sensors

MDPI

Article

Learning-Based Control of Autonomous Vehicles Using an Adaptive Neuro-Fuzzy Inference System and the Linear Matrix Inequality Approach

Mohammad Sheikhsamad and Vicenç Puig *

Institute of Robotics and Industrial Informatics (CSIC-UPC), Llorens i Artigas, 4-6, 08028 Barcelona, Spain; mohammad.sheikhsamad@upc.edu
* Correspondence: vicenc.puig@upc.edu

Abstract: This paper proposes a learning-based control approach for autonomous vehicles. An explicit Takagi–Sugeno (TS) controller is learned using input and output data from a preexisting controller, employing the Adaptive Neuro-Fuzzy Inference System (ANFIS) algorithm. At the same time, the vehicle model is identified in the TS model form for closed-loop stability assessment using Lyapunov theory and LMIs. The proposed approach is applied to learn the control law from an MPC controller, thus avoiding the use of online optimization. This reduces the computational burden of the control loop and facilitates real-time implementation. Finally, the proposed approach is assessed through simulation using a small-scale autonomous racing car.

Keywords: ANFIS controller; linear matrix inequality; Takagi–Sugeno; autonomous driving

Citation: Sheikhsamad, M.; Puig, V. Learning-Based Control of Autonomous Vehicles Using an Adaptive Neuro-Fuzzy Inference System and the Linear Matrix Inequality Approach. *Sensors* **2024**, *24*, 2551. https://doi.org/10.3390/s24082551

Academic Editors: Agapito Ledezma Espino and Araceli Sanchis de Miguel

Received: 23 February 2024
Revised: 8 April 2024
Accepted: 12 April 2024
Published: 16 April 2024

1. Introduction

Autonomous driving is one of the top ten technologies that will change the lives of citizens, according to the European Parliament Research Service (EPRS) [1]. The National Highway Traffic Safety Administration (NHTSA) currently states in its technical report that 94% of road accidents are the consequence of human error [2]. By 2050, it is projected that 68% of the world's population will live in urban regions, up from 55% in 2018 [3], increasing traffic congestion. Autonomous driving emerges as a solution to these challenges, and it facilitates the following [4]:

- Attaining close to zero traffic accidents.
- Improving accessibility for people with low physical mobility.
- Lessening congestion through shared routes for both passengers and goods, coupled with intelligent motion.
- Lowering energy consumption and pollution [1].

Consequently, industrialized countries are actively engaged in a competitive race to develop autonomous driving technology, and leading research institutions and companies are achieving great success. Recent progress in software (artificial intelligence, planning and control, telecommunications, etc.), hardware (sensors, embedded supercomputers, etc.), laws, and user acceptance suggests that autonomous driving is just a matter of time, although achieving full autonomy presents many challenges.

To address these challenges, the Society of Automotive Engineers (SAE) outlines five progressive levels of automation [5], ranging from driver assistance (level 1) to full autonomy (level 5). In levels 2 through 5, the autonomous vehicle driving system is in charge of steering, braking, and accelerating. This autonomous driving system consists of multiple components that necessitate seamless integration to operate as a cohesive unit. These components include perception, motion planning, vehicle localization, pedestrian detection, traffic-sign detection, road-marking detection, automated parking, vehicle cyber

security, fault diagnosis, and automatic control. Automatic control is in charge of driving the vehicle between two points as well as generating smooth control actions to ensure a comfortable ride [6] by controlling the lateral and/or longitudinal dynamics. This is a challenging task that puts the vehicle into motion using sensors (GPS, IMU, encoders, cameras, LIDAR, etc.) to measure the environment and the vehicle variables, and then provides the appropriate signals to the actuators (steering motor, electric engine, and braking system).

To address the problem of automatic control in autonomous driving systems, a variety of automatic control strategies have been developed and implemented, e.g., proportional–integral–derivative (PID) control [7–9], robust control (H_∞) [10], fuzzy logic control [7,11,12], sliding-mode control (SMC) [13,14], Lyapunov-based control [14,15], linear parameter-varying (LPV) control [16], Takagi–Sugeno control [17], and linear quadratic regulator (LQR) control [18]. Model predictive control (MPC) stands out as a highly effective control strategy, relying on the dynamic model of the system to anticipate future states. Thus, MPC has the capability to predict upcoming events and take appropriate control actions. It is based on an optimal control law that minimizes a cost function in real time during each iteration. Numerous notable efforts have been undertaken in this field [19–22]. Since the vehicle model is non-linear, non-linear MPC (NL-MPC) can be applied but real-time implementation is still an issue because of the small sampling times used [23–26].

The linear parameter-varying (LPV) approach mentioned in [27] is a control strategy utilized in many control applications (see [16] for a recent review). It enables the transformation of a non-linear system into a linear-like representation by embedding the system's non-linearities inside variable parameters. LPV-MPC [28,29] uses LPV to obtain the vehicle model, and despite its merits, the control still needs online optimization for MPC during iteration, but with less computational load than NL-MPC. Another versatile and effective tool in automatic control is the Linear Matrix Inequalities (LMIs) approach [30], which allows for systematic design of closed-loop systems with guarantees of stability and performance without requiring online optimization [31–33].

The Adaptive Neuro-Fuzzy Inference System (ANFIS) [34] is a learning approach that synergizes the capabilities of two soft computing frameworks: Artificial Neural Networks (ANNs) and fuzzy logic (FL). The ANFIS can model complex, non-linear functions that may not be easily described using physical mathematical equations. This model can also be represented explicitly and interpreted in Takagi–Sugeno (TS) form. Several studies have used ANFISs in various applications. Ref. [35] applied an ANFIS to enhance vehicle route selection in uncertain conditions. Ref. [36] aimed to build an ANFIS driver model that could replace a real one. Ref. [37] used an ANFIS for navigation and target acquisition for an autonomous robot in both static and dynamic environments. Ref. [38] applied an ANFIS to establish a systematic process to access the complex operations of working vehicles. Ref. [39] proposed an ANFIS to design a controller with self-position azimuth correction (SPAC) for trajectory tracking and obstacle avoidance. Ref. [40] presented an ANFIS-based approach to mobile robot navigation and obstacle avoidance in unknown static environments, considering obstacle distances and steering angles as the ANFIS input and output, respectively.

The TS fuzzy modeling approach, as proposed by Takagi and Sugeno [41], serves as a universal approximator for any smooth non-linear system [42]. It provides a systematic approach for generating fuzzy rules from input and output data, and it is able to represent the local dynamics of each fuzzy rule through a linear system model. This enables the representation of a large family of non-linear dynamical systems with a high degree of precision. Using the TS approach, several notable developments are actively moving forward [17,43,44]. One common application is the use of the TS model to represent the *vehicle model* for control tasks [45,46]. Ref. [47] proposed TS MPC for motion planning. However, in both cases, the controller/planner still needs to perform online optimization to minimize the MPC cost function during each iteration using the vehicle TS model. To the best of the authors' knowledge, there is a lack of evidence supporting the direct application of TS to learn the control law (*control model*) with stability guarantees.

This paper proposes a learning-based controller for autonomous vehicles. We use the ANFIS as a learning method to directly obtain control laws from data (*control model*) in TS model form. Another TS model is obtained to represent vehicle behavior (*vehicle model*). The application of TS representation for the vehicle model has already been considered for control design in the literature. However, this paper uses this model to verify the closed-loop stability of the learned TS controller in an innovative manner using Lyapunov theory and LMIs. A reliable working controller (in this paper, an MPC controller) is used as a data generator to provide the required input/output data for obtaining a TS model for the controller using the ANFIS. The proposed approach is validated through simulation with a small-scale autonomous race car. The obtained results show that the proposed approach has the merit of reducing computational complexity by removing online optimization.

This paper is organized as follows. Section 2 outlines the proposed approach. Section 3 provides details on the learning-based control design and introduces the autonomous vehicle considered as a case study. The simulation results are presented in Section 4. Finally, Section 5 summarizes the key findings of this paper and suggests potential paths for future research.

2. Proposed Approach

The core concept behind the proposed approach is to develop a controller for an autonomous vehicle by relying on machine learning and data rather than conventional model-based control strategies, which mostly rely on physical models. When employing the ANFIS as a machine learning method, the resulting controller is referred to as the ANFIS controller in the remainder of this paper. It is used as a feedback controller to provide appropriate control actions (controller output) to carry out the planned motion and correct tracking errors (controller input). Tracking errors are generated during the execution of a planned motion. Hence, the term "data" refers to the input and output values of the controller, and the objective is to design a controller based only on these data. Various methods exist to generate data, including simulation (using different control strategies) or conducting real-world experiments (application of various control actions to the vehicle). In this work, as shown in Figure 1, an existing MPC controller is used as a data generator. This means that there already exists an autonomous vehicle system that works with an MPC controller (previously designed and validated in [47]). This MPC controller functions effectively, and throughout its operation, the input and output data are recorded. By using these data to train an ANFIS structure, we can obtain an ANFIS controller. In other words, this ANFIS controller is intended as a potential substitute for the MPC controller, and a comparison of the operations and parameters of these two controllers is investigated. Furthermore, the stability of the new autonomous vehicle system using the ANFIS controller is examined. It should be noted that for stability assessment, both the control and vehicle models must be derived from two distinct TS representations.

The proposed approach comprises the following steps:

To design such a machine learning-based control strategy and ensure closed-loop stability, the following procedure is briefly outlined:

- Step 1: Generate the input and output data.
- Step 2: Employ the ANFIS to learn the control law from the data.
- Step 3: Validate the learned controller through simulation.
- Step 4: Obtain the TS model of the control model and vehicle model.
- Step 5: Stability proof of the closed-loop system.

2.1. Generate the Input and Output Data

The initial step consists of acquiring the data. Regardless of the data generation method used, the quality of the data directly influences the effectiveness of the controller. In this paper, an MPC controller functions as a data-generating tool. The controller inputs are the tracking errors measured during the operation of the autonomous vehicle using the MPC controller, $x_c = [v_e \quad y_e \quad \theta_e]$, which are the errors in the longitudinal speed, lateral speed, and angular velocity of the vehicle, respectively. The controller outputs are the

control actions of the MPC controller, $u = [\delta \quad a]^T$, which correspond to the steering wheel angle and acceleration applied to the autonomous vehicle, respectively.

