Received 10 January 2024, accepted 31 January 2024, date of publication 5 February 2024, date of current version 16 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3362236

RESEARCH ARTICLE

Neural-Network Based MPC for Enhanced Lateral Stability in Electric Vehicles

AHMED HASSAN, SARA RUIZ-MORENO, JOSE RAMON D. FREJO, JOSE M. MAESTRE, AND EDUARDO FERNÁNDEZ CAMACHO, (Life Fellow, IEEE)

Department of System Engineering and Automatic Control, University of Seville, 41092 Seville, Spain

Corresponding author: Ahmed Hassan (ahmhasahm@alum.us.es)

This work was supported in part by the Spanish MCIN/AEI C3PO-R2D2 Project under Grant PID2020-119476RB-I00; and in part by the Egyptian Government, the Spanish Ministry of Science, Innovation, and Universities under Grant FPU20/01958 and Grant PID2022-142069OB-I00/AEI/10.13039/501100011033/FEDER, UE.

ABSTRACT Distributed electric drive vehicles offer maneuver-ability but face stability challenges under different driving conditions. Model Predictive Control (MPC) algorithms can improve lateral stability, but their high computational demands hinder real-time implementation. To address this, the proposed strategy combines Nonlinear Autoregressive Exogenous (NARX) neural networks with MPC in two ways, namely, Nonlinear Prediction-Nonlinear Optimization (NMPC-NO) and Nonlinear Prediction-Linearization (MPC-NPL). While NMPC-NO involves online nonlinear optimization, MPC-NPL uses local linearization, reducing both the computational load significantly to about 40% of the computation time of MPC and 0.05% of that of nonlinear model predictive control (NMPC). The neural networks are trained and validated on 20 different datasets, with alternative training methods investigated. MATLAB/Simulink simulations under various standardized tests demonstrate the effectiveness of the proposed techniques, highlighting improved handling performance, reduced computation time, and real-time deployment capabilities.

INDEX TERMS Artificial intelligence (AI), nonlinear model predictive control (NMPC), model predictive control (MPC), machine learning (ML), nonlinear prediction-nonlinear optimization (NMPC-NO), nonlinear prediction-linearization (MPC-NPL).

I. INTRODUCTION

Recent research has focused on the four-wheel independent-motor-drive electric vehicle (4WIMDEV) due to its environmental benefits and precise control of vehicle movements through independent torque distribution on each wheel [1], [2] (see Fig. 1). The research has particularly emphasized vehicle stability control during emergency situations, aiming to enhance handling and safety in dynamic driving conditions including various strategies. Electronic Stability Control (ESC) is highlighted for maintaining the intended path through selective brake application, but it may have limitations on low-friction surfaces [3]. The Traction Control System (TCS) aims to prevent wheel spin during acceleration, enhancing traction and stability, though it may affect engine power [4]. The Anti-lock Braking System (ABS) prevents wheel lock during hard braking, maintaining steering control but potentially increasing stopping distances on rough surfaces [5]. Furthermore, negotiating corner conditions may result in oversteering

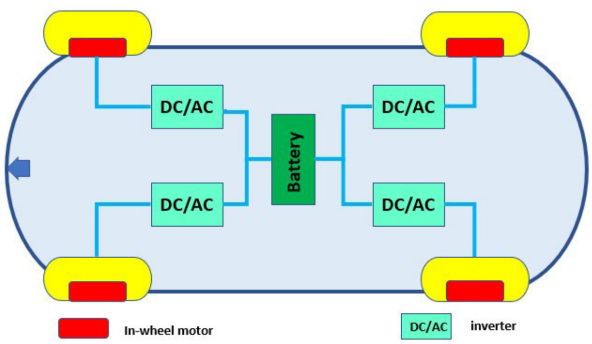


FIGURE 1. The 4 in-wheel motor.

The associate editor coordinating the review of this manuscript and approving it for publication was JJun Cheng.

and understeering challenges, prompting researchers to investigate and tackle these phenomena.

Recent research underscores Direct Yaw Moment Control (DYC) as effective for improving handling stability [6], [7], [8]. DYC comprises two control levels, with the upper controller determining the desired yaw moment, and the lower controller allocating it to the four wheels using Torque Vector Control (TVC) [9], [10]. Despite DYC's effectiveness, its nonlinear and uncertain characteristics pose challenges in designing the upper controller. Traditional PID controllers, commonly used for yaw moment generation, face issues in tuning parameters and handling nonlinearity [11]. Alternative methods like Linear Quadratic Regulator (LQR) as in [12] and fuzzy control as in [13] have been explored, but may have complex control rules. Model Predictive Control (MPC) is gaining attention for its potential in predicting states and handling diverse control objectives [14], [15], [16], [17], while Nonlinear Model Predictive Control (NMPC) has shown effectiveness in managing complex systems, though its implementation demands careful consideration of computing resources [18], [19].

For example, solving NMPC takes about 15 s for the problem considered, raising concerns about meeting the recommended 30 milliseconds sample time for vehicle stability [20]. Moreover, in other related works, a 10 milliseconds sample time has been used [21], [22], stressing computational limitations for real-time implementation.

The primary objective of this article is to reduce the computational burden of predictive controllers considering the limits imposed by real-time applications. Additionally, we deal with the challenges posed by vehicle nonlinearity and aim to improve handling performance by using neural networks for nonlinear prediction, effectively capturing system dynamics while reducing computation time.

Various strategies have been employed to enhance MPC's computational efficiency. Specialized fast MPC algorithms like explicit methods offer real-time capabilities with custom implementations [23]. However, these methods may demand customized implementations and offer less flexibility than general-purpose solvers. Suboptimal (MPC) achieves a compromise between computational efficiency and control performance, rendering it appropriate for dynamic systems and applications characterized by the model uncertainties [24]. Nevertheless, its effectiveness relies on appropriate initialization and may result in suboptimal control performance, thereby diminishing its suitability for high-precision control demands. Mixed-integer Linear Programming (MILP), as exemplified by [25], provides a robust framework for handling constraints and optimizing MPC but may become computationally expensive for large-scale systems. Horizon reduction, as introduced by [26], shortens prediction horizons for faster computations, albeit at the expense of a trade-off between computation time and long-term performance. Sparse optimization techniques aim to minimize active constraints and variables, significantly reducing computation time as demonstrated by [27], though they require careful selection and tuning of sparsity-inducing penalties. Warm start and online methods, as utilized by [28], reduce computation by reusing previous solutions, but are more suitable for slowly varying dynamics. The authors in [29] discuss the utilization of Graphics Processing Units (GPUs) for MPC computations, offering a significant speedup but demanding access to GPU hardware and complex implementation. Active-set methods, as explored by [30], focus on efficiently solving Quadratic Programming (QP) problems through active constraint selection and updating, thereby reducing computation time, though they may require careful tuning and implementation. In [31] the benefits of parallelism in solving large-scale MPC problems, although complex synchronization and load balancing may be needed for distributed computing. Event-triggered (MPC) offers advantages such as reduced computational load, enhanced energy efficiency, lower communication overhead, and improved robustness to disturbances [32], [33], [34]. However, it involves complex design requirements and potential trade-offs between computation time reduction and control performance as in [35], [36], and [37].

A significant contributor to the high computational power requirements of MPC is the inherent complexity of the model, often characterized by nonlinearity and a high order. Various approaches have been devised to address this challenge, including the development of linearized building models [38], online-linearization techniques [39], and developing reduced-order models of system dynamics, as seen in [40] and [41], which enable faster optimization and control computations. Leveraging artificial intelligence to emulate MPC and reduce the computational burden is another effective approach, as demonstrated by [42], which utilized artificial neural networks to dramatically shorten the computation time of MPC in solar parabolic-trough plants, offering an efficient solution for computational optimization. Reduced online computational requirements are one benefit of integrating Machine Learning (ML) techniques, particularly artificial neural networks (ANNs) with MPC. In particular, machine learning (ML) techniques improve MPC computational costs by enabling data-driven modeling [43], [44]. The integration of ML and MPC has experienced consistent growth in recent years, resulting in various categories of applications [45], [46], [47], e.g., offline modeling utilizes measurement data to create ML models for MPC [2], [48], [49]; online learning adjusts MPC model coefficients in real-time [26], [50], [51]; ML in imitation of MPC replicates MPC behavior in real-time, with successful applications in various industries [52], [53], [54] and improvements of computational efficiency [52], [55], [56]; ML in control structure of MPC involves ML as an add-on or embedded controller [57], [58], [59]; finally, MPC can also work as a safe learning controller in learning algorithms to address control constraints [60], [61]. Complex nonlinear interactions that may be difficult

FIGURE 2. **The fox vehicle, an experimental test car at university of seville.**

for conventional mathematical and statistical models to capture, particularly in complex systems, can be captured by ML-based models such as nonlinear autoregressive models with exogenous inputs (NARX) [62], [63], feed-forward neural networks (FNNs) [64], deep neural networks (DNNs) [65], and recurrent neural networks (RNNs) [66], can be used as process models, have the potential to effectively represent complicated physical systems, have demonstrated the ability to successfully simulate dynamic processes inside the MPC framework, giving precise approximations and quicker convergence in MPC. In this regard, it would be beneficial to place a stronger emphasis on MPC controllers utilizing NARX models, as this approach aligns with the methodology outlined in our paper.

Specifically, this study proposes the utilization of Nonlinear Autoregressive Exogenous (NARX) Artificial Neural Networks (ANNs). The NARX model is particularly suitable for handling nonlinearities and uncertainties, making it a popular choice in time-series forecasting, control systems, and nonlinear system identification [67]. Two approaches are employed here combining NARX neural networks with MPC [68], [69]:

1) The first approach, Nonlinear Prediction and Linearization (MPC-NPL), involves online linearization of the neural network model and solving an online quadratic optimization problem.
2) The second approach, Nonlinear Prediction and Nonlinear Optimization (NMPC-NO), utilizes neural network models directly for nonlinear prediction without simplifications. It derives the optimal control policy by solving a nonlinear optimization problem.

