

Model predictive path tracking control for automated road vehicles: A review

Original

Model predictive path tracking control for automated road vehicles: A review / Stano, P.; Montanaro, U.; Tavernini, D.; Tufo, M.; Fiengo, G.; Novella, L.; Sorniotti, A.. - In: ANNUAL REVIEWS IN CONTROL. - ISSN 1367-5788. - 55:(2023), pp. 194-236. [10.1016/j.arcontrol.2022.11.001]

Availability:

This version is available at: 11583/2990748 since: 2024-07-13T10:47:38Z

Publisher:

PERGAMON-ELSEVIER SCIENCE

Published

DOI:10.1016/j.arcontrol.2022.11.001

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)



Model predictive path tracking control for automated road vehicles: A review

P. Stano^a, U. Montanaro^a, D. Tavernini^a, M. Tufo^{b,c}, G. Fiengo^{b,c}, L. Novella^{b,c}, A. Sorniotti^{a,*}

^a Department of Mechanical Engineering Sciences, University of Surrey, GU2 7XH, Guildford, UK

^b Department of Engineering, University of Sannio, 82100 Benevento, Italy

^c Kineton R&D, Kineton S.r.l., 80146, Napoli, IT

ARTICLE INFO

Keywords:

Automated vehicles
Path tracking
Model predictive control
Prediction model
Optimal control problem

ABSTRACT

Thanks to their road safety potential, automated vehicles are rapidly becoming a reality. In the last decade, automated driving has been the focus of intensive automotive engineering research, with the support of industry and governmental organisations. In automated driving systems, the path tracking layer defines the actuator commands to follow the reference path and speed profile. Model predictive control (MPC) is widely used for trajectory tracking because of its capability of managing multi-variable problems, and systematically considering constraints on states and control actions, as well as accounting for the expected future behaviour of the system. Despite the very large number of publications of the last few years, the literature lacks a comprehensive and updated survey on MPC for path tracking. To cover the gap, this literature review deals with the research conducted from 2015 until 2021 on model predictive path tracking control. Firstly, the survey highlights the significance of MPC in the recent path tracking control literature, with respect to alternative control structures. After classifying the different typologies of MPC for path tracking control, the adopted prediction models are critically analysed, together with typical optimal control problem formulations. This is followed by a summary of the most relevant results, which provides practical design indications, e.g., in terms of selection of prediction and control horizons. Finally, the most recent development trends are analysed, together with likely areas of further investigations, and the main conclusions are drawn.

1. Introduction

In the last decades, self-driving vehicles, also known as automated vehicles (AVs), have become a topic of study and interest for academic researchers, industry, and governmental organisations.

The main benefit of automated driving (AD) is that, by replacing humans with sophisticated control systems, AVs have the potential of drastically reducing the number of road accidents and casualties. In 2017, ~10% of road accidents on the U.S. roads had fatal outcomes (De Oliveira Faria & Gabbard, 2020). In 2018, the yearly global road death toll amounted to 1.35 million, and road traffic injuries are currently the leading fatalities of people aged from 5 to 29 years old, according to the World Health Organisation (2018). According to the 2015 report of the U.S. Highway Traffic Safety Administration & U.S. Department of Transportation (2015), 94% of car crashes are estimated to be caused by human errors, while vehicle component failure or degradation, environmental factors (e.g., slick roads, bad weather), and other unknown

critical reasons are each responsible only for 2% of crashes. The safety potential of AVs has led several governments and organisations to set the ambitious target of reaching zero road fatalities by 2050, as indicated by the European Transport Safety Council (2016).

The second advantage of AD is that the internal spaces of the vehicle can be re-organised to enable other activities while travelling, i.e., to allow the occupants to rest or work, thus reducing tedium and increasing productivity (Litman & Litman, 2013; Salter et al., 2019, 2020). The exploitation of this opportunity requires an enhancement of the vehicle comfort level, which, however, is currently only marginally considered in AV designs (Iskander et al., 2019). Moreover, motion sickness (MS), which is the uncomfortable sensation caused by the effect of actual or perceived vehicle motions, is likely to affect AVs more than existing human-driven vehicles (Diels & Bos, 2016; Saruchi et al., 2020), and therefore represents a major challenge to tackle (Diels et al., 2017; Iskander et al., 2019).

From a control perspective, AD architectures can be classified into:

* Corresponding author.

E-mail address: a.sorniotti@surrey.ac.uk (A. Sorniotti).

List of symbols

A	State-space system matrix	$F_{y,ij}$	Lateral tyre force at the ij corner
a_{eq}	Equivalent acceleration in the frequency domain	$F_{y,0,ij}$	Lateral tyre force at the ij corner in pure slip condition
a_{MSDV}	Motion sickness dose value	$F_{z,ij}$	Vertical tyre force at the ij corner
a_{w_d,i_d}	Acceleration component in the frequency domain	$F_{z,ij}^{stat}$	Static vertical load at the ij wheel
a_x	Longitudinal acceleration	$F_{z,x}$	Longitudinal load transfer
$\dot{a}_{x,ref}$	Reference longitudinal jerk	$F_{z,y}$	Lateral load transfer
a_y	Lateral acceleration	$F_{\alpha}, F_{\mu,F}$	Adjustment factors
$a_{y,des}$	Desired lateral acceleration	f_{π}	Feature vector of a human driving trajectory
$a_{y,ref}$	Reference lateral acceleration	F_{π}	Feature vector of all human driving trajectories
B	Control action matrix	g	Gravitational acceleration
b_f, b_r	Front/rear track width	$g()$	System output function
B_{pac}	Stiffness factor	h	Distance between the centre of gravity and the roll axis
C	External disturbance matrix	$h()$	Inequality constraints function
c_i	Damping coefficient of the shock absorbers of the i axle	H_c	Control horizon
C_{pac}	Shape factor	h_{CG}	Centre of gravity height
C_x	Longitudinal slip stiffness	$h^l()$	Stage cost of the learning optimal control problem
C_y	Cornering stiffness	H_p	Prediction horizon
\tilde{C}_y	Cornering stiffness variation	h_r	Roll centre height
$C_{y,0}$	Nominal cornering stiffness	$i = f, r$	Subscript indicating the front or rear axles
CI	Computational index	$i_{\alpha} = x, y$	Subscript indicating the x or y axes
D_p	Look-ahead distance	i_t	Fuzzy logic rule
D_{pac}	Peak factor	i_v	Index referring to a sample of a specific variable
$D_{p,l}, D_{p,h}$	Look-ahead distance boundaries	I_w	Wheel mass moment of inertia
d_{safe}	Safe distance	I_x	Vehicle roll mass moment of inertia
E	Expectation	I_z	Vehicle yaw mass moment of inertia
e_{GP}	Gaussian process error	$j = l, r$	Subscript indicating the left or right sides
e_{nom}	Nominal model error	J	Cost function
E_{pac}	Curvature factor	j_c	Index referring to the current time step
e_x	Longitudinal offset error	j_l	Index referring to a system linearisation
E^x	Model uncertainty variance	J^{LMPC}	Cost function of the learning optimal control problem
e_y	Lateral offset error	J_{stage1}, J_{stage2}	Stage cost contributions
$e_{y,lim}$	Lateral offset error threshold	$J_{terminal}$	Terminal cost
$e_{y,min}, e_{y,max}$	Lateral offset error boundaries	k	Discretisation step along the prediction horizon
$\dot{e}_{y,p}$	Lateral offset error rate at the look-ahead distance	k_i	Spring stiffness at the i axle
$e_{y,p,min}, e_{y,p,max}$	Lateral offset error boundaries at the look-ahead distance	k_m	Motion sickness constant coefficient
e_{v_x}	Speed profile error	k_t	Vertical tyre stiffness
e_{ψ}	Heading angle error	k_{i_d}	Squared coefficient of the i_d axis
$e_{\psi,lim}$	Heading angle error threshold	k_{α}	Cornering stiffness adjustment coefficient
$e_{\psi,p}$	Heading angle error at the look-ahead distance	K_{δ}	Understeer coefficient
E_1, E_2	Calibration parameters that depend on the tyre-road friction coefficient	K_{ϕ}	Roll stiffness
$f()$	Prediction model function	k_{μ}	Lateral force adjustment coefficient
$F_{b,ij}$	Braking force at the ij corner	l	Learning strategy iteration
$f_C()$	Cornering stiffness function	l_f, l_r	Front and rear semi-wheelbases
$F_{c,ij}$	Suspension damper force at the ij corner	l_x	Index corresponding to the current lane of the vehicle
$F_{ellipse,max}$	Tyre force corresponding to the friction ellipse envelope	LTR	Lateral transfer ratio
$F_{k,ij}$	Suspension spring force at the ij corner	m	Total vehicle mass
f_t	Feature vector of the controller trajectory	m_s	Sprung mass
$f_{NN}()$	Neural network prediction model function	m_{tr}	Number of trials
f_r	Rolling resistance coefficient	m_{u}	Unsprung mass
F_r	Longitudinal resistance force	M^l	Set of indices associated with a successful iteration
$F_{T,ij}$	Traction force at the ij wheel	M_z	Yaw moment
$F_{T,tot}$	Total traction force at the wheels	min, max	Subscripts indicating the minimum and maximum value of a variable
$f_{tyre}()$	Tyre function	n	Number of samples of a specific variable
$f_{width}()$	Effective vehicle width function	N_c	Number of steps of the control horizon
$F_{x,ij}$	Longitudinal tyre force at the ij corner	N_p	Number of steps of the prediction horizon
$F_{x,0,ij}$	Longitudinal tyre force at the ij corner in pure slip condition	p	Parameter matrix
$F_{x,ij}^{slip,max}, F_{x,ij}^{slip,min}$	Longitudinal tyre force limitations at the ij corner	p_{long}	Ellipse shape parameter
		P_x	Diagonal weight matrix of the terminal cost
		q_k	Neural network input data
		$Q^l()$	Terminal cost of the learning optimal control problem
		Q_x	Diagonal weight matrix of the system output vector
		R	Diagonal weight matrix of the control action vector

r_c	Number of scheduling variables	α_{ij}	Tyre slip angle at the ij corner
r_w	Tyre radius	$\alpha_{min}, \alpha_{max}$	Lower and upper boundaries of the tyre slip angle
r_α	Tyre slip angle coefficient	$\bar{\alpha}_r$	Average rear cornering stiffness
s	Distance covered along the path	α_r^*	Predicted rear cornering stiffness
s^p	Distance covered along the path by the preceding vehicle	α_{road}	Angle expressing the road gradient
sf	Safety factor	α_{sat}	Saturation value of the tyre slip angle
SI	Stability index	β	Vehicle sideslip angle
SS^l	Sample safe set	β_{lim}	Vehicle sideslip angle threshold
T_{ave}	Average runtime	β_{ref}	Reference vehicle sideslip angle
$T_{b,ij}$	Braking torque at the ij corner	$\beta_{ref,SS}$	Steady-state reference vehicle sideslip angle
$T_{b,ij,act}$	Actual braking torque at the ij corner	Γ	Scheduling variable vector
$T_{b,ij,cal}$	Braking torque before actuation dynamics at the ij corner	$\hat{\Gamma}$	Predicted scheduling variable vector
T_c	CPU execution time	δ	Front wheel steering angle
t	Time	ξ	Front wheel steering angle rate
T_{ij}	Driving torque at the ij wheel	δ_h	Hand wheel steering angle
t_{ini}, t_{fin}	Initial and final time of the lap	$\dot{\delta}_h$	Hand wheel steering angle rate
T^l	Time at which the closed-loop system reaches the terminal set	$\delta_{min}, \delta_{max}$	Lower and upper boundaries for the steering angle
t_{lap}	Lap time	δ_r	Rear wheel steering angle
T_s	Sampling time	$\dot{\delta}_{ref}$	Reference front steering angle rate
$T_{s,c}$	Controller sampling time	Δa_x	Longitudinal acceleration variation
T_{set}	System settling time	$\Delta F_{b,ij}$	Braking force variation at the ij corner
t_{sim}	Simulation duration time	$\Delta F_{x,ij}$	Longitudinal tyre force variation at the ij corner
$T_{s,long}$	Long discretisation time step	$\Delta F_{y,ij}$	Lateral tyre force variation at the ij corner
$T_{s,short}$	Short discretisation time step	ΔM_z	Yaw moment variation
T_{tot}	Total driving torque	$\Delta T_{b,ij}$	Braking torque variation at the ij corner
T_δ	Steering torque	ΔT_{ij}	Driving torque variation at the ij corner
TI	Tracking index	Δt	Look-ahead time
u	Control action vector	ΔT	Total driving torque command rate
U	Decision variable vector	ΔT_δ	Steering torque variation
u_{min}, u_{max}	Lower and upper boundaries of the control action vector	$\Delta u_{min}, \Delta u_{max}$	Lower and upper boundaries for the control action variation
v	Vehicle speed	Δv	Vehicle speed variation
v_{rl}	Anti-rollover speed limit	$\Delta \delta$	Front wheel steering angle variation
v_x, v_y	Longitudinal and lateral vehicle speed components	$\Delta \delta_{min}, \Delta \delta_{max}$	Lower and upper boundaries for the steering angle variation
$v_{x,des}$	Desired longitudinal speed	$\Delta \delta_r$	Rear wheel steering angle variation
$v_{x,des}^{max}$	Maximum desired longitudinal speed	$\Delta \mathcal{L}$	Cost function of gradient-based optimisation
$v_{x,min}, v_{x,max}$	Vehicle speed limits	ϵ	Slack variable
$v_{x,ref}$	Reference longitudinal speed	ϵ	Exponential weight factor
$v_{z,b}, v_{z,u}$	Vertical speeds of the sprung and unsprung masses	ζ	Trajectory of the data set
W_d, W_f	Weighting functions	η	Derating coefficient
W_{env}	Diagonal weight matrix for collision avoidance	θ	Pitch angle
W_{ex}	Diagonal weight matrix for vehicle speed tracking	ϑ	Parametrised centreline position
$W_{ex,0}$	Initial value of the diagonal weight matrix for collision avoidance	λ	Scheduling parameter
$W_{F,ij}$	Element of the diagonal weight matrix on the longitudinal tyre forces	$\hat{\lambda}$	Predicted scheduling parameter
$W_{F,0,ij}$	Initial value of the ij element of the diagonal weight matrix on the longitudinal tyre forces	μ	Tyre-road adhesion coefficient
W_{NN}	Neural network weights and biases	μ_x, μ_y	Longitudinal and lateral components of the road adhesion coefficient
W_p	Weights on the polytope vertices	Π	Data set
w_r	Effective road height	ρ	Road curvature
$W_t()$	Membership function	ρ_p	Road curvature at the look-ahead distance
W_{veh}	Diagonal weight matrix for vehicle stability	σ	Tyre slip ratio
x	State vector	σ_{max}	Maximum tyre slip ratio
X, Y	Coordinates in the global reference system	σ_w	Adaptation parameter
X_f	Terminal set	τ	Path parametrisation variable
x_{in}	Initial value of the state vector	φ	Roll angle
x_{min}, x_{max}	Lower and upper boundaries for the state vector	χ	Parameter expressing tyre nonlinearities
X_{min}, X_{max}	Lower and upper boundaries for the X coordinate	ψ	Yaw angle
X_{ref}, Y_{ref}	Reference values of the X and Y coordinates	$\dot{\psi}_{lim}$	Yaw rate threshold
Y_{min}, Y_{max}	Lower and upper boundaries for the Y coordinate	ψ_{ref}	Reference yaw angle
z	Predicted system output vector	$\dot{\psi}_{ref}$	Reference yaw rate
z_b, z_u	Vertical coordinates of the sprung and unsprung masses	ω_{ij}	Angular speed of the ij wheel
z_{ref}	Reference output vector		

Acronyms

ABS	Anti-lock braking system	LTI	Linear time-invariant
AD	Automated driving	LTV	Linear time-varying
ARV	Automated road vehicle	MMAPC	Multi-model adaptive controller
AV	Automated vehicle	MPC	Model predictive control
BS	Back-stepping	MS	Motion sickness
CG	Centre of gravity	NL	Nonlinear
CPH	Constant prediction horizon	NNMPC	Neural network model predictive control
CPU	Central processing unit	NN	Neural network
DoF	Degree of freedom	NNMPC	Neural network model predictive control
DT	Double track	OCP	Optimal control problem
DYM	Direct yaw moment	PI	Proportional integral
FCC	Feedforward-feedback compensation control	PID	Proportional integral derivative
FL	Fuzzy logic	PHFA	Prediction horizon fitting algorithm
FMPC	Flatness model predictive control	PT	Path tracking
FOH	First order hold	RGC	Robust guaranteed-cost
GP	Gaussian process	RMS	Root mean square
H_∞	H-infinity	RTV	Real-time verification
HiL	Hardware-in-the-Loop	SMC	Sliding mode controller
KPI	Key performance indicator	ST	Single track
LPV	Linear parameter-varying	TV	Torque vectoring
LQ	Linear quadratic	VPH	Variable prediction horizon
		ZOH	Zero order hold

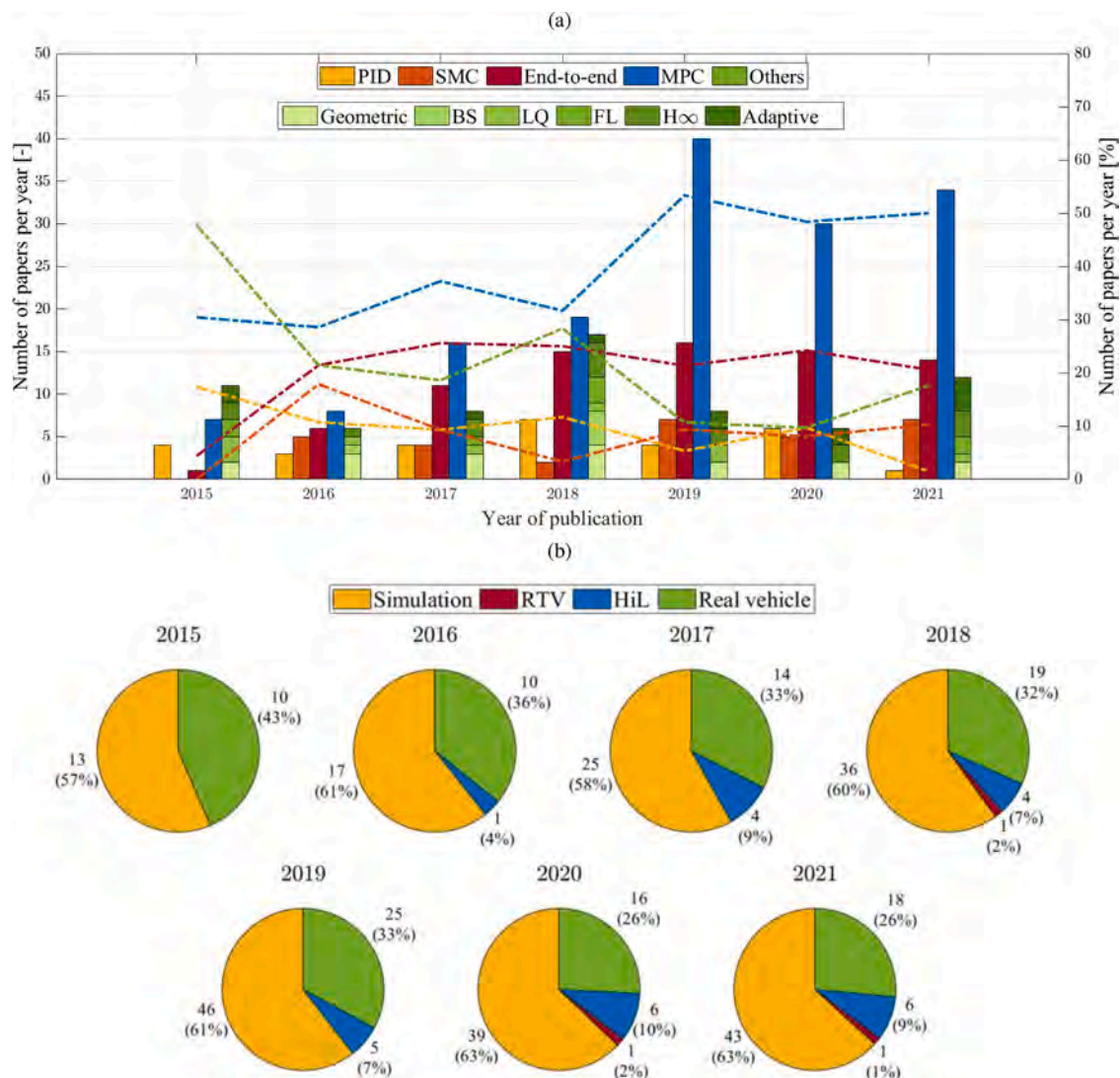


Fig. 1. (a) Distribution of the considered PT controllers for ARVs over time: the histograms refer to the number of papers published per year; the dashed lines refer to the percentage of implementations for each control technique per year; and (b) Controller assessment methods: simulation only, with or without open-loop real-time verification (RTV), HiL, or real vehicle experiments.

- Multi-layer architectures, typically consisting of (Gordon & Lidberg, 2015; Paden et al., 2016; Pendleton et al., 2017):
 - 1 The perception layer, which detects the conditions of the environment surrounding the vehicle, e.g., by identifying possible obstacle-free paths (Gruyer et al., 2017; Rosique et al., 2019).
 - 2 The reference generation layer, also known as motion or path planning layer, which, based on the inputs from the perception layer, defines the reference path and speed profile (Claussmann et al., 2020; González et al., 2016; Huang et al., 2020; Sharma et al., 2021).
 - 3 The control layer, also referred to as motion or path tracking (PT) layer, which defines the actuator commands, typically expressed in terms of steering angle and traction/braking torques (L. Li et al., 2021; Rokonuzzaman et al., 2021; Sorniotti et al., 2016) to follow the reference path and speed profile.
- End-to-end strategies based on neural network (NN) applications for AVs (Kuutti et al., 2021; Ma et al., 2020; Tampuu et al., 2020), where a single algorithm generates the steering and acceleration commands starting directly from the sensor measurements (J. Chen et al., 2021).

This review focuses on the PT layer of multi-layer architectures, which is a deeply studied topic. The main requirements of PT control are (Dixit et al., 2018; Rupp & Stolz, 2016):

- Operating range, i.e., the controller should operate with desirable performance over the entire range of vehicle speeds, as well as longitudinal and lateral accelerations, including the non-linear cornering response regions.
- Robustness, i.e., the controller must operate effectively also in case of vehicle parameter variations, for the whole range of environmental/driving conditions and external disturbances.
- Tunability, i.e., the controller must be associated with a systematic and predictable parameter calibration procedure.
- Real-time capability, i.e., the controller must be real-time implementable on available control hardware, at a suitable frequency to be effective on a real vehicle.

While PT control for low-speed AD operation has been extensively addressed (Wetzenig & Horn, 2016), controllers for high-speed manoeuvring at the limit and beyond the limit of handling are still under investigation, with implementations already showing promising results (Funke et al., 2016; Hajiloo et al., 2021; Rosolia & Borrelli, 2020).

By considering a comprehensive set of papers published from 2015 to 2021, Fig. 1(a) shows the distribution of PT controllers for AVs, based on the publication year of the respective study, and the adopted control technique. The histograms report the number of papers published per year per control category, while the lines refer to the respective annual percentage. The studies covered by the figure meet the following criteria, which are used also in the remainder of this review:

- Only journal and conference papers are considered.
- Only automated road vehicles (ARVs) with non-articulated layouts are included, e.g., robots, agricultural, aerial, and marine AVs, as well as automated articulated heavy goods vehicles, are excluded.
- Only publications in English language are included.

Moreover, in Fig. 1(a):

- The control techniques that are at least once covered by more than five papers published in the same year are considered individually, while the remaining controllers are grouped under the category “Others”.
- Papers dealing with a combination of multiple techniques are included individually within the category of the main control technique that is responsible for the PT control input.

- Papers comparing two or more PT strategies are accounted for in each category.

The histogram plot highlights that the research topic of PT for ARVs has significantly grown in recent years, with a peak reached in 2019. A reason for the partial reduction of PT-related papers in 2020 and 2021 could be the lack of ability of presenting innovative controllers in comparison with the previous publications. A second clear trend is that MPC is the most widely used control technique for PT, and covers ~50% of the publications of the last three years.

Fig. 1(b) highlights the controller assessment methods, i.e., only based on simulations, potentially including some form of open-loop real-time verification (RTV), or hardware-in-the-loop (HiL) testing, or real vehicle implementation. When simulations are used concurrently with HiL or experimental vehicle tests, in Fig. 1(b) the controller validation is respectively classified only as HiL or real vehicle assessment, according to the most advanced form of validation adopted within the same study. In the majority of the research literature of the last few years, vehicle dynamics simulations are the only form of path tracking control assessment; in fact, state-of-the-art control strategies, e.g., end-to-end, tend to be initially evaluated through numerical simulations, before being transferred – if the results are promising – to experimental vehicle demonstrators.

The recent literature reviews by L. Li et al. (2021) and Rokonuzzaman et al. (2021a) compare the different control techniques for PT, and discuss why MPC is highly competitive with respect to (w.r.t.) the other possible solutions. MPC implementations based on dynamic vehicle models including tyre slip behaviour allow for better tracking performance (Zu et al., 2018), especially during high speed and high slip angle conditions, than the kinematic or geometry-based PT methods, such as the pure pursuit (Gamez Serna et al., 2017) and Stanley (Zhu et al., 2016) methods. MPC is also capable of concurrently and effectively managing a greater number of variables and control inputs than more basic control structures, such as proportional integral derivative (PID) controllers, and therefore is ideal for MIMO (multiple-input multiple-output) systems, e.g., for AVs with multiple actuators. Another advantage of MPC over other PT strategies, such as those based on PID, linear quadratic (LQ), fuzzy logic (FL), H_∞ , and backstepping (BS) controllers, is that MPC can easily and systematically consider constraints on the states, actuation commands and their rates, and on functions of the states and control inputs. Hence, for example, MPC enables smooth control action, without the chattering that can jeopardise the effectiveness of sliding mode controllers (SMCs) (Hwang et al., 2018; Y. Wu et al., 2019). MPC does not require specific assumptions on the system model; this is an advantage compared to adaptive controllers, which commonly require the plant model to satisfy conditions (e.g., matching conditions) that can limit their applicability to PT control (Dixit et al., 2021). Differently from end-to-end strategies, MPC, as well as the other classical control techniques, do not require training with large amounts of real-world data, and, depending on the formulation, can enable systematic stability and robustness analyses of the closed-loop system.

In summary, the main benefits of MPC solutions are (see Rokonuzzaman et al., 2021; Sorniotti et al., 2016; S. Yu et al., 2021):

- Systematic management of a large number of variables and constraints (Cao et al., 2021; Xiang et al., 2021; W. Zhang et al., 2020).
- Possibility of considering multiple actuators and models at different levels of complexity within the same control design framework (Hamid et al., 2017b; Ren et al., 2018; H. Wang et al., 2021a; H. Yuan et al., 2018).
- Enhanced tracking performance at medium-to-high lateral accelerations and during emergency conditions, achievable through appropriate selection of the prediction model complexity (Funke et al., 2016; Hajiloo et al., 2021; Rosolia & Borrelli, 2020).

The main disadvantage of MPC algorithms is the computational cost of the online solution of their constrained optimisation problems (J. Lee & Chang, 2018; Liang et al., 2021; Siampis et al., 2018). However, the last generations of solvers (e.g., ForcesPRO, see Alsterda et al., 2019, Kabzan et al., 2019, and Suh et al., 2018; qpOASES QP, see Chowdhri et al., 2021, Qin et al., 2021, and Seccamonte et al., 2019; PRESAS, see Berntorp et al., 2019, and Quirynen et al., 2018) and control hardware solutions (e.g., dSPACE MicroAutoBox II, see Pereira et al., 2017, Rosolia et al., 2017, and Suh et al., 2018; NI PXI, see Luan et al., 2020, and H. Wang et al., 2021b) help mitigate the problem of the real-time capability compared to the past. Another important factor influencing the computational burden is the optimal control problem (OCP) formulation. Siampis et al. (2018) shows that by softening the constraints and reducing the number of solver iterations, the computational time is reduced without significant performance loss at the limit of handling.

Despite the broad literature on MPC implementations for PT control, there is a lack of systematic reviews on the topic. For example, S. Yu et al. (2021) is a literature review on MPC applications for motion planning and tracking, for single and multiple automated ground vehicles. However, the paper lacks a detailed analysis on PT, e.g., on the OCP and prediction model formulations, and their effect on performance and computational effort. Paden et al. (2016) is a survey on motion planning and control techniques for ARVs, and includes a simulation-based comparison of simple PT controllers based on kinematic models, which is followed by the discussion on other approaches, such as predictive and linear parameter-varying controllers. Sorniotti et al. (2016) provides an overview of PT controllers for AVs, ranging from basic kinematic controllers to robust MPC, and includes the analysis of a selection of experimental results on full-scale vehicles. However, both Paden et al. (2016) and Sorniotti et al. (2016) were published in 2016. Hence, given the high number of papers on the topic in the last few years, see Fig. 1, a more recent survey is required. Although the more recent reviews by L. Li et al. (2021), Rokonuzzaman et al. (2021a) and Yao et al. (2020) highlight the benefits and drawbacks of the individual PT controllers, they consider an extensive period of time, and exclude a significant part of the last generation of implementations. In conclusion, the existing reviews: a) do not comprehensively cover the most recent publications on PT control for ARVs; and b) do not include detailed analyses of the different aspects of MPC PT formulations, e.g., the effect of prediction horizon and prediction model on performance and computational effort.

This literature review targets the identified gap, and provides categorisation, description, and critical analysis of the MPC solutions for PT, including prediction models, cost functions, constraints, computational power demand, performance indicators, and achieved results. The considered PT control papers deal with: i) lateral vehicle dynamics control, commonly based on steering actuation (e.g., see Leman et al., 2019; Pereira et al., 2017; Z. Tang et al., 2019), with the option of using further actuators, e.g., rear-wheel-steering and/or wheel torque distribution control (Berntorp et al., 2019; Qin et al., 2021; W. Zhang et al., 2020); and ii) combined longitudinal (through traction and braking force control) and lateral control (e.g., see Karimshoushtari et al., 2021; Liang et al., 2021b; Peng et al., 2019; Ren et al., 2018; C. Zhang et al., 2019). On the contrary, the papers dealing with longitudinal PT control on its own are not the object of this review.

The remainder is organised as follows: Section 2 classifies the MPC typologies; Section 3 describes the prediction models; Section 4 deals with the cost functions, constraints, weight scheduling solutions, and PT architectures for ARVs with multiple chassis actuators; Section 5 summarises the solvers, hardware solutions, use cases, key performance indicators, and main results; Section 6 outlines the development trends, which is followed by the conclusions in Section 7.

2. Model predictive control typologies

2.1. Optimal control problem

At each time step j_c , an MPC algorithm computes an optimal control input sequence U that minimises a cost function accounting for the prediction of the system dynamics over a predefined horizon H_p , while considering system constraints (Camacho & Bordons, 1999; Grüne & Pannek, 2011; Kouvaritakis & Cannon, 2016). The discrete form of the OCP formulation is (J. Guo et al., 2017; M. Kim et al., 2021; Z. Wang, Bai, et al., 2019; H. Wu et al., 2020):

$$\begin{aligned} \min_U J &:= J_{terminal} + J_{stage1} + J_{stage2} \\ &= \frac{1}{2} \| z_{N_p} - z_{ref,N_p} \|_{P_x}^2 + \frac{1}{2} \sum_{k=j_c}^{j_c+N_p-1} \| z_k - z_{ref,k} \|_{Q_x}^2 + \frac{1}{2} \sum_{k=j_c}^{j_c+N_c-1} \| u_k \|_R^2 \end{aligned} \quad (1)$$

s.t.

$$x_0 = x_{in} \quad (1a)$$

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

$$z_k = g(x_k, u_k, p_k) \quad (1c)$$

$$u_{min} \leq u_k \leq u_{max} \quad (1d)$$

$$x_{min} \leq x_k \leq x_{max} \quad (1e)$$

$$h(x_k, u_k) \leq 0 \quad (1f)$$

where J is the cost function, consisting of a terminal cost, $J_{terminal}$, which aims to minimise the response error at the end of the prediction horizon, and a stage cost contribution, corresponding to $J_{stage1} + J_{stage2}$, which aims to optimise the response along H_p ; j_c is the current time step; k indicates a step along the prediction horizon; N_p is the number of steps of H_p , i.e., $H_p = N_p T_s$, with T_s being the sampling time used for discretising the OCP; N_c is the number of steps of the control horizon H_c , i.e., $H_c = N_c T_s$; z is the vector of the predicted system outputs, whose corresponding reference vector is z_{ref} ; P_x , Q_x and R are positive diagonal weighting matrices; x is the state vector; u is the control input vector; $U = [u_{j_c} \ u_{j_c+1} \ \dots \ u_{j_c+N_c-1}]$ is the decision variable vector; p is the parameter or external disturbance vector; x_{in} is the initial value of the state vector; f is the discretized vector field describing the prediction model; g is the function expressing the system outputs; u_{min} and u_{max} , and x_{min} and x_{max} are respectively the bounds on the control action u and state vector x ; and h is the inequality constraint function.

