta_{f,dyn}$ (the difference between δ_f and the kinematic steering angle), as a function of the lateral acceleration a_y. Figure 3(b) reports the time profiles of the main variables along a double-step steer manoeuvre.

3. The imitation-learning toolchain

3.1. Assumptions

This preliminary study is conducted under assumptions A1–A4.

A1. The vehicle operates on a flat surface with negligible slope and bank angles.

A2. All the relevant vehicle state variables (speed, sideslip angle, yaw rate and wheel speeds) are available in real-time, which is reasonable, given the presence of the relevant on-board sensors and observers in the case study EV but also in production vehicles.

A3. The tyre-road friction factor of the proving ground is spatially mapped offline, and is available to the controller as road preview information, which represents the future frontier of chassis control, e.g. see the analyses in [35] and [36], applied to anti-lock braking and traction control, and here to be extended to the TVC case.

A4. The reference steering wheel angle is available as preview information, which can be obtained through exteroceptive sensing on the road curvature ahead (e.g. through cameras) and/or V2X communication [37,38].

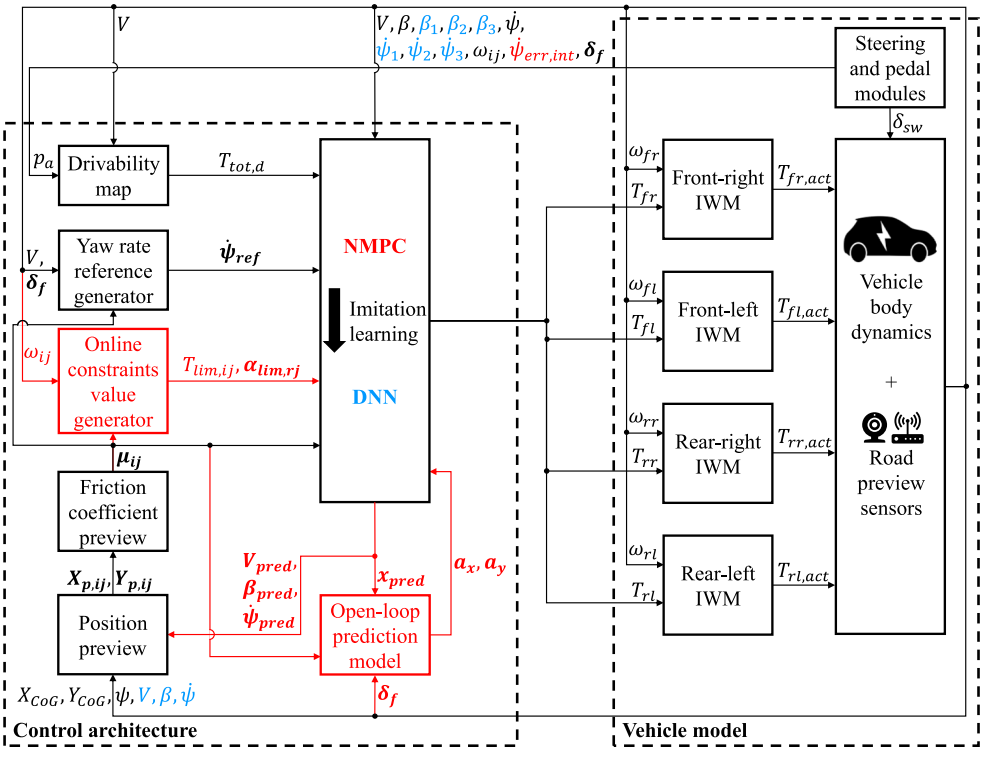


Figure 4. Control and simulation architectures for $\text{NMPC}_{\text{prev}}$ and $\text{DNN}^{\beta}_{\text{prev,4}}$. The items that are NMPC-specific are in red, those that are DNN-exclusive are in light blue, while the shared items are in black.

3.2. Control system and simulation architecture

The implicit solution of complex NMPC to be run at low sampling times t_c of vehicle dynamics control applications implies high computational effort. Hence, the proposed real-time control architecture consists of a DNN implementation for preview-based TVC, which is trained to imitate a high-performance NMPC algorithm. The control system and simulation architecture are presented in Figure 4. Since both the NMPC and resulting IL-derived DNN are included, the diagram distinguishes the items that are NMPC-specific from those that are DNN-exclusive or those that are shared. The scheme refers to two of the TVC configurations considered in the study: (i) $\text{NMPC}_{\text{prev}}$, an advanced model predictive algorithm including tyre-road friction and steering input preview; and (ii) $\text{DNN}^{\beta}_{\text{prev,4}}$, the corresponding agent incorporating the sideslip angle input. Other configurations, along with their characteristics, will be introduced later. Here and in the remainder, the bolded symbols refer to vectors whose components are the variables having the same name as the vector, discretised along the prediction horizon t_h.

The main blocks that are common to both TVC configurations are:

- The drivability map, which provides the total powertrain torque demand $T_{tot,d}$, starting from V and the accelerator pedal position p_a. To achieve regenerative braking, for $V > 10$ km/h, $T_{tot,d}$ is set to zero for a p_a value corresponding to 15% of the pedal travel,

while the torque demand linearly reaches its maximum negative (i.e. regenerative) and positive (i.e. traction) values at 0% and 100%. At very low speed, such a condition is progressively varied to prevent negative powertrain torque values when the vehicle is at a standstill.

- The yaw rate reference generator, which outputs the reference yaw rate vector $\dot{\boldsymbol{\psi}}_{\boldsymbol{ref}}$ along t_h. Firstly, a sequence of steady-state values of reference yaw rate is computed as a function of the sequence of steering inputs $\boldsymbol{\delta_f}$ and V, through a nonlinear map for high-friction conditions. The map is designed to provide a similar cornering response to the passive vehicle – i.e. the one without any control – in quasi-static conditions, and is obtained by postprocessing the results of ramp steer manoeuvres at different V values. Then, to ensure stability on low-friction surfaces, the previewed sequence of tyre-road friction factors $\boldsymbol{\mu_{ij}}$ at the individual EV corners sets a safety limit to $\dot{\boldsymbol{\psi}}_{\boldsymbol{ref}}$:

$$|\dot{\boldsymbol{\psi}}_{\boldsymbol{ref,lim}}| = \frac{\boldsymbol{a_{y,max}}}{V} = \frac{F_c g \sum_{i=f,r} \sum_{j=l,r} \boldsymbol{\mu_{ij}}}{4V} \tag{6}$$

where $\boldsymbol{a_{y,max}}$ is the vector of the maximum lateral acceleration magnitudes that are deemed achievable by the vehicle, for the sequence of average friction factors defined by $\boldsymbol{\mu} = \sum_{i=f,r} \sum_{j=l,r} \boldsymbol{\mu_{ij}}/4$; g is the gravitational acceleration; and F_c is a corrective factor.

- The position preview block, which provides the sequence of future wheel corner positions, $\boldsymbol{X_{p,ij}}$ and $\boldsymbol{Y_{p,ij}}$, in the global reference system, as part of the friction preview module. $\boldsymbol{X_{p,ij}}$ and $\boldsymbol{Y_{p,ij}}$ are determined by computing the expected future position vectors, $\boldsymbol{X_{CoG,p}}$ and $\boldsymbol{Y_{CoG,p}}$, of the vehicle CoG in the global reference frame (see the trajectory examples in Figure 5), as well as the heading angle vector $\boldsymbol{\psi_p}$, according to the following formulation:

For $k = 1$ to N_h **do**

$$X_{CoG,p}(k) = X_{CoG,p}(k-1) + t_s(k-1)V(k-1) \cdot \cos[\beta(k-1) + \psi_p(k-1)]$$

$$Y_{CoG,p}(k) = Y_{CoG,p}(k-1) + t_s(k-1)V(k-1) \cdot \sin[\beta(k-1) + \psi_p(k-1)]$$

$$\psi_p(k) = \psi_p(k-1) + t_s(k-1)\dot{\psi}(k-1)$$

End

$$\boldsymbol{X_{p,ij}} = \boldsymbol{X_{CoG,p}} \pm a_i \cos \boldsymbol{\psi_p} \pm \frac{b}{2} \sin \boldsymbol{\psi_p}$$

$$\boldsymbol{Y_{p,ij}} = \boldsymbol{Y_{CoG,p}} \pm a_i \sin \boldsymbol{\psi_p} \pm \frac{b}{2} \cos \boldsymbol{\psi_p} \tag{7}$$

where the initial values of $X_{CoG,p}$, $Y_{CoG,p}$, and ψ_p, corresponding to the current vehicle states, are obtained from the vehicle model; k is a generic step along t_h, while N_h is the number of prediction steps; $\boldsymbol{t_s}$, defined within the optimal control problem (OCP) formulation, is the sampling time vector, according to which the control inputs are updated along t_h; and $\boldsymbol{V}$, $\boldsymbol{\beta}$ and $\dot{\boldsymbol{\psi}}$ change along t_h in the case of $\text{NMPC}_{\text{prev}}$, i.e. they are part of the NMPC-predicted state vector $\boldsymbol{x_{pred}}$, hence the notations $\boldsymbol{V_{pred}}$, $\boldsymbol{\beta_{pred}}$ and $\dot{\boldsymbol{\psi}}_{\boldsymbol{pred}}$ in Figure 4, but remain constant – equal to the current values given the absence of a prediction – in the case of $\text{DNN}^{\beta}_{\text{prev,4}}$. The latter assumption implies a usually acceptable approximation, given the relatively short prediction horizon.

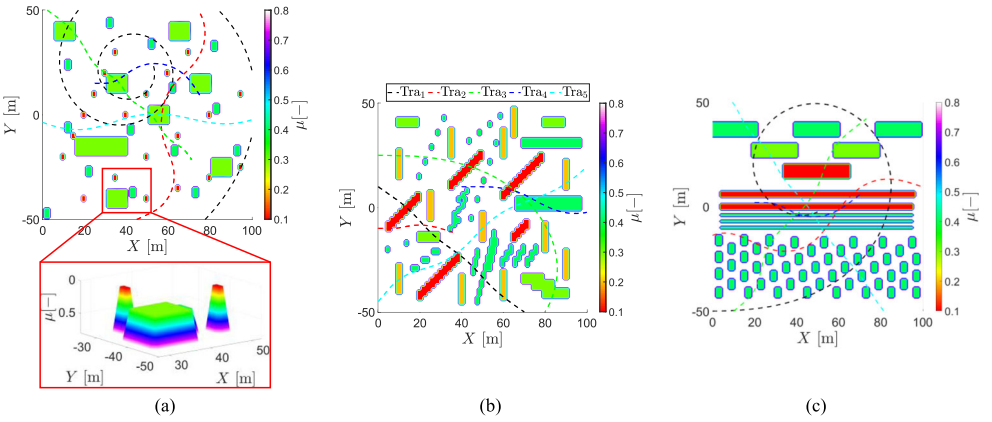


Figure 5. Examples of friction factor maps with a three-dimensional visualisation of the friction variations in a selected region of the first map, and resulting vehicle trajectories ($\text{Tra}_1, \ldots, \text{Tra}_5$) from the training phase.

- The friction coefficient preview block, including the relevant μ-maps, see Figure 5, generates $\boldsymbol{\mu_{ij}}$ as a function of $\boldsymbol{X_{ij}}$ and $\boldsymbol{Y_{ij}}$. Low-friction patches are randomly distributed across the maps, with their sizes and friction values selected to ensure high variability in the scenarios used for DNN training, see Section 3.4. To further diversify the friction values experienced at the four wheels throughout the simulations, each manoeuvre is initiated from different initial positions, with varying heading angles, initial speeds and torque profiles. This setup results in highly diverse training scenarios, as shown by the resulting trajectories in Figure 5, extrapolated from the training dataset.
- The $\text{NMPC}_{\text{prev}}$ and $\text{DNN}^{\beta}_{\text{prev,4}}$ TVC block, i.e. the core of the control architecture, which determines the reference torque distribution. $\text{NMPC}_{\text{prev}}$ computes the four EM torque values based on (i) the reference values of the total powertrain torque and yaw rate; (ii) the preview information on the tyre-road friction factor and steering input; (iii) the preview information on the longitudinal and lateral accelerations; (iv) the online constraint threshold values; and (v) the current vehicle states, together with the vehicle state augmentation represented by the integral of the yaw rate error, $\dot{\psi}_{err,int}$. $\text{DNN}^{\beta}_{\text{prev,4}}$ allocates the torque among the EMs using the inputs from (i), (ii) and (v), as well as the past three-time-step values of sideslip angle (β_1, β_2 and β_3) and yaw rate ($\dot{\psi}_1$, $\dot{\psi}_2$ and $\dot{\psi}_3$), with a 25 ms sampling, with respect to the current one t_0, i.e. $t_1 = t_0 - 0.025$ s, $t_2 = t_0 - 0.050$ s and $t_3 = t_0 - 0.075$ s.