By learning the system dynamics, it is possible to accelerate the computation of the control action while preserving the capacity to adjust the behavior of the controller by modifying its tuning parameters such as the prediction horizon and the cost's weighting matrices. Moreover, the approach followed also allows considering explicitly the satisfaction of the problem constraints along the prediction horizon. The proposed combination of MPC and NN controllers is assessed through the utilization of a simulated vehicle featuring four electric wheel motors, this model is based on the physical prototype shown in Fig. 2. It functions as a testing platform for control experimentation.

Our paper makes two main contributions:

We propose a novel approach that integrates Nonlinear Autoregressive with Exogenous Inputs (NARX) neural networks and Model Predictive Control (MPC) algorithms to enhance the lateral stability of electric vehicles. This integration offers several notable advantages. Firstly, by combining NARX neural networks and MPC algorithms, our approach allows for more efficient control computations, reducing the computational burden significantly. This, in turn, leads to improved real-time performance, ensuring that the controller can effectively respond to dynamic driving conditions. Additionally, the integration of NARX networks facilitates better tracking accuracy, as they excel at capturing complex, non-linear relationships in the vehicle dynamics, resulting in enhanced control performance. As a result, our approach not only optimizes computational efficiency but also bolsters the overall stability and handling performance of electric vehicles, making it a promising solution for real-world deployment.

The rest of this article is structured as follows. Section II illustrates both the vehicle's linear and non-linear models. Section III explains the effective control methods that have been applied and introduces the utilization of the torque vector control (TVC) method as a low-level control strategy. Section IV introduces ANNs, the training process, and their integration with control systems. Section V provides a numerical assessment of the proposed strategies. Finally, conclusions are summarized in Section VI.

II. PROBLEM FORMULATION AND SYSTEM MODELING

In this study, we utilize both linear and nonlinear models. The nonlinear model is a 14-degree-of-freedom representation built upon vehicle dynamics, encompassing components such as the vehicle itself, tires, and in-wheel motors. The linear model, on the other hand, is employed specifically for developing control strategies for electric vehicles equipped with in-wheel motors. All the symbols in the modeling process and their related physical meanings are listed in Table 1.

A. LINEAR VEHICLE MODEL

We initially introduce a two-degree-of-freedom vehicle model focusing on yaw and lateral motion. To facilitate control analysis, we employ the kinematic bicycle model [70], [71] to represent the lateral dynamics. This model operates on the assumption that a moment M_z can be produced through the difference in longitudinal tire forces between the wheels (as depicted in Fig. 3). The equations governing the vehicle's lateral dynamics are

$$mv_x(\dot{\beta} + r) = F_{yr} + F_{yf}, \tag{1}$$

$$I_z\dot{r} = l_f F_{yf} - l_r F_{yr} + M_z. \tag{2}$$

Under turning conditions, the dynamic equations of the bicycle have been linearized to focus exclusively on small values of lateral drift and sideslip angles, simplifying

TABLE 1. Vehicle parameters and model variables.

Symbol	Description	Value	[Units]
m	Vehicle mass	400.238	[kg]
I_z	Yaw moment of inertia	1047.52	[kg m2]
l_f	Distance from CG to front axle	1.3004	[m]
l_r	Distance from CG to rear axle	1.2204	[m]
L	Wheelbase	2.5208	[m]
C_f	Cornering stiffness of front tire	47000	[N/rad]
C_r	Cornering stiffness of rear tire	53000	[N/rad]
μ	Tire-road friction coefficient	[-]	[-]
β	Sideslip angle	[-]	[rad]
δ_f	Front steer angle	[-]	[rad]
α_f	Front side slip angle	[-]	[rad]
α_r	Rear side slip angle	[-]	[rad]
F_{yr}	Rear axle lateral tire force	[-]	[N]
F_{yf}	Front axle lateral tire force	[-]	[N]
F_g	The force of gravity	[-]	[N]
F_z	The vertical load	[-]	[N]
F_{roll}	The rolling resistance	[-]	[N]
F_w	The horizontal road	[-]	[N]
F_{air}	The air resistance	[-]	[N]
v_x	Vehicle longitudinal speed	[-]	[km/h]
r	Vehicle yaw rate	[-]	[rad/s]
τ_{lag}	Relaxation time	[-]	[s]
a_x	Horizontal acceleration	[-]	[r/s^2]
a_y	Vertical acceleration	[-]	[r/s^2]
V	Vehicle velocity	[-]	[km/h]
t_f	Width of the front axle	[-]	[m]
t_r	Width of the rear axle	[-]	[m]
c_1, c_2, c_3	Wheel model constants	[-]	[-]

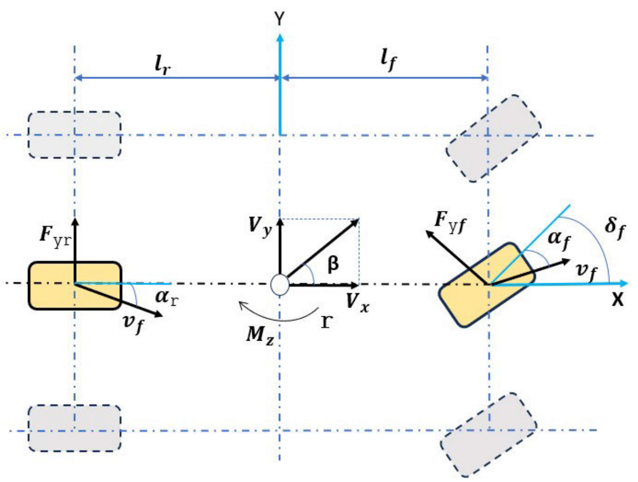


FIGURE 3. Vehicle bicycle model.

expressions to approximate $\sin(\alpha) \approx \alpha$, $\cos(\alpha) \approx 1$. Following this lineari-zation of equations (1) and (2), along with the incorporation of a dynamic tire model, we derive the subsequent state-space model [72]:

$$\dot{x}_l = Ax_l + B_\delta \delta_f + B_M M_z, \tag{3}$$

where (4)–(6), as shown at the bottom of the next page.

B. NONLINEAR VEHICLE MODEL

A nonlinear model has also been developed in Sim-Mechanics to faithfully replicate the real vehicle's path [21] (see Fig. 4). The FOX vehicle's SimMechanics block

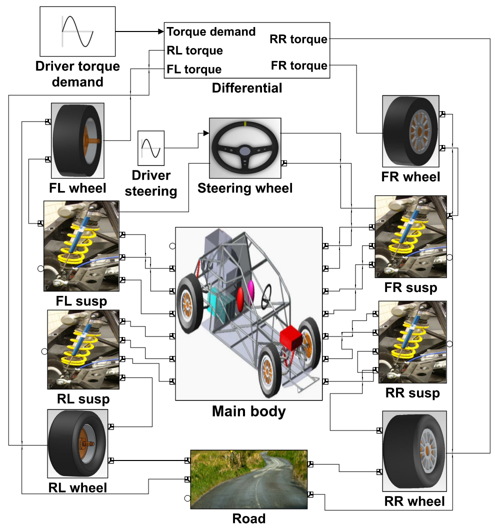


FIGURE 4. The vehicle's simmechanics model.

diagram incorporates 35 bodies and 38 joints from Solid-works CAD and uses 14-degree-of-freedom (longitudinal, lateral, vertical, roll, pitch, bounce, rotationas, and vertical oscillations of 4 wheels). We also assume rigid vehicle parts, tire-road force at the wheel's lowest point, flat surface, and account for longitudinal air friction. This non-linear model has been employed both as an internal model within the NMPC controller and as a digital replica of the actual vehicle. Similarly, Fig. 5a shows the external forces applied to the vehicle, whereas Fig. 5b illustrates the tire-road interaction, determining forces generated at the contact surface. The wheel movement is determined using longitudinal S_L and lateral S_s coefficients

$$S_L = \frac{v_r \sin\alpha}{v_w}, \quad S_S = \frac{v_r \cos\alpha - v_w}{v_r \cos\alpha}$$

where v_w is the speed of the wheel, v_r is a virtual speed whose direction is coincidental to the x_i axis, and α is the slip angle between v_r and v_w. The longitudinal F_L and lateral F_S forces are calculated as

$$F_L = \mu_L F_Z, \quad F_S = \mu_S F_Z \tag{7}$$

with μ_L and μ_S being the lateral and side friction coefficients, where

$$\mu_L = \mu_{Res}\frac{S_L}{S_{Res}}, \mu_S = \mu_{Res}\frac{S_S}{S_{Res}}, S_{Res} = \sqrt{(S_L)^2 + (S_S)^2},$$
$$\mu_{Res} = c_1 * (1 - e^{c_2 S_{Res}}) - c_3 S_{Res}$$

This Simmechanics model is very versatile since it allows us to choose several types of roads according to their adhesion (dry asphalt, rain, snow, etc.). Finally, a discretization time of 0.01 seconds is employed following [21] and [22].