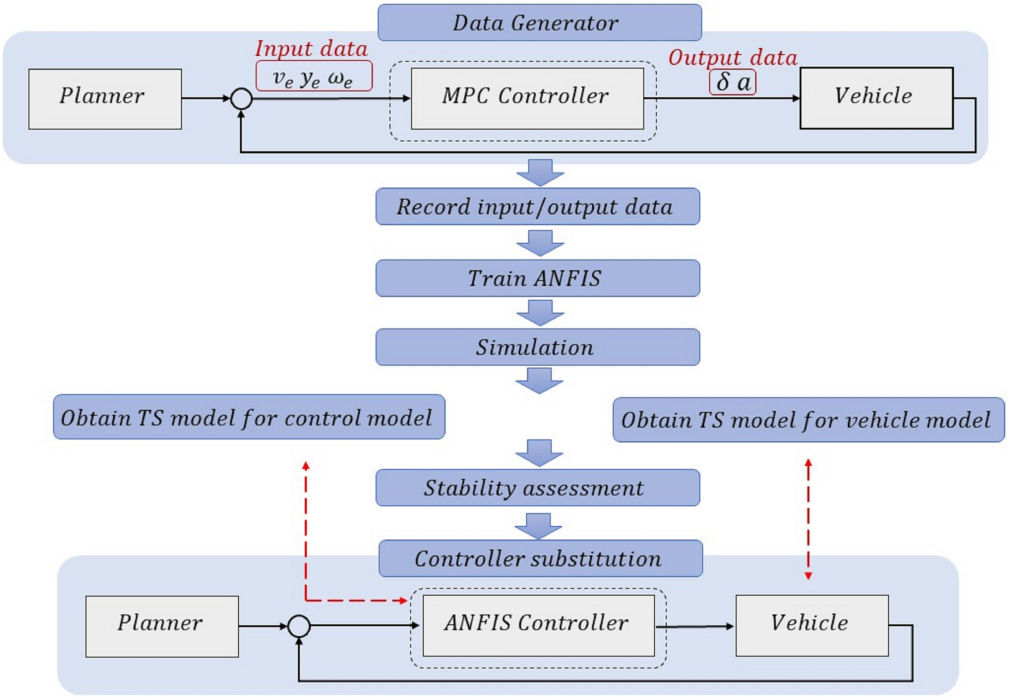

Figure 1. Introduction of the main idea and proposed approach.

2.2. Employ ANFIS to Learn the Control Law from the Data

The ANFIS is used to learn a control law from the input and output data. This modeling tool configures a neural network that learns the dynamic behavior of the vehicle using the backpropagation technique and the least squares (RLS) method for adjusting additional parameters. The ANFIS enables the generation of an interpretable law in the form of a TS model, providing sets of linear parameters (consequent parameters), nonlinear parameters (premise parameters), and membership functions (MFs). Given that the vehicle controller has two outputs, it is divided into two multi-input single-output (MISO) subsystems for the application of the ANFIS.

2.3. Validate the Learned Controller through Simulation

The closed-loop validation of autonomous vehicles is based on the conformity between the outcomes of the ANFIS controller and those attained by the MPC controller. The precise alignment during the simulation serves as a validation of the proposed approach. However, compared to the MPC controller, the proposed approach has the merit of removing the necessity of online optimization, thereby reducing computational complexity. Stability and performance are assessed in the next step.

2.4. Obtain the Takagi–Sugeno Model of the Control Model and Vehicle Model

After obtaining, learning, and validating the controller, our objective is to derive the explicit formula for the control law. The procedure is based on performing some inverse steps that the ANFIS internally performs. While the algorithm efficiently computes the consequent and premise parameters, we build two polytopic TS state-space representations for the control model (see Figure 2a) and the vehicle model (see Figure 2b). The vehicle model uses its related ANFIS structure, which is essential for evaluating the stability of the vehicle closed-loop system using the ANFIS controller.

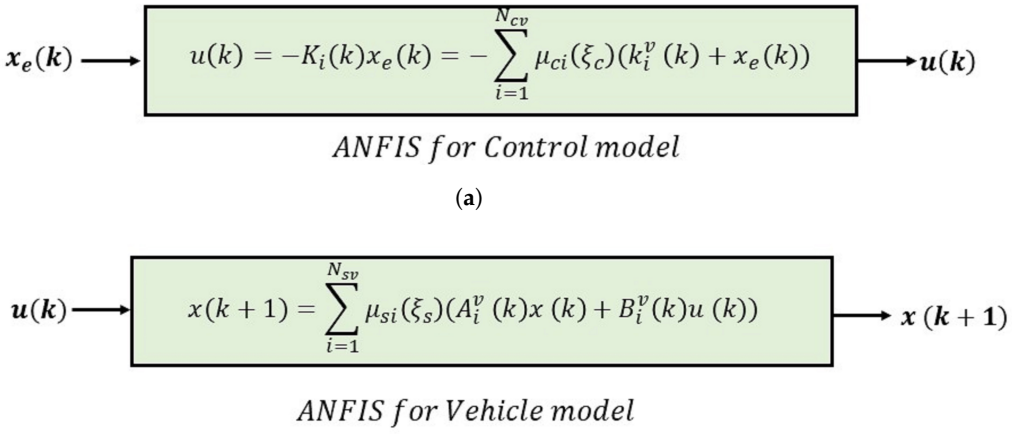

Figure 2. Illustrations of the ANFIS Takagi–Sugeno (TS) representations for (**a**) the control model and (**b**) the vehicle model. A detailed explanation of the parameters is provided in Section 3.

2.5. Stability Proof of the Closed-Loop System

The final step involves analyzing the stability of the autonomous vehicle closed-loop system using the ANFIS controller. This is achieved by employing all vertices of both TS representations of the control and vehicle models and applying the Lyapunov stability theorem to the closed-loop system using LMIs [25]. This is presented in detail later in Section 3.7.

3. Learning-Based Control Design Description

3.1. Considered Autonomous Vehicle

The case study considered is an autonomous race car, which is a developed platform for autonomous driving. This is a rear-wheel drive (RWD) electric remote control (RC) vehicle (see Figure 3) that has been modified to operate autonomously. Mechanically speaking, it has been equipped with some decks to protect the on-board electronics and sensors [47].

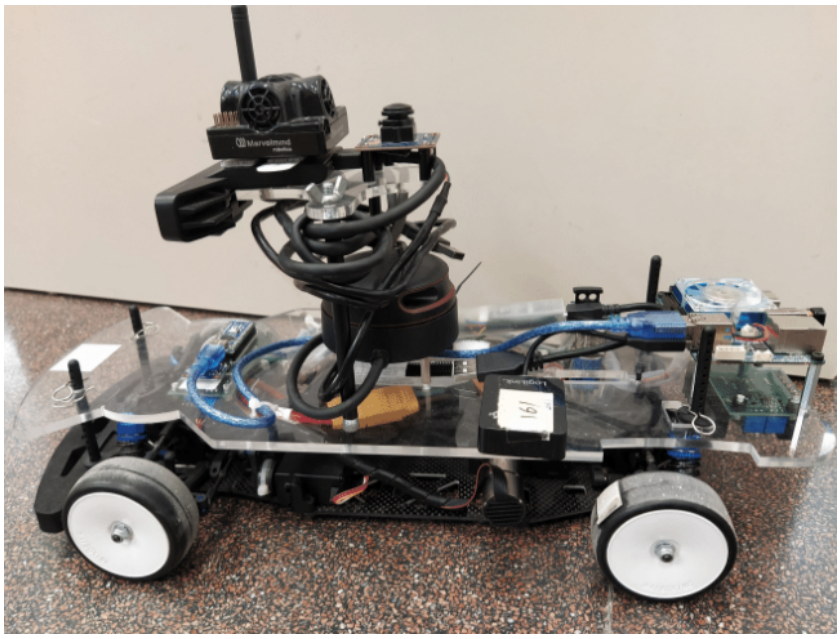

Figure 3. A real picture of the autonomous vehicle used for simulation.

The non-linear model used for simulating the considered autonomous vehicle is based on the bicycle model presented in Figure 4 and introduced in [48]:

$$\begin{aligned}
\dot{v}_x &= a_r + \frac{-F_{yf}\sin\delta - \mu g}{m} + \omega v_y \\
\dot{v}_y &= \frac{F_{yf}\cos\delta + F_{yr}}{m} - \omega v_x \\
\dot{\omega} &= \frac{F_{yf} l_f \cos\delta - F_{yr} l_r}{I} \\
\alpha_f &= \delta - \tan^{-1}\left(\frac{v_y}{v_x} - \frac{l_f \omega}{v_x}\right) \\
\alpha_r &= -\tan^{-1}\left(\frac{v_y}{v_x} + \frac{l_r \omega}{v_x}\right) \\
F_{yf} &= d\sin\left(c\tan^{-1}(b\alpha_f)\right) \\
F_{yr} &= d\sin\left(c\tan^{-1}(b\alpha_r)\right)
\end{aligned} \tag{1}$$

Table 1 lists the system variables. The lateral forces produced in the front and rear tires, denoted F_{yf} f and F_{yr}, are determined using the simplified "Magic Formula" model to simulate lateral tire forces. The parameters b, c, and d in this model shape the force curve and are obtained through an identification procedure. The front and rear sliding angles are represented as α_f and α_r, while m and I represent the mass and inertia of the vehicle. Furthermore, l_f and l_r are the distances from the vehicle center of mass to the front and rear wheel axes, respectively. The static friction coefficient and gravity constant are denoted as μ and g. Table 2 lists the specific values for all dynamic vehicle parameters.

Table 1. List of symbols.

Symbol	Description
v_x	Longitudinal velocity of the vehicle in the center of gravity (CoG) frame (C) in $(\frac{m}{s})$; see Figure 4.
v_y	Lateral velocity of the vehicle in the (CoG) frame (C) in $(\frac{m}{s})$.
ω	Angular velocity of the vehicle in the (CoG) frame (C) in $(\frac{rad}{s})$.
X	Global position of the vehicle in the x-axis frame (O) in (m).
Y	Global position of the vehicle in the y-axis frame (O) in (m).
θ	Orientation of the vehicle with respect to the x-axis of the frame (O) in (rad).
a	Longitudinal acceleration vector on the rear wheels in $(\frac{m}{s^2})$.
δ	Steering angle on the front wheels in (rad).