Because the DNN does not require the full set of inputs needed by the NMPC, the following blocks are exclusively associated with the NMPC:

- The online constraint value generator, which defines the constraint thresholds that vary along time, namely: (i) the EM torque saturation limits, $T_{lim,ij}$ (considered constant throughout the prediction), as a function of the corresponding wheel speed ω_{ij}; and (ii) the rear slip angle limit vectors $\boldsymbol{\alpha_{lim,rj}}$, obtained as a function of the corresponding preview-based $\boldsymbol{\mu_{rj}}$, to improve safety. The individual components of $\boldsymbol{\alpha_{lim,rj}}$ are given by

a saturated linear function of the friction factor:

$$\alpha_{lim,rj}(k) = \begin{cases} \alpha_{lim,rj,min}, \mu_{rj}(k) < \mu_{rj,ths1} \\ G_\mu \mu_{rj}(k) + \alpha_{lim,rj,mid}, \mu_{rj,ths1} \leq \mu_{rj}(k) < \mu_{rj,ths2} \\ \alpha_{lim,rj,max}, \mu_{rj}(k) \geq \mu_{rj,ths2} \end{cases} \tag{8}$$

where $\mu_{rj,ths1} = 0.2$ and $\mu_{rj,ths2} = 1$ are the friction factor values that are used to determine the rear slip angle limits; $\alpha_{lim,rj,min} = 1.5$ deg and $\alpha_{lim,rj,max} = 4$ deg are the minimum and maximum threshold values of the constraint; and $G_\mu = 3.125$ deg and $\alpha_{lim,rj,mid} = 0.875$ deg are the parameters defining the constraint for friction factor values between $\mu_{rj,ths1}$ and $\mu_{rj,ths2}$.

- The open-loop prediction model, which is a reduced version of the NMPC prediction model, takes as input $\boldsymbol{x_{pred}}$, $\boldsymbol{\mu_{ij}}$ and $\boldsymbol{\delta_f}$, to estimate the sequences of the future lateral and longitudinal accelerations, $\boldsymbol{a_y}$ and $\boldsymbol{a_x}$, which are used by the controller for the vertical tyre load computation.

The vehicle model blocks are:

- The steering and pedal modules, generating p_a and the steering wheel angle δ_{sw}.
- The four IWM models, see their description in Section 2.2, providing the actual powertrain torque levels to the vehicle body model.
- The vehicle body model in Section 2.2, set to provide the relevant variables to the control architecture.

3.3. Optimal control problem and road preview implementation

The NMPC OCP is formulated as the minimisation of a cost function J, subject to a set of constraints:

$$\underset{U}{\text{argmin}}\ J := J_t + J_s = \frac{1}{2}||z_{N_h} - z_{ref,N_h}||^2_{Q_t} + \frac{1}{2}\sum_{k=0}^{N_h-1}[||z_k - z_{ref,k}||^2_Q + ||u_k||^2_R]$$

s.t.

$$x_0 = x_{in}$$

$$x_{k+1} = f(x_k, u_k, p_k)$$

$$z_k = g(x_k, u_k, p_k)$$

$$\text{(i)} \quad T_{tot,lb} \leq T_{tot} \leq T_{tot,ub}$$

$$\text{(ii)} \quad -T_{lim,ij} \leq T_{ij} \leq T_{lim,ij}$$

$$\text{(iii)} \quad -\sigma_{lim,ij} - \epsilon_\sigma \leq \sigma_{ij} \leq \sigma_{lim,ij} + \epsilon_\sigma$$

$$\text{(iv)} \quad -\alpha_{lim,ij} - \epsilon_{\alpha,i} \leq \alpha_{ij} \leq \alpha_{lim,ij} + \epsilon_{\alpha,i}$$

$$\text{(v)} \quad \epsilon_\sigma \geq 0$$

$$\text{(vi)} \quad \epsilon_{\alpha,f} \geq 0$$

$$\text{(vii)} \quad \epsilon_{\alpha,r} \geq 0 \tag{9}$$

where J consists of the terminal cost J_t and the stage cost contribution J_s; z_k and z_{N_h} are the prediction model outputs at the k-th step and the last time step of t_h; $z_{ref,k}$ and z_{ref,N_h} are the corresponding reference values; U is the optimal sequence of control inputs u_k along t_h, i.e. $U = [u_0, u_1, \ldots, u_k, \ldots, u_{N_h}]$; Q and Q_t are the stage and terminal cost function weight matrices associated with the outputs; and R is the stage cost weight matrix penalising the control effort.

The notation $f(\cdot)$ indicates the prediction model formulation, corresponding to the discretised version of (1)–(4) and the related tyre and load transfer models, while $g(\cdot)$ refers to the output equation, providing z as a function of the states x (having initial condition $x_0 = x_{in}$), control actions u and online parameters p. In particular, the output vector z is:

$$z = [\dot{\psi}_{err},\ T_{tot}] \tag{10}$$

where T_{tot} is the sum of the IWM torque levels and the equivalent yaw rate error, $\dot{\psi}_{err}$, includes an integral term:

$$\dot{\psi}_{err} = \dot{\psi} - \dot{\psi}_{ref} + w\dot{\psi}_{err,int} \tag{11}$$

where w is a constant weight. The corresponding reference output vector, z_{ref}, is:

$$z_{ref} = [0,\ T_{tot,d}] \tag{12}$$

where $T_{tot,d}$ instigates the tracking of the wheel torque demand. At the last step of the prediction horizon, z_{N_h} and z_{ref,N_h} only include the component related to the yaw rate tracking error. x, u and p are defined as:

$$\begin{aligned} x &= [V, \beta, \dot{\psi}, \omega_{fl}, \omega_{fr}, \omega_{rl}, \omega_{rr}, \psi_{err,int}] \\ u &= [T_{fl}, T_{fr}, T_{rl}, T_{rr}, \epsilon_\sigma, \epsilon_{\alpha,f}, \epsilon_{\alpha,r}] \\ p &= [T_{tot,lb}, T_{tot,ub}, T_{lim,fl}, T_{lim,fr}, T_{lim,rl}, T_{lim,rr}, \\ &\quad a_x, a_y, \delta_f, \mu_{fl}, \mu_{fr}, \mu_{rl}, \mu_{rr}, \alpha_{lim,rl}, \alpha_{lim,rr}, \dot{\psi}_{ref}] \end{aligned} \tag{13}$$

where ϵ_σ, $\epsilon_{\alpha,f}$ and $\epsilon_{\alpha,r}$ are the slack variables associated with the vehicle dynamics constraints (iii) and (iv) in (9). Such constraints are implemented with a soft formulation to guarantee smooth numerical operation of the algorithm, also when the vehicle plant is not able to enforce the imposed limitations. $T_{tot,lb}$ and $T_{tot,ub}$ are the lower and upper boundaries on the total powertrain torque, which are adaptively set depending on the sign of $T_{tot,d}$, to limit the correction on the driver demand.

The constraints in (9) are (i) a hard constraint that imposes T_{tot} from the TVC to be smaller than or equal to $T_{tot,d}$; (ii) a hard constraint on the maximum EM torque magnitude on each corner; (iii) soft constraints on the longitudinal tyre slip ratio σ_{ij}, which implement the traction control and anti-lock braking control functions, with the threshold $\sigma_{lim,ij}$ being set to 0.1 in the specific implementation; (iv) soft constraints on the slip angles α_{ij}, where $\alpha_{lim,fj}$ is kept equal to $\sim$12 deg, while the rear limits depend on μ, see (8); and (v)–(vii) impose the slack variables to be non-negative.

The online parameters p are kept constant over the prediction horizon, unless preview information is available. Constant values bring prediction inaccuracies and performance

degradation. In the next analyses, $\text{NMPC}_{\text{base}}$ and $\text{NMPC}_{\text{prev}}$ refer to NMPC configurations with and without preview. In $\text{NMPC}_{\text{prev}}$, the second row of the p components in (13) varies over t_h. $\boldsymbol{t_{ph}}$ is the prediction horizon vector, whose last element is t_h, and which is discretised according to the discretisation step vector $\boldsymbol{t_s}$. Given that the constant p setting implies prediction errors that are magnified by high t_h values, $\text{NMPC}_{\text{base}}$ is implemented with a short t_h of 75 ms, obtained through $\boldsymbol{t_s} = [25,\ 25,\ 25]$ ms. On the other hand, given its higher prediction potential, $\text{NMPC}_{\text{prev}}$ is implemented with $t_h = 200$ ms and variable discretisation according to $\boldsymbol{t_s} = [25,\ 25,\ 25,\ 25,\ 50,\ 50]$ ms. A detailed explanation of the preview functionality of $\text{NMPC}_{\text{prev}}$ can be found in [39].

In this respect, for a baseline setting of the controller cost function weights, Figure 6 analyses the quality of the predictions of the relevant variables during controller operation along a sinusoidal steering simulation from an initial speed of 50 km/h, with a steering wheel angle amplitude of 100 deg applied at a 0.6 Hz frequency. The road surface is the one with the friction profile in Figure 5(c), where the initial EV coordinates are (20,20) m. To highlight the prediction accuracy along t_h, Figure 6(a) compares the actual $\dot{\psi}$ profiles with the controller-predicted profiles at the time instant t_0, corresponding to the bigger marker of each line, e.g. see the inset. Moreover, Figure 6(b) reports key performance indicators (KPIs) of the prediction accuracy level along the entire simulation, with initial and final times T_1 and T_2. The KPIs are the root mean square error values of the predictions, at each prediction step k. The considered variables are the vehicle speed, yaw rate, sideslip angle, and the vertical tyre loads, corresponding to the indicators $V_{pred,\text{RMSE}}(k)$, $\beta_{pred,\text{RMSE}}(k)$, $\dot{\psi}_{pred,\text{RMSE}}(k)$ and $F_{z,pred,\text{RMSE}}(k)$, which are computed through the following routine:

For $k = 1$ to N_h **do**

$$V_{pred,\text{RMSE}}(k) = \sqrt{\frac{1}{T_2 - T_1}\int_{T_1}^{T_2} [V_{pred}(t_0, k) - V(t_0 + t_{ph}(k))]^2 dt}$$

$$\beta_{pred,\text{RMSE}}(k) = \sqrt{\frac{1}{T_2 - T_1}\int_{T_1}^{T_2} [\beta_{pred}(t_0, k) - \beta(t_0 + t_{ph}(k))]^2 dt}$$

$$\dot{\psi}_{pred,\text{RMSE}}(k) = \sqrt{\frac{1}{T_2 - T_1}\int_{T_1}^{T_2} [\dot{\psi}_{pred}(t_0, k) - \dot{\psi}(t_0 + t_{ph}(k))]^2 dt}$$

$$F_{z,pred,\text{RMSE}}(k) = \frac{1}{4}\sum_{i=f,r}\sum_{j=l,r}$$

$$\sqrt{\frac{1}{T_2 - T_1}\int_{T_1}^{T_2} [F_{z,ij,pred}(t_0, k) - F_{z,ij}(t_0 + t_{ph}(k))]^2 dt}$$

End (14)

where $V_{pred}(t_0, k)$, $\beta_{pred}(t_0, k)$, $\dot{\psi}_{pred}(t_0, k)$, and $F_{z,ij,pred}(t_0, k)$ are the NMPC-predicted values of the variables, according to the computation carried out at the current time t_0 for the prediction step k; and $V(t_0 + t_{ph}(k))$, $\beta(t_0 + t_{ph}(k))$, $\dot{\psi}(t_0 + t_{ph}(k))$, and $F_{z,ij}(t_0 + t_{ph}(k))$ are the corresponding actual values, taken from the simulation model results at the time $t_0 + t_{ph}(k)$, to match the time of the prediction. The dependency on k through t_{ph} enables

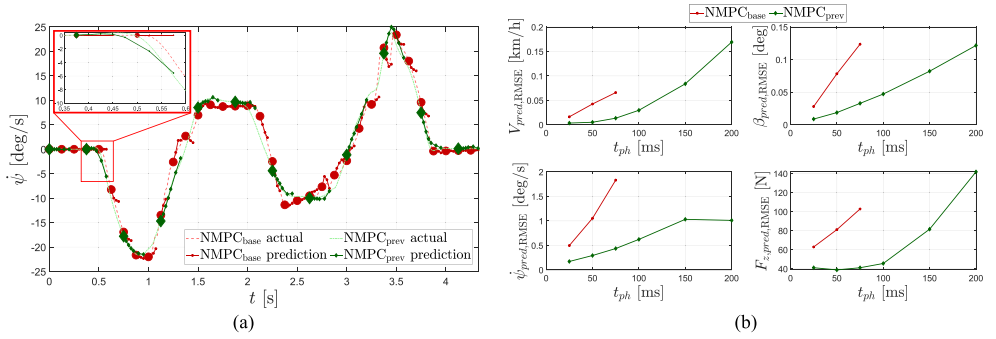


Figure 6. Benefit of road preview on prediction accuracy: (a) yaw rate profiles in the time domain; and (b) considered KPIs as a function of $t_{ph}(k)$ during a sinusoidal steering manoeuvre on a variable friction surface.

