



Lateral control for autonomous wheeled vehicles: a technical review

Yassine Kebbat, Naïma Aït Oufroukh, Dalil Ichalal, Vincent Vigneron

► To cite this version:

Yassine Kebbat, Naïma Aït Oufroukh, Dalil Ichalal, Vincent Vigneron. Lateral control for autonomous wheeled vehicles: a technical review. Asian Journal of Control, 2023, 25 (4), pp.2539–2563. <10.1002/asjc.2980>. <hal-04371558>

HAL Id: hal-04371558

<https://hal.science/hal-04371558v1>

Submitted on 3 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Lateral control for autonomous wheeled vehicles: a technical review

Yassine Kebbaty^a, Naima Ait-Oufroukh^a, Dalil Ichalal^a, Vincent Vigneron^{a,b}

^a*IBISC-EA4526, Université Paris-Saclay, Evry, France*

^b*School of Applied Sciences (FCA), UNICAMP, Limeira, Brazil*

Abstract

Autonomous driving has the ability to reshape mobility and transportation by reducing road accidents, traffic jams and air pollution. This can yield energy efficiency, convenience and more productivity as significant driving time will be gained and used in other activities instead. Autonomous vehicles are complex systems consisting of several modules that perform perception, decision making, planning and control. Control is essential for achieving automatic driving, it is basically divided into longitudinal control that handles speed tracking, and lateral control which ensures accurate steering. The latter is primordial in path tracking applications and recent research has witnessed a huge leap in this field. The aim of this paper is to provide a technical survey of the latest research on the lateral control of autonomous vehicles as well as to highlight technical challenges and limits for further developments.

Key words: Automated driving, Model-based control, Artificial intelligence, Path tracking, Optimization, Modeling.

ACRONYMS

CG Center of Gravity
DoF Degrees of Freedom
MPC Model Predictive Control
NMPC Nonlinear Model Predictive Control
PSO Particle Swarm Optimization
GA Genetic Algorithms
ANFIS Adaptive Neuro-Fuzzy Inference System
LPV Linear Parameter Varying
LTV Linear Time Varying
NLP Nonlinear Optimization Problem
LMI Linear Matrix Inequality
PID Proportional Integral Derivative
SMC Sliding Mode Control
LQR Linear Quadratic Regulator
RBFNN Radial Basis Function Neural Network
CNN Convolutional Neural Network
RNN Recurrent Neural Network
MLP Multi-layer Perceptron
MDP Markov Decision Process

MIMO Multi-Input Multi-Output
QP Quadratic Programming

1 INTRODUCTION

Self-driving technology has been the trend for the last decades, it is one of the technologies that will completely change people's lifestyle. The shift towards autonomous driving is encouraged by the recent developments in artificial intelligence, big data and information processing techniques. Autonomous driving will considerably reduce road accidents and traffic congestion, air pollution will be mitigated, and energy consumption will be optimized. Consequently, researchers are working towards full autonomy to reduce the human error and mitigate the risks and dangers of manual driving which accounts for almost 96% of all car accidents in the US [1]. Researchers in self-driving vehicles have been inspired by the DARPA challenge [2] that was organized by the USA which is a leading giant in the field. Waymo [3] has achieved great advancements in the field, with over 10 million miles of road testing and 7 billion miles of virtual testing.

Automatic driving systems comprise many modules that work in a coordinated way. The perception module consists of multiple sensors such as: cameras, GPS,

* Corresponding author Yassine Kebbaty.

Email addresses: `yassine.kebbati@univ-evry.fr` (Yassine Kebbaty), `naima.aitoufroukh@univ-evry.fr` (Naima Ait-Oufroukh), `dalil.ichalal@univ-evry.fr` (Dalil Ichalal), `vincent.vigneron@univ-evry.fr` (Vincent Vigneron).

LIDAR, RADAR and IMU. A variety of algorithms are used by the perception module to sense the environment and obtain relevant information. The planning and decision-making module uses sensor data to make decisions and plan the speed and trajectory profiles. These decisions are executed by the control module. Control is one of the most important tasks to achieve autonomous driving. Generally, the control module is divided into longitudinal and lateral control, longitudinal control handles speed tracking and lateral control ensures accurate steering. This is achieved by commanding the different actuators such as the accelerator, the brakes and the steering wheel. Self-driving cars are divided into 5 automation levels, level zero with no automation represents the conventional cars. In level one, simple automatic driving systems like adaptive cruise control and electronic stability control are deployed into the vehicle. Advanced systems like speed and steering control or emergency braking are introduced in level two. In level three, the vehicle is capable of sensing the environment through multiple sensors and it is able to drive autonomously while requiring driver supervision and intervention in infeasible situations. The vehicle is autonomous in level four, with only occasional driver intervention for certain driving modes only. Level five vehicles are completely autonomous, they can handle all driving modes and without any driver supervision [4].

Several studies have been conducted to develop models that best describe the behavior of vehicles. The vehicle model is essential for developing controllers and simulating vehicle systems in general. Simulations are useful for tuning controllers and observing their performance. Generally, realistic vehicle models are highly nonlinear and complex, but linearized variants are usually used for controller design. The state space representation of the model is the most commonly used for developing controllers since it is much more convenient and facilitates the task. The latter is derived from the model mathematical equations after some manipulations.

Existing vehicle modeling techniques in the literature range from simple methods like the geometric model used in pure pursuit [5] and Stanley [6] controllers, to complicated multi-body models as in [7] and [8]. The geometric models are well suited for path tracking, they are simple and require few parameters since they are mainly based on the geometry of the vehicle (position/dimensions). Although they are low on computation, they lack performance since they do not consider the motion of the vehicle and the forces applied on it. Complicated multi-body models treat the vehicle as a multi-articulate system consisting of multiple bodies, this robotics formalism considers the vehicle chassis as the movable base with the wheels being the terminals. When it comes to controller design two models are distinguished; the kinematic model and

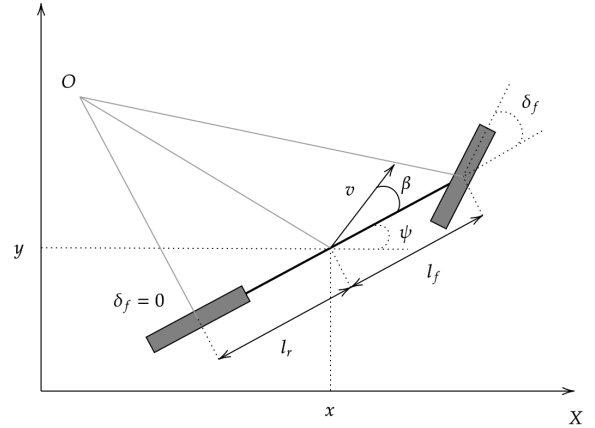


Fig. 1. Kinematic bicycle model.

the dynamic model, these are the widespread models in the literature, since they accurately depict vehicle behavior while still being relatively simple.

This article presents a technical review of lateral control methods, it provides an overview of the most widely used steering controllers and illustrates their design methods. Model-free controllers are also reviewed, the advantages and disadvantages of the lateral controllers are detailed in a summarized comparison. Section II of this article presents the most widely used models for controller design. The different model-based control methods and their design approach are discussed in section III and section IV presents model-free control techniques. Finally, section V concludes the article.

2 VEHICLE MODELING

2.1 Kinematic Bicycle Model

The kinematic model takes into consideration the kinematics of the vehicle, hence the name kinematic. The model describes the motion of the vehicle in terms of position, velocity and acceleration and disregards the forces acting on the vehicle. Despite the fact that the kinematic model is more sophisticated compared to simple geometric models, it is restricted and describes the vehicle motion under certain assumptions. The model works well when there are no slip angles, that is to say the velocity vector and the orientation of the wheels are identical. Such assumption holds true for low speeds, typically below 5 m/s [9].