C. CONSTRAINTS

The brushless motors in the system have 7 KW of power and are supplied with alternating current through a converter,

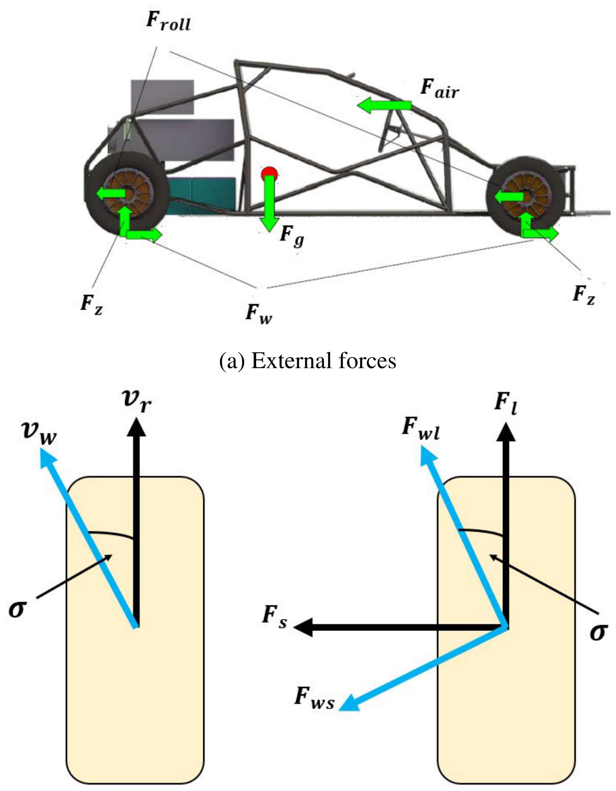


FIGURE 5. **External and tire forces on the vehicle.**

which is more efficient for prolonged use. These motors are chosen for their performance benefits, including reduced friction heat loss, and noise. They are directly connected to the wheels for maximum torque delivery efficiency. However, the manufacturer has set a maximum torque limit of $T_{max} = 80Nm$, which serves as the ultimate constraint on torque application. The actuator's primary function is to ensure control and safety by imposing constraints on the permissible additional yaw moment. These constraints may include limiting torque output, ensuring a specified response speed, or respecting the actuator's physical limitations. In essence, the actuator's role is to guarantee that the system operates within these defined limits while making necessary adjustments to the yaw moment. The following constraints need to be satisfied regardless of the model employed:

$$\begin{aligned} |\Delta M_z| &\leq 100 \\ |M_z| &\leq 1000 \\ |T_{max}| &\leq 80 \end{aligned} \tag{8}$$

III. YAW STABILITY CONTROL

Actual road vehicle dynamics exhibit substantial non-linearity and are susceptible to model uncertainties and disturbances, particularly when encountering demanding driving scenarios like sharp cornering maneuvers. These dynamics can be influenced by parameter fluctuations (such as changes in road surface adhesion coefficients, tire cornering stiffness, vehicle mass, vehicle speed, and moment of inertia [73]) as well as external disturbances (for instance, lateral crosswinds that can affect the ability to maintain the desired yaw rate and sideslip response [74]). Consequently, it is imperative to employ suitable control strategies and algorithms to effectively manage these challenges.

In particular, when aiming to achieve yaw stability control in an electric vehicle equipped with in-wheel motors, several objectives must be taken into consideration. These include ensuring excellent handling and stability, preventing wheel slip on surfaces with low friction coefficients, and delivering a smooth driving experience for the driver's comfort. The proposed controller for yaw stability adopts a hierarchical structure, as illustrated in Fig. 6. In this configuration, the linear/nonlinear MPC serves as the upper-level controller. It takes desired sideslip and yaw rate as inputs and calculates the necessary moment M_z. Meanwhile, the TVC functions as the lower-level controller, responsible for distributing torque among the four motors to generate the desired angular momentum.

A. SET POINT

The generation of references for both sideslip angle and yaw rate is determined as follows. In order to enhance stability and

$$x_l = \begin{bmatrix} \beta & \dot{\beta} & r & \dot{r} \end{bmatrix}^T, \tag{4}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\dfrac{C_f + C_r}{\tau_{lag} m v_x} & -\dfrac{1}{\tau_{lag}} & \left(\dfrac{C_r l_r - C_f l_f}{\tau_{lag} m v_x^2} - \dfrac{1}{\tau_{lag}}\right) & -1 \\ 0 & 0 & 0 & 1 \\ \dfrac{C_r l_r - C_f l_f}{\tau_{lag} I_z} & 0 & -\dfrac{C_f l_f^2 + C_r l_r^2}{\tau_{lag} I_z v_x} & -\dfrac{1}{\tau_{lag}} \end{bmatrix}, \tag{5}$$

$$B_\delta = \begin{bmatrix} 0 \\ \dfrac{C_f}{\tau_{lag} m v_x} \\ 0 \\ \dfrac{C_f l_f}{\tau_{lag} I_z} \end{bmatrix}, \quad B_M = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dfrac{1}{\tau_{lag} I_z} \end{bmatrix}. \tag{6}$$

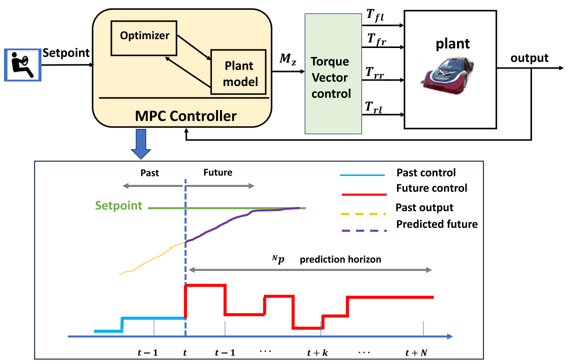


FIGURE 6. **MPC and yaw stability control structure.**

enhance tracking performance, it is recommended to establish the desired sideslip angle as zero [75]:

$$\beta_{des} = 0, \tag{9}$$

Furthermore, supposing that the vehicle is following a cur-vilinear path with a consistent turning radius R, speed V, a constant angular velocity $\dot{r} = 0$, and a sideslip angle rate of change $\cdot\dot{\beta} = 0$ [76], the necessary yaw rate can be expressed as:

$$r_{des} = \frac{V}{L + \frac{m}{L}(\frac{l_r}{C_{\alpha f}} - \frac{l_f}{C_{\alpha r}})V^2}\delta_f, \tag{10}$$

This equation establishes the connection between the yaw rate and the front wheel angle of rotation when the vehicle is at a standstill.

B. MPC CONTROLLER

As depicted in Fig. 6, the objective of the MPC controller is to determine the required moment M_z so that the sideslip and yaw rate follow the desired output reference. By utilizing a model of the car, MPC makes predictions about how these variables will behave in the future for a given horizon N_p. It also incorporates an optimizer that steers the predicted outputs towards the desired reference. To optimize the increments of the control signal ΔM_z the optimization problem formulated is

$$\min_{\Delta M_z} (Y_{ref} - Y)^T Q(Y_{ref} - Y) + \Delta M_z^T R \Delta M_z \tag{11}$$

subject to the constrains in (8). Here Q, R are weighting matrices,

$$\Delta M_z = [\Delta M_z(k), \Delta M_z(k+1), \Delta M_z(k+2), \ldots, \Delta M_z(k+N_c)]^T$$

is the input sequence, the system output sequence is

$$Y = [\beta(k+1), r(k+1), \beta(k+2), r(k+2), \ldots, \beta(k+N_p), r(k+N_p)]^T$$

and the reference signal for both outputs is given by

$$Y_{ref} = [\beta_{ref}(k+1), r_{ref}(k+1), \beta_{ref}(k+2), r_{ref}(k+2), \ldots, \beta_{ref}(k+N_p), r_{ref}(k+N_p)]^T$$

In linear MPC, the predicted output sequence Y is obtained using model (3), which was discretized in MATLAB using a zero-order hold, and the problem is solved using the Matlab function *quadprog*. In NMPC, the Simmechanics model predicts how the system behaves, providing outputs $\beta[k]$ and $r[k]$ during the prediction horizon N_p. Then, we utilize the interior-point algorithm through *fmincon* to minimize the cost function. As customary, at each time step, we apply solely the initial input from the sequence of optimal inputs, and this process is reiterated in every time step.

In our work, to tune controller parameters for MPC and NMPC, a systematic approach is followed. This process begins with defining an all-encompassing cost function that incorporates elements associated with tracking error, control effort, and state constraints. The primary objective is to discover parameter values that minimize a cost function while adhering to control objectives and constraints. Then, an initial estimate of Q and R based on engineering expertise and prior knowledge is taken. Subsequently, the adjustment of Q and R values has been accomplished through systematic trial-and-error tuning. The system is then subjected to diverse scenarios and disturbances to assess the controllers' performance. Evaluation is based on Root Mean Square Error (RMSE) cal-culations between the desired and measured yaw rate and side slip angle, serving as a performance indicator. Fine-tuning is achieved through iterative refinement. Moreover, trade-offs between conflicting control goals are carefully considered, ensuring an equilibrium between various objectives, such as improving tracking accuracy while minimizing control effort. Ultimately, a comparative evaluation determines which controller aligns better with the desired criteria.

C. TORQUE VECTOR CONTROLLER

The torque allocation for each individual motor is calculated using a low-level TVC system as outlined in [77]. The TVC is responsible for apportioning the torque among the four motors to produce the desired angular momentum, achieving this by distributing the generated angular momentum as follows:

$$M_{z,front} = 0.5M_z, \quad M_{z,rear} = 0.5M_z, \tag{12}$$

$$M_{z,front} = F_{x,fr}\frac{t_f}{2} + F_{x,fl}\frac{t_f}{2}. \tag{13}$$

The longitudinal forces on the right and left sides are allocated, following the principles outlined in [78], based on the vertical forces and road friction, and are determined as:

$$F_{x,f} = \sqrt{(\mu F_z)^2 - F_y^2}, \tag{14}$$

$$F_{x,f} = \frac{M_{z,front}}{t_f}. \tag{15}$$

here, $F_{x,fr}, F_{x,fl}, F_{x,rr}, F_{x,rl}$ represent the longitudinal forces acting on the front-right, front-left, rear-right, and rear-left wheels, respectively. Additionally, t_f and t_r denote the width of the front and rear axles, while μ represents the tire-road friction coefficient. Employing a similar approach for the rear wheels, we obtain

$$F_{x,r} = \frac{M_{z,rear}}{t_r}. \tag{16}$$

The upper-level controller supplies the necessary angular momentum needed to adjust the vehicle's trajectory, and the electric motor's requisite torque is determined as

$$T_m - F_x r_{dyn} = I_{yy}\dot{\omega}, \tag{17}$$

$$T_{m,i} = \frac{r_{dyn}}{t_f} M_{z,front} + I_{yy}\dot{\omega}, \tag{18}$$

here, T_m represents the torque produced by the electric motor, r_{dyn} stands for the dynamic tire radius, I_{yy} denotes the moment of inertia of the wheel about its axis of rotation, and ω represents the wheel's angular velocity.