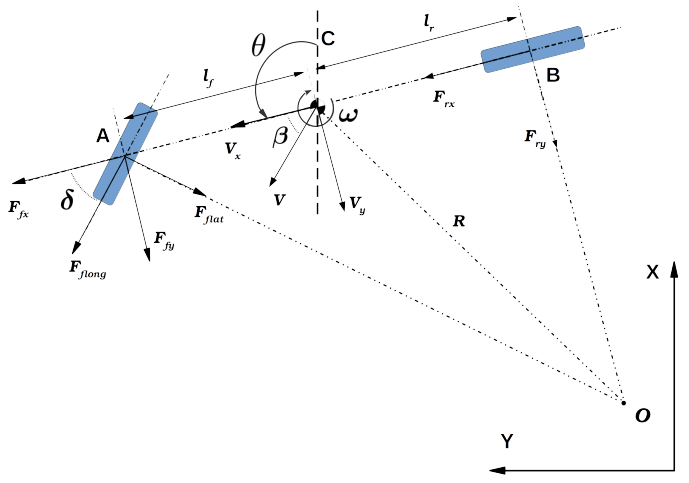

Figure 4. Representation of the bicycle model in 2D space.

Table 2. Model parameters.

Parameter	Value	Parameter	Value
l_f	0.1377 m	C_f	45
l_r	0.1203 m	C_r	45
m	2.424 kg	I	0.02 kg m^2
b	6.0	c	1.6
d	7.76	μ	0.006

3.2. Generate the Input and Output Data

The autonomous vehicle equipped with an MPC controller, as described in [47], is used as the data generator. The MPC problem is formulated as a quadratic optimization problem that is solved at each time k to determine control actions, given the values of $x_s(k)$ and $u(k-1)$:

$$\begin{aligned}
\min_{\Delta U(k)} J(k) = & \sum_{i=0}^{H_p-1} \Bigg(\Big(r(k+i) - x_s(k+i) \Big)^T Q \Big(r(k+i) - x_s(k+i) \Big) \\
& \cdots + \Delta u(k+i)^T R \Delta u(k+i) \Bigg) + x_s(k+H_p)^T P x_s(k+H_p) \\
s.t.: \quad & x_s(k+i+1) = \sum_{j=1}^{N_v} \mu_{sN_j}(\zeta_s(k)) \Big(A_j \hat{x_s}(k+i) + B_j u(k+i) + C_j \Big) \\
& u(k+i) = u(k+i-1) + \Delta u(k+i) \\
& \Delta U(k) \in \Delta \prod \\
& \Delta \prod = \{ \Delta u(k) \mid A_{\Delta u} \Delta u(k) = b_{\triangle u}, \Delta u(k) \geq 0 \} \\
& U(k) \in \prod \\
& \prod = \{ u(k) \mid A_u u(k) = b_u, u(k) \geq 0 \} \\
& x_s(k+H_p) \in \chi \\
& y_e \in \left[\underline{y_e}, \overline{y_e} \right] \\
& x_s(k) = \hat{x_s}(k)
\end{aligned} \tag{2}$$

where ζ_s:= $[v_x \; v_y \; \omega \; \delta \; a]$ is the vector of vehicle scheduling variables, $\hat{x}$ is the estimated state vector, $r = [v_{xr} \quad 0 \quad \omega_r]^T$ is the reference vector provided by the trajectory planner, and H_p is the prediction horizon. The tuning matrices $Q \in \mathbb{R}^{3\times3}$ and $R \in \mathbb{R}^{2\times2}$ are positive definite in order to obtain a convex cost function. Thus, the values of the variables $x_c = [v_{xe} \quad v_{ye} \quad \omega_e]$ and $u = [\delta \quad a]^T$ are recorded as inputs and outputs during the operation of the autonomous vehicle with the MPC controller. These data are employed as training data for the ANFIS in the following step.

3.3. Learn the Control Law from Data Using the ANFIS Algorithm

This section outlines the methodology used to obtain the TS representation of the autonomous vehicle control model (see Figure 5). To achieve this, the ANFIS is utilized to learn the structure from the input and output data. In more detail, it learns the MPC controller behavior of the vehicle from the input and output data using the backpropagation technique and a set of membership functions (MFs). A typical membership function is the generalized Gaussian Bell (GB) function. The ANFIS algorithm can only be used for multi-input single-output (MISO) systems. Thus, the control model, which has two outputs, is split into two MISO subsystems to apply the ANFIS. Since our control model is a second-order system, two subsystems are obtained and two learning procedures are carried out. To do this, first, the polynomial representation of each subsystem is formulated as:

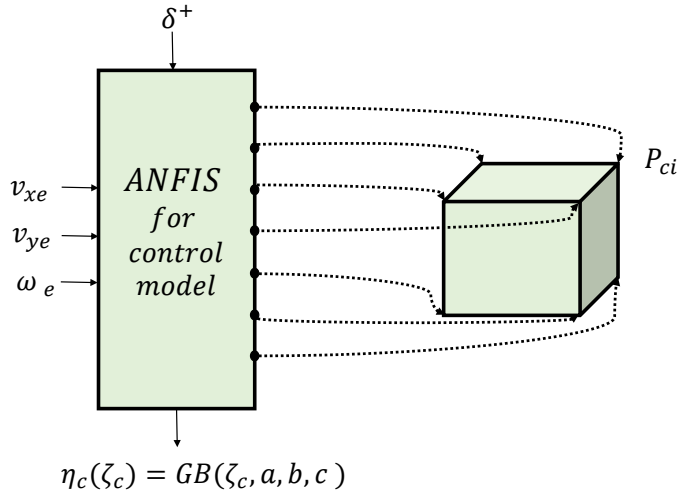

Figure 5. Control model TS representation: TS polytopic learning ANFIS scheme for subsystem δ, using the Gaussian Bell membership function with parameters a, b, and c.

$$\begin{aligned} &P_{ci} = p_{c1i}v_{xe} + p_{c2i}v_{ye} + p_{c3i}\omega_e + p_{c4i} \\ &\forall i = 1, \ldots, N_{cv} \end{aligned} \tag{3}$$

where P_{ci} is a linear polynomial representation of the controller of a subsystem at a particular output configuration. P_{cji}, $\forall j = 1, \ldots, N_{c\zeta}$ represent the consequent parameters obtained from the ANFIS; $N_{c\zeta}$ is the number of scheduling variables; and N_{cv} represents the number of polytopic vertices. v_{xe}, v_{ye}, and ω_e are the trajectory errors that are used as controller inputs. Reorganizing the terms in this equation yields

$$P_{ci} = [p_{c1i} \quad p_{c2i} \quad p_{c3i}]x_c + [p_{c4i}] \tag{4}$$

where $x_c = [v_{x_e} \quad v_{y_e} \quad \omega_e]^T$ is the controller state, and the polynomial structure is transformed into the discrete-time controller representation given by

$$\begin{aligned} &u_i(k) = -\Big(K_i(k)x_e(k) + C_{ci}(k)\Big) \\ &\forall i = 1, \ldots, N_{cv} \end{aligned} \tag{5}$$

where step $u_i(k)$ is the output of subsystem i. K_i and C_{ci} define the so-called vertex systems, with $u = [\delta \quad a]^T$. The generalized Gaussian Bell (GB) membership function, which is defined by three parameters (a, b, and c), is employed as a membership function.

$$\begin{aligned} &\eta_{cm} = \frac{1}{1 + \frac{\zeta_{co} - c_{mo}}{a_{mo}}^{2b_{mo}}} \\ &\forall m = 1, \ldots, N_{cMF}, \forall o = 1, \ldots, N_{c\zeta} \end{aligned} \tag{6}$$

where ζ_c represents the ANFIS input vector of the variables and is referred to as the scheduling variables. Moreover, N_{cMF} and $N_{c\zeta}$, respectively, represent the number of controller MFs per scheduling variable and the number of scheduling variables. In this case, if N_{cMF} is two, the normalized weights η_{cNi} are computed as follows:

$$\begin{aligned} &\mu_{ci}(\zeta_c) = \prod_{j=1}^{N_{c\zeta}} \xi_{cij}(\eta_{c0}, \eta_{c1}) \\ &\forall i = 1, \ldots, N_{cv} \end{aligned} \tag{7}$$

where $\xi_{cij}(0)$ represents any of the weighting functions that depend on each rule i. Then, by applying

$$\mu_{cNi}(\zeta_c) = \frac{\mu_{ci}(\zeta_c)}{\sum_{j=1}^{N_{cv}} \mu_{cj}(\zeta_c)} \quad \forall i = 1, \ldots, N_{cv} \tag{8}$$

the normalized weights are obtained. Each scheduling variable ζ_{co} is known and varies within a defined interval $\zeta_{co} \in [\underline{\zeta_{co}}, \overline{\zeta_{co}}] \in \mathbb{R}$. Finally, the polytopic TS model for each subsystem is represented as:

$$u_j(k) = -\sum_{i=1}^{N_{cv}} \mu_{cNji}(\zeta_c(k)) \left(K_{ji}^v(k) x_e(k) + C_{cji}^v(k) \right) \quad \forall i = 1, ..., N_{cG} \tag{9}$$

where N_{cG} is the number of subsystems. Accordingly, the overall TS system is represented as follows:

$$u_j(k) = -\sum_{i=1}^{N_{cv}} \mu_{cNji}(\zeta_c(k)) \left(\begin{bmatrix} K_{1i}^v(k) \\ K_{2i}^v(k) \end{bmatrix} x_e(k) + \begin{bmatrix} C_{c1i}^v(k) \\ C_{c2i}^v(k) \end{bmatrix} \right) \tag{10}$$

For the sake of clarity, Equation (10) can be expressed as:

$$u(k) = -\sum_{i=1}^{N_{cv}} \mu_{cNi}(\zeta_c(k)) (K_i^v(k) x_e + C_{ci}^v(k)) \tag{11}$$

and the ANFIS controller gains $K \in \mathbb{R}^{2\times 3}$ in Equation (5) are given by

$$K_i(k) = \sum_{i=1}^{N_{cv}} \begin{bmatrix} \mu_{cN1i}(\zeta_c(k)) K_{1i}^v(k) \\ \mu_{cN2i}(\zeta_c(k)) K_{2i}^v(k) \end{bmatrix} \tag{12}$$

3.4. Validate the Learned Controller through Simulation

In this step, the closed-loop validation methodology of the ANFIS controller is presented. It is executed through simulation under conditions identical to those under which the MPC controller operates. This means that the ANFIS controller is replaced with the MPC controller as follows:

$$\begin{aligned} \delta &= evalfis(outFIS\_\delta, [v_{xe} \quad v_{ye} \quad \omega_e]) \\ a &= evalfis(outFIS\_a, [v_{xe} \quad v_{ye} \quad \omega_e]) \end{aligned} \tag{13}$$

where *evalfis* is the Evaluate Fuzzy Inference System function in the MATLAB fuzzy toolbox; $outFIS\_\delta$ and $outFIS\_a$ are the fuzzy inference systems (FISs) to be evaluated, specified as TS fuzzy inference systems, as described in Section 3.3, and $x_e = [v_{xe} \quad v_{ye} \quad \omega_e]^T$ and $u = [\delta \quad a]^T$ are, respectively, the inputs and outputs of the ANFIS controller. The outcomes obtained with the ANFIS controller are expected to closely follow those achieved with the MPC controller.