The simplified bicycle kinematic model is well developed and very common in the literature [10], [11], [12], yet it is mostly limited to low speed applications and cannot be relied on for highly dynamic maneuvers. Rajamani [9] provided a full description of the model in its simplest form, it has been used in the design of numerous controllers in [13], [14] and [15]. The simplicity of the

bicycle model is in reducing the two front and rear wheels to a single front wheel and a single rear wheel which resembles a bicycle. As described in [9], the bicycle kinematic model can be summarized by the following equations expressed in the inertial frame:

$$\dot{x} = v \cos(\psi + \beta) \quad (1)$$

$$\dot{y} = v \sin(\psi + \beta) \quad (2)$$

$$\dot{\psi} = \frac{v}{l_r} \sin(\beta) \quad (3)$$

$$\dot{v} = \alpha \quad (4)$$

$$\beta = \tan^{-1} \left(\frac{l_r}{l_f + l_r} \tan(\delta_f) \right) \quad (5)$$

As can be seen in Fig. 1, x and y represent the coordinates of the vehicle's Center of Gravity (CG) in the inertial frame (OXY). The heading is described by ψ , while v represents the velocity of the vehicle. The front and rear wheel axles are distant from the CG by l_f and l_r respectively. β denotes the side-slip angel (angle between velocity vector and vehicle longitudinal axis), and α is the vehicle longitudinal acceleration. The inputs to the model are the steering angles δ_f and δ_r for the front and rear wheels, but most vehicles are only steerable through the front wheel, therefore in most case δ_r is null. The kinematic model requires the identification of only two parameters which are the distances from the wheel axles to the CG (l_f and l_r), hence applying the same controller to other vehicles with different wheelbases is straightforward.

2.2 Dynamic Bicycle Model

A more accurate representation of the vehicle requires modeling its dynamics, which considers behaviors like side-slipping, over-steering and friction. The dynamic model is more accurate than the kinematic model in the sense that it includes the applied forces on the vehicle and especially the tire forces. Newton's second law of motion and Euler Lagrange methods are applied on the vehicle system to obtain the dynamic model. The complete dynamic model is very complex and nonlinear accounting for translation and rotation motions in the 3D space and considering a full vehicle with four wheels [16]. Such models are used mainly for validation purposes and are too complicated for controller design (see section 2.3), simplified two-wheel dynamic models are used instead. The dynamic bicycle model accounts for 2D planar motion; translation in the X and Y axis (Longitudinal\Lateral) and rotation around the Z axis (Yaw) of the reference frame. This yields a 3 Degrees of Freedom (DoF) vehicle model [17] and in some cases the longitudinal dynamics are disregarded such as in path tracking where the task is reduced to controlling lateral dynamics and yaw motion resulting in a 2DoF model. Considering the simple bicycle model in Fig. 2 and applying newton's laws results in the following

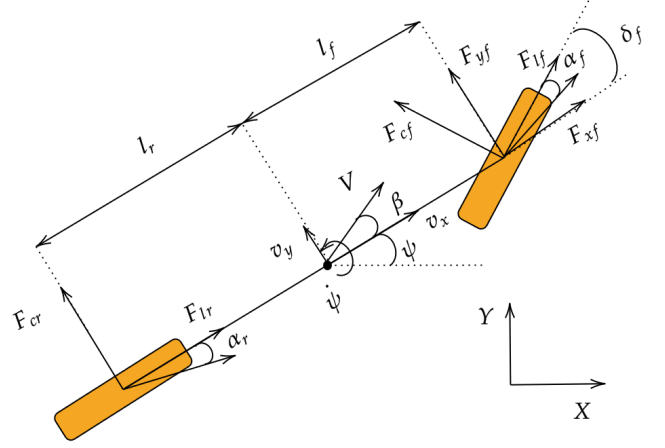


Fig. 2. Dynamic bicycle model.

model equations:

$$\sum F_x = m\alpha_x \quad (6)$$

$$\sum F_y = m\alpha_y \quad (7)$$

$$\sum M_z = I_z \ddot{\psi} \quad (8)$$

where I_z represents the moment of inertia, M_z the rotation moment around the Z axis and F_x and F_y are the longitudinal and lateral forces respectively. These are applied on the front and rear wheels. α_x and α_y are the longitudinal and lateral inertial accelerations and are expressed in terms of longitudinal\lateral accelerations (\ddot{x} , \ddot{y}), the yaw rate $\dot{\psi}$, the longitudinal\lateral velocities (\dot{x} , \dot{y}) as:

$$\alpha_x = \ddot{x} - \dot{\psi}\dot{y} \quad (9)$$

$$\alpha_y = \ddot{y} + \dot{\psi}\dot{x} \quad (10)$$

From equations (6) to (10) and considering the nonholonomic constraints, the full model can be derived as follows [18]:

$$\ddot{x} = \dot{y}\dot{\psi} + \frac{2}{m}(F_{xf} + F_{xr}) \quad (11)$$

$$\ddot{y} = -\dot{x}\dot{\psi} + \frac{2}{m}(F_{yf} + F_{yr}) \quad (12)$$

$$\ddot{\psi} = \frac{2}{I_z}(l_f F_{yf} + l_r F_{yr}) \quad (13)$$

$$\dot{Y} = \dot{x} \sin \psi + \dot{y} \cos \psi \quad (14)$$

$$\dot{X} = \dot{x} \cos \psi - \dot{y} \sin \psi \quad (15)$$

From Fig. 2 the relationship between the lateral and longitudinal forces acting on the tires and those acting on the vehicle's CG is as follows:

$$F_{xf} = F_{lf} \cos \delta + F_{cf} \sin \delta \quad (16)$$

$$F_{yf} = F_{cf} \cos \delta - F_{lf} \sin \delta \quad (17)$$

$$F_{xr} = F_{lr} \quad (18)$$

$$F_{yr} = F_{cr} \quad (19)$$

The tires of the vehicle interact with the road surface and undergo deformations with different maneuvers resulting in tire forces in the longitudinal and lateral directions. Although these forces are nonlinear functions, they are often linearized to be used in the simple bicycle model. The linearization of tire forces is possible under the assumption of small slip angles, and it is done by considering that slip angles and tire forces are proportional. Typically the linear region for longitudinal slip angles is under $\frac{1}{2}g$ accelerations and for lateral slip angles under 5° [19]. In such cases, the lateral tire forces can be expressed as:

$$F_{cf} = C_f \alpha_f \quad (20)$$

$$F_{cr} = C_r \alpha_r \quad (21)$$

The parameters C_f and C_r represent the lateral tire cornering stiffness coefficients for the front and rear wheels respectively, which can be seen as the tangent line of the curve representing tire lateral force with respect to the slip angle. α_f and α_r are the respective front and rear wheel side-slip angles, they represents the deviation between the velocity vector and the direction of the wheel see Fig. 2. In general, both the vehicle model and the tire forces are nonlinear, the Pacejka model known as magic formula works well with nonlinear models and has been extensively used in previous studies [19]. The formula is given by:

$$F(k) = d \sin(c \tan^{-1}(b\Theta(k))) + S_v \quad (22)$$

$$\Theta(k) = (1 - e)(k + S_h) + \frac{e}{b} \tan^{-1}(b(k + S_h)) \quad (23)$$

where F can be the longitudinal or the lateral tire force, k is the slip ratio in the case of F representing the longitudinal force and the slip angle when F represents the lateral force. The slip ratio defines the difference between the theoretical and the actual vehicle speed, which is mainly due to tire slipping phenomenon. b, c, d, e, S_h , and S_v are parameters fit from experimental data that define certain characteristics like the stiffness factor, the shape factor, the peak value, etc. In lateral control applications, the slip angle is important and is defined as the angle between the orientation and the velocity vector of the wheel. Under the assumption that only the front wheel is steerable, the front and rear wheel slip angles are given as $\alpha_f = \delta_f - \gamma_f$ and $\alpha_r = -\gamma_r$, with the rear steering angle set to zero ($\delta_r = 0$), and $\gamma_{f,r}$ being the angles between the wheel longitudinal

axis and longitudinal velocity which are obtained from the ratio of lateral to longitudinal velocity as follows:

$$\tan(\gamma_f) = \frac{\dot{y} + l_f \dot{\psi}}{\dot{x}} \quad (24)$$

$$\tan(\gamma_r) = \frac{\dot{y} - l_r \dot{\psi}}{\dot{x}} \quad (25)$$

Using angle approximations for linearization and replacing equations (24) and (25) in (20) and (21) results in the following tire forces.