IV. ARTIFICIAL NEURAL NETWORKS AND NARX MODEL

ANNs are versatile tools utilized for a wide range of applications, including clustering, recognition, pattern classification, optimization, function approximation, and prediction [79], [80], [81], [82]. Inspired by the biological neural system, ANNs possess remarkable abilities to learn, store, and recall information [83], [84]. As black-box modeling tools, they excel at performing nonlinear mappings from an m-dimensional input space to an n-dimensional output space, even when the relationships between the input and output spaces are unknown [47], [85], [86]. In our vehicle simulation model, continuous measurements of yaw rate and side slip angle, used for comprehending, controlling, and simulating dynamic behavior, create a time series dataset, which can be used to train NARX neural networks yielding a powerful predictor for the corresponding time series data due to their feedback connections to improve its performance in nonlinear time series prediction [83], [84], [87].

Fig. 7 shows two NARX neural network architectures: series-parallel (open-loop) and parallel (closed-loop), respectively represented in (19) and (20),

$$\hat{x}(k+1) = F(x(k), \ldots, x(k-d), u(k), \ldots, u(k-d)) \tag{19}$$

$$\hat{x}(k+1) = F(\hat{x}(k), \ldots, \hat{x}(k-d), u(k), \ldots, u(k-d)) \tag{20}$$

where $F(\cdot)$ is the mapping function of the neural network, $\hat{x}(k+1)$ is the predicted output of the NARX at time $k+1$, $\hat{x}(k)$ is the past predicted output of the NARX, $x(k)$ is the corresponding true value, $u(k)$ is the input of the NARX, and

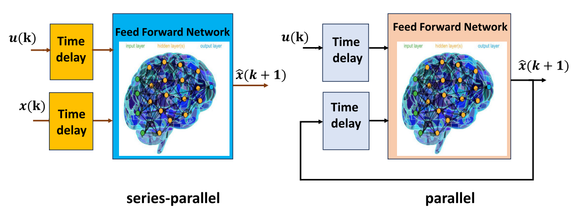


FIGURE 7. Architectures of the NARX neural network.

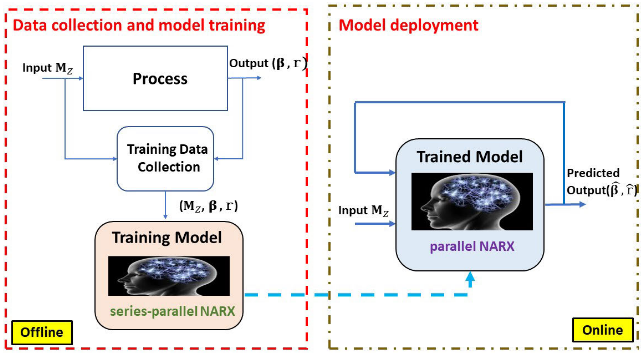


FIGURE 8. Training and deployment phases.

d is the number of delays. In the series-parallel architecture, predictions for each time step k are made by providing both present and past exogenous inputs $u(k), \ldots, u(k-d)$ and the past true values of the time series $x(k), \ldots, x(k-d)$. These inputs are utilized to forecast the next time step, $k+1$. Conversely, in the parallel architecture, the prediction is made using the present and past values of exogenous inputs $u(k), \ldots, u(k-d)$ and the past predicted values of the time series $\hat{x}(k), \ldots, \hat{x}(k-d)$. The series-parallel architecture is chosen for training due to the availability of true past values of side slip β, yaw rate r, and control input M_z (see Fig.8). It offers two advantages: more precise predictions using true values as inputs and a pure feedforward structure allowing the use of standard training algorithms for Multi-Layer Perceptron (MLP). After training, the NARX model is converted to parallel architecture via the feedback connection of the predicted outputs, benefiting multi-step ahead prediction for $(\hat{\beta}, \hat{r})$. The NARX neural network model uses the structure of MLP to approximate the mapping function, a powerful structure capable of learning continuous nonlinear mappings [88]. It consists of three layers: input, hidden, and output, where each neuron's output is given by

$$a_i^{(l)} = g^{(l)}\left(\sum_{j=1}^{n^{(l-1)}} w_{ji}^{(l-1)} a_j^{(l-1)} + b_i^{(l-1)}\right) \tag{21}$$

where a_i^l is the output of node i and layer l, w_{ji}^l is the weight vector between neurons j and i, $g^{(l)}$ is the activation function, and b_i^l is the bias unit.

A. NARX TRAINING

The neural network parameters are determined iteratively through initialization, error calculation, and weight adjustment until a specified criterion is met. Back-propagation is used to compute errors for networks with hidden layers. The appropriate number of nodes in an MLP is found through a trial-and-error process [89], [90].

The data gathering process involved conducting 20 separate simulations under different conditions on various roadways with profiles as those depicted in Fig. 9. A total of 20020 samples were generated from these simulations,

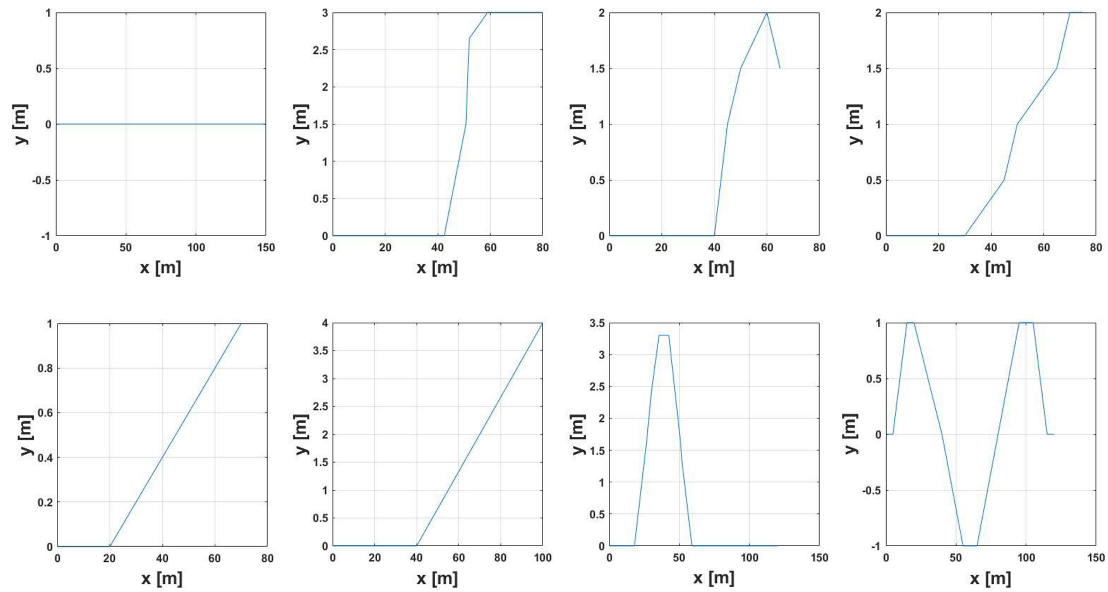


FIGURE 9. Roads schemes for training and validation.

comprising input values of M_z and the corresponding output data for slip angle β and yaw rate r. The dataset has been split into a training set (70%), a validation set (15%) and a test set (15%), randomly chosen. The neural dynamics prediction model can be described as

$$\hat{x}(k+1) = F(x[k], M_z[k]), \tag{22}$$

where $x[k] = \left[\beta[k], r[k]\right]^T$. In this way, the equations of the neural network are planned as follows [68], [69], and [91]:

$$\begin{aligned} Z_1 &= W_1(x[k], M_z[k]) + B_1 \\ H_1 &= g(Z_1) \\ \hat{x}(k+1) &= W_2 H_1 + B_2 \end{aligned} \tag{23}$$

where W_1, W_2, B_1 and B_2 are the weight matrices of the neural network, Z_1 is the weighted sum of inputs, H_1 is the activation value, and g is the activation function, which in this case is the hyperbolic tangent function. Eq. (23) can be extended for multiple layers. To learn the matrices, the mean square error function is defined, and the gradient descent algorithm is utilized to optimize the parameters.

The training process adjusts neural network weights iteratively to match the targets closely using the given inputs and desired outputs. *Tanh* activation function is used in all layers, except the output layer, which employs a linear function. For more details, see Table 2.

In this work, several neural networks underwent training with various hidden layer configurations. After utilizing the neural network toolbox, we obtained the Pearson correlation coefficient (R) results, which measure the correlation between outputs and targets of an open-loop NARX model. Subsequently, a feedback NARX model is employed in the simulation. The resulting values are shown in Table 3, where a regression value close to $R = 1$ indicates a very strong positive linear relationship between the variables, suggesting an excellent fit for the data. In Section V, the chosen trained neural networks are thoroughly evaluated and compared to each other. This evaluation encompasses an assessment of their overall performance, accuracy, and efficiency, considering the complete model. The objective is to identify the most effective and suitable model by making comparisons among these neural networks.

B. MACHINE LEARNING WITH MPC

MPC-NPL stands for a MPC algorithm with nonlinear prediction and linearization. The approach involves finding a local linear approximation of the nonlinear neural model online at each sampling instant, allowing the output predictions to depend on the calculated control policy in a linear manner. This feature enables the MPC-NPL algorithm to solve a quadratic programming problem online efficiently [68], [69]. That is, MPC-NPL improves conventional linear MPC by handling nonlinear behavior, adapting to complexity, and maintaining accuracy.