The majority of MPC-based PT applications have $N_c < N_p$ (e.g., Y. Chen et al., 2020; Choi et al., 2018; Hamid et al., 2017b; S. Li, S. Wang et al., 2019; Y. Xu et al., 2021; Yakub et al., 2016). A longer prediction horizon tends to reduce the oscillations of the closed-loop system, thus improving the PT performance (C. Hu & Zhao, 2021; H. Wang et al., 2021b; Z. Wang, Bai, et al., 2019). However, if H_c increases for a given H_p , vehicle stability tends to decrease (Z. Wang, Bai, et al., 2019; B. Zhang et al., 2019b). Shorter control horizons also have the benefit of reducing the computational load, thus enabling longer prediction horizons. The impact of N_p , N_c , and T_s on the PT results is discussed in Section 5.4.1.

2.2. MPC typologies used for path tracking control

In the framework of the OCP formulation in (1), different MPC typologies have been adopted for PT control:

- Linear time-invariant MPC (LTI-MPC), using a discrete linear state-space model (C. Huang et al., 2017; Kouvaritakis & Cannon, 2016;

Lu et al., 2018; Shen et al., 2017; Yakub & Mori, 2015), typically obtained from the continuous state-space formulation of the system through zero-order or Euler discretisation:

$$x_{k+1} = Ax_k + Bu_k + Cp_k \quad (2)$$

where A , B and C are the dynamic, control input and external disturbance matrices, which remain constant during controller operation. Equivalent LTI systems, within a control architecture defined as flatness MPC (FMPC), are proposed by Z. Wang et al. (2019, 2020). The flatness algorithm is based on the expression of the system inputs and states as functions of flat outputs and their finite-order derivatives, with the aim of converting a nonlinear model into an equivalent LTI model without using any local linearisation. The idea is to maintain the nonlinear features with reduced computational load w.r.t. a nonlinear model. However, the complex tyre-road friction forces, which are nonlinear functions of tyre slip angles and slip ratios, are not modelled in the high-level FMPCs (i.e., which compute the total longitudinal and lateral forces and yaw moment at the vehicle level) in Z. Wang et al. (2019, 2020), as they are the control inputs, and are converted into each wheel's torque and steering commands by a separate low-level control allocation algorithm.

- Linear time-varying MPC (LTV-MPC), based on discrete linear state-space models, whose matrices are obtained by linearising the system dynamics when the controller is called. Therefore, A , B and C vary with time during controller operation, but remain constant along the prediction horizon (Y. Huang, et al., 2019; K. Yuan et al., 2018). The discretized state-space model can be written as (Y. Chen et al., 2020; H. Wang, 2019; Xiang et al., 2020):

$$x_{k+1} = A_{j_i}x_k + B_{j_i}u_k + C_{j_i}p_k \quad (3)$$

with A_{j_i} , B_{j_i} and C_{j_i} being the system matrices for the j_i -th linearisation of the model, where $j_i = j_c$ holds in most implementations.

- Linear parameter-varying MPC (LPV-MPC), based on prediction models that are linear in the state-space, but nonlinear in the parameter space. The linear state transition map depends on the scheduling variable $\lambda = \lambda(k)$, which is bounded and known at the beginning of the prediction horizon, e.g., through online parameter measurements or estimations (Morato et al., 2020). The state-space formulation can be rewritten as:

$$x_{k+1} = A(\lambda(k))x_k + B(\lambda(k))u_k + C(\lambda(k))p_k \quad (4)$$

The system response in the N_p steps ahead of the current instant j_c directly depends on the vector of the future scheduling parameters, $\Gamma_{j_c} = [\lambda(j_c + 1) \lambda(j_c + 2) \dots \lambda(j_c + N_p - 1)]$. Because of the unavailability of the future scheduling values, LPV-MPC approaches opt for a prediction guess $\hat{\Gamma}_{j_c}$ (Morato et al., 2020), which often results in the assumption of a frozen guess, i.e., $\hat{\lambda}(k) = \lambda(j_c)$ for $k = j_c, \dots, N_p - 1$, enabling the use of simplified LTI models. However, the feasibility and closed-loop stability properties expected from LPV-MPC require the design of formulations that are formally robust against the possible future scheduling parameter variations (Hanema, et al. 2017). Some LPV approaches express the nonlinear model of the system as a combination of linear models, see Sename et al. (2013) and Gimondi et al (2021), where the latter – although not using MPC – explicitly reports the formulation of an LPV single track model. Liang et al. (2021a, 2021b) propose a multi-model MPC approach that assumes the pair $[A(\lambda(k)), B(\lambda(k))]$ to be within a polytope Ω along the entire prediction horizon. Any pair is a convex combination of the four polytope vertices such that:

$$A(\lambda(k))x_k + B(\lambda(k))u_k = \sum_{i_p=1}^4 W_{p,i_p} [A_{i_p}x_k + B_{i_p}u_k] \quad (5)$$

where $W_{p,i}$ is a weighting variable that determines how much each vertex LTI-model of the polytope represents the uncertain LPV model for the future N_p steps. In the Takagi-Sugeno approach (Takagi & Sugeno, 1985) proposed by Alcalá (2019a, 2019b), the weighting variable is based on fuzzy logic, and $A(\lambda(k))$ is computed as:

$$A(\lambda(k)) = \sum_{i_r=1}^{r_c} W_{r,i_r}(\lambda(k))A_{i_r} \quad (6)$$

where r_c is the number of scheduling variables; and $W_{r,i_r}(\lambda(k))$ is the relevant membership function.

- Nonlinear MPC (NMPC), based on nonlinear state-space models, typically accounting for the non-linearities of the vehicle and tyres (Grüne & Pannek, 2011). The state-space model is defined accordingly to (1b), where f is a nonlinear discrete vector field. The benefit is a more accurate prediction of the system dynamics, at the price of increased computational load, which can limit the real-time implementability (Ren et al., 2018; Rokonzuzaman et al., 2020; W. Zhang et al., 2020).
- Hybrid MPC (HMPC), where a logic-based decision module can be integrated into the algorithm to switch among a family of controllers (Liberzon, 2003). For example, K. Zhang et al. (2015) proposes an HMPC that switches between a kinematic and a dynamic prediction model. The switching conditions are determined by the divergence of the plant w.r.t. the two models. The aim is to reduce the computational load while consistently achieving the performance of an accurate dynamic vehicle model.
- Neural network MPC (NNMPC), i.e., a newly developed MPC typology based on a neural network (NN) prediction model, which learns and predicts the vehicle dynamics from the measured states and input variables, q_k . The discretized NN model can be expressed as (Rokonzuzaman et al., 2021b):

$$x_{k+1} = f_{NN}(q_k, W_{NN,k}) \quad (7)$$

where f_{NN} is the NN-based vehicle model function, and W_{NN} is the set of weights and biases. The use of NN-based prediction models enables the path tracking NNMPC: i) to capture the tangential tyre force variation as a function of the tyre-road friction coefficient with increased accuracy and without the need for a tyre-road friction level estimator (Spielberg et al., 2021); and ii) to more accurately predict the system dynamics in presence of uncertainties in the vehicle parameters, e.g., the vehicle mass, w.r.t. rigidly defined vehicle models (Rokonzuzaman et al., 2021b), while still remaining computationally efficient for real-time implementation (Spielberg et al., 2021).

The previous MPC typologies can be used in specific applications targeting controller robustness or the implementation of learning algorithms, thus giving origin to two further categories:

- Robust MPC, which aims to tackle the parameter uncertainties and disturbances caused by non-linearities, external sources, and driving conditions, by guaranteeing stability and controller performance (Kouvaritakis & Cannon, 2016; Mata et al., 2019; Peng et al., 2019; J. Yu et al., 2019). For example, Peng et al. (2019) uses robust MPC within a coordinated PT and direct yaw moment (DYM) controller. A polytopic uncertain state-space linear parameter-varying (LPV) discrete model accounts for the uncertainty of the tyre cornering stiffness and the time-varying vehicle speed. Four polytypic vertices cover the possible conditions of the state matrix, and are updated

according to vehicle speed. The result is that the predicted LPV model can accurately reflect the influence of time-varying parameters. A different robust approach is adopted by Mata et al. (2019) to deal with the typical mismatch of oversimplified models in limit handling conditions. A tube-based technique compensates for the difference between real and nominal predicted trajectories due to the unmodeled dynamics, whose effect is included through unknown disturbances. A local low-level controller forces the nominal states to be within a more restrictive set, defined as a trajectory tube, which is achieved by considering the initial time step of the nominal states as a decision variable.

- Learning MPC, which embeds experimental data previously collected by the plant in the OCP formulation (Hewing et al., 2020). The learning strategy can be simply used to adjust the cost function weights (Rokonuzzaman et al., 2020), or – in case of autonomous racing applications – to define a minimum time iterative control task (Brunner et al., 2017; Kabzan et al., 2019; Rosolia et al., 2017; Rosolia & Borrelli, 2019). In this context, Rosolia et al. (2017) presents a learning MPC strategy, where the terminal cost and constraints are updated at each iteration l , corresponding to the completion of a racetrack lap, with the aim of improving controller performance until the lap time converges to a local optimal solution. At each time t of the l -th iteration, the controller solves the following learning OCP:

$$J_{t \rightarrow t+N_p}^{LMPC,l}(x_k^l) = \min_{u_k^l} \left[\sum_{k=j_c}^{j_c+N_p-1} h^l(x_{k|t}^l, u_{k|t}^l) + Q^{l-1}(x_{k+N_p|t}^l) \right] \quad (8)$$

where the sum of the stage costs h^l aims to minimise the time to drive the system to the terminal set X_f , i.e., the end of the current lap, while the terminal cost Q^{l-1} , updated at each iteration based on data from previous laps, is designed to guarantee the recursive feasibility and stability of the successive iterations, as well as non-increasing iteration cost, thus satisfying the convergence property (Rosolia & Borrelli, 2017). Kabzan et al. (2019) presents a similar PT approach for autonomous racing, which minimises the lap time through a terminal cost that is function of the collected data. An online data driven MPC uses Gaussian processes (GP) regression to account for the residual model uncertainty, and achieve safe driving behaviour. The aim is to improve performance and enable automatic model adaptation by calculating the error between nominal model and measurement data during operation, and updating the input matrix B accordingly. The vehicle position constraints are imposed through a first term linked to the track radius at the parametrised centreline position θ_k , and a second term that includes the model uncertainty variance E^x . θ_k and E^x are precomputed based on the trajectory of the previous solution, and the optimisation problem can be solved by using a predetermined data set that is continuously updated.

2.3. Summary

Table 1 reports an overview of the different MPC approaches for PT control, and the respective control inputs. The front steering angle δ is the most frequently used control input. Alternative formulations define the control action in terms of time derivative of the steering angle (Chowdhri et al., 2021; Ercan et al., 2017; Laurence & Gerdes, 2021; Wurts et al., 2021), or in terms of steering torque (Ercan et al., 2017), to directly account for the steering actuation dynamics. A widely explored case is the adoption of integrated systems controlling multiple actuators to improve the PT performance (J. Guo et al., 2017; Peng et al., 2019; Yakub et al., 2016), and/or to combine the control of lateral and longitudinal dynamics (Brunner et al., 2017; C. Hu et al., 2020; K. Yang et al., 2021), see Section 4.4 for a summary on the available multi-actuator architectures.

Table 1
Overview of the MPC-based PT formulations and respective control inputs.

Controller type	Control input	References
Linear MPC	δ	J. Cao et al., 2020; S. Chen & Chen, 2020; S. Li et al., 2020; S. Li, G. Wang et al., 2019; S. Li, S. Wang et al., 2019; Lu et al., 2018; Shen et al., 2017
	$\dot{\delta}$	Nam et al., 2019
	δ, v	Ye et al., 2019
	$\delta, F_{x,tot}$	H. Wang, Huang et al., 2019
	δ, δ_r, M_z	Yakub et al., 2016; Yakub & Mori, 2015
	$\delta, \delta_r, T_{fl}, T_{fr}, T_{rl}, T_{rr}$	C. Huang et al., 2016
	Linear time-varying MPC	δ
F_{yf}		Brown et al., 2016; Eriien et al., 2016; Funke et al., 2015, 2016; C. Sun et al., 2018, 2019; Yao & Tian, 2019
$\dot{\delta}, \dot{\delta}_r$		Wurts et al., 2021
δ, v		C. Zhang et al., 2019
δ, a_x		Cesari et al., 2017; Kong et al., 2015; Luciani et al., 2020; Taherian et al., 2019; Vallon et al., 2017
$\dot{\delta}, a_x$		Qian et al., 2016
$\delta, F_{x,tot}$		Y. Huang et al., 2019; K. Yang et al., 2021
δ, T_r		Y. Chen et al., 2019; Y. Chen & Wang, 2019; Z. Wang & Wang, 2020
δ, T_{tot}		Farag, 2020
δ, M_z		J. Guo et al., 2017; H. Wang et al., 2021a; H. Wang & Liu, 2021; Y. Zou et al., 2019
F_{yf}, M_z		Hajiloo et al., 2021
$\delta, F_{x,tot}, M_z$		Xiang et al., 2020
$F_{y,tot}, M_z, F_{x,tot}$		Z. Wang, et al., 2020; Z. Wang, Zha et al., 2019
$\delta, F_{x,fl}, F_{x,fr}, F_{x,rl}, F_{x,rr}$		H. Wu et al., 2020; H. Yuan et al., 2018
$\delta, T_{fl}, T_{fr}, T_{rl}, T_{rr}$		M. Cao et al., 2021; Hashemi et al., 2021
$\delta, T_{b,fl}, T_{b,fr}, T_{b,rl}, T_{b,rr}$		Qin et al., 2021
Linear parameter-varying MPC		δ
	δ, a_x	Alcalá et al., 2019a, 2019b; Suh et al., 2018

(continued on next page)

Table 1 (continued)

Controller type	Control input	References	
Nonlinear MPC	$\delta, F_{x,fl}, F_{x,fr}, F_{x,rl}, F_{x,rr}$	Liang et al., 2021b	
	δ	J. Cao et al., 2020; Y. Chen et al., 2020; Codrean et al., 2020; Leman et al., 2019; S. Li, G. Wang et al., 2019; S. Li, S. Wang et al., 2019; F. Lin, Chen et al., 2019; Quirynen et al., 2018; Rafaila & Livint, 2015; Rokonzuzaman et al., 2019; Taghavifar, 2019; K. Zou et al., 2021	
	$\dot{\delta}$	Dallas et al., 2020; J. Liu et al., 2017	
	T_{δ}	Ercan et al., 2017	
	δ, v	Du et al., 2016	
	δ, a_x	Batkovic et al., 2019; Dawood et al., 2020; T. Lee & Kang, 2021	
	$\delta, F_{x,tot}$	M. Chen & Ren, 2017; Ren et al., 2018; Y. Yu et al., 2021	
	$\delta, F_{b,tot}$	Hamid et al., 2017a, 2017b	
	$\dot{\delta}, F_{x,tot}$	Laurense & Gerdes, 2021	
	$\delta, F_{x,r}$	Qi et al., 2021	
	$\dot{\delta}, \dot{\delta}_r$	C. Yu et al., 2021	
	$\delta, F_{x,f}, F_{x,r}$	C. Hu et al., 2020	
	$\dot{\delta}, T_f, T_r$	Berntorp et al., 2019	
	$F_{y,tot}, M_z, F_{x,tot}$	Z. Wang, Zha et al., 2020	
	$\delta, T_{fl}, T_{fr}, T_{rl}, T_{rr}$	W. Zhang et al., 2020	
	$\dot{\delta}, T_{b,fl}, T_{b,fr}, T_{b,rl}, T_{b,rr}$	Chowdhri et al., 2021	
	Hybrid MPC	δ, v	K. Zhang et al., 2015
	Neural network MPC	δ	Rokonzuzaman et al., 2021b
$\dot{\delta}$		Spielberg et al., 2021	
Robust MPC	δ	D. Kim et al., 2018; Mata et al., 2019; J. Yu et al., 2020	
	β	Law et al., 2018	
	δ, M_z	Peng et al., 2019	
	δ, δ_r, M_z	Hang et al., 2021	
Learning MPC	δ	Rokonzuzaman et al., 2020	
	δ, a_x	Brunner et al., 2017; Karimshoushtari et al., 2021; Rosolia et al., 2017; Rosolia & Borrelli, 2019; Vallon & Borrelli, 2021	
	$\Delta\delta, \Delta T$	Kabzan et al., 2019	

3. Prediction models

This section deals with the prediction models to approximate the vehicle behaviour along the prediction horizon. The formulations describing the position of the vehicle w.r.t. the reference path are provided in Section 3.1, which is followed by: i) the physics-based vehicle model formulations, presented according to their increasing complexity level in Sections 3.2-3.4; ii) the tyre modelling approaches, in Section 3.5; and iii) the discussion on neural network prediction models (Section 3.6).

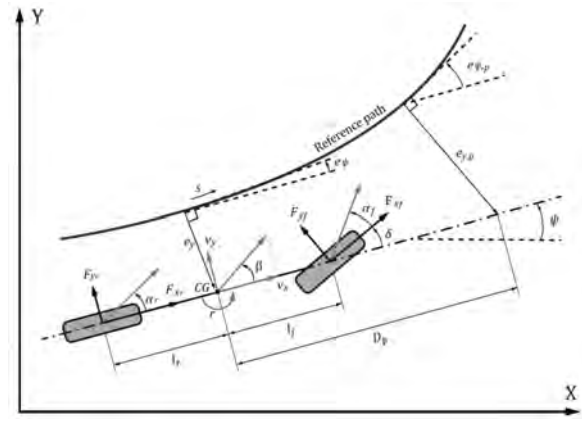


Fig. 2. Single track model schematic.

3.1. Vehicle position with respect to the reference path

Fig. 2 shows the schematic of a generic single track model, also known as bicycle model, commonly adopted for PT applications, which includes indication of the reference path. With a few exceptions (Yakub et al., 2016; Yakub & Mori, 2015; Hang et al., 2021; C. Huang et al., 2016; Wurts et al., 2021), a common assumption is to neglect the rear wheel steering, δ_r . In these conditions, the longitudinal axes of the front and rear wheels form angles equal to $\psi + \delta$ and ψ , w.r.t. the X-axis of the global reference system, with ψ being the heading (or yaw) angle.

A curvilinear coordinate system or the global reference system XY are used to describe the vehicle position w.r.t. the reference path. In the first case, the distance along the path s , the lateral position error e_y , defined as the shortest distance of the centre of gravity (CG) from the reference path, and the heading angle error e_ψ , defined as the body frame orientation w.r.t. the reference path, are commonly selected to describe the relative position between the vehicle and the reference path. In this approach, s , e_y , and e_ψ are system states, which are described by the following equations (Alcalá et al., 2019b; Hatem, 2018, Laurense & Gerdes, 2021; Suh et al., 2018; H. Wang et al., 2021b):

$$\begin{aligned} \dot{s} &= \frac{v_x \cos e_\psi - v_y \sin e_\psi}{1 - \rho e_y} \\ \dot{e}_y &= v_x \sin e_\psi + v_y \cos e_\psi \\ \dot{e}_\psi &= \dot{\psi} - \rho \dot{s} \end{aligned} \quad (9)$$

where v_x and v_y are the longitudinal and lateral components of vehicle velocity; and ρ is the reference road curvature, which is a function of s . As shown in Fig. 2, the lateral position and heading angle errors can also be defined at a look-ahead distance D_p in front of the vehicle centre of gravity, with corresponding reference path curvature ρ_p , according to (Cheng et al., 2020; Liang et al., 2021b; Peng et al., 2019):

$$\begin{aligned} \dot{e}_{y,p} &= v_x \sin e_{\psi,p} + v_y \cos e_{\psi,p} + D_p \dot{\psi} \\ \dot{e}_{\psi,p} &= \dot{\psi} - \rho_p \dot{s} \end{aligned} \quad (10)$$

Liang et al. (2021b) defines D_p as a function of v_x through:

$$D_p = \begin{cases} D_{p,l} & v_x \leq v_{x,min} \\ v_x \Delta t & v_{x,min} < v_x < v_{x,max} \\ D_{p,h} & v_x \geq v_{x,max} \end{cases} \quad (11)$$

where Δt is the look-ahead time; $D_{p,l} = v_{x,min} \Delta t$ and $D_{p,h} = v_{x,max} \Delta t$ are the lower and upper limits of D_p ; and $v_{x,min}$ and $v_{x,max}$ are lower and upper longitudinal speed limits in generating the preview distance.

s , e_ψ and e_y can be replaced by the position of the vehicle w.r.t. a global reference system XY (Geng et al., 2020; Y. Huang, Ding, et al., 2019; K. Yuan et al., 2018; W. Zhang et al., 2020). For example, the time derivatives of the global coordinates X and Y associated with the centre

of gravity can be expressed as:

$$\begin{aligned} \dot{X} &= v_x \cos\psi - v_y \sin\psi \\ \dot{Y} &= v_x \sin\psi + v_y \cos\psi \end{aligned} \quad (12)$$

The vehicle position states can be used in both kinematic and dynamic models. Hence, in all vehicle models of the following sections, the vehicle position states are expressed through (11)-(14) or equivalent formulations w.r.t. the curvilinear coordinate system or the global reference system.

3.2. Kinematic single track model

The kinematic single track (ST) model assumes zero slip angles on the front and rear tyres, and neglects the inertial effects. Based on Fig. 2, the longitudinal and lateral positions of the centre of gravity, X and Y , and the yaw angle ψ can be used as states. The time derivatives of the three states can be expressed as (Kong et al., 2015; Law et al., 2018; L. Tang et al., 2020):

$$\begin{aligned} \dot{X} &= v \cos(\psi + \beta) \\ \dot{Y} &= v \sin(\psi + \beta) \\ \dot{\psi} &= \frac{v}{l_f + l_r} \tan\delta \end{aligned} \quad (13)$$

where v is the vehicle speed; and l_f and l_r are the front and rear semi-wheelbases. The vehicle sideslip angle β can be expressed as a function of the steering angle δ :

$$\beta = \tan^{-1} \left(\frac{l_r \tan\delta}{l_f + l_r} \right) \quad (14)$$

The model in (15)-(16) is limited to the prediction of the lateral motion caused by steering angle control. The combined control on the longitudinal dynamics requires vehicle speed v as an additional state, the derivative of which is expressed by (Cesari et al., 2017, Farag, 2020; T. Lee & Kang, 2021; Taherian et al., 2019):

$$\dot{v} = a \quad (15)$$

where a is the acceleration of the centre of gravity in the same direction as the velocity. To facilitate the design of the constraint on the steering wheel rate, Batkovic et al. (2019) and Qian et al. (2016) define the following additional state:

$$\dot{\delta} = \xi \quad (16)$$

where ξ is the steering angle rate, used as control input.

The main advantage of kinematic model formulations is the low computational effort of the resulting controllers, which are suitable for real-time applications for low-speed (Batkovic et al., 2019; Pereira et al., 2017; H. Yu et al., 2015; C. Zhang et al., 2019) or parking (Ye et al., 2019) manoeuvres. In high speed and/or high slip angle conditions, the accuracy of kinematic models deteriorates w.r.t. dynamic models. For this reason, the most recent model predictive PT applications using kinematic prediction models do not directly control the steering angle, but generate reference signals for other controllers, such as the reference yaw rate, $\dot{\psi}_{ref}$, in Alcalá et al. (2019b) and L. Tang et al. (2020), or the reference longitudinal speed, $v_{x,ref}$, in Alcalá et al. (2019a), or train a neural network, see T. Lee & Kang (2021).

3.3. Dynamic single track model

The dynamic ST model is well-known for the design of vehicle stability and PT controllers (Luan et al., 2020; Rosolia et al., 2017; K. Yuan et al., 2018). In its most frequent declination, under the assumption of constant v_x , the ST model has two degrees of freedom, i.e., the states are the lateral velocity v_y and yaw rate $\dot{\psi}$, and their dynamics are expressed

by (Mata et al., 2019; Y. Xu et al., 2017):

$$\begin{aligned} \dot{v}_y &= \frac{1}{m} [F_{x,f} \sin\delta + F_{y,f} \cos\delta + F_{y,r}] - v_x \dot{\psi} \\ \ddot{\psi} &= \frac{1}{I_z} [l_f F_{x,f} \sin\delta + l_f F_{y,f} \cos\delta - l_r F_{y,r}] \end{aligned} \quad (17)$$

where m is the vehicle mass; I_z is the yaw mass moment of inertia; and $F_{x,i}$ and $F_{y,i}$ are the longitudinal and lateral tyre forces. Alternatively, v_y can be replaced by the vehicle sideslip angle, β , whose dynamics are given by (H. Guo, Cao et al., 2018; Hang et al., 2021; Shen et al., 2017; H. Wang et al., 2021b):

$$\dot{\beta} = \frac{1}{m v_x} [F_{y,f} \cos\delta + F_{y,r}] - \dot{\psi} \quad (18)$$

The choice of β as a state is reasonable under the constant speed assumption (Brown et al., 2016). However, when incorporating the speed dynamics, the derivative of β introduces a nonconvex term, $\beta \dot{v}_x / v_y$, in the governing equation, which is easily avoided by using v_y as a state (Funke et al., 2016).

The inclusion of a further degree of freedom for the longitudinal vehicle dynamics becomes necessary when the MPC is also responsible for generating a reference longitudinal acceleration (Karimshoushtari et al., 2021; Suh et al., 2018; Vallon et al., 2017; Yao & Tian, 2019) or longitudinal force or wheel torque demand (Berntorp et al., 2019; Y. Chen et al., 2019; K. Yang et al., 2021). In this case, the longitudinal force balance equation has the following form (Brunner et al., 2017; Cesari et al., 2017; Rosolia & Borrelli, 2019):

$$\dot{v}_x = \frac{1}{m} [F_{x,f} \cos\delta - F_{y,f} \sin\delta + F_{x,r}] + v_y \dot{\psi} \quad (19)$$

Specific ST prediction models include further degrees of freedom (DoFs), e.g., in terms of roll dynamics (Yakub et al., 2016), and roll and vertical dynamics (Taghavifar, 2019). For example, the 7-DoF prediction model in Taghavifar (2019) considers the vertical dynamics of equivalent left and right unsprung masses, as well as the roll and heave dynamics of the sprung mass:

- Vertical force balance equations of the sprung and unsprung masses

$$\begin{aligned} \dot{v}_{z,b} &= \frac{1}{m_s} [F_{k,l} + F_{k,r} + F_{c,l} + F_{c,r}] \\ \dot{v}_{z,u,l} &= \frac{1}{m_{u,l}} \{F_{k,l} + F_{c,l} + k_t [w_{r,l} - z_{u,l}]\} \\ \dot{v}_{z,u,r} &= \frac{1}{m_{u,r}} \{F_{k,r} + F_{c,r} + k_t [w_{r,r} - z_{u,r}]\} \end{aligned} \quad (20)$$

where m_s is the sprung mass; m_{u,j_s} refers to the unsprung mass of the left and right vehicle sides j_s ; $v_{z,b}$ and v_{z,u,j_s} are the vertical speeds of the sprung and unsprung masses; k_t is the vertical tyre stiffness; w_{r,j_s} is the effective road displacement, assumed to be zero in the implementation; and F_{k,j_s} and F_{c,j_s} are the suspension spring and damper forces, expressed through:

$$\begin{aligned} F_{k,l} &= k_l \left[z_b - \frac{b}{2} \sin\varphi - z_{u,l} \right] \\ F_{k,r} &= k_r \left[z_b + \frac{b}{2} \sin\varphi - z_{u,r} \right] \\ F_{c,l} &= c_l \left[v_{z,b} - \frac{b}{2} \dot{\varphi} \cos\varphi - v_{z,u,l} \right] \\ F_{c,r} &= c_r \left[v_{z,b} + \frac{b}{2} \dot{\varphi} \cos\varphi - v_{z,u,r} \right] \end{aligned} \quad (21)$$

where k_j and c_j are the suspension stiffness and damping coefficient; z_b and z_{u,j_s} are the vertical coordinates of the sprung mass and the

left/right unsprung masses; φ is the roll angle; and b is the track width.

- Roll moment balance equation

$$\ddot{\varphi} = \frac{1}{I_x + m_s h_g^2} \left\{ m_s h a_y \cos\varphi + m_s h g \sin\varphi + \frac{b}{2} [F_{c,l} - F_{c,r} + F_{k,l} - F_{k,r}] \right\} \quad (22)$$

where h is the distance of the centre of gravity from the roll axis; a_y is the lateral acceleration; and g is the gravitational acceleration.

- Longitudinal and lateral force balance equations

$$\begin{aligned} \dot{v}_x &= \frac{1}{m} [F_{x,fl} \cos\delta - F_{y,fl} \sin\delta + F_{x,rl}] + v_y \dot{\psi} - \dot{\theta} v_{z,b} \\ \dot{v}_y &= \frac{1}{m} [F_{x,fl} \sin\delta + F_{y,fl} \cos\delta + F_{y,rl}] + v_x \dot{\psi} + \dot{\theta} v_{z,b} \end{aligned} \quad (23)$$

where $\dot{\theta}$ is the pitch rate; and h_{CG} is the effective height of the centre of gravity. The yaw moment balance equation is the same as (19).

The results in Yakub et al. (2016), which neglects the vertical dynamics of the sprung and unsprung masses and assumes constant speed, show that the inclusion of the roll dynamics in the prediction model improves the resulting vehicle stability and PT performance for high-speed manoeuvres at different tyre-road friction coefficients ($\mu = 0.1$ and 0.7), w.r.t. the case without roll dynamics.

3.4. Dynamic double track model

The dynamic double track (DT) model is an expansion of the dynamic ST model, in which each of the four tyres is individually considered. The state-space formulation of the planar DT model with 3 DoFs (longitudinal, lateral, and yaw) can be expressed as (Hamid et al., 2017b; S. Li, G. Wang et al., 2019; H. Wu et al., 2020; H. Yuan et al., 2018):

$$\begin{aligned} \dot{v}_x &= \frac{1}{m} \{ [F_{x,fl} + F_{x,fr}] \cos\delta - [F_{y,fl} + F_{y,fr}] \sin\delta + F_{x,rl} + F_{x,rr} \} + v_y \dot{\psi} \\ \dot{v}_y &= \frac{1}{m} \{ [F_{x,fl} + F_{x,fr}] \sin\delta + [F_{y,fl} + F_{y,fr}] \cos\delta + F_{y,rl} + F_{y,rr} \} - v_x \dot{\psi} \\ \ddot{\psi} &= \frac{1}{I_z} \left\{ F_{x,fl} \left[l_f \sin\delta - \frac{b_f}{2} \cos\delta \right] + F_{x,fr} \left[l_f \sin\delta + \frac{b_f}{2} \cos\delta \right] - F_{x,rl} \frac{b_r}{2} \right. \\ &\quad \left. + F_{x,rr} \frac{b_r}{2} + F_{y,fl} \left[l_f \cos\delta + \frac{b_f}{2} \sin\delta \right] + F_{y,fr} \left[l_f \cos\delta - \frac{b_f}{2} \sin\delta \right] - [F_{y,rl} + F_{y,rr}] l_r \right\} \end{aligned} \quad (24)$$

where b_f and b_r are the front and rear track widths.

In real tyres, the longitudinal and lateral forces are functions of the vertical load. In Hamid et al. (2017b), Liu & Kang (2021) and H. Yuan et al. (2018), the calculation of the vertical tyre forces, $F_{z,ij}$, where the subscripts $i = f, r$ and $j = l, r$ indicate the individual vehicle corners, is not reported. Nevertheless, one of the main advantages of DT models w.r.t. ST models is the possibility of including the effect of the lateral load transfers in the calculation of $F_{z,ij}$ (Leman et al., 2019; Y. Chen et al., 2020; H. Wu et al., 2020; W. Zhang et al., 2020). A common assumption in PT applications is to neglect the suspension dynamics in the load transfer modelling, e.g., the nonlinear damping behaviour of the suspension shock absorbers (Subsits & Gerdes, 2021). Hence, $F_{z,ij}$ can be calculated as (Chowdhri et al., 2021):

$$\begin{aligned} F_{z,fl} &= F_{z,f}^{stat} - F_{z,x} - F_{z,yf} \\ F_{z,fr} &= F_{z,f}^{stat} - F_{z,x} + F_{z,yf} \\ F_{z,rl} &= F_{z,r}^{stat} + F_{z,x} - F_{z,yr} \\ F_{z,rr} &= F_{z,r}^{stat} + F_{z,x} + F_{z,yr} \end{aligned} \quad (25)$$

where the static contributions are:

$$\begin{aligned} F_{z,f}^{stat} &= \frac{l_r m g}{2[l_f + l_r]} \\ F_{z,r}^{stat} &= \frac{l_f m g}{2[l_f + l_r]} \end{aligned} \quad (26)$$

while the load transfer contributions are calculated as:

$$\begin{aligned} F_{z,x} &= \frac{h_{CG} m [\dot{v}_x - v_y r]}{2[l_f + l_r]} \\ F_{z,yf} &= \frac{m [\dot{v}_y + v_x r]}{b_f} \left[\frac{l_r h_{r,f}}{l_f + l_r} + \frac{K_{\varphi,f} h}{K_{\varphi,f} + K_{\varphi,r} - h m g} \right] \\ F_{z,yr} &= \frac{m [\dot{v}_y + v_x r]}{b_r} \left[\frac{l_f h_{r,r}}{l_f + l_r} + \frac{K_{\varphi,r} h}{K_{\varphi,f} + K_{\varphi,r} - h m g} \right] \end{aligned} \quad (27)$$

where $h_{r,f/r}$ is the front/rear roll centre height; and $K_{\varphi,f/r}$ is the roll stiffness of the front/rear axles. The distance h between the centre of gravity and the roll axis is given by:

$$h = h_{CG} - \frac{l_r h_{r,f} + l_f h_{r,r}}{l_f + l_r} \quad (28)$$

If including the wheel moment balance equations, the DT model has the possibility of splitting the driving and braking torque levels across each axle, and combining steering angle control with wheel torque control (Cao et al., 2021; Hajiloo et al., 2021; Qin et al., 2021; Xiang et al., 2020; W. Zhang et al., 2020), including wheel slip limitation. The additional four degrees of freedom associated with the wheel moment balance equations are expressed as:

$$\dot{\omega}_{ij} = \frac{1}{I_{w,ij}} [T_{ij} - r_{w,ij} F_{x,ij}] \quad (29)$$

where $I_{w,ij}$ is the mass moment of inertia of the wheel ij ; $r_{w,ij}$ is the tyre radius; ω_{ij} is the angular wheel speed; and T_{ij} is the drive or braking torque.