$$F_{cf} = C_f(\delta_f - \frac{\dot{y} + l_f \dot{\psi}}{\dot{x}}) \quad (26)$$

$$F_{cr} = C_r(-\frac{\dot{y} - l_r \dot{\psi}}{\dot{x}}) \quad (27)$$

By manipulating the dynamic model equations stated previously and the obtained tire forces, a state space model for lateral dynamics can be derived as follows:

$$\dot{x}(t) = A.x(t) + B.u(t) \quad (28)$$

$$Y = C.x(t) \quad (29)$$

where the control $u = \delta_f$, the state vector denoted by $x = [y \ \dot{y} \ \psi \ \dot{\psi}]^T$ with y being the vehicle's lateral position and the output vector $Y = [\psi \ y]^T$. The matrices A, B and C are defined as follows:

$$A = \begin{bmatrix} 0 & 1 & \dot{x} & 0 \\ 0 & -2\frac{C_f + C_r}{m\dot{x}} & 0 & -\dot{x} - 2\frac{C_f l_f - C_r l_r}{m\dot{x}} \\ 0 & 0 & 0 & 1 \\ 0 & -2\frac{C_f l_f - C_r l_r}{I_z \dot{x}} & 0 & -2\frac{C_f l_f^2 + C_r l_r^2}{I_z \dot{x}} \end{bmatrix},$$

$$B = \begin{bmatrix} 0 \\ 2\frac{C_f}{m} \\ 0 \\ 2\frac{C_f l_f}{I_z} \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

2.3 Full Dynamic Model

Although the single track bicycle dynamic model is fit for control design, it lacks precision due to the simplifications and assumptions to build the model. As discussed earlier, full vehicle dynamic models are a more realistic representation of the vehicle dynamics. Such models take into account all four wheels of the vehicle and do not neglect the vertical dynamics as presented in [16, 20]. Fig. 3 shows a 3DoF dual-track full vehicle model that is defined by equations (30-36)

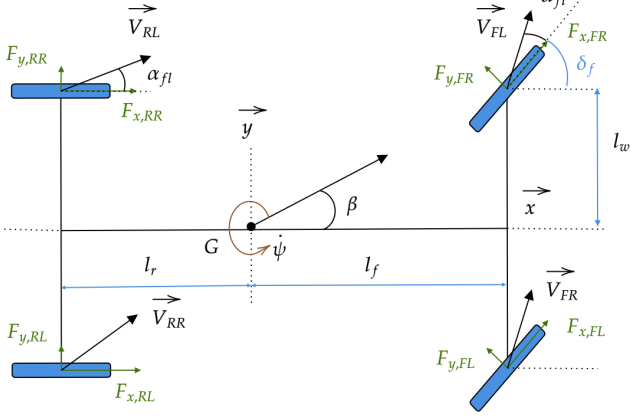


Fig. 3. Full dynamic model.

$$\dot{Y} = \dot{x} \sin \psi + \dot{y} \cos \psi \quad (30)$$

$$\dot{X} = \dot{x} \cos \psi - \dot{y} \sin \psi \quad (31)$$

$$m\ddot{x} = F_{x,FL} + F_{x,FR} + F_{x,RL} + F_{x,RR} - F_{aero} \quad (32)$$

$$m\ddot{y} = F_{y,FL} + F_{y,FR} + F_{y,RL} + F_{y,RR} \quad (33)$$

$$I_z\ddot{\psi} = L_f(F_{y,FL} + F_{y,FR}) - L_r(F_{y,RL} + F_{y,RR}) - l_\omega(F_{x,FR} + F_{x,RR} - F_{x,FL} - F_{x,RL}) \quad (34)$$

$$I_x\ddot{\theta} = l_\omega(F_{z,FL} + F_{z,RL} - F_{z,FR} - F_{z,RR}) + Z(F_{y,FL} + F_{y,FR} + F_{y,RL} + F_{y,RR}) \quad (35)$$

$$I_y\ddot{\phi} = L_r(F_{z,RL} + F_{z,RR}) - L_f(F_{z,FL} + F_{z,FR}) - Z(F_{x,FL} + F_{x,FR} + F_{x,RL} + F_{x,RR}) \quad (36)$$

Where F_x and F_y are the longitudinal and lateral tire forces respectively. F_z is given in equation(37) and represents the displacement of suspension that models the vertical dynamics and load transfer between the four tires. The indices (F,R) and (R,L) refer to front/rear and right/left wheels respectively. I_z , I_y and I_x represent the respective yaw, pitch and roll inertias of the vehicle.

$$F_z = -k_s\zeta(\theta, \phi) - d_s(\dot{\zeta})(\theta, \phi) \quad (37)$$

with $\zeta(\theta, \phi)$ being the displacement of the suspension for the given roll (ϕ) and pitch (θ) angles, while k_s and d_s define the suspension stiffness and damping.

The longitudinal and lateral tire forces are functions of the slip-ratio and side-slip angle as discussed in section 2.2, which is usually modeled by the Pacejka combined slip formula [21], taking into account the coupled interaction between longitudinal and lateral slipping. Note that this model only represents the chassis dynamics, More elaborate models may incorporate

engine and brake dynamics as well. In addition, it is assumed that the road remains flat which means that the effect of slope and banking angle can be neglected.

3 VEHICLE CONTROL

Automatic control is an essential task for autonomous driving systems, it solves the problem of following a set reference defined as a trajectory or a speed profile depending on the type of control. Control is a complex task for autonomous vehicles because it must ensure the stability of the vehicle and certain levels of performance. In general, the control task accounts for three aspects [22]:

- The type of control which can be lateral control, longitudinal control or both at the same time.
- The vehicle model used for implementing the control which can be kinematic or dynamic and linear or nonlinear.
- The control strategy which can be coupled, coordinated, in cascade, etc.

Lateral control deals with the yaw motion of the vehicle by acting on the steering angle of the front wheels, the goal is to follow a trajectory by reducing the angular and lateral position errors to zero. The trajectory can be predefined and generated offline or it can be computed online such as for obstacle avoidance and lane change maneuvers [13]. Lane keeping is completely dependent on lateral control where lanes can be detected from camera footage using image processing techniques or advanced deep learning methods.

3.1 Model-based Controllers

Model-based lateral controllers require a model that describes the lateral motion of the vehicle, These models are usually kinematic or dynamic depending on the type of controller. Numerous studies have been carried out for lateral control, the following subsections present the most common model-based lateral controllers and the corresponding latest research.

3.1.1 Model predictive control

Model Predictive Control (MPC) is a well-established strategy known as receding horizon control. MPC is based on the formulation of an optimization problem in the form of a finite horizon open-loop optimal control problem. The main idea of MPC is in using the discrete model of the system for predicting its behavior into the future up to a certain time step called prediction horizon. A control input sequence is generated along several discrete time steps known as the control horizon. The control sequence solves the optimization problem by minimizing a cost function while obeying to certain

constraints. At each time step, a finite horizon window (prediction horizon) is shifted, and new measurements of the system and the environment are used to solve the optimization problem resulting in a new control sequence. The whole process is repeated iteratively and only the first control input is applied to the system. The advantage of MPC control is its ability to handle Multi-Input Multi-Output (MIMO) constraints that represent limitations on the actuator or physical and safety constraints. More details about MPC, its variants and its applications can be found in the following surveys [23], [24], [25] and [26].

