POLITECNICO DI TORINO
Repository ISTITUZIONALE

Energy-efficient eco-driving for human-driven electric vehicles via adaptive drivability maps with predictive driver modeling

Original
Energy-efficient eco-driving for human-driven electric vehicles via adaptive drivability maps with predictive driver modeling / Ciravegna, L., Dimauro, L., Frison, G., Alberti, F., Galvagno, E., Sorniotti, A.. - In: APPLIED ENERGY. - ISSN 1872-9118. - 420:(2026). [10.1016/j.apenergy.2026.128077]

Availability:
This version is available at: 11583/3011534 since: 2026-05-28T17:24:31Z

Publisher:
Elsevier

Published
DOI:10.1016/j.apenergy.2026.128077

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

19 November 2026

Applied Energy 420 (2026) 128077

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

Energy-efficient eco-driving for human-driven electric vehicles via adaptive drivability maps with predictive driver modeling

Luca Ciravegna, Luca Dimauro, Gianluca Frison, Fabio Alberti, Enrico Galvagno, Aldo Sorniotti *

Department of Mechanical and Aerospace Engineering (DIMEAS), Politecnico di Torino, 10129 Turin, Italy

HIGHLIGHTS

- Adaptive drivability maps reduce energy consumption in human-driven electric vehicles.
- Real-time driver assistance improves efficiency without reducing user acceptance.
- Driving behavior prediction enables less intrusive eco-driving support.
- Human driver modeling allows personalized energy-efficient assistance.
- Energy consumption is reduced by up to 4% with improved driver acceptance.

ARTICLE INFO

Keywords:
Driving assistance system
Eco-driving
Driver intent prediction
Electric vehicles
Energy efficiency
Human driver modeling
Control intrusiveness

ABSTRACT

This study proposes a proof-of-concept eco-driving assistance system (EDAS) for human-driven electric vehicles (EVs), based on adaptive drivability maps (ADMs) integrated with predictive driver intent modeling. Differently from existing eco-driving systems that either rely on advisory feedback or apply torque corrections without anticipating human actions, the proposed approach explicitly embeds a preview of the driver's future pedal inputs within a nonlinear model predictive control (NMPC) formulation, where energy efficiency and driver acceptance are jointly addressed at the control level. Driver intent preview is generated through neural network (NN) models, employing two complementary architectures: i) feedforward NNs (FFNNs), offline trained using a physics-based human driver model (PB-HDM) – preliminarily validated on a static driving simulator – for longitudinal control, corresponding to distinct driving styles; and ii) an online-trained long short-term memory (LSTM) NN, continuously updated to capture intermediate or evolving driver behaviors that cannot be strictly categorized. Online switching algorithms select the most appropriate NN configuration in real time. By leveraging driver intent prediction, the proposed ADM framework achieves energy savings through smoother and driver-aware torque corrections. Real-time capable simulation results show that: i) energy consumption is reduced by ~4% compared to the same vehicle with static drivability maps; and ii) for comparable energy savings, control intrusiveness indicators are approximately halved by the inclusion of driver intent preview, in comparison with the same ADM algorithm excluding the NN-based preview. The results demonstrate the effectiveness of predictive, driver-centered drivability adaptation as a promising and scalable pathway for energy-efficient assistance in human-driven EVs.

1. Introduction

On-road transportation remains a major contributor to global energy consumption and greenhouse gas emissions [1], motivating the progressive adoption of electric vehicles (EVs) [2]. However, their limited driving range as well as the required charging time and infrastructure still represent an obstacle to their widespread adoption [3]. In this context, improving the energy efficiency of EVs through operational and control-oriented strategies remains a central challenge.

Eco-driving has consequently emerged as a prominent research topic, encompassing strategies that optimize speed profiles, powertrain operation, and interactions with the road infrastructure [4–8]. Within automated and connected vehicles frameworks, extensive research has

* Corresponding author.
E-mail address: aldo.sorniotti@polito.it (A. Sorniotti).

https://doi.org/10.1016/j.apenergy.2026.128077
Received 25 January 2026; Received in revised form 12 April 2026; Accepted 18 May 2026
Available online 4 June 2026
0306-2619/© 2026 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Nomenclature

Abbreviations

A	Active configuration
A+P	Active with preview configuration
ADM	Adaptive drivability map
CNS	Central nervous system
DoE	Design of experiments
DT	Digital twin
DTW	Dynamic time warping
Eco-ACC	Ecological adaptive cruise control
EDAS	Eco-driving assistance system
EM	Electric machine
EV	Electric vehicle
FFD	Full factorial design
FFNN	Feedforward neural network
FFNN-DT	Feedforward neural network based digital twin
FPC	Foot-pedal contact
GTO	Golgi tendon organ
KPI	Key performance indicator
LHS	Latin hypercube sampling
LSTM	Long short-term memory
LSTM-DT	Long short-term memory based digital twin
LUT	Look-up table
MSE	Mean squared error
NPMC	Nonlinear model predictive control
NMS	Neuromuscular system
NN	Neural network
NN-DT	Neural network digital twin
NLP	Nonlinear programming
OCP	Optimal control problem
P	Passive configuration
PB-HDM	Physics-based human driver model
RMS	Root mean square
RNN	Recurrent neural network
SQP	Sequential quadratic programming
V2X	Vehicle-to-everything

Parameters and symbols

A	Vehicle frontal area
a_y	Lateral acceleration
$a_{y,ratio}$	Ratio between $\overline{a}_y$ and $\overline{a}_{y,lim}$
B	Batch size
$b_{\eta NN,l}$	Bias matrix of the layer l of the NN
$b_{LSTM,f/i/c/o,l}$	Bias matrices of the LSTM-DT
c_{drag}	Aerodynamic drag coefficient
$c_{LSTM,t,l}$	LSTM cell state
$\widetilde{c}_{LSTM,t,l}$	LSTM cell candidate
d_{safe}	Safety distance
d_{static}	Static safety distance
DS	Driving style
ds_q	Driving style identified based on KPI q
$\widehat{E}$	Normalized term associated with energy consumption in $J_{opt,w}$
E_d	Energy consumed per unit distance
$F_{p,brk}$	Brake pedal force
F_{res}	Total resistance force
F_s	Correction factor used in $\dot{x}_{max,a_y}$ computation
$F_{x,brk}$	Friction braking longitudinal force
$F_{x,corr}$	Corrective longitudinal powertrain force
$\widehat{F}_{x,corr}$	Nondimensional cumulated corrective longitudinal powertrain force
$F_{x,pwt}$	Powertrain longitudinal tire force contribution
$F_{x,pwt,reg,min}$	Minimum available EM force during regeneration
$F_{x,req,drv}$	Total longitudinal tire force request by the driver
$F_{x,tot}$	Total longitudinal force
$F_{x,\mu,max}$	Maximum traction force compatible with the available μ
f_0, f_2	Rolling resistance coefficients
$f_{LSTM,t,l}$	LSTM forget gate
$f_{\eta NN,l}$	Activation function of the layer l of the NN
g	Gravitational acceleration
H	Transfer function
$h_{drv,FFNN}$	Input vector of the FFNN-DT
$h_{LSTM,t,l}$	LSTM hidden state output
$h_{\eta NN,l}$	Input vector of the layer l of the NN
$i_{LSTM,t,l}$	LSTM input gate
J_{diff}	Mass moment of inertia of the mechanical differential
J_E	ADM NMPC cost function contribution associated with E_d
J_{EM}	Mass moment of inertia of the electric motor
J_{F_x}	ADM NMPC cost function contribution associated with the tracking of $F_{x,pwt,drv}$
$J_{F_{x,corr}}$	ADM NMPC cost function contribution associated with $F_{x,corr}$
$J_{\dot{F}_{x,corr}}$	ADM NMPC cost function contribution associated with $\dot{F}_{x,corr}$
$J_{gear,eq}$	Equivalent mass moment of inertia of the gearbox
J_{opt}	NMPC cost function
$J_{opt,w}$	Cost function for ADM weights optimization
J_u	CNS NMPC cost function contribution associated with $\dot{F}_{x,req,drv}$
J_{VT}	CNS NMPC cost function contribution associated with the tracking of $\dot{x}_{ref}$
$J_{w,eq}$	Individual wheel mass moment of inertia
$J_{w,eq,tot}$	Equivalent wheel mass moment of inertia
$J_{\dot{x}}$	ADM NMPC cost function contribution associated with the tracking of $\dot{x}_{pred}(N)$
K	Stiffness
k	Discrete time step along T_p
L	Wheelbase
$\mathscr{L}_t$	Loss to be minimized during online training of the LSTM
l	NN layer
$l(\boldsymbol{x}(k), u(k), p(k))$	NMPC stage cost
$l_N(\boldsymbol{x}(N), p(N))$	NMPC terminal cost
M	Vehicle mass
M_{eq}	Equivalent vehicle mass
N	Number of prediction steps
n	Rotational speed
$o_{LSTM,t,l}$	LSTM output gate
P	Power
$p(t)$	NMPC external parameters vector
R_w	Wheel radius
s	Laplace operator
T_{brk}	Friction braking torque
T_{bump}	Bump stop torque
$T_{corrected}$	Corrected powertrain torque
T_{EM}	EM torque
T_p	NMPC prediction horizon
$T_{preload}$	Preload torque
$T_{ref\ acc\ pedal}$	Reference accelerator pedal torque
$T_{ref,NMS}$	Reference torque generated by the NMS
t	Time
t_h	Headway time
$\tanh(\cdot)$	Hyperbolic tangent activation function

U	Sequence of optimal control inputs
$U_{LSTM,f/i/c/o,l}$	Weight matrices of the LSTM-DT
$u(t)$	NMPC control inputs vector
$v_{lim,ratio}$	Ratio between $\overline{\dot{x}}_h$ and $\overline{\dot{x}}_{lim}$
W	NMPC cost function weights vector
$W_{LSTM,f/i/c/o,l}$	Weight matrices of the LSTM-DT
w_E	Weight associated with J_E
w_{en}	Weight associated with $\widehat{E}$
w_{F_x}	Weight associated with J_{F_x}
$w_{F_{x,corr}}$	Weight associated with $J_{F_x,corr}$
$w_{F_{x,track}}$	Weight associated with $\widehat{\Delta F}_{x,track}$
$w_{\dot{F}_{x,corr}}$	Weight associated with $J_{\dot{F}_x,corr}$
$w_{\dot{x}}$	Weight associated with $J_{\dot{x}}$
$w_{\dot{x}_{pred}}$	Weight associated with $\widehat{\Delta\dot{x}}_{pred}$
w_{ss}	Brake blending smoothness coefficient
w_u	Weight associated with J_u
w_{VT}	Weight associated with J_{VT}
$w_{\eta NN,l}$	Weight matrix of the layer l of the NN
$x(t)$	NMPC state vector
x_0	Initial value of the states
x_{LSTM}	Input vector of the LSTM
x_{rel}	Relative distance between host and preceding vehicle
$x_{rel,ratio}$	Ratio between $\overline{x}_{rel}$ and $\overline{x}_{rel,thr}$
$\dot{x}_h$	Host vehicle speed
$\dot{x}_{lim}$	Road speed limit
$\dot{x}_{max,a_y}$	Maximum speed considering the maximum a_y
$\dot{x}_p$	Preceding vehicle speed
$\dot{x}_{pred}(N)$	Predicted host vehicle speed
$\dot{x}_{ref}$	Expected reference vehicle speed
$y_{drv,FFNN}$	Output vector of the FFNN-DT
$y_{drv,LSTM}$	Output vector of the LSTM-DT
$y_{drv,LSTM,ref,k,b}$	Time-aligned α_p value used for comparison with predicted value
$y_{\eta NN,l}$	Output vector of the layer l of the NN
$A_{p,pred}$	Predicted accelerator pedal position sequence
α_f	Angle of the driver foot
α_{musc}	Desired angle of the driver foot
α_p	Accelerator pedal angle
$\alpha_{p,ratio}$	Ratio between $\overline{\alpha}_p$ and $\overline{\alpha}_{p,av}$
$\alpha_{p,r,DTW}$	Aligned α_p time profiles through DTW
α_{road}	Longitudinal road slope
β	Damping coefficient
δ	Hysteresis-based switching threshold
$\Delta\alpha_{p,r,DTW}$	RMS of $\overline{\alpha}_{p,exp}$ w.r.t. $\alpha_{p,r,DTW}$
ΔE	Percentage variation of $E_{d,A/A+P}$ w.r.t. $E_{d,P}$
$\widehat{\Delta F}_{x,track}$	Normalized term associated with longitudinal tire force tracking in $J_{opt,w}$
ΔT	NMPC discretization time
$\Delta\dot{x}_{pred}$	RMS of $\dot{x}_h(t)$ w.r.t. $\dot{x}_{pred}(N)(t-T_p)$
$\widehat{\Delta\dot{x}}_{pred}$	Normalized term associated with speed tracking in $J_{opt,w}$
$\frac{\partial\mathscr{L}_t}{\partial\theta_t}$	Gradient of $\mathscr{L}_t$ w.r.t. θ_t
ϵ	Slack variable vector
η_{dvl}	Driveline efficiency
η_{LSTM}	LSTM learning rate
η_{tot}	Total powertrain efficiency
λ	L2 regularization factor
μ	Tire-road friction coefficient
$\sigma(\cdot)$	Sigmoid activation function
ρ_{air}	Air density
ρ_{road}	Road curvature
θ_t	LSTM parameters including weights and biases
τ	Powertrain time constant
τ_1, τ_2	Time constants of the activation dynamics
τ_{diff}	Final reduction ratio
τ_{tot}	Total transmission ratio
$\dot{(\cdot)}$	Time derivative
$\ddot{(\cdot)}$	Second-order time derivative
$\overline{(\cdot)}$	Average value
$\lvert\cdot\rvert$	Absolute value
$\odot$	Element-wise product

Subscripts and superscripts

act	Activation dynamics
av	Average
b	Batch number
drv	Requested by the driver
dvt	Drivetrain
emb	Embedding layer
exp	Experimental test
FC	Fully connected layer
FFNN	Quantity obtained by the FFNN-DT
fin	Final
GTO	GTO dynamics
h	Host vehicle
in	Initial
int	Intrinsic dynamics
lb	Lower boundaries
LSTM	Quantity obtained by the LSTM-DT
max	Maximum
min	Minimum
p	Pedal
PB − HDM	Quantity obtained by the PB-HDM
q	Considered KPI for the FFNN switching algorithm
r	Quantity obtained by PB-HDM, FFNN-DT or experimental tests
s	Shoe sole
tend	Tendon dynamics
thr	Threshold
ub	Upper boundaries
w	Wheel

addressed ecological adaptive cruise controllers (Eco-ACCs) and predictive speed planning strategies [9–15]. These approaches leverage vehicle automation and vehicle-to-everything (V2X) connectivity to directly regulate longitudinal motion, achieving significant energy savings under well-defined operating conditions.

However, for the foreseeable future, the vast majority of driving time will still involve human drivers [16–17]. Consequently, eco-driving assistance systems (EDAS) for human-driven vehicles remain highly relevant. Existing solutions primarily rely on driver training, providing individuals with the knowledge and skills for more efficient driving [18–19], or on advisory feedback provided through visual, auditory, or haptic interfaces [6,20–24]. Although such systems can positively influence driving behavior, their effectiveness is limited by driver compliance, acceptance, and situational variability.

On top of providing advisories, the most recent EDAS implementations dynamically adjust the driver torque demand [25–26]. These approaches represent an important step toward stronger energy optimization under human control. However, their limitation is the

absence of a driver model, i.e., the systems operate without considering the driving style or predicting the driver intention, leading to a control action independent from the driver, who could perceive the EDAS as intrusive. Ref. [27] partially addresses this issue through customized drivability maps based on identified driving styles, but without predictive capabilities, online torque adaptation, or the use of preview information such as road slope, curvature, and speed limits.

In parallel, a broad literature has investigated driver modeling, particularly in the context of steering control, in the form of neuromuscular systems (NMS) including the arm dynamics and steering wheel torque feedback [28–31]. Refs. [32] and [33] are among the few studies considering the lower limbs dynamics during driving. Although NMS and foot-pedal contact models have been employed for the design of haptic gas pedal feedback systems [34–35], they are not formulated to predict driver intent for longitudinal energy optimization.

Recent studies have explored data-driven approaches to predict EV energy consumption and velocity profiles. For instance, machine-learning-based efficiency modeling has been proposed to improve the accuracy of EV energy estimation [36]. Other studies adopt transformer-based architectures to jointly predict vehicle speed and energy demand under varying operating conditions [37], while personalized prediction methods incorporating road features have also been investigated [38].

In conclusion, the existing literature lacks EDAS implementations that integrate powertrain torque correction with predictive modeling of human driver behavior based on realistic V2X information. Even more, to the best of the authors' knowledge, the available driver models are not formulated for predicting the human control of the longitudinal vehicle dynamics. This study aims to bridge the identified gap by proposing an innovative eco-driving assistance framework based on adaptive drivability maps (ADMs), where energy optimization and driver acceptance are jointly managed through a predictive control architecture.

Differently from existing EDAS that either provide advisory feedback or apply torque corrections without explicit driver behavior modeling, the proposed framework embeds driver intent prediction directly within the NMPC optimization problem, and dynamically corrects the torque demand to concurrently address energy efficiency and control intrusiveness. In particular, this study proposes a hybrid neural network (NN) approach for driver behavior prediction, according to which: i) offline-trained feedforward neural networks (FFNNs) provide stable emulation of predefined driving styles; and ii) a recurrent neural network (RNN) based on a long short-term memory (LSTM) structure is continuously trained online, to capture intermediate and evolving driving styles. The online LSTM approach builds upon concepts introduced in [39], where NNs are trained online for control adaptation, while differing significantly in methodology and application.

