



Article **Data-Driven Enhancements for MPC-Based Path Tracking Controller in Autonomous Vehicles**

Jianhua Guo^{1,*}, Zhihao Xie¹, Ming Liu², Jincheng Hu³, Zhiyuan Dai¹ and Jinqiu Guo¹

- 1 National Key Laboratory of Automotive Chassis Integration and Bionics, Changchun 130022, China; jiezh22@mails.jlu.edu.cn (Z.X.); daizy22@mails.jlu.edu.cn (Z.D.); jqguo23@mails.jlu.edu.cn (J.G.)
- 2 School of Automotive Studies, Tongii University, Shanghai 201804, China; 2110215@tongji.edu.cn 3

Department of Aeronautical and Automotive Engineering, Loughborough University,

- Loughborough LE11 3TU, UK; j.hu2@lboro.ac.uk
- Correspondence: xhhmail@jlu.edu.cn

Abstract: The accuracy of the control model is essential for the effectiveness of model-based control methods. However, factors such as model simplification, parameter variations, and environmental noise can introduce inaccuracies in vehicle state descriptions, thereby compromising the precision of path tracking. This study introduces data-driven enhancements for an MPC-based path tracking controller in autonomous vehicles (DD-PTC). The approach consists of two parts: firstly, Kolmogorov-Arnold Networks (KANs) are utilized to estimate tire lateral forces and correct tire cornering stiffness, thereby establishing a dynamic predictive model. Secondly, Gaussian Process Regression (GPR) is deployed to accurately capture the unmodeled dynamics of the vehicle to form a comprehensive control model. This enhanced model allows for precise path tracking through steering control. The superiority of DD-PTC is confirmed through extensive testing on the Simulink-CarSim simulation platform, where it consistently surpasses normal MPC and Linear Quadratic Regulator (LQR) strategies, especially in minimizing lateral distance errors under challenging driving conditions.

Keywords: data-driven; model predictive control; path tracking

check for updates

Citation: Guo, J.; Xie, Z.; Liu, M.; Hu, J.; Dai, Z.; Guo, J. Data-Driven Enhancements for MPC-Based Path Tracking Controller in Autonomous Vehicles. Sensors 2024, 24, 7657. https://doi.org/10.3390/s24237657

Academic Editor: Alex Alexandridis

Received: 8 October 2024 Revised: 1 November 2024 Accepted: 5 November 2024 Published: 29 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/).

1. Introduction

Driven by rapid advancements in technology, autonomous vehicles have emerged as a significant focus of research, primarily due to their vast commercial potential and broad spectrum of applications. An autonomous driving system mainly consists of core modules such as perception, decision-making, planning, and tracking [1]. Among these, path tracking is especially critical as it directly influences the vehicle's ability to precisely adhere to the planned path. Accurate path tracking is not only crucial for driving safety but also serves as an important indicator for evaluating the overall reliability of autonomous driving technologies [2]. Therefore, in-depth research and optimization of path tracking algorithms are crucial for enhancing the navigational accuracy and safety of autonomous vehicles. In recent years, a considerable amount of research has been conducted on path tracking technology for autonomous vehicles.

Currently, diverse algorithms are utilized in the design and optimization of path tracking controllers. These include proportional-integral-derivative control (PID), fuzzy logic control (FL), sliding mode control (SMC), linear quadratic regulator (LQR), robust control, and model predictive control (MPC) [3]. Compared to other controllers such as PID, LQR, and FL, MPC can more effectively handle constraints on state variables, control variables, and their rates of change, thereby achieving optimal control. Unlike SMC, MPC avoids potentially unsafe chattering phenomena, ensuring the smoothness of control actions [3].

In the design of path tracking controllers, the predictive model of MPC is a core component, essential for achieving high precision and stability in control. Kinematic singletrack models, employed by researchers for path tracking applications [4–7], simplify the dynamics by assuming zero tire slip angles at both the front and rear tires and disregarding the effects of inertia. The primary benefits of this model are its minimal computational requirements and its effectiveness at lower speeds. However, the model's accuracy diminishes with increased vehicle speed, limiting its effectiveness in high-speed scenarios. Dynamics single-track models, adopted by some researchers for path tracking [5,8-10], assume that the vehicle consistently maintains a constant speed and the tires remain within the linear region. Under these conditions, the lateral tire force maintains a linear relationship with the slip angle. At high speeds and with small steering angles, an MPC-based controller using the dynamics single-track model generally maintains good path tracking accuracy. When the tires exhibit nonlinear characteristics, the linear assumption of cornering stiffness leads to increased model error, which in turn severely affects tracking accuracy. Kabzan et al. [11] developed a hybrid model by combining the single-track kinematic model and the single-track dynamic model using an adjustable scaling factor. This approach improved the model's adaptability but still faced the previously mentioned challenges. To more accurately capture real vehicle dynamics and enhance model precision, several researchers [12–15] have turned to the dynamics double-track model for path tracking, recognizing it as an effective approach. This model accounts for the forces exerted by each tire and can precisely simulate the effects of lateral load transfer, along with the impact of suspension springs and dampers, significantly enhancing the accuracy of path tracking. However, the complexity of the dynamics double-track model may restrict its widespread adoption in practical applications.

The aforementioned predictive models do not account for model errors resulting from factors such as model simplification, parameter variations, and environmental noise. This leads to inaccuracies in describing vehicle states, thereby affecting the accuracy of tracking. Researchers have developed various strategies to improve the accuracy and adaptability of models beyond the linear region. Several approaches [16–20] directly estimate tire cornering stiffness or precisely utilize tire force observers to estimate lateral forces and effectively adjust tire cornering stiffness to accommodate the nonlinear characteristics of the tires, thus significantly enhancing model performance and ensuring the accuracy of path tracking. In several studies [21–25], learning-based MPC strategies were proposed. These strategies integrate a simplified mechanistic model with residual models that are learned through data-driven methods. Employing an MPC framework, it effectively minimizes path tracking errors and has been validated for its effectiveness and practicality under the conditions of high speed and high acceleration. These advanced methods highlight the critical need to account for both nonlinear characteristics and unmodeled dynamics in dynamics models. By accurately estimating tire cornering stiffness or implementing learning-based MPC, these studies significantly improve the adaptability and predictive accuracy of the models. Despite individual advancements in enhancing path tracking through tire cornering stiffness estimation and data-driven approaches, integrated strategies that combine these methods are still relatively underexplored in the current research.