In autonomous vehicle control, the MPC controller is generally composed of an optimizer and a vehicle model (see Fig. 4). The optimizer oversees finding the optimal input and minimizing the cost function. The type of MPC controller depends on the vehicle model used for prediction. Nonlinear Model Predictive Control (NMPC) is computationally expensive and often unsuitable for real-time applications while linear MPC provides a better choice for real-time applications at the expense of less precision.

The formulation of MPC requires a descriptive system model that captures its significant dynamics while still being simple enough for solving the optimization problem in real time. Consider the system dynamics presented in equations (28) and (29) (vehicle lateral dynamics model for the case of lateral control) with X and u being the state and input vectors. Since MPC requires a discrete model, the system can be discretized with a sampling time T_s giving:

$$X_{k+1} = A_k \cdot X_k + B_k \cdot U_k \quad (38)$$

$$Y_k = C_k \cdot X_k \quad (39)$$

Given the model above with $A_k \in \mathbb{R}^{n \times n}$, $B_k \in \mathbb{R}^{n \times m}$ and $C_k \in \mathbb{R}^{m \times n}$, the simple MPC controller is obtained

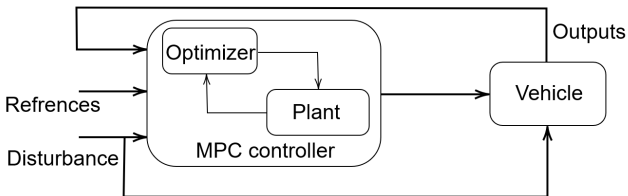


Fig. 4. MPC block diagram

as the following optimization problem:

$$\min_u J = \sum_{j=1}^{N_p} \|Y_p(k+j|k) - Y_{ref}(k+j|k)\|_{Q_y} + \quad (40)$$

$$\sum_{j=0}^{N_c-1} \|\Delta U(k+j|k)\|_{R_u}$$

$$\text{subj.to } X_{k+j+1|k} = A_k \cdot X_{k+j|k} + B_k \cdot U_{k+j|k} \quad (41)$$

$$X_{k|k} = X_k \quad (42)$$

$$Y_{k+j|k} = C_k \cdot X_{k+j+1|k} \quad (43)$$

$$U_{min} \leq U_{k+j|k} \leq U_{max} \quad (44)$$

$$\Delta U_{min} \leq \Delta U_{k+j|k} \leq \Delta U_{max} \quad (45)$$

$$X_{min} \leq X_{k+j|k} \leq X_{max} \quad (46)$$

$U \in \mathbb{R}^{m \times N_c}$ defines the optimized variable which is the control input, $Q_y \in \mathbb{R}^{N_p \times N_p}$ and $R_u \in \mathbb{R}^{N_c \times N_c}$ are weighting matrices for the outputs and the inputs. J is the cost function to be minimized in the form of a quadratic sum of weighted norms of the error and the input. The error is the difference between the predicted state $Y_p \in \mathbb{R}^{m \times N_p}$ and the reference Y_{ref} along the prediction horizon N_p and the input vector U is optimized along the control horizon N_c . Equations (42-46) specify the constraints which can be equality or inequality constraints and can be imposed on the outputs, the inputs and their rates. The weighted norms $\|Y_p(k+j|k) - Y_{ref}(k+j|k)\|_{Q_y}$ and $\|\Delta U(k+j|k)\|_{R_u}$ are defined as $(Y_p(k) - Y_{ref}(k))^T \cdot Q_y \cdot (Y_p(k) - Y_{ref}(k))$ and $\Delta U^T \cdot R_u \cdot \Delta U$ respectively, with Q_y and R_u being diagonal matrices whose weights penalize the corresponding variables. The cost function J can therefore be reformulated in a more convenient matrix form:

$$J_k = (Y_p(k) - Y_{ref}(k))^T \cdot Q_y \cdot (Y_p(k) - Y_{ref}(k)) + \Delta U^T \cdot R_u \cdot \Delta U \quad (47)$$

with:

$$Y_{ref} = [y_{ref}(k), y_{ref}(k+1), \dots, y_{ref}(k+N_p-1)]$$

$$Y_p = [y_p(k), y_p(k+1), \dots, y_p(k+N_p-1)]$$

$$\Delta U = [\delta u_k, \delta u_{k+1}, \dots, \delta u_{k+N_c-1}]$$

Solving the MPC problem is then resolved by finding the control sequence U that minimizes the cost function J and obeys to the constraints listed in equations (41-46).

Model predictive control is very suitable for the lateral control of autonomous vehicles since it can systematically handle constraints, several studies have been published recently that show the good performance achieved by MPC and its ability to overcome model imprecisions and external disturbances. In [27] Yao *et*

al. proposed an MPC path tracking controller with longitudinal speed compensation to overcome the assumption of constant longitudinal speed along the control horizon. The speed compensation aims to reduce the control deviation due to rapid speed and acceleration variations which affect the path tracking accuracy. The authors used the single-track bicycle model with Fiala brush tire model [9] in the design of MPC controller with constraints set on the tire cornering capabilities, the vehicle side-slip angle, the input and input rate. The compensation for the longitudinal speed is ensured by varying the velocity with constant acceleration. To check the performance of the designed controller, the authors used **MATLAB** and **Carsim** to simulate the system and compared it with an MPC controller without speed compensation. The simulations showed similar performance during gentle curve variations, but the designed controller performed much better in sudden curvature variations, where lateral and course deviation were significant for the normal MPC leading to greater tracking errors.

Kebbati *et al.* [28] developed an adaptive and optimized MPC controller for path tracking. The controller was based on the linear bicycle dynamics model, the authors used Laguerre functions to reduce the computation load and perform long control and prediction horizons. The authors proposed an improved Particle Swarm Optimization (PSO) algorithm to optimize the parameters of the MPC controller including prediction, control horizons and the cost function weighting matrices (see Fig. 5), the PSO algorithm is further explained and used with PI controllers in [29]. Their proposed controller adapts its parameters to varying longitudinal velocity, trajectory and external disturbances through a lookup table approach. The authors tested their controller against the classic MPC and pure-pursuit with different velocity profiles and trajectories and under varying external disturbances. The results showed that the proposed MPC is indeed adaptive and has superior performance and robustness. The same authors further enhanced their controller design in [30] by using Adaptive Neuro-Fuzzy Inference System (ANFIS) and fully-connected neural networks to learn the optimal MPC parameters and perform online controller adaptation. The optimization was carried for variable velocity profiles, wind disturbance profiles and road adhesion coefficients. The goal is to readjust the MPC controller to these varying working conditions and external disturbances. Significant tracking improvements were achieved.

Alcala *et al.* developed in his work [31] a solution for the mixed lateral and longitudinal control for trajectory tracking. They divided the control strategy into a cascade control scheme with an internal layer to control the vehicle dynamics (velocity) and an external layer which controls the vehicle kinematics (position and orientation) as can be seen in Fig. 6. The controller of