The main contributions are:

- A novel ADM functionality based on nonlinear model predictive control (NMPC), which assists the driver with a customized powertrain torque correction to reduce energy consumption, while accounting for the adopted driving style, thereby minimizing intrusiveness.
- A physics-based human driver model (PB-HDM) for longitudinal vehicle control, used to generate data for training the corresponding NNs for the online driver behavior prediction. The NNs consider exteroceptive inputs on the scenario ahead, e.g., in terms of road curvature and slope, speed limits, as well as relative distance and speed of the preceding vehicle, to predict the driver accelerator and brake pedal inputs along the prediction horizon.
- A hybrid and adaptive driver modeling framework, combining offline-trained FFNNs for stable baseline emulation, an online-adaptive LSTM RNN, and a real-time switching strategy based on key performance indicators (KPIs). The integrated approach enables dynamic selection of the most appropriate NN during vehicle operation, thus enhancing prediction accuracy and customization of the resulting control action.
- A real-time capable simulation-based proof-of-concept implementation demonstrating that the proposed ADM functionality enables energy savings with minor corrections of the human driver input, outperforming the same NMPC algorithm without driver behavioral modeling.

The remainder is organized as follows: Section 2 outlines the simulation architecture; Section 3 presents the ADM formulation; Section 4 describes the driver model; Section 5 analyzes the simulation results; finally, Section 6 summarizes the main conclusions.

2. Simulation architecture

The case study EV, developed in the European project CliMAFlux, is equipped with a centralized electric machine (EM) on the front axle, a single-speed gearbox, and an open differential, connected to the wheels through half-shafts. The main vehicle and drivetrain parameters are in Table 1, while the powertrain efficiency map is in Fig. 1(a).

Fig. 2 is the simplified schematic of the simulation and control architecture. Depending on the specific analysis, in the simulation environment the human driver model is represented either by PB-HDM or the FFNN-based digital twin (FFNN-DT), see Section 4.5. The model sets the accelerator position α_p (more specifically referred to as $\alpha_{p,PB-HDM}$ in the PB-HDM case, and $\alpha_{p,FFNN}$ for FFNN-DT), and brake pedal force $F_{p,brk}$ ($F_{p,brk,PB-HDM}$ for PB-HDM, and $F_{p,brk,FFNN}$ for FFNN-DT). The latter is converted into brake caliper pressures by the electro-hydraulic unit of the by-wire braking system, and results into friction braking torque levels T_{brk}, which are not affected by the ADM functionality. The considered EV adopts a one-pedal regenerative strategy, in which the regenerative braking torque is only determined by the accelerator pedal position, i.e., the brake pedal input solely affects the friction brakes. This assumption reflects the calibration of the reference vehicle used in the proof-of-concept analysis, and does not restrict the generality of the proposed ADM formulation, which can accommodate alternative regenerative braking strategies.

Through a drivability map, α_p is converted into the reference longitudinal tire force contribution of the powertrain, $F_{x,pwt,drv}$. The map is a look-up table (LUT) receiving as inputs α_p and the EV speed $\dot{x}_h$, according to the conventional setup of production EVs. $F_{x,pwt,drv}$ is then adjusted through the corrective longitudinal powertrain force, $F_{x,corr}$, computed by the NMPC-based ADM algorithm. Such correction is influenced by the human driver intention preview, in the form of a predicted accelerator pedal position sequence, $A_{p,pred}$, obtained through the driver behavior prediction system, see Fig. 2. The driver behavior predictor is fed with information from V2X, exteroceptive sensors, and state estimators, namely: i) the road speed limit $\dot{x}_{lim}$; ii) the road curvature ρ_{road}; iii) the longitudinal road slope α_{road}; iv) the relative distance x_{rel} between the host and a potential preceding vehicle, as well as the speed and acceleration of the latter, $\dot{x}_p$ and $\ddot{x}_p$; and v) $\dot{x}_h$.

The driving style identifier recognizes the driving behavior, and selects the most appropriate pre-trained FFNN calibration among those stored in the control unit, e.g., 'Soft' or 'Aggressive' in Fig. 2. In parallel, an LSTM-based RNN is continuously trained online using data generated

Table 1
Main vehicle and drivetrain parameters.

Parameter	Symbol	Value	Unit
Vehicle mass	M	1500	kg
Wheelbase	L	2.57	m
Overall transmission ratio	τ_{tot}	7	–
Wheel radius	R_w	0.34	m
Individual wheel mass moment of inertia	$J_{w,eq}$	1.45	kg m^2
Peak EM power	$P_{EM,max}$	134	kW
Peak EM torque	$T_{EM,max}$	300	Nm
Maximum EM speed	$n_{EM,max}$	10,000	rpm

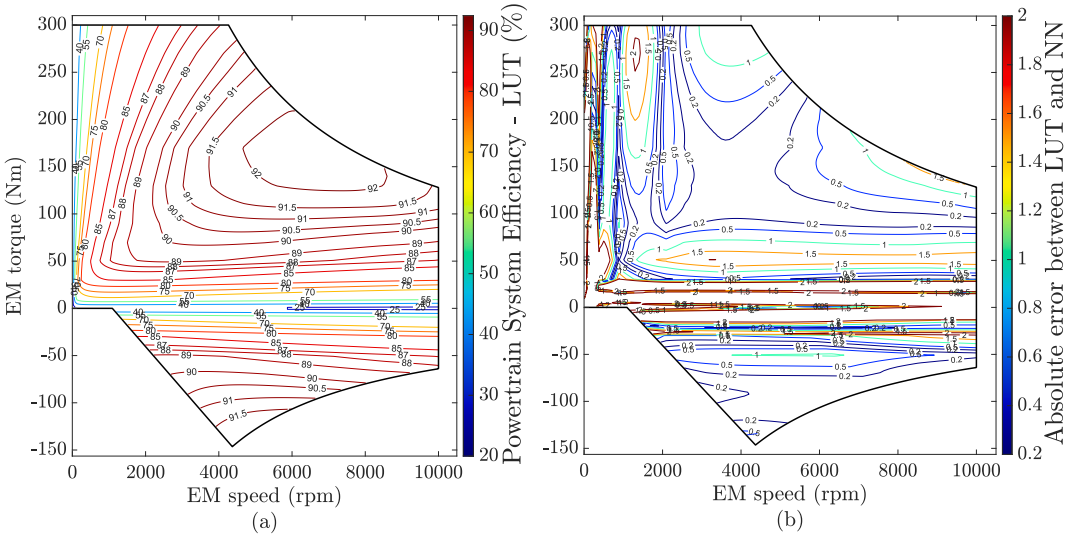


Fig. 1. Powertrain system (including inverter, EM, and driveline) efficiency map; (a) LUT; (b) Absolute error between the LUT and NN model.

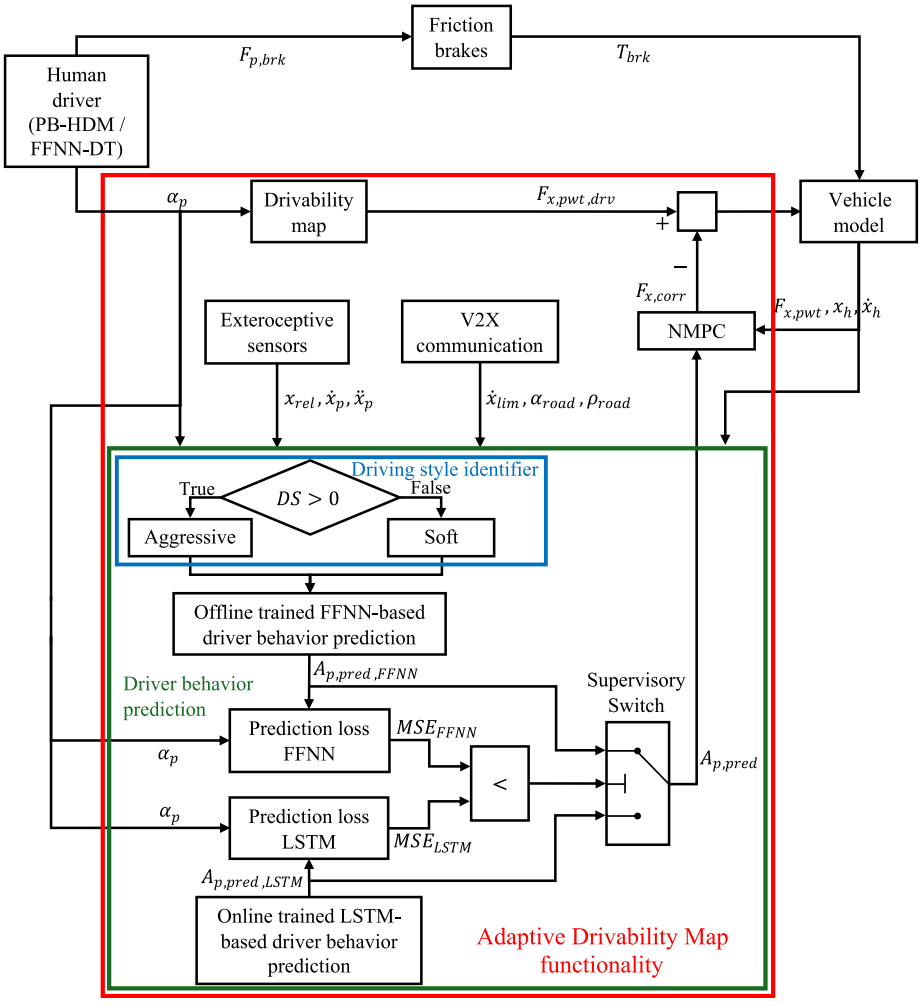


Fig. 2. Schematic of the simulation and control framework.

by the current human driver. This online adaptation enables the modeling framework to capture intermediate driving behaviors that cannot be strictly classified as 'Soft' or 'Aggressive'. The prediction performance of the selected FFNN and online-trained LSTM is evaluated through the respective mean squared error (MSE) values, MSE_{FFNN} and MSE_{LSTM}. If $MSE_{FFNN} < MSE_{LSTM}$, the supervisory switch selects the predicted accelerator pedal input profile from FFNN, i.e., $A_{p,pred} = A_{p,pred,FFNN}$; otherwise, if $MSE_{FFNN} > MSE_{LSTM}$, the LSTM-based

prediction is adopted, i.e., $A_{p,pred} = A_{p,pred,LSTM}$, see Section 4.5.

With the $A_{p,pred}$ knowledge, the NMPC can make the powertrain torque correction less disturbing for the driver, and maintain a spontaneous driving feeling. The ADM is the system consisting of the drivability map LUT, the NMPC, and the driver behavior prediction system. Following the ADM-generated correction, the traction force demand is sent to a high-fidelity Simscape model of the vehicle including the drivetrain dynamics.

3. The adaptive drivability map functionality

3.1. Vehicle prediction model formulation

This section describes the simplified model used by the NMPC to predict the relevant vehicle dynamics and resulting power losses and energy consumption, along the prediction horizon T_p.

The longitudinal vehicle dynamics are expressed by the force balance in (1):

$$F_{x,pwt} + F_{x,brk} - F_{res} = M_{eq}\ddot{x}_h \tag{1}$$

where $F_{x,pwt}$ is the total powertrain-related longitudinal tire force contribution; $F_{x,brk} \leq 0$ is the total longitudinal force caused by the friction brakes; F_{res} is the total resistance force, including the aerodynamic drag as well as the rolling resistance and road slope effects; M_{eq} is the apparent mass of the EV; and $\ddot{x}_h$ is the longitudinal vehicle acceleration, with the subscript h referring to the host (or ego) vehicle, to distinguish it from a potential preceding vehicle, detected by the radar sensor on the host EV. Within the internal model, the brake blending algorithm prioritizes regenerative braking over the friction brakes, by adopting the smooth saturation formulation from [40], which is implemented as

$$F_{x,pwt} = \frac{F_{x,tot} - F_{x,pwt,reg,min}}{1 + e^{-w_{ss}\left[F_{x,tot} - F_{x,pwt,reg,min}\right]}} + F_{x,pwt,reg,min} \tag{2}$$

where $F_{x,pwt,reg,min}$ is the minimum available longitudinal force during regeneration; and w_{ss} is a positive coefficient that defines the desired degree of smoothness. (2) approximates $F_{x,pwt} = \max\left(F_{x,tot}, F_{x,pwt,reg,min}\right)$. If $F_{x,tot} < 0$, $F_{x,brk}$ is computed as

$$F_{x,brk} = \text{deadband}^{0}_{-\infty}\left(F_{x,tot} - F_{x,pwt}\right) \tag{3}$$

In (1), M_{eq} includes four contributions:

$$\begin{aligned} M_{eq} &= M + M_{eq,w} + M_{eq,dvt} + M_{eq,EM} \\ &= M + \frac{J_{w,eq,tot}}{R_w^2} + \left[\frac{J_{diff}}{R_w^2}\tau_{diff}^2 + \frac{J_{gear,eq}}{R_w^2}\tau_{tot}^2\right] + \frac{J_{EM}}{R_w^2}\tau_{tot}^2 \end{aligned} \tag{4}$$

where M is the EV mass; $M_{eq,w}$, $M_{eq,dvt}$ and $M_{eq,EM}$ are the apparent masses of the wheels, drivetrain, and electric machine; $J_{w,eq,tot}$ is the equivalent wheel mass moment of inertia, including all relevant components, e.g., the brake discs; R_w is the wheel radius; J_{diff} is the mass moment of inertia of the mechanical differential; $J_{gear,eq}$ is the equivalent mass moment of inertia of the gearbox, referred to the EM rotor, having a mass moment of inertia J_{EM}; and τ_{tot} and τ_{diff} are the total transmission ratio and final reduction ratio. F_{res} is a function of x_h and $\dot{x}_h$:

$$F_{res}(x_h, \dot{x}_h) = \left[f_0 + f_2\dot{x}_h^2\right] M\cos(\alpha(x_h)) + Mg\sin(\alpha(x_h)) + \frac{1}{2}\rho_{air}Ac_{drag}\dot{x}_h^2 \tag{5}$$

where f_0 and f_2 are the rolling resistance coefficients, derived from coast down tests; g is the gravitational acceleration; ρ_{air} is the air density; A is the vehicle's frontal area; and c_{drag} is the aerodynamic drag coefficient. $F_{x,pwt}$ in (1) is obtained from a first order dynamic model of the powertrain-related longitudinal force response:

$$F_{x,pwt} + \tau\dot{F}_{x,pwt} = F_{x,pwt,drv} - F_{x,corr} \tag{6}$$

where τ is the equivalent time constant.

The prediction model accounts for the electric power consumption, P_{EM}, of the powertrain:

$$P_{EM} = \begin{cases} \dfrac{F_{x,pwt}\dot{x}_h}{\eta_{tot}}, & \text{if } F_{x,pwt} \geq 0 \\ F_{x,pwt}\dot{x}_h\eta_{tot}, & \text{if } F_{x,pwt} < 0 \end{cases} \tag{7}$$

where the 'if' condition discriminates between traction and regeneration; and η_{tot} is the total efficiency of the powertrain system, including the traction inverter, EM, and driveline. In the algorithm, η_{tot} is expressed as a function of the EM speed, n_{EM} ($n_{EM} \approx \dot{x}_h\tau_{tot}/R_w$), and the wheel torque referred to the EM, $T_{w,EM}$ ($T_{w,EM} \approx F_{x,pwt}R_w/\tau_{tot}$, under the simplifying assumption of neglecting the driveline efficiency), i.e., $\eta_{tot} = \eta_{tot}\left(n_{EM}, T_{w,EM}\right)$. Within the NMPC optimization solver, the drivetrain efficiency cannot be inserted in the form of a nonlinear map, but through a function. Since it was verified that in the specific application polynomial functions of n_{EM} and T_{EM} up to the eighth degree still bring relatively large mean efficiency errors, within the internal model $\eta_{tot}(n_{EM}, T_{EM})$ is approximated through a fully connected feedforward perceptron-based NN featuring two hidden 16-neuron layers, where the output $y_{\eta,NN,l}$ of each layer is given by:

$$y_{\eta,NN,l} = f_{\eta NN,l}\left(w_{\eta NN,l}^T h_{\eta NN,l} + b_{\eta NN,l}\right) \tag{8}$$

In (8), the subscript $l = 1, 2$ indicates the hidden layer; $y_{\eta NN,l}$ is the layer output vector; $f_{\eta NN,l}$ is the activation function, which is logarithmic in the first layer, and a swish function in the second layer; $w_{\eta NN,l}$ is the weight matrix; $h_{\eta NN,l}$ is the input vector; and $b_{\eta NN,l}$ is the bias matrix. The resulting NN accuracy is shown in Fig. 1(b) through the efficiency error distribution, having an average value of 1.3%. This represents a substantial improvement with respect to (w.r.t.) the 6.7% obtained with an eighth-order polynomial regression, and the 2.9% of a piecewise linear model, which were evaluated as alternatives in the controller design phase. Hence, the NN surrogate provides superior accuracy while preserving the smoothness of the resulting function, which facilitates the optimization carried out by the NMPC solver. The efficiency-approximating NN is used exclusively within the ADM NMPC prediction model, and thus is independent of the driver neural network digital twins (NN-DTs) described in Section 4. Consequently, the efficiency approximation error does not affect the training or prediction accuracy of the driver models.

For NMPC implementation, the prediction model is expressed in a nonlinear state-space form:

$$\dot{x}(t) = f(x(t), u(t), p(t)) \tag{9}$$

where t is time; and $x(t)$, $u(t)$, and $p(t)$ are the state, control input, and external parameter vectors, defined as

$$x(t) = \left[x_h, \dot{x}_h, F_{x,pwt}, F_{x,corr}\right]^T \tag{10}$$

$$u(t) = \dot{F}_{x,corr} \tag{11}$$

$$p(t) = \left[F_{x,pwt,drv}, \dot{x}_{pred}(N), W\right]^T \tag{12}$$

Setting $\dot{F}_{x,corr}$ as control action, rather than $F_{x,corr}$, facilitates the achievement of comfortable drivability through the reduction of the drivetrain torque oscillations. In $p(t)$, the terms $F_{x,pwt,drv}$ and $\dot{x}_{pred}(N)$, where the latter is the predicted EV speed based on the driver torque demand, enable to consider the preview of the driver's intent within the NMPC, see Section 3.2, while the inclusion of the cost function weight vector W permits the dynamic variation of the control priorities during algorithm operation.