3.5. Tyre models

The most significant and challenging feature of the PT prediction models is the tyre model, since tyre behaviour is highly nonlinear in limit handling conditions, and depends on several factors, such as the tyre-road friction condition, inflation pressure, vertical load, and camber angle (Chowdhri et al., 2021; Hamid et al., 2017a; Liang et al., 2021b).

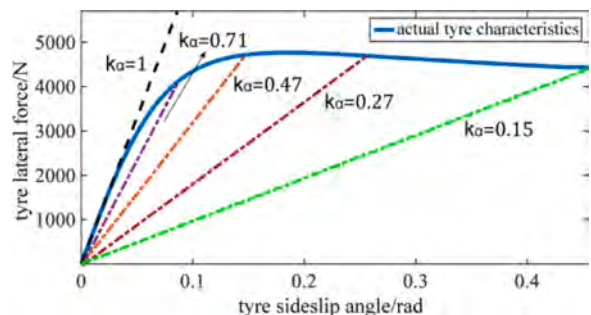


Fig. 3. Tyre cornering stiffness characteristics with adjustment factor (from Peng et al., 2019).

When tyres operate at low slip ratios and slip angles, a linear model can accurately compute the longitudinal and lateral forces (Ren et al., 2018; Xie et al., 2021; Y. Xu et al., 2021):

$$\begin{aligned} F_x &= C_x \sigma \\ F_y &= C_y \alpha \end{aligned} \quad (30)$$

where C_x and C_y are the longitudinal slip stiffness and cornering stiffness of the considered tyre or axle; σ is the slip ratio; α is the slip angle, which, under reasonable assumptions, can be expressed for the front and rear axles through:

$$\begin{aligned} \alpha_f &= -\delta + \tan^{-1} \left(\frac{v_y + \dot{\psi} l_f}{v_x} \right) \approx -\delta + \frac{v_y}{v_x} + \frac{\dot{\psi} l_f}{v_x} \\ \alpha_r &= -\tan^{-1} \left(\frac{\dot{\psi} l_r - v_y}{v_x} \right) \approx -\frac{\dot{\psi} l_r}{v_x} + \frac{v_y}{v_x} \end{aligned} \quad (31)$$

The linear formulations become insufficient when approaching the limit of handling (Chowdhri et al., 2021; S. Li, G. Wang et al., 2019). Embedding a high-fidelity tyre model into the MPC internal model improves prediction accuracy (V. Zhang et al., 2018), at the cost of increasing computational load and affecting real-time feasibility. To find a compromise between accuracy and computational effort, several tyre linearisation approaches have been proposed. For example, a quasi-linear function is used in Peng et al. (2019), see also Fig. 3, which calculates the cornering stiffness as:

$$C_y = k_\alpha(\alpha, \mu) C_{y,0} \quad (32)$$

where $C_{y,0}$ is the nominal cornering stiffness; and k_α , with $0.15 \leq k_\alpha(\alpha, \mu) \leq 1$, is an adjustment factor, which varies as a function of the slip angle α and the tyre-road friction coefficient μ . Although in this formulation the physical meaning of cornering stiffness as gradient of the lateral force w.r.t. the slip angle is lost, the model in (34) can accurately compute the lateral tyre force also in the nonlinear operating region of the tyre or axle.

Similarly, Cheng et al. (2020) uses the following relationship:

$$C_y = C_{y,0} + \chi \tilde{C}_y \quad (33)$$

where \tilde{C}_y is the maximum cornering stiffness correction; and χ , with $|\chi| \leq 1$, is a time-varying parameter that ensures that the resulting cornering stiffness for the specific application ranges from 180 kN/rad to 400 kN/rad.

In S. Li, G. Wang et al. (2019), the lateral axle forces along the prediction horizon are estimated as:

$$F_{y,i} = k_{\mu,F} \frac{m \dot{v}_y l_{r/ff} \pm I_z \ddot{\psi} + m v_x l_{r/ff}}{l_f + l_r} \quad (34)$$

where \dot{v}_y and $\ddot{\psi}$ are expressed as functions of the reference lateral position and heading angle, Y_{ref}^k and ψ_{ref}^k , i.e., they do not depend on the

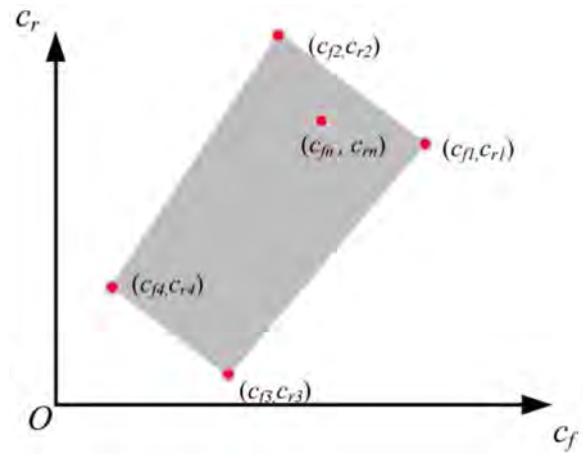


Fig. 5. Convex polytope of tyre cornering stiffness (from Liang et al., 2021b).

predicted states or inputs; and $k_{\mu,F}$ is an adjustment factor. The force from (36) is constrained through:

$$|F_{y,i}| \leq \mu F_{z,i} \quad (35)$$

The predicted slip angles are obtained from the lateral forces, through inverse tyre models expressed as look-up tables that are computed offline:

$$\alpha_i = f_{tyre}^{-1}(F_{y,i}) \quad (36)$$

The future state stiffness and its increment at the step k along the prediction horizon are expressed as:

$$\begin{aligned} C_{y,i}^k &= f_C(Y_{ref}^k, \psi_{ref}^k) \\ \Delta C_{y,i}^k &= C_{y,i}^k - C_{y,i}^{k-1} \end{aligned} \quad (37)$$

where f_C incorporates (36)-(38). Hence, the cornering stiffness and lateral force along the prediction horizon are:

$$\begin{aligned} C_{y,i}^k &= C_{y,i,Lp}^k + \sum_{i=1}^{N_p} \Delta C_{y,i}^k \\ F_{y,i}^k &= C_{y,i}^k \alpha_i^k \end{aligned} \quad (38)$$

where $C_{y,i,Lp}^{k+n}$ is the cornering stiffness from a look-up table based on the estimated slip angle and vertical load at the beginning of the prediction horizon. The expected effect and improvement w.r.t. the linear tyre model case are shown in Fig. 4.

Liang et al. (2021a, 2021b) propose a multiple-model adaptive law for the cornering stiffness uncertainty. A polytope with four vertices covers the possible combinations of front and rear cornering stiffness

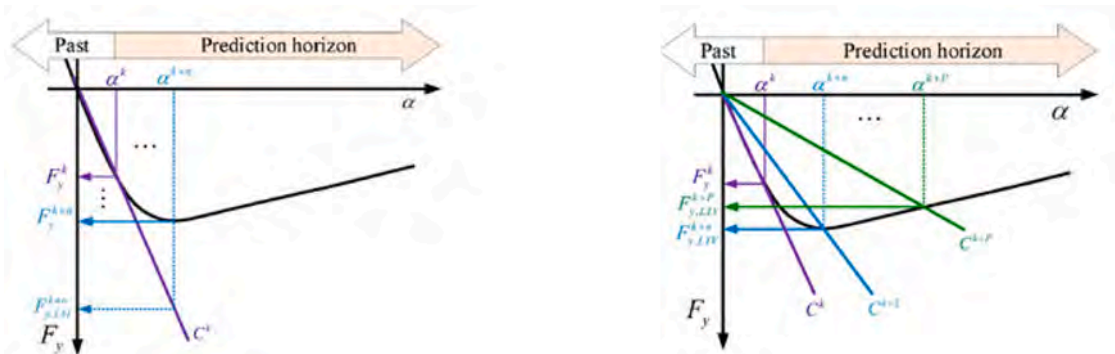


Fig. 4. Constant and variable cornering stiffness models (from S. Li, G. Wang et al., 2019).

values, see Fig. 5. The uncertainties are expressed starting from the four vertices, through the multiple-model theory. The cornering stiffness approximation and the system matrices change according to the weights associated with each vertex.

V. Zhang et al. (2018) uses a linearisation of the brush tyre model to predict the lateral force, through a first-order Taylor series expansion that expresses the force as a function of the vehicle states and control input, i.e., v_y , $\dot{\psi}$, and δ for the front axle:

$$F_{y,f}(\delta, v_y, \dot{\psi}) = F_{y,f}(\delta_0, v_{y,0}, \dot{\psi}_0) + \frac{\partial F_{y,f}}{\partial \delta} \Big|_p [\delta - \delta_0] + \frac{\partial F_{y,f}}{\partial v_y} \Big|_p [v_y - v_{y,0}] + \frac{\partial F_{y,f}}{\partial \dot{\psi}} \Big|_p [\dot{\psi} - \dot{\psi}_0] \quad (39)$$

where $p = (\delta_0, v_{y,0}, \dot{\psi}_0)$ is a generic operating point, and the partial derivatives are rewritten by using the chain rule:

$$\begin{aligned} \frac{\partial F_{y,f}}{\partial \delta} &= \frac{\partial F_{y,f}}{\partial \tan \alpha_f} \frac{\partial \tan \alpha_f}{\partial \delta} \\ \frac{\partial F_{y,f}}{\partial v_y} &= \frac{\partial F_{y,f}}{\partial \tan \alpha_f} \frac{\partial \tan \alpha_f}{\partial v_y} \\ \frac{\partial F_{y,f}}{\partial \dot{\psi}} &= \frac{\partial F_{y,f}}{\partial \tan \alpha_f} \frac{\partial \tan \alpha_f}{\partial \dot{\psi}} \end{aligned} \quad (40)$$

which results into:

$$\begin{aligned} \frac{\partial F_{y,f}}{\partial \tan \alpha_f} &= -C_{y,f} + \frac{2C_{y,f}^2}{3\mu F_z} |\tan \alpha_f| - \frac{C_{y,f}^3}{9\mu^2 F_z^2} |\tan^2 \alpha_f| \\ \frac{\partial \tan \alpha_f}{\partial \delta} &= \frac{-\{v_x^2 + [v_y + l_f \dot{\psi}]^2\}}{\{v_x \cos \delta + [v_y + l_f \dot{\psi}] \sin \delta\}^2} \\ \frac{\partial \tan \alpha_f}{\partial v_y} &= \frac{v_x}{\{v_x \cos \delta + [v_y + l_f \dot{\psi}] \sin \delta\}^2} \\ \frac{\partial \tan \alpha_f}{\partial \dot{\psi}} &= \frac{l_f v_x}{\{v_x \cos \delta + [v_y + l_f \dot{\psi}] \sin \delta\}^2} \end{aligned} \quad (41)$$

A saturation value, ϵ , is added to the expressions of the partial derivatives where appropriate, to avoid that the denominator becomes zero when the vehicle comes to a standstill.

Funke et al. (2016) starts from the modified Fiala model, according to which the lateral axle forces are calculated as:

$$F_y = f_{lyre}(\alpha) = \begin{cases} -C_y \tan \alpha + \frac{C_y^2}{3\eta \mu F_z} |\tan \alpha| \tan \alpha \\ -\frac{C_y^3}{27\eta \mu^2 F_z^2} \tan^3 \alpha & \text{if } |\alpha| < \alpha_{sat} \\ -\eta \mu F_z \text{sign} \alpha & \text{otherwise} \end{cases} \quad (42)$$

$$\text{with } \alpha_{sat} = \tan^{-1} \left(\frac{3\eta \mu F_z}{C_y} \right)$$

where the cornering stiffness, C_y , and friction coefficient, μ , are experimentally determined along ramp steer tests; α_{sat} is the saturating tyre slip angle; and η is a derating coefficient that captures the reduced lateral force capability due to F_x , according to:

$$\eta = \frac{\sqrt{\mu^2 F_z^2 - F_x^2}}{\mu F_z} \quad (43)$$

where the vertical axle loads, $F_{z,i}$, vary according to a simplified longitudinal weight transfer model:

$$F_{z,f} = \frac{1}{l_f + l_r} [ml_r g - h_{CG} F_x] \quad (44)$$

$$F_{z,r} = \frac{1}{l_f + l_r} [ml_f g + h_{CG} F_x]$$

Since in the specific implementation the front lateral force is a control input, the lateral axle force formulation is used only as an inverse model outside the MPC algorithm, to compute the front slip angle, and thus the reference front steering angle. On the contrary, the model in (42)-(44) is embedded in the MPC algorithm for the rear lateral force prediction along H_p . Firstly, the algorithm computes nominal rear slip angles along the prediction horizon, $\bar{\alpha}_r^k$, through a regularisation approach based on an averaging algorithm:

$$\bar{\alpha}_r^k = (1 - r_\alpha) \bar{\alpha}_{r,prev}^k + r_\alpha \alpha_{r,prev}^{*k} \quad k = 1, \dots, n \quad (45)$$

where $\bar{\alpha}_{r,prev}^k$ is the previous rear slip angle prediction; $\alpha_{r,prev}^{*k}$ is the rear slip angle solution at the previous controller execution; and r_α is a coefficient equal to 0.5. (45) significantly reduces the jitter in the generation of the $\bar{\alpha}_r$ values that are used for the linearisation of the nonlinear tyre curve resulting from the Fiala model in (42)-(44) at the current step. Then the linearisation of the tyre characteristic is carried out as an interpolation between $\bar{\alpha}_r^k$ and $\bar{\alpha}_r^{k+1}$, according to Fig. 6. This approach tends to underestimate the force generation capabilities between k and $k+1$, which in turn causes the controller to react earlier to potential threats and stability violations, and improves safety. However, if the slip angle predictions significantly change between successive controller executions, i.e., if $\alpha_r^{*k} < \bar{\alpha}_r^k$, where α_r^{*k} is the current solution, see Fig. 6, the linearisation could overestimate the rear lateral forces. For this reason, in the MPC algorithm trust regions bound the vehicle states to allowable deviations from the pre-computed predictions. Such regions are defined in terms of $\alpha_{r,sat}$ and n_{tr} , so that n_{tr}/r_α control executions are necessary to traverse the entire range of the rear tyre force:

$$\bar{\alpha}_r^k - \frac{2}{n_{tr}} \alpha_{r,sat} \leq \alpha_r^{*k} \leq \bar{\alpha}_r^k + \frac{2}{n_{tr}} \alpha_{r,sat} \dots, n \quad (46)$$

As mentioned in Funke et al. (2016), “when an emergency occurs, the controller will require up to n_{tr}/r_α executions to reach the final solution. The key, however, is that during each of those steps, an accurate model still allows the controller to determine an appropriate input, enabling an immediate response even as the model converges.”

For PT applications at or beyond the limit of handling, the effect of tyre nonlinearities becomes more significant, and to satisfy collision avoidance and stabilisation criteria with linearised models becomes

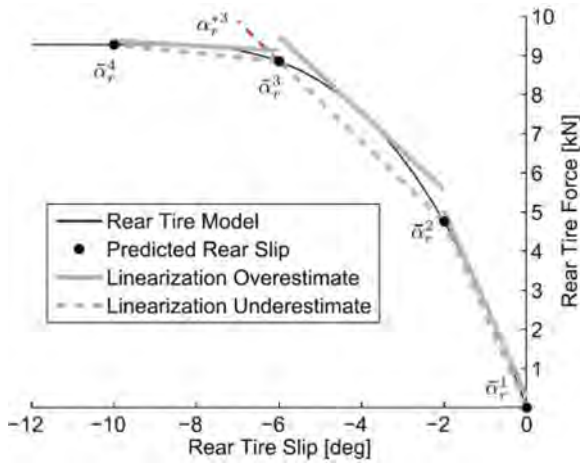


Fig. 6. Predicted operating points of the rear tyres at four time steps, and potential linearization of the rear tyre model for these points (from Funke et al., 2016).

more difficult (Y. Chen et al., 2020). For this reason, nonlinear tyre models with high fitting accuracy have been explored in recent years. Berntorp et al. (2019), Y. Chen et al. (2020), Dallas et al. (2020), Kabzan et al. (2019) and Karimshoushtari et al. (2021) adopt the well-known Pacejka Magic Formula to approximate the longitudinal and lateral slip conditions are expressed as:

$$F_0 = D_{pac} \sin\{C_{pac} \arctan\{B_{pac} \lambda - E_{pac} [B_{pac} \lambda - \arctan(B_{pac} \lambda)]\}\} \quad (47)$$

where λ is generic parameter that indicates the slip ratio σ or slip angle α under pure slip conditions; and B_{pac} , C_{pac} , D_{pac} , and E_{pac} are the stiffness factor, shape factor, peak factor, and curvature factor. The peak factor is a function of the vertical tyre load and the friction coefficient. Under combined slip conditions, the coupling effect between longitudinal and lateral forces can be represented by the friction ellipse (Berntorp et al., 2019; Hamid et al., 2017b):

$$F_{x,i} = F_{0,x} \sqrt{1 - \left[\frac{F_{0,y}}{\mu_y F_z}\right]^2} \quad (48)$$

$$F_{y,i} = F_{0,y} \sqrt{1 - \left[\frac{F_{0,x}}{\mu_x F_z}\right]^2}$$

or a specific declination of the Magic formula (Y. Chen et al., 2020):

$$\begin{cases} F_x = \frac{p_x}{\sqrt{p_x^2 + p_y^2}} F_{x,0} \\ F_y = \frac{p_y}{\sqrt{p_x^2 + p_y^2}} F_{y,0} \end{cases} \quad \begin{cases} p_x = \frac{\sigma_x}{1 + \sigma_x} \\ p_y = \frac{\tan \alpha}{1 + \sigma_x} \end{cases} \quad (49)$$

3.6. Neural network prediction models

In contrast with the physics-based models of conventional MPCs, the prediction models of NN MPC implementations learn the vehicle dynamics from a set of states and variables from vehicle measurements or simulations. The use of comprehensive series of past inputs and states allows multi-layer NN prediction models (Rokonuzzaman et al., 2021b; Spielberg et al., 2021) to capture higher order vehicle dynamics effects, and to predict the vehicle motion on both high and low friction surfaces (Spielberg et al., 2019). Rokonuzzaman et al. (2021b) adopts a feed-forward NN with two hidden layers to predict the time derivatives of the lateral slip speed and yaw rate, which are system states resulting from the time integration of the NN outputs. The inputs to the network are the longitudinal and lateral velocities, yaw rate, and steering angle. Spielberg et al. (2021), see Fig. 7, uses a similar feedforward NN with two hidden layers, with the addition of the front longitudinal force as input, which makes the learned model robust to the effect of wheel slip. The NN model is combined with the kinematic equations for the computation of the e_y and e_ψ dynamics, $x^{(b)} = [e_y, e_\psi]$, resulting into a prediction model combining an NN component, corresponding to f_{NN} , and a conventional physics-based component, corresponding to f :

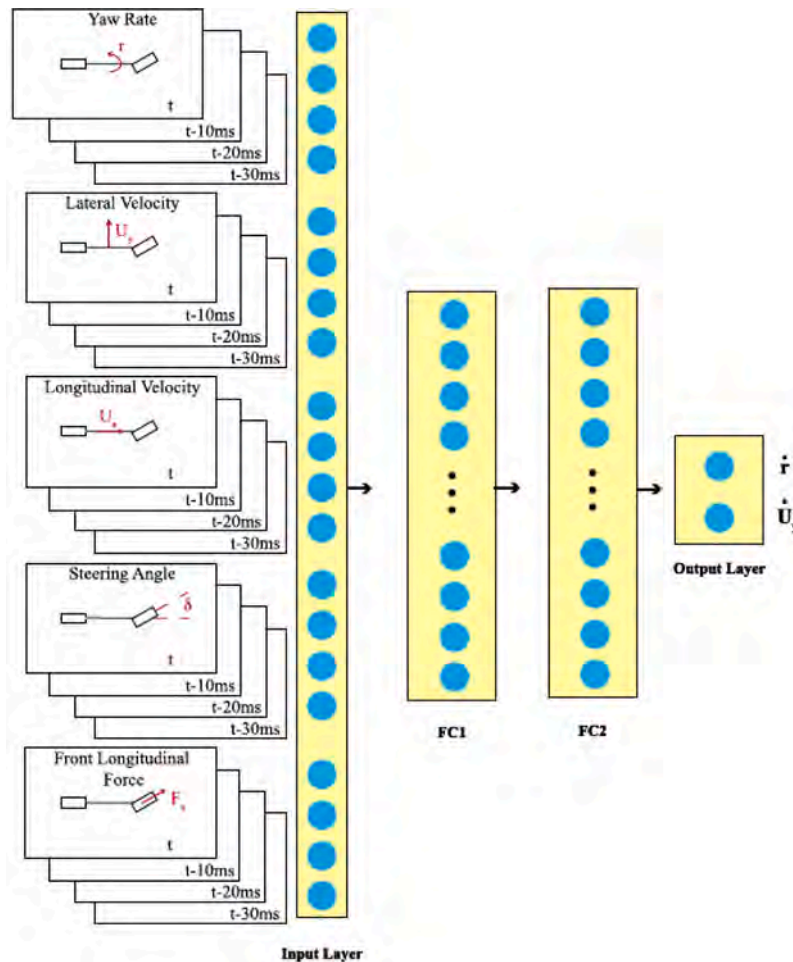


Fig. 7. Neural network prediction model (from Spielberg et al., 2021).

Table 2
Overview of the prediction model characteristics in the considered MPC studies.

Prediction model	Tyre model	States	References
Kinematic single track model		X, Y, ψ	Alcalá et al., 2019a; Du et al., 2016; Law et al., 2018; Lima et al., 2015, 2017; Pereira et al., 2017; Ye et al., 2019; H. Yu et al., 2015; C. Zhang et al., 2019; K. Zhang et al., 2015
		$X, Y, \psi, \rho, \rho_{ref}$	Seccamonte et al., 2019
		v, X, Y, ψ	Kong et al., 2015; T. Lee & Kang, 2021; Taherian et al., 2019
		$v, X, Y, \psi, \delta, \dot{\delta}$	Batkovic et al., 2019
		$\dot{\psi}, e_y, \dot{e}_{yp}, e_{\psi}$	Quan & Chung, 2019
		$v, X, Y, \psi, e_y, e_{\psi}$	Farag, 2020
		e_y, e_{ψ}, s	Hatem, 2018
		$v, e_y, e_{\psi}, s, \delta$	Qian et al., 2016
		v, e_y, e_{ψ}, s, l_x	Cesari et al., 2017
	Dynamic single track model with 2 DoFs	Linear or linearised	$v_y, \dot{\psi}, \psi$
$v_y, \dot{\psi}, Y, \psi$			Cheng et al., 2020; Mata et al., 2019; H. Guo, Shen et al., 2018; Z. He et al., 2020; Liang et al., 2021a; F. Lin, Chen et al., 2019; Y. Xu et al., 2017; Yakub & Mori, 2015; K. Yuan et al., 2018
$\beta, \dot{\psi}, Y, \psi$			Ahn et al., 2021; Bo et al., 2019; Ji et al., 2017; Shen et al., 2017; Sun et al., 2019
$\beta, \dot{\psi}, X, Y, \psi$			Cui et al., 2017
$\dot{e}_y, e_y, \dot{e}_{\psi}, e_{\psi}$			M. Kim et al., 2021; J. Lee et al., 2018; Lu et al., 2018; S. Xu & Peng, 2020; J. Yu et al., 2019
$\dot{\psi}, \dot{e}_{yp}, e_y, e_{\psi}$			D. Kim et al., 2018; Choi et al., 2018
$v_y, \dot{\psi}, e_y, e_{\psi}$			Alsterda et al., 2019; Alsterda & Gerdes, 2021; Dai et al., 2020; Hajiloo et al., 2021; Massera et al., 2020; Yao & Tian, 2019; J. Yu et al., 2020; V. Zhang et al., 2018
Nonlinear		$\beta, \dot{\psi}, e_y, e_{\psi}$	H. Guo, Cao et al., 2018; J. Guo et al., 2017; Peng et al., 2019; Sun et al., 2018; Y. Zou et al., 2019
		$v_y, \dot{\psi}, e_y, e_{\psi}, s$	Funke et al., 2015, 2016
		$\beta, \dot{\psi}, e_y, e_{\psi}, s$	Brown et al., 2016; Erlien et al., 2016; H. Wang et al., 2021a, 2021b; H. Wang & Liu, 2021
		$v_y, \dot{\psi}, Y, e_{\psi}, \delta$	Nam et al., 2019
		$v_y, \dot{\psi}, Y, \psi$	Rafaila & Livint, 2015; K. Zou et al., 2021
		$v_y, \dot{\psi}, X, Y, \psi$	S. Li, S. Wang et al., 2019; Rokonzuzaman et al., 2019
		$v_y, \dot{\psi}, X, Y, \psi, \alpha_y$	Rokonzuzaman et al., 2020
			J. Liu et al., 2017

Table 2 (continued)

Prediction model	Tyre model	States	References
		$v_y, \dot{\psi}, X, Y, \psi, \delta$	Taghavifar, 2019
		$v_y, \dot{\psi}, v_{z,b}, v_{z,u,f}, v_{z,u,r}, \dot{\varphi}$	
Dynamic single track model with 3 DoFs	Linear or linearised	$v_x, v_y, \dot{\psi}, X, Y$	Kong et al., 2015
		$v, \dot{\psi}, X, Y, \psi$	K. Zhang et al., 2015
		$v_x, v_y, \dot{\psi}, X, Y, \psi$	J. Cao et al., 2020; M. Chen & Ren, 2017; S. Chen & Chen, 2020; W. Chen et al., 2021; X. Chen et al., 2021; Y. Chen et al., 2019; Y. Chen & Wang, 2019; Dawood et al., 2020; Feng et al., 2021; Geng et al., 2020; H. He et al., 2021; J. Hu et al., 2020; Y. Huang et al., 2019; F. Lin, Zhang et al., 2019; Luan et al., 2020; Z. Tang, 2018; Ren et al., 2018; H. Wang et al., 2019; Xie et al., 2021; Y. Xu et al., 2021; Y. Yu et al., 2021; B. Zhang et al., 2019a, 2019b
		$v_x, v_y, \dot{\psi}, X, Y, \psi, \delta, \delta_r$	C. Yu et al., 2021
		$v_y, \dot{\psi}, Y, \psi, \dot{\varphi}, \varphi$	Yakub et al., 2016
	Nonlinear	$v_y, \dot{\psi}, e_{v_x}, e_y, e_{\psi}$	Liang et al., 2021b; Z. Wang et al., 2020; K. Yang et al., 2021
		$v_x, v_y, \dot{\psi}, e_y, e_{\psi}, s$	Alcalá et al., 2019b; Brunner et al., 2017; Rosolia et al., 2017; Rosolia & Borrelli, 2019; Suh et al., 2018; Vallon et al., 2017
		$v_x, v_y, \dot{\psi}, e_y, e_{\psi}, a_x$	Luciani et al., 2020
		$v_x, \beta, \dot{\psi}, e_y$	Qi et al., 2021
		$v_x, v_y, \dot{\psi}, X, Y, \psi$	J. Cao et al., 2020; Codrean et al., 2020; C. Hu & Zhao, 2020
Dynamic double track model without load transfer effects	Linear or linearised	$v_x, v_y, \dot{\psi}, X, Y, \psi, \delta$	Dallas et al., 2020
		$v_x, v_y, \dot{\psi}, X, Y, \psi, \delta, T_{tot}$	Kabzan et al., 2019
		$v_x, v_y, \dot{\psi}, X, Y, \psi, \delta_f, \delta_r$	Wurts et al., 2021
		$v_x, v_y, \dot{\psi}, e_y, e_{\psi}, s, \delta$	Laurense & Gerdes, 2021
		$v_x, v_y, \dot{\psi}, e_y, e_{\psi}, s, \delta_h, \dot{\delta}_h$	Ercan et al., 2017
	Nonlinear	$v_x, v_y, \dot{\psi}, e_y, e_{\psi}, X, Y, \psi$	Karimshoushtari et al., 2021
		$v_x, v_y, \dot{\psi}, X, Y, \psi, \delta, \omega_f, \omega_r, \alpha_f, \alpha_r, T_{ef}, T_{lr}, \tau$	Berntorp et al., 2019
		$v_x, v_y, \dot{\psi}, X, Y, \psi$	C. Huang et al., 2016; H. Yuan et al., 2018
		$v_y, \dot{\psi}, e_{v_x}, e_y, e_{\psi}$	Z. Wang, Zha, et al., 2019
		$\beta, \dot{\psi}, \dot{\varphi}, \varphi, Y, \psi$	Hang et al., 2021
	$v_y, \dot{\psi}, \omega_n, \omega_{fr}, \omega_n, \omega_{rr}$	Qin et al., 2021	
	$\alpha_f, \alpha_r, X, Y, \psi$		
	$v_x, v_y, \dot{\psi}, Y, \psi$	Hamid et al., 2017b	

(continued on next page)

Table 2 (continued)

Prediction model	Tyre model	States	References
Dynamic double track model with load transfer effects	Linear or linearised	$v_x, v_y, \dot{\psi}, X, Y, \psi$	Z. Liu & Kang, 2019
		$v_x, \dot{\psi}, X, Y, \psi, \alpha_x$	Hamid et al., 2017a
		$v_y, \dot{\psi}, Y, \psi$	S. Li, G. Wang et al., 2019
		$v_x, v_y, \dot{\psi}, X, Y, \psi$	H. Wang, Huang et al., 2019; Z. Wang, Bai, et al., 2019; H. Wu et al., 2020
		$v_x, v_y, \dot{\psi}$	M. Cao et al., 2021
	Nonlinear	$\omega_{fl}, \omega_{fr}, \omega_{rl}, \omega_{rr}, X, Y, \psi$	Hashemi et al., 2021
		$v_y, \dot{\psi}, \omega_{fl}, \omega_{fr}, \omega_{rl}, \omega_{rr}$	Hashemi et al., 2021
		$\alpha_f, \alpha_r, e_y, e_\psi, \beta, \dot{\psi}, X, Y$	Leman et al., 2019
		$v_x, v_y, \dot{\psi}, X, Y, \psi$	X. Chen et al., 2020
		$v_x, v_y, \beta, \dot{\psi}, X, Y, \psi$	Xiang et al., 2020
Neural network model		$v_x, v_y, \dot{\psi}, X, Y, \psi, \omega_{fl}, \omega_{fr}, \omega_{rl}, \omega_{rr}$	W. Zhang et al., 2020
		$v_x, v_y, \dot{\psi}, X, Y, \psi$	Chowdhri et al., 2021
		$T_{b,flact}, T_{b,fract}, T_{b,rlact}, T_{b,rract}, T_{b,flcal}, T_{b,frcal}, T_{b,rlcal}, T_{b,rrcal}$	Chowdhri et al., 2021
		$v_y, \dot{\psi}, X, Y, \psi$	Rokonuzzaman et al., 2021b
		$v_x, v_y, \dot{\psi}, e_y, e_\psi, \delta, F_{xf}$	Spielberg et al., 2021

$$\begin{aligned}
 x_{k+1}^{(a)} &= f_{NN}(q_k, W_{NN,k}) \\
 x_{k+1}^{(b)} &= f(x_k^{(a)}, x_k^{(b)}, u_k)
 \end{aligned}
 \tag{50}$$

The longitudinal vehicle speed (Rokonuzzaman et al., 2021b) and force (Spielberg et al., 2021) are calculated outside the optimisation problem.

3.7. Summary

As a summary, Table 2 reports the prediction model set-ups in the considered MPC PT literature, in terms of states, tyre model approximation, and inclusion or exclusion of the load transfers.

4. Cost functions, weights, constraints, and controller architectures for multiple actuators

4.1. Cost functions

With respect to the vehicle model in Fig. 2, the lateral control objectives can be typically achieved by defining the MPC cost function in terms of: i) lateral position error, e_y , at the vehicle centre of gravity, w.r.t. the closest point on the reference path; and ii) heading angle error, e_ψ , between the body frame orientation and the tangent to the reference path, evaluated at the centre of gravity. Both e_y and e_ψ are defined w.r.t. a curvilinear reference system. Specific cost function formulations can include the evaluation of these errors at a look-ahead distance, see $e_{y,p}$ and $e_{\psi,p}$ in Fig. 2 (Cheng et al., 2020; Liang et al., 2021b; Peng et al.,

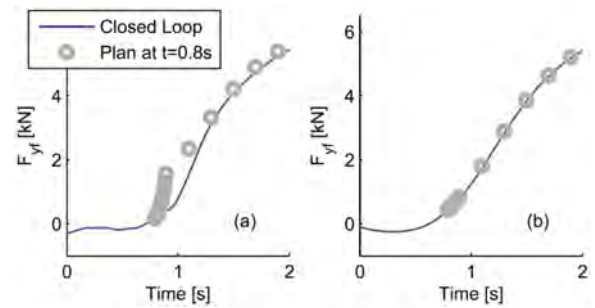


Fig. 8. Simulation results of the planned control input ('Plan') at a given time, and the closed-loop response, with: (a) unscaled weights and ZOH discretisation; and (b) scaled weights and combined ZOH-FOH discretisation (from Funke et al., 2016).