Inspired by the methods discussed above, this study introduces a data-driven enhancements for an MPC-based path tracking controller architecture that aims to improve path tracking accuracy through precise modeling of the predictive model. As shown in Figure 1, the architecture includes three main components: the state estimation layer, the augmented predictive model layer, and the MPC solver layer. Firstly, we utilize the KAN network [26] to estimate the lateral forces on the front and rear axles of the vehicle, which are employed to correct the vehicle dynamics model. Based on this, GPR is employed as the residual model to describe the deviation between the simplified mechanistic model and the actual vehicle state. Finally, the MPC-based controller is employed to solve for the optimal control solutions. The main contributions of this paper include the following:

 In response to the challenge that the relationship between the lateral force and slip angle becomes nonlinear as the tires exhibit nonlinear characteristics, this study employs the KAN network to estimate the lateral forces to correct the tire cornering stiffness, thereby enhancing the accuracy of the predictive model.

• GPR is utilized to learn the residual dynamics between the simplified mechanistic model and the actual vehicle state. This strategy significantly enhances the predictive model accuracy of the MPC and further improves the precision of path tracking.

The remainder of this paper is organized as follows: Section 2 introduces the establishment of the vehicle prediction model; Section 3 elaborates on the data-driven enhancements for the MPC-based path tracking controller; Section 4 presents the simulation results to demonstrate the effectiveness of the strategy; and Section 5 discusses the conclusions.



Figure 1. Framework of DD-PTC.

2. Nominal Vehicle Dynamics Model

When autonomous vehicles follow a path, the primary emphasis is placed on assessing the vehicle's lateral dynamics performance. The effects of air resistance and load transfer between the front and rear axles on tire characteristics are ignored. A three-degree-offreedom vehicle dynamics model encompassing longitudinal, lateral, and yaw motion is established, as shown in Figure 2.



Figure 2. Dynamic model.

The vehicle dynamics equations are as follows:

$$\begin{cases} m\ddot{x} = m\dot{y}\dot{\phi} + 2F_{xf} + 2F_{xr} \\ m\ddot{y} = -m\dot{x}\dot{\phi} + 2F_{yf} + 2F_{yr} \\ I_{z}\ddot{\phi} = 2l_{f}F_{yf} - 2l_{r}F_{yr} \end{cases}$$
(1)

where *m* represents the mass of the vehicle, and l_f and l_r denote the distances from the vehicle's center of mass to the front and rear axles, respectively. In the vehicle coordinate system, *x* and *y* represent the longitudinal and lateral positions, \dot{x} and \dot{y} represent the longitudinal and lateral positions, \dot{x} and \dot{y} represent the longitudinal and lateral speeds, and \ddot{x} and \ddot{y} represent the longitudinal and lateral accelerations. φ , $\dot{\varphi}$, and $\ddot{\varphi}$ correspond to the vehicle's yaw angle, yaw rate, and yaw acceleration, respectively. I_z denotes the vehicle's moment of inertia. F_{xf} and F_{yf} represent the decomposition of the front tire forces into lateral and longitudinal components, while F_{xr} and F_{yr} represent the decomposition of the rear tire forces into lateral and longitudinal components, respectively, as follows:

$$\begin{cases} F_{xf} = F_{lf} \cos \delta_f - F_{cf} \sin \delta_f \\ F_{xr} = F_{lr} \\ F_{yf} = F_{lf} \sin \delta_f + F_{cf} \cos \delta_f \\ F_{yr} = F_{cr} \end{cases}$$

where F_{lf} and F_{lr} represent the longitudinal forces of the front and rear wheels, respectively, while F_{cl} and F_{cr} represent the lateral forces of the front and rear wheels, respectively. Considering the slip angle and slip rate are within a small range near the origin, the lateral and longitudinal forces of the tires can be simplified into a linear relationship, as follows:

$$\begin{cases} F_l = C_l s \\ F_{cf} = C_{cf} \alpha_f = C_{cf} \left(\frac{l_f \dot{\varphi} + \dot{y}}{\dot{x}} - \delta_f \right) \\ F_{cr} = C_{cr} \alpha_r = C_{cr} \frac{\dot{y} - l_r \dot{\varphi}}{\dot{x}} \end{cases}$$
(2)

where C_l represents the longitudinal stiffness of the tires, and $C_{c,i}$ with $i \in \{f, r\}$ represents the cornering stiffness of the tires, respectively. α_i with $i \in \{f, r\}$ represents the tire slip angle, μ denotes the ground adhesion coefficient, and s indicates the tire slip rate.

Considering the transformation relationship between the coordinate system of the vehicle and the inertial coordinate system, as shown in Figure 3, the coordinate transformation formula can be expressed as follows:

$$\begin{cases} \dot{Y} = \dot{x}\sin\varphi + \dot{y}\cos\varphi \\ \dot{X} = \dot{x}\cos\varphi - \dot{y}\sin\varphi \end{cases}$$
(3)

Finally, the single-track model of vehicle dynamics can be described as follows:

$$\dot{\xi}(t) = f_{dyn}(\xi(t), u(t)) = \begin{cases} \dot{x} \cos \varphi - \dot{y} \sin \varphi \\ \dot{x} \sin \varphi + \dot{y} \cos \varphi \\ \dot{\varphi} \\ \frac{1}{m} \dot{y} \dot{\varphi} + 2(C_{lf} s_f + C_{lr} s_r + C_{cf} \alpha_f \delta_f) \\ \frac{1}{m} (-m \dot{x} \dot{\varphi} + 2C_{cf} \alpha_f + 2C_{cr} \alpha_r) \\ \frac{2}{I_z} (a C_{cf} \alpha_f - b C_{cr} \alpha_r) \end{cases}$$
(4)

In Equation (4), the state variables and control variable are as follows:

$$\begin{cases} \boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{X} & \boldsymbol{Y} & \dot{\boldsymbol{x}} & \dot{\boldsymbol{y}} & \boldsymbol{\varphi} & \dot{\boldsymbol{\varphi}} \end{bmatrix}^T \\ \boldsymbol{u} = \delta_f \end{cases}$$
(5)