3.2. Nonlinear optimal control problem formulation

At each time step, the NMPC computes the solution of an optimal control problem (OCP) that minimizes the cost function J_{opt} along T_p, subject to (s.t.) a set of constraints:

$$\min_{U} J_{opt}(x_0, u, p) := l_N(x(N), p(N)) + \sum_{k=0}^{N-1} l(x(k), u(k), p(k)) \tag{13}$$

s.t.

$$x(k+1) = f_d(x(k), u(k), p(k)), \quad x(0) = x_0 \tag{13a}$$

$$x_{lb} - \epsilon < x < x_{ub} + \epsilon \tag{13b}$$

$$u_{lb} < u < u_{ub} \tag{13c}$$

$$\epsilon_j > 0 \tag{13d}$$

where $U = [u(0), u(1), \ldots, u(k), \ldots, u(N-1)]$ is the sequence of optimal control inputs; $x_0 = x(0)$ includes the initial value of the states; $l_N(x(N), p(N))$ is the terminal cost, facilitating system stabilization; $l(x(k), u(k), p(k))$ is the stage cost; N is the number of steps of T_p, i.e. $T_p = N\,\Delta T$, with ΔT being the discretization time; k indicates the discretization step along T_p; f_d is the discretized version of f; x_{lb} and x_{ub} are the vectors including the lower and upper boundaries of the elements of x; u_{lb} and u_{ub} define the boundaries of the elements of u; and ϵ is the slack variable vector, whose j-th element is ϵ_j, which is used to define soft constraints.

Instead, $l_N(x(N), p(N))$ and $l(x(k), u(k), p(k))$ are defined as

$$l_N(x(N), p(N)) = \frac{1}{2}\left[w_E(N) J_E^2(N) + w_{F_x}(N) J_{F_x}^2(N) + w_{F_{x,corr}}(N) J_{F_{x,corr}}^2(N) + w_{\dot{x}}(N) J_{\dot{x}}^2(N) + \sum_{j=1}^{4} W_{\epsilon,j}(N) \epsilon_j^2(N) \right] \tag{14a}$$

$$l(x(k), u(k), p(k)) = \frac{1}{2}\left[w_E(k) J_E^2(k) + w_{F_x}(k) J_{F_x}^2(k) + w_{F_{x,corr}}(k) J_{F_{x,corr}}^2(k) + w_{\dot{F}_{x,corr}}(k) J_{\dot{F}_{x,corr}}^2(k) + \sum_{j=1}^{4} W_{\epsilon,j}(k) \epsilon_j^2(k) \right] \tag{14b}$$

where the terms $J_i(N)$, with $i = E, F_x, F_{x,corr}, \dot{x}$, are the normalized terminal cost contributions; the terms $J_m(k)$, with $m = E, F_x, F_{x,corr}, \dot{F}_{x,corr}$, are the normalized stage cost contributions; and $w_i(N)$ and $w_m(k)$ are the respective weights. $W_\epsilon(N)$ and $W_\epsilon(k)$ are the weights of the slack variables in the terminal and stage costs. The normalization is based on the maximum expected value of the respective variable, such that the resulting values are in the [0,1] range. In particular, the individual terms are:

$$J_E(k/N) = \frac{E_d(k/N)}{E_{d,max}} = \frac{\frac{P_{EM}(k/N)}{\dot{x}_h(k/N)N}}{E_{d,max}} \tag{15a}$$

$$J_{F_x}(k/N) = \frac{F_{x,pwt}(k/N) - F_{x,pwt,drv}(k/N)}{F_{x,pwt,max}} \tag{15b}$$

$$J_{F_{x,corr}}(k/N) = \frac{F_{x,corr}(k/N)}{F_{x,corr,max}} \tag{15c}$$

$$J_{\dot{x}}(N) = \frac{\dot{x}_h(N) - \dot{x}_{pred}(N)}{\dot{x}_{max}} \tag{15d}$$

$$J_{\dot{F}_{x,corr}}(k) = \frac{\dot{F}_{x,corr}(k)}{\dot{F}_{x,corr,max}} \tag{15e}$$

where the notation k/N indicates that the specific contribution is used both in l and l_N; (15a) refers to the energy consumption per unit distance, indicated as E_d, where N is included to obtain the average value along T_p throughout the sum of the cost terms; (15b) and (15c) target the reduction of the intrusiveness level of the algorithm, in terms of correction of the expected driver-related powertrain force demand profile, which is computed along T_p starting from the elements $\alpha_{p,pred}(k)$ of $A_{p,pred}$, predicted by the driver emulation NNs, in case of activated driver preview; (15d) ensures that the energy saving is achieved without a significant speed reduction w.r.t. the predicted level, $\dot{x}_{pred}(N)$, at the end of T_p, for the same EV without ADM correction; the $\dot{F}_{x,corr}(k)$ term in (15e) brings a reduction of the drivetrain torque oscillations, and thus desirable drivability. The term $\dot{x}_{pred}(N)$ in (15d) is obtained externally to the NMPC algorithm, through the discrete integration – repeated N times – of the predicted EV acceleration, $\ddot{x}_{pred}(k+1)$, in absence of ADM, based on the following open-loop formulations:

$$\ddot{x}_{pred}(k+1) = \frac{F_{x,pwt,drv}(k+1)}{M_{eq}} + \frac{F_{x,brk}(k) - F_{res}(k)}{M_{eq}}$$

$$\dot{x}_{pred}(k+1) = \dot{x}_{pred}(k) + \ddot{x}_{pred}(k+1)\Delta T \tag{16}$$

where $F_{x,pwt,drv}(k+1)$ is computed through the drivability map from $\alpha_{p,pred}(k+1)$.

In terms of constraints, (13b) is implemented as

$$x_h \le x_{max}, \text{with } x_{max} = x_p - d_{safe} \tag{17a}$$

$$\dot{x}_h \le \dot{x}_{max} \tag{17b}$$

$$F_{x,pwt,min} \le F_{x,pwt} \le F_{x,pwt,max} \tag{17c}$$

$$F_{x,corr,min} \le F_{x,corr} \le F_{x,corr,max} \tag{17d}$$

where (17a) prevents the collision of the host vehicle with a potential preceding one, located at the coordinate x_p, while considering a safety distance $d_{safe} = d_{static} + t_h \dot{x}_h$, with d_{static} being the static safety distance, and t_h the headway time; $\dot{x}_{max}$ accounts for road speed limits, with adjustments to reflect different driving styles (see Section 4); $F_{x,pwt,min}$ and $F_{x,pwt,max}$ ensure that the EM limits (see Fig. 1) are met; and $F_{x,corr,min} < 0$ and $F_{x,corr,max} > 0$ are the lower and upper boundaries on the corrective action. During system operation, these can vary depending on the control priorities, e.g., in case of preceding vehicle interference, the bounds are modified to enable decisive corrections of the driver intent to avoid collisions. $x_p(0)$ is directly measured by the host vehicle, while $x_p(k)$ comes from the numerical integration of $\dot{x}_p(k)$, which is obtained from the radar sensor measurement at $k = 0$, and – similarly to the implementation in [25] – is assumed constant along T_p. This assumption simplifies the prediction model and supports real-time implementation, while remaining consistent with the relatively short prediction horizon of the algorithms of this study. Although highly dynamic events such as emergency braking applications cannot be fully anticipated, the closed-loop radar-based nature of the proposed framework continuously updates the predictions, by using the measured or estimated vehicle states at each time step, according to the receding horizon approach. (13c) assumes the form:

$$\dot{F}_{x,corr,min} \le \dot{F}_{x,corr} \le \dot{F}_{x,corr,max} \tag{18}$$

where $\dot{F}_{x,corr,min}$ and $\dot{F}_{x,corr,max}$ are the lower and upper bounds on $\dot{F}_{x,corr}$. Finally, ϵ is implemented as

$$\epsilon = \left[\epsilon_{x_h}, \epsilon_{\dot{x}_h}, 0, 0\right]^T \tag{19}$$

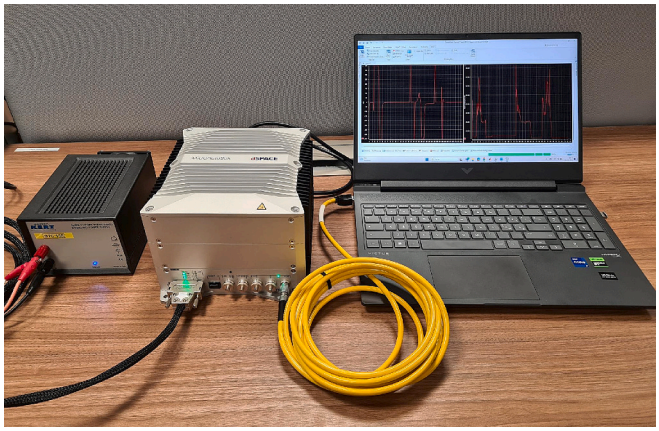


Fig. 3. Real-time NMPC implementation on the considered dSPACE MicroAutoBox III unit.

Table 2
NMPC parameters for dSPACE implementation.

Parameter	Symbol	Value	Unit
Number of prediction steps	N	20	–
Prediction horizon	T_p	2	s
Controller time step	ΔT	100	ms
Mean turnaround time	–	14.8	ms
Peak turnaround time	–	27.4	ms

3.3. Controller implementation

The OCP solution is implicitly (i.e., online) obtained through the acados toolkit [41]. The prediction model is integrated through an implicit Runge-Kutta method. The considered nonlinear programming solver (NLP) is sequential quadratic programming (SQP), with a maximum number of NLP iterations of 3, and an NLP tolerance of 1e-3.

A dSPACE MicroAutoBox III unit (1.4 GHz, 64 Mb flash memory, Fig. 3) was used to verify the real-time capability of the controller with the parameters in Table 2. The mean and peak values – respectively 14.8 and 27.4 ms – of the turnaround time, i.e., the time to generate the control input, are both significantly lower than the controller time step ($\Delta T = 100$ ms), which provides desirable safety margins for real-time implementation. Such values refer to the whole set of ADM blocks in Fig. 2, e.g., including the NN-based driver behavior prediction in Section 4.5.

4. Human driver models and switching algorithm

4.1. Physics-based human driver model

The PB-HDM schematic, inspired by [34], is represented in Fig. 4. Its three main subsystems are:

- The central nervous system (CNS) model and pedal reference generator. CNS consists of the brain and spinal cord, which process visual feedback and information on the road ahead [34,35], make control decisions, and send neural commands to the muscles to contract or relax, based on the separation between the desired and actual vehicle behavior. In the context of driving simulation, CNS is modeled as a path-following or car-following controller [28–33]. In the proposed architecture, where such model is used to generate preview information, the target is to emulate the driver behavior starting from a set of variables that are realistically available on the vehicle, such as: i) the expected speed value $\dot{x}_{ref}$, e.g., based on data from the vehicles ahead; ii) the position and speed of the preceding vehicle; iii) information on the road ahead, e.g., in terms of $\dot{x}_{lim}$, α_{road}, ρ_{road}, and tire-road friction factor μ. The CNS model outputs the total longitudinal tire force request, $F_{x,req,drv}$, which is then converted into reference pedal variables, namely: i) $T_{ref\ acc\ pedal}$, i.e., the reference accelerator pedal torque; and ii) $F_{p,brk}$, i.e., the reference brake pedal force.
- The neuromuscular system (NMS) model, which emulates the dynamics from $T_{ref\ acc\ pedal}$ to the desired angle of the driver foot, α_{musc}, on the accelerator pedal.
- The foot-pedal contact (FPC) model, which converts α_{musc} into the accelerator pedal angle, α_p.

Since the study focuses on powertrain torque control, the NMS and FPC models only consider the accelerator pedal actuation.

4.2. Central nervous system and pedal reference generator

Similarly to [31,32], CNS is modeled through an NMPC, whose prediction model is defined in (20). Although being based on a similarly simplified longitudinal dynamics vehicle model, this NMPC formulation differs from the ADM NMPC described in Section 3.2 in terms of states, control inputs, cost function, parameters, and constraints, since it is used exclusively within the PB-HDM to emulate the human driver behavior and generate the $F_{x,req,drv}$ profile.

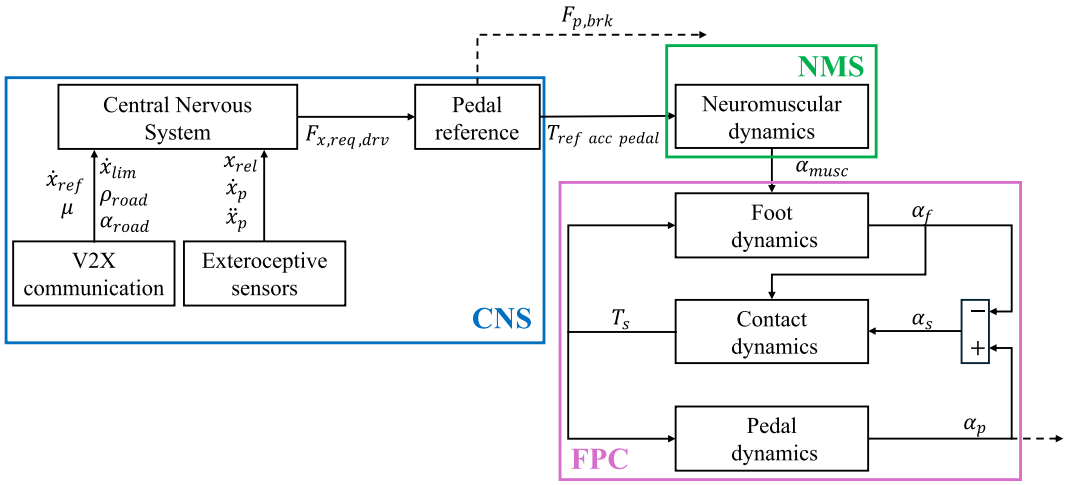


Fig. 4. Simplified schematic of the PB-HDM architecture.

$$\dot{x}_{CNS}(t) = f_{CNS}(x_{CNS}(t), u_{CNS}(t), p_{CNS}(t)) = \begin{bmatrix} \dot{x}_h \\ \frac{F_{x,pwt} + F_{x,brk} - F_{res}}{M_{eq}} \\ \dot{F}_{x,req,drv} \end{bmatrix} \tag{20}$$

where the subscript 'CNS' distinguishes the symbols from those in (13), when these are different; $x_{CNS}(t) = [x_h, \dot{x}_h, F_{x,req,drv}]^T$ is the state vector; $u_{CNS}(t) = \dot{F}_{x,req,drv}$ is the control input; and $p_{CNS}(t) = [\alpha_{road}, \rho_{road}, \dot{x}_{ref}, \eta_{dvl}, W_{CNS}]^T$ is the external parameter vector, providing a preview of the road scenario ahead, since α_{road}, ρ_{road} and $\dot{x}_{ref}$ vary along T_p. For reducing the model mismatch, the expected driveline efficiency profile, η_{dvl}, is externally provided based on the wheel torque and speed profiles. Also the weight vector W_{CNS} can vary along T_p, to enable behavioral flexibility depending on the driving condition.

The OCP has similar structure to the one in (13):

$$\min_{U_{CNS}} J_{opt,CNS}(x_{CNS,0}, u_{CNS}, p_{CNS}) := l_{CNS,N}(x_{CNS}(N), p_{CNS}(N)) + \sum_{k=0}^{N-1} l_{CNS}(x_{CNS}(k), u_{CNS}(k), p_{CNS}(k)) \tag{21}$$

s.t.

$$x_{CNS}(k+1) = f_{d_{CNS}}(x_{CNS}(k), u_{CNS}(k), p_{CNS}(k)), \quad x_{CNS}(0) = x_{CNS,0} \tag{21a}$$

$$x_{lb,CNS} - \epsilon_{CNS} < x_{CNS} < x_{ub,CNS} + \epsilon_{CNS} \tag{21b}$$

$$u_{lb,CNS} < u_{CNS} < u_{ub,CNS} \tag{21c}$$

$$\epsilon_{CNS,j} > 0 \tag{21d}$$

The terminal and stage costs are defined as

$$l_{CNS,N}(x_{CNS}(N), p_{CNS}(N)) = \frac{1}{2}\left[w_{VT}(N) J_{VT}^2(N) + \sum_{j=1}^{3} W_{\epsilon,CNS,j}(N) \epsilon_{CNS,j}^2(N)\right] \tag{22a}$$

$$l_{CNS}(x_{CNS}(k), u_{CNS}(k), p_{CNS}(k)) = \frac{1}{2}\left[w_{VT}(k) J_{VT}^2(k) + w_u(k) J_u^2(k) + \sum_{j=1}^{3} W_{\epsilon,CNS,j}(k) \epsilon_{CNS,j}^2(k)\right] \tag{22b}$$

where J_{VT} is the $\dot{x}_{ref}$ tracking term, included both in $l_{CNS,N}$ and l_{CNS}; J_u penalizes $\dot{F}_{x,req,drv}$; ϵ_{CNS} is the slack variable vector, whose j-th element is indicated as ϵ_{CNSj}; and $w_{VT}(k/N)$, $w_u(k)$ and $W_{\epsilon,CNS}(k/N)$ are the respective weights. In particular, the individual contributions are defined as

$$J_{VT}(k/N) = \frac{\dot{x}_{ref}(k/N) - \dot{x}_h(k/N)}{\dot{x}_{max}} \tag{23a}$$

$$J_u(k) = \frac{\dot{F}_{x,req,drv}(k)}{\dot{F}_{x,req,drv,max}} \tag{23b}$$

On top of including the constraint on x_h defined in (17a), the constraints in (21b) depend on the preview information and driving style:

$$\dot{x}_h \le \dot{x}_{max}, \text{with } \dot{x}_{max} = \min\left(K_1 \dot{x}_{lim}, \dot{x}_{max,a_y}\right) \tag{24a}$$

$$F_{x,req,drv,min} \le F_{x,req,drv} \le F_{x,req,drv,max}, \text{ with } F_{x,req,drv,max} = \min\left(F_{x,pwt,max}, F_{x,\mu,max}\right) \tag{24b}$$

where (24a) considers the minimum between $\dot{x}_{lim}$, corrected by a factor K_1, and the speed, $\dot{x}_{max,a_y}$, corresponding to the maximum admissible lateral acceleration, corrected by a factor F_S, see [42–44]. Both terms depend on the driving style, through K_1 and F_S, which are higher for aggressive drivers, who can decide, for example, to operate the vehicle very close to or also above the speed limits. $\dot{x}_{max,a_y}$ depends on the preview values of ρ_{road} and μ:

$$\dot{x}_{max,a_y} = \sqrt{\frac{F_s \mu g}{\rho_{road}}} \tag{25}$$

In (24b), the upper bound depends on the powertrain traction force capability, $F_{x,pwt,max}$, and the maximum traction force compatible with the available tire-road friction level, $F_{x,\mu,max}$. The lower bound is the maximum achievable braking force.