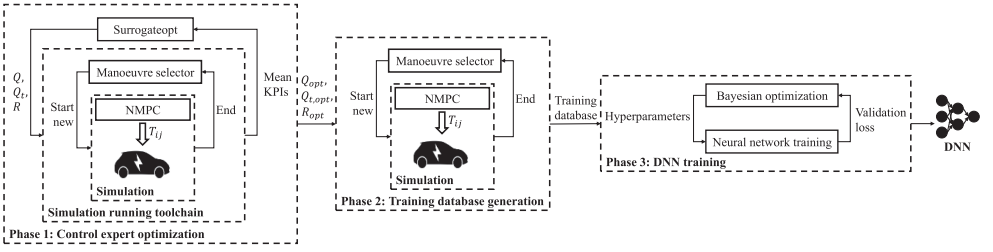


Figure 7. Simplified schematic of the imitation learning process based on Phases 1–3.

to assess how much the prediction accuracy deteriorates along the horizon. Interestingly, thanks to the preview setup, $\text{NMPC}_{\text{prev}}$ provides an overall better prediction at 150 ms than $\text{NMPC}_{\text{base}}$ at 75 ms.

3.4. Imitation learning-based control design

The IL process to obtain a DNN for TVC is represented in Figure 7 and consists of three phases.

Phase 1: Control expert optimisation. In this phase, $\text{NMPC}_{\text{prev}}$ is optimised over $n_{test} = 200$ representative tests, along with 5 manoeuvre typologies, corresponding to the lines of Table 2. Each manoeuvre was simulated 40 times: (i) 20 times on a surface with $\mu = 0.8$, among which 10 times with constant – but different for each simulation – $T_{tot,d}$ values and 10 times with a variable $T_{tot,d}$ profiles and (ii) 20 times on a surface with variable μ from 0.1 to 0.8, according to the maps in Figure 5, with the same distribution between constant and variable $T_{tot,d}$ profiles as for (i). For each simulation run, where relevant for the specific manoeuvre, the following parameters were varied in the range indicated in the table: (a) $|\delta_{sw,max}|$, i.e. the δ_{sw} amplitude; (b) $\dot{\delta}_{sw}$, i.e. the steering wheel rate; (c) $f_{\delta_{sw}}$, i.e. the steering frequency; and (d) $T_{\delta_{sw,const}}$, i.e. the duration of the constant steering phases. To achieve a database with good coverage of the operating conditions, the tests have been simulated from different initial speeds, in the 20–60 km/h range.

Table 2. Considered manoeuvre parameters.

Manoeuvre	$\lvert\delta_{sw,max}\rvert$ [deg]	$\dot{\delta}_{sw}$[deg/s]	$f_{\delta_{sw}}$[Hz]	$T_{\delta_{sw,const}}$[s]
Ramp steer	200	2–20	–	–
Sinusoidal steering	100–200	–	0.1–0.6	–
Sweep steering	100–200	–	0–2	–
Multiple-step steer	100–200	200–400	–	0.25–2

The MATLAB 'SurrogateOpt' function [40] for global minimisation of time-consuming objective functions is used to find the optimal NMPC cost function weights, according to the following set-up:

$$\begin{aligned}
&\underset{Q,Q_t,R}{\operatorname{argmin}}\ J_{WT}: \\
&= w_{WT1}\frac{\sum_{l_{test}=1}^{n_{test}} \dot{\psi}_{err,\mathrm{RMS},l_{test}}}{n_{test}\ \dot{\psi}_{err,\mathrm{RMS,max}}} + w_{WT2}\frac{\sum_{l_{test}=1}^{n_{test}} \dot{\psi}_{err,\mathrm{ISD},l_{test}}}{n_{test}\ \dot{\psi}_{err,\mathrm{ISD,max}}} \\
&+ w_{WT3}\frac{\sum_{l_{test}=1}^{n_{test}} T_{tot,\mathrm{RMSE},l_{test}}}{n_{test}\ T_{tot,\mathrm{RMSE,max}}} + \\
&+ w_{WT4}\frac{\sum_{l_{test}=1}^{n_{test}} \Delta\alpha_{r,\mathrm{RMS},l_{test}}}{n_{test}\ \Delta\alpha_{r,\mathrm{RMS,max}}} + w_{WT5}\frac{\sum_{l_{test}=1}^{n_{test}} \Delta\alpha_{r,\mathrm{max},l_{test}}}{n_{test}\ \Delta\alpha_{r,\mathrm{max,max}}} \\
&\text{s.t.} \\
&Q, Q_t \in [Q_{\min}, Q_{\max}] \\
&R \in [R_{\min}, R_{\max}]
\end{aligned} \tag{15}$$

where l_{test} is an index referring to the considered test, and the auxiliary cost function J_{WT} for controller calibration, computed outside the NMPC algorithm along the 200 simulations, is based on:

- $\dot{\psi}_{err,\mathrm{RMS}}$, i.e. the root mean square value of the error between $\dot{\psi}_{ref}$ and $\dot{\psi}$:

$$\dot{\psi}_{err,\mathrm{RMS}} = \sqrt{\frac{1}{T_2 - T_1}\int_{T_1}^{T_2} [\dot{\psi} - \dot{\psi}_{ref}]^2 dt} \tag{16}$$

- $\dot{\psi}_{err,\mathrm{ISD}}$, i.e. the time integral of the squared time derivative of the yaw rate error, to reduce control input fluctuations:

$$\dot{\psi}_{err,\mathrm{ISD}} = \int_{T_1}^{T_2} \left[\ddot{\psi} - \ddot{\psi}_{ref}\right]^2 dt \tag{17}$$

- $T_{tot,\mathrm{RMSE}}$, i.e. the root mean square error value between T_{tot} and $T_{tot,d}$:

$$T_{tot,\mathrm{RMSE}} = \sqrt{\frac{1}{T_2 - T_1}\int_{T_1}^{T_2} [T_{tot} - T_{tot,d}]^2 dt} \tag{18}$$

- $\Delta\alpha_{r,\mathrm{RMS}}$, i.e. the root mean square value of the rear slip angle limit violation $\Delta\alpha_r$:

$$\Delta\alpha_{r,\mathrm{RMS}} = \sqrt{\frac{1}{T_2 - T_1}\int_{T_1}^{T_2} [\Delta\alpha_r]^2 dt} \tag{19}$$

  with $\Delta\alpha_r$ being calculated as the average between the left and right violations:

$$\Delta\alpha_r = \max\left(0, \frac{\sum_{j=l,r} |\alpha_{rj}| - \alpha_{lim,rj}}{2}\right) \tag{20}$$

- $\Delta\alpha_{r,\max}$, i.e. the maximum value of $\Delta\alpha_r$ along the manoeuvre:

$$\Delta\alpha_{r,\max} = \max(\Delta\alpha_r) \tag{21}$$

In (15), the maximum expected values of the KPIs, indicated with the subscript 'max', are used to normalise the individual terms of J_{WT}.

Phase 2: Training database generation. Following its optimisation, the $\mathrm{NMPC_{prev}}$ calibration with the optimal weight matrices (Q_{opt}, $Q_{t,opt}$, and R_{opt}) is tested to collect its inputs and outputs. In addition to the previously introduced manoeuvres with the continuously running controller, half of the simulations are repeated with a specific controller activation logic. In fact, $\mathrm{NMPC_{prev}}$ is switched on and off with a predefined frequency over the same manoeuvre. In this way, the controller tries to bring the vehicle back to a stable condition, starting from the critical ones that tend to occur when the controller is inactive. Hence, the DNN can learn how to act also when the vehicle shows unstable or undesirable behaviour. To ensure smooth TVC operation during normal driving, 32 additional straight-line manoeuvres with different combinations of constants $T_{tot,d}$ and μ are included in this phase. As a result, the collected training database consists of $\sim$152,000 sampling data points, obtained through 332 simulations. The database is split into training and validation sets with a 90:10 ratio, and normalised with respect to the physically plausible limits of the operating conditions of the vehicle, e.g. with the yaw rate limited to $\pm$90 deg/s and the angular wheel speeds ranging between 0 and 1800 rpm.

Phase 3: DNN training. During the feedforward DNN training process, a Bayesian optimisation algorithm [41] is used to select the best hyperparameter sets within the ranges in Table 3. The hyperparameters include the initial learning rate λ, mini-batch size m_{batch}, number of hidden layers n_{hid}, number of neurons per layer n_{neu} and the output layer activation function f_o. For the sake of time saving, each training run is limited to 100 epochs. The adaptive moment estimation (Adam) optimiser [42] – with a gradient decay factor of 0.9 and a squared gradient decay factor of 0.999 – is employed to update the network parameters by minimising a loss function defined as the mean squared prediction error. The parameter set achieving the minimum validation loss across all epochs – which does not necessarily correspond to the last iteration – is saved. Furthermore, L_2 regularisation is applied across the training runs with a fixed value of $1{\cdot}10^{-4}$. The Bayesian optimisation procedure is executed for 168 h, corresponding to approximately 230 trainings for the $\mathrm{DNN}^{\beta}_{\mathrm{prev},4}$ and $\mathrm{DNN}^{\beta\backslash}_{\mathrm{prev},4}$ configurations (the most complex networks of this study), on a Lenovo 16p Gen 4 ThinkBook with Intel Core i9-13900H 2.60 GHz processor, 8 GB GPU and 32 GB 5.6 GHz RAM.

Table 3. Hyperparameters of the DNN training process.

Description	Symbol	Values
Initial learning rate	λ	$5{\cdot}10^{-4} \div 1{\cdot}10^{-2}$
Mini-batch size	m_{batch}	64, 128, 258, 512
Number of the hidden layers	n_{hid}	2, 3, 4
Number of neurons per layer	n_{neu}	32, 64, 128, 258
Output function	f_o	linear, tanh

Upon completion of the Bayesian optimisation process, the trained networks are firstly evaluated in open-loop (i.e. in parallel with $\text{NMPC}_{\text{prev}}$, which sends the control inputs to the vehicle model) within the simulation architecture in Figure 4, implemented in Simulink, under several previously unseen scenarios. These scenarios involve unexplored combinations of steering and pedal profiles as well as different initial vehicle conditions, thereby driving the vehicle through friction paths not encountered during training and operating under unseen driving conditions. If this evaluation is successful, the networks are subsequently tested in a closed-loop by replacing $\text{NMPC}_{\text{prev}}$. If any of these validation stages fail, the Bayesian optimisation process is repeated, by incorporating additional scenarios to explore unseen DNN inputs-outputs relations, expanding the hyperparameter ranges, or modifying the network architecture. For instance, in the implementation phase of the specific NNs of this study, during the initial trials, when the training dataset consisted solely of the manoeuvres introduced in Phase 1, the DNNs were unable to complete the final closed-loop validation. Their performance was noteably improved by including data from the manoeuvres introduced in Phase 2, in which the NMPC is repeatedly switched on and off.

The toolchain based on Phase 1–Phase 3 was used to generate several DNNs. All of them have in common the architecture with fully connected layers, the ReLU [43] activation functions of the hidden layers and the linear activation function of the output layer. In the remainder, the trained networks are referred to as follows:

- $\text{DNN}^{\beta}_{\text{prev},4}$, already introduced in Section 3.2, targeting the simulated AWD EV configuration, where the subscript '4' identifies the number of control inputs, i.e. the reference IWM torque levels. The resulting network has the architecture in Figure 8, with 4 hidden layers of 128 neurones each. The $\text{DNN}^{\beta}_{\text{prev},4}$ input vector, $x_{0,\text{DNN}^{\beta}_{\text{prev},4}}$, consists of 56 components:

$$\begin{aligned} x_{0,\text{DNN}^{\beta}_{\text{prev},4}} = [&V, \beta, \beta_1, \beta_2, \ \beta_3, \dot{\psi}, \dot{\psi}_1, \dot{\psi}_2, \dot{\psi}_3, \omega_{fr}, \omega_{fl}, \omega_{rr}, \omega_{rl}, T_{tot,d}, \\ &\delta_f(1), \ldots, \delta_f(7), \dot{\psi}_{ref}(1), \ldots, \dot{\psi}_{ref}(7), \mu_{fr}(1), \ldots, \mu_{fr}(7), \\ &\mu_{fl}(1), \ldots, \mu_{fl}(7), \mu_{rr}(1), \ldots, \mu_{rr}(7), \mu_{rl}(1), \ldots, \mu_{rl}(7)] \end{aligned} \tag{22}$$

  where the front steering and friction preview vectors, $\boldsymbol{\delta_f}$ and $\boldsymbol{\mu_{ij}}$, are sampled according to the sampling time vector $\boldsymbol{t_s} = [25,\ 25,\ 25,\ 25,\ 50,\ 50]$ ms. The $\boldsymbol{\mu_{ij}}$ vectors are computed from the predicted future wheel positions as in (7), under the assumption of constant V, β and $\dot{\psi}$ over the preview horizon.