3.5. TS Representation for Control Model

Once the algorithm discussed in Section 3.3 has computed the consequent and premise parameters for each of the MISO subsystems, we construct the polytopic TS representation for the control model (for each one of the MISO subsystems). This results in an explicit formula for the control law (see Equation (12)). This TS model has to be validated through the conformity of the parameters obtained with those achieved by the ANFIS. Using the same approach, the TS representation of vehicle states (vehicle model) is achieved in the next step.

3.6. TS Representation for Vehicle Model

Obtaining the state space of the vehicle (*vehicle model*) is required for the stability assessment of the system. This can be accomplished through two approaches: the TS approach [47,49] and the linear parameter-varying (LPV) approach [50]. In this work, the vehicle model is represented by the TS model, as shown in Figure 6.

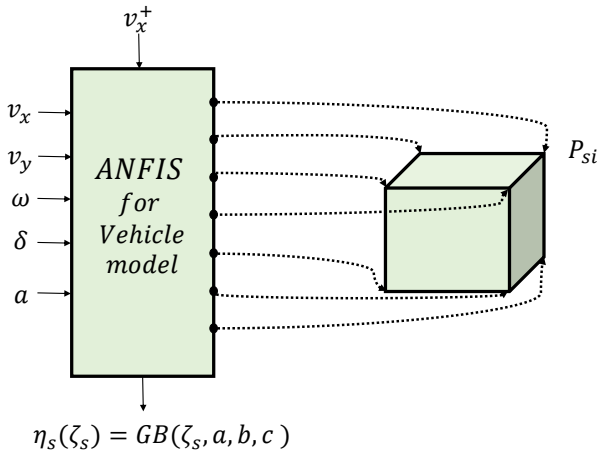

Figure 6. Vehicle model TS representation: TS polytopic learning ANFIS scheme for subsystem v_x, using the Gaussian Bell membership function with parameters a, b, and c [47].

$$\begin{aligned} &P_{si} = p_{s1i}v_x + p_{s2i}v_y + p_{s3i}\omega + p_{s4i}\delta + p_{s5i}a + p_{s6i} \\ &\forall i = 1, \ldots, N_{sv} \end{aligned} \tag{14}$$

where the linear polynomial P_{si} represents the output configuration of the state of the vehicle for a particular subsystem. P_{sji}, $j = 1, \ldots, N_{s\zeta}$ represent the consequent parameters obtained from the ANFIS; $N_{s\zeta}$ is the number of scheduling variables; N_{sv} represents the number of polytopic vertices; and v_x, v_y, and ω are the system states, which, respectively, represent the longitudinal speed, lateral speed, and angular velocity of the vehicle at each time step. Equation (14) can be rewritten as follows:

$$P_{si} = [p_{s1i} \quad p_{s2i} \quad p_{s3i}]x_s + [p_{s4i} \quad p_{s5i}]u + [p_{s6i}] \tag{15}$$

where $x_s = [v_x \quad v_y \quad \omega]^T$ represents the vehicle state with discrete-time representation

$$\begin{aligned} &x_i(k+1) = A_i x(k) + B_i u(k) + C_{si} \\ &\forall i = 1, \ldots, N_{sv} \end{aligned} \tag{16}$$

where A_i, B_i, and C_{si} are vertex systems, and $u = [\delta \quad a]^T$. The generalized Gaussian Bell (GB) membership function is defined by three parameters (a, b, and c), as follows

$$\begin{aligned} &\eta_{sm} = \frac{1}{1 + \frac{\zeta_{so} - c_{mo}}{a_{mo}}^{2b_{mo}}} \\ &\forall m = 1, \ldots, N_{sMF}, \forall o = 1, \ldots, N_{s\zeta} \end{aligned} \tag{17}$$

where ζ_s represents the ANFIS input vector of the variables or scheduling variables and N_{sMF} and $N_{s\zeta}$, respectively, represent the number of membership functions per scheduling variable and the number of scheduling variables. In this case, the normalized weights η_{sNi} are computed as follows:

$$\mu_{si}(\zeta_s) = \prod_{j=1}^{N_{s\zeta}} \xi_{sij}(\eta_{s0}, \eta_{s1}) \quad \forall i = 1, \ldots, N_{sv} \tag{18}$$

where ξ_{sij} stands for any of the weighting functions that depend on each rule i

$$\mu_{sNi}(\zeta_s) = \frac{\mu_{si}(\zeta_s)}{\sum_{j=1}^{N_{sv}} \mu_{sj}(\zeta_s)} \quad \forall i = 1, \ldots, N_{sv} \tag{19}$$

Each scheduling variable ζ_{so} is known and varies within a defined interval $\zeta_{so} \in [\underline{\zeta_{so}}, \overline{\zeta_{so}}] \in \mathbb{R}$. Finally, the polytopic TS model for each subsystem is

$$x_{sj}(k+1) = -\sum_{i=1}^{N_{sv}} \mu_{sNji}(\zeta_s(k)) \left(A_{ji}^v(k)x(k) + B_{ji}^v(k)u(k) + C_{sji}^v(k) \right) \quad \forall i = 1, \ldots, N_{sG} \tag{20}$$

where N_{sG} is the number of subsystems. The overall TS system is represented as:

$$x_{sj}(k+1) = \sum_{i=1}^{N_{sv}} \mu_{sNji}(\zeta_s(k)) \left(\begin{bmatrix} A_{1i}^v(k) \\ A_{2i}^v(k) \\ A_{3i}^v(k) \end{bmatrix} x(k) + \begin{bmatrix} B_{1i}^v(k) \\ B_{2i}^v(k) \\ B_{3i}^v(k) \end{bmatrix} u(k) + \begin{bmatrix} C_{1i}^v(k) \\ C_{2i}^v(k) \\ C_{3i}^v(k) \end{bmatrix} \right) \tag{21}$$

For clarity of presentation, Equation (21) can be expressed as

$$x_s(k+1) = \sum_{i=1}^{N_{sv}} \mu_{sNi}(\zeta_s(k)) \left(A_i^v(k)x(k) + B_i^v(k)u(k) + C_{si}^v(k) \right) \tag{22}$$

and the matrices $A \in \mathbb{R}^{3\times3}$ and $B \in \mathbb{R}^{3\times2}$ are as follows:

$$A_i(k) = \sum_{i=1}^{N_{sv}} \begin{bmatrix} \mu_{sN1i}(\zeta_s(k))A_{1i}^v(k) \\ \mu_{sN2i}(\zeta_s(k))A_{2i}^v(k) \\ \mu_{sN3i}(\zeta_s(k))A_{3i}^v(k) \end{bmatrix} \tag{23}$$

$$B_i(k) = \sum_{i=1}^{N_{sv}} \begin{bmatrix} \mu_{sN1i}(\zeta_s(k))B_{1i}^v(k) \\ \mu_{sN2i}(\zeta_s(k))B_{2i}^v(k) \\ \mu_{sN3i}(\zeta_s(k))B_{3i}^v(k) \end{bmatrix} \tag{24}$$

3.7. Stability Assessment Using LMIs

Finally, the stability condition of the closed-loop system using the ANFIS controller is presented and proved.

Proposition 1. *Consider the following TS closed-loop system $x(k+1) = (A(k) - B(k)K(k))x(k)$ in discrete time, where the controller K is the ANFIS controller in TS form (12), and the vehicle model is also expressed in TS form (see Equations (23) and (24)). Then, according to the Lyapunov stability theorem, the previous TS closed-loop system will be stable if there exists a matrix $P > 0$, $P = P^T \in \mathbb{R}^{n\times n}$, satisfying*

$$\begin{bmatrix} -P & (A_i(k) - B_i(k)K_j(k))^T \\ A_i(k) - B_i(k)K_j(k) & -P^{-1} \end{bmatrix} < 0 \quad \forall i = 1, \ldots, N_{sv}, \forall j = 1, \ldots, N_{cv} \tag{25}$$

Proof. The stability of the autonomous vehicle using the ANFIS controller is assessed through the Lyapunov stability theorem and LMIs by introducing $V(x)$ as a Lyapunov function [25]:

$$\begin{aligned} V(x) &= x^T P x \\ \Delta V \quad &= V\Big(x(k+1)\Big) - V\Big(x(k)\Big) \\ &= x(k)^T \Bigg(\Big(A_i(k) - B_i(k)K_j(k)\Big)^T P \Big(A_i(k) - B_i(k)K_j(k)\Big) \Bigg) x(k) - x(k)^T P x(k) \\ &= x(k)^T \Bigg(\Big(A_i(k) - B_i(k)K_j(k)\Big)^T P P^{-1} P \Big(A_i(k) - B_i(k)K_j(k)\Big) - P \Bigg) x(k) < 0 \end{aligned} \tag{26}$$