2019), and/or consideration of the time derivatives of the errors, i.e., \dot{e}_y , $\dot{e}_{y,p}$, \dot{e}_ψ and $\dot{e}_{\psi,p}$ (Yakub & Mori, 2015; V. Zhang et al., 2018). An alternative approach is to express the errors w.r.t. an inertial reference system, through the global reference coordinates X_{ref} and Y_{ref} (C. Hu & Zhao, 2020; H. Wang, Liu, et al., 2019; Xie et al., 2021). This solution tends to be limited to applications for manoeuvres, such as the lane change, in which the final heading angle is the same as the initial one.

The longitudinal control task manages the vehicle acceleration to follow the reference speed (Ren et al., 2018; K. Yang et al., 2021; Y. Yu et al., 2021), while keeping a safe distance from the preceding vehicle (Suh et al., 2018; Vallon et al., 2017), and avoiding rear-end collisions (Cui et al., 2019; C. Huang et al., 2016). The cost function is typically expressed in terms of longitudinal speed error, e_{v_x} , defined as:

$$e_{v_x} = v_x - v_{x,ref} \tag{51}$$

where $v_{x,ref}$ is the reference speed generated by the path planning layer.

4.2. Weight selection

The choice of cost function weights and the relaxation level of the constraints allow to fulfil different objectives, e.g., collision avoidance, vehicle stability and PT, by assigning a priority scale. The obstacle avoidance goal must have top priority, to ensure passenger safety in emergency scenarios. For example, Funke et al. (2016) and Hajiloo et al. (2021) select the weights for collision avoidance (W_{env}), vehicle stability (W_{veh}), and path tracking (Q_x) such that:

$$W_{env} \gg W_{veh} \gg \|Q_x\|_\infty \tag{52}$$

For (52) to be meaningful, the weights have to be normalised according to the maximum expected value of the corresponding variable, which is explicitly mentioned in Funke et al. (2016). The prioritisation allows the controller to selectively violate stability criteria if necessary to avoid a collision. Moreover, Funke et al. (2016) scales the weights according to the two adopted time step values along H_p :

$$Q_{k,short} = \frac{T_{s,short}}{T_{s,long}} Q_{k,long} \tag{53}$$

where $T_{s,short}$ and $T_{s,long}$ are the discretisation times for the initial and final part of the prediction horizon. While zero order hold (ZOH) is used for signal discretisation in the initial part of the prediction horizon, first order hold (FOH) is adopted in the second portion of the horizon. To show the effectiveness of the combination of the variable weights and the concurrent ZOH-FOH approach, Fig. 8 compares the simulated planned control input ('Plan') at a given time, and the closed-loop system response with the MPC controller, in terms of front lateral axle force, for: (a) the conventional case of unscaled weights and ZOH; and (b) the proposed weight scaling algorithm with ZOH-FOH combination, which significantly reduces the discrepancy between planned and

closed-loop response.

A different weight selection approach is proposed by B. Zhang et al. (2019a, 2019b). For PT applications, long prediction horizons facilitate vehicle stability in severe driving conditions. However, the corresponding large magnitude of the Hessian matrix of the MPC algorithm tends to generate numerical instability and ill-conditioning, resulting in high sensitivity to even minor disturbances (Emmart et al., 2015). To address this issue, B. Zhang et al. (2019a, 2019b) scale the weights with a descending exponential function along H_p , which provides numerical stability margin by placing less weighting on the future tracking errors and control input increments.

Liang et al. (2021b) defines an online adaptive mechanism on the weights referring to vehicle speed tracking and the longitudinal tyre forces, which are among the control inputs of the specific implementation. The speed tracking weight, W_{ex} , is based on the system outputs, i.e., the lateral position error e_y , the heading angle error e_ψ , and the sideslip angle β , according to a hyperbolic tangent function:

$$Q_s = \max\left(\frac{|e_y|}{e_{y,lim}}, \frac{|e_\psi|}{e_{\psi,lim}}, \frac{|\beta|}{\beta_{lim}}\right)$$

If $Q_s \leq 1$: $W_{ex} = W_{ex,0}$ (54)

else : $W_{ex} = a_s + b_s \tanh\left(\frac{k_s}{Q_s}\right)^{c_s}$

where $e_{y,lim}$, $e_{\psi,lim}$ and β_{lim} are thresholds on $|e_y|$, $|e_\psi|$, and $|\beta|$; $W_{ex,0}$ is the baseline value of W_{ex} ; and a_s , b_s , c_s and k_s are the parameters of the weight adaptation mechanism. When Q_s exceeds 1, which means that the controller struggles maintaining the PT performance or vehicle stability, W_{ex} substantially decreases from $W_{ex,0}$, i.e., velocity tracking becomes a low priority, which facilitates PT and vehicle stabilisation. The weight adaptation on the longitudinal tyre forces aims to prevent the excessive tyre slip ratios σ_{ij} , according to:

if $|\sigma_{ij}| \leq 0.1$: $W_{Fij} = W_{Fij,0}$ (55)

else : $W_{Fij} = W_{Fij,0} e^{\sigma_w [|\sigma_{ij}| - 0.1]}$

where $W_{Fij,0}$ is the baseline longitudinal force weight of each wheel; and σ_w is the adaptation parameter. When $|\sigma_{ij}|$ is larger than 0.1, W_{Fij} sharply grows from $W_{Fij,0}$, which means the corresponding longitudinal tyre force will be rapidly released to prevent excessive longitudinal slip.

Rokonuzzaman et al. (2020) proposes an offline learning strategy based on inverse optimal control to find MPC weights capable of replicating desirable human driving features. To this purpose, the expected profiles of human demonstration tests, i.e., the distance from the centre of the lane, heading angle error, lateral velocity, yaw rate and lateral acceleration, are expressed as:

$$F_\pi = \sum_{k_r=1}^{p_{tr}} \left[\frac{1}{m_{tr}} \sum_{j_r=1}^{m_{tr}} f_\pi(\zeta_{k_r, j_r}) \right]$$
 (56)

where F_π is the feature vector for all considered demonstrations; f_π is the feature vector of a demonstrated trajectory; ζ_{k_r} is the k^{th} trajectory of the data set Π ; and m_{tr} being the number of trials for each driving scenario. For learning the MPC weights Q_x , the expected features of the trajectories generated by the controller are calculated starting from random weights:

$$E(f_i|Q_x) = \sum_{k_r=i}^{p_{tr}} f_i(\zeta_k)$$
 (57)

where f_i is the feature vector for a fixed Q_x . The optimised weight value, Q_x^* , is found through gradient-based optimisation, in which Q_x is updated until convergence to minimise the cost function $\Delta \mathcal{L}(Q_x)$:

$$\Delta \mathcal{L}(Q_x) = F_x - E(f_i|Q_x)$$
 (58)

4.3. Constraints

A strength of MPC is the possibility of systematically considering constraints (Kouvaritakis & Cannon, 2016). In the MPC-based PT implementations, the equality constraints typically refer to the kinematic and dynamic equations of the prediction models in Section 3. The inequality constraints in (1f) can be classified into: i) hard constraints, which are inviolable, and ii) soft constraints, for which violations by a relaxation factor ϵ , also called slack variable, is permitted. ϵ facilitates the feasibility of the solution and the numerical processes associated with the OCP (J. Guo et al., 2017; Z. Tang et al., 2018; B. Zhang et al., 2019a).

The typical inequality constraints for PT applications cover:

- Physical limits of the actuators, i.e., in the form of hard constraints on the control inputs and their variation rates (M. Chen & Ren, 2017; C. Huang et al., 2016; Yakub & Mori, 2015):

$$\begin{aligned} u_{\min} &\leq u \leq u_{\max} \\ \Delta u_{\min} &\leq \Delta u \leq \Delta u_{\max} \end{aligned}$$
 (59)

which typically results into constraints on the steering angle and/or its variation at each discretisation step to account for actuator dynamics, control smoothness (S. Li et al., 2020; Z. Tang et al., 2018; Z. Wang, Bai, et al., 2019; Y. Xu et al., 2021), and indirectly resulting comfort (Alcalá et al., 2019b; F. Lin et al., 2020; V. Zhang et al., 2018):

$$\begin{aligned} \delta_{\min} &\leq \delta \leq \delta_{\max} \\ \Delta \delta_{\min} &\leq \Delta \delta \leq \Delta \delta_{\max} \end{aligned}$$
 (60)

where $\delta_{\max/\min}$ and $\Delta \delta_{\max/\min}$ are the limits on steering angle and steering angle rate.

- Vehicle stability constraints, which can take multiple forms. For example, the lateral acceleration a_y can be limited through hard (F. Lin et al., 2020; Taghavifar, 2019; K. Yuan et al., 2018) or soft constraints (Berntorp et al., 2019; Liu & Kang, 2019), according to the tyre-road friction limits:

$$-\mu g \leq a_y \leq \mu g$$
 (61)

Dai et al. (2020) mentions that this restriction tends to limit δ when the vehicle enters a large curvature turn, which may result in insufficient steering with poor tracking performance. Therefore, it can be convenient to introduce a variable vehicle speed constraint along the prediction horizon to pre-emptively slow down the vehicle, based on the relationship between the maximum lateral acceleration, the known road curvature profile, and vehicle speed:

$$v_{\max} = \sqrt{\frac{a_{y,\max}}{\rho}}$$
 (62)

where v_{\max} is the speed limit. Many authors consider constraints in terms of longitudinal (Y. Chen et al., 2019; Y. Chen & Wang, 2019; Dawood et al., 2020; Suh et al., 2018; H. Wu et al., 2020) and/or lateral speed (Y. Chen et al., 2019; Y. Chen & Wang, 2019; Funke et al., 2015) components, where the latter is equivalent to a sideslip angle constraint. M. Chen & Ren (2017) sets an anti-rollover constraint in terms of maximum vehicle speed to prevent one vehicle side from losing contact with the road. By considering the lateral load transfer, this anti-rollover constraint is expressed as:

$$\mu \left[\frac{mg}{2} - \frac{mh_g |a_y|}{b} \right] \geq \frac{F_r}{2}$$
 (63)

where F_r approximates the total longitudinal resistance force:

$$F_r = mg[f_r \cos \alpha_{road} + \sin \alpha_{road}] \quad (64)$$

with f_r being the rolling resistance coefficient, and α_{road} the road inclination. From (64), the anti-rollover speed limit v_{rl} is given by:

$$v_{rl} = \sqrt{\frac{[mg - \frac{F_z}{\mu}] b}{2mh_g \rho}} \quad (65)$$

Different rollover prevention approaches are proposed by Hang et al. (2021) and Taghavifar (2019). Hang et al. (2021) adopts the phase-plane method:

$$|C_1 \varphi + C_2 \dot{\varphi}| \leq C_3 \quad (66)$$

on top of which a further roll angle constraint is imposed through:

$$|\varphi| \leq \varphi_{max} = \frac{m_s h a_y}{2[K_\varphi - m_s g h]} \quad (67)$$

Taghavifar (2019) sets a constraint in terms of the lateral transfer ratio, LTR , calculated by considering an equivalent single-axle roll model, based on the vertical deflections of the left and right tyres, $w_{r,l} - z_{u,l}$ and $w_{r,r} - z_{u,r}$, see also the formulations in Section 3.3:

$$LTR = \frac{k_{t,l} [w_{r,l} - z_{u,l}] - k_{t,r} [w_{r,r} - z_{u,r}]}{k_{t,l} [w_{r,l} - z_{u,l}] + k_{t,r} [w_{r,r} - z_{u,r}]} \quad (68)$$

LTR ranges from 0 to 1, where 0 corresponds to equal vertical forces on both vehicle sides, while 1 refers to the condition in which either the right or left tyre loses contact with the ground.

In the literature further vehicle stability limitations are imposed in terms of yaw rate and/or sideslip angle as hard (J. Hu et al., 2020; Liu & Kang, 2019; Zou et al., 2019) or soft constraints (Liang et al., 2021b; Sun et al., 2018; H. Wang et al., 2021b). The typical yaw rate constraint is a direct consequence of (63), under the assumption of steady-state cornering conditions:

$$|\dot{\psi}| \leq \frac{\mu g}{v_x} \quad (69)$$

The sideslip angle constraint is often expressed in terms of rear slip angle at which the rear lateral tyre force saturates, $\alpha_{r,sat}$, combined with the kinematic relationship between rear slip angle and sideslip at the centre of gravity:

$$|\beta| \leq \alpha_{r,sat} + \frac{l_r \dot{\psi}}{v_x} \quad (70)$$

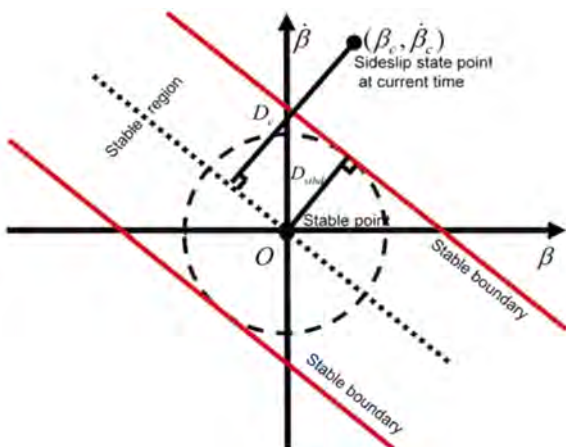


Fig. 9. Sideslip stability envelope (from B. Zhang et al., 2019a).

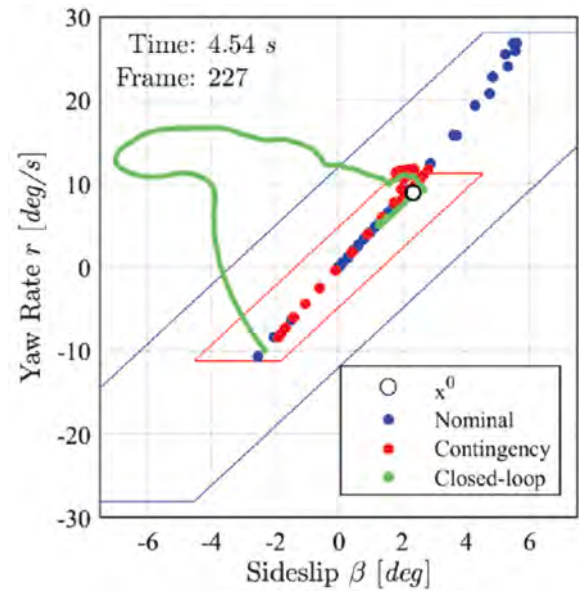


Fig. 10. Stability envelopes for snow (blue) and ice (red) conditions (from Alsterda et al., 2019).

while in several cases the sideslip limitations are directly in terms of hard (Z. He et al., 2020; J. Hu et al., 2020; F. Lin et al., 2020; Luan et al., 2020) or soft (Erlie et al., 2016; Sun et al., 2018; Z. Wang, Bai, et al., 2019) slip angle constraints at the tyre level:

$$\alpha_{min} \leq \alpha \leq \alpha_{max} \quad (71)$$

B. Zhang et al. (2019a) proposes a phase-plane-based constraint in terms of β and $\dot{\beta}$, to identify stable and unstable operating regions according to the linear boundaries in Fig. 9:

$$|\dot{\beta} + E_1 \beta| \leq E_2 \quad (72)$$

where E_1 and E_2 are functions of the tyre-road friction coefficient, μ , and longitudinal speed, v_x . Alternatively, Funke et al. (2016) and Hajiloo et al. (2021) define a soft stability constraint in terms of yaw rate as a function of lateral slip speed, while Alsterda et al. (2019) uses a soft stability constraint in terms of r as a function of β , see Fig. 10.

Cao et al. (2021) and Liang et al. (2021b) also set constraints on the longitudinal tyre slip ratios ($|\sigma_{ij}| \leq \sigma_{max}$), in the context of PT implementations embedding also the anti-lock braking and traction control functions through individual wheel torque control.

The important observation of Funke et al. (2016) and Hajiloo et al. (2021) is that softening the stability constraints not only avoids unfeasible solutions, but can be necessary when the vehicle must avoid an obstacle in emergency conditions, in which, for example, larger sideslip angle values, beyond those of conventional stability controllers for human-driven vehicles, can facilitate the generation of high-curvature trajectories.

- Environmental envelope constraints. To prevent collision with road boundaries or other vehicles, constraints can be expressed in terms of vehicle position in the global reference system (Hamid et al., 2017a; H. Guo, Shen et al., 2018; Kabzan et al., 2019; B. Zhang et al., 2019b):

$$\begin{aligned} X_{min} &\leq X \leq X_{max} \\ Y_{min} &\leq Y \leq Y_{max} \end{aligned} \quad (73)$$

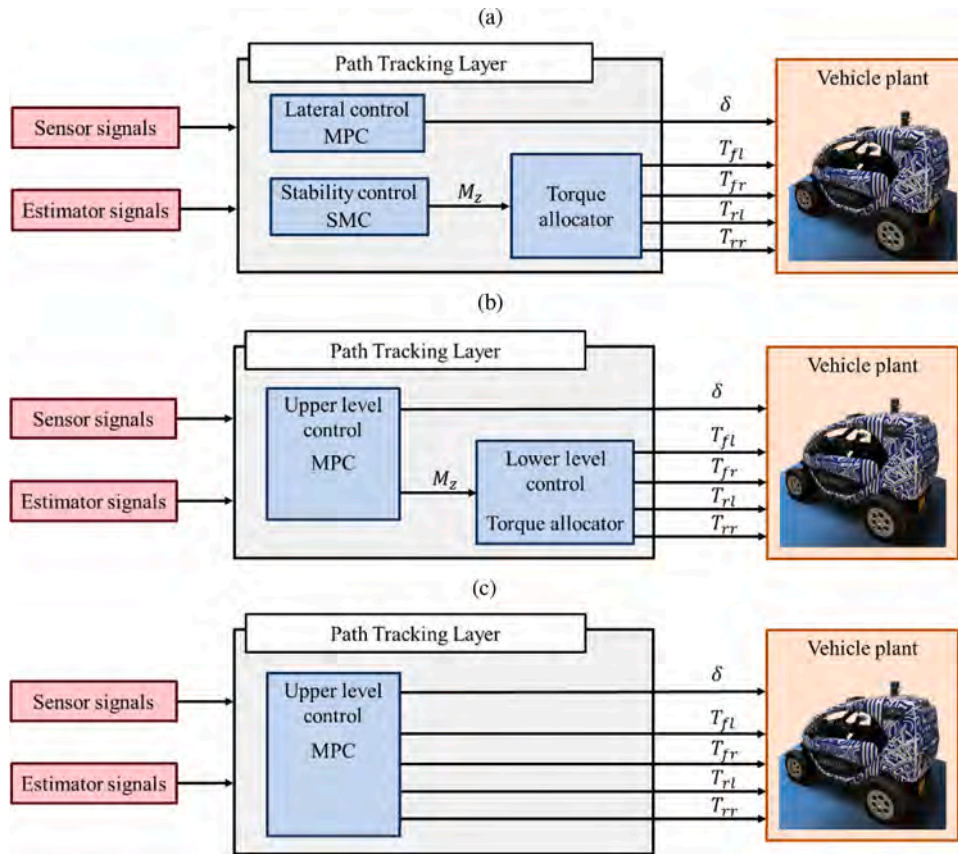


Fig. 11. Overview of typical MPC-based PT architectures for vehicles with multiple chassis actuators: (a) MPC for front steering angle control, and a separate controller (e.g., an SMC) for DYM control; (b) integration of δ and M_z control in a single MPC layer; and (c) integration of δ and T_{ij} control in a single MPC implementation.

or in terms of lateral position error in a curvilinear reference system (Y. Chen et al., 2020; Erlien et al., 2016; Liang et al., 2021b):

$$\begin{aligned} e_{y,\min} &\leq e_y \leq e_{y,\max} \\ e_{y,p,\min} &\leq e_{y,p} \leq e_{y,p,\max} \end{aligned} \quad (74)$$

Funke et al. (2016) defines soft environmental boundaries in terms of lateral position error, including the effect of the body frame orientation:

$$\begin{aligned} e_y + f_{width}(e_\psi) &\leq e_{y,\max} \\ e_y - f_{width}(e_\psi) &\geq e_{y,\min} \end{aligned} \quad (75)$$

through the function $f_{width}(e_\psi)$, which approximates the width and orientation of the vehicle. Additional constraints on the safe following distance d_{safe} can be set to prevent collision with the preceding vehicle, as defined in Suh et al. (2018), Vallon et al. (2017), and H. Yuan et al. (2018):

$$s^p - s \geq d_{safe} \quad (76)$$

where s^p is the predicted position of the preceding vehicle, and d_{safe} is the minimum safety distance. The choice of soft environmental constraints guarantees safety in emergency scenarios, allowing the vehicle to temporally violate the lane boundaries to avoid the obstacle (Funke et al., 2016; Hajiloo et al., 2021).

4.4. MPC-based path tracking and integrated chassis control

MPC-based PT through front axle steering is often integrated with other actuation methods, e.g., DYM and/or rear-wheel-steering. Fig. 11

shows three typical implementation options:

- The MPC-based PT algorithm generates the reference value of δ , while in parallel a different controller generates the other high-level control inputs for vehicle dynamics control (Fig. 11(a)), i.e., typically a reference direct yaw moment M_z to be actuated through multiple electric motors and/or the friction brakes (Ahn et al., 2021; F. Lin, Zhang et al., 2019; Ren et al., 2018; Xie et al., 2021). A commonly investigated solution for M_z generation is the adoption of a sliding mode controller (SMC), where the sliding variable can be defined in terms of yaw rate error (F. Lin, Zhang et al., 2019), lateral position error (Ahn et al., 2021; Xie et al., 2021), and/or sideslip angle (Ren et al., 2018), either individually or concurrently. Based on the M_z value from the high-level stability controller, a low-level control allocator computes the individual reference wheel torque levels, T_{ij} . A further low-level control layer can correct the individual wheel torque from the control allocation algorithm to prevent excessive wheel slip in traction or braking. The benefit of this architecture is the substantial independence of the PT and vehicle stability control functions, which allows the implementation of driving automation without any substantial modification of the vehicle dynamics control architecture w.r.t. the one of human-driven vehicles.
- The MPC-based PT controller outputs both δ and M_z , see Fig. 11(b), while a control allocation algorithm computes the individual wheel torque levels, also in this case generated through differential braking (Hajiloo et al., 2021; H. Wang et al., 2021a; H. Wang & Liu, 2021), or in-wheel motor torque distribution (J. Guo et al., 2017; Peng et al., 2019; Xiang et al., 2020; Zou et al., 2019). Examples of low-level algorithms use pseudo-inverse (J. Guo et al., 2017) or optimal allocation (Peng et al., 2019; Zou et al., 2019) formulations, which can

Table 3
Overview of the MPC-based PT control tasks and respective inequality constraints.

Control task	Constraint	Formulation	References		
Lateral control	Physical limits of the actuators	$\delta_{min} \leq \delta \leq \delta_{max}$	Ahn et al., 2021; Alsterda et al., 2019; Alsterda & Gerdes, 2021; Bo et al., 2019; J. Cao et al., 2020; S. Chen & Chen, 2020; W. Chen et al., 2021; X. Chen et al., 2021; Y. Chen et al., 2020; Choi et al., 2018; Cheng et al., 2020; Codrean et al., 2020; Cui et al., 2017; Dai et al., 2020; Dallas et al., 2020; S. Feng et al., 2021; Geng et al., 2020; H. Guo, Cao et al., 2018; H. Guo, Shen et al., 2018; J. Guo et al., 2017; Hatem, 2018*; Hang et al., 2021; H. He et al., 2021; Z. He et al., 2020; J. Hu et al., 2020; Ji et al., 2017; M. Kim et al., 2021; J. Lee & Chang, 2018; Leman et al., 2019; S. Li et al., 2020; S. Li, G. Wang et al., 2019; S. Li, S. Wang et al., 2019; Liang et al., 2021a; Lima et al., 2017; F. Lin et al., 2020; F. Lin, Chen et al. 2019; F. Lin, Zhang et al. 2019; J. Liu et al., 2017; Z. Liu & Kang, 2019; Lu et al., 2018; Luan et al., 2020; Mata et al., 2019; Massera et al., 2020; Nam et al., 2019; Peng et al., 2019; Quan & Chung, 2019; Quirynen et al., 2018; Reda et al. 2020; Rokonzaman et al., 2019, 2020, 2021b; Seccamonte et al., 2019; Spielberg et al., 2021; Taghavifar, 2019; Z. Tang et al., 2018; H. Wang et al., 2019, 2021b; H. Wang & Liu, 2021; Z. Wang, Bai, et al., 2019; Wurts et al., 2021; Xie et al., 2021; S. Xu & Peng, 2020; Y. Xu et al., 2021; Yakub et al., 2016; Yakub & Mori, 2015; Yu et al., 2015; J. Yu et al., 2019, 2020; K. Yuan et al., 2018; C. Yu et al., 2021; B. Zhang et al., 2019a, 2019b; V. Zhang et al., 2018; K. Zou et al., 2021; Y. Zou et al., 2019		
		$\Delta\delta_{min} \leq \Delta\delta \leq \Delta\delta_{max}$	Ahn et al., 2021; Alsterda et al., 2019; Alsterda & Gerdes, 2021; S. Chen & Chen, 2020; W. Chen et al., 2021; X. Chen et al., 2021; Y. Chen et al., 2020; Choi et al., 2018; Cui et al., 2017; Dallas et al., 2020; S. Feng et al., 2021; Geng et al., 2020; H. Guo, Cao et al., 2018; J. Guo et al., 2017; Hatem, 2018; Ji et al., 2017; D. Kim et al., 2018; M. Kim et al., 2021; Hang et al., 2021; H. He et al., 2021; Z. He et al., 2020; J. Hu et al., 2020; S. Li et al., 2020; S. Li, G. Wang et al., 2019; S. Li, S. Wang et al., 2019; Lima et al., 2017; F. Lin et al., 2020; F. Lin, Chen et al. 2019; F. Lin, Zhang et al. 2019; J. Liu et al., 2017; Z. Liu & Kang, 2019; Lu et al., 2018; Mata et al., 2019; Quan & Chung, 2019; Reda et al. 2020; Rokonzaman et al., 2019, 2020, 2021b; Spielberg et al., 2021; Taghavifar, 2019*; Z. Tang et al., 2018; H. Wang et al., 2019, 2021b; H. Wang & Liu, 2021; Z. Wang, Bai, et al., 2019; Xie et al., 2021; Yu et al., 2015; J. Yu et al., 2019, 2020; K. Yuan et al., 2018; C. Yu et al., 2021; B. Zhang et al., 2019a, 2019b; V. Zhang et al., 2018; Y. Zou et al., 2019*		
		$\delta_{r,min} \leq \delta_r \leq \delta_{r,max}$	Hang et al., 2021; Wurts et al., 2021; Yakub et al., 2016; Yakub & Mori, 2015; C. Yu et al., 2021		
		$\Delta\delta_{r,min} \leq \Delta\delta_r \leq \Delta\delta_{r,max}$	Hang et al., 2021; Wurts et al., 2021; Yakub et al., 2016; Yakub & Mori, 2015; C. Yu et al., 2021		
		$F_{y,f,min} \leq F_{y,f} \leq F_{y,f,max}$	Brown et al., 2016; Erlie et al., 2016; Funke et al., 2015, 2016; Hajiloo et al., 2021; C. Sun et al., 2018, 2019; Yao & Tian, 2019		
		$\Delta F_{y,f,min} \leq \Delta F_{y,f} \leq \Delta F_{y,f,max}$	Brown et al., 2016; Erlie et al., 2016; Funke et al., 2015, 2016; Hajiloo et al., 2021; C. Sun et al., 2018; Yao & Tian, 2019		
		$M_{z,min} \leq M_z \leq M_{z,max}$	J. Guo et al., 2017; Hajiloo et al., 2021; Hang et al., 2021; Peng et al., 2019; H. Wang & Liu, 2021; Yakub et al., 2016; Yakub & Mori, 2015; Zou et al., 2019		
		$\Delta M_{z,min} \leq \Delta M_z \leq \Delta M_{z,max}$	J. Guo et al., 2017; Hajiloo et al., 2021; H. Wang & Liu, 2021; Yakub et al., 2016; Yakub & Mori, 2015		
		$T_{\delta,min} \leq T_{\delta} \leq T_{\delta,max}$	Ercan et al., 2017		
		$\Delta T_{\delta,min} \leq \Delta T_{\delta} \leq \Delta T_{\delta,max}$	Ercan et al., 2017		
		Vehicle stability		$\alpha_{min} \leq \alpha \leq \alpha_{max}$	J. Cao et al., 2020*; Dai et al., 2020; Dallas et al., 2020; Deng et al., 2020; Ercan et al., 2017; Geng et al., 2020; Z. He et al., 2020; J. Hu et al., 2020; F. Lin et al., 2020; Massera et al., 2020; Nam et al., 2019; Taghavifar, 2019*; Z. Wang, Bai, et al., 2019*; Wurts et al., 2021; K. Yuan et al., 2018; Zou et al., 2019
				$\beta_{min} \leq \beta \leq \beta_{max}$	J. Cao et al., 2020; Cui et al., 2017; S. Feng et al., 2021; Ji et al., 2017; H. He et al., 2021; J. Hu et al., 2020; Law et al., 2018; Z. Liu & Kang, 2019; Peng et al., 2019; Taghavifar, 2019; H. Wang et al., 2021b*; Zou et al., 2019
				$ \beta \leq \alpha_{r,sat} + \frac{l_r \dot{\psi}}{v_x}$	Alsterda et al., 2019*; Alsterda & Gerdes, 2021*; Brown et al., 2016*; Erlie et al., 2016*; Funke et al., 2015*; F. Lin et al., 2020; Sun et al., 2018*, 2019*; Yao & Tian, 2019
$ \dot{\beta} + E_1 \beta \leq E_2$	B. Zhang et al., 2019a				
$ C_1 \varphi + C_2 \dot{\varphi} \leq C_3$	Hang et al., 2021				
$ \varphi \leq \varphi_{max} = \frac{m_s h a_y}{2[K_{\varphi} - m_s g h]}$	Taghavifar, 2019				
$LTR_{min} \leq LTR \leq LTR_{max}$					
$-\frac{\mu g}{v_x} \leq \dot{\psi} \leq \frac{\mu g}{v_x}$	Alsterda et al., 2019*; Alsterda & Gerdes, 2021*; Brown et al., 2016*; Codrean et al., 2020; Deng et al., 2020; Erlie et al., 2016*; Funke et al., 2015*; Hajiloo et al., 2021*; H. He et al., 2021; Ji et al., 2017; F. Lin et al., 2020; F. Lin, Chen et al. 2019; Massera et al., 2020; Sun et al., 2018*, 2019*; H. Wang et al., 2021b*; Xie et al., 2021; Yao & Tian, 2019; Zou et al., 2019				

(continued on next page)

Table 3 (continued)