In this approach, the NN model is linearized yielding the incremental space state representation:

$$\begin{aligned} \hat{x}_e(k+1) &= A_e x_e[k] + B_e \Delta M_z \\ y[k] &= C_e x[k] \end{aligned} \tag{24}$$

where $x_e[k]$, A_e, B_e and C_e are:

$$x_e[k] = \begin{bmatrix} \Delta x[k] \\ y[k] \end{bmatrix} \quad A_e = \begin{bmatrix} A_m & 0_m \\ C_m A_m & I \end{bmatrix}$$

$$B_e = \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} C_e = \begin{bmatrix} 0_m \; I \end{bmatrix}$$

TABLE 2. Training parameters of the neural networks.

Range	Transfer function	μ	μ increase ratio	μ decrease ratio	Max μ	Max epochs	Min gradient	Max validation checks
$[1, -1]$	tansig pureling	10^{-3}	10	10^{-1}	10^{10}	10^{3}	10^{-7}	6

TABLE 3. The regression R values of different NNS.

	R (training)	R (validation)	R (test)
5	0.9974	0.9985	0.9979
10-5	0.9978	0.9983	0.9981
15-10	0.9983	0.9981	0.9987
20-15-10	0.9987	0.9977	0.9973

To obtain the system matrices, the partial derivative of the function $F(x[k], u[k])$ with respect to the current state $x[k]$ and the input $M_z[k]$ is found [68], [69] as follows

$$A_m = \frac{\partial \hat{x}[k+1]}{\partial x[k]}, \quad B_m = \frac{\partial \hat{x}[k+1]}{\partial M_z[k]}, C_m = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix}$$

Therefore, we can predict these future states using the augmented model as

$$\hat{Y} = Fx[k] + \Phi \Delta M_z \tag{25}$$

here

$$F = \begin{bmatrix} C_eA_e & C_eA_e^2 & C_eA_e^3 & \dots & C_eA_e^{N_p} \end{bmatrix}^T,$$
$$\Phi = \begin{bmatrix} C_eB_e & 0 & \cdots & 0 \\ C_eA_eB_e & C_eB_e & \cdots & 0 \\ C_eA_e^2B_e & C_eA_eB_e & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C_eA_e^{N_p-1}B_e & C_eA_e^{N_p-2}B_e & \cdots & C_eA_e^{N_p-N_c}B_e \end{bmatrix} \tag{26}$$

To optimize the increments of the control signal ΔM_z, the optimization problem is formulated as (11) and solved using the Matlab function *quadprog*.

C. MACHINE LEARNING WITH NMPC

NMPC–NO denotes the optimal control action obtained providing NMPC with the nonlinear ML model, with *fmincon* performing the nonlinear optimization. The proposed approach generates control signals that exhibit almost the same characteristics as NMPC, but with significantly lower execution time [69]. Here, the future states are iteratively predicted as

$$\begin{aligned} \hat{x}[k+1] &= W_2 \cdot \tanh(W_1 \cdot (x[k], (M_z[k] \\ &\quad + \Delta M_z[1]) + B_1) + B_2 \\ \hat{x}[k+2] &= W_2 \cdot \tanh(W_1 \cdot (\hat{x}[k+1], (M_z[k] \\ &\quad + \Delta M_z[2]) + B_1) + B_2 \\ &\vdots \\ \hat{x}[k+N_p] &= W_2 \cdot \tanh(W_1 \cdot (\hat{x}[k+N_p-1], (M_z[k] \\ &\quad + \Delta M_z[k+N_c]) + B_1) + B_2 \end{aligned} \tag{27}$$

The predicted output of the system at each time step is

$$y[k+1] = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} \cdot \hat{x}[k+1] \tag{28}$$

Again, the increments of the control signal ΔM_z are optimized following Eq. (11).

V. SIMULATION RESULTS

The control strategies are tested through a double-lane change and a slalom maneuver through both wet and dry roads as in Fig. 10. The simulation is conducted within SimMechanics, demonstrating the vehicle's ability to navigate a curve effectively [92]. The tire model parameters in SimMechanics were configured as follows: $c_1 = 1.1973, c_2 = 25.168, c_3 = 0.5373$ in case of dry road and $c_1 = 0.857, c_2 = 33.822, c_3 = 0.347$ in case of wet road [21].

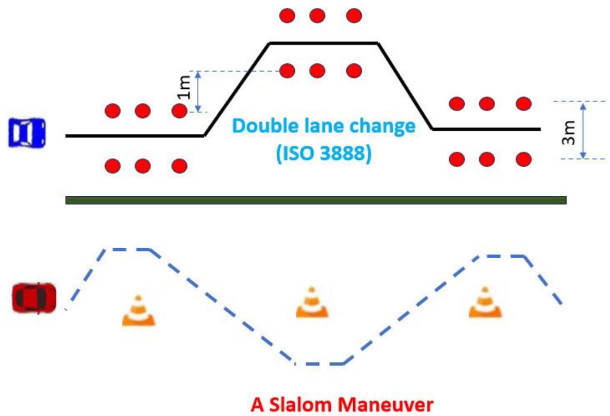
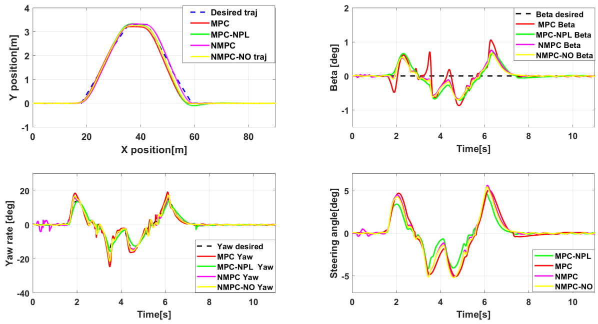


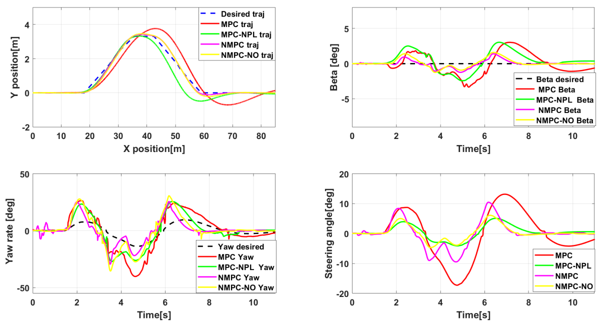
FIGURE 10. A double lane change and a slalom maneuvers.

A. DOUBLE-LANE CHANGE ROAD TEST

As depicted in Fig. 11, the MPC-NPL controller outperforms conventional linear MPC, demonstrating superior performance with decreased tracking error and oscillations, resulting in smoother driving behavior. Meanwhile, the performance of NMPC-NO approaches that of NMPC in terms of stability when tracking the desired values.

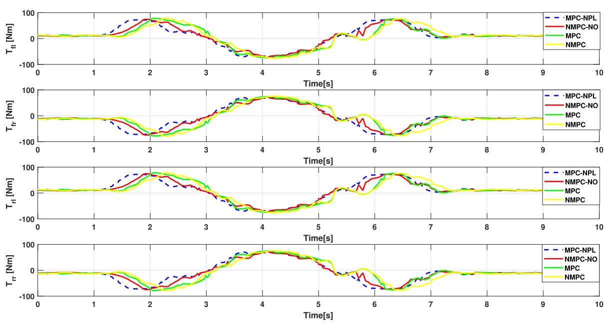


(a) Dry road

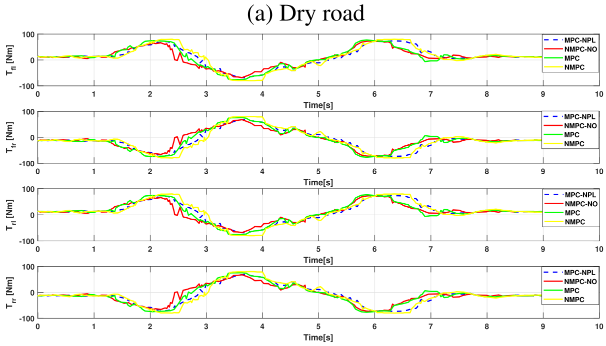


(b) Wet road

FIGURE 11. Double-lane change response.

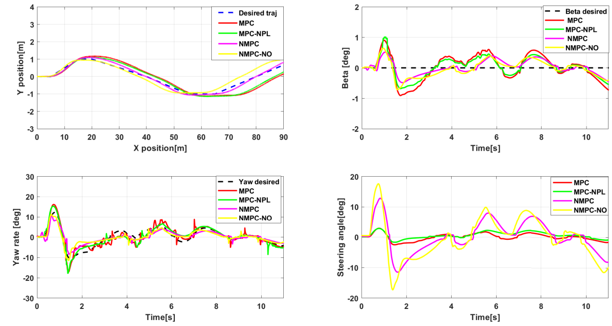


(a) Dry road

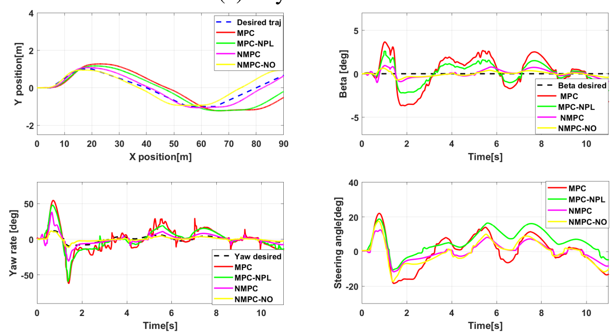


(b) Wet road

FIGURE 12. Double-lane change response for motors' torques.