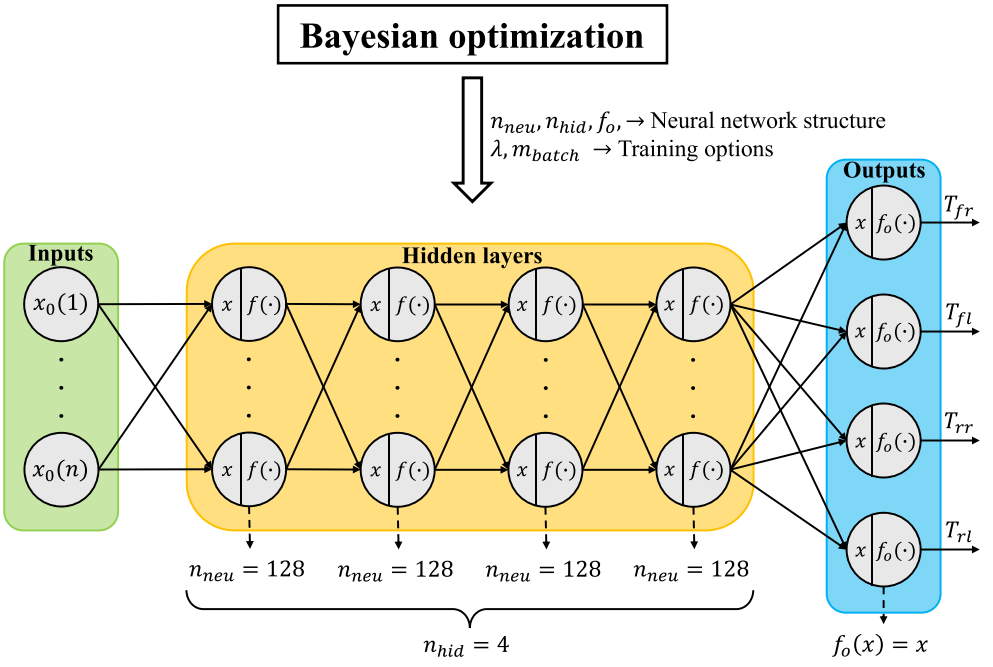


Figure 8. Simplified schematic of the $\text{DNN}^{\beta\backslash}_{\text{prev},4}$ and $\text{DNN}^{\beta}_{\text{prev},4}$ configurations.

- $\text{DNN}^{\beta\backslash}_{\text{prev},4}$, which has the same characteristics as $\text{DNN}^{\beta}_{\text{prev},4}$, apart from the different number of inputs (52 instead of 56), since it does not require the sideslip angle information (β, β_1, β_2 and β_3). Accordingly, $\boldsymbol{\mu_{ij}}$ are computed with the same method as for $\text{DNN}^{\beta}_{\text{prev},4}$, but assuming $\beta = 0$.
- $\text{DNN}^{\beta}_{\text{prev},2}$, which is the equivalent of $\text{DNN}^{\beta}_{\text{prev},4}$ for the experimental RWD EV configuration, i.e. with only two control inputs. The network consists of 3 hidden layers of 64 neurons each. With respect to $x_{0,\text{DNN}^{\beta}_{\text{prev},4}}$, the $\text{DNN}^{\beta}_{\text{prev},2}$ input vector excludes ω_{fr} and ω_{fl}.
- $\text{DNN}^{\beta\backslash}_{\text{prev},2}$, which is the same as $\text{DNN}^{\beta}_{\text{prev},2}$, apart from the exclusion of the sideslip angle information from the inputs.
- $\text{DNN}^{\beta\backslash}_{\text{prev},\mu\backslash,2}$, which only uses the preview information on $\boldsymbol{\delta_f}$, but not on $\boldsymbol{\mu_{ij}}$, i.e. $\mu_{ij}(2), \dots, \mu_{ij}(7)$ are all equal to the current value $\mu_{ij}(1)$.
- $\text{DNN}^{\beta\backslash}_{2}$, which does not have any preview information, i.e. neither on $\boldsymbol{\delta_f}$ nor on $\boldsymbol{\mu_{ij}}$.

For the $\text{DNN}^{\beta\cdot}_{\cdot,2}$ configurations, the whole IL process was simplified with respect to the one of $\text{DNN}^{\beta\cdot}_{\cdot,4}$, i.e. the simulated manoeuvres were set by considering the experimental safety constraints, involving a maximum EV speed of 40 km/h and the smaller proving ground dimensions.

3.5. Controller implementation

The NMPCs are implicitly implemented in the MATLAB Simulink environment through the ACADO toolkit [44], which supports autogenerated C code and real-time operation. The solution of the nonlinear OCP is computed by the qpOASES 3 solver. The prediction model is discretised with a 1 ms time step. The DNNs are coded in MATLAB functions that include only the neural network structure equations. For both the NMPCs and DNNs, the control action sampling time, t_c, is set to 25 ms.

4. Simulation evaluation

This section presents the simulation results, obtained with the AWD vehicle model of Section 2.2, to assess the performance, computational effort and robustness of $\text{DNN}^{\beta\setminus}_{\text{prev},4}$ and $\text{DNN}^{\beta}_{\text{prev},4}$, including comparisons with the corresponding NMPCs and the passive (i.e. without TV) case.

4.1. Performance analysis in the time and frequency domains

Figure 9 reports the time profiles of the relevant variables along a sinusoidal steering test, with the steering input applied at 0.6 Hz from the following initial conditions: (i) 40 km/h of vehicle speed; (ii) (0,2) m coordinates in Figure 5(a) scenario; and (iii) and zero yaw angle. A variable wheel torque demand is applied, transitioning from full acceleration to maximum regenerative braking after 1 s, then returning to full traction after 2 s. The combination of negative wheel torque and variable friction conditions brings the passive EV to the verge of instability, with a sideslip angle peak in excess of 30 deg. Here, the reference direct yaw moment $M_{z,ref}$ is calculated as follows:

$$M_{z,ref} = \frac{a}{2R_w}\cos(\delta_f)[T_{fr} - T_{fl}] + \frac{b}{2R_w}[T_{rr} - T_{rl}] \tag{23}$$

The associated KPIs are shown in Table 4. With respect to the passive case, the controlled configurations reduce $\dot{\psi}_{err,\text{RMS}}$ from 55% to 84%, depending on the considered algorithm. Interestingly, although making the cornering response prompter than for the passive configuration, $\text{NMPC}_{\text{base}}$ experiences a time delay in tracking the reference yaw rate with respect to the preview-based algorithms, e.g. see the inset of Figure 9(a). Moreover, with respect to $\text{NMPC}_{\text{base}}$, the preview controllers reduce $\dot{\psi}_{err,\text{RMS}}$, $\Delta\alpha_{r,\text{RMS}}$ and $\Delta\alpha_{r,\text{max}}$, respectively by 61-to-65%, 36-to-39% and 28-to-39%. The cornering response improvement of the preview algorithms is associated with a marginal $T_{tot,\text{RMSE}}$ increase. The important conclusion is that even if $\text{NMPC}_{\text{prev}}$ provides the best results, the DNN algorithms are very close, i.e. the response difference is negligible and clearly superior to that of $\text{NMPC}_{\text{base}}$. Moreover, the integrated traction control and anti-lock braking functionality – implemented internally in the NMPC formulation via (9iii) and (9v) – is successfully learned by the DNN configurations, which maintain tyre slip ratios within the ± 0.1 limit, whereas the passive vehicle experiences wheel spinning or locking throughout the simulation.

The computational performance of the algorithms, see Figure 9(b), was tested on a dSPACE MicroAutobox III unit, in terms of turnaround time (TAT), i.e. the time required

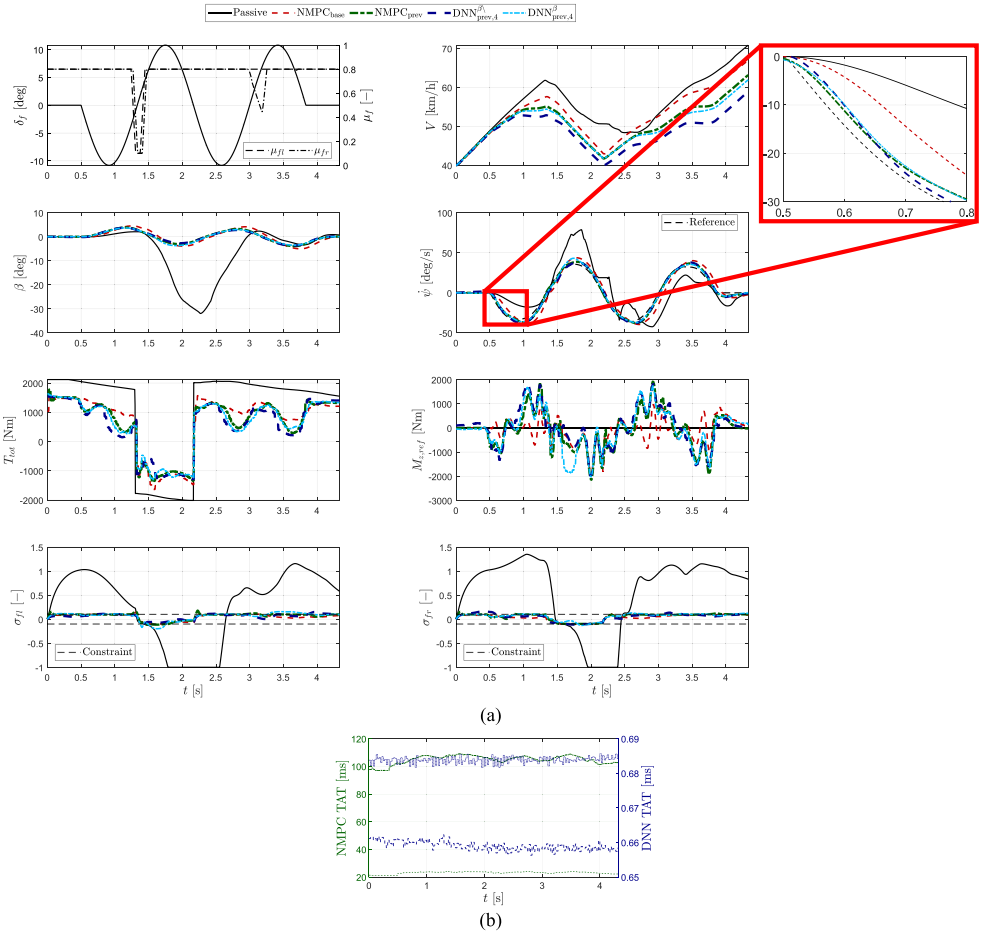


Figure 9. Controller comparison along the sinusoidal steering manoeuvre on a variable friction surface. Time domain results in terms of: (a) relevant vehicle dynamics variables, with the reference yaw rate being one of $\text{NMPC}_{\text{prev}}$; and (b) turnaround times.

Table 4. Key performance indicators along the selected sinusoidal steering manoeuvre with variable friction conditions.