Outside the NMPC-based CNS model, the reference pedal generator model includes:

- An inverse brake blending algorithm, to obtain the friction braking force contribution $F_{x,brk}$, by prioritizing regenerative braking over friction braking, through a deadband ranging from $-\infty$ to 0:

$$F_{x,brk} = \text{deadband}_{-\infty}^{0}\left(F_{x,req,drv} - F_{x,pwt,reg,min}\right) \tag{26}$$

where $F_{x,pwt,reg,min}$ is the longitudinal tire force contribution associated with the maximum achievable regeneration.

- The inverse static drivability map, which, starting from the saturated powertrain force and current vehicle speed, obtains the reference accelerator pedal position, and thus $T_{ref\ acc\ pedal}$, through the stiffness characteristic of the accelerator pedal.
- The inverse brake-by-wire system map, which permits to obtain $F_{p,brk}$ from $F_{x,brk}$.

4.3. Neuromuscular system model

The NMS model (Fig. 5), based on the adaptation of formulations from [31,32,35], includes: i) the activation dynamics; ii) the Golgi Tendon Organs (GTO) dynamics; iii) the tendon dynamics; and iv) the intrinsic dynamics.

The activation dynamics deal with the muscle fiber recruitment in response to input signals. A sequence of phases is involved: i) signal transmission to the alpha motor neurons in the spine, described by first order dynamics with time constant τ_1; ii) activation and deactivation of the muscle fibers by the motor neurons, corresponding to first order response with time constant τ_2; and iii) conversion from muscular action to an equivalent angular displacement, expressed through the stiffness K_{act}. i)-iii) result in the following transfer function, where s is the Laplace operator:

$$H_{act} = \frac{1}{[1 + \tau_1 s]\,[1 + \tau_2 s]\,K_{act}} \tag{27}$$

The Golgi Tendon Organs [45] inhibit or stimulate muscle movement through the feedback torque T_{GTO}, obtained through:

$$H_{GTO} = \frac{1}{1 + \tau_{GTO} s} K_{GTO} \tag{28}$$

where τ_{GTO} and K_{GTO} are the GTO time constant and stiffness.

The intrinsic dynamics, parametrized through the stiffness and damping coefficient K_{int} and β_{int}, cover the co-contraction of the muscle fibers, according to:

$$H_{int} = \frac{1}{\beta_{int} s + K_{int}} \tag{29}$$

The tendons are described as springs with stiffness K_{tend}, i.e., $H_{tend} = K_{tend}$, generating the torque $T_{ref,NMS}$. The latter is converted into α_{musc} by considering the inverse of the equivalent FPC stiffness resulting from the three springs in series in Fig. 6.

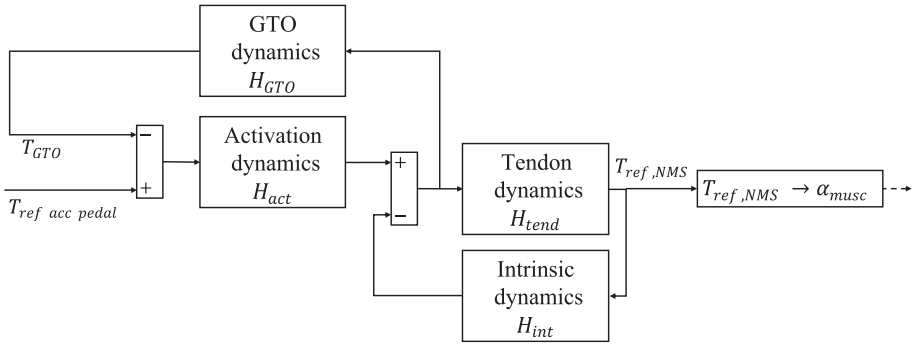


Fig. 5. The NMS model structure.

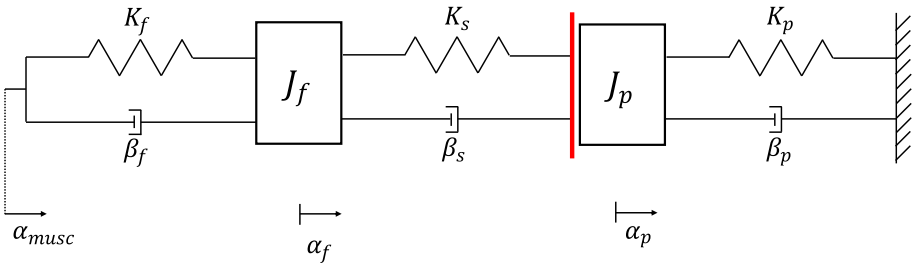


Fig. 6. Schematic of the FPC dynamics model.

4.4. Foot-pedal contact dynamics

The FPC dynamics are modeled through the equivalent mechanical system in Fig. 6, inspired by [34] and [35], where the notations $K_{f/s/p}$ refer to stiffness values, $\beta_{f/s/p}$ to damping coefficients, and $J_{f/s/p}$ to mass moments of inertia, with the subscripts f, s, and p standing for foot, shoe sole, and pedal. The pedal moves only if its preload, $T_{preload}$, is overcome. The preload effect is modeled through a bump stop generating the torque T_{bump}. The governing equations are:

$$J_f\ddot{\alpha}_f = T_{musc} + T_s \tag{30}$$

$$J_p\ddot{\alpha}_p + \beta_p\dot{\alpha}_p + K_p\alpha_p = T_{bump} - T_{preload} - T_s \tag{31}$$

where:

$$T_{musc} = K_f\left[\alpha_{musc} - \alpha_f\right] + \beta_f\left[\dot{\alpha}_{musc} - \dot{\alpha}_f\right] \tag{32}$$

$$T_s = K_s\left[\alpha_p - \alpha_f\right] + \beta_s\left[\dot{\alpha}_p - \dot{\alpha}_f\right] \tag{33}$$

$$T_{bump} = \mathrm{sat}_0^{+\infty}\left(T_{preload} - K_s\alpha_f - \beta_s\dot{\alpha}_f\right) \tag{34}$$

4.5. Neural network digital twinning of human driving behavior

4.5.1. Methodology framework

The computational effort associated with PB-HDM, which is a numerically stiff model, is excessively high for its real-time implementation in the control architecture in Fig. 2. To overcome this limitation while preserving model fidelity, NN-DTs of the human driver are introduced. PB-HDM is therefore run offline to generate representative training data, whereas NN-DTs are deployed online to emulate and forecast the individual driver behavior in real-time. To this purpose, two complementary architectures have been developed: a) one using a couple of FFNN models, referred to as FFNN-DTs in the remainder; and b) one based on an LSTM implementation, referred to as LSTM-DT. In the proposed framework, LSTM-DT operates as an adaptive digital twin through continuous online learning from driver-specific data, whereas the pre-trained FFNN-DT implementations are selected online based on driver-dependent KPIs, thereby enabling some form of driver-specific representation. More specifically:

- The FFNNs are designed with a dual purpose: i) to emulate PB-HDM providing the driver inputs – α_p and $F_{p,brk}$ – at the current time instant, within the plant model; and ii) to generate the predicted future accelerator pedal trajectory, $A_{p,pred,FFNN}$, along T_p, for preview-based NMPC operation. Since driving behavior significantly depends on the individual driver characteristics, e.g., age, sex, background, and specific circumstances [46], the preliminary proof-of-concept implementation uses two distinct offline-trained FFNN-DTs, corresponding to generically ‘Soft’ and ‘Aggressive’ driving behaviors. The decision to train the FFNNs exclusively offline is motivated by robustness and functional safety considerations, i.e., the FFNN-DTs act as stable and certifiable baseline models under predefined behavioral classes. The low granularity of the FFNNs does not represent a significant limitation during control operation, given the automated selection of the individualized LSTM-DT when this is more accurate.
- LSTM-DT generates the respective individualized accelerator pedal trajectory, $A_{p,pred,LSTM}$, for control purposes. Unlike the FFNNs, following a preliminary offline training, the LSTM is continuously trained online using data generated by the current human driver. This NN choice is motivated by the intrinsic capability of LSTM networks to capture temporal dependencies and long-term correlations in sequential data, which is fundamental for accurately modeling the evolution of driver behavior over time. The online adaptation enables the LSTM-DT to progressively adjust to intermediate or evolving driving styles that cannot be strictly categorized as either ‘Soft’ or ‘Aggressive’, thus enhancing the personalization and predictive accuracy of the FFNN-DTs.

As a result, the combination of offline-trained FFNN-DTs and an online-adaptive LSTM-DT – together with the respective switching mechanisms – results in a hybrid modeling framework that balances real-time feasibility, robustness, and behavioral adaptability.

4.5.2. FFNN-DT formulation

Each layer of the implemented FFNNs follows the formulation in (8), with the resulting FFNN-DTs providing the following output vector:

$$y_{drv,FFNN} = \left[\alpha_{p,FFNN}, A_{p,pred,FFNN}, F_{p,brk,FFNN}\right]^T \tag{35}$$

where $\alpha_{p,FFNN}$ is the estimated pedal position at the current time step; $A_{p,pred,FFNN}$ includes the predicted α_p sequence along T_p, i.e., $A_{p,pred,FFNN} = \left[\alpha_{p,pred,FFNN}(1), \alpha_{p,pred,FFNN}(2), \ldots, \alpha_{p,pred,FFNN}(k), \ldots, \alpha_{p,pred,FFNN}(N)\right]$; and $F_{p,brk,FFNN}$ is the expected brake pedal force at the current time step. The NN input vector, $h_{drv,FFNN}$, is:

$$h_{drv,FFNN} = \left[\dot{X}_{lim}, \mathrm{P}_{road}, X_{rel}, \dot{X}_p, \ddot{X}_p, \dot{X}_h\right]^T \tag{36}$$

where the capital letter notations indicate vectors with time sequences of the respective variables, which have been previously defined with the corresponding lower-case notations. Namely, $\dot{X}_{lim}$ and P_{road} are 24-element vectors, including the information for the previous three time steps, the current step, and the preview values along T_p, while X_{rel}, $\dot{X}_p$, $\ddot{X}_p$ and $\dot{X}_h$ are 4-element vectors, including the current value and three preceding values of the respective variable. The FFNN-DT hyperparameters are reported in Table 3.

4.5.3. LSTM-DT formulation and online adaptation

The LSTM-based RNN structure consists of two stacked LSTM layers, followed by an embedding layer, a LeakyReLU activation layer, a fully connected layer, and a tanh output activation function. The input vector, indicated as x_{LSTM}, is:

$$x_{LSTM} = \left[\dot{X}_{lim,LSTM}, \mathrm{P}_{road,LSTM}, x_{rel}, \dot{x}_p, \ddot{x}_p, \dot{X}_h, A_p\right]^T \tag{37}$$

where $\dot{X}_{lim,LSTM}$ and $\mathrm{P}_{road,LSTM}$ are 5-element vectors, including the information for the current step, and four preview values along T_p; $x_{rel}, \dot{x}_p, \ddot{x}_p$ are the values at the current step of the relative distance from a preceding vehicle, the preceding vehicle speed and acceleration; $\dot{X}_h$ and A_p are 4-element vectors, including the current value and three previous values of the respective variable.

Each LSTM layer is described by the following set of equations, with (38a) formulating the forget gate; (38b) the input gate; (38c) the cell candidate; (38d) the cell state update; (38e) the output gate; and (38f) the hidden state output:

$$f_{LSTM,t,l} = \sigma\left(W_{LSTM,f,l} x_{LSTM,l} + U_{LSTM,f,l} h_{LSTM,t-1,l} + b_{LSTM,f,l}\right) \tag{38a}$$

$$i_{LSTM,t,l} = \sigma\left(W_{LSTM,i,l} x_{LSTM,l} + U_{LSTM,i,l} h_{LSTM,t-1,l} + b_{LSTM,i,l}\right) \tag{38b}$$

$$\tilde{c}_{LSTM,t,l} = \tanh\left(W_{LSTM,c,l} x_{LSTM,l} + U_{LSTM,c,l} h_{LSTM,t-1,l} + b_{LSTM,c,l}\right) \tag{38c}$$

$$c_{LSTM,t,l} = f_{LSTM,t,l} \odot c_{LSTM,t-1,l} + i_{LSTM,t,l} \odot \tilde{c}_{LSTM,t,l} \tag{38d}$$

$$o_{LSTM,t,l} = \sigma\left(W_{LSTM,o,l} x_{LSTM,l} + U_{LSTM,o,l} h_{LSTM,t-1,l} + b_{LSTM,o,l}\right) \tag{38e}$$

$$h_{LSTM,t,l} = o_{LSTM,t,l} \odot \tanh\left(c_{LSTM,t,l}\right) \tag{38f}$$

where the subscript l, with $l = 1,2$, indicates the LSTM layer; $x_{LSTM,l}$ is the input vector of the layer ($x_{LSTM,1} = x_{LSTM}$ and $x_{LSTM,2} = h_{LSTM,t,1}$); $\sigma(\cdot)$ and $\tanh(\cdot)$ are the sigmoid and hyperbolic tangent activation functions; $W_{LSTM,z,l}$, $U_{LSTM,z,l}$ and $b_{LSTM,z,l}$, with $z = f, i, c, o$, are the corresponding weights and biases; and $\odot$ indicates the element-wise product. The output of the second LSTM layer, $h_{LSTM,t,2}$, is then processed by an embedding layer, a LeakyReLU activation layer, a fully connected layer and a tanh output activation function, resulting into the output vector, $y_{drv,LSTM} = A_{pred}$:

$$y_{drv,LSTM} = \tanh\left(W_{LSTM,FC}\,\mathrm{LeakyReLU}\left(W_{LSTM,emb} h_{LSTM,t,2} + b_{LSTM,emb}\right) + b_{LSTM,FC}\right) \tag{39}$$

where $W_{LSTM,FC}$, $W_{LSTM,emb}$, $b_{LSTM,FC}$ and $b_{LSTM,emb}$ are the weights and biases of the embedding and fully connected layers.

The LSTM-DT is trained in two phases: i) an offline pre-training aimed at obtaining a suitable initialization of the network parameters; and ii) an online update to adapt to changes in driver behavior. The LSTM hyperparameters are reported in Table 4. During online operation, the loss minimized by the network is:

$$\mathscr{L}_t = \frac{1}{2NB} \sum_{b=1}^{B} \sum_{k=1}^{N} \left[y_{drv,LSTM,ref,k,b} - y_{drv,LSTM,k,b}\right]^2 \tag{40}$$

where N is the number of prediction steps; B is the batch size; and $y_{drv,LSTM,ref,k,b}$ is the current value of α_p, shifted in time to be comparable with the corresponding predicted value.

The gradients of the loss are propagated backward through the stacked LSTM layers, the embedding layer, and the fully connected layer, by means of backpropagation through time for the LSTM layers and standard backpropagation for the remaining layers. The network parameters θ_t, which include the weights and biases of the stacked LSTM layers, the embedding layer, and the fully connected layer, are updated using gradient descent with momentum and L2 regularization:

$$\theta_{t+1} = \theta_t - \eta_{LSTM}\left[\frac{\partial \mathscr{L}_t}{\partial \theta_t} + \lambda \theta_t\right] \tag{41}$$

where η_{LSTM} is the learning rate; $\frac{\partial \mathscr{L}_t}{\partial \theta_t}$ is the gradient of the loss w.r.t. each network parameter; and λ is the L2 regularization factor. This online training algorithm allows the network to continuously refine its parameters based on the observed performance. In the online adaptation phase, the learning rate is lower w.r.t. the offline training phase (from 1e-3 to 1e-4). Since the online updates rely on limited sequential data, reducing the learning rate improves the stability of the parameter updates, preventing abrupt variations while allowing gradual adaptation to changes in driver behavior.

Table 3
Human driver FFNN hyperparameters.

Hyperparameter	Value
Number of hidden layers	3
Number of neurons in the first and third layers	32
Number of neurons in the second layer	64
Activation function of the hidden layers	Swish
Activation function of the output layer	Tanh
Mini batch size	6400
Maximum number of epochs	800
Initial learning rate	1e-3
Optimization algorithm	Adam
Loss function	MSE

Table 4
Human driver LSTM hyperparameters.

Training	Offline	Online
Hyperparameter	Value	
Number of hidden LSTM layers	2	2
Number of embedding layer	1	1
Number of neurons in the hidden LSTM layers	16	16
Number of neurons in the embedding layers	8	8
Mini batch size	64	10
Maximum number of epochs	500	–
Learning rate	1e-3	1e-4
L2 regularization factor	1e-5	1e-5
Optimization algorithm	Adam	Adam
Loss function	MSE	MSE

Table 5
KPIs for the NN switching algorithm.

KPI	Formula	Condition
$v_{lim,ratio}$	$\frac{\overline{\dot{x}}_h}{\overline{\dot{x}}_{lim}}$	$> ds_{v_{lim,ratio,thr}} \rightarrow ds_{v_{lim,ratio}} = +1$ $\leq ds_{v_{lim,ratio,thr}} \rightarrow ds_{v_{lim,ratio}} = -1$
$a_{y,ratio}$	$\frac{\overline{a}_y}{\overline{a}_{y,lim}}$	$> ds_{a_{y,ratio,thr}} \rightarrow ds_{a_{y,ratio}} = +1$ $\leq ds_{a_{y,ratio,thr}} \rightarrow ds_{a_{y,ratio}} = -1$
$\alpha_{p,ratio}$	$\frac{\overline{\alpha}_p}{\overline{\alpha}_{p,av}}$	$> ds_{\alpha_{p,ratio,thr}} \rightarrow ds_{\alpha_{p,ratio}} = +1$ $\leq ds_{\alpha_{p,ratio,thr}} \rightarrow ds_{\alpha_{p,ratio}} = -1$
$\dot{\alpha}_{p,av}$	$\overline{\lvert\dot{\alpha}_p\rvert}$	$> ds_{\dot{\alpha}_{p,av,thr}} \rightarrow ds_{\dot{\alpha}_{p,av}} = +1$ $\leq ds_{\dot{\alpha}_{p,av,thr}} \rightarrow ds_{\dot{\alpha}_{p,av}} = -1$
$x_{rel,ratio}$	$\frac{\overline{x}_{rel}}{\overline{x}_{rel,thr}}$	$< ds_{x_{rel,ratio,thr}} \rightarrow ds_{x_{rel,ratio}} = +1$ $\geq ds_{x_{rel,ratio,thr}} \rightarrow ds_{x_{rel,ratio}} = -1$

4.5.4. Training scenarios and preliminary validation through a driving simulator

For PB-HDM calibration and NN-DT training, two different driving style have been considered, corresponding to generically 'Soft' and 'Aggressive' driving behaviors. For example, as discussed in [46–49], w.r.t. to its soft counterpart, an aggressive driver tends to: 1) abruptly vary the pedal commands after speed limit variations, and also to exceed the limits; 2) leave lower safety distances w.r.t. preceding vehicles; and 3) operate the vehicle at larger lateral accelerations on curvy roads. The driving style is identified depending on conditions determined through dedicated KPIs, see Table 5.