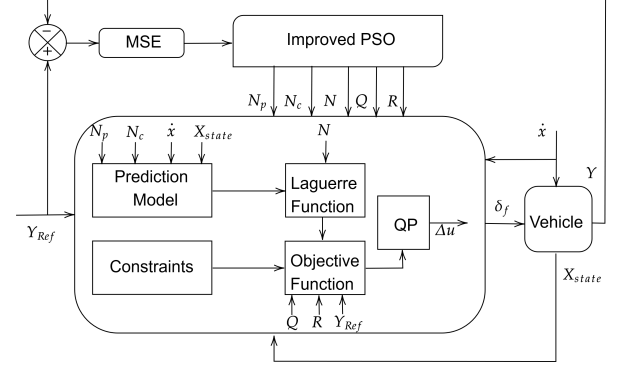


Fig. 5. Adaptive MPC control scheme [28]

the internal loop is a Linear Quadratic Regulator (LQR) formulated using Linear Parameter Varying (LPV) approach as a dynamic (LPV-LQR) design, it was transformed from the nonlinear model presented previously in [22]. The controller in the external loop was formulated as a quadratic optimization problem using the nonlinear kinematic model in an LPV form (Kinematic LPV-MPC), it controls the position and orientation of the vehicle. The designed control strategy was simulated under **MATLAB** for a curved circuit using the dynamics of an electric rear wheel drive car. The performances were compared to the NMPC controller, and the results showed that the LPV-MPC had similar performances to NMPC model except at handling external disturbances. On the other hand, the LPV-MPC proved to be 50 times faster than the NMPC, which is very important for real-time applications. Kebbati *et al.* [32] developed a coordinated lateral and longitudinal control strategy. The authors addressed the lateral control by designing an LPV-MPC controller that takes into account variable cornering stiffness coefficients, these were adapted online using recursive least squares estimators. The longitudinal control was ensured by a PSO-optimized Proportional Integral Derivative (PID) controller (PSO-PID), that guarantees accurate speed tracking. Speed and lateral dynamics stability conditions were satisfied with regard to road curvatures. Furthermore, the authors enhance the MPC cost function by adding an exponential weight which improved trajectory tracking precision.

The nonlinear model predictive control NMPC has been used in combination with multi-sensor fusion

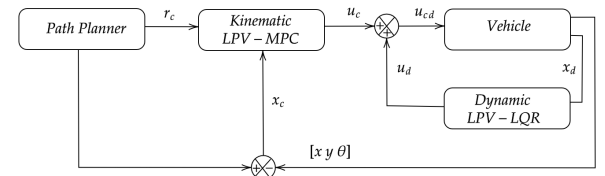


Fig. 6. Proposed cascade control scheme [31]

in [33], the authors aimed at exploring a parking lot autonomously and performing a parking maneuver. Three modules have been employed; a sensor fusion module for localization, mapping and obstacle detection, a decision-making module for detecting parking spaces and for guidance and a control module based on the NMPC. The kinematic single-track model was used in this study as it is sufficient for the low speeds and small external forces encountered in a parking maneuver. The different maneuvers were planned as a search for an optimal control problem that can be generalized as:

$$\begin{aligned}
& \min_{x,u,T} J(x,u,T) \\
& s.t. \\
& \dot{x}(t) = f(x(t), u(t)), \\
& x_{min} \leq x(t) \leq x_{max}, \\
& u_{min} \leq u(t) \leq u_{max}, \\
& \Psi_1(x(0), x(T)) = 0, \\
& \Psi_2(x(0), x(T)) \leq 0, \\
& C(x(t), u(t), t) \leq 0 \text{ for all } t \in [0, T].
\end{aligned} \tag{48}$$

The solution to this problem is the set of states x , controls u and time T that minimize the objective function J . The solutions obey the dynamics of f and the constraints set by Ψ_1 and Ψ_2 . Path constraints such as obstacle avoidance are defined by C . The authors applied a direct method using the software framework (TransWORHP [34]) to transform the optimal control problem into a Nonlinear Optimization Problem (NLP) and solve it with NLP solvers. Tests on a real autonomous vehicle (modified Volkswagen Passat GTE) showed that the computational load is heavy, but the implemented NMPC was able to solve the task of autonomously exploring the parking lot of the University of Bremen with high precision thanks to the consideration of the nonlinear dynamics of the vehicle. Li *et al.* [35] proposed a NMPC for trajectory tracking, the controller was based on nonlinear vehicle dynamics and Pacejka tire model [21], and it tracks the yaw angle and lateral position. The authors used the Sigmoid function to generate a smoothly curved reference. The NMPC is formulated as a standard optimization problem by discretizing the system dynamic equations, constraints were imposed on the control input and incremental control to prevent actuator saturation. The performance evaluation was done through simulations in **Carsim** and **MATLAB** by comparing the NMPC to the Linear Time Varying (LTV) version of MPC; (LTV-MPC). The simulation results showed better performance for NMPC against LTV-MPC which exhibited frequent oscillations with higher overshoots and less precision. However, the NMPC required very long computations compared to the LTV-MPC controller. LPV-MPC was used by Alcala *et al.* in [36] as a novel approach to perform autonomous racing, where an offline nonlinear model predictive planner was used to compute the optimal racing trajectory. The LPV-MPC approach was then

used online to track the generated trajectory, it uses the standard nonlinear dynamic bicycle model to capture vehicle dynamics. The proposed control approach was evaluated through simulations and real experiments in a proposed circuit, where the trajectory planner aims to minimize the lap time and the LPV-MPC aims to track the optimal trajectory. The results show fast lap times and good trajectory tracking with certain errors attributed to non-modeled dynamics. The main disadvantage of the proposed strategy is the required initialization of the LPV model, since the state space matrices of the LPV model change with the scheduling variables. In [37] Wang *et al.* proposed an improved MPC control strategy that includes an adaptive fuzzy controller to change the weights of the cost function. The aim of this design is to eliminate the problem of ride comfort caused by fixed weights in classical MPC. The fuzzy controller uses five fuzzy sets, it has as inputs the errors of the lateral position and the heading and generates the weight matrices for the cost function. The designed improved MPC was compared to pure-pursuit and classical MPC to evaluate its performance and ability to handle the ride comfort problem. The comparison was done through simulations in **Carsim** and **MATLAB** and the results proved that it is more accurate than the pure-pursuit controller while providing smooth steering angle changes compared to classical MPC especially with high errors. Xu *et al.* [38] designed a lane keeping system based on the LTV-MPC controller. In their work, the reference trajectory was generated by fitting five preview points to reduce the number of sensors needed. The LTV-MPC was based on the linearized bicycle model. The trajectory tracking was solved as a Quadratic Programming (QP) problem, where the solution is a vector of optimal steering angles. The LTV-MPC controller was evaluated through co-simulation in **MATLAB** and **Carsim**, and the results showed good tracking performance in low and high speeds. However, some acceptable overshoots in steering angles were produced at high speeds. Sun *et al.* [39] proposed an MPC with switched tracking error to tackle the tracking deviation and overshoot at high speeds. A fuzzy-logic classifier was used to determine the switching phase and instant and the controller used the velocity heading deviation as the tracking error instead of the vehicle heading deviation. The fuzzy classifier switches between the real-time side-slip and steady-state side-slip so that the MPC computes the velocity heading deviation in the transient and steady-state conditions respectively. The MPC controller was based on the single-track bicycle model with Fiala brush tire model to capture vehicle lateral dynamics. The latter was compared to regular MPC controller and MPC controllers with fixed side-slip (steady state or real-time). The designed controller achieved very high tracking accuracy. It scored the lowest tracking deviation RMS value, but failed to reject disturbances and model uncertainties. Guo *et al.* [40] proposed an MPC path following controller that considers road regions delimited

by curves and measurable disturbances accounting for model mismatch and small angle assumptions. The MPC was based on the bicycle dynamic model that includes measurable disturbances. The authors introduced a differential evolution algorithm to ensure faster online optimization. The proposed path following controller was tested on an experimental platform (HQ430 autonomous car) where it outperformed PID and regular MPC showing promising computation and control performances. The authors of [41] developed an online learning MPC controller for autonomous racing. They adopted a simple bicycle vehicle model and improved it by learning the model errors online using Gaussian process regression [42]. The control problem was formulated as a contouring-based MPC with a learning-based extension [43], and solved using FORCES Pro solver [44]. The authors used the AMZ autonomous car [45] to test the proposed learning-based controller where it achieved improved performance and 10% shorter lap times compared to the normal controller without the learning extension. Paper [46] also dealt with autonomous race driving using NMPC for the F1/10 platform, to keep the vehicle inside the track the authors interpolated the circuit boundaries using 3rd order polynomials and implemented them as inequality constraints on the lateral position. A feed-forward neural network was introduced in the paper to learn to mimic the NMPC using input-output data. In the same racing context, Verschueren *et al.* [47] handled autonomous racing as a time-optimal problem, in which a slip-free bicycle model is used to build the NMPC controller. The key objective is minimizing lap time, which is achieved by a spatial reformulation of the model to include time as an optimization variable. Simulation and experimentation was conducted using ACADO for solving the NLP. In [48], the same authors extended their work by including a full nonlinear dynamic bicycle model and using the same time-optimal control approach with spatial reformulation. In a similar approach, Kloeser *et al.* [49] addressed autonomous racing for a 1:43 scale race car using a singularity-free path parametric model for NMPC predictions. Contrary to [47], they used partial spatial reformulation of the model to exclude singularities. The authors implemented obstacle avoidance in the optimization problem as constraints with the objective of maximizing progress on the path. The designed controller was validated both in simulation and experimentation, leading to good racing performance. Paper [50] provides a thorough survey in the field of autonomous racing, where vehicles are driven at the limits of their performance with high speeds and accelerations.