Configuration	$\dot{\psi}_{err,\text{RMS}}$[deg/s]	$\Delta\alpha_{r,\text{RMS}}$[deg]	$\Delta\alpha_{r,\text{max}}$[deg]	$T_{tot,\text{RMSE}}$[Nm]
Passive	$1.77{\cdot}10^{1}$	$1.02{\cdot}10^{1}$	$2.94{\cdot}10^{1}$	$4.52{\cdot}10^{2}$
$\text{NMPC}_{\text{base}}$	7.96	1.31	3.20	$8.46{\cdot}10^{2}$
$\text{NMPC}_{\text{prev}}$	2.82	$8.13{\cdot}10^{-1}$	1.95	$1.03{\cdot}10^{3}$
$\text{DNN}^{\beta\backslash}_{\text{prev},4}$	3.14	$8.01{\cdot}10^{-1}$	1.95	$1.19{\cdot}10^{3}$
$\text{DNN}^{\beta}_{\text{prev},4}$	3.00	$8.43{\cdot}10^{-1}$	2.29	$1.10{\cdot}10^{3}$

by the considered rapid control prototyping hardware to compute the control input. The DNN configurations achieve very low TAT values, amounting to $\sim$0.6 ms, i.e. they are much faster than the NMPCs. While $\text{NMPC}_{\text{base}}$ is still real-time implementable, with a TAT of $\sim$24 ms, the TAT of $\text{NMPC}_{\text{prev}}$ is approximately four times longer than the

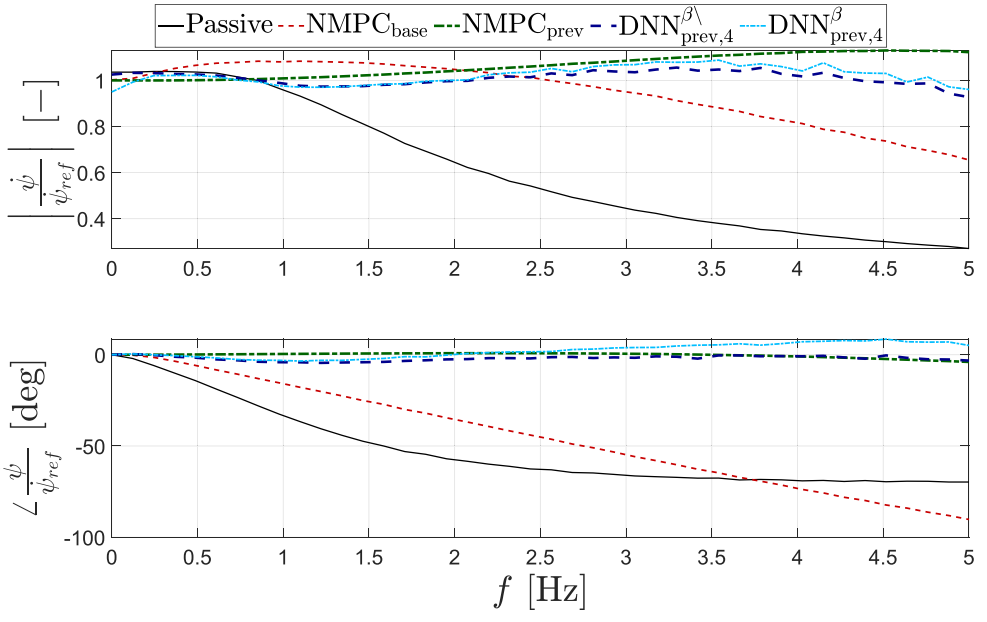


Figure 10. Frequency response function of the yaw rate tracking performance during a sweep steering manoeuvre on a high-friction surface.

implementation time, i.e. 25 ms, and ∼150 times longer than the TAT of the DNN configurations, which highlights the DNN merit.

Figure 10 reports the frequency response characteristics of the yaw rate tracking performance, expressed by the magnitude and phase of $\dot{\psi}/\dot{\psi}_{ref}$, during a sweep steering manoeuvre with 20 deg δ_{sw} amplitude, 60 km/h initial EV speed and zero $T_{tot,d}$, simulated in high tyre-road friction conditions. Desirable performance implies tracking gain and phase angle values, respectively close to 1 and 0 for the relevant frequency range. In this scenario, the bandwidth was defined as the lowest frequency at which the yaw rate tracking gain stays below 0.9, which amounts to 1.2 Hz for the passive EV. NMPC$_{\text{base}}$ significantly improves the performance by extending the bandwidth to 3.4 Hz, while the preview configurations further increase it to ∼5 Hz and beyond. The preview benefit is even more noticeable on the phase delay, which is negligible across the relevant frequency range for all the preview-based setups, i.e. including the DNN cases, while the passive EV and NMPC$_{\text{base}}$ have a phase delay of 50 deg at ∼1.5 and ∼2.75 Hz.

The previous analyses highlighted that the IL approach is able to provide similar performance to a preview-based NMPC algorithm, also without the need for the estimated sideslip angle as a feedback input. This would represent a major implementation benefit for the industry, given the complexity and challenges associated with accurate β estimation. Hence, the next simulation analyses will focus on DNN$^{\beta\backslash}_{\text{prev},4}$.

4.2. Robustness analysis

The robustness of the DNN and NMPC algorithms is evaluated through > 500 Monte Carlo simulations [45], considering the performance sensitivity to the vehicle and preview

parameters. In fact, parametrisation errors affect the prediction accuracy of $\text{NMPC}_{\text{base}}$ and $\text{NMPC}_{\text{prev}}$, as well as the performance of $\text{DNN}^{\beta\backslash}_{\text{prev},4}$, which – in the implemented toolchain – was trained in the absence of inaccuracies.

The considered parameter error distribution is reported in Figure 11. Specifically, $\text{NMPC}_{\text{prev}}$ and $\text{DNN}^{\beta\backslash}_{\text{prev},4}$ are subject to the errors corresponding to the plots in Figure 11(a,b), while $\text{NMPC}_{\text{base}}$ is subject to those in Figure 11(a,c). m_{add} is the additional mass included in the EV model. k_x and k_y are the scaling factors on the longitudinal slip stiffness and cornering stiffness of the tyres, respectively. ϵ_δ is the percentage error applied to the steering angle preview at the last prediction step, which is used to generate a linear steering error distribution along t_h:

$$\begin{aligned} &\textbf{For } k = 0 \text{ to } N_h \textbf{ do} \\ &\delta_f(k) = \left[1 + \frac{\epsilon_\delta}{100}\frac{k}{N_h}\right]\delta_{f,id}(k) \\ &\textbf{End} \end{aligned} \tag{24}$$

where $\delta_{f,id}(k)$ is the ideal value (i.e. without preview errors) of the k-th component of the front steering angle preview vector, and $\delta_f(k)$ is the corresponding actual value that is provided to the control architecture, including the preview error. k_μ is a scaling coefficient on the tyre-road friction factors that are provided as road preview to the algorithms. The $\boldsymbol{\mu_{ij}}$ preview is also affected by the vehicle localisation accuracy, since it depends on the EV coordinates on the road maps in Figure 5. In this respect, $s_{err,max}$ in Figure 11 represents the maximum position preview error, which is randomly picked at the beginning of each simulation. Then, during the individual manoeuvre, the preview position errors, d_X and d_Y, randomly vary according to a uniform probability law U:

$$\begin{aligned} X_{CoG,p}(0) &= X_{CoG,id,0} + d_X, \text{ with } \; d_X \in U(-s_{err,max}, s_{err,max}) \\ Y_{CoG,p}(0) &= Y_{CoG,id,0} + d_Y, \text{ with } \; d_Y \in U(-s_{err,max}, s_{err,max}) \end{aligned} \tag{25}$$

where $X_{CoG,id,0}$ and $Y_{CoG,id,0}$ are the EV coordinates at the current time, before the addition of the error. The error is only applied to the TVC inputs, and not to the simulation model, and affects the position preview values through (7). In the absence of preview information from V2X or road preview sensors, tyre-road friction estimators are necessary. To simulate the friction level estimation delay, in $\text{NMPC}_{\text{base}}$ a first-order transfer function with a time constant τ_μ is considered, with the distribution in Figure 11.

Figure 12 shows the Monte Carlo results during a multiple-step steer from 40 km/h, where each constant steering angle phase following the ± 120 deg δ_{sw} step application lasts 0.5 s. The initial conditions are (50,50) m in the map of Figure 5(c), with $\psi = 90$ deg. A variable $T_{tot,d}$ profile is implemented through a tip-in (positive torque demand step) followed by a tip-out (negative torque demand step). Clear benefits are brought by the preview-based algorithms, with $\text{NMPC}_{\text{prev}}$, providing average values of $\dot{\psi}_{err,\text{RMS}}$ and $\Delta\alpha_{r,\text{RMS}}$ amounting to 2.01 deg/s and 1.30 deg, and $\text{DNN}^{\beta\backslash}_{\text{prev},4}$ with 2.76 deg/s and 1.46 deg, with respect to 5.66 deg/s and 1.89 deg of $\text{NMPC}_{\text{base}}$. The $\dot{\psi}_{err,\text{RMS}}$ range is the highest for $\text{DNN}^{\beta\backslash}_{\text{prev},4}$, with 2.63 deg/s against 1.54 and 1.36 deg/s of $\text{NMPC}_{\text{base}}$ and $\text{NMPC}_{\text{prev}}$. However, (i) the highest $\dot{\psi}_{err,\text{RMS}}$ range of $\text{DNN}^{\beta\backslash}_{\text{prev},4}$ is caused by only 5% of the entire test set (in particular,

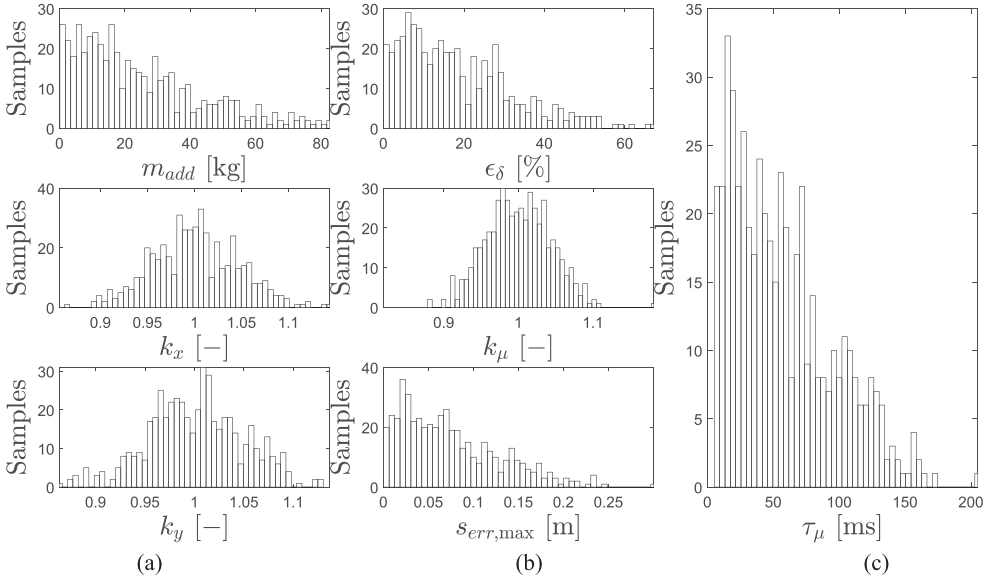


Figure 11. Parameter distribution of the Monte Carlo analysis.

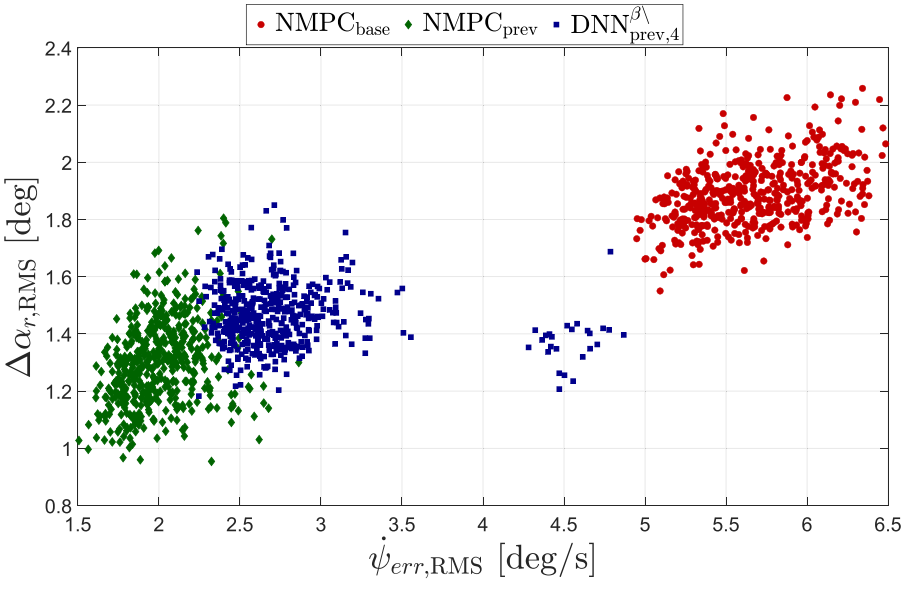


Figure 12. KPI distribution for the Monte Carlo analysis along the multiple-step steer on a variable friction surface.

by the parametrisations with the highest values of k_x), and (ii) when considering the same error combination of the common parameters for DNN$^{\beta\setminus}_{\text{prev},4}$ and NMPC$_{\text{base}}$, the former always provides significantly better results. On the contrary, there is no significant difference among the configurations in terms of $\Delta\alpha_{r,\text{RMS}}$ range, with an average value of $\sim$0.7 deg.