To meet computational requirements, the model described above must be linearized. The linearized equations can be rewritten as follows:

$$\dot{\xi} = f(\xi(t), u(t)) \tag{6}$$

$$\dot{\xi} = A\xi + Bu \tag{7}$$

where

$$A = \frac{\partial f}{\partial x}|_{\xi_{t},u_{t}} = \begin{bmatrix} -\frac{2(C_{cf}+C_{cr})}{m\dot{x}_{t}} & \frac{\partial f_{\dot{y}}}{\partial \dot{x}} & 0 & -\dot{x}_{t} + \frac{2(l_{t}C_{cr}-l_{f}C_{cf})}{m\dot{x}_{t}} & 0 & 0\\ \dot{\varphi} - \frac{2C_{cf}\delta_{f,t-1}}{m\dot{x}_{t}} & \frac{\partial f_{\dot{x}}}{\partial \dot{x}} & 0 & \dot{y}_{t} - \frac{2l_{f}C_{cf}\delta_{f,t-1}}{m\dot{x}_{t}} & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0\\ \frac{2(l_{t}C_{cr}-l_{f}C_{cf})}{l_{z}\dot{x}_{t}} & \frac{\partial f_{\dot{\varphi}}}{\partial \dot{x}} & 0 & -\frac{2(l_{f}^{2}C_{cf}+l_{r}^{2}C_{cr})}{l_{z}\dot{x}_{t}} & 0 & 0\\ \cos(\varphi_{t}) & \sin(\varphi_{t}) & \dot{x}_{t}\cos(\varphi_{t}) - \dot{y}_{t}\sin(\varphi_{t}) & 0 & 0 & 0\\ -\sin(\varphi_{t}) & \cos(\varphi_{t}) & -\dot{y}_{t}\cos(\varphi_{t}) - \dot{x}_{t}\sin(\varphi_{t}) & 0 & 0 & 0 \end{bmatrix}$$
(8)

$$B = \frac{\partial f}{\partial u}|_{\xi_t, u_t} = \left[\frac{2C_{cf}}{m}, \frac{2C_{cf}\left(2\delta_{f, t-1} - \frac{\dot{y}_t + l_f \dot{\phi}_t}{\dot{x}_t}\right)}{m}, 0, \frac{2l_f C_{cf}}{I_z}, 0, 0\right]$$
(9)

$$\frac{\partial f_{\dot{y}}}{\partial \dot{x}} = \left(2C_{cf}\left(\dot{y}_t + l_f\dot{\phi}_t\right) + 2C_{cr}(\dot{y}_t - l_r\dot{\phi}_t)\right) / m\dot{x}_t^2 - \dot{\phi}_t \tag{10}$$

$$\frac{\partial f_{\dot{\phi}}}{\partial \dot{x}} = \left(2l_f C_{cf} \left(\dot{y}_t + l_f \dot{\phi}_t\right) - 2l_r C_{cr} (\dot{y}_t - l_r \dot{\phi}_t)\right) / l_z \dot{x}_t^2 \tag{11}$$



Figure 3. Coordinate transform.

3. DD-PTC Architecture

3.1. Augmented Predictive Model Layer

3.1.1. Gaussian Process Regression

Inspired by [21,22], we employ GPR to augment the nominal dynamics model in the MPC framework. A GPR ensemble predicts dynamics model errors and makes adjustments at each time step. Similar to most GP-based learning problems, we assume a true vehicle dynamics model, f_{true} . We obtain \tilde{y}_{k+1} by measuring at discrete time points t_k , where the measurements are subject to noise.

$$\tilde{y}_{k+1} = f_{true}(x_k, u_k) + w_k \tag{12}$$

$$\kappa(z_i, z_j) = \sigma_f^2 exp\left(-\frac{1}{2}(z_i - z_j)^T L^{-2}(z_i - z_j)\right) + \sigma_n^2 \tag{13}$$

where *L* represents the diagonal length scale matrix, σ_f and σ_n denote the data variance and the prior noise variance, respectively, while z_i and z_j represent data features.

The vehicle system dynamics is redefined as a combination of the normal dynamics plus the mean posterior μ_{gp} from the GPR, and μ_{gp} is determined by the selection matrix B_d and the input feature vector z_k .

$$f_{cor}(x_k, u_k) = f_{dyn}(x_k, u_k) + B_d \mu_{gp}(z_k)$$
(14)

In this paper, the input to the GPR is selected as $z_k = \begin{bmatrix} \dot{y} & \varphi & \dot{\varphi} \end{bmatrix}^T$, with $B_d \mu(z_k)$ representing the residual compensation for the state variables. This paper focuses on correcting $\begin{bmatrix} \dot{y} & \varphi & \dot{\varphi} \end{bmatrix}$. Given the training feature samples *Z* and the query feature sample z_k , the mean and covariance predicted by GPR can be derived as follows:

$$\mu_{gp}(z_k) = K_k^T K^{-1} Z \tag{15}$$

$$\Sigma_{uk} = K_{kk} - K_k^T K^{-1} K_k \tag{16}$$

where K, K_k , and K_{kk} , respectively, represent:

$$K = \kappa(Z, Z) + \sigma_n^2 I \tag{17}$$

$$K_k = \kappa(Z, Z_k) \tag{18}$$

$$K_{kk} = \kappa(Z_k, Z_k) \tag{19}$$

3.1.2. Data Collection and Practical Implementation

1

To fit the GPR ensemble, data are collected in Carsim using the nominal dynamics model. For each sampling moment t_k , the actual state variables at the next time step x_{k+1}^{true} , the state variables predicted by the prediction model x_{k+1} , and the time step δt_k are recorded. The error in the state variables of the vehicle per unit time is then calculated as follows:

$$x_{ek} = \frac{x_{k+1}^{true} - x_{k+1}}{\delta t_k}$$
(20)

During the fitting process of GPR, outlier processing is initially conducted on the collected data. Subsequently, to reduce the fitting errors of GPR, we use a Gaussian Mixture Model (GMM) to cluster the data, and a separate GPR is fitted for each category to form a GPR ensemble. Better fitting results are achieved by appropriately selecting the hyperparameters of the kernel function. During this process, we utilize the maximum likelihood estimation method to optimize the hyperparameters of the kernel function, ensuring that the predictive model can accurately reflect the actual vehicle dynamics. In practical applications, by calculating the Euclidean distance between the current vehicle state and each cluster center, the GPR model from the closest category is chosen from the GPR ensemble to correct state variable errors in MPC. This method effectively improves prediction accuracy and optimizes control performance.