3.1.2 Sliding mode controller

Sliding Mode Control (SMC) is a robust control technique suitable for nonlinear systems with parametric

uncertainties and subject to external disturbances. In SMC the states and control signals are not affected by the disturbances and uncertainties, because they are considered to be discontinuous. This technique forces the system to slide (converge) in a sliding surface (Fig. 7 where x is the controlled variable) by applying a fast switching control signal. The design of SMC controllers is a two-part task; the desired dynamics are defined by the sliding surface in the first part such that the design specifications are satisfied by the sliding motion. The second part is the design of a switching function that ensures the system slides in the desired sliding surface. The theory of SMC is reviewed in details in [51], [52] and [53].

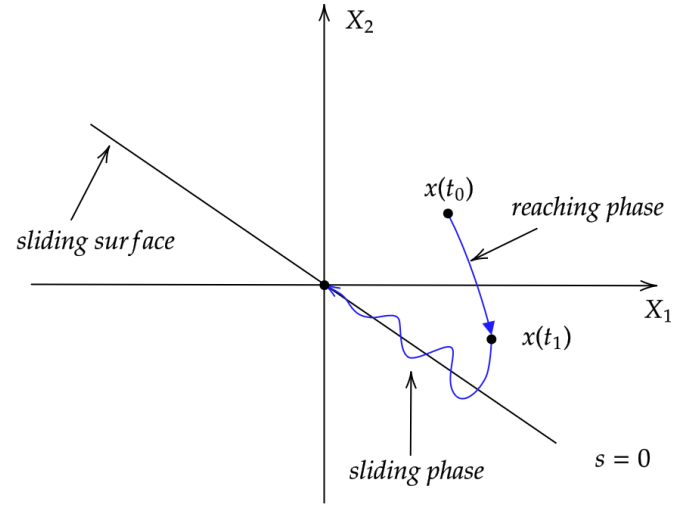


Fig. 7. SMC principle

The problem of lateral control with sliding mode has been studied in the literature [54], [55]. For instance, Talj *et al.* [56] addressed the lateral control of autonomous vehicles using a higher order sliding mode controller, the authors minimized the lateral displacement by using the super-twisting algorithm. The control problem can be formulated by considering equations (12, 13, 26) and (27) of the dynamic model given previously. From these equations, the model can be summarized as follows:

$$\begin{cases} \ddot{y} = -\frac{(C_f + C_r)}{mv_x} \dot{y} - \left(\frac{L_f C_f - L_r C_r}{mv_x} + v_x \right) \dot{\psi} + \frac{C_f}{m} \delta \\ \ddot{\psi} = -\frac{L_f C_f - L_r C_r}{I_z v_x} \dot{y} - \frac{L_f^2 C_f + L_r^2 C_r}{I_z v_x} \dot{\psi} + \frac{L_f C_f}{I_z} \delta \end{cases} \quad (49)$$

with y and ψ being the lateral position and the yaw angle of the vehicle. The super-twisting algorithm ensures robust stability and reduces the chattering phenomenon, let us consider a system of the form:

$$\dot{x} = f(t, x) + g(t, x)u(t) \quad (50)$$

with f and g being continuous functions, x and u the state vector and the control input respectively. Let us now define a sliding variable s with a derivative expressed as:

$$\dot{s}(t, s) = \phi(t, s) + \Phi(t, s)u(t) \quad (51)$$

The sliding mode controller aims for the system to converge to the sliding surface which can be defined as $s = 0$. To achieve this, we assume that there exist s_0 , b_{min} , b_{max} and C_0 as positive constants such that the system satisfies the following:

$$\begin{cases} x \in R^n, |s(t, x)| < s_0 \\ |u(t)| \leq U_{max} \\ 0 < b_{min} \leq |\phi(t, s)| \leq b_{max} \\ |\phi(t, s)| < C_0 \end{cases} \quad (52)$$

Thus the sliding mode control can be given as [56]:

$$u(t) = u_1 + u_2 \begin{cases} u_1 = -\alpha|s|^\tau \text{sign}(s), \tau \in]0, 0.5] \\ \dot{u}_2 = -\beta \text{sign}(s) \end{cases} \quad (53)$$

where α and β are positive constants, the convergence in finite time is very important and it can be assured by the conditions below:

$$\begin{cases} \beta > \frac{C_0}{b_{min}} \\ \alpha \geq \sqrt{\frac{4C_0(b_{max}\beta + C_0)}{b_{min}^2(b_{min}\beta - C_0)}} \end{cases} \quad (54)$$

To apply this to the lateral control of autonomous vehicles, we consider the lateral error dynamics ($\ddot{e} = \alpha_y - \alpha_{y_{ref}}$) where we assume that the desired lateral acceleration can be expressed as $\alpha_{y_{ref}} = \frac{v_x^2}{R}$ with R being the radius of the road curvature. Using equation (10) we obtain:

$$\ddot{e} = \ddot{y} + v_x \dot{\psi} - \frac{v_x^2}{R} \quad (55)$$

Replacing \ddot{y} by equation (49) yields:

$$\ddot{e} = -\frac{(C_f + C_r)}{mv_x} \dot{y} - \frac{L_f C_f - L_r C_r}{mv_x} \dot{\psi} - \frac{v_x^2}{R} + \frac{C_f}{m} \delta \quad (56)$$

Here δ represents the control input which is the steering angle, the objective is to cancel the displacement error. Therefore, the sliding surface can be chosen as $s = \dot{e} + \lambda e$. The derivative is then obtained as: $\dot{s} = \ddot{e} + \lambda \dot{e}$. By replacing \ddot{e} by its expression in equation (56) we obtain

the following:

$$\dot{s} = -\frac{(C_f + C_r)}{mv_x} \dot{y} - \frac{L_f C_f - L_r C_r}{mv_x} \dot{\psi} - \frac{v_x^2}{R} + \frac{C_f}{m} \delta + \lambda \dot{e} \quad (57)$$

Using identification between equations (57) and (51) we find that:

$$\begin{cases} \phi(t, s) = -\frac{C_f + C_r}{mv_x} \dot{y} - \frac{L_f C_f - L_r C_r}{mv_x} \dot{\psi} - \frac{v_x^2}{R} + \lambda \dot{e} \\ \Phi(t, s) = \frac{C_f}{m} \end{cases} \quad (58)$$

Hence, the super-twisting algorithm (equation 54) yields the following control signal:

$$\delta = u_1 + u_2 \begin{cases} u_1 = -\alpha|s|^{\frac{1}{2}} \text{sign}(s) \\ u_2 = -\beta \text{sign}(s) \end{cases} \quad (59)$$

The downside of SMC technique is its application limit due to the fast switching which results in the chattering phenomenon. In reality, the actuators suffer from imperfections and delays, therefore, the chattering from SMC method may damage the actuators, and results in discomfort or energy loss which also creates additional disturbances [19]. Researchers have adopted several methods to deal with this phenomenon, these methods include using higher order sliding mode [57], the use of smooth functions to replace the discontinuous function (sign) and the inclusion of observers [56].