Control task	Constraint	Formulation	References
		$ \dot{\psi} \leq \min\left(\frac{F_{y,f,max}\left[1 + \frac{l_f}{l_r}\right]}{mv_x}, \frac{F_{y,r,max}\left[1 + \frac{l_r}{l_f}\right]}{mv_x}\right)$ $v_{y,min} \leq v_y \leq v_{y,max}$ $-v_x\alpha_{r,sat} - l_r\dot{\psi} \leq v_y \leq v_x\alpha_{r,sat} + l_r\dot{\psi}$ $a_{y,min} \leq a_y \leq a_{y,max}$	<p>Funke et al., 2016*</p> <p>Deng et al., 2020; Liang et al., 2021a*</p> <p>Funke et al., 2016*; Hajiloo et al., 2021*</p> <p>J. Cao et al., 2020*; Dai et al., 2020; S. Feng et al., 2021; J. Hu et al., 2020*; S. Li, S. Wang et al., 2019; F. Lin et al., 2020; F. Lin, Chen et al. 2019; Z. Liu & Kang, 2019*; Taghavifar, 2019; K. Yuan et al., 2018</p>
	Environmental envelope	$e_{y,min} \leq e_y \leq e_{y,max}$ $e_y + f_{width}(e_\psi) \leq e_{y,max}$ $e_y - f_{width}(e_\psi) \geq e_{y,min}$ $e_{\psi,min} \leq e_\psi \leq e_{\psi,max}$ $X_{min} \leq X \leq X_{max}$ $Y_{min} \leq Y \leq Y_{max}$ $\Psi_{min} \leq \Psi \leq \Psi_{max}$	<p>Alsterda et al., 2019*; Alsterda & Gerdes, 2021*; Ercan et al., 2017; M. Kim et al., 2021*; J. Lee & Chang, 2018; Liang et al., 2021a*; Peng et al., 2019; Quirynen et al., 2018; Seccamonte et al., 2019; Sun et al., 2019*; H. Wang & Liu, 2021; Y. Xu et al., 2021*; J. Yu et al., 2019</p> <p>Brown et al., 2016*; Ertien et al., 2016*; Funke et al., 2015*, 2016*; Hajiloo et al., 2021*</p> <p>M. Kim et al., 2021*; J. Lee & Chang, 2018; Liang et al., 2021a*; Peng et al., 2019; Sun et al., 2019*; H. Wang & Liu, 2021; Y. Xu et al., 2021*; J. Yu et al., 2019</p> <p>S. Chen & Chen, 2020; Dallas et al., 2020; Lima et al., 2017; J. Liu et al., 2017; Wurts et al., 2021</p> <p>Bo et al., 2019; S. Chen & Chen, 2020; X. Chen et al., 2021*; Y. Chen et al., 2020; Cui et al., 2017; Dai et al., 2020*; Dallas et al., 2020; Deng et al., 2020; H. Guo, Cao et al., 2018; H. Guo, Shen et al., 2018; Ji et al., 2017; Lima et al., 2017; F. Lin, Chen et al. 2019; J. Liu et al., 2017; Z. Liu & Kang, 2019; Taghavifar, 2019; Z. Tang et al., 2018*; H. Wang et al., 2019; Wurts et al., 2021; K. Yuan et al., 2018; B. Zhang et al., 2019a, 2019b</p> <p>Bo et al., 2019; S. Chen & Chen, 2020; X. Chen et al., 2021*; Y. Chen et al., 2020; Cui et al., 2017; F. Lin, Chen et al. 2019; H. Wang et al., 2019; Wurts et al., 2021</p>
Combined control	Physical limits of the actuators	$\delta_{min} \leq \delta \leq \delta_{max}$ $\Delta\delta_{min} \leq \Delta\delta \leq \Delta\delta_{max}$ $\delta_{r,min} \leq \delta_r \leq \delta_{r,max}$ $\Delta\delta_{r,min} \leq \Delta\delta_r \leq \Delta\delta_{r,max}$ $v_{min} \leq v \leq v_{max}$ $\Delta v_{min} \leq \Delta v \leq \Delta v_{max}$ $a_{x,min} \leq a_x \leq a_{x,max}$ $\Delta a_{x,min} \leq \Delta a_x \leq \Delta a_{x,max}$ $F_{x,ij,min} \leq F_{x,ij} \leq F_{x,ij,max}$ $\Delta F_{x,ij,min} \leq \Delta F_{x,ij} \leq \Delta F_{x,ij,max}$ $F_{x,tot,min} \leq F_{x,tot} \leq F_{x,tot,max}$ $\Delta F_{x,tot,min} \leq \Delta F_{x,tot} \leq \Delta F_{x,tot,max}$ $F_{y,tot,min} \leq F_{y,tot} \leq F_{y,tot,max}$ $F_{b,tot,min} \leq F_{b,tot} \leq F_{b,tot,max}$ $\Delta F_{b,tot,min} \leq \Delta F_{b,tot} \leq \Delta F_{b,tot,max}$	<p>Alcalá et al., 2019a, 2019b; Batkovic et al., 2019; Berntorp et al., 2019; Brunner et al., 2017; M. Cao et al., 2021; Cesari et al., 2017; M. Chen & Ren, 2017; Y. Chen et al., 2019; Y. Chen & Wang, 2019; Chowdhri et al., 2021; Dawood et al., 2020; Du et al., 2016; Farag, 2020; Hamid et al., 2017a, 2017b; Hashemi et al., 2021; C. Hu & Zhao, 2020; Huang et al., 2016; Y. Huang et al., 2019; Kabzan et al., 2019; Karimshoushtari et al., 2021; Kong et al., 2015; Laurence & Gerdes, 2021; T. Lee & Kang, 2021; Liang et al., 2021b; Luciani et al., 2020; Qian et al., 2016; Qin et al., 2021; Ren et al., 2018; Rosolia et al., 2017; Rosolia & Borrelli, 2019; Suh et al., 2018; Taherian et al., 2019; Tan et al., 2018; Vallon et al., 2017; H. Wang, Huang et al., 2019; H. Wu et al., 2020; Xiang et al., 2020; K. Yang et al., 2021; Ye et al., 2019*; Y. Yu et al., 2021; H. Yuan et al., 2018; W. Zhang et al., 2020</p> <p>Alcalá et al., 2019a, 2019b; Batkovic et al., 2019; Berntorp et al., 2019; M. Cao et al., 2021; Cesari et al., 2017; M. Chen & Ren, 2017; Chowdhri et al., 2021; Du et al., 2016; Hamid et al., 2017a, 2017b; Huang et al., 2016; Y. Huang et al., 2019; Kabzan et al., 2019; Kong et al., 2015; Laurence & Gerdes, 2021; Liang et al., 2021b; Qian et al., 2016; Ren et al., 2018; Tan et al., 2018; Vallon et al., 2017; H. Wang, Huang et al., 2019; H. Wu et al., 2020; Xiang et al., 2020; K. Yang et al., 2021; Ye et al., 2019*; Y. Yu et al., 2021; W. Zhang et al., 2020</p> <p>Huang et al., 2016; Tan et al., 2018</p> <p>Huang et al., 2016; Tan et al., 2018</p> <p>Du et al., 2016; Qian et al., 2016; Ye et al., 2019*</p> <p>Du et al., 2016; Ye et al., 2019*</p> <p>Alcalá et al., 2019a, 2019b; Brunner et al., 2017; Cesari et al., 2017; Dawood et al., 2020; Karimshoushtari et al., 2021; Kong et al., 2015; T. Lee & Kang, 2021; Luciani et al., 2020; Qian et al., 2016; Rosolia et al., 2017; Rosolia & Borrelli, 2019; Suh et al., 2018; Taherian et al., 2019; Vallon et al., 2017</p> <p>Alcalá et al., 2019a, 2019b; Cesari et al., 2017; Kong et al., 2015; Vallon et al., 2017</p> <p>C. Hu & Zhao, 2020; H. Wu et al., 2020; H. Yuan et al., 2018;</p> <p>H. Wu et al., 2020</p> <p>M. Chen & Ren, 2017; Y. Huang et al., 2019; Laurence & Gerdes, 2021; Ren et al., 2018; Tan et al., 2018; H. Wang, Huang et al., 2019; Z. Wang, Zha et al., 2019, 2020; Xiang et al., 2020; K. Yang et al., 2021; Y. Yu et al., 2021</p> <p>M. Chen & Ren, 2017; Y. Huang et al., 2019; Ren et al., 2018; Tan et al., 2018; Xiang et al., 2020; K. Yang et al., 2021; Y. Yu et al., 2021</p> <p>Z. Wang, Zha et al., 2019, 2020</p> <p>Hamid et al., 2017a, 2017b</p> <p>Hamid et al., 2017a, 2017b</p>

(continued on next page)

Table 3 (continued)

Control task	Constraint	Formulation	References
		$M_{z,min} \leq M_z \leq M_{z,max}$ $\Delta M_{z,min} \leq \Delta M_z \leq \Delta M_{z,max}$ $T_{ij,min} \leq T_{ij} \leq T_{ij,max}$ $\Delta T_{ij,min} \leq \Delta T_{ij} \leq \Delta T_{ij,max}$ $T_{f,min} \leq T_f \leq T_{f,max}$ $\Delta T_{f,min} \leq \Delta T_f \leq \Delta T_{f,max}$ $T_{r,min} \leq T_r \leq T_{r,max}$ $\Delta T_{r,min} \leq \Delta T_r \leq \Delta T_{r,max}$ $T_{tot,min} \leq T_{tot} \leq T_{tot,max}$ $\Delta T_{tot,min} \leq \Delta T_{tot} \leq \Delta T_{tot,max}$ $T_{b,ij,min} \leq T_{b,ij} \leq T_{b,ij,max}$ $\Delta T_{b,ij,min} \leq \Delta T_{b,ij} \leq \Delta T_{b,ij,max}$ $\frac{T_{b,rl} + T_{b,rr}}{T_{b,fl} + T_{b,fr} + e} \leq \frac{\frac{l_f}{l_f + l_r} + h_g(\dot{v}_x - v_y\dot{\psi})}{1 - \frac{l_f}{l_f + l_r} - h_g(\dot{v}_x - v_y\dot{\psi})}$	Z. Wang, Zha et al., 2019, 2020; Xiang et al., 2020 Xiang et al., 2020 M. Cao et al., 2021; Hashemi et al., 2021; Huang et al., 2016; W. Zhang et al., 2020 M. Cao et al., 2021; Huang et al., 2016; W. Zhang et al., 2020 Berntorp et al., 2019 Berntorp et al., 2019 Berntorp et al., 2019; Y. Chen et al., 2019; Y. Chen & Wang, 2019 Berntorp et al., 2019 Farag, 2020; Kabzan et al., 2019 Kabzan et al., 2019 Chowdhri et al., 2021; Qin et al., 2021 Chowdhri et al., 2021 Chowdhri et al., 2021
Vehicle stability		$\alpha_{min} \leq \alpha \leq \alpha_{max}$ $\beta_{min} \leq \beta \leq \beta_{max}$ $\dot{\beta}_{min} \leq \dot{\beta} \leq \dot{\beta}_{max}$ $-\frac{\mu g}{v_x} \leq \dot{\psi} \leq \frac{\mu g}{v_x}$ $v_{x,min} \leq v_x \leq v_{x,max}$ $v_x \leq sf \min \left(\sqrt{\frac{\mu g}{\rho}}, \sqrt{\frac{[mg - F_r] b}{2mh_g \rho}}, v_{max} \right)$ $v_{y,min} \leq v_y \leq v_{y,max}$ $[\dot{v}_x - v_y\dot{\psi}]^2 + [\dot{v}_y + v_x\dot{\psi}]^2 \leq [\mu g]^2$ $a_{y,min} \leq a_y \leq a_{y,max}$ $\sigma_{x,min} \leq \sigma_x \leq \sigma_{x,max}$ $[p_{long} F_x]^2 + F_{ly}^2 \leq F_{ellipse,max}^2$	Hamid et al., 2017a, 2017b; Hashemi et al., 2021*; Qin et al., 2021*; H. Wu et al., 2020 Chowdhri et al., 2021; C. Hu & Zhao, 2020; Xiang et al., 2020; W. Zhang et al., 2020* Chowdhri et al., 2021 Y. Chen et al., 2019; Y. Chen & Wang, 2019; Dawood et al., 2020; Hashemi et al., 2021*; C. Hu & Zhao, 2020; Qin et al., 2021*; Z. Wang, Zha et al., 2019*, 2020*; Xiang et al., 2020; W. Zhang et al., 2020* Y. Chen et al., 2019; Y. Chen & Wang, 2019; Chowdhri et al., 2021; Dawood et al., 2020; Luciani et al., 2020; H. Wang, Huang et al., 2019*; H. Wu et al., 2020*; Xiang et al., 2020; K. Yang et al., 2021; Y. Yu et al., 2021 M. Chen & Ren, 2017 Y. Chen et al., 2019; Y. Chen & Wang, 2019; Liang et al., 2021b*; Z. Wang, Zha et al., 2019*, 2020* Chowdhri et al., 2021 Berntorp et al., 2019*; Qian et al., 2016 M. Cao et al., 2021 Chowdhri et al., 2021; Kabzan et al., 2019; Laurence & Gerdes, 2021
Environmental envelope		$e_{y,min} \leq e_y \leq e_{y,max}$ $e_{y,p,min} \leq e_{y,p} \leq e_{y,p,max}$ $e_{\psi,min} \leq e_{\psi} \leq e_{\psi,max}$ $X_{min} \leq X \leq X_{max}$ $Y_{min} \leq Y \leq Y_{max}$ $\psi_{min} \leq \psi \leq \psi_{max}$ $s^p - s \geq d_{safe}$	Alcalá et al., 2019a, 2019b; Berntorp et al., 2019*; Brunner et al., 2017; Cesari et al., 2017; Farag, 2020; Karimshoushtari et al., 2021; Luciani et al., 2020; Qian et al., 2016; Rosolia et al., 2017; Rosolia & Borrelli, 2019; Suh et al., 2018; K. Yang et al., 2021; Vallon et al., 2017 Liang et al., 2021b* Brunner et al., 2017; Farag, 2020; Karimshoushtari et al., 2021; Liang et al., 2021b*; Luciani et al., 2020; Qian et al., 2016; Rosolia et al., 2017; Rosolia & Borrelli, 2019; K. Yang et al., 2021 Kabzan et al., 2019*; H. Yuan et al., 2018 M. Chen & Ren, 2017; Hamid et al., 2017a, 2017b; Kabzan et al., 2019*; Ren et al., 2018; H. Wang, Huang et al., 2019*; H. H. Wu et al., 2020*; Xiang et al., 2020; Y. Yu et al., 2021; Yuan et al., 2018 M. Chen & Ren, 2017; Ren et al., 2018; H. Wu et al., 2020*; Xiang et al., 2020; Y. Yu et al., 2021 Cesari et al., 2017; Suh et al., 2018; Vallon et al., 2017

Table 4
Overview of toolboxes and solvers in the considered MPC studies.

Toolbox	Solver	References
ACADO	qpOASES	Chowdhri et al., 2021; Quirynen et al., 2018; Seccamonte et al., 2019
	High performance MPC (HPMPC)	Batkovic et al., 2019; Quirynen et al., 2018
	PQP	Quirynen et al., 2018
	ADMM	Quirynen et al., 2018
	PRESAS	Berntorp et al., 2019; Quirynen et al., 2018
	Gradient-based augmented Lagrangian approach (GRAMPC)	Z. Wang et al., 2020
	Interior Point Optimizer (IPOPT)	Qian et al., 2016; Ren et al., 2018; C. Yu et al., 2021; Y. Yu et al., 2021 Brunner et al., 2017; Rokonzaman et al., 2021b; L. Tang et al., 2020; Wurts et al., 2021 Xiang et al., 2020; K. Zou et al., 2021
CasADi		Dawood et al., 2020; Rokonzaman et al., 2019, 2020
	IPOPT - solver MA27	Laurence & Gerdes, 2021; Spielberg et al., 2021; W. Zhang et al., 2020
CVXGEN	Majorization-Minimization (MM)	Lima et al., 2015 Brown et al., 2016; Erlen et al., 2016; Funke et al., 2015, 2016; M. Kim et al., 2021; Lin et al., 2020; Pereira et al., 2017; Z. Wang et al., 2020; Z. Wang, Zha, et al., 2019; J. Yu et al., 2019, 2020; V. Zhang et al., 2018
		Alsterda et al., 2019; Kabzan et al., 2019; Suh et al., 2018
FORCES Pro	NA	
YALMIP	GUROBI	Alcalá et al., 2019a
	IPOPT	Alcalá et al., 2019a
	POP	J. Lee & Chang, 2018 Hatem, 2018
Matlab optimization toolbox	Fmincon – Sequence Quadratic Programming (SQP)	S. Li, G. Wang et al., 2019
MPC toolbox in Matlab/Simulink		Yakub & Mori, 2015
NA	Backpropagation Through Time	Quan & Chung, 2019 M. Chen & Ren, 2017
NA	Active-set	Z. Wang, Bai, et al., 2019
NA	qpOASES	Hashemi et al., 2021; Qin et al., 2021
NA	Backpropagation Through Time	Quan & Chung, 2019
NA	Operator Splitting Quadratic Programming (OSQP)	Alcalá et al., 2019b; Rosolia & Borrelli, 2019
NA	NPSOL	Cesari et al., 2017; Kong et al., 2015; Rosolia et al., 2017
NA	Nelder-Mead method	J. Kim et al., 2018
NA	Interior Point Algorithm	S. Xu & Peng, 2020
NA	Hildreth's algorithm	Ji et al., 2017
NA	KWIK	Luciani et al., 2020

be supported, in a further layer, by the wheel slip control function. Although this solution implies a revision of the vehicle stability control architecture w.r.t. the established one, based on yaw rate and sideslip angle, of human-driven vehicles, the integration of steering and DYM control ensures coherent action of the involved actuators, with the possibility of effectively using M_z for PT.

- The MPC-based PT algorithm directly generates both δ and T_{ij} , see Fig. 11(c), according to a centralised approach (Cao et al., 2021; Chowdhri et al., 2021; Qin et al., 2021; W. Zhang et al., 2020). In this case, the adoption of a DT prediction model is a necessary requirement; moreover, all mentioned implementations under this sub-category also include the wheel moment balance equations in their internal models, which enables integration of individual tyre slip control in traction and braking. These enhanced model formulations significantly increase the computational load, with a trade-off between performance and real-time implementability to be carefully evaluated in further analyses.

4.5. Summary

As a summary of Section 4, Table 3 offers an overview of the MPC control tasks, i.e., lateral vehicle dynamics control or combined longitudinal and lateral control (which is strictly related to the adopted cost function), and the typical inequality constraints for PT applications, classified into: i) physical limits of the actuators; ii) vehicle stability; and iii) environmental envelope. The superscript “*” indicates the adoption of a soft constraint. The multiple actuator configurations in the considered literature are reported in Table 1 (see the list of control inputs).

5. Controller implementation and results

5.1. Solvers and hardware

In general, MPC can be implemented: i) explicitly (Grancharova & Johansen, 2012), if the solution of the optimal control problem is obtained offline, while the online implementation reduces to a computationally light function evaluation; or ii) implicitly, if the solution of the optimal control problem is obtained online, at each time step. Given that i) implies memory requirements that are beyond the current practical feasibility for problems with a relatively high number of states and parameters, such as those associated with PT control including the preview of the road curvature ahead, explicit MPC has been adopted in a very limited number of studies, e.g., see Choi et al. (2018) and Lee & Chang (2018). The vast majority of the MPC implementations for PT are based on implicit algorithms, which require significant computing infrastructure for the online optimization. This might not be available for processes with fast sampling time and limited computational resources, such as the PT algorithms for AVs (Lee & Chang, 2018; Siampis et al.,

Table 5
Average and maximum NMPC runtimes with ST and DT prediction models, on a Raspberry Pi 2 device (Quirynen et al., 2018).

Solver	$N_p = 20$ mean/max	$N_p = 30$ mean/max	$N_p = 40$ mean/max	$N_p = 60$ mean/max
ST model				
PQP	23.0/25.7	105/117	209/226	578/599
ADMM	14.7/27.5	39.6/71.9	85.5/150	268/440
qpOASES	12.2/21.0	26.1/50.1	48.5/81.7	132/240
HPMPC	11.9/16.9	18.1/21.9	24.5/29.2	37.0/45.1
PRESAS	6.44/9.61	9.14/10.7	12.4/19.8	18.4/19.8
DT model				
PQP	45.1/55.5	142/159	266/288	672/704
ADMM	33.4/50.2	69.7/101	129/190	344/506
qpOASES	34.5/40.4	63.7/82.1	102/131	229/354
HPMPC	39.44/48.0	58.7/77.5	82.6/126	122/136
PRESAS	26.0/30.2	39.0/41.9	52.5/65.3	78.7/82.5

2018).

Several toolboxes and solvers have been adopted in the literature to deal with the OCPs of MPC-based PT, see the overview in **Table 4**. The most frequently used tools are:

- ACADO, i.e., an open-source tool based on sequential quadratic programming type iteration of the generalised Gauss-Newton method. A Matlab interface is available, which allows to directly export, compile, and use autogenerated C code for NMPC, and enables real-time implementations (Quirynen et al., 2014).
- CasADi, i.e., an open-source framework to formulate and solve optimisation problems, with special interest for those constrained by differential equations. CasADi provides flexibility for solving OCPs that can be efficiently transposed into a nonlinear program (direct approach), and is written in self-contained C++ code, but is normally used via full-featured interfaces to Python, Matlab, or Octave (Andersson et al., 2019). CasADi is suitable for NNMPC implementations (Spielberg et al., 2021).
- CVXGEN, i.e., a software tool for convex optimisation problems, which are compiled into high-speed automatically generating solvers. The solvers have no library dependencies, and are almost branch free, making them suitable for real-time applications. CVXGEN targets problem families that can be transformed into convex quadratic programs of modest size (Mattingley & Boyd, 2012).
- YALMIP, i.e., a free general-purpose toolbox for modelling and solving optimisation problems in Matlab, including linear, quadratic, semidefinite, mixed integer, and multiparametric programming. YALMIP can be interfaced with approximately 20 solvers, such as GUROBI, POP, SeDuMi, SDPT3, PENNON, MILAB 141, and CPLEX (Lofberg, 2004).
- FORCES Pro, i.e., a commercial tool to generate tailor-made solvers from high-level mathematical descriptions of optimisation problems. The software is specifically designed for fast and embedded optimisation. The generated C code is library-free and does not use dynamic memory allocation, which makes it suitable for safe deployment on real-time automated systems (Domahidi & Jerez, 2014).

New more performing toolboxes have recently been developed, even if they have not been used in the PT literature yet. An example is ACADOS, which is an evolution of ACADO, for rapid testing and deployment of NMPC algorithms on embedded hardware platforms. ACADOS: i) can accelerate the NMPC simulation and sensitivity propagation tasks through the GNSF-IRK integrator; ii) can be easily interfaced with Matlab and Python; iii) uses BLASFEO as linear algebra backend; iv) can adopt CasADi as modelling language; and v) is particularly suitable for AD applications, thanks to its high performance and flexibility (Verschuere et al., 2021).

The real-time capability of the controller is significantly affected by the optimisation solver. To analyse this effect, Quirynen et al. (2018) presents HiL tests of PT NMPC implementations on a Raspberry Pi 2 platform, by considering $T_s = 50$ ms and different $N_p = N_c$ values. The comparison is made by implementing several solvers in ACADO, and evaluating the average and worst-case runtimes. **Table 5** summarises the results for ST and DT prediction models, where the combinations highlighted in bold are real-time implementable. The runtimes are subject to quadratic increase as a function of N_p for dense solvers, such as PQP, ADMM and qpOASES, and linear increase for sparse solvers, such as HPMP and PRESAS.

The performance of the controllers considered in this survey has been tested on a variety of platforms, see **Table 6**, which categorises them into: i) simulation models, in some cases accompanied by real-time verification (RTV), obtained by testing the control strategy on control hardware in open-loop; ii) HiL set-ups, in which the controllers are assessed in real-time in closed-loop; and iii) experimental set-ups involving scale or real vehicles. i)-iii) are ordered according to

Table 6
Overview of testing platforms for MPC-based PT assessment.

Platform	Testing environment	References
Simulation	Matlab	Alcalá et al., 2019a; Hamid et al., 2017a, 2017b; Law et al., 2018; J. Liu et al., 2017; Tan et al., 2018; Yakub et al., 2016
	Matlab – OpenDLV	Batkovic et al., 2019
	Python	Wurts et al., 2021
	Simulink	Geng et al., 2020; F. Lin, Chen et al., 2019; Luciani et al., 2020; Rafalia & Livint, 2015; Reda et al. 2020; Rokonzaman et al., 2020, 2021b; Samuel et al., 2021; Yakub & Mori, 2015
	Simulink – CarSim	J. Cao et al., 2020; M. Cao et al., 2021; M. Chen & Ren, 2017; S. Chen & H. Chen, 2020; W. Chen et al. 2021; X. Chen et al., 2021; Y. Chen et al., 2019; Y. Chen & Wang, 2019; Choi et al., 2018; Cui et al., 2017; S. Feng et al., 2021; Hajiloo et al., 2021; He et al., 2020; C. Hu et al., 2020; J. Hu et al., 2020; Huang et al., 2019; Ji et al., 2017; Lee & Chang, 2018; S. Li et al., 2020; S. Li, G. Wang et al., 2019; S. Li, S. Wang et al., 2019; Liang et al., 2021a, 2021b; F. Lin et al., 2020; F. Lin, Chen et al. 2019; F. Lin, Zhang et al. 2019; Liu & Kang, 2019; Qi et al., 2021; Qin et al., 2021; Ren et al., 2018; C. Sun et al., 2018, 2019; Taghavifar, 2019; H. Wang, 2019; Z. Wang, Bai et al. 2019; Z. Wang, Zha et al. 2019; H. Wu et al., 2020; Xiang et al., 2020; Y. Xu et al., 2017, 2021; Yang et al., 2021; Yao & Tian, 2019; H. Yu et al., 2015; J. Yu et al., 2020; Y. Yu et al., 2021; K. Yuan et al., 2018; H. Yuan et al., 2018; B. Zhang, 2019a, 2019b; K. Zhang et al., 2015; Zou et al., 2019
	IPG CarMaker	Ahn et al., 2021; Chowdhri et al., 2021; Mata et al., 2019; Taherian et al., 2019
	veDYNA – Simulink	Bo et al., 2019; Shen et al., 2017
	Dymola	W. Zhang et al., 2020
	Webots	Qian et al., 2016
	RTV - dSPACE Scalexio	C. Yu et al., 2021
RTV - ARM based Raspberry Pi 2	Quirynen et al., 2018	
HiL	dSPACE MicroAutoBox I	Y. Chen et al., 2020; H. Guo, Shen et al., 2018; Hang et al., 2021; Peng et al., 2019; Z. Wang et al., 2020
	dSPACE MicroAutoBox II	Rosolia et al., 2017
	NI PXI	Du et al., 2016; Luan et al., 2020; H. Wang et al., 2021b
	Speedgoat real-time mobile target machine	Xie et al., 2021 Cesari et al., 2017
Experimental vehicle	Scale car	Nam et al., 2019
	1/43 scale car	T. Lee & Y. Kang, 2021
	1/18 scale car	Z. Wang & J. Wang, 2020
	1/10 scale race car	Hatem, 2018
	1/10 scale Berkeley Autonomous Race Car	Alcalá et al., 2019b; Brunner et al., 2017; Rosolia & Borrelli, 2019
	X1- RWD electric vehicle	Alsterda & Gerdes, 2021; Brown et al., 2016; Erlien et al., 2016; Funke et al., 2015, 2016
	Volkswagen Golf GTI	Alsterda et al., 2019; Laurence & Gerdes, 2021; Spielberg et al., 2021; Zhang et al., 2018
	Fiat Palio Adventure	Massera et al., 2020
	Hyundai-Kia motors K7	Suh et al., 2018
	Hyundai Azera	Kong et al., 2015

(continued on next page)

Table 6 (continued)

Platform	Testing environment	References
	Renault Zoe	Seccamonte et al., 2019
	Hongqi HQ430 autonomous car	H. Guo, Cao et al., 2018
	BYD Qin	C. Zhang et al., 2019
	Dongfeng A60EV	Tang et al., 2020
	Hybrid Lincoln MKZ	Xu & Peng, 2020
	Gotthard – 4WD Race Car	Kabzan et al., 2019
	DYROS	M. Kim et al., 2021
	Real vehicle – not specified	Ye et al., 2019
	Prototype vehicle	J. Guo et al, 2017; Pereira et al., 2017; Tang et al., 2018
	Scania G480 truck	Lima et al., 2017
	LCK6105GZ bus	Cheng et al., 2020

increasing level of sophistication. For each controller implementation, Table 6 reports only the most advanced assessment scenario among those mentioned in the respective publication.

The experimental tests on the vehicle demonstrators in Table 6 have been carried out through rapid control prototyping hardware, e.g., dSPACE MicroAutoBox or Raspberry Pi units, which guarantee superior computing performance and flexibility compared to the current industrial electronic control units for production vehicles. The technological limitations of the available hardware, in conjunction with the high computational power associated with typical implicit MPC implementations, require additional work for industrial MPC-based PT applications. For the same reasons, together with the cost and effort associated with the set-up of demonstrator vehicles, most of the recent MPC PT examples are limited to simulation assessments, see also Fig. 33 (d).

5.2. Use cases

Table 7 is an overview of the testing scenarios for PT control assessment, which are categorised into:

- PT for vehicle operation below the limits of handling, which can be further divided into low-to-medium speed applications ($v < 60$ km/h), and medium-to-high speed applications ($v \geq 60$ km/h).
- PT at the limit or beyond the limit of handling, i.e., in which the lateral tyre forces reach their saturation level, and the slip angles are large (Funke et al., 2016; Massera et al., 2020).

If a control strategy has been tested in different scenarios, the same reference is placed under all relevant categories in the table.

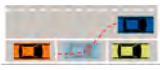
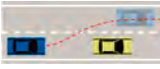

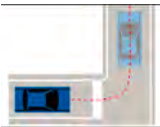
Despite the scenarios in Table 7 do not explicitly refer to specific automotive standards, some requirements and procedures for (semi-) automated vehicle testing – which are therefore relevant to path tracking control – are already available, e.g., see the standards ISO 16787:2017 for assisted parking systems, ISO 21202:2020 for automated lane change, and ISO 22737:2021 for low-speed (max. 8.89 m/s) automated driving on predefined routes. In general, the standardization activities allow: i) to incorporate the minimum safety requirements into the AD system design; and ii) end users, operators and regulators to have a minimum set of performance requirements for their procurements.

5.3. Key performance indicators

The performance of MPC implementations for PT is commonly evaluated through key performance indicators (KPIs). The PT KPIs usually consider the absolute value of the lateral position error, $|e_y|$, and heading angle error, $|e_\psi|$, w.r.t. the curvilinear reference system (Cheng et al., 2020; H. Wang et al., 2021b; K. Zou et al., 2021). The position errors can be alternatively defined w.r.t. a global reference system, in







Table 7

Overview of testing scenarios and path tracking objective in the considered MPC studies.

Scenario	Objective	References
Parking	PT at low-to-medium speed	Ye et al., 2019
		
Lane change	PT at low-to-medium speed	Geng et al., 2020; J. Guo et al., 2018; Lima et al., 2017; Pereira et al., 2017; Rafaila & Livint, 2015
		
	PT at medium-to-high speed	Chen & Ren, 2017; Chowdhri et al., 2021; H. Guo, Cao et al., 2018; Huang et al., 2019; Leman et al., 2019; S. Li, G. Wang et al., 2019; S. Li, S. Wang et al., 2019; Peng et al., 2019; Ren et al., 2018; Rokonuzzaman et al., 2021b; Suh et al., 2018; Vallon et al., 2017; H. Wang et al., 2019; Z. Wang et al., 2019b; C. Yu et al., 2021; Y. Yu et al., 2021; K. Yuan et al., 2018
	PT at the limit of handling	Funke et al., 2015; Wurts et al., 2021
Double lane change	PT at low-to-medium speed	Alsterda & Gerdes, 2021; Berntrorp et al., 2019; Hu et al., 2020; C. Huang et al., 2016; Li et al., 2020; Nam et al., 2019; Qian et al., 2016; Quirynen et al., 2018; H. Wang et al., 2019; K. Yang et al., 2021
		
	PT at medium-to-high speed	Ahn et al., 2021; Bo et al., 2019; Cao et al., 2020; X. Chen et al., 2021; W. Chen et al., 2021; Y. Chen et al., 2020; Dai et al., 2020; Dallas et al., 2020; Feng et al., 2021; Hamid et al., 2017a, 2017b; Hatem, 2018; He et al., 2020, 2021; Huang et al., 2019; Ji et al., 2017; Karimshoushtari et al., 2021; Li et al., 2019a; Liang et al., 2021a; F. Lin, Chen et al. 2019; F. Lin, Zhang et al. 2019; Luan et al., 2020; Mata et al., 2019; Rokonuzzaman et al., 2021b; Shen et al., 2017; Taghavifar, 2019; Taherian et al., 2019; H. Wang & Liu, 2021; Z. Wang et al., 2019a; H. Wu et al., 2020; Xie et al., 2021; Y. Xu et al., 2021; Yakub et al., 2016; C. Yu et al., 2021; J. Yu et al., 2019; Y. Yu et al., 2021; H. Yuan et al., 2018; B. Zhang et al., 2019; W. Zhang et al., 2020; K. Zou et al., 2019; Y. Zou et al., 2019
	PT at the limit of handling	Erlie et al., 2016; Hajiloo et al., 2021; Hang et al., 2021; Wurts et al., 2021
Turn, J-turn, S-turn, U-turn	PT at low-to-medium speed	Batkovic et al., 2019; Chen et al., 2019; Chen & Ren, 2017; Chen & Wang, 2019; Hashemi et al., 2021; Law et al., 2018; Liu & Kang, 2019; Qin et al., 2021; Wang et al., 2020
		
	PT at medium-to-high speed	Ahn et al., 2021; M. Cao et al., 2021; Choi et al., 2018; Peng et al., 2019; Ren et al., 2018; Rokonuzzaman et al., 2019; Tonoli et al., 2020; Yakub & Mori, 2015; Y. Yu et al., 2021
	PT at the limit of handling	Alsterda et al., 2019; Brown et al., 2016; Funke et al., 2015, 2016; Liang et al., 2021b; Sun et al., 2018; V. Zhang, 2018
Elliptical or sinusoidal paths	PT at low-to-medium speed	Hajiloo et al., 2021; Xu & Peng, 2020; Yu et al., 2015

(continued on next page)

Table 7 (continued)

Scenario	Objective	References
	PT at medium-to-high speed PT at the limit of handling	J. Guo et al., 2017; Kim et al., 2018; Kong et al., 2015; Samuel et al., 2021 Hang et al., 2021; Qi et al., 2021
	PT at low-to-medium speed	Chen & Chen, 2020; Codrean et al., 2020; Dawood et al., 2020; H. Guo, Cao et al., 2018; He et al., 2021; T. Lee & Kang, 2021; Tan et al., 2018; Rokouzzaman et al., 2020; C. Zhang et al., 2019
	PT at medium-to-high speed PT at the limit of handling	Dai et al., 2020; Lu et al., 2018; Mata et al., 2019 Alcalá et al., 2019a, 2019b; Brunner et al., 2017; Kabzan et al., 2019; Laurence & Gerdes, 2021; Massera et al., 2020; Rosolia et al., 2017; Rosolia & Borrelli, 2019; Spielberg et al., 2021; Sun et al., 2018
	PT at low-to-medium speed	Cheng et al., 2020; He et al., 2021; M. Kim et al., 2021; Lima et al., 2015; Seccamonte et al., 2019; C. Yu et al., 2021; K. Zhang et al., 2015
	PT at medium-to-high speed	Lee & Chang, 2018; Tang et al., 2018; Sun et al., 2019; H. Wang et al., 2021a, 2021b; Xu et al., 2017; Yao & Tian, 2019
	PT at the limit of handling	Spielberg et al., 2021

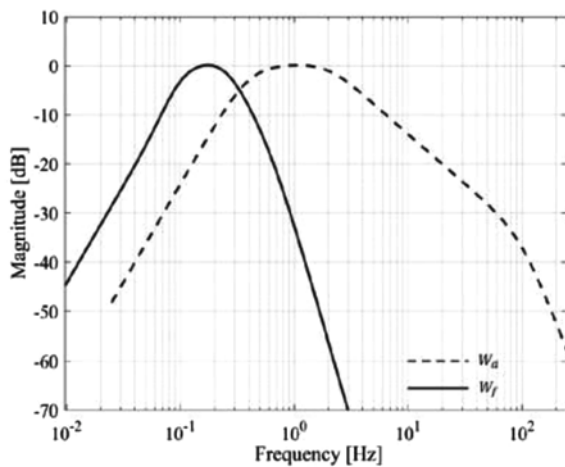


Fig. 12. Magnitude plot of the weighting functions W_d and W_f in ISO 2631-1:1997 (from Luciani et al., 2020).

terms of $|X_{ref} - X|$ and $|Y_{ref} - Y|$ (Ji et al., 2017; Suh et al., 2018; Taghavifar, 2019; Z. Tang et al., 2019; B. Zhang, 2019a).