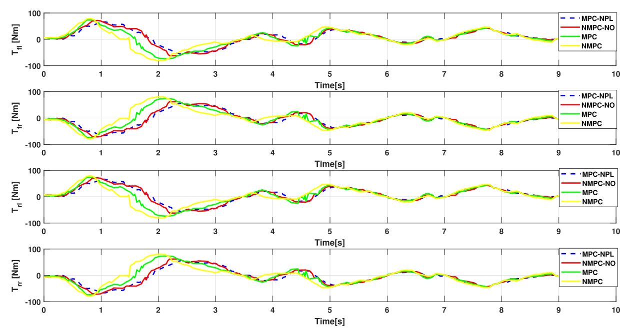


(a) Dry road

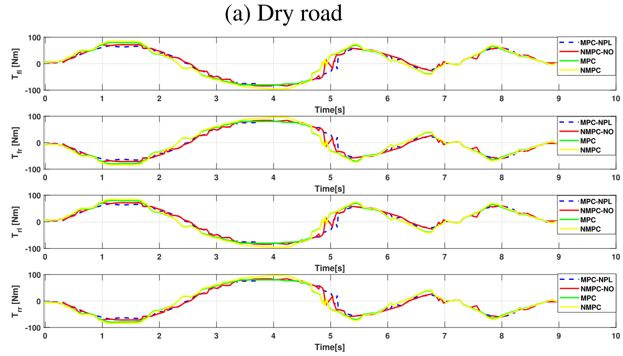


(b) Wet road

FIGURE 13. Slalom response.



(a) Dry road



(b) Wet road

FIGURE 14. Slalom response for motors' torques.

B. SLALOM ROAD TEST

As Fig. 13 shows, the stability of NMPC-NO is close to that of NMPC. On the other hand, the MPC-NPL controller surpasses the performance of conventional MPC, exhibiting reduced tracking error, oscillations, and smoother driving behavior. In terms of RMSE performance as shown in Table 4, different neural network models were employed to evaluate controllers across various road conditions, with NMPC-NO showing convergence to NMPC stability levels and improved computational efficiency as in the second column. Additionally, a deep learning neural network with two hidden layers highlighted the superior RMSE performance of the MPC-NPL controller over the linear MPC controller. As illustrated in Figs. 12,14 these findings provide additional perspective on the controller's efficacy in maintaining vehicle stability across diverse scenarios, all while ensuring that each motor remains below the maximum torque threshold of 80 $N.m$.

C. COMPUTATION TIME

In control systems, computation time typically refers to the time required for a controller to calculate and execute its control actions and is a relevant indicator in determining the performance in real-time applications where the system must respond quickly to changes in the environment. The computation time depends on several factors, including the processing speed of the control hardware, the complexity of the control algorithm and the system being controlled. In the context of vehicle lateral stability, the sample time is also related to the collection of lateral dynamics measurements, influencing the frequency of adjustments that determine the real-time response and its stability.

Considering a 10 milliseconds sample time, only MPC-NPL and NMPC-NO fit the requirement, with MPC-NPL being much faster. However, note that our simulations were performed in Matlab, which is good for mathematical prototyping but slow, therefore admitting much faster execution times if implemented in programming languages such as C. In particular, the computation times shown in Table 4 and Fig. 15 were evaluated using Matlab on a computer equipped with 16 GB RAM, a 2.6 GHz Intel Core i7 processor, and a 512 GB SSD. Also, note that the computation resources available at the car play a fundamental role in the computation time. Bearing all these issues in mind, there are grounds for thinking that both MPC-NPC and NMPC-NO can be suitable for real-time implementation in a dedicated control system with sufficient computation power.

TABLE 4. Computation time of the obtained controllers.

	Mean computation time [s]	Min [s]	Max [s]
MPC	0.0052	0.0019	0.0115
MPC-NPL	0.0021	0.0018	0.0065
NMPC	14.15	13.195	15.13
NMPC-NO	0.007	0.0006	0.015

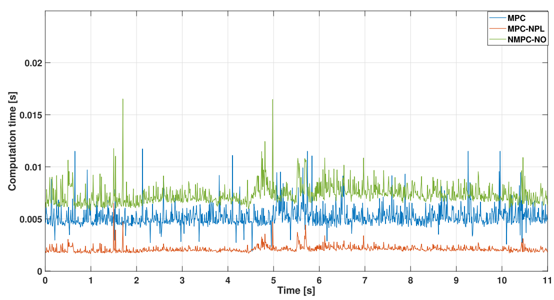


FIGURE 15. Computation time calculation.

VI. CONCLUSION AND FUTURE WORK

The application of MPC for lateral stability control can optimize handling performance while satisfying different specifications. However, nonlinear MPC entails significant computational costs. To address this limitation, this study proposes utilizing artificial neural networks to approximate the internal model of the model predictive controller. To evaluate this approach, a simulated vehicle with four electric wheel motors based on a real-life prototype was utilized. A dataset was created comprising more than 20 thousand data, which were employed to train and validate various neural network architectures, and several tests were conducted.

Our tests show that the NMPC controller delivers superior results regarding the yaw rate and side slip angle, but it is computationally demanding, making it unfeasible for real-time implementation. Combining artificial neural networks with both linear and nonlinear MPC reduces the computation time, facilitating its utilization in real-time scenarios. In particular, the NMPC controller calculation time is reduced from roughly 15 s to 0.007 s.

In future works, training an ANN that establishes a direct connection between the MPC controller's input and the control action. The suggested methodologies will be assessed using the real vehicle depicted in Fig. 2. The case of failures will also be analyzed to enhance the vehicle's handling performance and prevent severe accidents.

REFERENCES

[1] K. Shi, X. Yuan, and L. Liu, "Model predictive controller-based multi-model control system for longitudinal stability of distributed drive electric vehicle," *ISA Trans.*, vol. 72, pp. 44–55, Jan. 2018.

[2] B. Huang, S. Wu, S. Huang, and X. Fu, "Lateral stability control of four-wheel independent drive electric vehicles based on model predictive control," *Math. Problems Eng.*, vol. 2018, pp. 1–15, Jan. 2018.

[3] M. Seo, C. Yoo, S.-S. Park, and K. Nam, "Development of wheel pressure control algorithm for electronic stability control (ESC) system of commercial trucks," *Sensors*, vol. 18, no. 7, p. 2317, Jul. 2018.

[4] M. L. De Klerk and A. K. Saha, "A comprehensive review of advanced traction motor control techniques suitable for electric vehicle applications," *IEEE Access*, vol. 9, pp. 125080–125108, 2021.

[5] Z. Zhao and X. Fan, "Review of vehicle active safety systems and their coordinated control," *Recent Patents Mech. Eng.*, vol. 14, no. 1, pp. 4–17, Feb. 2021.

[6] B. Lenzo, M. Zanchetta, A. Sorniotti, P. Gruber, and W. De Nijs, "Yaw rate and sideslip angle control through single input single output direct yaw moment control," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 1, pp. 124–139, Jan. 2021.

[7] P. Sun, A. S. Trigell, L. Drugge, and J. Jerrelind, "Energy efficiency and stability of electric vehicles utilising direct yaw moment control," *Vehicle Syst. Dyn.*, vol. 60, no. 3, pp. 930–950, Mar. 2022.

[8] M. De Bernardis, G. Rini, F. Bottiglione, A. E. Hartavi, and A. Sorniotti, "On nonlinear model predictive direct yaw moment control for trailer sway mitigation," *Vehicle Syst. Dyn.*, vol. 61, no. 2, pp. 445–471, Feb. 2023.

[9] L. Zhang, H. Ding, Y. Huang, H. Chen, K. Guo, and Q. Li, "An analytical approach to improve vehicle maneuverability via torque vectoring control: Theoretical study and experimental validation," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4514–4526, May 2019.