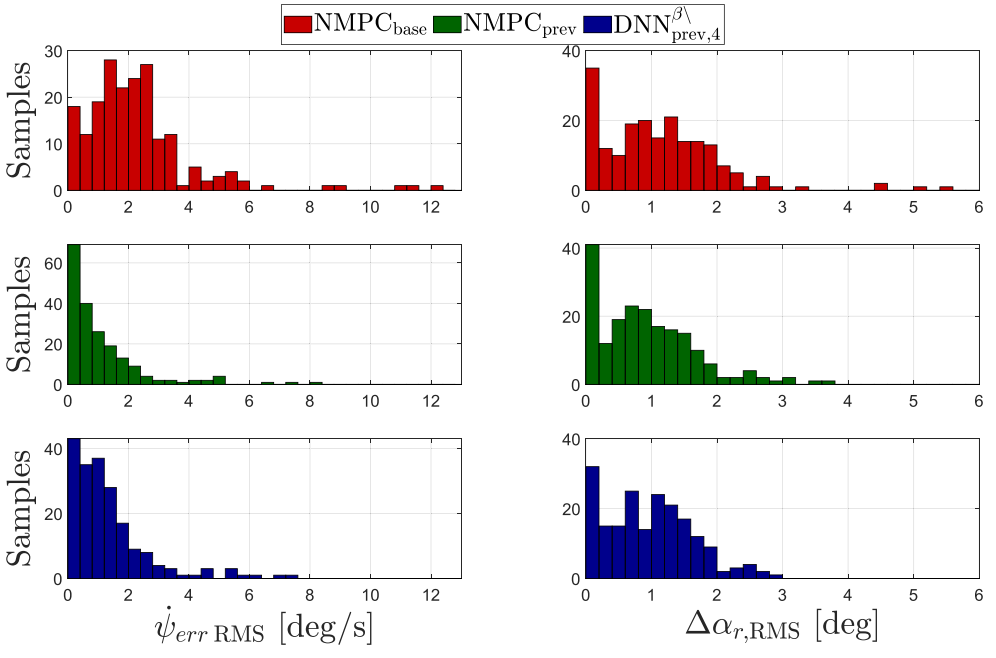


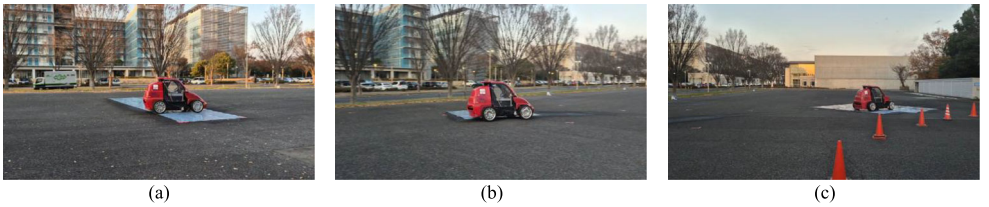
Figure 13. KPI distribution over the considered 200 manoeuvres.

To further analyse robustness, Figure 13 collects the KPIs across the 200 manoeuvres of Phase 1 in Section 3.4. The best performance is obtained by NMPC$_{\text{prev}}$, with average values of 1.11 deg/s and 0.93 deg for $\dot{\psi}_{err,\text{RMS}}$ and $\Delta\alpha_{r,\text{RMS}}$, closely followed by DNN$^{\beta\backslash}_{\text{prev},4}$ with 1.35 deg/s and 0.97 deg, while NMPC$_{\text{base}}$ only achieves 2.27 deg/s and 1.13 deg. Hence, on average, DNN$^{\beta\backslash}_{\text{prev},4}$ improves $\dot{\psi}_{err,\text{RMS}}$ by $> 40\%$ with respect to NMPC$_{\text{base}}$.

5. Experimental evaluation

5.1. Proof-of-concept experimental setting

Through vehicle experiments along two manoeuvres – a modified version of the sine-with-dwell test and a double step steer – involving highly variable tyre-road friction conditions, this section analyses: (i) the performance and robustness of DNN-based TVC, and in particular of DNN$^{\beta\backslash}_{2}$, DNN$^{\beta\backslash}_{\text{prev},\mu\backslash,2}$, DNN$^{\beta\backslash}_{\text{prev},2}$ and DNN$^{\beta}_{\text{prev},2}$; and (ii) the benefits of including road preview information. For safety reasons, the maximum motor torque is limited to 230 Nm. The low-friction areas, shown in Figure 14, are characterised by ultra-high-molecular-weight polyethylene panels, which were covered by water to achieve a friction coefficient of ~ 0.2. Before testing the controlled vehicle, the low-friction levels were inferred from the experimental response of the passive vehicle, based on the observation of the longitudinal acceleration profiles of the vehicle as well as the rotational wheel dynamics during wheel spinning and locking events, while considering the vertical tyre load on the relevant axles. The same tyre-road friction factor value for the considered vehicle operating on identical panels had already been obtained in previous studies on



(a) (b) (c)

Figure 14. Most relevant sections – with the low-friction patches – of the experimental testing environment during: (a) the modified sine-with-dwell test and (b)–(c) the double step steer.

tyre-road friction estimation [46]. For improving repeatability, the desired steering profile is given as a reference to the power steering module and is also used to generate the steering preview information. Due to the low accuracy and slow sampling rate of the involved GPS, in the experiments, the tyre-road friction factor profiles at the front wheels – instead of being evaluated through the blocks in Figure 4 – are provided to the controllers through an offline-determined time-based lookup table. The friction profile for the rear corners is considered to be the same as one of the front corners, with a time shift $\Delta t = [a_f + a_r]/V(0)$. Without $\boldsymbol{\mu_{ij}}$ preview, i.e. for $\mathrm{DNN}_2^{\beta\backslash}$ and $\mathrm{DNN}_{\mathrm{prev},\mu\backslash,2}^{\beta\backslash}$, a friction estimator should be implemented. Hence, for these configurations, first-order dynamics with a 100 ms time constant are applied to the lookup table-generated μ time history to simulate an estimation delay.

During the experiments, the adapted version of the yaw moment observer TVC (YMO-TVC) architecture from [47], which had already been comprehensively tested and tuned on the specific EV demonstrator before this study, and integrated with rule-based traction control and anti-lock braking system algorithms, is used as a benchmark.

For a comprehensive evaluation, further KPIs are introduced:

- $\dot{\psi}_{err,\max}$, i.e. the maximum yaw rate error magnitude:

$$\dot{\psi}_{err,\max} = \max(|\dot{\psi} - \dot{\psi}_{ref}|) \tag{26}$$

- $\beta_{\max}$, i.e. the maximum sideslip angle magnitude:

$$\beta_{\max} = \max(|\beta|) \tag{27}$$

- $\sigma_{r,\max}$, i.e. the maximum slip ratio magnitude at the rear wheels:

$$\sigma_{r,\max} = \max(|\sigma_{rr}|, |\sigma_{rl}|) \tag{28}$$

- $M_{z,ref,\mathrm{RMS}}$, i.e. the root mean square value of the reference direct yaw moment, $M_{z,ref}$, corresponding to the reference rear wheel torque level distribution:

$$M_{z,ref,\mathrm{RMS}} = \sqrt{\frac{1}{T_2 - T_1}\int_{T_1}^{T_2} [M_{z,ref}]^2 dt} \tag{29}$$

where $M_{z,ref}$ considers only the contribution provided by the rear motor torque levels included in (23).

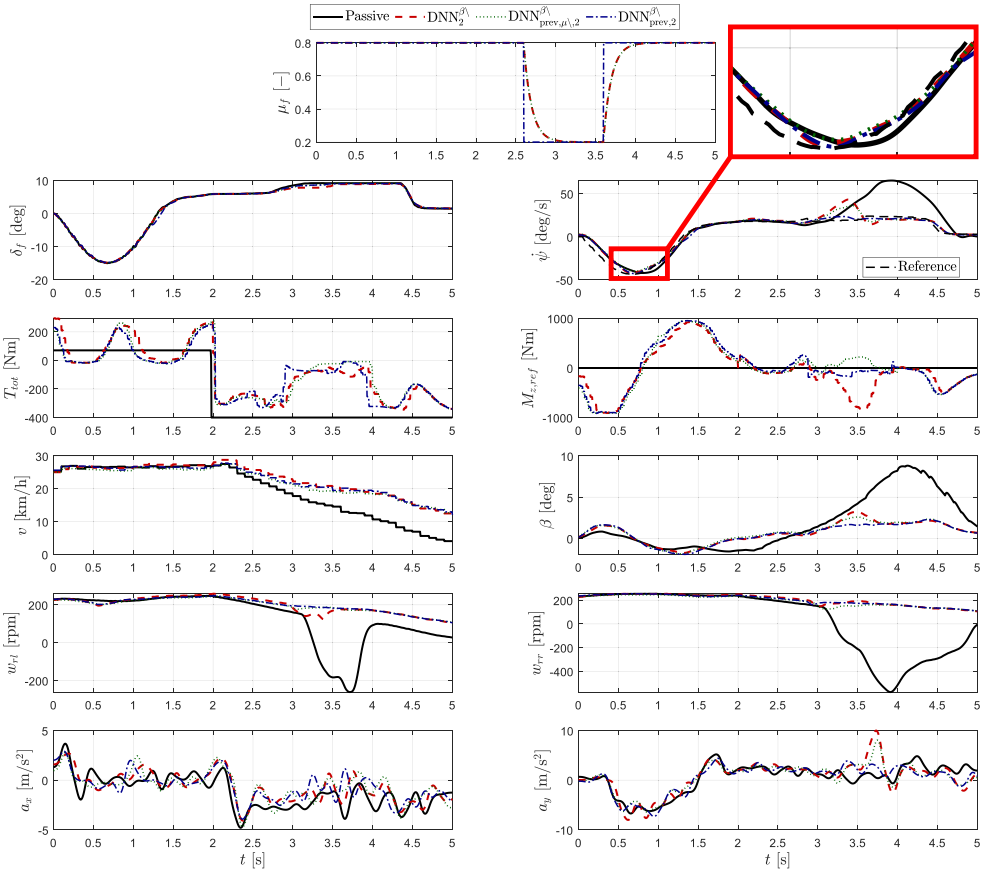


Figure 15. Comparison of the DNN-based control configurations during the considered sine-with-dwell with variable tyre-road friction conditions.

5.2. Results and discussion

Figure 15 reports the sine-with-dwell results. The manoeuvre is executed with a $T_{tot,d}$ profile, having a transition from constant positive to constant negative values at ~ 2 s, immediately after the steering input reversal, before the transition to the low-friction area. Approximately 1 s before the steering input is brought down to zero, the EV goes back to high-friction operation. For fairness of comparison, to have the same initial speed in the most relevant part of the manoeuvre, i.e. while the vehicle is on the low-friction surface, the torque demand of the passive vehicle in the first 2 s is lower than for the controlled versions. This behaviour derives from the NMPC traction control and anti-lock braking functionality in (9iii) and (9v), which has been learnt by the DNNs. As a result, the torque demand $T_{tot,d}$ does not necessarily correspond to the actual torque delivered by the controlled vehicles. The friction factor profiles in the first subplot correspond to (i) the output of the time-based lookup table before the preview block for the $\text{DNN}^{\beta\setminus}_{\text{prev},2}$ configuration and (ii) the delayed profile affected by the simulated estimation delay for the configurations without preview.

During the transient phase from 0.7 to 1 s, all DNNs achieve better $\dot{\psi}_{ref}$ tracking with respect to the passive case, which has a tracking delay during the $|\delta_f|$ reduction phase. The controllers operate both through their TVC and longitudinal slip control functionalities. In fact, while the EV is on the low-μ panels with the rear wheels, i.e. from $\sim$2.8 to $\sim$3.8 s, the controllers reduce the total torque to limit the longitudinal tyre slip. In this section of the manoeuvre, the yaw rate error peak decreases thanks to the addition of the preview information, and the configuration ranking, from the worst to the best one, is Passive, $\text{DNN}_2^{\beta\backslash}$, $\text{DNN}_{\text{prev},\mu\backslash,2}^{\beta\backslash}$ and $\text{DNN}_{\text{prev},2}^{\beta\backslash}$.

The manoeuvre was repeated 10 times for each controller configuration, to assess repeatability and robustness against the inevitable experimental uncertainties and disturbances. Also $\text{DNN}_{\text{prev},2}^{\beta}$ was tested, although its results are not included in Figure 15, due to their very marginal improvement with respect to those of $\text{DNN}_{\text{prev},2}^{\beta\backslash}$, which would make them indistinguishable in the plot. As a summary, for each KPI, Figure 16 shows the maximum, mean and minimum values – hence the subscripts 'max', 'mean' and 'min' in the notations – obtained by the controllers across the 10 attempts. Both configurations without the $\boldsymbol{\mu_{ij}}$ preview show instances of significantly oversteering EV behaviour, with $\dot{\psi}_{err,\max}$ exceeding 180 deg/s, at least once. Nevertheless, by comparing the mean values of the KPIs, the $\boldsymbol{\delta_f}$ preview enhances the cornering response, e.g. for $\text{DNN}_{\text{prev},\mu\backslash,2}^{\beta\backslash}$ $\dot{\psi}_{err,\text{RMS}}$ is 51% lower than for $\text{DNN}_2^{\beta\backslash}$. $\text{DNN}_{\text{prev},2}^{\beta\backslash}$, including the $\boldsymbol{\mu_{ij}}$ preview as well, reduces $\dot{\psi}_{err,\text{RMS}}$ by a further 82% with respect to $\text{DNN}_{\text{prev},\mu\backslash,2}^{\beta\backslash}$, and exhibits consistently stable behaviour. The difference between $\text{DNN}_{\text{prev},2}^{\beta\backslash}$ and $\text{DNN}_{\text{prev},2}^{\beta}$ is in terms of total wheel torque and direct yaw moment, but not in terms of resulting cornering responses (these are overlapped), i.e. for $\text{DNN}_{\text{prev},2}^{\beta}$ the average $T_{tot,\text{RMSE}}$ and $M_{z,ref,RMS}$ values are 10% smaller and 7% greater.