The identified KPIs are: i) $v_{lim,ratio}$, i.e., the ratio between the average vehicle speed, $\overline{\dot{x}}_h$, and the average speed limit, $\overline{\dot{x}}_{lim}$, along the considered time interval; ii) $a_{y,ratio}$, i.e., the ratio of the average lateral acceleration in the cornering phases, $\overline{a}_y$, to the corresponding limit value, $\overline{a}_{y,lim}$, based on the available tire-road friction level; iii) $\alpha_{p,ratio}$, i.e., the ratio of the average accelerator pedal position of the considered driver, $\overline{\alpha}_p$, to the one, $\overline{\alpha}_{p,av}$, of the average driver archetype, corresponding to a pedal input that is the arithmetic average between the one of the soft and aggressive driver models at each time step; iv) $\dot{\alpha}_{p,av}$, i.e., the time-normalized average of $\lvert\dot{\alpha}_p\rvert$, computed only when $\lvert\dot{\alpha}_p\rvert > \dot{\alpha}_{p,thr}$, where $\dot{\alpha}_{p,thr}$ is a constant threshold; and v) $x_{rel,ratio}$, i.e., the ratio of the average relative distance, $\overline{x}_{rel}$, to its respective average safety threshold value, $\overline{x}_{rel,thr}$, in presence of a reasonably close preceding vehicle. To distinguish between aggressive and soft driving, each KPI corresponds to a

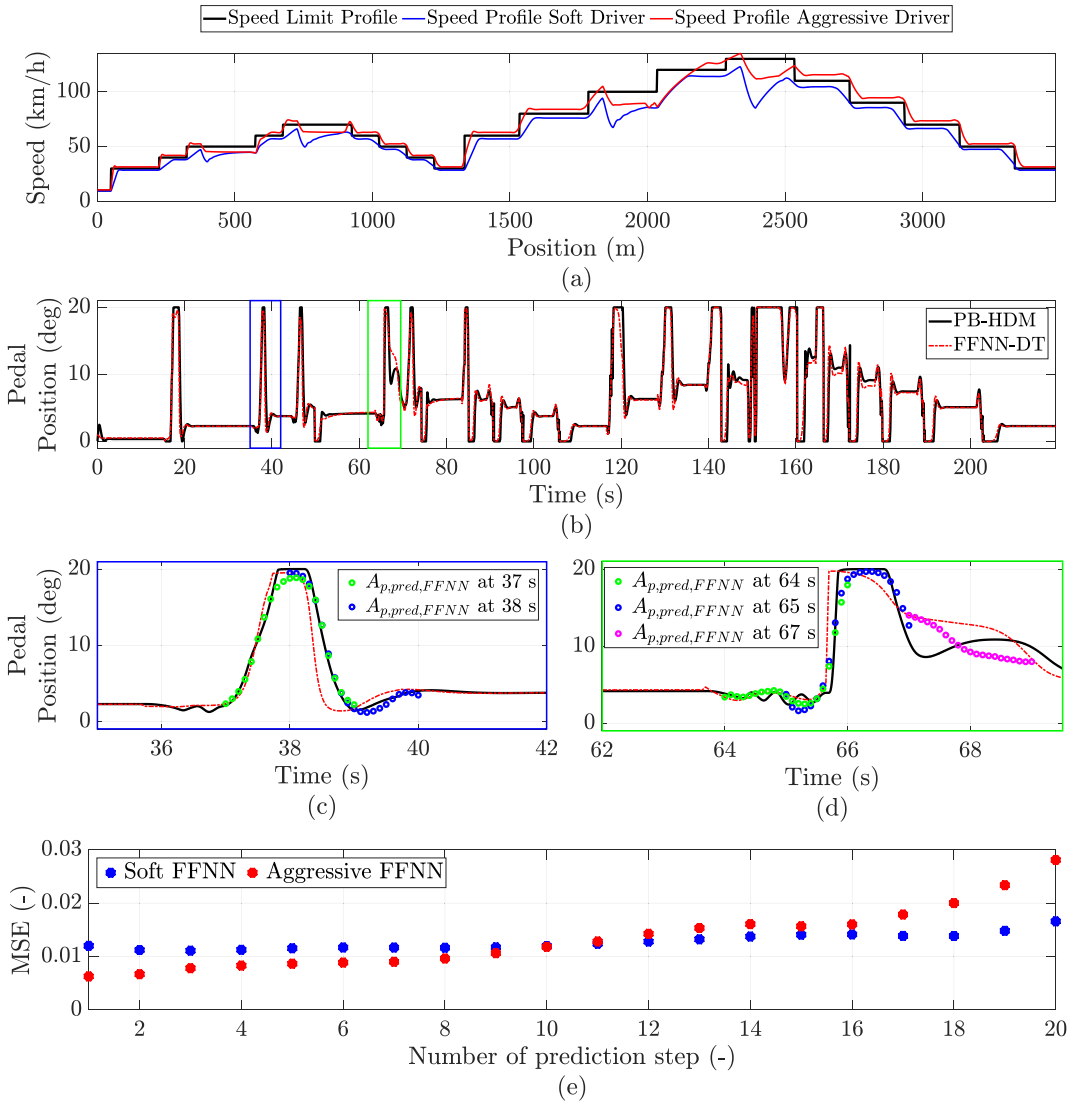


Fig. 7. Example of NN-DT training and validation results: (a) Considered speed limit profile, and actual speed profiles along the traveled distance, for the soft and aggressive driving styles, generated through PB-HDM; (b) Time profiles of the accelerator pedal position for PB-HDM and NN-DT (for the latter, the plot reports $\alpha_{p,FFNN}$), for the aggressive driving style; (c), (d) Zoomed views of relevant sections of the graph in (b), with the addition of the discretized $A_{p,pred,FFNN}$ points at $t = 37$, 38, 64, 65, and 67 s, highlighting the prediction capability of FFNN-DT along T_p; and (e) MSE between the normalized pedal position from PB-HDM, $\alpha_{p,PB-HDM}$, and the corresponding predicted elements of $A_{p,pred,FFNN}$, for the soft and aggressive driving styles, as a function of the prediction step number.

Boolean value, ds_q (with $q = v_{lim,ratio}$, $a_{y,ratio}$, …, $x_{rel,ratio}$). For $v_{lim,ratio}$, $a_{y,ratio}$, $a_{p,ratio}$, and $\dot{\alpha}_p$, a value above the threshold $ds_{q,thr}$ means aggressive driving from the viewpoint of that indicator, and is assigned $ds_q = +1$; vice versa, for $x_{rel,ratio}$, aggressive driving corresponds to a value below $ds_{q,thr}$. Soft driving conditions from each KPI perspective are assigned ds_q = -1. The thresholds were determined based on the KPIs computed from the driving data collected during experimental tests conducted with a sample of human drivers on the static driving simulator, which were used to characterize the driving styles. Then, the driving style indicator, DS, is obtained through majority voting:

$$DS = \sum_{\forall q} ds_q \rightarrow \begin{cases} DS > 0: & \text{aggressive} \\ DS \leq 0: & \text{soft} \end{cases} \tag{42}$$

Two PB-HDM calibrations were set, corresponding to the average behavior of the identified aggressive and soft driving styles. The resulting PB-HDM tunings were used to generate datasets for offline NN training. This approach provides greater flexibility in scenario emulation w.r.t. generating data only through the static simulator. Two FFNN-DTs were trained separately, each corresponding to one of the two driving styles. Conversely, the datasets associated with both driving styles were combined and used for the preliminary offline training of the LSTM-DT.

Fig. 7(a) reports an extract of training data for the two driver archetypes. The scenario includes several speed limit variations on a straight flat road. In the first part of the test, a preceding vehicle is present, with a variable speed that tends to be lower than the one of the host vehicle. This leads to braking actions of the host vehicle to prevent collisions at ~500 m, ~800 m, ~1900 m and ~2400 m. Once the preceding vehicle re-accelerates and largely exceeds the safety distance, the human driver of the host vehicle can resume tracking the set speed for the defined speed limit profile. This setup is intentionally designed to reproduce frequent acceleration and deceleration phases, which are typical of both urban and extra-urban driving conditions.

For the aggressive driver calibration, Fig. 7(b) reports the time profiles of α_p from PB-HDM, and $\alpha_{p,FFNN}$ for the corresponding FFNN-DT. Moreover, Fig. 7(c) and (d) highlight the prediction capability of FFNN-DT along T_p, by including the entire $A_{p,pred,FFNN}$ sequence at t = 37, 38, 64, 65, and 67 s, see the markers in the plots, which are overlapped to the profiles in Fig. 7(b). Fig. 7(e) shows the prediction error – expressed as the MSE between the normalized current pedal input $\alpha_{p,PB-HDM}$ and the corresponding $A_{p,pred,FFNN}$ elements – as a function of the prediction step, for both FFNN-DT calibrations. The prediction accuracy is high in the short-term portion of the horizon, which is the most relevant region for the intended control application. Moreover, the increasing trend – with rather low MSE values across the horizon – does not represent a limitation for the intended NMPC application. In fact, NMPC operates according to the receding horizon principle: at each sampling instant, an optimal control sequence is computed over the entire prediction horizon, but only the first control action is applied to the system. At the next sampling instant, the optimization problem is solved again, starting from updated system measurements. Therefore, while the accuracy levels of the preview and related prediction naturally decrease for longer horizons, the NMPC structure inherently mitigates the impact of these errors on the closed-loop performance.

The driving behavior analyses were supported by experiments covering a limited sample of drivers (a comprehensive driver behavior classification is not the focus of this research), on the static driving simulator of Politecnico di Torino, see its setup in Fig. 8. The experimental campaign involved 22 licensed drivers (17 males and 5 females) aged between 23 and 63 years old (with a mean age of 30 years), with driving experience ranging from 2 to 45 years. This variability allows the dataset to capture a sufficiently broad range of behavioral responses. None of the participants reported extensive prior experience with driving simulators.

The static driving simulator is a fixed-base driver-in-the-loop platform integrating real steering and pedal hardware with a virtual vehicle dynamics model implemented in IPG CarMaker. The human-machine interface includes a force-feedback steering wheel actuator and a sensorized accelerator and brake pedal unit. Driver inputs are acquired, and transmitted via CAN communication to a real-time machine (Xpack 4), which executes the vehicle dynamics simulation with a fixed integration time step of 1 ms. The simulated vehicle motion is visualized on a dedicated host PC, ensuring real-time interaction between driver inputs, vehicle dynamics, and haptic feedback under controlled and repeatable experimental conditions.

The driving simulator tests allowed:

- To verify the relevance of the driving style KPIs in Table 5, together with the quantification of the respective threshold values.
- To generate experimental data to calibrate and validate the PB-HDM and FFNN versions corresponding to the soft and aggressive driving behaviors.
- To generate experimental data to validate the predictions obtained through the online trained LSTM-DT.

Fig. 9 refers to a scenario with the same speed limit profile as in Fig. 7, dealt with by drivers with aggressive (subplot (a)) and soft (subplots (b) and (c)) driving styles, but without the interference from the preceding vehicle. After a familiarization phase with the experimental setup, each of the 22 human subjects carried out two tests, one per driving style. To this purpose, the same drivers were given instructions to behave more or less aggressively, hence the different accelerator pedal position profiles in the subplots. Although each driver has different reactions to speed limit variations, the overall response dynamics display a good match of PB-HDM and FFNN-DT with the



Fig. 8. The static driving simulator setup at the Politecnico di Torino.

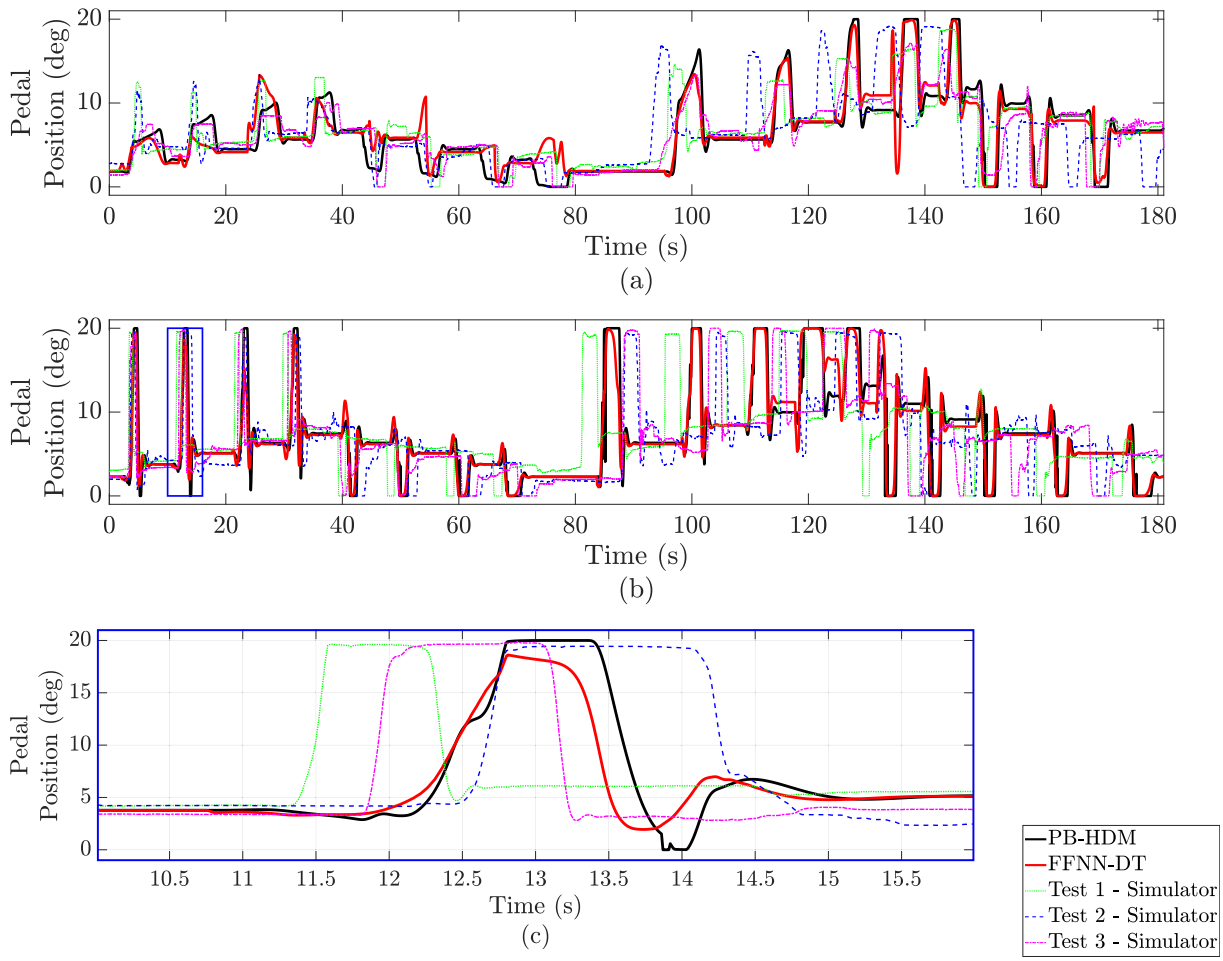


Fig. 9. Example of experimental validation results of the driver models. Comparison of the time profiles of the accelerator pedal position from the driving simulator used by different human drivers (each of them explicitly asked to repeat the test while driving in a relaxed or aggressive manner), PB-HDM, and FFNN-DT (for which the reported profiles correspond to $\alpha_{p,FFNN}$): (a) Soft driving style; (b) Aggressive driving style; and (c) Zoomed view of a relevant section of (b).

driving simulator tests. Among the 22 experiments per driving style, only 3 are included in Fig. 9, for ease of readability. To quantitively validate PB-HDM and FFNN-DT w.r.t the measurements, the dynamic time warping (DTW) algorithm [50] is used to align the time profiles before comparing them. For each driving behavior, DTW is applied to the $\alpha_p(t)$ sequences of PB-HDM, FFNN-DT ($\alpha_{p,FFNN}(t)$), and the individual experiments (exp1, …, exp22), to match the respective time profiles with the instantaneous average pedal position, $\overline{\alpha}_{p,exp}$, from the 22 driving simulator experiments, which has been considered as reference. The aligned time profiles through DTW are indicated as $\alpha_{p,r,DTW}$, with r = PB-HDM, FFNN-DT, exp1, …, exp22. The root mean square value $\Delta\alpha_{p,r,DTW}$ of the residual is obtained as indicator of the signal fidelity level:

$$\Delta\alpha_{p,r,DTW} = \mathrm{RMS}\left(\overline{\alpha}_{p,exp} - \alpha_{p,r,DTW}\right) \tag{43}$$

The rows of Table 6 report: i) $\overline{\Delta\alpha}_{p,exp,DTW}$, i.e., the average of the $\Delta\alpha_{p,r,DTW}$ values for the 22 experimental tests; ii) the range of variation of $\Delta\alpha_{p,r,DTW}$, for the individual experiments, i.e., $[\Delta\alpha_{p,exp,DTW,\min} \div \Delta\alpha_{p,exp,DTW,\max}]$; iii) $\Delta\alpha_{p,PB-HDM,DTW}$; and iv) $\Delta\alpha_{p,NN-DT,DTW}$. For both driving styles, PB-HDM and FFNN-DT provide driver model fidelity values that are well within the ranges in ii), and are aligned to or lower than $\overline{\Delta\alpha}_{p,exp,DTW}$.