3.2. State Estimation Layer

The tire lateral force calculated based on constant cornering stiffness only provides good approximations under stable conditions. When there are drastic changes in the vehicle's vertical load, significant discrepancies arise between the calculated tire lateral force using constant stiffness and the actual lateral force. Therefore, fixed stiffness values C_{cf} and C_{cr} struggle to adapt to scenarios with significant changes in vertical load. To enhance path tracking performance in autonomous vehicles, a lateral force observer based on the KAN network has been designed. This observer can precisely estimate the tire's lateral force, and then the cornering stiffness is dynamically adjusted.

The KAN network is based on the Kolmogorov–Arnold representation theorem, which proves that any multivariate function can be represented by a sum of single-variable functions. Compared to Multilayer Perceptrons (MLPs) [27], KAN networks offer several advantages. Firstly, KAN networks have better interpretability and exhibit faster neural scaling laws than MLPs. By leveraging the locality of splines, KAN networks can avoid the catastrophic forgetting issue, which is a severe problem observed in MLPs. Secondly, KAN networks achieve comparable performance to MLPs with fewer parameters and exhibit good generalization capabilities [26].

Assuming the structure of the KAN is $[n_0, n_1, \dots, n_L]$, where n_i represents the number of nodes in the i^{th} layer of the computational graph, we denote i^{th} the neuron in the l^{th} layer as (l, i), and its activation value as $x_{l,i}$. The activation function connecting (l, i) and (l, j + 1) is denoted as follows:

$$\phi_{l,j,i}, l = 0, \cdots, L - 1, i = 1, \cdots, n_l, j = 1, \cdots, n_{l+1}.$$
 (21)

Given the pre-activation value of $x_{l,i}$, the post-activation value $\tilde{x}_{l,j,i}$ is obtained through the activation function $\phi_{l,i,i}$. The activation value (l + 1, j) is the sum of all $\tilde{x}_{l,i,i}$.

$$x_{l+1,j} = \sum_{i=1}^{n_l} \tilde{x}_{l,j,i} = \sum_{i=1}^{n_l} \phi_{l,j,i}(x_{l,i}), i = 1, \cdots, n_l, j = 1, \cdots, n_{l+1}.$$
 (22)

The matrix representation of the above Equation (22) is given by the following:

$$\mathbf{x}_{l+1} = \underbrace{\begin{pmatrix} \phi_{l,1,1}(\cdot) & \phi_{l,1,2}(\cdot) & \cdots & \phi_{l,1,n_{l}}(\cdot) \\ \phi_{l,2,1}(\cdot) & \phi_{l,2,2}(\cdot) & \cdots & \phi_{l,2,n_{l}}(\cdot) \\ \vdots & \vdots & & \vdots \\ \phi_{l,n_{l+1},1}(\cdot) & \phi_{l,n_{l+1},2}(\cdot) & \cdots & \phi_{l,n_{l+1},n_{l}}(\cdot) \end{pmatrix}}_{\Phi_{l}} \mathbf{x}_{l}$$
(23)

Given an input vector $x_0 \in R^{n_0}$, the output of the KAN network is denoted by the following:

$$KAN(x) = (\Phi_{L-1} \circ \Phi_{L-2} \circ \dots \circ \Phi_1 \circ \Phi_0)x \tag{24}$$

where Φ_l is the function matrix corresponding to the l^{th} KAN layer. The selected input features for the KAN network are $z_k = [vx vy yaw_rate ax ay delta]$, and the output of the KAN network is as follows:

$$[\hat{F}_{cf}, \hat{F}_{cr}] = KAN(\mathbf{z}_k) \tag{25}$$

where \hat{F}_{cf} and \hat{F}_{cr} are the estimated lateral forces. The tire cornering stiffness is adjusted based on the lateral tire force estimated by the KAN network. The tire cornering stiffness correction coefficient is defined as follows:

$$\begin{cases} \lambda_f = \frac{\hat{F}_{cf} - F_{cf}}{|\hat{F}_{cf}|} = \frac{\hat{F}_{cf} - C_{cf} \left(\frac{l_f \phi + y}{\hat{x}} - \delta_f\right)}{\hat{F}_{cf}} \\ \lambda_r = \frac{\hat{F}_{cr} - F_{cr}}{|\hat{F}_{cr}|} = \frac{\hat{F}_{cr} - C_{cr} \left(\frac{y - l_r \phi}{\hat{x}}\right)}{\hat{F}_{cr}} \end{cases}$$
(26)

The corrected tire cornering stiffness is given by the following equation:

$$\begin{cases} \hat{C}_{cf} = (1 + \lambda_f)C_{cf} \\ \hat{C}_{cr} = (1 + \lambda_r)C_{cr} \end{cases}$$
(27)

To avoid the failure of tire lateral force estimation and the deterioration of the path tracking controller's performance due to excessive correction by the correction factor, the correction coefficients for the cornering stiffness of the front and rear axles are constrained as follows:

$$\begin{cases} \lambda_{f\min} \le \lambda_{f} \le \lambda_{f\max} \\ \lambda_{r\min} \le \lambda_{r} \le \lambda_{r\max} \end{cases}$$
(28)

where $\lambda_{f \max}$ and $\lambda_{f \min}$ are the upper and lower limits of the front axle cornering stiffness correction coefficient, while $\lambda_{r \max}$ and $\lambda_{r \min}$ are the upper and lower limits for the rear axle cornering stiffness correction coefficient. The lower and upper limits are set as $\lambda_{f \min} = \lambda_{r \min} = -0.4$ and $\lambda_{f \max} = \lambda_{r \max} = 0.4$, respectively. The cornering stiffness in the path tracking controller is replaced with the corrected cornering stiffnesses \hat{C}_{cf} and \hat{C}_{cr} . To prevent singularities in the correction factor when the tire lateral force is too low, the correction coefficient is set to zero if the tire slip angle's absolute value is less than 0.1° .