Figure 17 reports the double step steer results in the time domain, while the trajectories, recorded through the on-board GPS of the EV demonstrator, are in Figure 18. Figure 18 also includes the reference trajectory, which was obtained in the postprocessing phase (i.e. the controllers did not use it), through the time integration of the velocity components $\dot{X}_{ref}$ and $\dot{Y}_{ref}$:

$$\begin{aligned} \dot{X}_{ref} &= V\cos(\psi_{ref}+\beta) \\ \dot{Y}_{ref} &= V\sin(\psi_{ref}+\beta) \end{aligned} \tag{30}$$

where ψ_{ref} is the reference yaw angle resulting from the integration of the $\dot{\psi}_{ref}$ profile recorded for $\text{DNN}_{\text{prev},2}^{\beta\backslash}$, which was also selected for the V and β profiles to be used in (30). Hence, such a trajectory represents the desirable one, based on the reference yaw rate profile, and thus the applied steering input.

While turning right, from 0.5 to 1 s, the vehicle passes through the first low-friction area with the left wheels only. Then, at $\sim$2.4 s, while changing direction, the vehicle reaches the second low-friction area with both front wheels. In parallel, the $T_{tot,d}$ profile – in this case, identical for all configurations – transitions from its maximum traction value (400 Nm) to maximum regeneration (−400 Nm) at $\sim$2.2 s, after the first low-friction patch, before going back to traction (165 Nm) at $\sim$2.7 s, while the rear wheels are on the low-friction panel. The passive vehicle spins at the end of the manoeuvre, and reaches a yaw rate magnitude exceeding 100 deg/s immediately after the second low-friction area. YMO-TVC

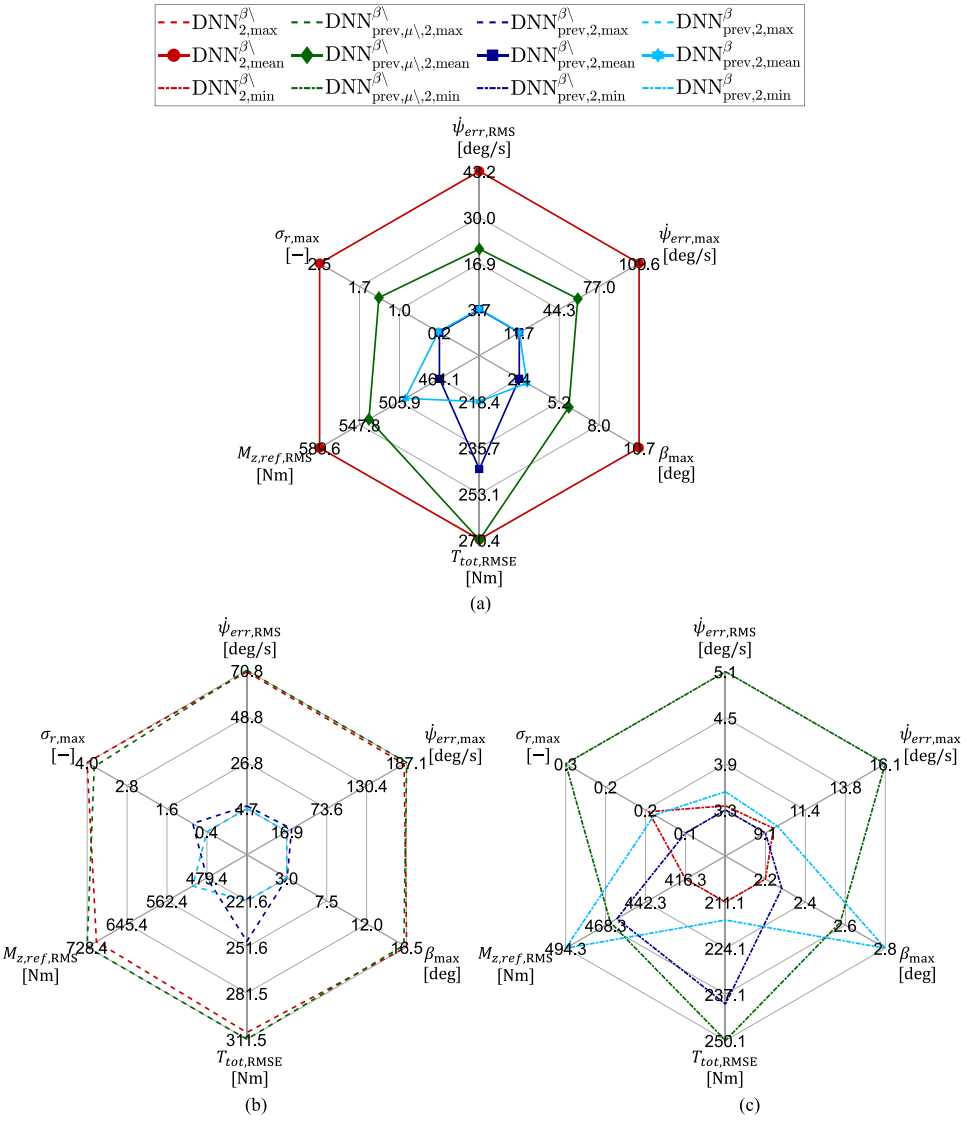


Figure 16. Mean (a), maximum (b) and minimum (c) values of the most relevant KPIs across the modified sine-with-dwell test repetitions with variable tyre-road friction conditions.

gets on the verge of instability, with a yaw rate peak of 70.6 deg/s at 4.6 s, but then brings the yaw rate response towards its reference profile. $\mathrm{DNN}^{\beta\backslash}_{\mathrm{prev},2}$ provides consistently desirable $\dot{\psi}_{ref}$ tracking across the experiment, with $\dot{\psi}_{err,\max}$ and $\dot{\psi}_{err,\mathrm{RMS}}$ of 11.8 and 5.0 deg/s, corresponding to 82% and 79% reductions with respect to YMO-TVC. This confirms the potential of the proposed TVC solution.

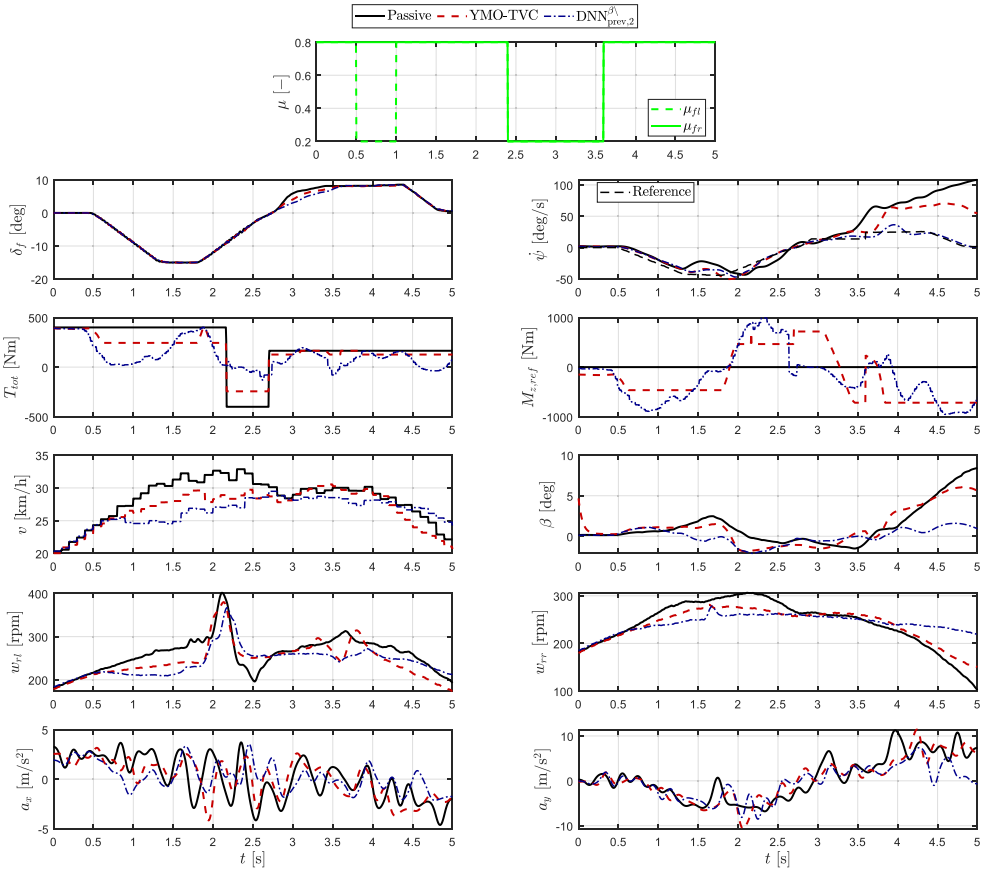


Figure 17. Comparison among the passive and active vehicles during the double step steer manoeuvre with variable tyre-road friction conditions.

6. Conclusions

Through extensive simulations and proof-of-concept experiments, this study implemented and evaluated novel IL-based TVC strategies, i.e. $DNN^{\beta\backslash}_{prev,4}$ and $DNN^{\beta\backslash}_{prev,2}$, respectively, for AWD and RWD EV architectures, including road preview and without the need for sideslip angle feedback. The main outcomes are:

- Across more than 200 simulated manoeuvres, $DNN^{\beta\backslash}_{prev,4}$ shows similar performance to an NMPC-based TVC algorithm with road preview, i.e. $NMPC_{prev}$. Moreover, $DNN^{\beta\backslash}_{prev,4}$ achieves a 40% reduction of the root mean square value of the yaw rate tracking error, with respect to the NMPC implementation without road preview, $NMPC_{base}$ and – very importantly – has 150 and 40 times lower values of the turnaround time than $NMPC_{prev}$ and $NMPC_{base}$.
- During the simulated sinusoidal steering manoeuvre, $DNN^{\beta}_{prev,4}$, including sideslip angle feedback, brings a very marginal (4%) yaw rate tracking improvement with respect to $DNN^{\beta\backslash}_{prev,4}$, which becomes negligible during the experimental sine-with-dwell test

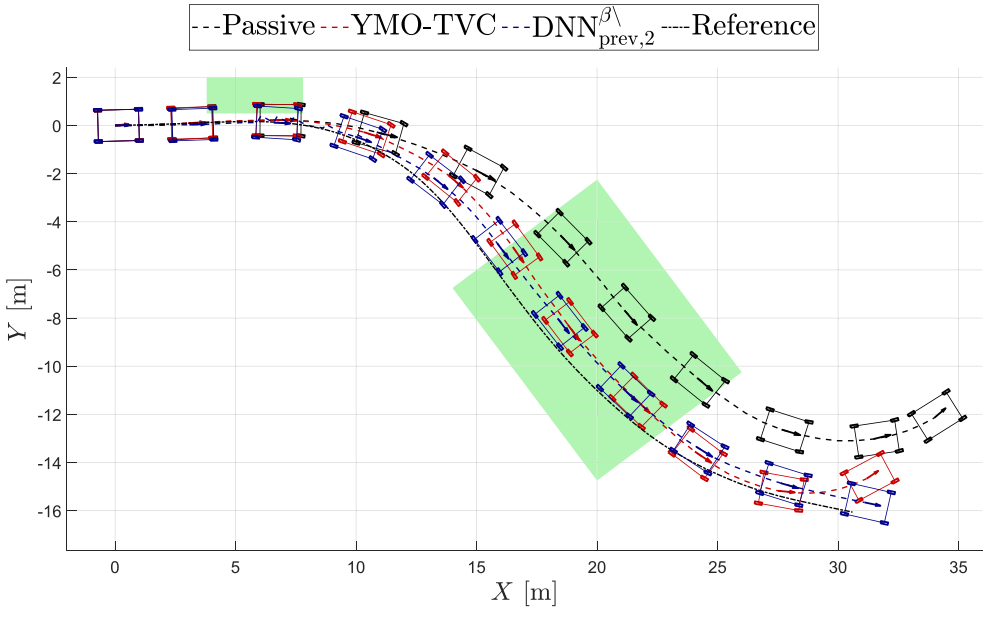


Figure 18. Trajectories of the passive and active vehicles during the double step steer manoeuvre with variable tyre-road friction conditions, where the green rectangles indicate the low-friction areas. The reference trajectory was obtained in the postprocessing phase through the method based on (30).

with the corresponding controller set-ups for the RWD EV. This outcome confirms that the DNN configuration can generate optimal control actions even without direct sideslip measurements, thanks to its ability to generalise from training datasets across a wide variety of vehicle states and control actions, and by leveraging a history of yaw rate inputs to implicitly capture the effect of sideslip on the vehicle dynamics.