[10] A. El Kharki, Z. Boulghasoul, L. Et-Taaj, and A. Elbacha, "A new intelligent control strategy of combined vector control and direct torque control for dynamic performance improvement of induction motor drive," *J. Electr. Eng. Technol.*, vol. 17, no. 5, pp. 2829–2847, Sep. 2022.
[11] Z. Zainal, W. Rahiman, and M. Baharom, "Yaw rate and sideslip control using pid controller for double lane changing," *J. Telecommun., Electron. Comput. Eng. (JTEC)*, vol. 9, nos. 3–7, pp. 99–103, 2017.
[12] Z. Y. Wang, U. Montanaro, S. Fallah, A. Sorniotti, and B. Lenzo, "A gain scheduled robust linear quadratic regulator for vehicle direct yaw moment control," *Mechatronics*, vol. 51, pp. 31–45, Jan. 2018.
[13] F. L. Silva, L. C. A. Silva, J. J. Eckert, and M. A. M. Lourenço, "Robust fuzzy stability control optimization by multi-objective for modular vehicle," *Mechanism Mach. Theory*, vol. 167, Jan. 2022, Art. no. 104554.
[14] G. Vasiljevic and S. Bogdan, "Model predictive control based torque vectoring algorithm for electric car with independent drives," in *Proc. 24th Medit. Conf. Control Autom. (MED)*, Jun. 2016, pp. 316–321.
[15] Y. Suzuki, Y. Kano, and M. Abe, "A study on tyre force distribution controls for full drive-by-wire electric vehicle," *Veh. Syst. Dyn.*, vol. 52, pp. 235–250, Jan. 2014.
[16] R. Dhamija, S. Basak, A. Singh, S. Sengupta, T. Q. Dinh, and J. I. Yoon, "Advanced control strategies for high performance four-wheel drive electric vehicle," in *Proc. 25th Int. Conf. Mechatronics Technol. (ICMT)*, 2022, pp. 1–6.
[17] G. D. Filippis, B. Lenzo, A. Sorniotti, P. Gruber, and W. D. Nijs, "Energy-efficient torque-vectoring control of electric vehicles with multiple drivetrains," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 4702–4715, Jun. 2018.
[18] E. Siampis, E. Velenis, S. Gariuolo, and S. Longo, "A real-time nonlinear model predictive control strategy for stabilization of an electric vehicle at the limits of handling," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 6, pp. 1982–1994, Nov. 2017.
[19] A. Parra, D. Tavernini, P. Gruber, A. Sorniotti, A. Zubizarreta, and J. Pérez, "On nonlinear model predictive control for energy-efficient torque-vectoring," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 173–188, Jan. 2021.
[20] H. Guo, F. Liu, F. Xu, H. Chen, D. Cao, and Y. Ji, "Nonlinear model predictive lateral stability control of active chassis for intelligent vehicles and its FPGA implementation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 1, pp. 2–13, Jan. 2019.
[21] D. M. Rodríguez, "Contributions to power management and dynamics control in hybrid vehicles," Ph.D. dissertation, Univ. de Sevilla, 2014.
[22] M. N. Alcántara, "Diseño de controladores para vehículo eléctrico con cuatro motores en rueda," Tech. Rep., 2017.
[23] M. Kvasnica, P. Bakarác, and M. Klauco, "Complexity reduction in explicit MPC: A reachability approach," *Syst. Control Lett.*, vol. 124, pp. 19–26, Feb. 2019.
[24] A. Pavlov, M. Müller, C. Manzie, and I. Shames, "Complexity minimisation of suboptimal MPC without terminal constraints," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6089–6094, 2020.
[25] D. Masti, T. Pippia, A. Bemporad, and B. D. Schutter, "Learning approximate semi-explicit hybrid MPC with an application to microgrids," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 5207–5212, 2020.
[26] H. Peng, W. Wang, Q. An, C. Xiang, and L. Li, "Path tracking and direct yaw moment coordinated control based on robust MPC with the finite time horizon for autonomous independent-drive vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 6053–6066, Jun. 2020.
[27] R. N. Liang, E. A. J. Anacleto, and C. N. Meneses, "Data structures for speeding up Tabu search when solving sparse quadratic unconstrained binary optimization problems," *J. Heuristics*, vol. 28, no. 4, pp. 433–479, Aug. 2022.
[28] F. Truger, J. Barzen, M. Bechtold, M. Beisel, F. Leymann, A. Mandl, and V. Yussupov, "Warm-starting and quantum computing: A systematic mapping study," 2023, *arXiv:2303.06133*.
[29] A. Cano, "A survey on graphic processing unit computing for large-scale data mining," *WIREs Data Mining Knowl. Discovery*, vol. 8, no. 1, p. e1232, Jan. 2018.
[30] T. M. Blaha, E. J. J. Smeur, and B. D. W. Remes, "A survey of optimal control allocation for aerial vehicle control," *Actuators*, vol. 12, no. 7, p. 282, Jul. 2023.
[31] J. Hou, G. Chen, J. Huang, Y. Qiao, L. Xiong, F. Wen, A. Knoll, and C. Jiang, "Large-scale vehicle platooning: Advances and challenges in scheduling and planning techniques," *Engineering*, vol. 28, pp. 26–48, Sep. 2023.
[32] Z. Zhou, C. Rother, and J. Chen, "Event-triggered model predictive control for autonomous vehicle path tracking: Validation using CARLA simulator," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 6, pp. 3547–3555, Jun. 2023.
[33] B. Wang, J. Huang, C. Wen, J. Rodriguez, C. Garcia, H. B. Gooi, and Z. Zeng, "Event-triggered model predictive control for power converters," *IEEE Trans. Ind. Electron.*, vol. 68, no. 1, pp. 715–720, Jan. 2021.
[34] C. Wu, S. Jiang, S. Gao, Y. Liu, and H. Han, "Event-triggered model predictive control for dynamic energy management of electric vehicles in microgrids," *J. Cleaner Prod.*, vol. 368, Sep. 2022, Art. no. 133175.
[35] S. Yang, W. Chen, and M. P. Wan, "A machine-learning-based event-triggered model predictive control for building energy management," *Building Environ.*, vol. 233, Apr. 2023, Art. no. 110101.
[36] J. Yoo and K. H. Johansson, "Event-triggered model predictive control with a statistical learning," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 4, pp. 2571–2581, Apr. 2021.
[37] J. Chen, X. Meng, and Z. Li, "Reinforcement learning-based event-triggered model predictive control for autonomous vehicle path following," in *Proc. Amer. Control Conf. (ACC)*, 2022, pp. 3342–3347.
[38] Y. Xiao, X. Zuo, I. Kaku, S. Zhou, and X. Pan, "Development of energy consumption optimization model for the electric vehicle routing problem with time windows," *J. Cleaner Prod.*, vol. 225, pp. 647–663, Jul. 2019.
[39] Y. Xu, W. Tang, B. Chen, L. Qiu, and R. Yang, "A model predictive control with preview-follower theory algorithm for trajectory tracking control in autonomous vehicles," *Symmetry*, vol. 13, no. 3, p. 381, Feb. 2021.
[40] L. Xiang, C. W. Lee, O. Zikanov, and C.-C. Hsu, "Efficient reduced order model for heat transfer in a battery pack of an electric vehicle," *Appl. Thermal Eng.*, vol. 201, Jan. 2022, Art. no. 117641.
[41] N. Rama and D. Robinette, "Computationally efficient reduced-order powertrain model of a multi-mode plug-in hybrid electric vehicle for connected and automated vehicles," Tech. Rep., 2019.
[42] S. Ruiz-Moreno, J. R. D. Frejo, and E. F. Camacho, "Model predictive control based on deep learning for solar parabolic-trough plants," *Renew. Energy*, vol. 180, pp. 193–202, Dec. 2021.
[43] D. Piga, M. Forgione, S. Formentin, and A. Bemporad, "Performance-oriented model learning for data-driven MPC design," *IEEE Control Syst. Lett.*, vol. 3, no. 3, pp. 577–582, Jul. 2019.
[44] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 3, no. 1, pp. 269–296, May 2020.
[45] A. Norouzi, H. Heidarifar, H. Borhan, M. Shahbakhti, and C. R. Koch, "Integrating machine learning and model predictive control for automotive applications: A review and future directions," *Eng. Appl. Artif. Intell.*, vol. 120, Apr. 2023, Art. no. 105878.
[46] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Vehicles*, vol. 1, no. 1, pp. 33–55, Mar. 2016.
[47] Y. Xiao, X. Zhang, X. Xu, X. Liu, and J. Liu, "Deep neural networks with Koopman operators for modeling and control of autonomous vehicles," *IEEE Trans. Intell. Veh.*, vol. 8, no. 1, pp. 135–146, 2023.
[48] M. T. Gillespie, C. M. Best, E. C. Townsend, D. Wingate, and M. D. Killpack, "Learning nonlinear dynamic models of soft robots for model predictive control with neural networks," in *Proc. IEEE Int. Conf. Soft Robot. (RoboSoft)*, Apr. 2018, pp. 39–45.
[49] L. A. Basina, B. K. Irdmousa, J. M. Velni, H. Borhan, J. D. Naber, and M. Shahbakhti, "Data-driven modeling and predictive control of maximum pressure rise rate in RCCI engines," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2020, pp. 94–99.
[50] L. C. Sousa and H. V. H. Ayala, "Nonlinear tire model approximation using machine learning for efficient model predictive control," *IEEE Access*, vol. 10, pp. 107549–107562, 2022.
[51] K. Bieker, S. Peitz, S. L. Brunton, J. N. Kutz, and M. Dellnitz, "Deep model predictive flow control with limited sensor data and online learning," *Theor. Comput. Fluid Dyn.*, vol. 34, no. 4, pp. 577–591, Aug. 2020.
[52] Y. Vaupel, N. C. Hamacher, A. Caspari, A. Mhamdi, I. G. Kevrekidis, and A. Mitsos, "Accelerating nonlinear model predictive control through machine learning," *J. Process Control*, vol. 92, pp. 261–270, Aug. 2020.
[53] M. Toub, M. Shahbakhti, R. D. Robinett, and G. Aniba, "MPC-trained ANFIS for control of MicroCSP integrated into a building HVAC system," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2019, pp. 241–246.
[54] M. Novak and T. Dragicevic, "Supervised imitation learning of finite-set model predictive control systems for power electronics," *IEEE Trans. Ind. Electron.*, vol. 68, no. 2, pp. 1717–1723, Feb. 2021.