Applying the Schur complement to the previous expression yields

$$\Delta V = \sum_{i=1}^{N_{sv}} \mu_{si}(\zeta_s) \sum_{j=1}^{N_{cv}} \mu_{ci}(\zeta_c) \begin{bmatrix} -P & (A_i(k) - B_i(k)K_j(k))^T \\ A_i(k) - B_i(k)K_j(k) & -P^{-1} \end{bmatrix} < 0 \tag{27}$$

where N_{cv} and N_{sv} are, respectively, the number of polytopic vertices employed in the ANFIS structure applied to the controller and vehicle TS models. The membership functions $\mu_{si}(\zeta_s)$ and $\mu_{ci}(\zeta_c)$ are positive between zero and one. In order to guarantee negativity for ΔV, the second term needs to be negative. This leads to the LMI condition (25). □

4. Results

4.1. Data Generation

The proposed approach is versatile and can be applied to any type of controller. In this study, we employ an autonomous vehicle system, with the MPC controller (detailed in Section 3.2) as a data generator. The autonomous vehicle (introduced in Section 3.1) was tested on the Verschueren track. Figure 7 provides a visual representation of its operation in the x- and y-axes. Throughout its operation, both the controller input (trajectory error variables: $x_c = [v_{xe} \quad v_{ye} \quad \omega_e]^T$) and controller output (control action: $u = [\delta \quad a]^T$) were systematically recorded. The diagonal terms for tuning the matrices and input constraints are as follows:

$$\begin{gathered} Q = 0.65[0.4 \quad 10^{-6} \quad 0.6] \quad R = 0.35[0.7 \quad 0.3] \quad H_p = 6 \\ A_u = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \quad A_{\Delta u} = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \quad b_u = \begin{bmatrix} 0.249 \\ 0.249 \\ 4 \\ 1 \end{bmatrix} \quad b_{\Delta u} = \begin{bmatrix} 0.05 \\ 0.05 \\ 0.5 \\ 0.5 \end{bmatrix} \end{gathered} \tag{28}$$

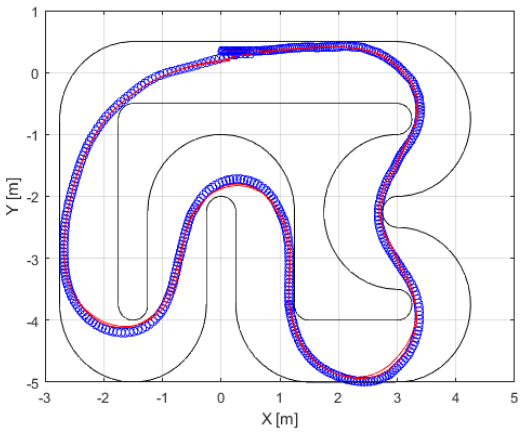

Figure 7. Closed-loopMPC controller simulations, with the positions of the autonomous vehicle system (blue line), trajectory (red line), and track boundaries (black line).

4.2. Learning the Control Law

By using the Neuro-Fuzzy Designer app in MATLAB R2021a, a data split of 20% for testing and 80% for training, the hybrid optimization method, and 100 epochs in the training phase, we obtained the ANFIS structure shown in Figure 8 separately for δ and a. The resulting neuro-fuzzy rules are illustrated in Figures 9 and 10, corresponding to the fuzzy rules for δ and a, respectively. The specifications of the applied ANFIS are also detailed in Table 3.

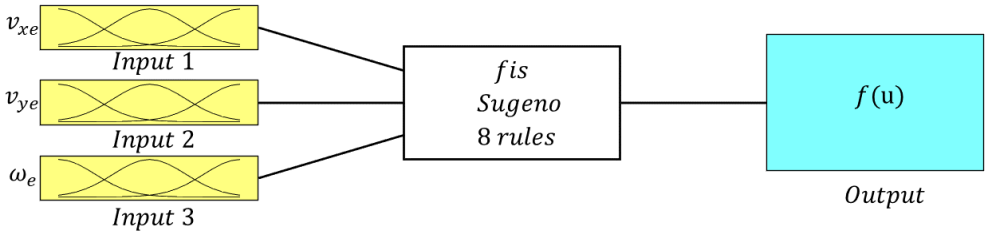

Figure 8. Representation of the ANFIS structure for the controller.

Table 3. Specifications of the ANFIS architecture.

Name	Type
Generate FIS type	Sugeno
The initial FIS model	Grid partition
Decision method for fuzzy logic operation AND (minimum)	Product
Decision method for fuzzy logic operation OR (maximum)	Probabilistic
Output defuzzification method	Weighted average
Number of membership functions for v_{xe}	2
Number of membership functions for v_{ye}	2
Number of membership functions for ω_e	2
Input membership function type	Gaussian Bell
Output membership function type	Constant
Number of rules	8
Train FIS optimization method	Hybrid
Number of epochs	100

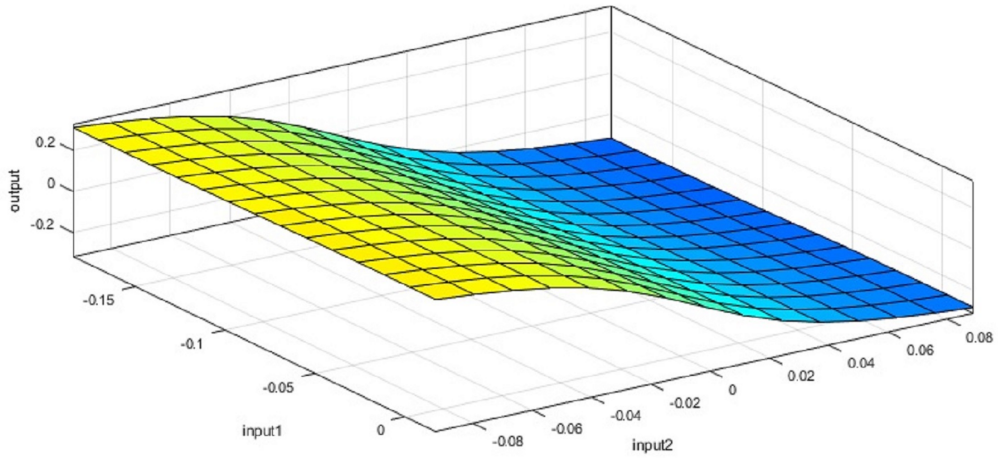

Figure 9. Representation of the fuzzy rules surface for the steering wheel angle (δ). This is one output of the ANFIS controller.The degree of membership is shown in a color contour from blue (lowest) to green (highest).

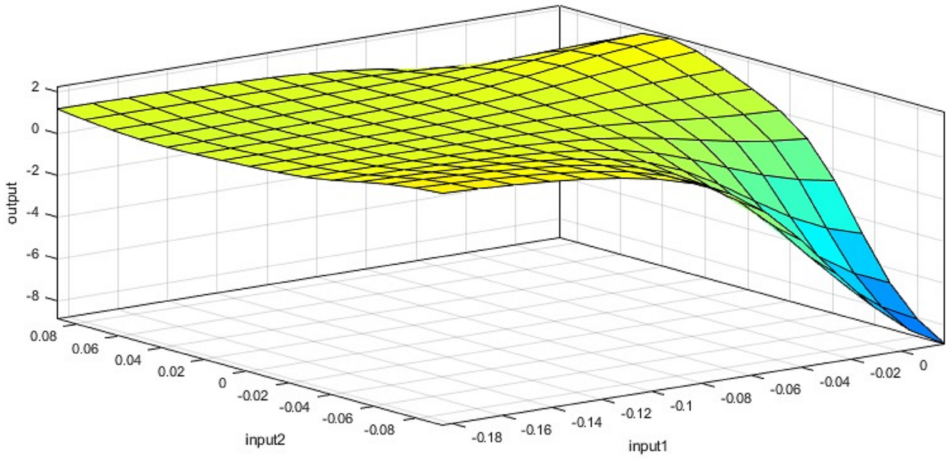

Figure 10. Representation of the fuzzy rules surface for the ANFIS controller acceleration (a). This is one output of the ANFIS controller. The degree of membership is shown in a color contour from blue (lowest) to green (highest).

4.3. Validation of the Learned ANFIS Controller

Following the methodology outlined in Section 3.4, the ANFIS controller was integrated into the autonomous vehicle system, and the closed-loop system was simulated. The autonomous vehicle operated under the same conditions as those previously established for the MPC controller.

This simulation involved implementing a control algorithm for a vehicle racing scenario aimed at finding a trajectory within the circuit. It considered a pre-defined trajectory plan along with the constraints of the circuit (Verschueren 2016 map), the ANFIS controller models obtained in Section 4.2, and other vehicle parameters mentioned in Section 3.1. The trajectory points defined the racing track boundaries and reference states, including the velocity, curvature, position, and orientation of the vehicle. The simulation loop iterated over time steps, where at each step, it calculated the errors between the current states and reference states. The simulation used the ANFIS controller (evalfis) to determine control actions based on these errors.

Figure 11 provides a visual representation of the trajectory followed in the x- and y-axes. To gain detailed insight into the newly designed controller, we conducted a comparative analysis of the other variables of the autonomous vehicle when using the MPC and ANFIS controllers. Figure 12 depicts a side-by-side comparison of the states and control actions between the reference values provided by the planner and those obtained using these two controllers. Furthermore, Figure 13 shows the state errors (relative to the planner) of both controllers. These errors are quantified as the Mean Square Error (MSE) and are shown in Table 4.

Table 4. Trajectory state errors (MSE) relative to the planner for both the ANFIS controller (evalfis) and the MPC controller.

MSE	v_{xe}	v_{ye}	ω_e
ANFIS	0.2144	0.0280	0.0417
MPC	0.0587	0.0323	0.0518

The trajectory-following performance of the autonomous vehicle was almost the same when using both the ANFIS controller and the MPC controller. Moreover, trajectory errors in two states (v_y and ω) were smaller when the system used the ANFIS controller. Differences in the evolution of one state (v_x) did not affect trajectory tracking. Furthermore,

the primary advantage is the notable reduction in computational time. Figure 14 shows the time elapsed for a one-cycle iteration of the autonomous vehicle on the Verschueren 2016 map under the same conditions but with different controllers:

(*a*) MPC controller using LPV to identify the vehicle model (LPV-MPC) [29].
(*b*) MPC controller using the non-linear technique (NL-MPC) [24].
(*c*) MPC controller using TS to identify the vehicle model (TS-MPC) [45].
(*d*) ANFIS controller using TS to identify the control and vehicle models.