Table 6
Validation of the PB-HDM and FFNN-DT models.

Parameter	Aggressive	Soft	Unit
$\overline{\Delta\alpha}_{p,exp,DTW}$	2.65	1.59	deg
$[\Delta\alpha_{p,r,DTW,\min} \div \Delta\alpha_{p,r,DTW,\max}]$	[1.78÷3.71]	[1.06÷2.46]	deg
$\Delta\alpha_{p,PB-HDM,DTW}$	2.99	2.24	deg
$\Delta\alpha_{p,FFNN-DT,DTW}$	3.08	2.21	deg

In terms of computational load, the benefit of replacing PB-HDM with FFNN-DT is very significant. In fact, on a personal computer with 16 GB RAM, a 1.8 GHz Intel(R) Core(TM) 7 150U processor, and a 1 TB solid-state drive, the implemented NN-DTs are characterized by an average simulation time that is 45 times lower than for PB-HDM. It was also verified that the latter would not have enabled the real-time implementation of the ADM architecture on the dSPACE MicroAutoBox III unit, which – instead – was possible with the presented FFNN-DTs, see Section 3.3.

The same 22 drivers performed further experimental tests on the static driving simulator of Fig. 8 exploring a more complex driving scenario, with a ~3.6 km length, including both urban and suburban sections, with complete freedom to drive according to their own preferences. As shown in Fig. 10(a), the route integrates several speed limit changes, varying road curvatures, and a signalized intersection in the urban area, requiring drivers to continuously adapt their speed. Traffic interactions are introduced through slower preceding vehicles that induce realistic braking and car-following behaviors. These interactions occur in different contexts, namely in the 70 km/h speed-limit section (for X ~800 m) and in the 90 km/h section. By combining infrastructure-related factors (speed limits, curvature, traffic lights) with interactions with surrounding traffic, the scenario exposes drivers to heterogeneous

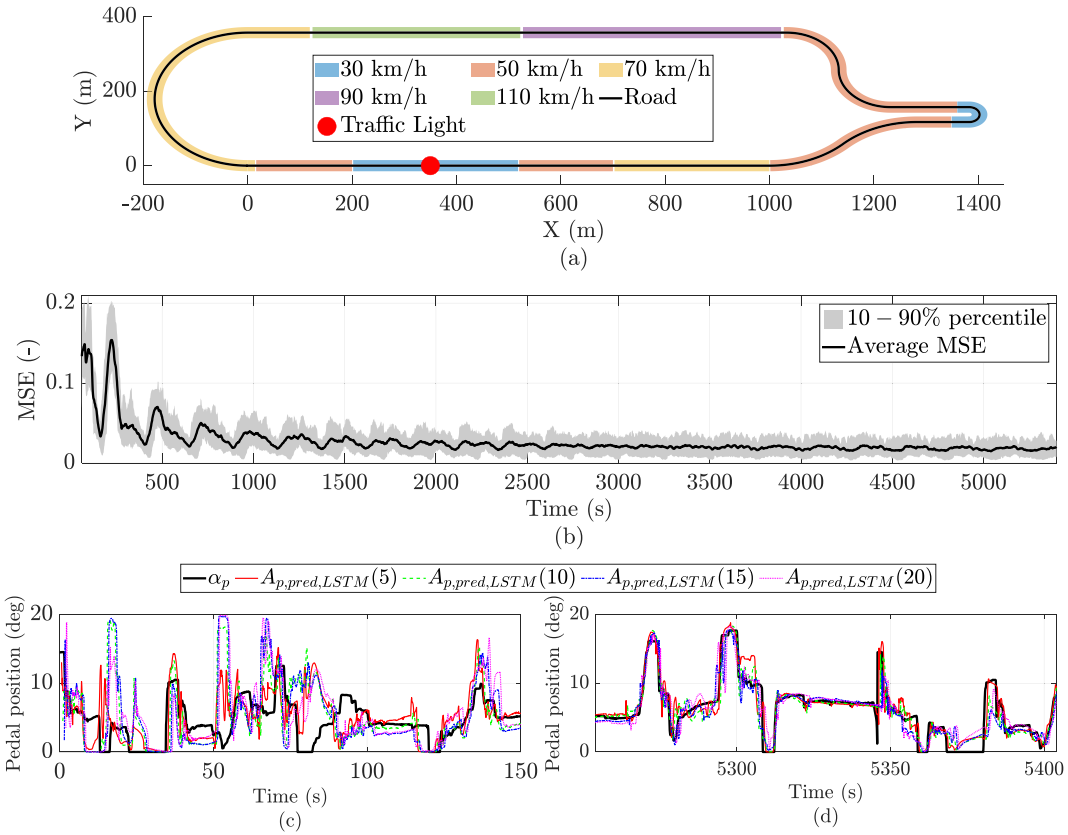


Fig. 10. Experimental tests for validating the online trained LSTM-DT. (a) Driving scenario, including urban and extra-urban sections, with associated speed limits, traffic light, road curvatures, and dynamic traffic interactions; (b) Average MSE of all experimental tests smoothened through a moving average over a 60 s time frame; and (c) and (d) Comparison between the experimental pedal input profile of one of the 22 drivers along with the predicted values from $A_{p,pred,LSTM}$ at 0.5, 1, 1.5 and 2 s, at the beginning and at the end of the online training procedure.

and dynamically evolving driving conditions, enabling the collection of driver behavior data under more representative and realistic situations. To evaluate the effectiveness of the online training procedure, the experimental data for a single circuit lap have been concatenated to emulate multiple track laps, and then provided to the LSTM-DT. Fig. 10 (b) plots the resulting MSE time profile, obtained from the comparison of the normalized experimental accelerator pedal input with the corresponding predicted values $A_{p,pred,LSTM}$. To reduce high-frequency fluctuations, the MSE is smoothed with a moving average over a 60 s temporal window. For an example of experimental test, subplots (c-d) show the time profile of the actual pedal input of the human driver along with the four prediction components of $A_{p,pred,LSTM}$ at 0.5, 1, 1.5 and 2 s, appropriately shifted in time. The comparison of the results at the beginning and the end of the online training process confirm the LSTM-DT adaptation benefits.

4.5.5. Switching logic

The objective of the switching logic in Fig. 2 is to determine which NN-DT provides the most accurate forecast of the driver intent, and consequently select the corresponding $A_{p,pred}$. Initially, the driving style identifier block selects the more appropriate offline-calibrated FFNN-DT based on the KPIs in Table 5. Such selection is performed among the stored behavioral alternatives, i.e., 'Soft' or 'Aggressive'. The selected FFNN-DT generates the predicted driver intent, denoted as $A_{p,pred,FFNN}$. In parallel, the LSTM-DT – continuously trained online during vehicle operation – produces its own prediction vector, $A_{p,pred,LSTM}$. The prediction accuracy of both NN-DTs is evaluated online through the MSE, obtained by comparing α_p with the corresponding predicted values from $A_{p,pred,FFNN}$ and $A_{p,pred,LSTM}$. To prevent chattering, which would lead to oscillatory switching and abrupt $A_{p,pred}$ variations when the alternative NN-DTs produce similar prediction accuracy, two mitigation measures are introduced:

- The prediction losses – MSE_{FFNN} and MSE_{LSTM} – are computed as moving-average values of the MSE over 60 s, to reduce the influence of short-term fluctuations in the prediction error, and obtain a more robust estimate of the relative performance of the predictors.
- A hysteresis-based switching condition varies the NN-DT selection only when the difference between the prediction losses of the two models exceeds a predefined threshold δ. This threshold-based criterion prevents unnecessary switching when the prediction errors are comparable, thereby improving the stability of the supervisory selection mechanism:

$$A_{p,pred} = \begin{cases} A_{p,pred,FFNN}, & \text{if } MSE_{FFNN} + \delta < MSE_{LSTM}, \\ A_{p,pred,LSTM}, & \text{if } MSE_{LSTM} + \delta < MSE_{FFNN}, \\ \text{previous selection}, & \text{otherwise.} \end{cases} \tag{44}$$

Fig. 11 is an example of the NN switching algorithm operation. Since the KPI computation is based on a time frame of a few minutes (240 s in the example in the figure), and to clearly highlight the benefits of the online adaptation of the LSTM-DT, a relatively long scenario is needed. To this purpose, the Fig. 7 scenario has been repeated 17 times. During the simulation, the driver model calibration within the plant model has been forced to abruptly vary three times, see the 'Actual *DS*' line in Fig. 11(a). The first transition corresponds to a change from soft ($DS = -1$) to aggressive ($DS = 1$) driving style; the second transition from aggressive to soft; and the third transition brings a shift from soft to an intermediate ($DS = 0$) behavior, i.e., a driving style between the soft and aggressive archetypes. Fig. 11(a) also reports the *DS* value from the driving style identifier in Fig. 2 (see the 'Identified *DS*' line), where the

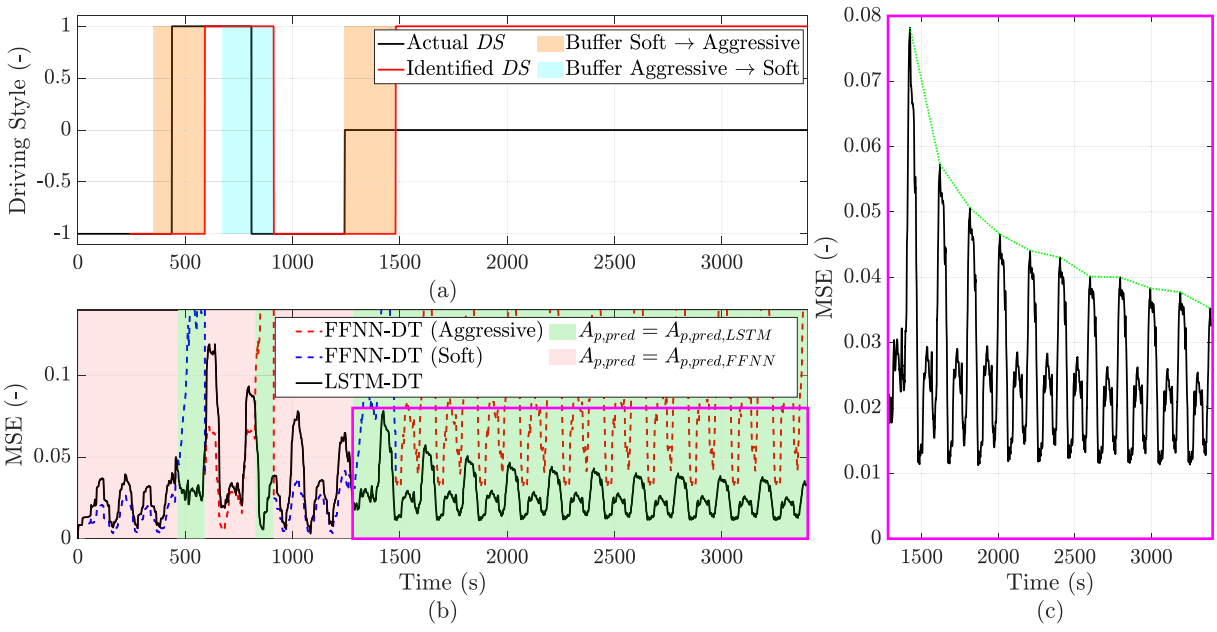


Fig. 11. Example of NN switching algorithm operation: (a) *DS* profile for the actual driver in the plant model ('Actual *DS*'), *DS* profile identified by the KPI-based voting algorithm of the driving style identifier ('Identified *DS*'), and time ranges corresponding to the data buffers used by the switching algorithm to determine the transitions; (b) Prediction errors – expressed by the MSE – between the normalized actual pedal input and the corresponding values predicted by the LSTM-DT (solid line) and the FFNN-DT (dashed line) selected by the driving style identifier; and (c) Zoomed view of the MSE profile for the LSTM-DT, highlighting its reduction as the online training progresses.

time needed for the identification of the varied driving style corresponds to the offset between the 'Actual *DS*' and'Identified *DS*' profiles. The colored areas highlight the time frames corresponding to the data buffers that are used by the algorithm to decide the transitions. When the driver exhibits intermediate behavior ($DS = 0$), the identified driving style is classified as aggressive. This is a direct consequence of the offline-trained FFNN-DTs being available only for the two discrete archetypes (soft and aggressive). As expected, this transition exhibits the largest identification delay, since the actual driving style does not exactly match either of the offline-trained classes.

Fig. 11(b) reports the prediction losses of the LSTM-DT and the FFNN-DT option currently selected by the driving style identifier. During the initial phases of the simulation, i.e., approximately in the first 500 s, the actual and identified driving styles correspond to the soft archetype. In this condition, the driver behavior matches one of the offline-trained FFNN-DTs, resulting in $MSE_{FFNN} < MSE_{LSTM}$, and therefore $A_{p,pred} = A_{p,pred,FFNN}$. When the driver transitions from soft to aggressive behavior, the KPI-based identification requires a finite time window to detect the change. During this transient phase, the FFNN-DT no longer accurately represents the driver behavior, and $MSE_{FFNN} > MSE_{LSTM}$. Consequentely, the switching logic selects the LSTM-DT, i.e., $A_{p,pred} = A_{p,pred,LSTM}$. Once the KPIs correctly classify the driver as aggressive, the corresponding offline-trained FFNN-DT provides the lowest prediction loss, and is therefore selected. A similar behavior is observed during the transition from the aggressive to the soft driving style, where the LSTM-DT temporarily achieves lower prediction error during the identification delay. In the final phase of the simulation, starting from ∼1300 s, the driver exhibits intermediate behavior ($DS = 0$). In this region $MSE_{FFNN} > MSE_{LSTM}$ holds, and the LSTM-DT remains selected. A zoomed view of this interval, shown in Fig. 11(c), highlights the benefit of the online adaptation of the LSTM architecture, with a progressive decrease of the prediction loss as the network adapts to the intermediate driving style.

5. Proof-of-concept simulation results

All the preliminary proof-of-concept results of this section are obtained with the simulation architecture in Fig. 2, by using the real-time capable controller setup from Section 3.3. The performance assessment is conducted through a structured comparison among three configurations of the same vehicle: i) the baseline configuration, equipped with conventional passive drivability maps (P); ii) the active vehicle (A) including ADMs but without the driver intent preview module, which is the benchmarking controller setup; and iii) the complete control framework (A+P), including the NN-based driver preview. Section 5.1 analyzes the influence of the cost function weights on the control system performance, while – based on the identified weight values – Section 5.2 discusses the trade-offs between energy savings and driver disturbance. The three considered configurations enable the isolation of the benefits of the adaptive drivability map functionality on its own – by means of the comparison between P and A – and those brought by the predictive driver model – by means of the comparison between A and A+P. Hence, the assessment setup allows consistent evaluation of the incremental added value of the proposed approaches within their intended human-in-the-loop application.

5.1. Effect and optimization of the cost function weights

Since the ADM performance is strongly influenced by the absolute and relative magnitude of the cost function terms in (14), a sensitivity analysis is carried out on the related weights. A low (i.e., 400 N) $F_{x,corr,max}$ value is selected, to assess the potential ADM benefit when operating with a low invasiveness level over the human driver input. The analysis is conducted along a 4.5 km scenario, including urban and extra-urban driving, without preceding vehicles, and with the speed limit profile in Fig. 13(c). Since PB-HDM is computationally heavy, to run a high number of simulations the sensitivity results are obtained by replacing PB-HDM with FFNN-DT in the human driver block in Fig. 2, hence the applied accelerator pedal input is $\alpha_{p,FFNN}$. A dedicated design of experiments (DoE) is set, based on:

- Full factorial design (FFD) [51] with 5 levels (logarithmic distribution) for the involved weights, which are condensed into only 3, i.e., w_E, $w_{\dot{x}} = w_{F_x}$, and $w_{F_{x,corr}} = w_u$. This choice is made to reduce the

computational burden, considering the normalized setting of (14), and the similar effect of $w_{\dot{x}}$ and w_{F_x} as well as $w_{F_{x,corr}}$ and w_u. The validity of this condensation strategy was confirmed by an a posteriori sensitivity analysis performed using the original five-weight formulation. Each weight involved in the condensation was varied independently, while keeping the remaining weights fixed at their optimized values obtained in this section. This allowed isolating the individual weight influence on the KPIs in (45)–(47). The results showed consistent monotonic and comparable trends between the paired weights (i.e., $w_{F_{x,corr}}$ and w_u; $w_{\dot{x}}$ and w_{F_x}), confirming their similar effect. In summary, the FFD with the condensed weights results into 125 unique combinations, where each weight is imposed to have equal values in the stage and terminal costs.

- Latin hypercube sampling (LHS) [52], to generate additional 175 samples with linear distribution, thus enriching the parameter space coverage. $W_{c,j}$ is not involved in the optimization, since it was verified that once it is appropriately defined to provide the expected soft constraint effect, its further increase does not produce any significant performance variation.

The sensitivity results are obtained for the configurations A and A+P. The results, obtained for an aggressive driver, are reported in Fig. 12, whose subplots have the involved weights on the axes, while the color scales of the dots, each corresponding to a simulation, relate to the value of the reported KPIs, namely:

- ΔE, i.e., the percentage variation (%) of the consumed energy per traveled distance ($E_{d,A/A+P}$) of the considered ADM configuration (A or A+P), w.r.t. the one ($E_{d,P}$) of the passive configuration, i.e. with static drivability maps, referred to as P:

$$\Delta E = \frac{E_{d,A/A+P} - E_{d,P}}{E_{d,P}} \times 100 \tag{45}$$

- The nondimensional cumulated longitudinal powertrain force correction, $\widehat{F}_{x,corr}$, imposed by the ADM algorithm:

$$\widehat{F}_{x,corr} = \frac{\int_{t_{in}}^{t_{fin}} |F_{x,corr}| dt}{F_{x,corr,max}\left[t_{fin} - t_{in}\right]} \tag{46}$$

which indicates the level of intrusiveness of the algorithm.