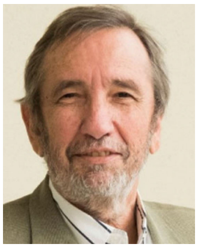
[55] R. Kohut, L. Galcíová, K. Fedorová, T. Ábelová, M. Bakosová, and M. Kvasnica, "Hidden Markov model-based warm-start of active set method in model predictive control," in *Proc. 23rd Int. Conf. Process Control (PC)*, Jun. 2021, pp. 60–65.
[56] D. Masti and A. Bemporad, "Learning binary warm starts for multiparametric mixed-integer quadratic programming," in *Proc. 18th Eur. Control Conf. (ECC)*, 2019, pp. 1494–1499.
[57] N. Karnchanachari, M. I. Valls, D. Hoeller, and M. Hutter, "Practical reinforcement learning for MPC: Learning from sparse objectives in under an hour on a real robot," in *Proc. 2nd Conf. Learn. Dyn. Control*, 2020, pp. 211–224.
[58] L. Wei, X. Wang, L. Li, Z. Fan, R. Dou, and J. Lin, "T-S fuzzy model predictive control for vehicle yaw stability in nonlinear region," *IEEE Trans. Veh. Technol.*, vol. 70, no. 8, pp. 7536–7546, Aug. 2021.
[59] C. Greatwood and A. G. Richards, "Reinforcement learning and model predictive control for robust embedded quadrotor guidance and control," *Auto. Robots*, vol. 43, no. 7, pp. 1681–1693, Oct. 2019.
[60] K. P. Wabersich and M. N. Zeilinger, "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems," *Automatica*, vol. 129, Jul. 2021, Art. no. 109597.
[61] M. Zanon and S. Gros, "Safe reinforcement learning using robust MPC," *IEEE Trans. Autom. Control*, vol. 66, no. 8, pp. 3638–3652, Aug. 2021.
[62] A. Cheng and Y. M. Low, "Improved generalization of NARX neural networks for enhanced metamodeling of nonlinear dynamic systems under stochastic excitations," *Mech. Syst. Signal Process.*, vol. 200, Oct. 2023, Art. no. 110543.
[63] P. Takyi-Aninakwa, S. Wang, H. Zhang, Y. Xiao, and C. Fernandez, "A NARX network optimized with an adaptive weighted square-root cubature Kalman filter for the dynamic state of charge estimation of lithium-ion batteries," *J. Energy Storage*, vol. 68, Sep. 2023, Art. no. 107728.
[64] A. Jain, N. Framke, A. Tiwari, and M. Spasov, "Application of feedforward neural networks in battery electric vehicle air conditioning system simulations," Tech. Rep., 2022.
[65] E. Šabanovič, V. Žuraulis, O. Prentkovskis, and V. Skrickij, "Identification of road-surface type using deep neural networks for friction coefficient estimation," *Sensors*, vol. 20, no. 3, p. 612, Jan. 2020.
[66] J. Zhu, Q. Jiang, Y. Shen, C. Qian, F. Xu, and Q. Zhu, "Application of recurrent neural network to mechanical fault diagnosis: A review," *J. Mech. Sci. Technol.*, vol. 36, no. 2, pp. 527–542, Feb. 2022.
[67] M. G. Sani, N. Abdul Wahab, Y. M. Sam, S. I. Samsudin, and I. W. Jamaludin, "Comparison of NARX neural network and classical modelling approaches," *Appl. Mech. Mater.*, vol. 554, pp. 360–365, Jun. 2014.
[68] M. Lawrynczuk, "A family of model predictive control algorithms with artificial neural networks," *Int. J. Appl. Math. Comput. Sci.*, vol. 17, no. 2, pp. 217–232, Jun. 2007.
[69] E. Szczerbicki, *Intelligent Systems for Knowledge Management*, vol. 252. Berlin, Germany: Springer, 2009.
[70] J. Liu, W. Zhuang, H. Zhong, L. Wang, H. Chen, and C.-A. Tan, "Integrated energy-oriented lateral stability control of a four-wheel-independent-drive electric vehicle," *Sci. China Technol. Sci.*, vol. 62, no. 12, pp. 2170–2183, Dec. 2019.
[71] H. Pacejka, *Tire and Vehicle Dynamics*. Amsterdam, The Netherlands: Elsevier, 2005.
[72] A. Hassan, J. R. D. Frejo, and J. M. Maestre, "Enhancement handling performance of 4-wheels drive electrical vehicle using advanced control technique," in *Proc. 43rd Jornadas de Automática*. Univ. da Coruña, 2022, pp. 530–536.
[73] X. Hu, H. Chen, Z. Li, and P. Wang, "An energy-saving torque vectoring control strategy for electric vehicles considering handling stability under extreme conditions," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 10787–10796, Oct. 2020.
[74] X. Zhang, P. Wang, Z. Li, F. Wang, and H. Chen, "Estimation and analysis of vehicle stability region under complex road conditions," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2020, pp. 3036–3041.
[75] H. Wang, Y. Huang, A. Khajepour, Y. Zhang, Y. Rasekhipour, and D. Cao, "Crash mitigation in motion planning for autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 9, pp. 3313–3323, Sep. 2019.
[76] R. Rajamani, *Vehicle Dynamics and Control*. Berlin, Germany: Springer, 2011.
[77] G. Hernández, C. B. Alba, D. Marcos, and C. Montero, "Control de estabilidad basado en mpc para un vehículo eléctrico con motores en rueda," *Jornadas de automática*, pp. 887–894, 2015.
[78] O. Mokhiamar and M. Abe, "Simultaneous optimal distribution of lateral and longitudinal tire forces for the model following control," *J. Dyn. Syst. Meas. Control*, vol. 126, pp. 753–763, Dec. 2004.
[79] B. Widrow, D. E. Rumelhart, and M. A. Lehr, "Neural networks: Applications in industry, business and science," *Commun. ACM*, vol. 37, no. 3, pp. 93–106, 1994.
[80] F. Mumali, "Artificial neural network-based decision support systems in manufacturing processes: A systematic literature review," *Comput. Ind. Eng.*, vol. 165, Mar. 2022, Art. no. 107964.
[81] E. F. Camacho and A. J. Gallego, "Optimal operation in solar trough plants: A case study," *Sol. Energy*, vol. 95, pp. 106–117, Sep. 2013.
[82] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, "A survey on trajectory-prediction methods for autonomous driving," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 3, pp. 652–674, Sep. 2022.
[83] S. Mohanty, P. K. Patra, and S. S. Sahoo, "Prediction of global solar radiation using nonlinear auto regressive network with exogenous inputs (NARX)," in *Proc. 39th Nat. Syst. Conf. (NSC)*, Dec. 2015, pp. 1–6.
[84] L. Ruiz, M. Cuéllar, M. Calvo-Flores, and M. Jiménez, "An application of non-linear autoregressive neural networks to predict energy consumption in public buildings," *Energies*, vol. 9, no. 9, p. 684, Aug. 2016.
[85] M. A. M. Sadeeq and A. M. Abdulazeez, "Neural networks architectures design, and applications: A review," in *Proc. Int. Conf. Adv. Sci. Eng. (ICOASE)*, Dec. 2020, pp. 199–204.
[86] A. Kusiak, "Convolutional and generative adversarial neural networks in manufacturing," *Int. J. Prod. Res.*, vol. 58, no. 5, pp. 1594–1604, Mar. 2020.
[87] E. Cadenas, W. Rivera, R. Campos-Amezcua, and C. Heard, "Wind speed prediction using a univariate ARIMA model and a multivariate NARX model," *Energies*, vol. 9, no. 2, p. 109, Feb. 2016.
[88] T. Fine, *Feedforward Neural Network Methodology* (Softcover Reprint of the Original), 1st ed. 2013.
[89] B. J. Erickson, "Basic artificial intelligence techniques: Machine learning and deep learning," *Radiologic Clinics*, vol. 59, no. 6, pp. 933–940, 2021.
[90] J. Spicer and A. N. Sanborn, "What does the mind learn? A comparison of human and machine learning representations," *Current Opinion Neurobiol.*, vol. 55, pp. 97–102, Apr. 2019.
[91] J. M. Zamarreño, P. Vega, L. D. Garcia, and M. Francisco, "State-space neural network for modelling, prediction and control," *Control Eng. Pract.*, vol. 8, no. 9, pp. 1063–1075, Sep. 2000.
[92] H. Kanchwala, J. Wideberg, C. B. Alba, and D. Marcos, "Control of an independent 4WD electric vehicle by DYC method," *Int. J. Vehicle Syst. Model. Test.*, vol. 10, no. 2, pp. 168–184, 2015.

AHMED HASSAN received the B.Sc. and M.Sc. degrees in electrical engineering from the Military Technical College, Cairo, Egypt, in 2010 and 2017, respectively. He is currently pursuing the Ph.D. degree in systems and automation with the University of Seville, Seville, Spain. His research interests include automotive systems, aerospace, and robotics.

SARA RUIZ-MORENO received the B.S. degree in industrial technologies engineering and the M.S. degree in industrial engineering from the University of Seville, in 2019 and 2021, respectively, where she is currently pursuing the Ph.D. degree in automatic control, robotics, and telematics. In 2019, she obtained a collaboration grant with GAMCO, SL, to apply machine learning techniques in the Self Growing Learning Knowledge (SEKAS) Project. Her research interests include control, machine learning, and fault detection and diagnosis.

JOSE RAMON D. FREJO received the Ph.D. degree in automatic, robotic, and telematics from the University of Seville, in 2015. He was an Assistant Professor with University Loyola Andalusia, from 2014 to 2015; a MSCA-IF Research Fellow with the Delft University of Technology, from 2017 to 2019; an Assistant Professor with the University of Seville, from 2009 to 2010, in 2014, and from 2015 to 2017; and a Juan de la Cierva Postdoctoral Researcher, from 2019 to 2022. He was a Visiting Scholar with the University of California at Berkeley, Imperial College of London, and the Technical University of Crete. He currently holds a tenured track professor position with the University of Seville. His research interests include model predictive control, freeway traffic control, machine learning, and solar energy systems. He was a recipient of the Outstanding Doctorate Award from the University of Seville and the First Prize of the IEEE ITS Best Dissertation Award in 2016.

JOSE M. MAESTRE received the Ph.D. degree from the University of Seville. He held various positions with universities, such as TU Delft, University of Pavia, University of Kyoto, and Tokyo Institute of Technology. He is currently a Full Professor with the University of Seville. He has published over 200 journal and conference papers, co-edited several books, and led several research projects. His research interests include the control of distributed cyber-physical systems, with a special emphasis on the integration of heterogeneous agents in the control loop. Finally, his achievements have been recognized through several awards and honors, including the Spanish Royal Academy of Engineering's Medal for his contributions to the predictive control of large-scale systems.



EDUARDO FERNÁNDEZ CAMACHO (Life Fellow, IEEE) received the Ph.D. degree in electrical engineering from the University of Seville, Seville, Spain, in 1977. He is currently a Full Professor with the Department of System Engineering and Automatic Control, University of Seville. He has authored several books, including *Model Predictive Control in the Process Industry* (Springer, 1995), *Advanced Control of Solar Plants* (Springer, 1997), *Model Predictive Control* (Springer, 1999 and 2004, second edition), *Control e Instrumentación de Procesos Químicos* (Editorial Sintesis, 1997), *Control of Dead-Time Processes* (Springer, 2007), and *Control of Solar Systems Energy System* (Springer, 2011). He was a member of the IEEE CSS Board of Governors. He is also a member of the IFAC Council. He is a fellow of IFAC. He was awarded the Advanced Grant by the European Research Council in 2018. He chaired the IFAC Publication Committee and the IFAC Policy Committee. He was the President of the European Union Control Association and chaired the IEEE Control Systems Society International Affairs Committee.