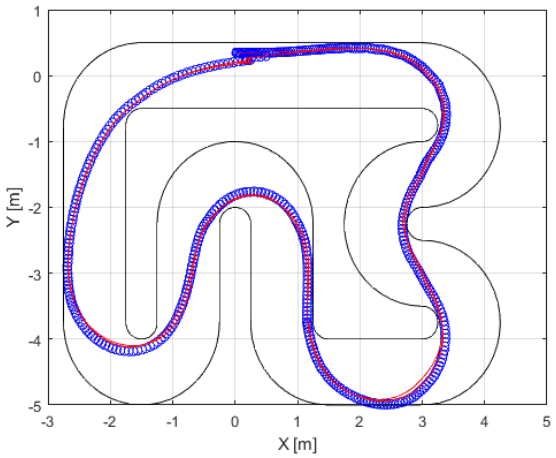

Figure 11. Closed-loop ANFIS controller simulations, with the positions of the autonomous vehicle system (blue line), trajectory (red line), and track boundaries (black line). This operation is known as ANFIS controller design validation.

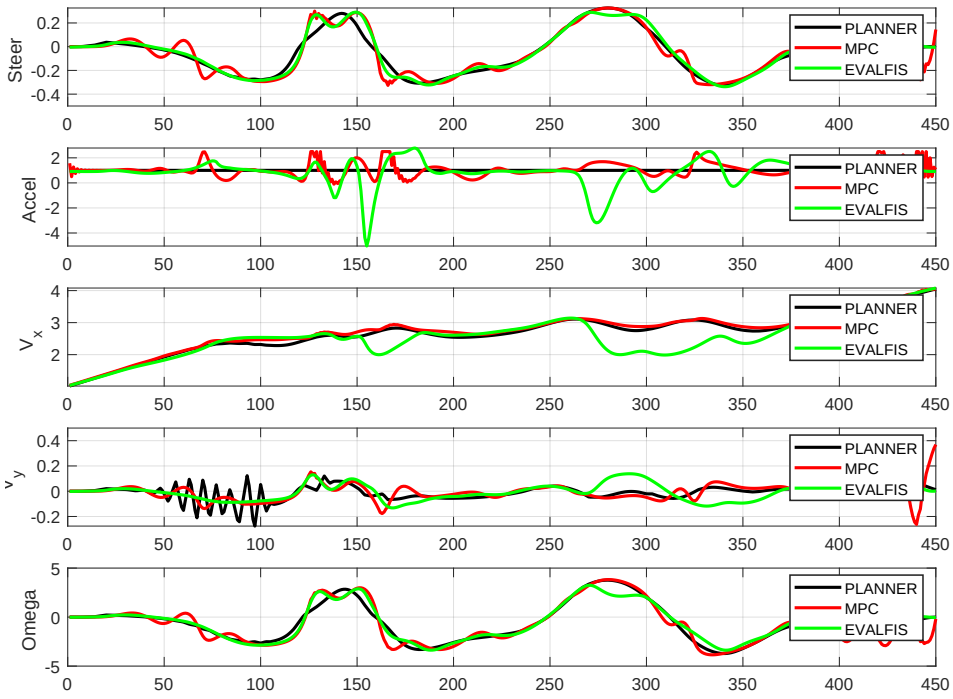

Figure 12. Parameters of the autonomous vehicle system navigating the Verschueren map, using an ANFIS controller (*evalfis*) and an MPC controller. The planner parameters are also highlighted.

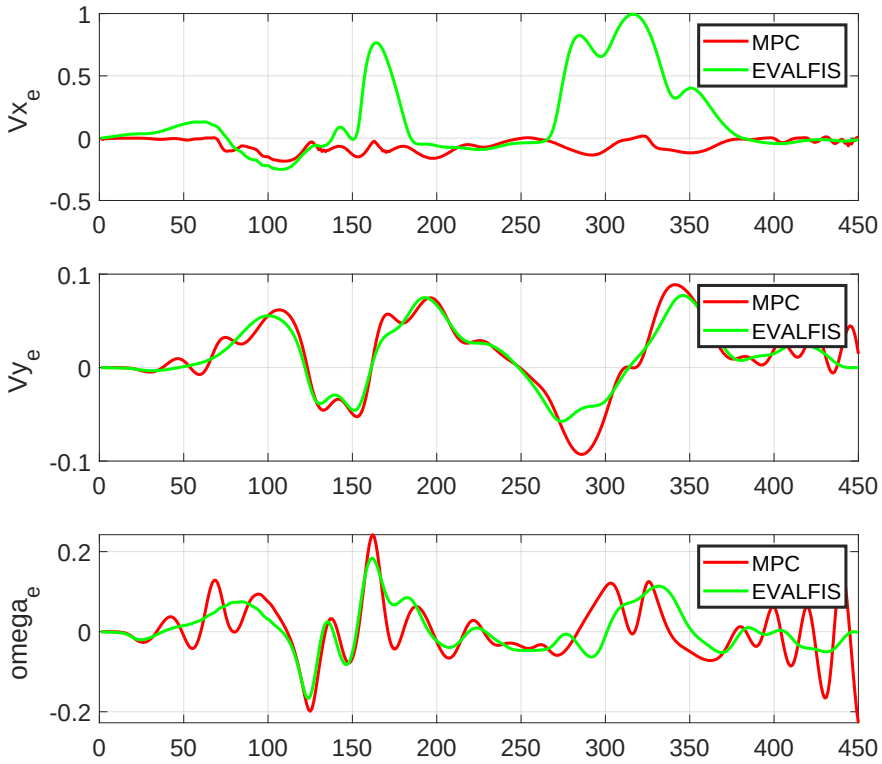

Figure 13. Visualization of the state errors (relative to the planner) for both the ANFIS controller (*evalfis*) and MPC controller.

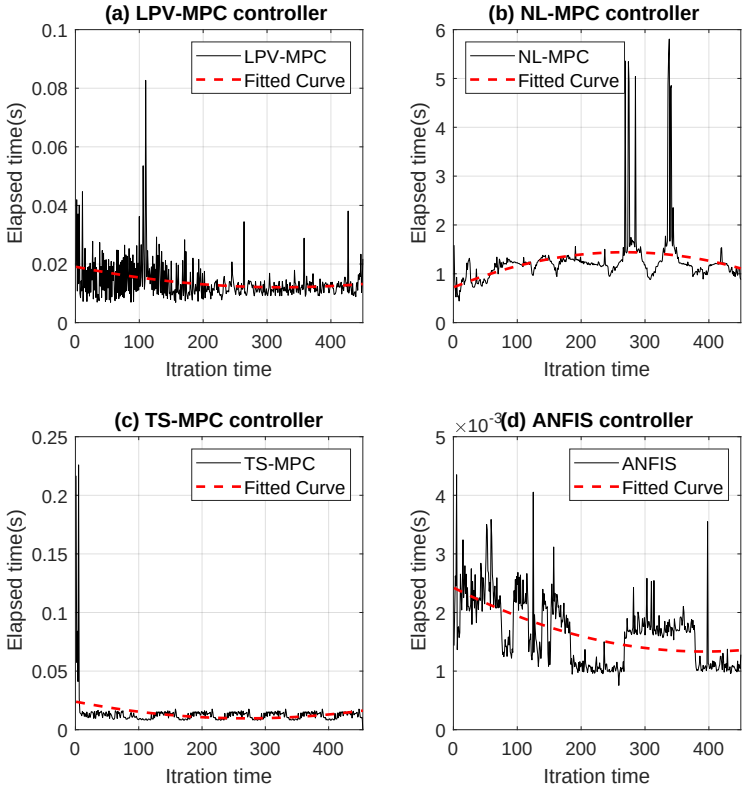

Figure 14. Computational times required for the simulation (one-cycle iteration of the autonomous vehicle on the Verschueren 2016 map) with different controllers: (**a**) LPV-MPC [29], (**b**) NL-MPC [24], (**c**) TS-MPC [45], (**d**) ANFIS.

Figure 15 depicts a comparison of the controllers, where subfigure (a) shows that the NL-MPC operated at a higher sampling rate of 20 ms, while the other controllers operated below that rate. However, subfigure (b) shows that the ANFIS controller was approximately 10 times faster than the LPV-MPC and TS-MPC controllers.

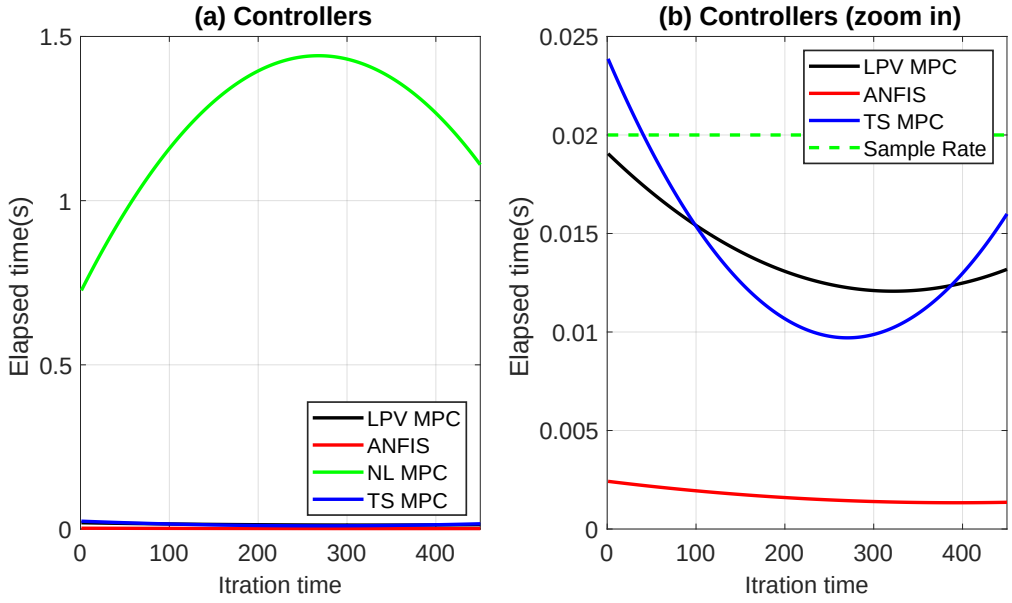

Figure 15. (**a**) Comparison of computational times for all controllers; (**b**) Comparison of elapsed times for controllers operating at a sampling rate of around 20 ms.