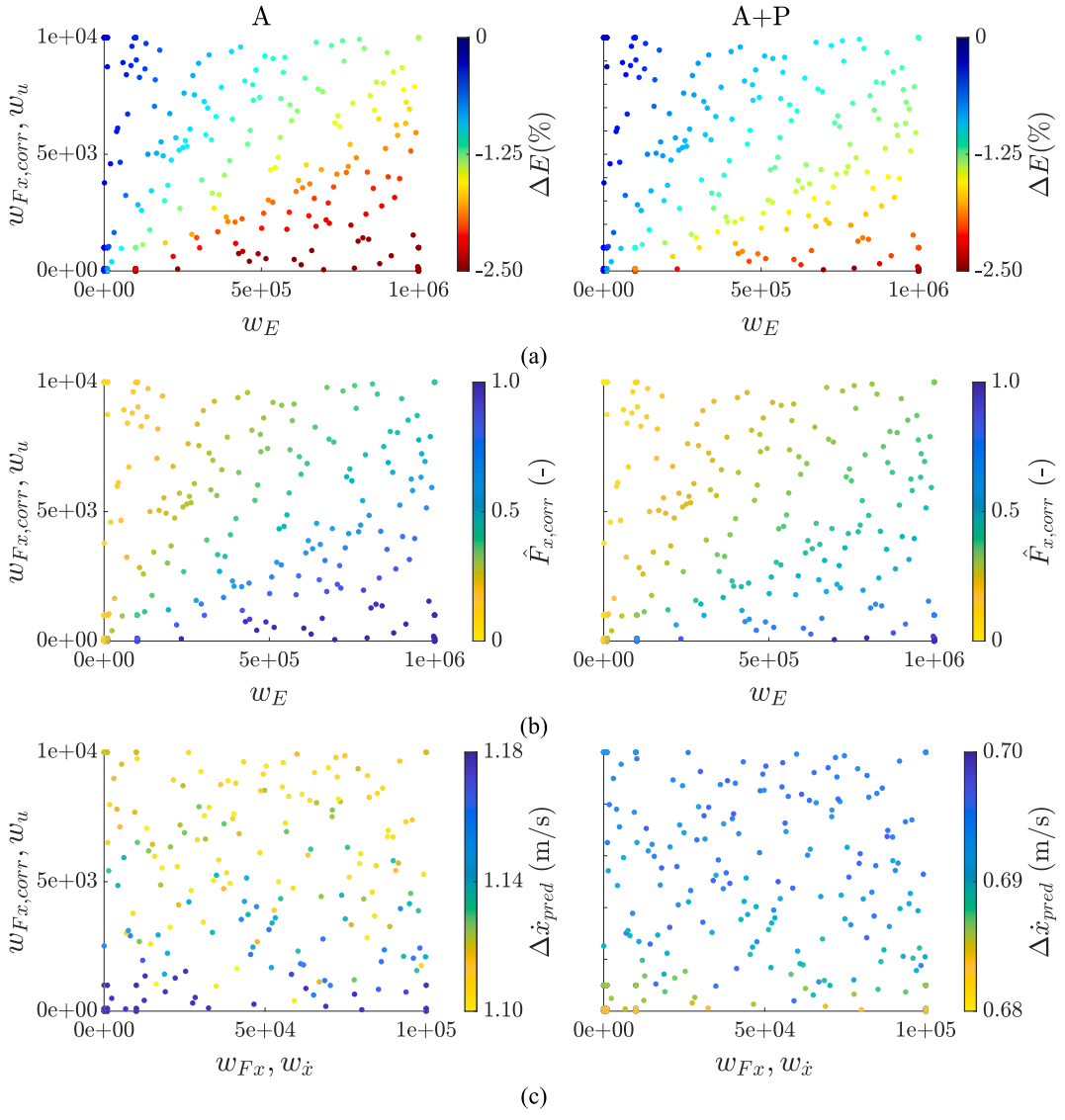


Fig. 12. Sensitivity analysis on the ADM cost function weights: influence on energy consumption reduction (ΔE in (a)) and KPIs of the intrusiveness level of the control function ($\widehat{F}_{x,corr}$ and $\Delta \dot{x}_{pred}$ in (b) and (c)) considering A (left) and A+P (right) configurations.

- The root mean square value of the difference between the current vehicle speed, $\dot{x}_h(t)$, and its predicted value for the final prediction horizon step N, computed at the time $t - T_p$ through (16), i.e., in absence of driver torque demand alteration by the ADM function:

$$\Delta\dot{x}_{pred} = \mathrm{RMS}\big(\dot{x}_h(t) - \dot{x}_{pred}(N)(t - T_p)\big) \tag{47}$$

In configuration A, $\dot{x}_{pred}(N)(t - T_p)$ is based on the assumption of a constant accelerator pedal input along T_p, while in A+P it accounts for the NN-DT generated $A_{p,pred}$ profile. Hence, $\Delta\dot{x}_{pred}$ measures the extent of the longitudinal vehicle performance variation brought by application of the online control sequence, w.r.t. the original target defined N steps earlier. A low $\Delta\dot{x}_{pred}$ value indicates an efficient operation of the algorithm, i.e., without significant preview and prediction errors, nor control action intrusiveness.

Fig. 12 only includes the most significant subplots in terms of axes and KPI combinations. As w_E increases and $w_{F_{x,corr}} = w_u$ decreases, $\Delta E_{A/A+P}$ in Fig. 12(a) spans from 0% to −2.5%. Configuration A has a marginally wider region with energy consumption reductions exceeding 2%, w.r.t. A+P. Such effect is predictable, since the preview aims at reducing the level of control-induced variation of the traction force w.r.t. the human driver request. In Fig. 12(b), $\widehat{F}_{x,corr}$ increases if w_E increases and/or $w_{F_{x,corr}}$ and w_u decrease. Interestingly, opposite trends are visible for A and A+P in terms of $\Delta\dot{x}_{pred}$ variations in Fig. 12(c), i.e., a reduction of $w_{F_{x,corr}}$ and w_u, corresponding to larger controller interventions, brings higher $\Delta\dot{x}_{pred}$ values for A (from 1.10 to 1.18 m/s), and marginally lower $\Delta\dot{x}_{pred}$ values (from 0.70 down to 0.68 m/s) for A+P. This means that in A+P the control action increase corresponds to approximately constant or reduced driver disturbance level. Moreover, the significantly lower – by ∼50% – $\Delta\dot{x}_{pred}$ values of A+P clearly highlight its overall higher effectiveness w.r.t. A.

The following analyses use an optimized set of weights, obtained through iterative simulation runs managed by the *surrogateopt* algorithm [53], with a maximum number of evaluations set to 100. The optimization cost function, $J_{opt,w}$, is:

$$J_{opt,w} = w_{en}\widehat{E} + w_{\dot{x}_{pred}}\widehat{\Delta\dot{x}}_{pred} + w_{F_{x,track}}\widehat{\Delta F}_{x,track} \tag{48}$$

where $\widehat{E}$, $\widehat{\Delta\dot{x}}_{pred}$, and $\widehat{\Delta F}_{x,track}$ are normalized terms accounting for the energy consumption, speed tracking performance, and longitudinal tire force tracking performance; and w_{en}, $w_{\dot{x}_{pred}}$, and $w_{F_{x,track}}$ are the respective weights, set to 0.8, 0.1, and 0.1, which brings non-negligible values of $w_{F_x,k/N}$, $w_{F_{x,corr},k/N}$, $w_{\dot{x},N}$, and $w_{\dot{F}_{x,corr},k}$, not to compromise the human driver experience. The optimization process was carried out for $F_{x,corr,max}$ = 400 N.

5.2. Trade-offs between energy consumption and control intrusiveness

Building upon the sensitivity analysis and weight optimization of Section 5.1, which systematically investigated the impact of the ADM

Table 7
Sensitivity analysis results on the maximum corrective action level.

	Soft		Aggressive	
$F_{x,corr,max}$	ΔE_A	ΔE_{A+P}	ΔE_A	ΔE_{A+P}
200 N	−1.73%	−1.12%	−1.30%	−1.32%
300 N	−2.38%	−2.11%	−1.90%	−1.90%
400 N	−2.91%	−2.54%	−2.43%	−2.46%
500 N	−3.41%	−2.92%	−2.97%	−2.97%
600 N	−3.92%	−3.25%	−3.44%	−3.49%

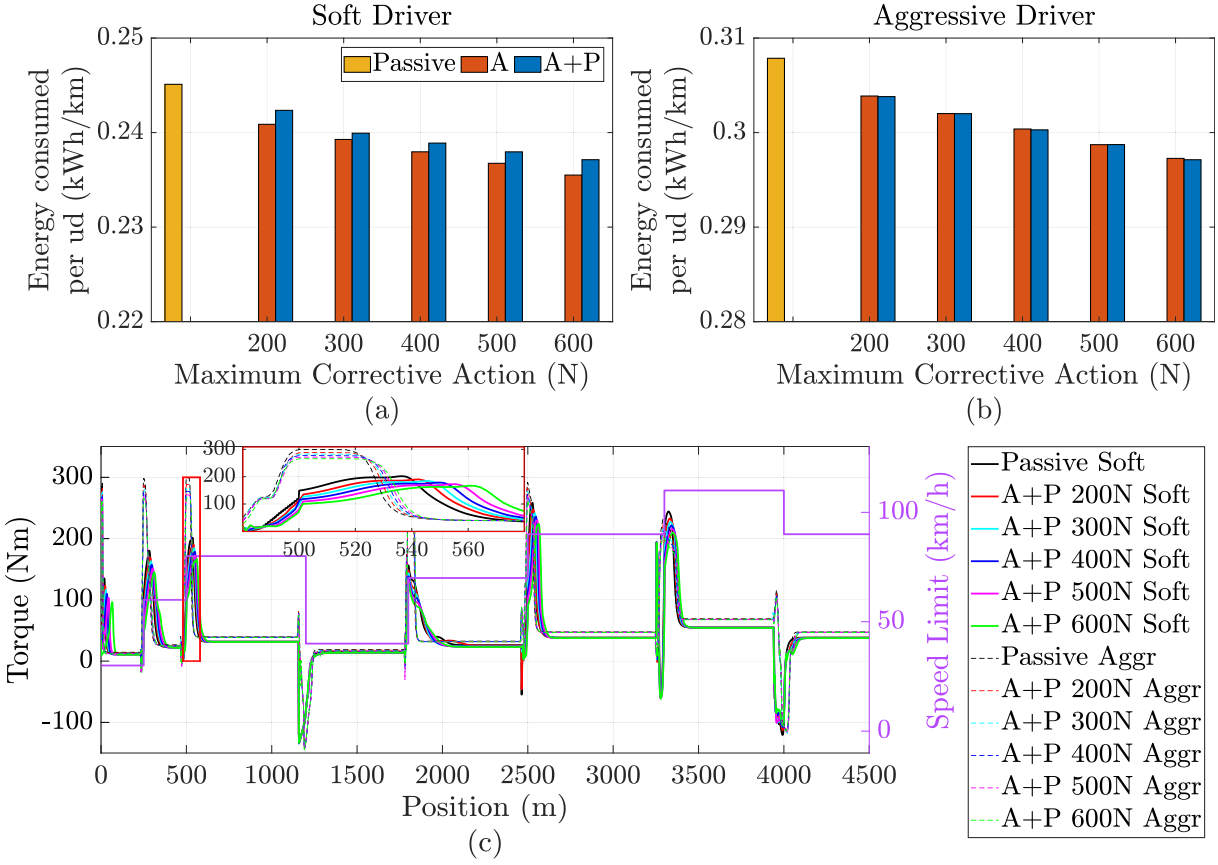


Fig. 13. Sensitivity analysis on the maximum corrective action: influence on energy consumption, considering soft (a) and aggressive (b) driving styles; (c) Corrected torque demand profiles of the A+P configuration compared to the torque request profiles for the passive case, along with the speed limits.

cost function weights on the energy consumption and performance metrics, the following analysis adopts the fixed set of optimized weights. Fig. 13 and Table 7 investigate the effect of the maximum corrective force, $F_{x,corr,max}$, along the scenario of Section 5.1, without varying the ADM cost function weights. In the sensitivity, $F_{x,corr,max}$ is varied from 200 to 600 N, in steps of 100 N. Consistently with the literature [54], the comparison of Fig. 13(a) and (b) highlights a significant energy consumption increment (amounting to 25.6%, i.e., from 0.245 to 0.308 kWh/km) in the passive vehicle configuration, when moving from the soft to the aggressive driver model. For the controlled cases, as $F_{x,corr,max}$ increases, the energy consumption monotonically decreases with a substantially linear and predictable behavior, for both the soft and aggressive drivers. Such reduction is caused by the larger correction of the driver torque request, as highlighted by the time profiles of the corrected torque, $T_{corrected}$, in Fig. 13(c), i.e., $T_{corrected} = \left[F_{x,pwt,drv} - F_{x,corr}\right]\eta_{dvl}^{\mp 1} R_w/\tau_{tot}$, where the exponent ∓ 1 depends on whether the system is operating in traction or regeneration. Table 7 reports the percentage variation of the energy consumption of the A and A+P variants w.r.t. the passive one. A higher reduction is often reached in the soft driver case, rather than the aggressive one. For the soft driver, a marginally higher consumption reduction is achieved by A rather than A+P. This is not concerning, as the main goal of the driver intent preview is to reduce controller intrusiveness, which has been quantified in Fig. 12, and will be further analyzed in the following scenario, comparing the control intervention level for approximately the same energy saving.

Fig. 14 refers to a 400 m long scenario with three speed limit variations. The short distance and limited duration enable the adoption of PB-HDM within the plant model, while NN-DT is used for predicting driver behavior within the ADM architecture. The results are reported for the aggressive driver. The ADM controller behaves similarly to the previous analyses. Fig. 14(b) shows the corrected torque demand, which tends to be smoothened by ADM. Fig. 14(c) confirms the accuracy of $A_{p,pred}$, in a different scenario from those used for network training. The spider plot of Fig. 14(d) highlights the benefits brought by the

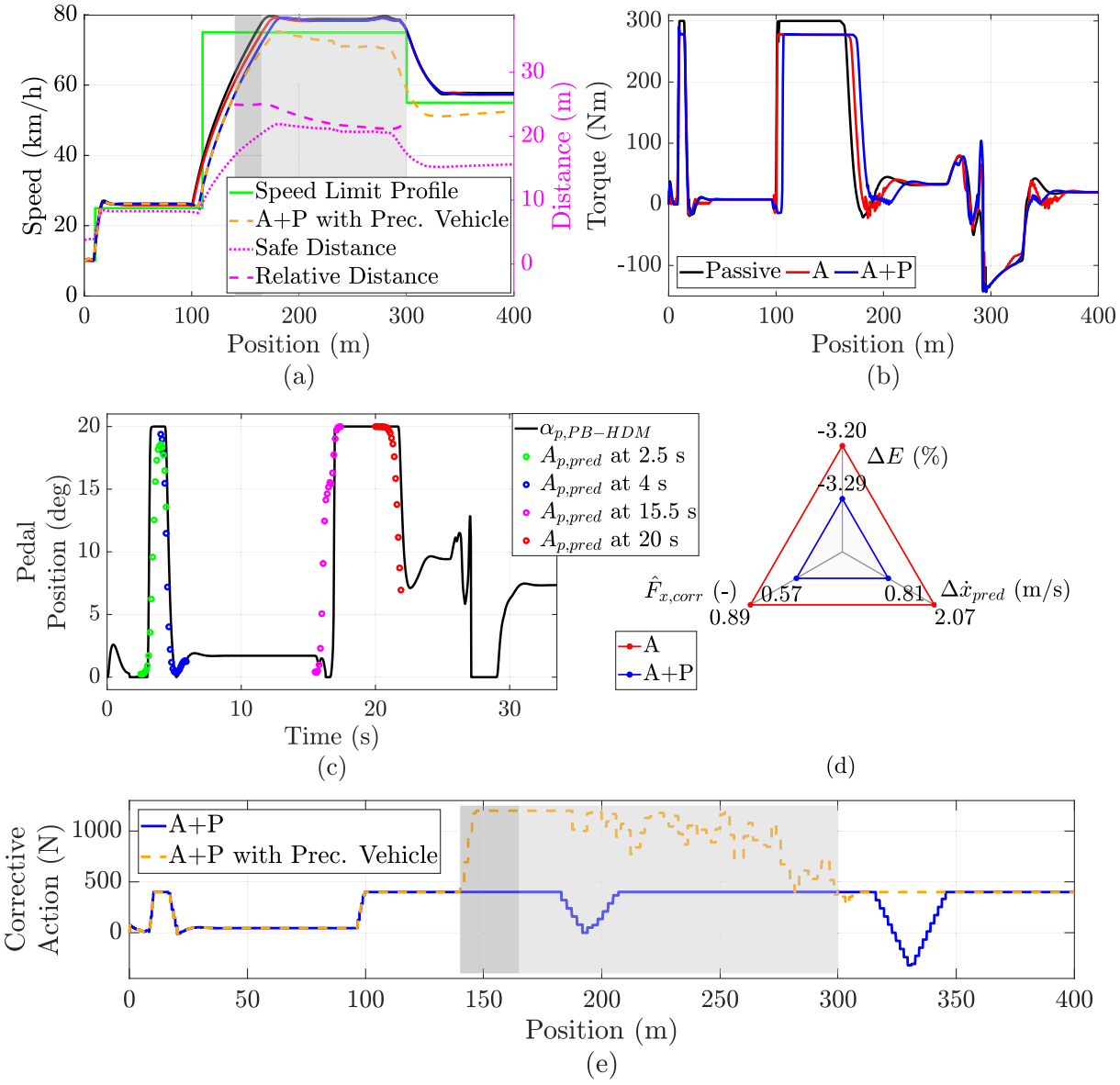


Fig. 14. ADM effects on the A and A+P vehicles with the aggressive PB-HDM calibration: (a) Speed profiles of A and A+P without the interference of a preceding vehicle, and A+P with the interference of a preceding vehicle along with the relative and safe distances; (b) Corrected torque of A and A+P, compared to the passive case; (c) Instantaneous pedal position along with its prediction in some relevant instants; (d) Comparison of energy reduction and KPIs describing the intrusiveness level of the control function, for the A and A+P vehicles; and (e) Comparison of the A+P corrective action with and without preceding vehicle interference.

introduction of the driver's intent preview. In fact, the driver disturbance KPIs are significantly reduced when moving from A to A+P. In particular, $\Delta\dot{x}_{pred}$ is more than halved, while – importantly – also the energy consumption is marginally lower.