Many researchers adopted the SMC control technique to perform lateral control of autonomous vehicles, For instance, Wu *et al.* [58] designed a path following strategy based on SMC. They used the non-singular terminal sliding mode technique. The control problem was simplified into a yaw tracking problem to ensure that the displacement deviation converges to zero. The linearized dynamic bicycle model was used in the design of the SMC controller. Fig. 8 shows the path following algorithm where the linear-ESO is used to estimate the state and disturbance of the simplified yaw tracking system which ensures that the control algorithm works well even without an accurate mathematical model. The NTSM block in Fig. 8 is the SMC controller based on non-singular terminal sliding surface. Moreover, the SMC controller was coupled with an exponential convergence law to improve the convergence speed. Simulations in *Carsim* showed solid performance of the designed SMC controller and proved its abilities to overcome uncertainties, reject disturbances and achieve very small tracking errors.

Kada *et al.* [59] proposed a robust SMC that eliminates the drawbacks of regular SMC controllers. The authors used a disturbance observer to enhance the controller, a neural network was used to estimate model uncertainties and a fuzzy logic system was added to handle parameter

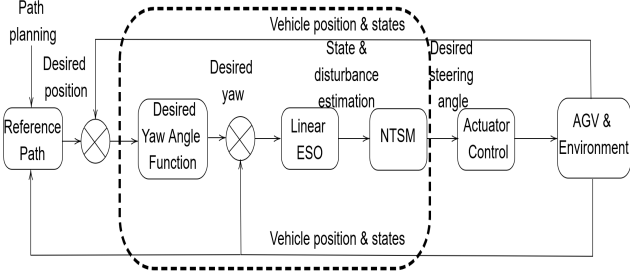


Fig. 8. Path following control algorithm [58].

variations. The disturbance observer which is detailed in [60], estimates the mismatch disturbances such as wind and unmodeled dynamics. The model uncertainties are estimated through a Radial Basis Function Neural Network (RBFNN) formed by three layers. The algorithm of this neural network approximates the different functions of the model. The fuzzy system was used for scheduling the gains of the sign function in the SMC controller [61]. The design of the controller is detailed in [59] with the chosen sliding surface and the stability analysis. For performance evaluation, the authors compared their controller to traditional SMC, observer-based SMC and back-stepping SMC. The proposed controller ensured very fast convergence thanks to gain scheduling with the fuzzy system. In addition, the controller provided very precise tracking that easily overcame the other controllers even in sharp turns while eliminating the chattering.

The authors of [62] designed an adaptive distributed sliding mode controller for vehicular platooning. A distributed sliding surface for all the followers was used to control the full platoon. To ensure ride comfort, the authors employed an adaptive law instead of the switching function to compensate for model errors and avoid jerks. They proposed a structural decomposition method to decouple the sliding motion dynamics of the platoon. Moreover, they ensured the smoothness and quickness of the control by placing the poles of the sliding motion in predefined regions using Linear Matrix Inequality (LMI) approach. It was found that the performance of the designed controller is satisfactory even though the platoon is composed of vehicles with uncertain parameters. However, the control performance was heavily affected by the interaction topology of the platoon. Norouzi *et al.* [63] designed an adaptive sliding mode controller for the task of autonomous lane change, the authors used a fuzzy boundary layer to address the chattering issues of SMC. The path planning for the lane change was performed using mathematical quintic function and vehicle boundary conditions, and the vehicle lateral dynamics were modeled using the bicycle model. The authors tested their controller in *Carsim* under different road conditions and compared it to regular SMC. The results showed a significant improvement, the use of adaptive rules improved the

controller such that the range of uncertainties were no longer needed to be known and the use of the fuzzy boundary layer achieved smaller tracking errors. Wang *et al.* [64] also worked on the path tracking problem, where they proposed an automatic steering control strategy based on the back-stepping sliding mode control. For higher precision modeling, the authors used both kinematic and dynamic vehicle models as well as a vehicle-road system model. To address the chattering problem, the authors replaced the sign function of the SMC by a saturation and introduced a sliding mode term that overcomes disturbances and ensures the system state is always on the sliding surface which in turn guarantees robustness. The proposed strategy was evaluated in *Carsim* and on a real vehicle at low speed and closed road conditions, the results showed that the proposed strategy significantly improved the steering performance. In the same context, Alika *et al.* [65] improved the work presented in [66] which addresses the lateral control of autonomous vehicles using a higher order SMC with the super twisting algorithm. The authors enhanced the SMC by optimizing the controller performance using the PSO algorithm. The authors perform the optimization for two cases; first they optimize the sliding surface parameters and then the controller parameters. The conducted simulations showed the proposed controller outperformed the SMC developed in [66] and guarantees much better trajectory tracking and more robustness.

3.1.3 H_∞ controller

H_∞ is a robust control method that easily handles modeling uncertainties and external disturbances. The objective is to minimize the H_∞ norm of the system to be controlled by solving an optimization problem that involves the Riccati equation. The control problem formulation usually accounts for noise, modeling uncertainties and disturbances and is solved through LMI methods. Although the H_∞ approach is known for robust stability and performance, it may easily become complex to realize in practice. More details about H_∞ control theory and design can be found in [67]. To apply the lateral control task based on H_∞ a vehicle-road system model is required (Fig. 9). The vehicle dynamics model given in equations (11-15) can be adopted along with external disturbances as follows:

$$\begin{cases} \dot{\beta} = a_{11}\beta + a_{12}\gamma + b_{11}\delta_f + b_{12}\delta_r + d_1 \\ \dot{\gamma} = a_{21}\beta + a_{22}\gamma + b_{21}\delta_f + b_{22}\delta_r + d_2 \end{cases} \quad (60)$$

where the terms a_{ij} and b_{ij} are given by:

$$\begin{cases} a_{11} = -\frac{C_f + C_r}{v_x m}, a_{12} = -\frac{l_f C_f - l_r C_r}{v_x^2 m} - 1 \\ a_{21} = -\frac{l_f C_f - l_r C_r}{I_z}, a_{22} = \frac{l_f^2 C_f + l_r^2 C_r}{v_x I_z} \\ b_{11} = \frac{C_f}{v_x m}, b_{12} = \frac{C_r}{v_x m}, b_{21} = \frac{l_f C_f}{I_z}, b_{22} = \frac{b C_r}{I_z} \end{cases}$$

To eliminate either the lateral error (e) or the heading error ($\Delta\Phi$), the state variables relevant to the reference path are obtained from the state variables of the vehicle dynamics as follows:

$$\begin{cases} \dot{s} = v_x \cos(\Delta\Phi) - v_y \sin(\Delta\Phi) \\ \dot{e} = v_y \cos(\Delta\Phi) + v_x \sin(\Delta\Phi) \end{cases} \quad (61)$$

with s being the distance along the path, v_y and v_x being the lateral and longitudinal velocities respectively. Paper [68] adopted the projected error e_p to combine both lateral and heading errors at once:

$$\begin{cases} e_p = e + x_p \sin(\Delta\Phi) \\ \Delta\Phi = \Phi - \Phi_r \end{cases} \quad (62)$$

By using small angle approximation and differentiating e_p we obtain:

$$\begin{cases} \dot{e}_p = v_x \beta + v_x \Delta\Phi \\ \dot{e}_p = \dot{e} + x_p \dot{\Delta\Phi} \\ \dot{\Delta\Phi} = \dot{\Phi} - \dot{\Phi}_r \end{cases} \quad (63)$$

Hence, we can obtain the following:

$$\begin{cases} \dot{e}_p = v_x \beta + x_p \gamma + v_x \Delta\Phi - x_p k \dot{s} + d_3 \\ \dot{\Delta\Phi} = \gamma - k \dot{s} + d_4 \end{cases} \quad (64)$$