- When considering parametric variations through Monte Carlo analyses, $\text{DNN}^{\beta\backslash}_{\text{prev},4}$ is more robust than $\text{NMPC}_{\text{base}}$, and more than halves the considered yaw rate tracking indicator, despite having a marginally wider dispersion range.
- The experimental assessment highlights the $\text{DNN}^{\beta\backslash}_{\text{prev},2}$ potential, which effectively implements the TVC and longitudinal tyre slip control functionalities on highly variable friction surfaces, providing significantly safer cornering than a state-of-the-art benchmarking direct yaw moment controller without preview, i.e. YMO-TVC.

Although the offline-trained DNN-based TVC demonstrated superior performance with respect to a more traditional TVC during the experimental session, its effectiveness may degrade over time due to vehicle ageing. To maintain or even enhance performance, future work will focus on the online adaptation of the neural network.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported in part by the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant 872907, in part by the European Union's Horizon Europe research and innovation program under grant 101138110, in part by the Industrial Technology Research Grant Program of the New Energy and Industrial Technology Development Organization (NEDO) of Japan under grant 05A48701d and in part by the Ministry of Education, Culture, Sports, Science, and Technology of Japan under grants 22246057, 26249061, 22K14283 and 25K07790.

ORCID

Davide Lazzarini http://orcid.org/0009-0001-8703-5040

References

[1] Asperti M, Vignati M, Sabbioni E. On torque vectoring control: review and comparison of state-of-the-art approaches. Machines. 2024;12(3):160. doi:10.3390/machines12030160
[2] Dalboni M, Tavernini D, Montanaro U, et al. Nonlinear model predictive control for integrated energy-efficient torque-vectoring and anti-roll moment distribution. IEEE/ASME Trans Mechatron. 2021;26(3):1212–1224. doi:10.1109/TMECH.2021.3073476
[3] Wang J, Gao S, Wang K, et al. Wheel torque distribution optimization of four-wheel independent-drive electric vehicle for energy efficient driving. Control Eng Pract. 2021;110:104779. doi:10.1016/j.conengprac.2021.104779
[4] Ricco M, Percolla A, Rizzo GC, et al. On the model-based design of front-to-total anti-roll moment distribution controllers for yaw rate tracking. Veh Syst Dyn. 2022;60(2):569–596. doi:10.1080/00423114.2020.1825753
[5] Cai L, Liao Z, Wei S, et al. Novel direct yaw moment control of multi-wheel hub motor driven vehicles for improving mobility and stability. IEEE Trans Ind Appl. 2023;59(1):591–600. doi:10.1109/TIA.2022.3211831
[6] Guo J, Wang J, Luo Y, et al. Takagi–Sugeno fuzzy-based robust $H\infty$ integrated lane-keeping and direct yaw moment controller of unmanned electric vehicles. IEEE/ASME Trans Mechatron. 2021;26(4):2151–2162. doi:10.1109/TMECH.2020.3032998
[7] Lee JE, Kim BW. A novel adaptive non-singular fast terminal sliding mode control for direct yaw moment control in 4WID electric vehicles. Sensors. 2025;25(3):941. doi:10.3390/s25030941
[8] Hu J, Zhang K, Zhang P, et al. Direct yaw moment control for distributed drive electric vehicles based on hierarchical optimization control framework. Math. 2024;12(11):1715. doi:10.3390/math12111715
[9] Lucchini A, Formentin S, Corno M, et al. Torque vectoring for high-performance electric vehicles: an efficient MPC calibration. IEEE Control Syst Lett. 2020;4(3):725–730. doi:10.1109/LCSYS.2020.2981895
[10] Amer NH, Dalboni M, Georgiev P, et al. Integrated torque-vectoring and anti-roll moment distribution strategies based on optimal control: influence of model complexity and road curvature preview. Veh Syst Dyn. 2024;62(10):2533–2566. doi:10.1080/00423114.2023.2296586
[11] Ueno T, Pousseur H, Nguyen BM, et al. Proposal of on-board camera-based driving force control method for autonomous electric vehicles. 2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM). Seattle, WA, USA: IEEE; 2023. p. 424–429. doi:10.1109/AIM46323.2023.10196211
[12] Takacs A, Haidegger T. A method for mapping V2X communication requirements to highly automated and autonomous vehicle functions. Future Internet. 2024;16(4):108. doi:10.3390/fi16040108
[13] Yim S. Preview controller design for vehicle stability with V2V communication. IEEE Trans Intell Transp Syst. 2016;18(6):1–10. doi:10.1109/TITS.2016.2607283

[14] Parra A, Tavernini D, Gruber P, et al. On pre-emptive vehicle stability control. Veh Syst Dyn. 2022;60(6):2098–2123. doi:10.1080/00423114.2021.1895229
[15] Tavernini D, Metzler M, Gruber P, et al. Explicit nonlinear model predictive control for electric vehicle traction control. IEEE Trans Control Syst Technol. 2019;27(4):1438–1451. doi:10.1109/TCST.2018.2837097
[16] Taherian S, Kuutti S, Visca M, et al. Self-adaptive torque vectoring controller using reinforcement learning. 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). Indianapolis, IN, USA: IEEE; 2021. p. 172–179. doi:10.1109/ITSC48978.2021.9564494
[17] Deng H, Zhao Y, Nguyen AT, et al. Fault-tolerant predictive control with deep-reinforcement-learning-based torque distribution for four in-wheel motor drive electric vehicles. IEEE/ASME Trans Mechatron. 2023;28(2):668–680. doi:10.1109/TMECH.2022.3233705
[18] Wei H, Zhang N, Liang J, et al. Deep reinforcement learning based direct torque control strategy for distributed drive electric vehicles considering active safety and energy saving performance. Energy. 2022;238:121725. doi:10.1016/j.energy.2021.121725
[19] Deng H, Zhao Y, Lin F, et al. Deep reinforcement learning-based torque vectoring control considering economy and safety. Machines. 2023;11(4):459. doi:10.3390/machines11040459
[20] Lin Y, McPhee J, Azad NL. Comparison of deep reinforcement learning and model predictive control for adaptive cruise control. IEEE Trans Intell Veh. 2021;6(2):221–231. doi:10.1109/TIV.2020.3012947
[21] Tahami F, Farhangi S, Kazemi R. A fuzzy logic direct yaw-moment control system for all-wheel-drive electric vehicles. Veh Syst Dyn. 2004;41(3):203–221. doi:10.1076/vesd.41.3.203.26510
[22] Dendaluce Jahnke M, Cosco F, Novickis R, et al. Efficient neural network implementations on parallel embedded platforms applied to real-time torque-vectoring optimization using predictions for multi-motor electric vehicles. Electronics. 2019;8(2):250. doi:10.3390/electronics8020250
[23] Chen H, Zhang J, Lv C. RHONN modelling-enabled nonlinear predictive control for lateral dynamics stabilization of an in-wheel motor driven vehicle. IEEE Trans Veh Technol. 2022;71(8):8296–8308. doi:10.1109/TVT.2022.3172870
[24] Wang X, Wang S, Liang X, et al. Deep reinforcement learning: a survey. IEEE Trans Neural Netw Learn Syst. 2024;35(4):5064–5078. doi:10.1109/TNNLS.2022.3207346
[25] Le Mero L, Yi D, Dianati M, et al. A survey on imitation learning techniques for end-to-end autonomous vehicles. IEEE Trans Intell Transp Syst. 2022;23(9):14128–14147. doi:10.1109/TITS.2022.3144867
[26] Zare M, Kebria PM, Khosravi A, et al. A survey of imitation learning: algorithms, recent developments, and challenges. IEEE Trans Cybern. 2024;54(12):7173–7186. doi:10.1109/TCYB.2024.3395626
[27] Kim S, Han K, Choi SB. Imitation learning of nonlinear model predictive control for emergency collision avoidance. IEEE Trans Intell Veh. 2024;9(1):2908–2922. doi:10.1109/TIV.2023.3309962
[28] Lee T, Kang Y. Performance analysis of deep neural network controller for autonomous driving learning from a nonlinear model predictive control method. Electronics. 2021;10(7):767. doi:10.3390/electronics10070767
[29] Lee T, Seo D, Lee J, et al. Real-time drift-driving control for an autonomous vehicle: learning from nonlinear model predictive control via a deep neural network. Electronics. 2022;11(17):2651. doi:10.3390/electronics11172651
[30] Ajak NA, Ong WH, Malik OA. A comparison of imitation learning pipelines for autonomous driving on the effect of change in Ego-vehicle. 2024 IEEE Intelligent Vehicles Symposium (IV). Jeju Island, South Korea: IEEE; 2024. p. 1693–1698. doi:10.1109/IV55156.2024.10588638
[31] Esmailzadeh E, Goodarzi A, Vossoughi GR. Optimal yaw moment control law for improved vehicle handling. Mechatronics. 2003;13(7):659–675. doi:10.1016/S0957-4158(02)00036-3
[32] Geng C, Mostefai L, Denai M, et al. Direct yaw-moment control of an in-wheel-motored electric vehicle based on body slip angle fuzzy observer. IEEE Trans Ind Electron. 2009;56(5):1411–1419. doi:10.1109/TIE.2009.2013737

[33] Tchamna R, Youn I. Yaw rate and side-slip control considering vehicle longitudinal dynamics. Int J Automot Technol. 2013;14:53–60. doi:10.1007/s12239-013-0007-1
[34] Nam K, Fujimoto H, Hori Y. Lateral stability control of in-wheel-motor-driven electric vehicles based on sideslip angle estimation using lateral tire force sensors. IEEE Trans Veh Technol. 2012;61(5):1972–1985. doi:10.1109/TVT.2012.2191627
[35] Tavolo G, So KM, Tavernini D, et al. On antilock braking systems With road preview through nonlinear model predictive control. IEEE Trans Ind Electron. 2024;71(8):9436–9448. doi:10.1109/TIE.2023.3314917
[36] Scamarcio A, Caponio C, Mihalkov M, et al. Predictive anti-jerk and traction control for V2X connected electric vehicles with central motor and open differential. IEEE Trans Veh Technol. 2023;72(6):7221–7239. doi:10.1109/TVT.2022.3143497
[37] Beliautsou V, Beliautsou A, Ivanov V, et al. Drone-to-vehicle integration of data: design concept and application to vehicle automation system. IEEE Veh Technol Mag. 2025;20(4):114–122. doi:10.1109/MVT.2025.3540749
[38] Jung C, Lee D, Lee S, et al. V2X-communication-aided autonomous driving: system design and experimental validation. Sensors. 2020;20(10):2903. doi:10.3390/s20102903
[39] Lazzarini D, Depalo M, Tota A, et al. On the benefits of road preview on integrated chassis control. Veh Syst Dyn. 2026: 1–31. doi:10.1080/00423114.2026.2634102
[40] López CP. Optimization techniques via the optimization toolbox. In: Dominic Shakeshaft, editor. MATLAB optimization techniques. Berkeley, CA: Apress; 2014. p. 85–108. doi:10.1007/978-1-4842-0292-0_6
[41] Snoek J, Larochelle H, Adams, RP. Practical Bayesian optimization of machine learning algorithms. arXiv prepreint arXiv:1206.2944. 2012. doi:10.48550/arXiv.1206.2944
[42] Kingma DP, Ba J. Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980. 2014. doi:10.48550/arXiv.1412.6980
[43] Nair V, Hinton GE. Rectified linear units improve restricted Boltzmann machines. In: Johannes Fürnkranz, Thorsten Joachims, editors. Proceedings of the 27th International Conference on Machine Learning (ICML 2010). Madison, WI, USA: Omnipress; 2010. p. 807–814. doi:10.5555/3104322.3104425
[44] Houska B, Ferreau HJ, Diehl M. ACADO toolkit – an open-source framework for automatic control and dynamic optimization. Optim Control Appl Met. 2011;32(3):298–312. doi:10.1002/oca.939
[45] Tavernini D, Vacca F, Metzler M, et al. An explicit nonlinear model predictive ABS controller for electro-hydraulic braking systems. IEEE Trans Ind Electron. 2020;67(5):3990–4001. doi:10.1109/TIE.2019.2916387
[46] Ueno T, Nguyen BM, Nagai S, et al. Wheel velocity based cascade driving force control for electric vehicles. IEEE/ASME Trans Mechatron. 2025;30(4):2620–2631. doi:10.1109/TMECH.2024.3456151
[47] Ueno T, Nguyen BM, Fujimoto H. Direct yaw moment control for electric vehicles with variable-rate-slip-ratio-limiter based driving force control. 2023 IEEE International Conference on Mechatronics (ICM). Loughborough: IEEE; 2023. p. 1–6. doi:10.1109/ICM54990.2023.10102086