Further indicators are often introduced to evaluate the handling and stability performance in terms of yaw rate and sideslip angle errors:

$$\begin{aligned} |\dot{e}_\psi| &= |\dot{\psi}_{ref} - \dot{\psi}| \\ |\Delta\beta| &= |\beta_{ref} - \beta| \end{aligned} \quad (77)$$

where the reference yaw rate, $\dot{\psi}_{ref}$, and sideslip angle, β_{ref} , can be calculated with the simplified steady-state formulations deriving from the linear ST model (Luan et al., 2020; Xiang et al., 2020):

$$\dot{\psi}_{ref}(t) = \frac{v_x}{l_f + l_r + K_\delta v_x^2} \quad (78)$$

$$\beta_{ref,ss} = \frac{l_r - \frac{v_x^2 l_r m}{2C_{y,r}(l_f + l_r)}}{l_f + l_r + K_\delta v_x^2} \delta \quad (79)$$

in which K_δ is the understeer coefficient. In the PT controllers that are also responsible for the longitudinal vehicle dynamics, an additional relevant variable is the speed tracking error, $|e_{v_x}|$.

To assess the real-time feasibility, a common indicator is the computational load index CI (Z. Wang, Bai et al., 2019, Z. Wang et al., 2020):

$$CI = \frac{T_c}{T_{s,c}} \quad (80)$$

where T_c is the total CPU execution time, also known as turnaround time, and $T_{s,c}$ is controller sampling time to find the OCP solution (which is generally different from the discretisation time of the internal model).

In autonomous racing application, the assessment is done in terms of lap time $t_{lap,i}$ (Alcalá et al., 2019b; Brunner et al., 2017; Kabzan et al., 2019; Rosolia et al., 2017; Rosolia & Borrelli, 2019):

$$t_{lap,i} = t_{fin,i} - t_{ini,i} \quad (81)$$

where $t_{ini,i}$ and $t_{fin,i}$ are the starting and finishing times, which are evaluated at each lap l .

A challenge of PT control is the integration of MS mitigation strategies, to provide not only safe but also comfortable user experience (Iskander et al., 2019). KPIs such as the lateral and longitudinal acceleration peaks (Luciani et al., 2020), the mean absolute value of the lateral jerk, $MAV_{\dot{a}_y}$ (Lima et al., 2015), the root mean square values of the lateral acceleration, RMS_{a_y} , and longitudinal jerk, $RMS_{\dot{a}_x}$ (Karimshoushtari et al., 2021), are sometimes used as indicators of comfort and MS aspects. For evaluating PT comfort, Luciani et al. (2020) uses two indices from ISO 2631-1:1997:

- a_{eq} , i.e., an equivalent acceleration that is calculated as the weighted sum, based on the squared axis coefficients k_x and k_y , of the root mean square values ($RMS_{a_{w_d,i_a}}$, with $i_a = x, y$) of the frequency weighted (with weighting function W_d , see Fig. 12) acceleration components along the longitudinal and lateral axes of the vehicle reference system:

$$\begin{aligned} RMS_{a_{w_d,i_a}} &= \sqrt{\frac{1}{T_2 - T_1} \int_{T_1}^{T_2} a_{w_d,i_a}^2 dt} \\ a_{eq} &= \sqrt{k_x^2 RMS_{a_{w_d,x}}^2 + k_y^2 RMS_{a_{w_d,y}}^2} \end{aligned} \quad (82)$$

The comfort performance evaluation is obtained according to Table 8.

- a_{MSDV} , i.e., the motion sickness dose value, which considers the cumulated effect of analysed acceleration component, and uses the weighting function W_f in Fig. 12:

Table 8

Likely user reactions to the magnitude of a_{eq} , according to ISO 2631-1:1997 (Luciani et al., 2020).

[m/s ²]	User acceptance
$a_{eq} \leq 0.315$	Not uncomfortable
$0.315 \leq a_{eq} \leq 0.63$	A little uncomfortable
$0.5 \leq a_{eq} \leq 0.1$	Fairly uncomfortable
$0.8 \leq a_{eq} \leq 1.6$	Uncomfortable
$1.25 \leq a_{eq} \leq 2.5$	Very uncomfortable
$a_{eq} \geq 2$	Extremely uncomfortable

Table 9
Overview of the key performance indicators (KPIs) from the considered MPC-based PT studies.

Definition	Formulation	References
Maximum lateral tracking error	$e_{y,max} = \max(e_y)$ $\Delta Y_{max} = \max(Y_{ref} - Y)$	Y. Chen et al., 2020; Hang et al., 2021; Lima et al., 2015; Lu et al., 2018; Luciani et al., 2020; Sun et al., 2018; Taghavifar, 2019; H. Wu et al., 2020; Yao & Tian, 2019; C. Yu et al., 2021; W. Zhang et al., 2020
Maximum heading angle error	$e_{\psi,max} = \max(e_{\psi})$	Y. Chen et al., 2020; Hang et al., 2021; Lu et al., 2018; Luciani et al., 2020; Taghavifar, 2019; H. Wu et al., 2020; Yao & Tian, 2019; C. Yu et al., 2021; W. Zhang et al., 2020
Maximum yaw rate error	$e_{\dot{\psi},max} = \max(\dot{\psi}_{ref} - \dot{\psi})$	Hang et al., 2021; W. Zhang et al., 2020
Maximum sideslip angle error	$\Delta\beta_{max} = \max(\beta_{ref} - \beta)$	Hang et al., 2021; W. Zhang et al., 2020
Maximum lateral acceleration	$a_{y,max} = \max(a_y)$	Luciani et al., 2020
Standard deviation of the lateral tracking error	$\sigma_{e_y} = \sqrt{\frac{1}{n} \sum_{i=1}^n e_{y,i}^2}$	Sun et al., 2018
Root mean square value of the lateral tracking error	$RMS_{e_y} = \sqrt{\frac{1}{T_2 - T_1} \int_{T_1}^{T_2} e_y^2 dt}$	J. Cao et al., 2020; Y. Chen et al., 2020; Chowdhri et al., 2021; Hang et al., 2021; Z. He et al., 2020; C. Hu et al., 2020; Karimshoushtari et al., 2021; Leman et al., 2019; Liang et al., 2021a; Lima et al., 2015; Rokonuzzaman et al., 2021b; Taghavifar, 2019; Z. Wang et al., 2020; H. Wu et al., 2020; Xie et al., 2021; Yakub & Mori, 2015; K. Yang et al., 2021; W. Zhang et al., 2020
Root mean square value of the heading angle error	$RMS_{\Delta Y} = \sqrt{\frac{1}{T_2 - T_1} \int_{T_1}^{T_2} [Y_{ref} - Y]^2 dt}$ $RMS_{e_{\psi}} = \sqrt{\frac{1}{T_2 - T_1} \int_{T_1}^{T_2} e_{\psi}^2 dt}$	J. Cao et al., 2020; Y. Chen et al., 2020; Chowdhri et al., 2021; Hang et al., 2021; Z. He et al., 2020; C. Hu et al., 2020; Liang et al., 2021a; Rokonuzzaman et al., 2021b; Taghavifar, 2019; Z. Wang et al., 2020; H. Wu et al., 2020; Yakub & Mori, 2015; K. Yang et al., 2021; W. Zhang et al., 2020
Root mean square value of the yaw rate error	$RMS_{\dot{\psi}} = \sqrt{\frac{1}{T_2 - T_1} \int_{T_1}^{T_2} \dot{\psi}_{ref} - \dot{\psi} ^2 dt}$	J. Cao et al., 2020; Chowdhri et al., 2021; Hang et al., 2021; Leman et al., 2019; K. Yang et al., 2021; W. Zhang et al., 2020
Root mean square value of the vehicle sideslip angle	$RMS_{\Delta\beta} = \sqrt{\frac{1}{T_2 - T_1} \int_{T_1}^{T_2} \beta_{ref} - \beta ^2 dt}$	Hang et al., 2021; Xie et al., 2021; W. Zhang et al., 2020
Root mean square value of the lateral acceleration	$RMS_{a_y} = \sqrt{\frac{1}{T_2 - T_1} \int_{T_1}^{T_2} [a_{y,ref} - a_y]^2 dt}$	Karimshoushtari et al., 2021
Root mean square value of the steering rate	$RMS_{\delta} = \sqrt{\frac{1}{T_2 - T_1} \int_{T_1}^{T_2} [\delta_{ref} - \delta]^2 dt}$	Karimshoushtari et al., 2021
Mean absolute value of the lateral jerk	$MAV_{\dot{a}_y} = \frac{1}{n} \sum_{i=1}^n \dot{a}_{y,ref,i} - \dot{a}_{y,i} $	Lima et al., 2015
Mean square value of the lateral trajectory tracking error	$MSE_{e_y} = \frac{1}{n} \sum_{i=1}^n e_{y,i}^2$	Farag, 2020; Lu et al., 2018; Yao & Tian, 2019
Mean square value of the yaw angle error	$MSE_{e_{\psi}} = \frac{1}{n} \sum_{i=1}^n e_{\psi,i}^2$	Lu et al., 2018; Yao & Tian, 2019
Lateral trajectory tracking index	$Q_{track,e_y} = \sqrt{\frac{\frac{t_{sim}}{T_s} \sum_{i=1}^{\frac{t_{sim}}{T_s}} [Y_{ref,i} - Y_{iv}]^2}{\frac{t_{sim}}{T_s} - 1}}$	B. Zhang, 2019b
Heading angle error index	$Q_{track,e_{\psi}} = \sqrt{\frac{\frac{t_{sim}}{T_s} \sum_{i=1}^{\frac{t_{sim}}{T_s}} e_{\psi,i}^2}{\frac{t_{sim}}{T_s} - 1}}$	B. Zhang, 2019b
Tracking index	$TI = \frac{\int_0^{t_{fin}} \min(Y_{max} - \max(Y_{ij}); \max(Y_{ij}) - Y_{min})_{norm} dt}{t_{fin}}$	Z. Wang, Bai et al., 2019
Stability index	$SI = \frac{\int_0^{t_{fin}} \min\left(\min\left(1 \pm \frac{\beta}{\beta_{lim}}\right); \min\left(1 \pm \frac{\dot{\psi}}{\dot{\psi}_{lim}}\right)\right)_{norm} dt}{t_{fin}}$	Z. Wang, Bai et al., 2019
Maximum value of the speed tracking error	$\Delta v_{x,max} = \max(v_{x,ref}(x) - v_x(x))$	H. Wu et al., 2020
Maximum absolute value of the longitudinal acceleration	$\Delta a_{x,max} = \max(a_x)$	Luciani et al., 2020
Root mean square value of the speed tracking error	$RMS_{v_x} = \sqrt{\frac{1}{T_2 - T_1} \int_{T_1}^{T_2} [v_{x,ref} - v_x]^2 dt}$	C. Hu et al., 2020; Z. Wang et al., 2020; H. Wu et al., 2020; K. Yang et al., 2021
Root mean square value of the longitudinal jerk	$RMS_{\dot{a}_x} = \sqrt{\frac{1}{T_2 - T_1} \int_{T_1}^{T_2} [\dot{a}_{x,ref} - \dot{a}_x]^2 dt}$	Karimshoushtari et al., 2021
Mean value of the longitudinal speed	$MAV_{v_x} = \frac{1}{n} \sum_{i=1}^n [v_{x,ref,i} - v_{x,i}]$	Z. He et al., 2020

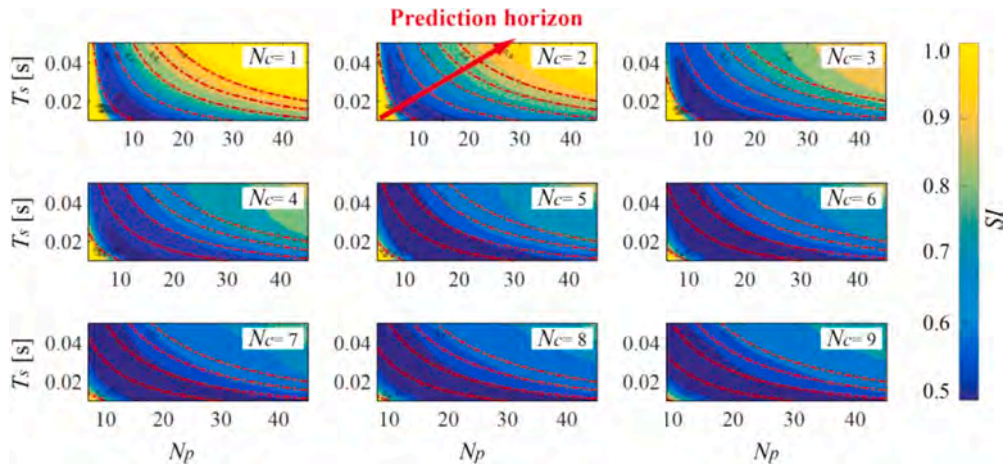


Fig. 13. Contour and colour plots of SI as a function of N_p and T_s , for different values of N_c (adapted from Z. Wang, Bai et al., 2019).

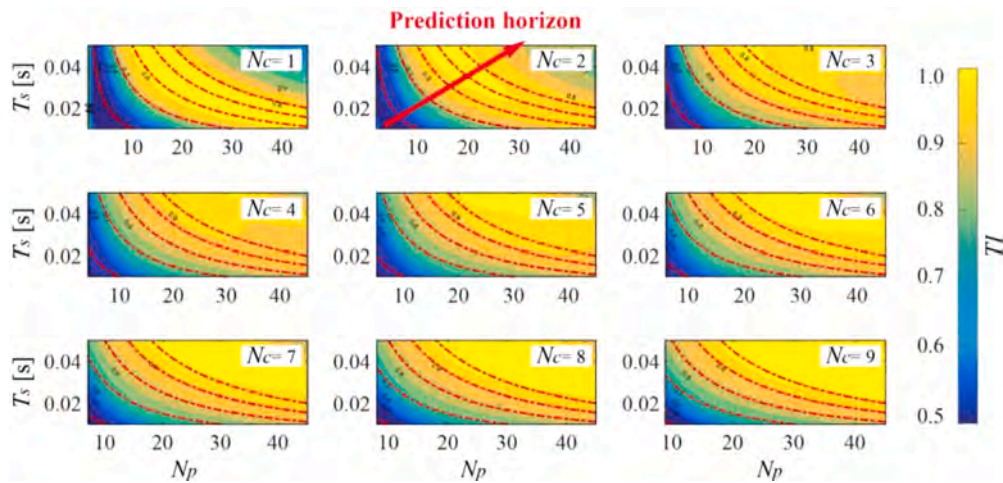


Fig. 14. Contour and colour plots of TI as a function of N_p and T_s , for different values of N_c (adapted from Z. Wang, Bai et al., 2019).

$$a_{MSDV} = k_m \sqrt{\int_{T_1}^{T_2} a_{w_{f,i_d}} dt} \tag{83}$$

where k_m is a constant coefficient.

Table 9 summarises the other KPIs that are used in the considered literature to evaluate the tracking performance as well as the handling stability and ride comfort levels provided by MPC-based PT.

5.4. Selection of relevant results

5.4.1. Effect of T_s , N_c and N_p on the path tracking performance

This section deals with the selection of the sampling time, T_s , and the numbers of steps, N_c and N_p , of the control and prediction horizons, to achieve desirable PT performance without sacrificing vehicle stability and real-time feasibility.

Table 10

Comparison of NMPC performance with different prediction horizon lengths (from C. Hu & Zhao, 2020).

Method	RMS_{e_y} [m]	RMS_{e_ψ} [rad]	RMS_{v_x} [m/s]	Computation time [s]
NMPC-VPH	2.4633	0.0092	0.0553	203.875
NMPC-CPH-3	2.6505	0.137	0.1007	167.376
NMPC-CPH-5	2.4396	0.0103	0.1180	185.112
NMPC-CPH-9	2.7287	0.0141	0.1208	371.268

Z. Wang, Bai et al. (2019) carries out a sensitivity analysis along double lane change manoeuvres from $v_x = 30$ m/s, at $\mu = 0.8$, by individually varying T_s (from 0.01 s to 0.05 s, with 5 ms increments), N_c (from 1 to 9, with unit increments), and N_p (from N_c to 45, with unit increments), within an LTV-MPC implementation. The assessment is based on the vehicle stability index, SI , reported in the contour plot in Fig. 13, the tracking index, TI , included in Fig. 14, and the computational load index, CI . SI (see Table 9) improves as the prediction horizon increases, regardless of the control horizon. In fact, higher H_p values bring reduced oscillations of the closed-loop system, as confirmed by H. Wang et al. (2021b), and C. Hu & Zhao (2020). On the contrary, longer H_c tends to increase the PT control aggressiveness, and to reduce vehicle stability. The dependency of TI on H_p is more complex. For short prediction horizons, the tracking index improves by increasing H_p , which should be larger than the system settling time, T_{set} , to comprehensively account for the plant dynamics. On the contrary, TI tends to worsen for long H_p , because of: i) the cumulative error caused by the mismatch

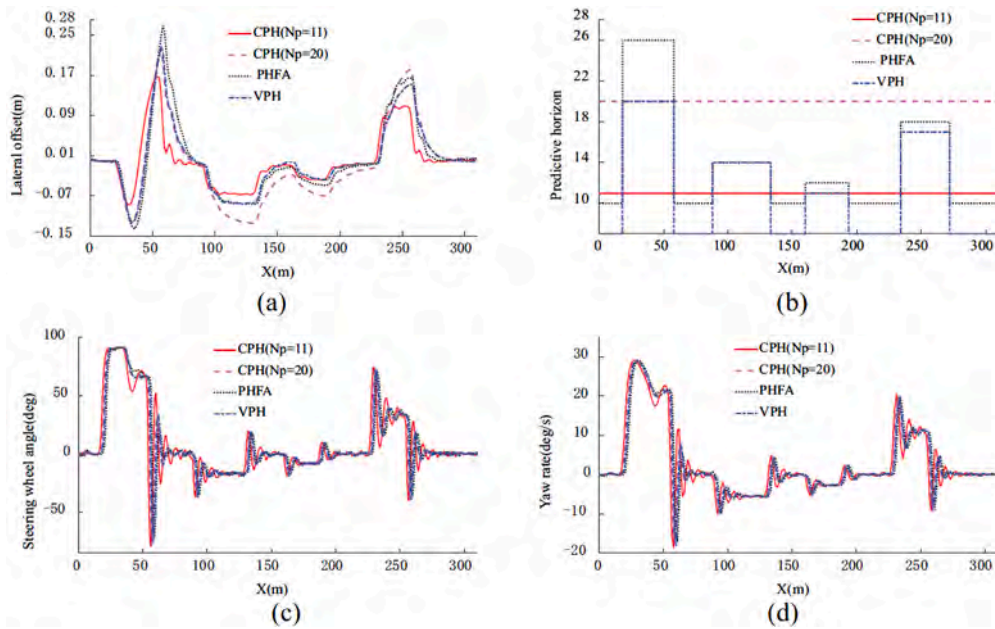


Fig. 15. Experimental results from H. Wang et al., 2021b: a) lateral offset; b) prediction horizon; c) steering wheel angle; d) yaw rate.

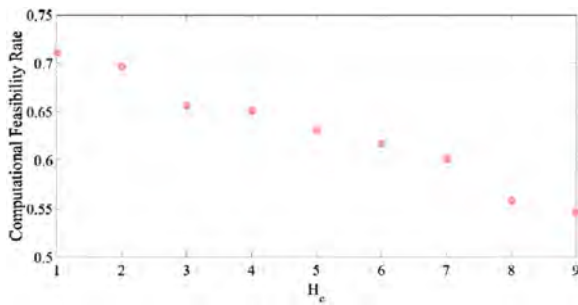


Fig. 16. Computational feasibility rate as a function of the control horizon H_c (from Z. Wang, Bai et al., 2019).

between real plant and prediction model; and ii) the reduction of the swiftness of the closed-loop system response, which impairs vehicle agility. The negative side effects of long H_p can be partially neutralised by long H_c . In fact, B. Zhang et al., 2019b shows that increasing N_c from 4 to 36 improves the tracking indices, Q_{track,e_y} and Q_{track,e_ψ} , by $\sim 20\%$ and

$\sim 13\%$, along a double lane change from 15 m/s.

The trend in Fig. 14 is also confirmed in C. Hu & Zhao (2020), which analyses the tracking performance during a simulated overtaking manoeuvre from 11 m/s, through the RMS values, reported in Table 10, of the lateral position error, yaw angle error, and velocity error, for three constant prediction horizon lengths (3, 5, and 9 steps), and one variable prediction horizon (VPH) set-up, where the variation is a function of the smoothness of the reference path. The VPH scope is to reduce the computational burden while still guaranteeing the high accuracy of long H_p for non-smooth trajectories (H. Wang et al., 2021b; B. Zhang et al., 2019b). H. Wang et al. (2021b) further investigates the VPH idea in the context of LTV-MPC, and adopts a particle swarm optimisation algorithm for selecting the best H_p , based on a cost function considering PT, comfort, and real-time feasibility indices. The VPH algorithm is experimentally tested on a HiL platform, and compared with two implementations with constant prediction horizon (CPH) lengths, and a controller with the prediction horizon fitting algorithm (PHFA) from Bai et al. (2020). Fig. 15(a) shows that the PT accuracy with VPH is slightly worse than with CPH and $N_p = 11$, but is higher than with the other two methods. The analysis of the steering frequency and amplitude, see

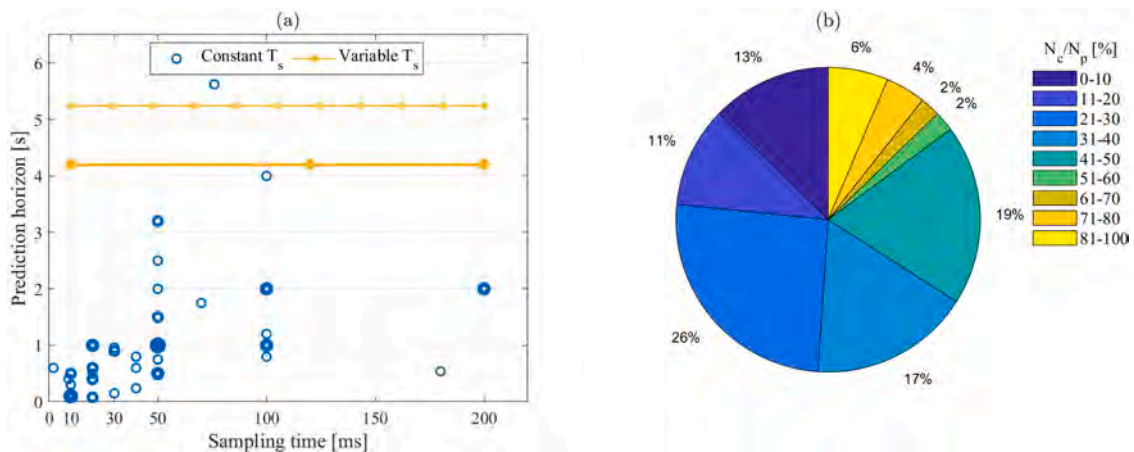


Fig. 17. Overview of the considered controller settings: (a) Sampling time and corresponding prediction horizon (explicitly stated for 74 implementations), where the thickness of the lines is proportional to the number of controllers with the same settings; and (b) N_c/N_p distribution (explicitly stated for 42 controllers).

Fig. 15(c), highlights that the proposed VPH strategy behaves similarly to the algorithm with CPH and $N_p = 20$ and the one with the PHFA, while the control input is significantly less aggressive than for CPH and $N_p = 11$. The yaw rate behaviour in Fig. 15(d) confirms that the VPH implementation provides good driving comfort. The average runtime, T_{ave} , required for a CompactRIO unit to process a single sampling period, and the number, N_{sum} , of sampling periods are analysed for each method, to evaluate the real-time capability of the controllers. The T_{ave} values are 5.42 ms, 6.79 ms, 6.15 ms, and 5.03 ms, respectively for the PT controllers with CPH and $N_p = 11$, CPH and $N_p = 20$, PHFA, and VPH, which proves the computational benefit of VPH. Berntorp et al. (2019) and Z. Wang, Bai et al. (2019) highlight that the computational load is significantly affected by the choice of H_c . For example, Fig. 16 shows that the computational feasibility rate, i.e., the fraction of combinations of H_p and $T_{s,c}$ (assumed to be equal to the sampling time of the prediction model, T_s) that ensure that CI is less than 1 significantly decreases as H_c increases.

In summary, as it is difficult to select values of H_c , H_p and T_s providing desirable response across the variety of vehicle operating conditions in terms of speed, shape of the reference path, and level of severity of the resulting cornering response, the control settings should be adapted accordingly, as recommended in F. Lin, Chen et al. (2019).

Fig. 17 summarises the T_s and H_p values, together with the N_c / N_p ratio (in percentage), for the controllers in the considered literature. In Fig. 17(a), each circle refers to an implementation with fixed discretisation along H_p , while the horizontal lines refer to controllers with variable discretisation. A higher concentration of points occurs for low values of T_s and H_p . Such combinations guarantee a good compromise between tracking accuracy and computational effort. Longer H_p , e.g., in excess of 4 s, can benefit performance in emergency scenarios. Many studies state that the control horizon is shorter than the prediction horizon, but only 42 papers explicitly report the value of H_c . Interestingly, Fig. 17(b) shows that ~86% of controller implementations assume the control horizon to be less than half than the prediction horizon. In fact, the partial reduction of tracking accuracy is largely compensated by the improvement of the real-time computational feasibility for the same H_p , and by the possibility of adopting longer H_p for the same computational load.

5.4.2. Comparisons of MPC implementations for path tracking control

Only a limited number of studies, reported in this section, includes comparisons among different MPC implementations for PT, in terms of tracking performance and computational effort. The following subsections focus on the comparisons between: i) MPC formulations using physics-based prediction models; and ii) NNMPCC and physics-based

MPC.

Comparisons of MPC formulations using physics-based prediction models – A comparison is presented by Z. Wang et al. (2020), which proposes a two-layer PT architecture consisting of: i) a high-level flatness model predictive controller (FMPC), responsible for generating the global longitudinal and lateral forces and yaw moment, respectively F_x , F_y and M_z , for stabilising the flat outputs defined in terms of lateral error \tilde{e}_y , heading angle error \tilde{e}_ψ , and velocity tracking error \tilde{v} ; and ii) a low-level controller, which allocates the global forces and moment into the reference steering angles and wheel torque levels. Real-time simulations on a HiL platform allow the comparison of the proposed FMPC with corresponding LTV-MPC and NMPC formulations. The RMS of the flat outputs for a cornering test from the initial speed of 20 m/s are summarised in Table 11. The FMPC reduces the RMS of the lateral position error by ~75% and ~45 %, and CI in (80) by 6.3% and ~75%, w.r.t. the LTV-MPC and the NMPC. Despite this, as stated by the authors, a fair comparison among the approaches is not entirely possible, because of: i) the lack of optimised weight tuning for the benchmarking LTV-MPC and NMPC; and ii) the different solver choices for the controllers, namely CVXGEN for the FMPC and LTV-MMPC, and GRAMPC in ACADO for the NMPC. Although the GRAMPC settings have been adjusted to make the two solvers as similar as possible, the numerical solver plays an important role in the PT and computational performance (Quirynen et al., 2018).

In Liang et al. (2021a), the advantages of an LPV-MPC formulation based on a multiple-model adaptive predictive controller (MMAPC) for addressing tyre cornering stiffness uncertainty, see Section 3.5, are assessed w.r.t. a nominal LTV-MPC, and the robust guaranteed-cost (RGC) PT controller by J. Guo et al. (2018), see the RMS_{e_y} and RMS_{e_ψ} values in Table 12, along double lane changes from 65 km/h, in high ($\mu = 0.8$) and low ($\mu = 0.3$) tyre-road friction conditions. While the specific LTV-MPC lacks robustness in low- μ conditions, in which it is not able to follow the reference path, the LPV-MPC provides the best performance among the controllers for $\mu = 0.8$, and is comparable to the RGC controller for $\mu = 0.3$. The inclusion of the multiple model algorithm increases the average turnaround time by 5.4 ms w.r.t. the nominal MPC, but does not compromise the real-time capability.

The aim of the comparison in J. Cao et al. (2020) is to verify whether the addition of tyre slip angle and sideslip angle constraints improves the PT performance. Three controllers are considered along simulated double lane changes with $\mu = 0.4$, from 12.5 m/s and 25 m/s: i) controller 1, i.e., the proposed LTV-MPC, which includes constraints on tyre slip angles, sideslip angle, and other variables, to improve vehicle stability; ii) controller 2, i.e., the same LTV-MPC as in i), which, however, excludes the cornering stability constraints; and iii) controller 3, i.

Table 11

Comparison of the tracking performance of different MPC approaches (from Z. Wang et al., 2021).

Method	RMS_{e_y} [m]	RMS_{e_ψ} [rad]	$RMS_{\dot{y}}$ [m/s]
LTV-MPC	2.6925	0.1244	0.3319
NMPC	0.6393	0.0953	0.2938
FMPC	0.3481	0.1141	0.2862

Table 12

Comparison of the tracking performance of different MPC approaches (from Liang et al., 2021a).

Tyre-road friction condition	Method	RMS_{e_y} [m]	RMS_{e_ψ} [deg]
$\mu = 0.8$	LPV-MPC (Liang et al., 2021a)	0.0153	0.3618
	LTV-MPC (Liang et al., 2021a)	0.0161	0.3669
	RGC (J. Guo et al., 2018)	0.0377	0.5199
$\mu = 0.3$	LPV-MPC (Liang et al., 2021a)	0.0481	1.6590
	LTV-MPC (Liang et al., 2021a)	0.5758	17.1125
	RGC (J. Guo et al., 2018)	0.0737	0.7822

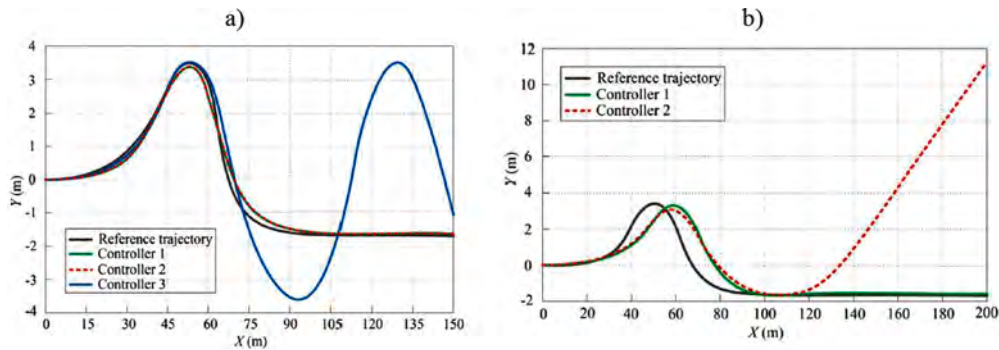


Fig. 18. Simulation results from J. Cao et al. (2020): reference and actual trajectories at 12.5 m/s (subplot (a)) and at 25 m/s (subplot (b)).

Table 13

Comparison of the computational performance of different MPC approaches (from J. Cao et al., 2020).

Method	Hardware	Average runtime [ms]	Maximum runtime [ms]
NMPC (Z. Wang et al., 2018)	4 GB RAM and an Intel Core i5-4590 3.30 GHz	58.4	350
LTV-MPC (Z. Wang et al., 2018)	4 GB RAM and an Intel Core i5-4590 3.30 GHz	23.8	67.0
NMPC (Abbas et al., 2012)	4 GB RAM and an Intel Core i7-2620M 2.27 GHz	27.8	39.3
LTV-MPC (J. Cao et al., 2020)	Not specified	22.1	35.8

e., the optimal preview-based PT controller available in CarSim. The results in Fig. 18(a)-(b) show that controller 1 – differently from controllers 2 and 3 – guarantees desirable trajectory tracking and vehicle stability, and is well adaptable to the different vehicle speeds. The same study includes a computational effort comparison with the NMPC and LTV-MPC based on quadratic programming in Z. Wang et al. (2018), and the NMPC in Abbas et al. (2012), see the average and maximum runtimes in Table 13. However, the data refer to simulations carried out by the different authors in their respective studies, using different

controller settings and solvers, which limits the reliability of the assessment and conclusions.