3.3. MPC Solver Layer

MPC controls a system subject to its dynamics $\dot{x} = f(x, u)$ by minimizing a cost function L(x, u) based on reference outputs $\eta_{ref}(t)$ and reference input controls $u^*(t)$. The general form of MPC is as follows:

$$\min_{u} \int L(x, u)
s.t.\dot{x} = f_{dyn}(x, u)
x_0 = x_{init}
r(x, u) = 0
h(x, u) \le 0$$
(29)

where x_0 is the initial state, and r and h represent the equality and inequality constraints, respectively. $\dot{x} = f_{dyn}(x, u)$ can be further written in the form of state space equations $\dot{x} = A_{dis}x + B_{dis}u$, and the output is $\eta = C_{dis}x + D_{dis}u$. The normal model introduced in Section 2 is employed here. The cost function and state space equations above are expressed in a continuous form. For practical applications, it is necessary to discretize the equations. Assuming the discretization time step is T, the output in this study is solely dependent on x. The forward Euler method is used for discretization to obtain the following:

$$\begin{cases} x(k+1) = A_k x(k) + B_k u(k) \\ \eta(k) = C_k x(k) \end{cases}$$
(30)

where $A_k = I + A_{dis}T$, $B_k = B_{dis}T$, $C_k = C_{dis}$, and T = 0.01. To prevent sudden changes in the control variables that could cause danger during the driving process, it is necessary to limit the control increments. Thus, the augmented state vector is defined as follows:

$$\chi(k) = \begin{bmatrix} \tilde{\xi}(k) & \tilde{u}(k-1) \end{bmatrix}^T$$
(31)

The new expression for the state space equations is obtained as follows:

$$\chi(k+1) = A\chi(k) + B\Delta u(k) \tag{32}$$

$$y(k) = C\chi(k) \tag{33}$$

9 of 18

where $A = \begin{bmatrix} A_k & B_k \\ 0_{m \times n} & I_m \end{bmatrix}$, $B = \begin{bmatrix} B_k \\ I_m \end{bmatrix}$, $C = \begin{bmatrix} C_k & 0_{m \times 1} \end{bmatrix}$, *n* is the number of state variables, and *m* is the number of control variables. The state prediction incorporating GPR-augmented dynamics can be expressed as follows:

$$\chi(k+1|k) = A\chi(k) + B\Delta u(k|k) + \mu_{gp}(\xi(k))$$

$$\chi(k+2|k) = A^{2}\chi(k) + AB\Delta u(k|k) + B\Delta u(k+1|k) + \mu_{gp}(\xi(k+1|k))$$

$$\chi(k+3|k) = A^{3}\chi(k) + A^{2}B\Delta u(k|k) + AB\Delta u(k+1|k) + B\Delta u(k+2|k) + \mu_{gp}(\xi(k+2|k))$$

$$\vdots$$

$$\chi(k+N_{p}|k) = A^{N_{p}}\chi(k) + A^{N_{p}-1}B\Delta u(k|k) + A^{N_{p}-2}B\Delta u(k+1|k)$$

$$+\dots + A^{N_{p}-N_{c}}B\Delta u(k+N_{c}-1|k) + \mu_{gp}(\xi(k+N_{p}-1|k))$$

(34)

where μ_{gp} is the mean of the state variable errors predicted by GPR, N_p is the prediction horizon of the system, N_c is the control horizon, and Δu represents the control increments. The output of the system at future time steps is given by the following:

$$\eta(k+1|k) = C\chi(k+1|k)$$

$$\eta(k+2|k) = C\chi(k+2|k)$$

$$\eta(k+3|k) = C\chi(k+3|k)$$

$$\vdots$$

$$\eta(k+N_p|k) = C\chi(k+N_p|k)$$

(35)

The primary goal of the cost function is to penalize deviations between the output and the reference target, and to minimize the rate of change in control actions as much as possible, thereby ensuring passenger comfort. Additionally, to prevent situations where no solution is feasible, a slack variable ε is introduced. In this study, we specify the cost function in a quadratic form.

$$L(x,u) = \sum_{i=1}^{N_p} ||\eta(k+i|t) - \eta_{ref}(k+i|t)||_Q^2 + \sum_{i=0}^{N_c-1} ||\Delta u(k+i|k)||_R^2 + \rho \varepsilon^2$$
(36)

The first summation term penalizes the difference between the expected vehicle state and the state within the prediction horizon of N_p steps. The second summation term penalizes the control increments over the control horizon of N_c steps. Q and R are the weight matrices. In addition, a set of constraints is applied to the control actions.

$$u_{\min} \le u(k) \le u_{max} \tag{37}$$

The sequence of control actions and the sequence of control increments satisfy the following relationship:

$$U(k) = M\Delta U(k) + \Gamma u(k-1)$$
(38)

where $U(k) = \begin{bmatrix} u(k|k) \\ u(k+1|k) \\ \vdots \\ u(k+N_c-1|k) \end{bmatrix}$, $M = \begin{bmatrix} I_m & 0 & \cdots & 0 \\ I_m & I_m & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ I_m & I_m & I_m \end{bmatrix}$, $\Gamma = \begin{bmatrix} I_m \\ I_m \\ \vdots \\ I_m \end{bmatrix}$, $\Delta U(k) = \begin{bmatrix} \Delta u(k|k) \\ \Delta u(k+1|k) \\ \vdots \\ \Delta u(k+N_c-1|k) \end{bmatrix}$.

$$U_{\min} \le U(k) \le U_{\max} \tag{39}$$

The constraints for the front wheel steering angle and the increment of the front wheel steering angle are set as follows:

$$\begin{cases} -30 \le u \le 30\\ -0.47 \le \Delta u \le 0.47 \end{cases}$$
(40)

In summary, the constraints can be represented by the following equation:

$$\begin{bmatrix} M \\ -M \\ I \\ -I \end{bmatrix} \Delta U(k) \leq \begin{bmatrix} U_{\max} - \Gamma u(k-1) \\ -U_{\min} + \Gamma u(k-1) \\ \Delta U_{\max} \\ -\Delta U_{\min} \end{bmatrix}$$
(41)

Finally, the solution is obtained using a nonlinear programming (NLP) solver.

4. Simulation and Analysis

To validate the effectiveness and robustness of the proposed method, we selected two typical scenarios for path tracking verification, as illustrated in Figure 4. These scenarios correspond to double-lane change and single-lane change maneuvers. Both scenarios were simulated using the Simulink-CarSim joint simulation platform. The parameters for the vehicle in the CarSim simulation model are detailed in Table 1. To demonstrate the superiority of DD-PTC, path tracking controllers based on MPC and LQR were developed as benchmarks.