For completeness, subplots (a) and (e) also include the profiles for A+P (referred to as 'A+P with Prec.') in case of interference from a preceding vehicle, which is detected by the radar sensor of the host vehicle. In particular, the preceding vehicle joins the host EV lane when the latter is at 140 m, with an initial 25 m x_{rel} value, and remains on the lane for 160 m, see the gray patches. The preceding vehicle speed profile is similar to the host EV one in the dark gray region, before saturating at 72 km/h, i.e., a lower value than for the host vehicle, in the light gray region. The A+P with Prec. simulation uses the same α_p profile as A+P, to emulate a scenario in which the human driver does not recognize the presence of the slower vehicle ahead. When x_{rel} goes below an appropriate time-varying safety critical threshold, safety constraints become active within the NMPC problem. In this condition, $F_{x,corr,max}$ is automatically increased to 1200 N, and the ADM NMPC weights are adjusted to prioritize the relative distance constraint satisfaction over energy consumption and driver intent tracking. This mechanism allows the controller to temporarily apply braking forces exceeding the nominal driver request to guarantee collision avoidance. Hence, in Fig. 14(e), $F_{x,corr}$ suddenly increases to slow down the host vehicle and avoid the collision, and the controller returns to standard operation when the preceding vehicle leaves the lane.

The simulation analysis also verified that: i) if the controller is given additional freedom by significantly reducing the driver disturbance weights, and by increasing $F_{x,corr,max}$, it is possible to drastically reduce the energy consumption, at the price of increasing ADM intrusiveness. For example, by doing so, in the Fig. 14 scenario, it is possible to achieve ~8.6% energy consumption reduction w.r.t. the passive configuration. This result highlights the combined effect of cost function tuning and control authority on the obtainable energy savings; and ii) the driving scenarios in which the ADM functionality is particularly effective are those with frequent speed variations. These results further confirm that the proposed ADM framework enables a flexible trade-off between energy efficiency and control intrusiveness, which can be tuned through both the cost function weights and the admissible corrective force.

6. Conclusions

This preliminary proof-of-concept study introduced a novel driver-aware EDAS, in the form of the adaptive drivability map (ADM) functionality, to reduce the energy consumption of human-driven EVs, while considering the intrusiveness level caused by the associated longitudinal force correction. A physics-based high-fidelity human driver model, validated through static driving simulator tests, was used to train computationally efficient feedforward neural networks (FFNNs) that predict archetypal driver behaviors corresponding to distinct driving styles. To capture intermediate or evolving behaviors that cannot be strictly categorized, an online-trained long short-term memory neural network complements the FFNNs, and is continuously updated during vehicle operation. A dedicated online switching algorithm selects the most appropriate driver digital twin configuration, which then generates preview information for the ADM algorithm, implemented as a real-time capable nonlinear model predictive controller. By embedding a preview of the driver's future pedal inputs, the framework addresses the trade-off between energy efficiency and control intrusiveness. The simulation analyses of the complete control architecture show: i) 2-to-4% energy consumption reductions w.r.t. static drivability maps; and ii) ~50% decrease of the driver intrusiveness indicators w.r.t. to the ADM functionality excluding driver behavior preview. These results highlight the importance of explicitly incorporating human driving behavior when designing EDAS for EVs, and suggest that predictive, human-in-the-loop drivability adaptation represents a promising approach for energy-efficient mobility systems.

Future developments will focus on: a) the experimental implementation and assessment of the proposed functionality on a vehicle prototype; b) the investigation of the potential integration of driver biometric signals (e.g., physiological or eye-tracking indicators) to further enhance driver state representation and intent prediction capabilities; and c) the development of enhanced preceding vehicle motion prediction strategies, including the estimation of acceleration profiles or probabilistic trajectory prediction methods, to improve robustness in highly dynamic and emergency scenarios.

CRediT authorship contribution statement

Luca Ciravegna: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Luca Dimauro:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Project administration, Methodology, Investigation, Data curation, Conceptualization. **Gianluca Frison:** Writing – review & editing, Validation, Software, Investigation, Data curation, Conceptualization. **Fabio Alberti:** Writing – review & editing, Visualization, Software, Methodology, Investigation, Data curation, Conceptualization. **Enrico Galvagno:** Data curation, Investigation, Methodology, Software, Visualization, Writing – review & editing. **Aldo Sorniotti:** Writing – review & editing, Visualization, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Horizon Europe Programme of the European Commission under Grant agreement numbers 101096062 (CLIMAFlux Project) and 101138266 (EFFEREST Project).

Data availability

Data will be made available on request.

References

[1] Tu R, Xu J, Li T, Chen H. Effective and acceptable eco-driving guidance for human-driving vehicles: a review. Int. J. Environ. Res. Public Health 2022;19(12):7310. https://doi.org/10.3390/ijerph19127310.
[2] Li W, Ding H, Xu N, Zhang J. Toward carbon-neutral transportation electrification: a comprehensive and systematic review of eco-driving for electric vehicles. IEEE Trans. Transp. Electrific. 2024;10(3):6340–60. https://doi.org/10.1109/TTE.2023.3331727.
[3] Åkerblom N, Chen Y, Chehreghani MH. Online learning of energy consumption for navigation of electric vehicles. Artif. Intell. 2023;317. https://doi.org/10.1016/j.artint.2023.103879.
[4] Vasebi S, Hayeri YM, Saghiri AM. A literature review of energy optimal adaptive cruise control algorithms. IEEE Access 2023;11:13636–13,646. https://doi.org/10.1109/ACCESS.2023.3241140.
[5] Li J, Fotouhi A, Liu Y, Zhang Y, Chen Z. Review on eco-driving control for connected and automated vehicles. Renew. Sust. Energ. Rev. 2024;189. https://doi.org/10.1016/j.rser.2023.114025.
[6] Sciarretta A, Vahidi A. Energy-efficient Driving of Road Vehicles. Cham, Switzerland: Springer International Publishing; 2020.
[7] Huang Y, Ng EC, Zhou JL, Surawski NC, Chan EF, Hong G. Eco-driving technology for sustainable road transport: a review. Renew. Sust. Energ. Rev. 2018;93:596–609. https://doi.org/10.1016/j.rser.2018.05.030.
[8] Boriboonsomsin K, Barth M. Impacts of road grade on fuel consumption and carbon dioxide emissions evidenced by use of advanced navigation systems. Transp. Res. Rec. 2009;2139(1):21–30. https://doi.org/10.3141/2139-03.
[9] Pan C, Huang A, Chen L, Cai Y, Chen L, Liang J, et al. A review of the development trend of adaptive cruise control for ecological driving. Proc. Inst. Mech. Eng., Part

D: J. Automob. Eng. 2022;236(9):1931–48. https://doi.org/10.1177/09544070211049068.
[10] Wang Q, Ju F, Zhuang W, Wang L. Ecological cruising control of connected electric vehicle: A deep reinforcement learning approach. SCIENCE CHINA Technol. Sci. 2022;65(3):529–40. https://doi.org/10.1007/s11431-021-1994-7.
[11] Lee H, Kim N, Cha SW. Model-based reinforcement learning for eco-driving control of electric vehicles. IEEE Access 2020;8:202886–96. https://doi.org/10.1109/ACCESS.2020.3036719.
[12] Jia Y, Jibrin R, Görges D. Energy-optimal adaptive cruise control for electric vehicles based on linear and nonlinear model predictive control. IEEE Trans. Veh. Technol. 2020;69(12):14173–87. https://doi.org/10.1109/TVT.2020.3044265.
[13] Ojeda LL, Han J, Sciarretta A, De Nunzio G, Thibault L. A real-time eco-driving strategy for automated electric vehicles. In: IEEE 56th Annual Conference on Decision and Control (CDC), Melbourne, VIC, Australia; 2017. p. 2768–74. https://doi.org/10.1109/CDC.2017.8264061.
[14] Bai Z, Hao P, ShangGuan W, Cai B, Barth MJ. Hybrid reinforcement learning-based eco-driving strategy for connected and automated vehicles at signalized intersections. IEEE Trans. Intell. Transp. Syst. 2022;23(9):15850–63. https://doi.org/10.1109/TITS.2022.3145798.
[15] Tajeddin S, Ekhtiari S, Faieghi M, Azad NL. Ecological adaptive cruise control with optimal lane selection in connected vehicle environments. IEEE Trans. Intell. Transp. Syst. 2020;21(11):4538–49. https://doi.org/10.1109/TITS.2019.2938726.
[16] Alatawneh A, Torok A. A predictive modeling framework for forecasting cumulative sales of euro-compliant, battery-electric and autonomous vehicles. Decis. Anal. J. 2024;11. https://doi.org/10.1016/j.dajour.2024.100483.
[17] Shladover SE. Connected and automated vehicle systems: introduction and overview. J. Intell. Transp. Syst. 2017;22(3):10–200. https://doi.org/10.1080/15472450.2017.1336053.
[18] Andrieu C, Saint Pierre G. Comparing effects of eco-driving training and simple advices on driving behavior. Procedia Soc. Behav. Sci. 2012;54:211–20. https://doi.org/10.1016/j.sbspro.2012.09.740.
[19] Jeffreys I, Graves G, Roth M. Evaluation of eco-driving training for vehicle fuel use and emission reduction: a case study in Australia. Transp. Res. Part D: Transp. Environ. 2018;60:85–91. https://doi.org/10.1016/j.trd.2015.12.017.
[20] McIlroy RC, Stanton NA, Godwin L, Wood AP. Encouraging eco-driving with visual, auditory, and vibrotactile stimuli. IEEE Trans. Human-Mach. Syst. 2017;47(5):661–72. https://doi.org/10.1109/THMS.2016.2608937.
[21] Hibberd DL, Jamson AH, Jamson SL. The design of an in-vehicle assistance system to support eco-driving. Transp. Res. Part C Emerg. Technol. 2015;58(Part D):732–48. https://doi.org/10.1016/j.trc.2015.04.013.
[22] McIlroy RC, Stanton NA, Godwin L. Good vibrations: using a haptic accelerator pedal to encourage eco-driving. Transp. Res. Part F Traffic Psychol. Behav. 2017;46 (part A):34–46. https://doi.org/10.1016/j.trf.2017.01.002.
[23] Chada SK, Görges D, Ebert A, Teutsch R, Min CG. Learning-based driver behavior modeling and delay compensation to improve the efficiency of an eco-driving assistance system. In: IEEE International Conference on Systems, Man, and Cybernetics (SMC). Prague, Czech Republic; 2022. p. 415–22. https://doi.org/10.1109/SMC53654.2022.9945577.
[24] Chada SK, Görges D, Ebert A, Teutsch R, Subramanya SP. Evaluation of the driving performance and user acceptance of a predictive eco-driving assistance system for electric vehicles. Transp. Res. Part C Emerg. Technol. 2023;153. https://doi.org/10.1016/j.trc.2023.104193.
[25] Hong J, Luo X, Wu H, Na X, Chu H, Gao B, et al. Energy-saving driving assistance system integrated with predictive cruise control for electric vehicles. IEEE Trans. Intellig. Vehicles 2024;9(3):4518–28. https://doi.org/10.1109/TIV.2024.3358797.
[26] Shi Q, He Z, Wei Y, Wang M, Zheng X, He L. Single pedal control of battery electric vehicle by pedal torque demand with dynamic zero position. IEEE Trans. Intell. Transp. Syst. 2022;23(11):21608–19. https://doi.org/10.1109/TITS.2022.3181042.
[27] Zhang Y, Huang Y, Chen H, Na X, Chen Z, Liu Y. Driving behavior oriented torque demand regulation for electric vehicles with single pedal driving. Energy 2021;228. https://doi.org/10.1016/j.energy.2021.120568.
[28] Pick AJ, Cole DJ. A mathematical model of driver steering control including neuromuscular dynamics. J. Dyn. Syst. Meas. Control. 2008;130(3). https://doi.org/10.1115/1.2837452.
[29] Hanbing W, Yanhong W, Xing C, Jin X, Sharma R. Human-vehicle dynamic model with driver's neuromuscular characteristic for shared control of autonomous vehicle. Proc. Inst. Mechan. Eng. Part D 2023;237(10−11):2402–14. https://doi.org/10.1177/0954407020977108.
[30] Gholami A, Majidi M. Development of a neuromuscular driver model with an estimation of steering torque feedback in vehicle steer-by-wire systems. Proc. Inst. Mechan. Eng. Part K 2019;233(3):657–67. https://doi.org/10.1177/1464419319829980.
[31] Cole DJ. A path-following driver-vehicle model with neuromuscular dynamics, including measured and simulated responses to a step in steering angle overlay. Veh. Syst. Dyn. 2011;50(4):573–96. https://doi.org/10.1080/00423114.2011.606370.
[32] Wu H, Wei H, Liu Z, Xu J. A simplified dynamic model with driver's NMS characteristic for human-vehicle shared control of autonomous vehicle. Proc. Inst. Mechan. Eng. Part D 2021;236(1):16–28. https://doi.org/10.1177/09544070211018944.
[33] Abbink DA, Mulder M, Van der Helm FCT, Mulder M, Boer ER. Measuring neuromuscular control dynamics during car following with continuous haptic feedback. IEEE Trans. Syst. Man. Cybernet. Part B (Cybernet.) 2011;41(5):1239–49. https://doi.org/10.1109/TSMCB.2011.2120606.
[34] Mulder M. Haptic Gas Pedal Feedback for Active Car-Following Support. Ph.D. dissertation. Delft, The Netherlands: Dept. Aerospace Eng., Delft Univ. Technol; 2007.
[35] Abbink DA. Neuromuscular Analysis of Haptic Gas Pedal Feedback during Car Following. Ph.D. dissertation. Delft, The Netherlands: Dept. Mech. Eng., Delft Univ. Technology; 2006.
[36] Shen H, Wang Z, Zhou X, Lamantia M, Yang K, Chen P, et al. Electric vehicle energy consumption estimation with consideration of longitudinal slip ratio and machine-learning-based powertrain efficiency. IFAC Papers Line 2022;55(37):158–63.
[37] Shen H, Wang Z, Zhou X, Lamantia M, Chen P, Wang J. Electric vehicle velocity and energy consumption predictions using transformer and Markov-Chain Monte Carlo. IEEE Trans. Transp. Electrific. 2022;8(3):3836–47. https://doi.org/10.1109/TTE.2022.3157652.
[38] Shen H, Zhou X, Ahn H, Lamantia M, Chen P, Wang J. Personalized velocity and energy prediction for electric vehicles with road features in consideration. IEEE Trans. Transp. Electrific. 2023;9(3):3958–69. https://doi.org/10.1109/TTE.2023.3241098.
[39] Frison G, Alberti F, Ciravegna L, Dimauro L, Sorniotti A. Self-adaptive neural network model predictive anti-jerk control of electric powertrains. Mech. Mach. Theory 2025;214. https://doi.org/10.1016/j.mechmachtheory.2025.106082.
[40] Dalboni M, et al. Nonlinear model predictive control for integrated energy-efficient torque-vectoring and anti-roll moment distribution. IEEE/ASME Trans. Mechatron. 2021;26(3):1212–24. https://doi.org/10.1109/TMECH.2021.3073476.
[41] Verschueren R, et al. Acados—a modular open-source framework for fast embedded optimal control. Math. Program. Comput. 2022;14(1):147–83. https://doi.org/10.1007/s12532-021-00208-8.
[42] Parra A, Tavernini D, Gruber P, Sorniotti A, Zubizarreta A, Pérez J. On pre-emptive vehicle stability control. Veh. Syst. Dyn. 2022;60(6):2098–123. https://doi.org/10.1080/00423114.2021.1895229.
[43] Guastadisegni G, So KM, Parra A, Tavernini D, Montanaro U, Gruber P, et al. Vehicle stability control through pre-emptive braking. Int. J. Automot. Technol. 2023;24(2):347–65. https://doi.org/10.1007/s12239-023-0029-2.
[44] So KM, Tavolo G, Tavernini D, Grosso M, Pozzato S, Perlo P, et al. Novel pre-emptive control solutions for V2X connected electric vehicles. In: Proc. Transport Res. Arena Conf.; 2022. p. 1–8. https://doi.org/10.48550/arXiv.2406.02211.
[45] Mileusnic MP, Loeb GE. Mathematical models of proprioceptors. II. Structure and function of the Golgi tendon organ. J. Neurophysiol. 2006;96(4):1789–802. https://doi.org/10.1152/jn.00869.2005.
[46] Martinez CM, Heucke M, Wang FY, Gao B, Cao D. Driving style recognition for intelligent vehicle control and advanced driver assistance: a survey. IEEE Trans. Intell. Transp. Syst. 2018;19(3):666–76. https://doi.org/10.1109/TITS.2017.2706978.
[47] Wang W, Xi J, Chong A, Li L. Driving style classification using a semisupervised support vector machine. IEEE Trans. Human-Mach. Syst. 2017;47(5):650–60. https://doi.org/10.1109/THMS.2017.2736948.
[48] Shi B, Xu L, Hu J, Tang Y, Jiang H, Meng W, et al. Evaluating driving styles by normalizing driving behavior based on personalized driver modeling. IEEE Trans. Syst. Man. Cybernet. Syst. 2015;45(12):1502–8. https://doi.org/10.1109/TSMC.2015.2417837.
[49] Karaduman M, Eren H. Deep learning based traffic direction sign detection and determining driving style. In: International Conference on Computer Science and Engineering (UBMK), Antalya, Turkey; 2017. p. 1046–50. https://doi.org/10.1109/UBMK.2017.8093453.
[50] Zhen D, Wang T, Gu F, Ball AD. Fault diagnosis of motor drives using stator current signal analysis based on dynamic time warping. Mech. Syst. Signal Process. 2013;34(1–2):191–202. https://doi.org/10.1016/j.ymssp.2012.07.018.
[51] Montgomery DC. Design and Analysis of Experiments. John Wiley & Sons; 2017.
[52] Santner TJ, Williams BJ, Notz WI. The Design and Analysis of Computer Experiments1. New York: Springer; 2003.
[53] MathWorks. Surrogate Optimization Algorithm. [Online]. Available, https://it.mathworks.com/help/gads/surrogate-optimization-algorithm.html; 2024.
[54] Miotti M, Needell ZA, Ramakrishnan S, Heywood J, Trancik JE. Quantifying the impact of driving style changes on light-duty vehicle fuel consumption. Transp. Res. Part D: Transp. Environ. 2021;98. https://doi.org/10.1016/j.trd.2021.102918.