4.4. Validated Takagi–Sugeno (TS) Representation for Both Control and Vehicle Models

The ANFIS output was calculated using the *evalfis* function in Matlab R2021a (from MathWorks, Massachusetts, United States)and the TS explicit representation of the ANFIS control law was also obtained for stability analysis. As a controller, the ANFIS involved three inputs, each associated with two membership functions. This resulted in eight matrices for $K \in \mathbb{R}^{2\times 3}$, representing the control model. In addition, the ANFIS was used to obtain the vehicle model, which, in this case, involved three inputs, each associated with two membership functions, resulting in eight matrices of $A \in \mathbb{R}^{3\times 3}$ and two matrices of $B \in \mathbb{R}^{3\times 2}$. The TS representations for both ANFIS models (controller and vehicle) were obtained using Equations (12), (23), and (24). These TS representations must be verified through the conformity of the TS representation output (explicit *evalfis*) and ANFIS output (*evalfis*). However, there is no guarantee that the TS representation precisely follows the ANFIS output. This validation for the TS representation for the control model is depicted in Figure 16, and for the vehicle model, in Figure 17. The overlap between *evalfis* and the TS explicit representation allowed us to confirm that the TS model corrected both the controller and vehicle.

4.5. Stability Assessment

The stability assessment relied on Lyapunov theory, as outlined in Section 3.7. This process utilized YALMIP with SeDuMi 1.3.4, involving 275 LMIs. The assessment concluded the stability of the autonomous vehicle system using the ANFIS controller by establishing a positive-definite matrix P, as follows:

$$P = \begin{bmatrix} 1 & -1.19 \times 10^{-12} & 1.15 \times 10^{-13} \\ -1.19 \times 10^{-12} & 1 & -5.64 \times 10^{-12} \\ 1.15 \times 10^{-13} & -5.64 \times 10^{-12} & 1 \end{bmatrix}$$

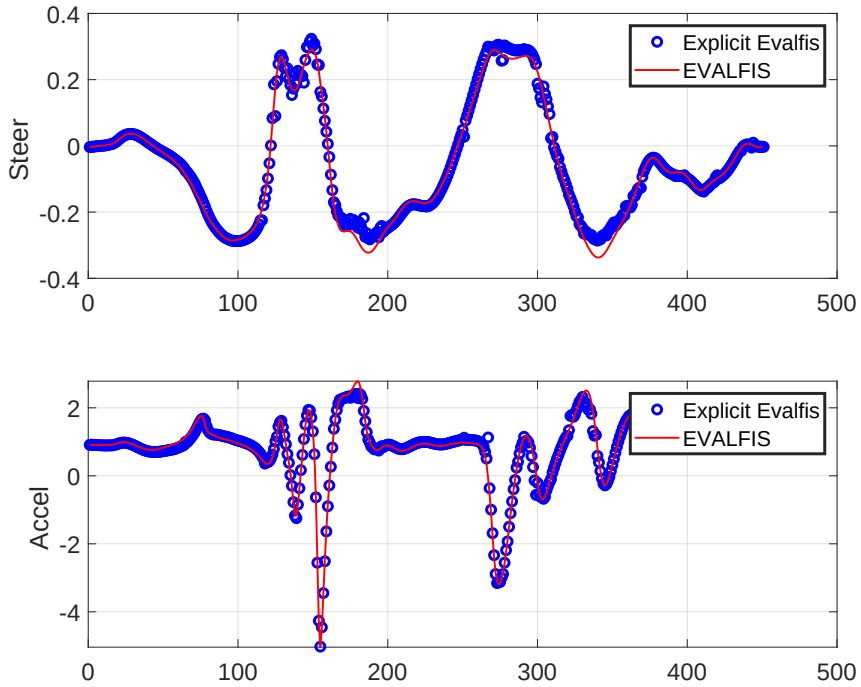

Figure 16. TS representation and validation for the control model: The control actions calculated by the TS model (explicit *evalfis*) closely match those calculated by the ANFIS (*evalfis*).

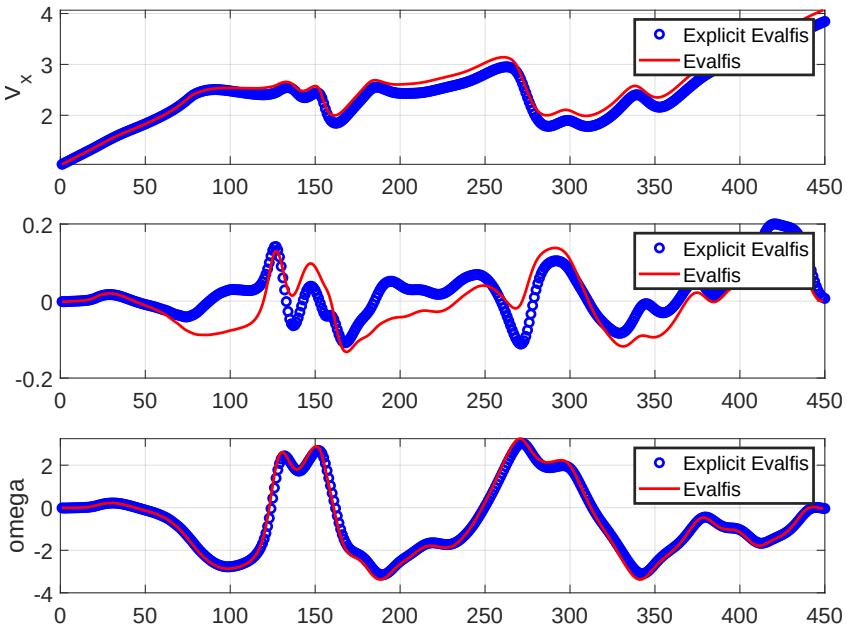

Figure 17. TS representation and validation for the vehicle model: The states calculated by the TS model (explicit *evalfis*) closely match those calculated by the ANFIS (*evalfis*).

5. Conclusions

In this article, a learning-based approach has been presented to design a controller for an autonomous vehicle under realistic conditions in real time. The ANFIS, as a learning method, is applied to learn a control law using training data obtained from a pre-existing controller. The control law learned from the data is formulated as a Takagi–Sugeno (TS) representation. At the same time, the vehicle model is also learned from the data using the ANFIS. This allows for the proof of closed-loop stability using Lyapunov theory and LMIs for the autonomous vehicle system when the ANFIS controller is used. The proposed approach is validated through simulation using racing-based references provided by an

external planner. The ANFIS controller enables the vehicle to perform in racing mode. In comparison to an autonomous vehicle controlled using the MPC controller under the same conditions, the ANFIS controller exhibits reduced errors in two vehicle states (V_y and ω) while performing satisfactory trajectory tracking with the added advantage of lower computational time. This strategy has been proposed as an approach to address both autonomous driving control problems and to serve as a parallel controller, enhancing system reliability in the event of a malfunction in the primary controller. The required data in this approach can be obtained from different methods, such as real-world experiments or through simulation. For future work, this approach can be implemented in a real-world experiment using the scale autonomous car and a full-sized vehicle. The application is straightforward since only data from the real vehicle are required. In addition, the proposed method can be applied to the platooning control of multiple vehicles. Some interesting fields of application include the co-design of a bandwidth-aware communication scheduler and cruise controller for multiple high-speed trains, as well as the secure and collision-free multi-platoon control of automated vehicles under data falsification attacks. It is possible to apply the approach using other machine learning methods, such as reinforcement learning [51].

Author Contributions: Methodology, M.S. and V.P.; Investigation, M.S.; Writing—original draft, M.S. and V.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by the Spanish State Research Agency (AEI) and the European Regional Development Fund (ERFD) through the project SaCoAV (Ref. MINECO PID2020-114244RB-I00).

Data Availability Statement: To inquire about the availability of the data used in this study, readers may contact the authors via email.

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

References

1. Van Woensel, L.; Archer, G.; Panades-Estruch, L.; Vrscaj, D. *Ten Technologies Which Could Change Our Lives: Potential Impacts and Policy Implications*; European Union Journal: Brussels, Belgium 2015.
2. Yurtsever, E.; Lambert, J.; Carballo, A.; Takeda, K. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access* **2020**, *8*, 58443–58469. [CrossRef]
3. Ritchie, H.; Roser, M. Urbanization. In *Our World in Data*; Global Change Data Lab, England and Wales, UK, 2018.
4. Montgomery, W.D.; Mudge, R.; Groshen, E.L.; Helper, S.; MacDuffie, J.P.; Carson, C. America's Workforce and the Self-Driving Future: Realizing Productivity Gains and Spurring Economic Growth, Washington, United States, 2018. Available online: https://trid.trb.org/View/1516782 (accessed on 22 February 2024).
5. Warrendale, P. *Levels of Automation for On-Road Vehicles*; Society of Automotive Engineers (SAE): Warrendale, PA, USA, 2014.
6. Bachute, M.R.; Subhedar, J.M. Autonomous driving architectures: insights of machine learning and deep learning algorithms. *Mach. Learn. Appl.* **2021**, *6*, 100164. [CrossRef]
7. Jin, L.; Zhang, R.; Tang, B.; Guo, H. A Fuzzy-PID scheme for low speed control of a vehicle while going on a downhill road. *Energies* **2020**, *13*, 2795. [CrossRef]
8. Chen, S.; Chen, H. MPC-based path tracking with PID speed control for autonomous vehicles. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2020; Volume 892, p. 012034.
9. Peicheng, S.; Li, L.; Ni, X.; Yang, A. Intelligent vehicle path tracking control based on improved MPC and hybrid PID. *IEEE Access* **2022**, *10*, 94133–94144. [CrossRef]
10. Yang, K.; Tang, X.; Qin, Y.; Huang, Y.; Wang, H.; Pu, H. Comparative study of trajectory tracking control for automated vehicles via model predictive control and robust H-infinity state feedback control. *Chin. J. Mech. Eng.* **2021**, *34*, 74. [CrossRef]
11. Awad, N.; Lasheen, A.; Elnaggar, M.; Kamel, A. Model predictive control with fuzzy logic switching for path tracking of autonomous vehicles. *ISA Trans.* **2022**, *129*, 193–205. [CrossRef] [PubMed]
12. Arifin, B.; Suprapto, B.Y.; Prasetyowati, S.A.D.; Nawawi, Z. Steering control in electric power steering autonomous vehicle using type-2 fuzzy logic control and PI control. *World Electr. Veh. J.* **2022**, *13*, 53. [CrossRef]
13. Sabiha, A.D.; Kamel, M.A.; Said, E.; Hussein, W.M. ROS-based trajectory tracking control for autonomous tracked vehicle using optimized backstepping and sliding mode control. *Robot. Auton. Syst.* **2022**, *152*, 104058. [CrossRef]
14. Alcalá, E.; Sellart, L.; Puig, V.; Quevedo, J.; Saludes, J.; Vázquez, D.; López, A. Comparison of two non-linear model-based control strategies for autonomous vehicles. In Proceedings of the 2016 24th Mediterranean Conference on Control and Automation (MED), Athens, Greece, 21–24 June 2016; pp. 846–851.