Along a lane change from 80 km/h with $\mu = 0.3$, and a double lane change from 110 km/h with $\mu = 0.5$, S. Li, G. Wang et al. (2019) compares the proposed LTV-MPC with cornering stiffness prediction, described in Section 3.5, with: i) an LTI-MPC with a linear tyre model; and ii) an NMPC with a nonlinear UniTyre model. The LTI-MPC is unable to maintain vehicle control, see Fig. 19(a)-(b) referring to the double lane change, and the steering angle reaches its saturation limit, see

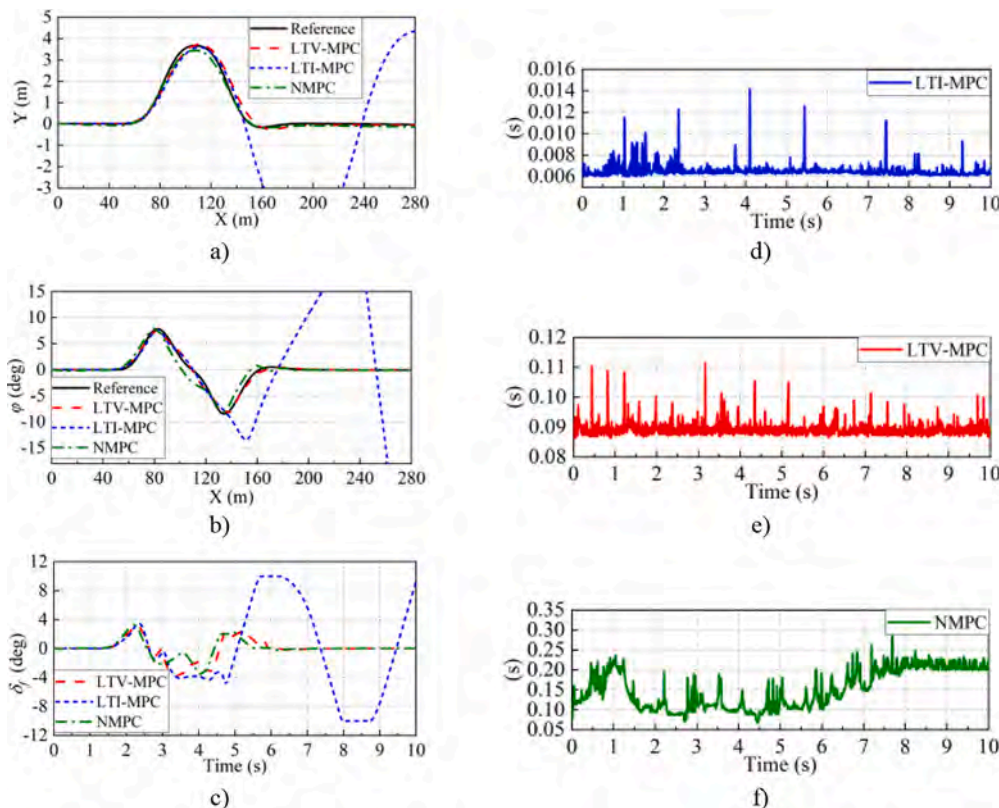


Fig. 19. Simulation results along double lane change tests: (a) lateral position; (b) heading angle; (c) steering angle; (d), (e), and (f) runtimes of LTI-MPC, LTV-MPC and NMPC (from S. Li, G. Wang et al., 2019).

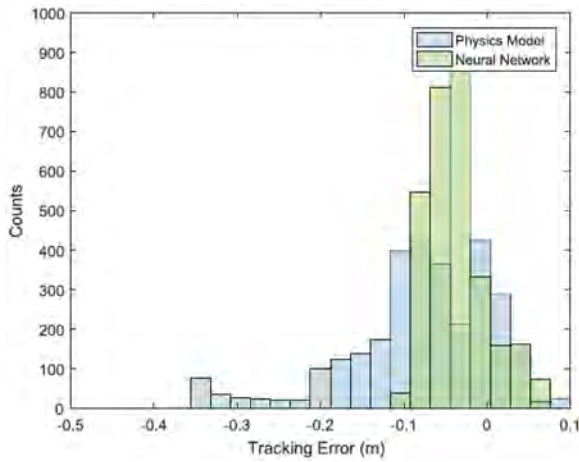


Fig. 20. Experimental results along an oval track with low tyre-road friction conditions: lateral tracking error distribution for the proposed NNMPC and a benchmarking physics-based MPC (from Spielberg et al., 2021).

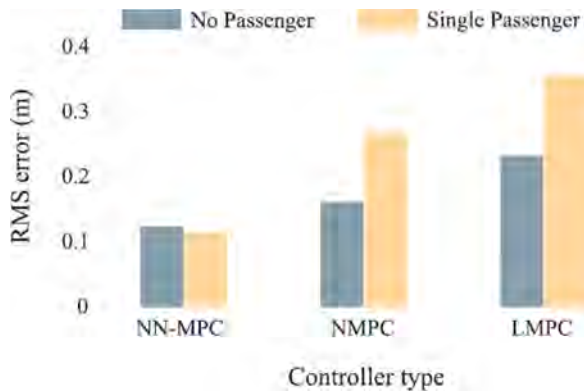


Fig. 21. Tracking error comparison of the considered controllers for two vehicle load conditions, along a double lane change (from Rokonzuzaman et al., 2021b).

Fig. 19(c). On the contrary, the LTV-MPC generates small steering angle fluctuations, and ensures good tracking of the reference position and heading angle, with comparable performance to the NMPC. The average computation time, see Fig. 19(d)-(f), increases by an order of magnitude from LTI-MPC to LTV-MPC, and by less than a factor 2 from LTV-MPC to NMPC. The conclusion is that the LTV-MPC approach provides a good trade-off between PT and computational performance, which is confirmed by the extreme popularity of this controller typology among the researchers, see the following Fig. 33.

Comparison of NNMPC and physics-based MPC – To highlight the NNMPC ability to operate in highly dynamic scenarios with significant uncertainties, Spielberg et al. (2021) presents experimental PT results on dry and icy pavements, and compares the trajectory tracking performance of the proposed NNMPC with that of an MPC using a bicycle model as prediction model. Differently from the physics-based MPC, the NNMPC does not need the estimation of the tyre-road friction level. Fig. 20 summarises the results on an oval test track with $\mu = 0.25$, where the NNMPC is associated with a maximum lateral tracking error of ~ 0.1 m, against the significant 0.38 m of the bicycle-model-based MPC, occurring at the entry of each turn, because of the understeer induced by the excessive steering angle magnitude.

Similarly to Spielberg et al. (2021), the simulation study in Rokonzuzaman et al. (2021b) presents a comparison between a proposed NNMPC using an offline-trained neural network, and conventional LTI-MPC (called LMPC in the study) and NMPC implementations, along

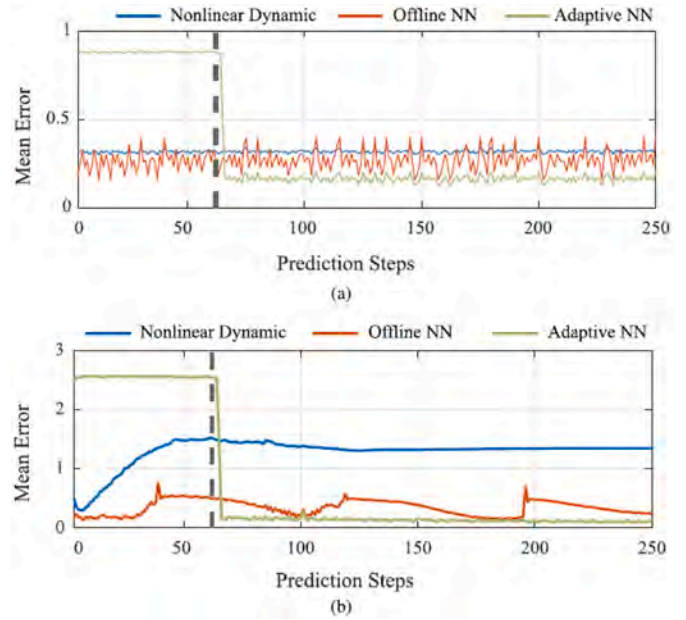


Fig. 22. Mean error of the 8-step prediction for vehicle operation at: (a) constant steering angle; and (b) increasing steering angle (from Rokonzuzaman et al., 2021b).

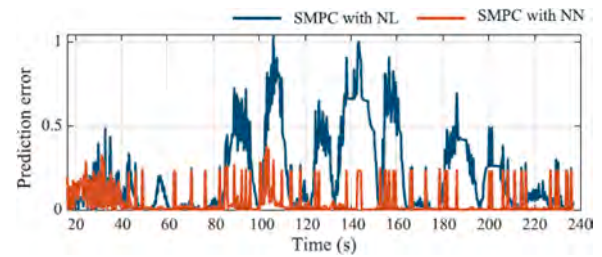


Fig. 23. Prediction error comparison during the SMPC tracking task (from Rokonzuzaman et al., 2021b).

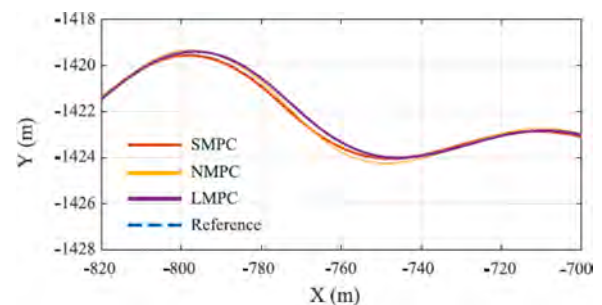


Fig. 24. Extract of the comparison of the trajectories generated by the considered PT controllers (from Rokonzuzaman et al., 2021b).

single and double lane changes for different tyre-road friction coefficients and vehicle payloads, at 60 km/h. The comparison of the trajectories and yaw angles, as well as the resulting RMS values of the lateral tracking error in Fig. 21, highlights that NNMPC improves tracking performance and robustness w.r.t. the vehicle load variation.

Further performance improvement is achievable through the adoption of online-trained adaptive neural networks (NNs) as prediction models. In this respect, Fig. 22 compares the prediction accuracy of: i) a physics-based nonlinear model; ii) an offline-trained NN; and iii) an online-trained NN. The graph reports the mean error along an 8-step

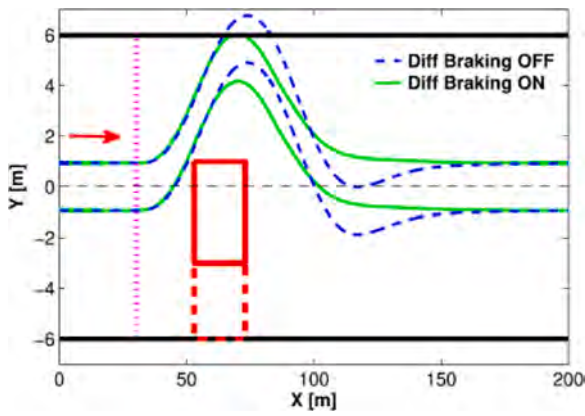


Fig. 25. Simulation results from Hajiloo et al. (2021): vehicle envelope profiles without and with differential braking, when an obstacle (in red) suddenly pops up on a straight road.

prediction, corresponding to the controller prediction horizon, during constant (Fig. 22(a)) and increasing (Fig. 22(b)) steering angle operation. The dotted vertical line represents the condition when the online NN has sufficient data, and starts adapting its weights and biases. The adaptive NN requires a certain number of data and learning iterations to provide better performance than the physics-based dynamic model and the offline-trained NN. To circumvent this problem, in Rokonzaman et al. (2021b) a switched MPC algorithm, referred to as SMPC, is designed, where the controller uses the physics-based nonlinear (NL) dynamic vehicle model until the online-trained NN provides better performance. The online NN training process can be conducted on a separate processing core, and therefore it is not part of the real-time MPC optimisation. The proposed strategy is tested on a generic trajectory, and compared with the benchmarking LMPC and NMPC. Fig. 23 shows the prediction error comparison of the NL and NN models, calculated as the difference between the predicted and measured states during the tracking task, while Fig. 24 reports an extract of the trajectories generated by the PT controllers.

5.4.3. MPC implementations involving multiple actuators and functionalities

This section reports a selection of the available results from PT implementations using additional actuators w.r.t. the front steering system, see Table 1, or integrating multiple functionalities.

Hajiloo et al. (2021) presents an integrated steering and differential braking controller, according to the architecture in Fig. 11(b). A relatively long prediction horizon with variable discretisation, together with the adoption of soft constraints for dealing with the lateral dynamics aspects, allows to combine PT, collision avoidance, and vehicle stability control. The performance of the integrated controller is compared with the one of the corresponding controller only actuating the steering system, in an emergency manoeuvre with an obstacle popping up on a straight road with $\mu = 0.5$, from an initial speed of 60 km/h, which pushes the vehicle to limit handling conditions. The differential braking actuation provides quick yaw response, which prevents the vehicle from violating the road boundaries, see Fig. 25. A similar solution, limited to the front steering angle actuation, had already been proposed by Funke et al. (2016), including experimental tests on an automated vehicle, which highlight swift and effective reactions to sudden emergency scenarios, achieved by relaxing the sideslip angle constraints.

To embed forms of obstacle avoidance in the PT layer is not common, despite the excellent response of the solutions in Funke et al. (2016) and Hajiloo et al. (2021). Most of the authors tackle this problem within the path planning algorithms, by defining safe reference trajectories for PT (Hamid et al., 2017b; Ji et al., 2017; H. Wang, Huang, et al., 2019). Y. Chen et al. (2020) generates a collision-free path in a dedicated path

replanning layer, to avoid obstacles and ensure handling stability. Then, an NMPC-based PT algorithm actuates the front steering system, while, in parallel, a transfer function receiving the front steering input generates a reference rear steering angle that stabilises the vehicle and reduces sideslip angle. The performance of this control framework, referred to as C0, has been tested on a HiL platform along a double lane change manoeuvre with $\mu = 0.3$, from a speed of 20 m/s, and is compared with those of: i) the same NMPC with the exclusion of rear-wheel-steering (C1); and ii) an unspecified MPC for concurrent front and rear steering control (C2). In Fig. 26, C1 and C2 cannot follow the reference path, which also results in excessive sideslip angle and stability loss. This is also evident from the RMS and maximum values of the lateral and heading angle errors in Table 14, e.g., C0, C1 and C2 bring maximum deviations from the reference path amounting to 0.40, 2.94, and 1.3 m. A limitation of this comparative analysis is the lack of details, e.g., in terms of controller settings and weight calibration of the C1 and C2 implementations.

Hang et al. (2021) presents a comprehensive multi-actuator PT solution combining four-wheel-steering and DYM control (δ , δ_r , and M_z actuations) into an integrated tube-based MPC. The effectiveness of the proposed controller w.r.t. tube-based MPCs with reduced number of actuators is shown along a double lane change manoeuvre with $\mu = 0.25$ and $v_x = 20$ m/s, see Table 15. The front steering actuation on its own is not sufficient to track the reference path in extreme conditions, which require multi-actuator set-ups. The case with $u = [\delta_f, \delta_r, M_z]$ reduces $|e_{y,max}|$ by 65% and 33%, and $|e_{\psi,max}|$ by 68% and 38%, w.r.t. the cases with $u = [\delta_f, M_z]$ and $u = [\delta_f, \delta_r]$. The rating of the actuation configurations is the same w.r.t. the vehicle stability KPIs.

In accordance with the architecture in Fig. 11(c), Cao et al. (2021) proposes an NMPC PT strategy for front steering and individual wheel torque actuation (differential TV). The architecture includes a feedforward-feedback compensation controller (FCC) to mitigate the risk of secondary collisions by improving post-impact stability, according to:

$$\begin{aligned} \delta &= \delta_{FCC} + \delta_{MPC} \\ T_{ij} &= T_{ij,FCC} + T_{ij,MPC} \end{aligned} \quad (84)$$

where δ_{FCC} and $T_{ij,FCC}$ are the FCC steering angle and wheel torque contributions, while δ_{MPC} and $T_{ij,MPC}$ are those from the MPC-based PT. The performance of the proposed controller (Controller A) is assessed in a rear-end off-centre collision on a straight lane, in which the impact forces are modelled as half-sine pulses, and is compared with that of the post-impact NMPC from D. Yang et al. (2012) (Controller B). Under the assumption that the actuators are not damaged after the first impact, in Fig. 27 Controller A can promptly drive the vehicle back to the original path, whereas Controller B entails an extensive deviation of the car into the adjacent lane, which could result in secondary accidents.

5.4.4. Learning MPC applications for autonomous racing

Autonomous racing has gained considerable interest in recent years. The objective is to implement algorithms capable of minimising lap times through optimal trajectory planning (Alcalà et al., 2019a, 2019b, which are beyond the scope of this survey), or by combining MPC-based PT with learning strategies (Brunner et al., 2017; Kabzan et al., 2019; Rosolia et al., 2017; Rosolia & Borrelli, 2019). In the latter case, the PT component ensures that the vehicle remains within the boundaries of the track, while the steering input is computed to generate a trajectory that progressively reduces lap time through the learning process.

The learning MPC strategy in Rosolia & Borrelli (2019) was experimentally implemented on the 1/10-scale BARC vehicle platform. The algorithm, controlling the steering angle and longitudinal acceleration, is initialised by performing two laps with PT at constant speed, while the following learning iterations use the data from the last two laps. The controller is assessed on the oval- and L-shaped tracks in Fig. 28, which are covered in counter-clockwise direction. The first row of subplots

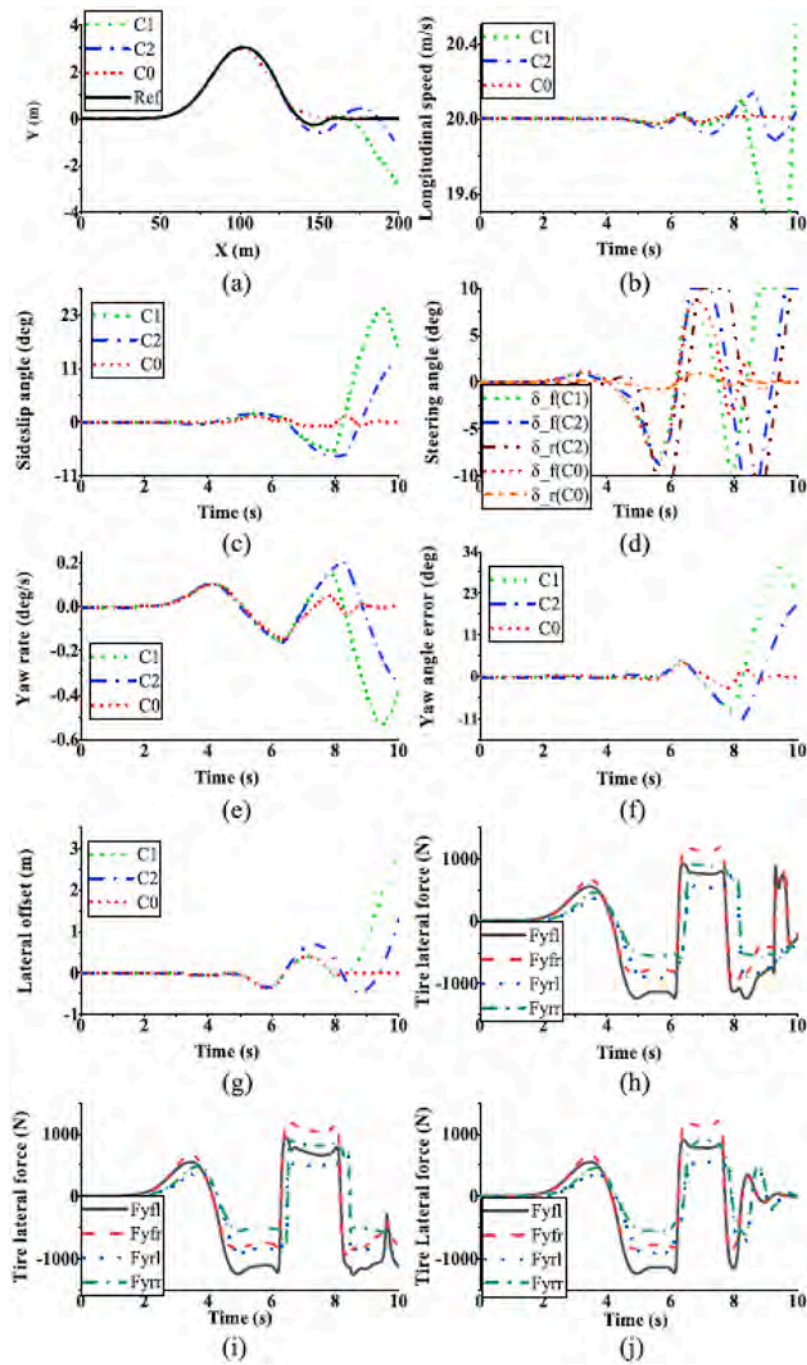


Fig. 26. Experimental results from Y. Chen et al. (2020).

Table 14

Comparison of the tracking performance of different collision avoidance approaches under extreme conditions (from Y. Chen et al., 2020).

Method	RMS_{e_y} [m]	$RMS_{e_{\psi}}$ [deg]	$ e_{y,max} $ [m]	$ e_{\psi,max} $ [deg]
C1	0.6566	9.7925	2.94	30.36
C2	0.3158	5.7243	1.30	20.05
C0	0.1422	1.1059	0.40	3.43

refers to the trajectories used for initialisation and those achieved at laps 7 and 15, while the second row refers to the conditions to which the controller converges. The learning process induces the vehicle to progressively deviate from the original trajectory, by adopting a more aggressive behaviour that ‘cuts’ the corners and increases vehicle speed,

with lateral acceleration peaks of $\sim 1g$ (Fig. 29). The effectiveness of the learning strategy is evident from the lap times in Fig. 30, showing a progressive improvement until reaching a steady-state condition after ~ 30 laps.

The learning MPC in Kabzan et al. (2019) is implemented on the AMZ vehicle used in 2018 Formula Student Driverless event. The experimental test is carried out on the track in Fig. 31, where the top speed is limited to 15 m/s for safety reasons. Also in this case, two laps are used for algorithm initialisation. The speed profile and acceleration comparison in Fig. 32 highlights that the learning-based controller enables more aggressive driving than a nominal MPC, with maximum lateral accelerations that increase from 1.3 g to 2 g, since the learning-based controller is able to make use of the increased tyre grip due to the aerodynamic package. After five laps, the learning approach guarantees

Table 15
Comparison of different PT approaches for collision avoidance under extreme handling conditions (from Hang et al., 2021).

Control input	$ e_{y,max} $ [m]	RMS_{e_y} [m]	$ e_{\psi,max} $ [deg]	$RMS_{e_{\psi}}$ [deg]	$ \Delta\beta_{max} $ [deg]	$RMS_{\Delta\beta}$ [deg]	$ e_{\dot{\psi},max} $ [rad/s]	$RMS_{e_{\dot{\psi}}}$ [rad/s]
δ	2.0483	0.8282	19.2026	6.4633	2.9175	2.0007	0.8906	0.2723
δ, M_z	0.8130	0.3276	3.6344	1.2246	2.6872	0.9682	0.1947	0.0456
δ, δ_r	0.4214	0.1479	2.8577	0.8490	2.6045	0.7943	0.0921	0.0220
δ, δ_r, M_z	0.2817	0.0795	1.7651	0.7275	1.7858	0.4978	0.0589	0.0212

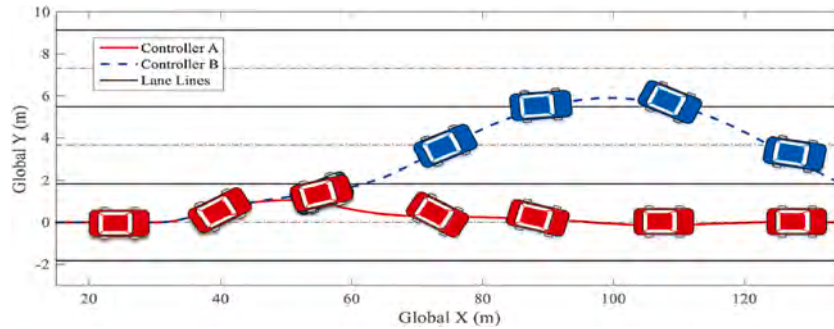


Fig. 27. Path following performance in a post-impact scenario on a road with straight lanes, from an initial speed of 30 m/s (from Cao et al., 2021).

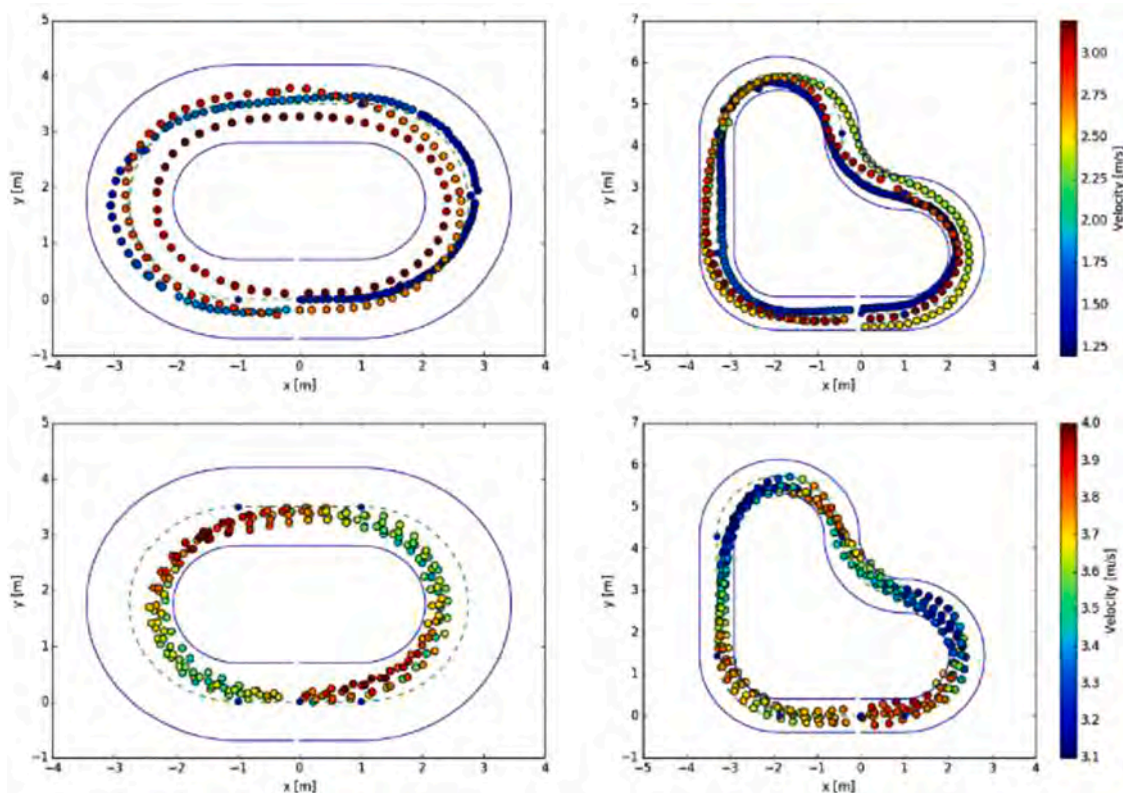


Fig. 28. Experimental results from Rosolia & Borrelli, 2019, along oval (left) and L-shaped (right) tracks. The top subplots show the closed-loop trajectory used to initialise the learning MPC, and the achieved trajectories after a few laps of learning. The bottom subplots report the final trajectories towards which the learning algorithm has converged. The scale of the colour bar changes from the first to the second row, as the vehicle runs at higher speed after the learning process.

a lap time improvement of almost 10% compared to the first lap.

6. Trends and potential future developments

6.1. Trends in recent model predictive path tracking control implementations

Fig. 33 summarises the recent development trends from the literature

on model predictive PT control, through the histograms in subplots (a)–(d), which classify the reviewed implementations based on: (a) the MPC typology, according to the categorisation in Section 2; (b) the prediction model complexity, see Section 3; (c) the inclusion of a single control input, namely the steering control action on its own (these implementations are referred to as ‘Single’ in the subplot), or the consideration of additional control inputs (referred to as ‘Multiple’ in the figure), e.g., associated with the control of the longitudinal vehicle dynamics, or

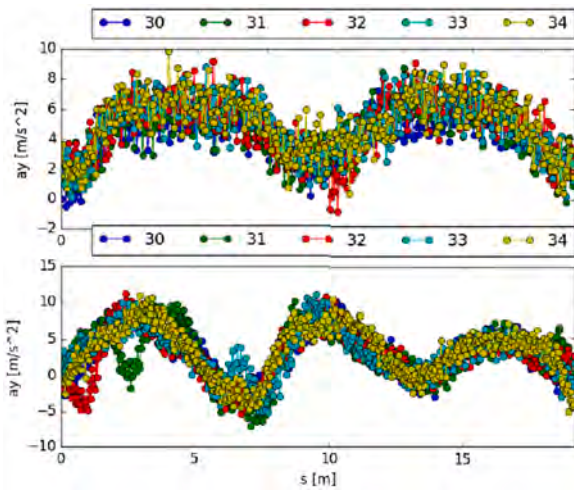


Fig. 29. Experimental lateral acceleration profiles of the vehicle running along the considered oval-shaped (top) and L-shaped (bottom) tracks (from Rosolia & Borrelli, 2019).

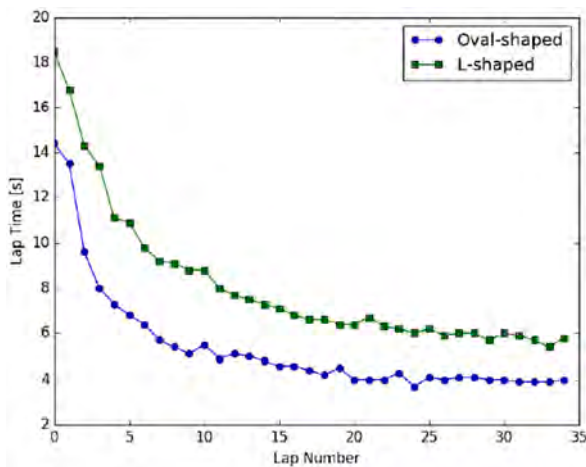


Fig. 30. Lap times as functions of the lap number for the considered learning MPC along oval-shaped and L-shaped tracks (from Rosolia & Borrelli, 2019).

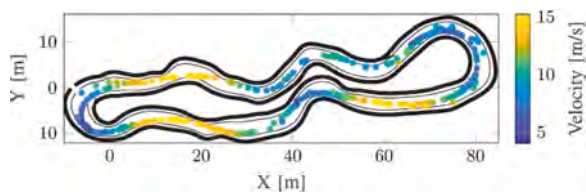


Fig. 31. Experimental results of the learning MPC in Kabzan et al. (2019), where the dots indicate the location of the points that are used in the dictionary of the Gaussian-based learning process after 9 laps. The dictionary points are dense in the most critical portions of the lap, i.e., within the sharp corners. The colour scale shows the velocity at which each point was recorded.

other chassis actuators (the slack variables associated with the soft constraints are not classified as control inputs); and (d) the MPC control assessment method, according to the same classification adopted for Fig. 1(b).

From Fig. 33, the main observations are that:

- The most complex MPC configurations, such as LPV-MPC and NMPC (see subplot (a)), and the implementations embedding the most

accurate prediction models, such as the 3-DoF ST and DT models (see subplot (b)), are becoming more frequent. For example, from 2018 to 2021, the yearly percentage of NMPC PT applications doubles from ~17% to ~34%, while the percentage of MPC formulations using 3-DoF ST models increases from ~15% to ~50%, which makes this prediction model typology the most widely adopted one in the last two years. In parallel, the percentage of implementations based on kinematic models has significantly decreased, and is limited to specific niche applications, as discussed in Section 3. These trends emphasise that the computational effort related to the online solution of the optimal PT control problem does not represent a major issue any longer, thanks to the most recent generation of high-performance solvers and control hardware, capable of significantly reducing controller runtimes. In this respect, the analysis in Quirynen et al. (2018) highlights that the selection of the appropriate solver can reduce the runtime time of an NMPC-based PT formulation by more than 70%.

- Subplot (c) shows an overall similar number of MPC-based PT implementations involving single (i.e., steering actuation only, corresponding to 81 controllers in total) or multiple control actions (corresponding to 71 controllers in total), which confirms that MPC is suitable for dealing with complex control problems involving multiple variables. Despite the histograms do not allow the identification of a clear trend, the 2021 data highlight a majority of multiple-actuator PT solutions, possibly also because MPC-based PT through exclusive control of the front steering angle has already reached a rather high maturity level, as highlighted by this review.
- Although more than 50% of the MPC-based PT implementations have been assessed only through vehicle dynamics simulations, without any verification of their real-time capability, in the last six years 32 controllers have been experimentally tested on real vehicle demonstrators, see subplot (d), which confirms: i) the progressively increasing availability of vehicle prototypes with AD capability; and ii) the relative ease of the online computation of the implicit MPC solution, as already discussed w.r.t. subplots (a) and (b).