Figure 4. The traffic scenes of two cases.

Table 1. Vehicle parameters.

Parameter	Value	Parameter	Value
Vehicle mass (kg)	1270	Yaw moment of inertia (kg·m ²)	1536.7
Wheelbase (m)	2.910	Wheel track (mm)	1.675 (front, rear)
Distance from front axle to center of mass (m)	1.015	Distance from rear axle to center of mass (m)	1.895
Centroid height (m)	0.54	Rolling radius (m)	0.325
Front axle cornering stiffness (N/rad)	60,000	Rear axle cornering stiffness (N/rad)	40,000

4.1. Estimation Results of the KAN Model

As discussed in Section 3, the lateral force was estimated using the KAN network. Data were gathered via Carsim across a variety of driving scenarios and conditions. Subsequently, these data were employed to train the KAN network, which is structured as follows: the input layer consists of six neurons, corresponding to the primary variables influencing the outputs. This is followed by a hidden layer containing 32 neurons, which allows for the extraction and processing of important features from the input data. The architecture culminates in an output layer of two neurons. These output neurons specifically represent the lateral forces at the front and rear axles, respectively. The KAN network, implemented using PyTorch, incorporates learnable activation functions to enhance model adaptability and performance. The network was trained for 500 epochs with a learning rate set at 0.001. For the dataset distribution, 80% was allocated for training purposes, while the remaining 20% was employed for testing. To further refine the model parameters during the later stages of training, the ExponentialLR learning rate scheduler from PyTorch was employed. This scheduler decreases the learning rate by a certain proportion every epoch. The decay factor of the scheduler was set to 0.99. Additionally, to address the dimensional discrepancies between features and labels, we employed the Z-score normalization method. This technique standardizes the data to a normal distribution, achieving a mean of 0 and a standard deviation of 1 for each feature. As depicted in Figure 5, the training curve of the KAN network demonstrated efficient learning, characterized by a steady decrease in MSE (Mean Squared Error) loss and no indications of overfitting. This suggests that the model generalizes well to new data.



Figure 5. Learning curve.

In the testing phase, we assessed the performance of the KAN network by comparing the estimated values of lateral forces at the front and rear axles with their corresponding actual values. The representative results from this evaluation are illustrated in Figure 6. The experimental results demonstrate that the KAN network has high predictive accuracy on the test dataset, accurately predicting the lateral forces of the front and rear axles of the vehicle.



Figure 6. Prediction vs. ground truth.

4.2. Double-Lane Change

The double-lane change maneuver is a critical scenario for assessing a vehicle's ability to safely execute overtaking maneuvers, as illustrated in Figure 4, Case 1. We set the vehicle's speed at 72 km/h and specified a road surface adhesion coefficient of 0.8. The design parameters for DD-PTC and MPC were identical, with a prediction horizon of $N_p = 35$, a control horizon of Nc = 15, weights for the output variables $Q = \begin{bmatrix} 2000 & 12000 \end{bmatrix}^T$, and weights for the control increments R = 5000. The weights of the MPC were set to be the same as those of the DD-PTC, and the preview time for LQR was 0.1 s. The lateral force estimation and comparative results of path tracking performance across the three controllers are presented below.

Figure 7a,b demonstrate that the KAN network exhibited excellent adaptability in estimating lateral forces. The KAN network effectively understood the characteristics of input data and accurately estimated the lateral forces of the front and rear axles of the vehicle. The root mean square errors (RMSEs) of the lateral force estimation for the front and rear axles were 190.1904 N and 136.3055 N, and the maximum estimation errors were 1.093×10^3 N and 466.854 N, respectively. The average absolute errors of the lateral force estimations were 94.5349 N for the front axle and 62.6035 N for the rear axle. Although these average errors were very small, the maximum values were relatively large. This discrepancy may be attributed to the limited data collected under extreme driving conditions, leading to insufficient training of the model. As a result, the network may struggle to accurately estimate lateral forces, so it is necessary to employ more data to thoroughly train the KAN network to enhance its robustness and accuracy under various driving conditions, which will be a focus of future work. Figure 7c demonstrates that all three methods are capable of tracking the reference path. However, in the scenario involving paths with slightly large curvatures, the LQR method exhibited significant lateral errors and slower convergence compared to the other methods. MPC, by predicting future states with its predictive model, converged over more quickly but still exhibited substantial lateral errors due to limitations in its predictive model. In contrast, DD-PTC, which utilizes a more accurate predictive model, demonstrated smaller lateral errors. This distinction underscores the importance of model accuracy in reducing prediction error and enhancing overall system performance. Figure 7d–f and Table 2 provide a detailed comparison of tracking errors for the three controllers. LDE (Lateral Distance Error) and HAE (Heading Angle Error) are key performance metrics. Employing DD-PTC for path tracking significantly improved the performance: the maximum lateral error decreased by 19.83% and 35.02% compared to MPC and LQR, respectively. Additionally, the average lateral error displacement was reduced by 29.56% and 54.51%, respectively. Regarding the heading angle error, the maximum and average values for DD-PTC were 2.0818° and 0.3564°, respectively, which are roughly equivalent to those recorded for MPC and LQR, indicating that both remained at a comparably low level.



Figure 7. Vehicle simulation results of double-lane change. (a) Lateral force of front axle; (b) lateral force of rear axle; (c) result of path tracking; (d) wheel angle; (e) lateral distance error; and (f) heading angle error.

Table 2. Double-lane change test.

	LDE (m) Max/Avg.	HAE (°) Max/Avg.
DD-PTC	0.0885/0.0162	2.0818/0.3564
LQR	0.1362/0.0357	2.0139/0.4071
MPC	0.1104/0.0230	1.4532/0.3080

4.3. Single-Lane Change

To further validate the robustness of the proposed algorithm, a lane changing simulation experiment was conducted, as illustrated in Figure 4, Case 2. The vehicle speed was fixed at 72 km/h, with a road surface adhesion coefficient of 0.8. The controller parameters were the same as those used for the double-lane change setup. The lateral force estimation and comparative results of path tracking performance across the three controllers are presented below.