15. Karafyllis, I.; Theodosis, D.; Papageorgiou, M. Lyapunov-based two-dimensional cruise control of autonomous vehicles on lane-free roads. *Automatica* **2022**, *145*, 110517. [CrossRef]
16. Atoui, H.; Sename, O.; Milanes, V.; Martinez-Molina, J.J. Toward switching/interpolating LPV control: A review. *Annu. Rev. Control* **2022**, *54*, 49–67. [CrossRef]
17. Wang, J.; Ding, B.; Wang, P. Event-triggered output feedback predictive control for Takagi-Sugeno model with bounded disturbance and redundant channels. *Int. J. Robust Nonlinear Control* **2023**, *33*, 3045–3063. [CrossRef]
18. Park, M.; Kang, Y. Experimental verification of a drift controller for autonomous vehicle tracking: A circular trajectory using LQR method. *Int. J. Control Autom. Syst.* **2021**, *19*, 404–416. [CrossRef]
19. Zhang, K.; Sun, Q.; Shi, Y. Trajectory tracking control of autonomous ground vehicles using adaptive learning MPC. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 5554–5564. [CrossRef] [PubMed]
20. Cheng, S.; Li, L.; Chen, X.; Wu, J. Model-predictive-control-based path tracking controller of autonomous vehicle considering parametric uncertainties and velocity-varying. *IEEE Trans. Ind. Electron.* **2020**, *68*, 8698–8707. [CrossRef]
21. Minh, V.T.; Moezzi, R.; Dhoska, K.; Pumwa, J. Model predictive control for autonomous vehicle tracking. *Int. J. Innov. Technol. Interdiscip. Sci.* **2021**, *4*, 560–603.
22. Pang, H.; Liu, N.; Hu, C.; Xu, Z. A practical trajectory tracking control of autonomous vehicles using linear time-varying MPC method. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2022**, *236*, 709–723. [CrossRef]
23. Allamaa, J.P.; Listov, P.; Van der Auweraer, H.; Jones, C.; Son, T.D. Real-time nonlinear mpc strategy with full vehicle validation for autonomous driving. In Proceedings of the 2022 American Control Conference (ACC), Atlanta, GA, USA, 8–10 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1982–1987.
24. Pereira, G.C.; Lima, P.F.; Wahlberg, B.; Pettersson, H.; Mårtensson, J. Nonlinear Curvature Modeling for MPC of Autonomous Vehicles. In Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–7.
25. Bernal, M.; Sala, A.; Lendek, Z.; Guerra, T.M. *Analysis and Synthesis of Nonlinear Control Systems*; Springer: Berlin/Heidelberg, Germany, 2022.
26. Zheng, Y.; Li, S.E.; Li, K.; Borrelli, F.; Hedrick, J.K. Distributed model predictive control for heterogeneous vehicle platoons under unidirectional topologies. *IEEE Trans. Control. Syst. Technol.* **2016**, *25*, 899–910. [CrossRef]
27. Shamma, J.S. An overview of LPV systems. In *Control of Linear Parameter Varying Systems with Applications*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 3–26.
28. Alcalá, E.; Puig, V.; Quevedo, J.; Sename, O. Fast zonotope-tube-based LPV-MPC for autonomous vehicles. *IET Control Theory Appl.* **2020**, *14*, 3676–3685. [CrossRef]
29. Alcalá, E.; Puig, V.; Quevedo, J.; Rosolia, U. Autonomous racing using linear parameter varying-model predictive control (LPV-MPC). *Control Eng. Pract.* **2020**, *95*, 104270. [CrossRef]
30. Wang, H.O.; Tanaka, K. *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*; John Wiley & Sons: Hoboken, NJ, USA, 2004.
31. Alcala, E.; Puig, V.; Quevedo, J.; Escobet, T.; Comasolivas, R. Autonomous vehicle control using a kinematic Lyapunov-based technique with LQR-LMI tuning. *Control Eng. Pract.* **2018**, *73*, 1–12. [CrossRef]
32. Tran, G.Q.B.; Pham, T.P.; Sename, O.; Gáspár, P. Design of an LMI-based Polytopic LQR Cruise Controller for an Autonomous Vehicle towards Riding Comfort. *Period. Polytech. Transp. Eng.* **2023**, *51*, 1–7. [CrossRef]
33. Cheng, S.; Li, L.; Liu, C.Z.; Wu, X.; Fang, S.N.; Yong, J.W. Robust LMI-Based H-Infinite Controller Integrating AFS and DYC of Autonomous Vehicles With Parametric Uncertainties. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 6901–6910. [CrossRef]
34. Jang, J.S. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Trans. Syst. Man Cybern. Syst.* **1993**, *23*, 665–685. [CrossRef]
35. Pamucar, D.; Ćirović, G. Vehicle route selection with an adaptive neuro fuzzy inference system in uncertainty conditions. *Decis. Mak. Appl. Manag. Eng.* **2018**, *1*, 13–37. [CrossRef]
36. Avdagic, Z.; Cernica, E.; Omanovic, S. Adaptive neuro-fuzzy inference system based modelling of vehicle guidance. *J. Eng. Sci. Technol.* **2019**, *14*, 2116–2131.
37. Patel, D.; Cohen, K. Obstacle Avoidance and Target Tracking by Two Wheeled Differential Drive Mobile Robot Using ANFIS in Static and Dynamic Environment. In *Fuzzy Information Processing 2020: Proceedings of the 2020 Annual Conference of the North American Fuzzy Information Processing Society, NAFIPS 2020, Virtual Event, 20–22 August 2020*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 337–348.
38. Ravikumar, S.; Kavitha, D. IOT based autonomous car driver scheme based on ANFIS and black widow optimization. *J. Ambient. Intell. Humaniz. Comput.* **2021**, 1–14. [CrossRef]
39. Xue, H.; Zhang, Z.; Wu, M.; Chen, P. Fuzzy controller for autonomous vehicle based on rough sets. *IEEE Access* **2019**, *7*, 147350–147361. [CrossRef]
40. Pandey, A.; Kumar, S.; Pandey, K.K.; Parhi, D.R. Mobile robot navigation in unknown static environments using ANFIS controller. *Perspect. Sci.* **2016**, *8*, 421–423. [CrossRef]
41. Takagi, T.; Sugeno, M. Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. Syst. Man Cybern.* **1985**, *SMC-15*, 116–132. [CrossRef]

42. Fantuzzi, C.; Rovatti, R. On the approximation capabilities of the homogeneous Takagi-Sugeno model. In Proceedings of the IEEE 5th International Fuzzy Systems, New Orleans, LA, USA, 11 September 1996; IEEE: Piscataway, NJ, USA, 1996; Volume 2, pp. 1067–1072.
43. Hu, J.; Li, X.; Xu, Z.; Pan, H. Co-Design of Quantized Dynamic Output Feedback MPC for Takagi-Sugeno Model. *IEEE Trans. Ind. Inform.* **2022**, *19*, 8049–8060. [CrossRef]
44. Ding, B.; Pan, H. Dynamic output feedback-predictive control of a Takagi–Sugeno model with bounded disturbance. *IEEE Trans. Fuzzy Syst.* **2016**, *25*, 653–667. [CrossRef]
45. Alcalá, E.; Bessa, I.; Puig, V.; Sename, O.; Palhares, R. MPC using an on-line TS fuzzy learning approach with application to autonomous driving. *Appl. Soft Comput.* **2022**, *130*, 109698. [CrossRef]
46. Chaubey, S.; Puig, V. Autonomous Vehicle State Estimation and Mapping Using Takagi–Sugeno Modeling Approach. *Sensors* **2022**, *22*, 3399. [CrossRef] [PubMed]
47. Alcala, E.; Sename, O.; Puig, V.; Quevedo, J. TS-MPC for Autonomous Vehicle using a Learning Approach. *IFAC-PapersOnLine* **2020**, *53*, 15110–15115. [CrossRef]
48. Rajamani, R. *Vehicle Dynamics and Control*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2011.
49. Samada, S.E.; Puig, V.; Nejjari, F. Battery Health-Aware MPC Planning for Autonomous Racing Vehicles. In Proceedings of the Conference on Latin America Control Congress, Havana, Cuba, 15–17 November 2022; Springer: Berlin/Heidelberg, Germany, 2023; pp. 177–188.
50. Alcala, E.; Puig, V.; Quevedo, J.; Escobet, T. Gain-scheduling LPV control for autonomous vehicles including friction force estimation and compensation mechanism. *IET Control Theory Appl.* **2018**, *12*, 1683–1693. [CrossRef]
51. Albarella, N.; Lui, D.G.; Petrillo, A.; Santini, S. A Hybrid Deep Reinforcement Learning and Optimal Control Architecture for Autonomous Highway Driving. *Energies* **2023**, *16*, 3490. [CrossRef]