6.2. Potential future developments

Based on the analysed PT control literature, possible future development lines and new trends in the subject area of MPC-based PT are:

- MPC implementations that integrate and enhance the PT and stability control functions, where the latter is usually achieved through the individual actuation of the friction brakes. Such integration could push the boundaries of the current generation of vehicle stability controllers. In fact, the paradigm of existing active safety systems is to restrict the response of the vehicle to make it predictable and intuitive for the average human driver. However, this proposition is rather conservative in the context of highly automated vehicles, which could become more effective in avoiding accidents when operating outside the envelope enforced by current chassis control systems. For example, the implementation of PT controllers capable of imitating race driving techniques (e.g., trail braking, pendulum turning, hand brake cornering, and power oversteer) beyond the limit of handling, and thus exploiting the full vehicle potential, would bring enhanced manoeuvrability and enable trajectories that cannot be achieved through the conventional slip angle limitations of stability controllers.

As an example of the potential benefit of this proposed development line, Fig. 34 shows the case of an impending crash between two vehicles at an intersection, where the ego vehicle, i.e., the one with an automated driving system including a PT controller, is the blue one. In Fig. 34(a) the ego vehicle attempts to avoid the collision only by braking. Even with the intervention of the anti-lock braking system (ABS), the collision may not be avoided. In Fig. 34(b), the blue vehicle attempts to avoid the accident by turning and braking. The

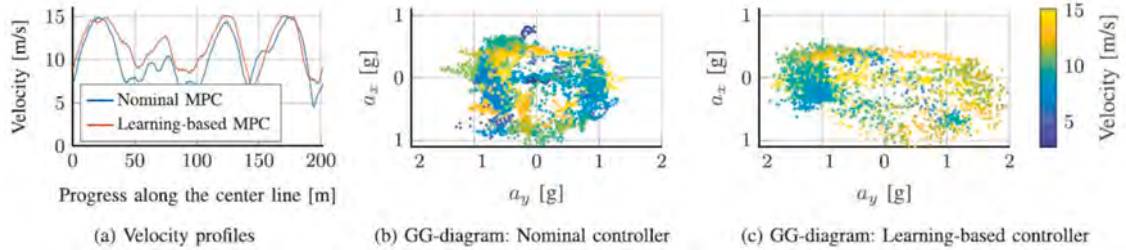


Fig. 32. Experimental performance comparison of a nominal MPC with the proposed learning MPC during the final (ninth) lap of the event (from Kabzan et al., 2019).

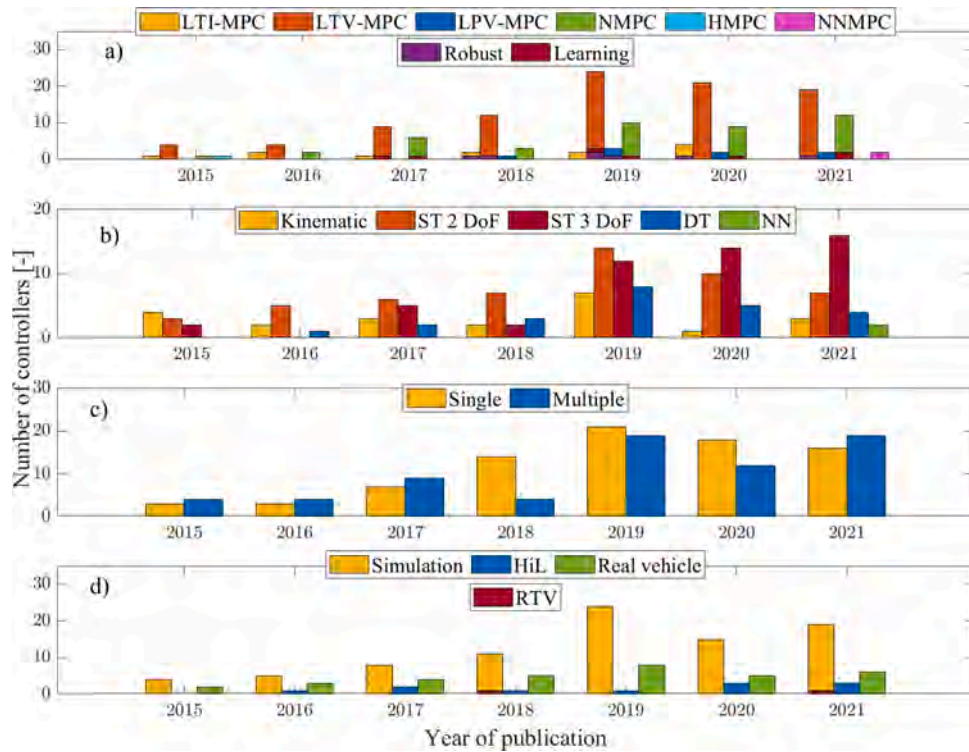


Fig. 33. Overview of the recent MPC-based PT development trends, in terms of number of controller implementations per year according to: (a) model predictive control typology; (b) adopted prediction model; (c) use of front steering actuation as single control input ('Single'), or control of multiple actuators ('Multiple'); (d) environment for generating the results: simulation only, with or without open-loop real-time verification (RTV), or HiL, or real vehicle experiments.

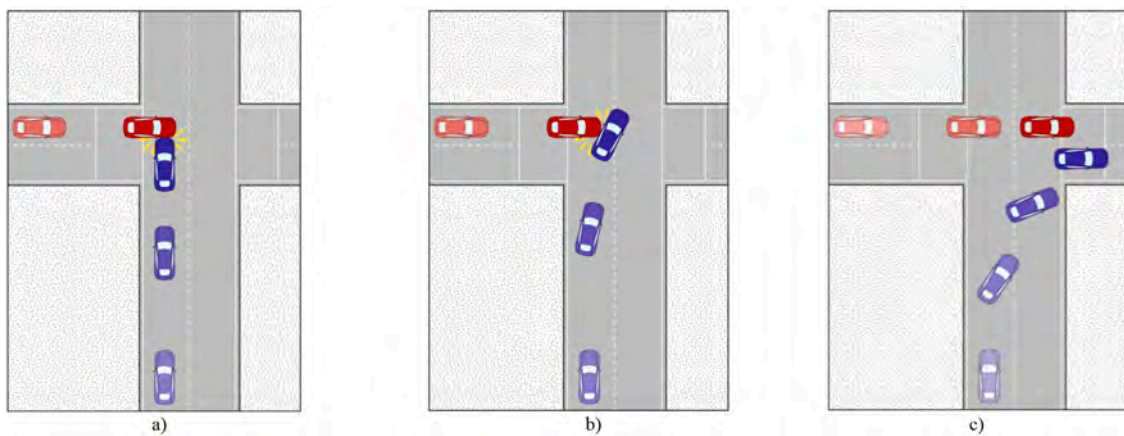


Fig. 34. Emergency manoeuvring of an automated ego vehicle (in blue) in a junction scenario: (a) anti-lock braking system (ABS) actuation without steering intervention; (b) combined steering and braking intervention, while keeping the vehicle within the operational domain of conventional stability controllers; and (c) automated driving control including forms of expert driving actuation, to achieve vehicle operation beyond the limit of handling with lower trajectory radius for accident prevention.

intervention of the conventional stability control system based on the actuation of the friction brakes prevents the vehicle from developing high sideslip angles and excessive levels of understeer or oversteer, but does not guarantee collision avoidance. One way for the ego vehicle to avoid the collision may be to perform a manoeuvre that significantly reduces the turning radius, see Fig. 34(c), by mimicking specialised techniques developed by race drivers. This approach would be transformational considering that industry leaders in AV technology, like Waymo in the US, BMW and Volkswagen in Europe, and Baidu in China, implement or are planning to couple their automated driving algorithms with conventional stability controllers.

Limited literature proposes a few examples of PT controllers (Cai et al., 2020, Goel et al., 2020, and Hou et al., 2021) that, although not being based on MPC, are capable of managing controlled drift conditions, and the authors of this review expect that further studies are going to be published shortly, including MPC formulations. Nevertheless, there is a fundamental knowledge gap in: i) establishing the specific road scenarios under which the extreme handling operation of AVs is beneficial to active safety; and ii) formulating, and efficiently and robustly solving the PT control problem in extreme handling conditions, including the case of vehicle configurations with multiple chassis actuators.

- MPC-based PT implementations embedding consideration of motion sickness and comfort aspects in their formulations. Ride quality refers to the response of the human body to excitations from 1-5 Hz (primary ride) and up to 80 Hz, whereas motion sickness (MS) occurs at lower frequencies. A survey performed in multiple countries indicates that 6-to-10% of adults riding in an ARV often, usually, or always experience some level of MS (Sivac et al., 2015). MS, together with the lack of adequate ride comfort, can severely affect the passengers' ability to use the travel time for work or leisure activities. Hence, MS needs to be effectively addressed for AVs to achieve significant market penetration and deliver their socio-economic and environmental benefits. The likelihood of MS fundamentally depends on the direction, amplitude, frequency and duration of the vibrations experienced by the passengers (Kato and Satoshi, 2006).

The available tools for the assessment of passenger comfort (ride comfort) and MS include standardised metrics and models based on the characteristics of the experienced accelerations. Comfort norms like ISO-2631-1:2017 are derived empirically by fitting mathematical functions to experimental data, and have been used with some success to evaluate MS through the integration of frequency-weighted accelerations. Crucially, the ISO norms lack any connection with the role of the human vestibular system.

Currently, the on-line PT implementations do not include consideration of effective measures to mitigate MS, as highlighted in Elbanhawi et al. (2015). As an exception, Saruchi et al. (2020) proposes a PT architecture based on a fuzzy-PID controller (i.e., without MPC) to compute a corrective steering angle to reduce the lateral acceleration, and consequently decrease the passengers' head roll angle. The controller was evaluated in a motion platform driving simulator and achieved reduced MS. However, the MS metrics were used only in the controller evaluation phase, rather than explicitly in the formulation phase. Ride quality (comfort) considerations in PT control have also been the focus of limited research efforts, mainly in the assessment phase of the algorithms.

Because of the significance of these aspects, some of the next-generation MPC-based PT implementations are likely to formally include simplified yet meaningful MS models within their prediction models, with their outputs to be embedded in the cost function formulations. Similarly, next-generation PT algorithms could formally consider ride comfort aspects, e.g., to penalise small-amplitude steering

input oscillations in critical frequency ranges, or to include innovative forms of compensation of the effect of road irregularities, for example through the integration of steering control, active/semi-active suspension control as well as powertrain control, in the context of PT and chassis control for enhanced user experience.

- MPC-based PT implementations using neural network prediction models, e.g., see the neural network MPCs proposed in Spielberg et al. (2021) and Rokonzaman et al. (2021b), which – without using any form of tyre-road friction information – show better PT performance than more traditional MPCs embedding a physics-based prediction model. In general, such artificial-intelligence-based MPCs could: i) enable consideration of second approximation dynamics (e.g., the longitudinal dynamics associated with road irregularities for a PT controller including powertrain control), while preventing the corresponding increase of computational load; and ii) simplify the state estimator frameworks and sensor fusion systems for PT and vehicle dynamics control at and beyond the limit of handling.
- MPC-based PT formulations for connected vehicles, see the review of use cases in Montanaro et al. (2019). For example, a system merging the connectivity information from the preceding vehicles could inform the ego vehicle about: i) the tyre-road friction coefficient ahead, which is the essential prediction model parameter in emergency conditions; and ii) the possibility of having to unexpectedly carry out an emergency manoeuvre at or beyond the limit of handling, possibly involving the adoption of expert driving techniques, e.g., because of an unforeseen obstacle located at the exit of a tight turn.

7. Conclusions

In the context of automated road vehicles (ARVs), path tracking (PT) control, i.e., the methodology to generate the actuator/s commands to follow the reference path and speed profile, has been the subject of broad research. Among the possible control techniques, model predictive control (MPC) is particularly suitable for PT, thanks to its ability of: i) embedding a prediction of the future behaviour of the system, and thus considering the road curvature preview information, to facilitate smooth and 'natural' steering actuation during normal driving operation, and to enhance active safety in limit handling conditions; ii) managing multiple variables in the framework of complex cost function formulations, which consider and weight different priorities; and iii) formally considering constraints.

This paper reviewed the literature on model predictive PT control for ARVs, and included: a) the presentation of the different MPC-based PT categories, with their control problems, constraints, and weight scheduling formulations; b) the classification of the adopted prediction models, with focus on the involved degrees of freedom and tyre model features; c) the overview of the adopted implementation tools and solvers, key performance indicators, and assessment scenarios; and d) the critical analysis of the main trends from the results.

The review highlighted that:

- MPC is the most widely adopted technique for PT control of ARVs, used in ~50% of the publications on the topic in the 2019-2021 period.
- Prediction models with increasing complexity and accuracy, e.g., including higher number of degrees of freedom and/or tyre nonlinearities, are progressively finding more frequent application, due to the increasing effectiveness and computational power of available solvers and control hardware. For instance, the publications on MPC-based PT using 2-DoF single track prediction models decreased from ~46% in 2018 to ~22% in 2021, compared to an increase in 3 DoF models from ~15% to ~50% in the same years.
- Most of the implementations use control horizons that are significantly shorter than the respective prediction horizons, i.e., >75% of

the considered controllers have control horizons that are less than half than the prediction horizons, to ensure computational efficiency without compromising tracking accuracy and vehicle stability. Typical controller sampling times range from 10 ms to 50 ms. A few controllers are characterised by increasing sampling times along the prediction horizon, to provide accurate prediction in proximity of the current operating condition of the vehicle without compromising their longer-term prediction capability.

- A significant portion of the considered real-time-implementable MPC-based PTs are capable of successfully controlling the vehicle operating at or beyond the limit of handling, in typical emergency scenarios.
- Nearly half of the considered controllers target applications with additional control inputs with respect to the front steering angle, e.g., the PT controller often integrates longitudinal and lateral vehicle control, and/or the vehicle stability control function (yaw rate and sideslip angle stabilisation through direct yaw moment control).

Ongoing and expected future developments include: i) a new generation of PT controllers integrating innovative vehicle dynamics control functions, capable of going beyond the conservative limits of the current generation of vehicle stability controllers for human-driven vehicles; ii) optimal PT control problem formulations embedding vehicle comfort and motion sickness aspects in their prediction models and cost functions; iii) MPC applications using neural networks within the prediction model, for increased adaptability to the actual road and vehicle conditions; and iv) MPC-based PT formulations benefitting from vehicle connectivity features.

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.

Data Availability

Data will be made available on request.

References

- Abbas, M. A., Eklund, J. M., & Milman, R. (2012). Real-time analysis for nonlinear model predictive control of autonomous vehicles. *2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 1–4. <https://doi.org/10.1109/CCECE.2012.6335014>
- Ahn, T., Lee, Y., & Park, K. (2021). Design of integrated autonomous driving control system that incorporates chassis controllers for improving path tracking performance and vehicle stability. *Electronics*, *10*(2), 1–23. <https://doi.org/10.3390/electronics10020144>
- Alcalá, E., Puig, V., & Quevedo, J. (2019a). TS-MPC for Autonomous Vehicles Including a TS-MHE-UIO Estimator. *IEEE Transactions on Vehicular Technology*, *68*(7), 6403–6413. <https://doi.org/10.1109/TVT.2019.2918237>
- Alcalá, E., Puig, V., Quevedo, J., & Rosolia, U. (2019b). Autonomous racing using Linear Parameter Varying-Model Predictive Control (LPV-MPC). *Control Engineering Practice*, *95*. <https://doi.org/10.1016/j.conengprac.2019.104270>
- Alsterda, J.P., Brown, M., & Gerdes, C. J. (2019). Contingency Model Predictive Control for Automated Vehicles. 2019 American Control Conference (ACC). arXiv: 1903.08818v1.
- Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J., & Diehl, M. (2019). CasADi: a software framework for nonlinear optimization and optimal control. *Math. Prog. Comp.*, *11*, 1–36. <https://doi.org/10.1007/s12532-018-0139-4>
- Bai, G., Meng, Y., Liu, L., et al. (2020). Path tracking control of vehicles based on prediction horizon and velocity. *China Mechanical Engineering*, *31*(11), 1277–1284.
- Alsterda, J.P. & Gerdes, J. (2021). Contingency Model Predictive Control for Linear Time-Varying Systems. ArXiv:2102.12045 .
- Batkovic, I., Zanon, M., Ali, M., & Falcone, P. (2019). Real-Time Constrained Trajectory Planning and Vehicle Control for Proactive Autonomous Driving With Road Users. 2019 18th European Control Conference (ECC). <https://doi.org/10.23919/ECC.2019.8796099>.
- Berntorp, K., Quirynen, R., Uno, T., & di Cairano, S. (2019). Trajectory tracking for autonomous vehicles on varying road surfaces by friction-adaptive nonlinear model predictive control. *Vehicle System Dynamics*, *58*(5), 705–725. <https://doi.org/10.1080/00423114.2019.1697456>
- Bo, Y., Guo, H., Shen, C. & Liu, J. (2019). MPC-Based Path Tracking Controller Design for Intelligent Driving Vehicle. 2019 Chinese Control Conference (CCC), 3006–3011. <https://doi.org/10.23919/ChiCC.2019.8866327>.
- Brown, M., Funke, J., Erlien, S., & Gerdes, J. C. (2016). Safe driving envelopes for path tracking in autonomous vehicles. *Control Engineering Practice*, *61*, 307–316. <https://doi.org/10.1016/j.conengprac.2016.04.013>
- Brunner, M., Rosolia, U., Gonzales, J., & Borrelli, F. (2017). Repetitive Learning Model Predictive Control: An Autonomous Racing Example. 2017 IEEE 56th Annual Conference on Decision and Control (CDC). <https://doi.org/10.1016/10.1109/CDC.2017.8264027>.
- Cai, P., Mei, X., Tai, L., Sun, Y., & Liu, M. (2020). High-speed autonomous drifting with deep reinforcement learning. *IEEE Robotics and Automation Letters*, *5*(2), 1247–1254. <https://doi.org/10.1109/lra.2020.2967299>
- Camacho, E. F., & Bordons, C. (1999). *Model Predictive Control*. Springer.
- Cao, J., Song, C., Peng, S., Song, S., Zhang, X., & Xiao, F. (2020). Trajectory Tracking Control Algorithm for Autonomous Vehicle Considering Cornering Characteristics. *IEEE Access*, *1*. <https://doi.org/10.1109/ACCESS.2020.2982963>
- Cao, M., Hu, C., Wang, R., Wang, J., & Chen, N. (2021). Compensatory model predictive control for post-impact trajectory tracking via active front steering and differential torque vectoring. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, *235*(4), 903–919. <https://doi.org/10.1177/0954407020979087>
- Cesari, G., Schildbach, G., Carvalho, A., & Borrelli, F. (2017). Scenario Model Predictive Control for Lane Change Assistance and Autonomous Driving on Highways. *IEEE Intelligent Transportation Systems Magazine*, *9*(3), 23–35. <https://doi.org/10.1109/ITITS.2017.2709782>
- Chen, J., Li, S. E., & Tomizuka, M. (2021). Interpretable End-to-End Urban Autonomous Driving With Latent Deep Reinforcement Learning. *IEEE Transactions on Intelligent Transportation Systems*. <https://doi.org/10.1109/ITITS.2020.3046646>.
- Chen, M., & Ren, Y. (2017). MPC based path tracking control for autonomous vehicle with multi-constraints. 2017 International Conference on Advanced Mechatronic Systems (ICAMechS), 477–482. <https://doi.org/10.1109/ICAMechS.2017.8316521>.
- Chen, S., & Chen, H. (2020). MPC-based path tracking with PID speed control for autonomous vehicles. *IOP Conference Series: Materials Science and Engineering*, *892*(1). <https://doi.org/10.1088/1757-899X/892/1/012034>
- Chen, W., Yan, M., Wang, Q., & Xu, K. (2021). Dynamic path planning and path following control for autonomous vehicle based on the piecewise affine tyre model. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, *235* (2-3), 881–893. <https://doi.org/10.1177/0954407020941729>
- Chen, X., Zhou, B., & Wu, X. (2021). Autonomous vehicle path tracking control considering the stability under lane change. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, *235*(8), 1388–1402. <https://doi.org/10.1177/0959651821991357>
- Chen, Y., Chen, S., Ren, H., Gao, Z., & Liu, Z. (2020). Path Tracking and Handling Stability Control Strategy with Collision Avoidance for the Autonomous Vehicle under Extreme Conditions. *IEEE Transactions on Vehicular Technology*, *69*(12), 14602–14617. <https://doi.org/10.1109/TVT.2020.3031661>
- Chen, Y., Hu, C., & Wang, J. (2019). Human-Centered Trajectory Tracking Control for Autonomous Vehicles with Driver Cut-In Behavior Prediction. *IEEE Transactions on Vehicular Technology*, *68*(9), 8461–8471. <https://doi.org/10.1109/TVT.2019.2927242>
- Chen, Y., & Wang, J. (2019). Trajectory Tracking Control for Autonomous Vehicles in Different Cut-in Scenarios. American Control Conference (ACC). <https://doi.org/10.23919/ACC.2019.8814985>
- Cheng, S., Li, L., Chen, X., Wu, J., & Wang, H. da. (2020). Model-Predictive-Control-Based Path Tracking Controller of Autonomous Vehicle Considering Parametric Uncertainties and Velocity-Varying. *IEEE Transactions on Industrial Electronics*, *68*(9), 8698–8707. <https://doi.org/10.1109/TIE.2020.3009585>
- Choi, W. Y., Kim, D. J., Kang, C. M., Lee, S. H., & Chung, C. C. (2018). Autonomous Vehicle Lateral Maneuvering by Approximate Explicit Predictive Control. Proceedings of the American Control Conference, 2018-June, 4739–4744. 10.23919/ACC.2018.8431889.
- Chowdhri, N., Ferranti, L., Iribarren, F. S., & Shyrokau, B. (2021). Integrated nonlinear model predictive control for automated driving. *Control Engineering Practice*, *106*. <https://doi.org/10.1016/j.conengprac.2020.104654>
- Claussmann, L., Revilloud, M., Gruyer, D., & Glaser, S. (2020). A Review of Motion Planning for Highway Autonomous Driving. *IEEE Transactions on Intelligent Transportation Systems*, *21*(5), 1826–1848. <https://doi.org/10.1109/ITITS.2019.2913998>
- Codrean, A. T., Mariani, T., & Engell, S. (2020). Design and Simulation of a Machine-learning and Model Predictive Control Approach to Autonomous Race Driving for the F1/10 Platform. *IFAC-PapersOnLine*, *53*(2), 6031–6036. <https://doi.org/10.1016/j.ifacol.2020.12.1669>
- Cui, Q., Ding, R., Wu, X., & Zhou, B. (2019). A new strategy for rear-end collision avoidance via autonomous steering and differential braking in highway driving. *Vehicle System Dynamics*, *58*(6), 955–986. <https://doi.org/10.1080/00423114.2019.1602732>
- Cui, Q., Ding, R., Zhou, B. & Wu, X. (2017). Path-tracking of an autonomous vehicle via model predictive control and nonlinear filtering. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, *232*(9), 1237–1252. <https://doi.org/10.1177/0954407017728199>
- Dai, C., Zong, C., & Chen, G. (2020). Path tracking control based on model predictive control with adaptive preview characteristics and speed-assisted constraint. *IEEE Access*, *8*, 184697–184709. <https://doi.org/10.1109/ACCESS.2020.3029635>

- Wang, Z., & Wang, J. (2020). Ultra-local model predictive control: A model-free approach and its application on automated vehicle trajectory tracking. *Control Engineering Practice*, 101. <https://doi.org/10.1016/j.conengprac.2020.104482>
- Wang, Z., Zha, J., & Wang, J. (2019). Flatness-based Model Predictive Control for Autonomous Vehicle Trajectory Tracking. The 2019 IEEE Intelligent Transportation Systems Conference - ITSC. <https://doi.org/10.1109/ITSC.2019.8917260>.
- Wang, Z., Zha, J., & Wang, J. (2020). Autonomous Vehicle Trajectory Following: A Flatness Model Predictive Control Approach with Hardware-in-the-Loop Verification. *IEEE Transactions on Intelligent Transportation Systems*, 22(9), 5613–5623. <https://doi.org/10.1109/TITS.2020.2987987>
- Watzgen, D., & Horn, M. (2016). Introduction to automated driving. In *Automated Driving: Safer and More Efficient Future Driving*. https://doi.org/10.1007/978-3-319-31895-0_1.
- World Health Organisation. (2018). Global Status Report on Road Safety 2018. <https://www.who.int/publications/i/item/9789241565684>.
- Wu, H., Si, Z., & Li, Z. (2020). Trajectory Tracking Control for Four-Wheel Independent Drive Intelligent Vehicle Based on Model Predictive Control. *IEEE Access*, 8, 73071–73081. <https://doi.org/10.1109/ACCESS.2020.2987812>
- Wu, Y., Wang, L., Zhang, J., & Li, F. (2019). Path Following Control of Autonomous Ground Vehicle Based on Nonsingular Terminal Sliding Mode and Active Disturbance Rejection Control. *IEEE Transactions on Vehicular Technology*, 68(7), 6379–6390. <https://doi.org/10.1109/TVT.2019.2916982>
- Wurts, J., Stein, J. L., & Ersal, T. (2021). Collision Imminent Steering at High Speeds on Curved Roads Using One-Level Nonlinear Model Predictive Control. *IEEE Access*, 9, 39292–39302. <https://doi.org/10.1109/ACCESS.2021.3063795>
- Xiang, C., Peng, H., Wang, W., Li, L., An, Q., & Cheng, S. (2020). Path tracking coordinated control strategy for autonomous four in-wheel-motor independent-drive vehicles with consideration of lateral stability. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 235(4), 1023–1036. <https://doi.org/10.1177/0954407020946884>
- Xie, J., Xu, X., Wang, F., Tang, Z., & Chen, L. (2021). Coordinated control based path following of distributed drive autonomous electric vehicles with yaw-moment control. *Control Engineering Practice*, 106. <https://doi.org/10.1016/j.conengprac.2020.104659>
- Xu, S., & Peng, H. (2020). Design, Analysis, and Experiments of Preview Path Tracking Control for Autonomous Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 21(1), 48–58. <https://doi.org/10.1109/TITS.2019.2892926>
- Xu, Y., Chen, B. Y., Shan, X., Jia, W. H., Lu, Z. F., & Xu, G. (2017). Model predictive control for lane keeping system in autonomous vehicle. *2017 7th International Conference on Power Electronics Systems and Applications - Smart Mobility* (pp. 1–5). Power Transfer & Security (PESA). <https://doi.org/10.1109/PESA.2017.8277758>
- Xu, Y., Tang, W., Chen, B., Qiu, L., & Yang, R. (2021). A model predictive control with preview-follower theory algorithm for trajectory tracking control in autonomous vehicles. *Symmetry*, 13(3), 1–16. <https://doi.org/10.3390/sym13030381>
- Yakub, F., Abu, A., Sarip, S., & Mori, Y. (2016). Study of model predictive control for path-following autonomous ground vehicle control under crosswind effect. *Journal of Control Science and Engineering*, 2016. <https://doi.org/10.1155/2016/6752671>
- Yakub, F., & Mori, Y. (2015). Comparative study of autonomous path-following vehicle control via model predictive control and linear quadratic control. In *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 229(12), 1695–1714. <https://doi.org/10.1177/0954407014566031>
- Yang, D., Gordon, T. J., Jacobson, B., & Jonasson, M. (2012). A nonlinear post impact path controller based on optimised brake sequences. *Vehicle System Dynamics*, 50(1), 131–149. <https://doi.org/10.1080/00423114.2012.660490>. SUPPL.
- Yang, K., Tang, X., Qin, Y., Huang, Y., Wang, H., & Pu, H. (2021). Comparative Study of Trajectory Tracking Control for Automated Vehicles via Model Predictive Control and Robust H-infinity State Feedback Control. *Chinese Journal of Mechanical Engineering (English Edition)*, 34(1). <https://doi.org/10.1186/s10033-021-00590-3>
- Yao, Q., & Tian, Y. (2019). A model predictive controller with longitudinal speed compensation for autonomous vehicle path tracking. *Applied Sciences (Switzerland)*, 9(22). <https://doi.org/10.3390/app9224739>
- Yao, Q., Tian, Y., Wang, Q., & Wang, S. (2020). Control Strategies on Path Tracking for Autonomous Vehicle: State of the Art and Future Challenges. *IEEE Access*, 8, 161211–161222. <https://doi.org/10.1109/ACCESS.2020.3020075>
- Ye, H., Jiang, H., Ma, S., Tang, B., & Wahab, L. (2019). Linear model predictive control of automatic parking path tracking with soft constraints. *International Journal of Advanced Robotic Systems*, 16(3). <https://doi.org/10.1177/1729881419852201>
- Yu, C., Zheng, Y., Shyrokau, B., & Ivanov, V. (2021). MPC-based Path Following Design for Automated Vehicles with Rear Wheel Steering. In *2021 IEEE International Conference on Mechatronics (ICM)*, 1–6. <https://doi.org/10.1109/ICM46511.2021.9385606>.
- Yu, H., Duan, J., Taheri, S., Cheng, H., & Qi, Z. (2015). A model predictive control approach combined unscented Kalman filter vehicle state estimation in intelligent vehicle trajectory tracking. *Advances in Mechanical Engineering*, 7(5), 1–14. <https://doi.org/10.1177/1687814015578361>
- Yu, J., Guo, X., Pei, X., Chen, Z., Zhu, M., & Gong, B. (2019). Robust model predictive control for path tracking of autonomous vehicle. *SAE Technical Papers*. <https://doi.org/10.4271/2019-01-0693>.
- Yu, J., Pei, X., Guo, X., Lin, J. G., & Zhu, M. (2020). Path tracking framework synthesizing robust model predictive control and stability control for autonomous vehicle. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 234(9), 2330–2341. <https://doi.org/10.1177/0954407020914666>
- Yu, S., Hirche, M., Huang, Y., Chen, H., & Allgöwer, F. (2021). Model predictive control for autonomous ground vehicles: a review. *Autonomous Intelligent Systems*, 1(1). <https://doi.org/10.1007/s43684-021-00005-z>
- Yu, Y., Li, Y., Liang, Y., Zheng, L., & Ren, Y. (2021). Decoupling motion tracking control for 4WD autonomous vehicles based on the path correction. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 236(1), 99–108. <https://doi.org/10.1177/09544070211015928>
- Yuan, H., Sun, X., & Gordon, T. (2018). Unified decision-making and control for highway collision avoidance using active front steer and individual wheel torque control. *Vehicle System Dynamics*, 57(8), 1188–1205. <https://doi.org/10.1080/00423114.2018.1535125>
- Yuan, K., Shu, H., Huang, Y., Zhang, Y., Khajepour, A., & Zhang, L. (2018). Mixed local motion planning and tracking control framework for autonomous vehicles based on model predictive control. *IET Intelligent Transport Systems*, 13(6), 950–959. <https://doi.org/10.1049/iet-its.2018.5387>
- Zhang, B., Zong, C., Chen, G., & Li, G. (2019a). An adaptive-prediction-horizon model prediction control for path tracking in a four-wheel independent control electric vehicle. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 233(12), 3246–3262. <https://doi.org/10.1177/0954407018821527>
- Zhang, B., Zong, C., Chen, G., & Zhang, B. (2019b). Electrical Vehicle Path Tracking Based Model Predictive Control with a Laguerre Function and Exponential Weight. *IEEE Access*, 7, 17082–17097. <https://doi.org/10.1109/ACCESS.2019.2892746>
- Zhang, C., Chu, D., Liu, S., Deng, Z., Wu, C., & Su, X. (2019). Trajectory Planning and Tracking for Autonomous Vehicle Based on State Lattice and Model Predictive Control. *IEEE Intelligent Transportation Systems Magazine*, 11(2), 29–40. <https://doi.org/10.1109/TITS.2019.2903536>
- Zhang, K., Sprinkle, J., & Sanfelice, R. G. (2015). Computationally aware control of autonomous vehicles: a hybrid model predictive control approach. *Autonomous Robots*, 39(4), 503–517. <https://doi.org/10.1007/s10514-015-9469-5>
- Zhang, V., Thornton, S. M., & Gerdes, J. C. (2018). Tyre Modeling to Enable Model Predictive Control of Automated Vehicles From Standstill to the Limits of Handling, 14th International Symposium on Advanced Vehicle Control (AVEC 2018).
- Zhang, W., Wang, Z., Drugge, L., & Nybacka, M. (2020). Evaluating Model Predictive Path following and Yaw Stability Controllers for Over-Actuated Autonomous Electric Vehicles. *IEEE Transactions on Vehicular Technology*, 69(11), 12807–12821. <https://doi.org/10.1109/TVT.2020.3030863>
- Zhu, Q., Huang, Z., Liu, D., and Dai, B. (2016). An adaptive path tracking method for autonomous land vehicle based on neural dynamic programming. In *2016 IEEE International Conference on Mechatronics and Automation*, 1429–1434. <https://doi.org/10.1109/ICMA.2016.7558773>.
- Zou, K., Cai, Y., Chen, L., & Sun, X. (2021). Event-triggered nonlinear model predictive control for trajectory tracking of unmanned vehicles. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*. <https://doi.org/10.1177/0954407021992165>.
- Zou, Y., Guo, N., & Zhang, X. (2019). An integrated control strategy of path following and lateral motion stabilization for autonomous distributed drive electric vehicles. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 235(4), 1164–1179. <https://doi.org/10.1177/0954407019884168>