Figure 8a,b demonstrate that the KAN network generally provided accurate estimates of the lateral forces on the front and rear axles of the vehicle. Most of the estimation results were close to the actual values, with only a few showing some deviations. The RMSE of the lateral force predictions for the front and rear axles were 406.6039 N and 234.9228 N, and the maximum prediction errors were 1.731×10^3 N and 1.009×10^3 N, respectively. The average absolute errors of the predicted lateral forces for the front and rear axles were 252.2186 N and 145.5211 N, respectively. Notably, in order to reduce the computational load, the KAN network estimated lateral forces without requiring historical data, considering only the current state of the vehicle. Despite this simplification, the network still achieved results that are relatively satisfactory. However, some estimates still exhibited errors, indicating potential for further optimization of the KAN network. Figure 8c-f and Table 3 display the path tracking performance of the three controllers, all demonstrating the ability to track the reference path. In the single-lane change scenario, both MPC and LQR exhibited significant lateral errors. LQR converged more slowly than MPC and DD-PTC, but due to its preview mechanism and slower convergence rate, its HAE was smaller; however, its LDE was significantly greater than those of MPC and DD-PTC. In comparison, DD-PTC performed the best among the three controllers, significantly enhancing path tracking precision, particularly in terms of lateral error. When using DD-PTC for path tracking, the maximum lateral error decreased by 32.64% and 46.30% compared to MPC and LQR, respectively, and the average lateral error decreased by 12.57% and 36.40%. Regarding HAE, the maximum and average values for DD-PTC were 7.1496° and 1.4074° , respectively, which represents a reduction of 21.60% and 21.92% compared to MPC.



Table 3. Single-lane change test.

Figure 8. Cont.





Figure 8. Vehicle simulation results of double-lane change. (a) Lateral force of front axle; (b) lateral force of rear axle; (c) result of path tracking; (d) wheel angle; (e) lateral distance error; and (f) heading angle error.

4.4. Ablation Experiment

To comprehensively demonstrate the effectiveness of our proposed method, we performed ablation experiments in both single-lane and double-lane change scenarios. Our method was compared with MPC using only cornering stiffness correction, GP-MPC, and standard MPC. The comparison results are presented below.

In Figure 9, GP-MPC refers to an MPC approach where only Gaussian Process Compensation was applied to the prediction equation, while Stiff Corr–MPC represents an MPC method with cornering stiffness correction. The results indicate that under both single-lane and double-lane change conditions, the DD-PTC method demonstrated strong performance. Compared to standard MPC, both Stiff Corr–MPC and GP-MPC performed better. The performance of both Stiff Corr–MPC and GP-MPC is presented in Table 4. In the double-lane change scenario, the mean lateral error of DD-PTC was approximately 16.60% lower than that of Stiff Corr–MPC and 9.07% lower than that of GP-MPC, with the maximum error reduced by around 16.67% and 0.65%, respectively. In the single-lane change scenario, although the mean lateral error of DD-PTC was approximately 10.10% higher than that of Stiff Corr–MPC, the maximum error in this scenario was reduced by approximately 13.23% compared to Stiff Corr–MPC. Additionally, the mean lateral error of DD-PTC was 5.58% lower than that of GP-MPC, with the maximum error reduced by 15.52% compared to GP-MPC.



Figure 9. Vehicle simulation results of double-lane change. (a) Path tracking results for DLC; (b) wheel angle for DLC; (c) lateral distance error for DLC; (d) heading angle error for DLC; (e) path tracking results for SIL; (f) wheel angle for SIL; (g) lateral distance error for SIL; and (h) heading angle error for SIL.

	Test Scenarios	LDE (m) Max/Avg.	HAE (°) Max/Avg.
Stiff Corr-MPC	Double-lane	0.1062/0.0194	2.1643/0.3705
	Single-lane	0.1181/0.0297	5.8537/1.1589
GP-MPC	Double-lane	0.0891/0.0178	1.6996/0.3610
	Single-lane	0.1213/0.0350	8.8269/1.7213

Table 4. Ablation study results of path tracking performance in single-lane and double-lane change scenarios.

5. Discussion and Conclusions

This paper developed and validated data-driven enhancements for the MPC-based path tracking controller, which integrates cornering stiffness corrections and Gaussian Process Regression as residual dynamics to improve both the adaptability of the prediction model and the accuracy of path tracking. Specifically, the strategy utilizes the KAN network to estimate the tire lateral forces for dynamically adjusting cornering stiffness, and combined Gaussian Process Regression to correct discrepancies between the simplified mechanistic model and the actual vehicle state. Our predictive model accurately captures and responds to dynamic changes in the vehicle. Consequently, this enhances the precision of path tracking. In comparative tests conducted on the Simulink-Carsim joint simulation platform, our DD-PTC method demonstrated superior performance in high adhesion double-lane change and single-lane change scenarios, outperforming normal MPC and LQR methods.

Although DD-PTC has demonstrated significant potential, there remains considerable room for optimization. The robustness and precision of the KAN network in estimating lateral forces require further enhancement. Additionally, the computational speed of DD-PTC presents opportunities for optimization. Future research will focus on enhancing the continuous learning capabilities of the KAN network and improving the computational efficiency of DD-PTC. These efforts aim to enable quicker controller responses to changes in complex environments, comprehensively enhancing the controller's performance and adaptability.

Author Contributions: Conceptualization and methodology, J.G. (Jinqiu Guo) and Z.X.; software, M.L. and J.H.; validation, Z.X. and Z.D.; writing—original draft preparation, Z.X.; writing—review and editing, J.G. (Jinqiu Guo); Supervision, J.G. (Jianhua Guo); funding acquisition, J.G. (Jianhua Guo). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

References

- 1. Teng, S.; Hu, X.; Deng, P.; Li, B.; Li, Y.; Ai, Y.; Yang, D.; Li, L.; Xuanyuan, Z.; Zhu, F.; et al. Motion planning for autonomous driving: The state of the art and future perspectives. *IEEE Trans. Intell. Veh.* **2023**, *8*, 3692–3711. [CrossRef]
- Rokonuzzaman, M.; Mohajer, N.; Nahavandi, S.; Mohamed, S. Review and performance evaluation of path tracking controllers of autonomous vehicles. *IET Intell. Transp. Syst.* 2021, 15, 646–670. [CrossRef]
- Stano, P.; Montanaro, U.; Tavernini, D.; Tufo, M.; Fiengo, G.; Novella, L.; Sorniotti, A. Model predictive path tracking control for automated road vehicles: A review. *Annu. Rev. Control* 2023, 55, 194–236. [CrossRef]
- Kong, J.; Pfeiffer, M.; Schildbach, G.; Borrelli, F. Kinematic and dynamic vehicle models for autonomous driving control design. In Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, Republic of Korea, 28 June–1 July 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1094–1099.

- Taherian, S.; Montanaro, U.; Dixit, S.; Fallah, S. Integrated trajectory planning and torque vectoring for autonomous emergency collision avoidance. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 2714–2721.
- 6. Law, C.K.; Dalal, D.; Shearrow, S. Robust model predictive control for autonomous vehicles/self driving cars. *arXiv* 2018, arXiv:1805.08551.
- 7. Tang, L.; Yan, F.; Zou, B.; Wang, K.; Lv, C. An improved kinematic model predictive control for high-speed path tracking of autonomous vehicles. *IEEE Access* 2020, *8*, 51400–51413. [CrossRef]
- Vallon, C.; Ercan, Z.; Carvalho, A.; Borrelli, F. A machine learning approach for personalized autonomous lane change initiation and control. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1590–1595.
- 9. Xie, J.; Xu, X.; Wang, F.; Tang, Z.; Chen, L. Coordinated control based path following of distributed drive autonomous electric vehicles with yaw-moment control. *Control Eng. Pract.* **2021**, *106*, 104659. [CrossRef]
- Hang, P.; Xia, X.; Chen, G.; Chen, X. Active safety control of automated electric vehicles at driving limits: A tube-based MPC approach. *IEEE Trans. Transp. Electrif.* 2021, *8*, 1338–1349. [CrossRef]
- 11. Kabzan, J.; Valls, M.I.; Reijgwart, V.J.; Hendrikx, H.F.; Ehmke, C.; Prajapat, M.; Bühler, A.; Gosala, N.; Gupta, M.; Sivanesan, R.; et al. AMZ driverless: The full autonomous racing system. *J. Field Robot.* **2020**, *37*, 1267–1294. [CrossRef]
- 12. Wu, H.; Si, Z.; Li, Z. Trajectory tracking control for four-wheel independent drive intelligent vehicle based on model predictive control. *IEEE Access* 2020, *8*, 73071–73081. [CrossRef]
- Hamid, U.Z.A.; Ariff, M.H.M.; Zamzuri, H.; Saito, Y.; Zakaria, M.A.; Rahman, M.A.A.; Raksincharoensak, P. Piecewise trajectory replanner for highway collision avoidance systems with safe-distance based threat assessment strategy and nonlinear model predictive control. J. Intell. Robot. Syst. 2018, 90, 363–385. [CrossRef]
- 14. Yuan, H.; Sun, X.; Gordon, T. Unified decision-making and control for highway collision avoidance using active front steer and individual wheel torque control. *Veh. Syst. Dyn.* **2018**, *57*, 1188–1205. [CrossRef]
- 15. Subosits, J.K.; Gerdes, J.C. Impacts of model fidelity on trajectory optimization for autonomous vehicles in extreme maneuvers. *IEEE Trans. Intell. Veh.* **2021**, *6*, 546–558. [CrossRef]
- 16. Zhang, Y.; Wang, W.; Yang, C.; Ma, M. An adaptive model predictive control strategy for path following of autonomous vehicles based on tire cornering stiffness estimation. In Proceedings of the 2021 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, 22–24 May 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1904–1909.
- 17. Xie, H.; Xi, Y.; Song, K.; Guo, F. A Trajectory Tracking Control Algorithm of Self-driving Tractor-trailer Based on Online Estimation of Cornering Stiffness. In Proceedings of the 2022 41st Chinese Control Conference (CCC), Hefei, China, 25–27 July 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 5414–5419.
- Wang, Y.; Duan, Y.; Zhang, Y. Local Path Planning and Tracking Control Considering Tire Cornering Stiffness Uncertainty; Technical Report, SAE Technical Paper; SAE International: Warrendale, PA, USA, 2021.
- Chu, Z.; Chen, N.; Zhang, N.; Li, G. Path-tracking for autonomous vehicles with on-line estimation of axle cornering stiffnesses. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 6651–6657.
- 20. Zhang, Z.; Zheng, L.; Li, Y.; Li, S.; Liang, Y. Cooperative strategy of trajectory tracking and stability control for 4WID autonomous vehicles under extreme conditions. *IEEE Trans. Veh. Technol.* **2023**, *72*, 3105–3118. [CrossRef]
- Hewing, L.; Kabzan, J.; Zeilinger, M.N. Cautious model predictive control using gaussian process regression. *IEEE Trans. Control Syst. Technol.* 2019, 28, 2736–2743. [CrossRef]
- Kabzan, J.; Hewing, L.; Liniger, A.; Zeilinger, M.N. Learning-based model predictive control for autonomous racing. *IEEE Robot. Autom. Lett.* 2019, 4, 3363–3370. [CrossRef]
- 23. Saviolo, A.; Li, G.; Loianno, G. Physics-inspired temporal learning of quadrotor dynamics for accurate model predictive trajectory tracking. *IEEE Robot. Autom. Lett.* 2022, *7*, 10256–10263. [CrossRef]
- Torrente, G.; Kaufmann, E.; Föhn, P.; Scaramuzza, D. Data-driven MPC for quadrotors. *IEEE Robot. Autom. Lett.* 2021, 6, 3769–3776. [CrossRef]
- Salzmann, T.; Kaufmann, E.; Arrizabalaga, J.; Pavone, M.; Scaramuzza, D.; Ryll, M. Real-time neural MPC: Deep learning model predictive control for quadrotors and agile robotic platforms. *IEEE Robot. Autom. Lett.* 2023, *8*, 2397–2404. [CrossRef]
- 26. Liu, Z.; Wang, Y.; Vaidya, S.; Ruehle, F.; Halverson, J.; Soljačić, M.; Hou, T.Y.; Tegmark, M. Kan: Kolmogorov-arnold networks. *arXiv* 2024, arXiv:2404.19756.
- 27. Pinkus, A. Approximation theory of the MLP model in neural networks. Acta Numer. 1999, 8, 143–195. [CrossRef]

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