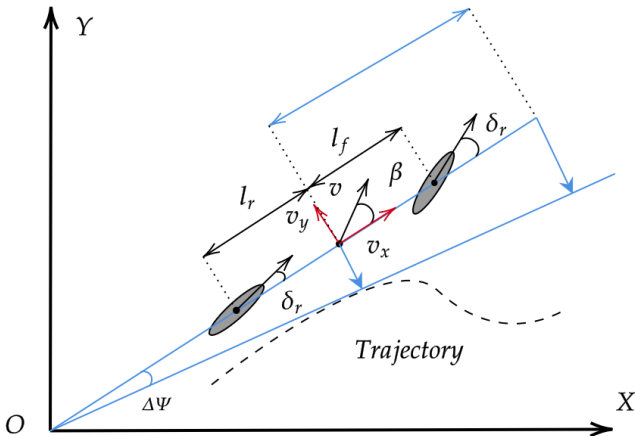


Fig. 9. Vehicle-road system

where d_3 and d_4 denote uncertain external disturbances, and k represents the path curvature. The following state space system with external disturbances is obtained by combining equations (60) and (64):

$$\begin{aligned} \dot{x} &= A x + B u + w \\ y &= x \end{aligned} \quad (65)$$

$$\text{where } A = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ v_x & x_p & 0 & v_x \\ 0 & 1 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$x = [\beta \ \dot{\beta} \ e_p \ \Delta\Phi]^T$ represents the state vector, $u = [\delta_f \ \delta_r]^T$ is the control vector (general case where both wheels are steerable) and $w = [d_1 \ d_2 \ d_3 \ -x_p K \dot{s} \ d_4 - K \dot{s}]^T$ is the uncertain external disturbance which can be defined as $w = B d_e$ with $d_e = [d_{ef} \ d_{er}]^T$ being the external disturbance acting on the front and rear axles. The robust feedback control law can then be expressed as:

$$u = K x \quad (66)$$

with K being a feedback gain matrix, for a positive symmetric matrix P , the Lyapunov function can then be selected as :

$$L = x^T P x \quad (67)$$

which when differentiated yields the following:

$$\begin{aligned} \dot{L} &= \dot{x}^T P x + x^T P \dot{x} \\ &= (Ax + Bu + Bd_e)^T P x + x^T P (Ax + Bu + Bd_e) \\ &= (Ax + BKx + Bd_e)^T P x + x^T P (Ax + BKx + Bd_e) \\ &= x^T Q^T x + x^T Q x + (Bd_e)^T P x + x^T P B d_e \end{aligned} \quad (68)$$

where $Q = P(A + BK)$. Then by setting $\lambda = [x^T d_e^T]^T$, one can obtain the following:

$$\begin{aligned} x^T Q^T x + x^T Q x &= x^T Q x \\ &= \lambda^T \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix} \lambda \end{aligned} \quad (69)$$

$$(Bd_e)^T P x + x^T P (Bd_e) = \lambda^T \begin{bmatrix} 0 & PB \\ B^T P & 0 \end{bmatrix} \lambda \quad (70)$$

Equation (68) can be reformulated by combining (69)

and (70):

$$\begin{aligned}\dot{L} &= \lambda^T \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix} \lambda + \lambda^T \begin{bmatrix} 0 & PB \\ B^T P & 0 \end{bmatrix} \lambda \\ &= \begin{bmatrix} Q & PB \\ B^T P & 0 \end{bmatrix} \lambda\end{aligned}\quad (71)$$

at this point, the performance index of the H_∞ can be selected based on an attenuation level $\rho > 0$:

$$\int_0^t y^T y dt \leq \rho^2 \int_0^t d_e^T d_e dt \quad (72)$$

$$\begin{aligned}y^T y dt - \rho^2 d_e^T d_e dt &= x^T x dt - \rho^2 d_e^T d_e dt \\ &= [x^T d_e^T] \begin{bmatrix} I & 0 \\ 0 & -\rho^2 I \end{bmatrix} \begin{bmatrix} x \\ d_e \end{bmatrix} \\ &= \lambda^T \begin{bmatrix} I & 0 \\ 0 & -\rho^2 I \end{bmatrix} \lambda\end{aligned}\quad (73)$$

adding (71) to (73) yields the following equation:

$$\dot{L} + y^T y dt - \rho^2 d_e^T d_e dt = \lambda^T \begin{bmatrix} Q + I & PB \\ B^T P & -\rho^2 I \end{bmatrix} \lambda \quad (74)$$

and by setting $\begin{bmatrix} Q + I & PB \\ B^T P & -\rho^2 I \end{bmatrix}$ and integrating (74) we obtain:

$$\begin{aligned}L &\leq \rho^2 \int_0^t d_e^T d_e dt - \int_0^t y^T y dt + L(0) \\ &\leq \rho^2 \int_0^t d_e^T d_e dt + L(0)\end{aligned}\quad (75)$$

Thus by using equation (67), it can be proven that system states are bounded under bounded disturbance and finite time horizon as follows:

$$\begin{aligned}P_{min} \|x\|^2 &\leq x^T P x \leq \rho^2 \int_0^t d_e^T d_e dt + L(0) \\ \|x\|^2 &\leq \frac{\rho^2 \int_0^t d_e^T d_e dt + L(0)}{P_{min}}\end{aligned}\quad (76)$$

using Schur complement theorem one can write:

$$\begin{bmatrix} Q + I & PB \\ B^T P & -\rho^2 I \end{bmatrix} = \begin{bmatrix} Q + I & PB & I \\ B^T P & -\rho^2 I & 0 \\ I & 0 & -I \end{bmatrix} < 0 \quad (77)$$

and based on equations (68) and (69) we obtain the following:

$$\begin{bmatrix} PA + PBK + A^T P + K^T B^T P & PB & I \\ B^T P & -\rho^2 I & 0 \\ I & 0 & -I \end{bmatrix} < 0 \quad (78)$$

$$\text{setting } S = \begin{bmatrix} P^{-1} & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}, F = KP^{-1} \text{ and } R = P^{-1}$$

and multiplying equation (78) by S we obtain the first LMI (79), and considering that P is a positive symmetric matrix yields the second LMI (80):

$$\begin{bmatrix} AR + BF + RA^T + F^T B^T & B & R \\ B^T & -\rho^2 I & 0 \\ R & 0 & -I \end{bmatrix} < 0 \quad (79)$$

$$R > 0 \quad (80)$$

The H_∞ state feedback gain is obtained by solving the above mentioned LMIs. Several research papers dealing with H_∞ for the lateral control task of autonomous vehicles exist in the literature. For instance, In [69], Sun *et al.* developed a fuzzy model-based H_∞ controller with feedforward for the lateral control of autonomous vehicles. The authors considered the desired yaw rate as external disturbance of the vehicles dynamics and used the feedforward loop to improve the performance of the closed-loop system. The authors used the bicycle kinematic model for the feedforward loop and the bicycle dynamic model for the feedback loop. Fuzzy modelling was used to express the variation of vehicle velocity and mass which can change with the number of passengers onboard. The effectiveness of the proposed controller was demonstrated by **Simulink-Carsim** co-simulations. He *et al.* [68], proposed a robust H_∞ coordination control strategy to improve lateral motion control at handling limits. The approach was based on LMI method and coordinates front and rear wheel steering. The proposed controller was verified on dry asphalt through **Simulink-Carsim** co-simulations where external disturbances on the front and rear vehicle axles were included. The results showed high path tracking capabilities close to driving limits and proved that

the proposed approach is robust to uncertain external disturbances. Hu *et al.* [70], proposed a robust H_∞ control strategy for path following where the vehicle lateral velocity is not required to be known. The developed output feedback controller was based on a mixed approach involving Genetic Algorithms (GA) and LMI to obtain the optimal feedback gain, and it accounts for model and road uncertainties and external disturbances. The designed controller was tested for a lane change maneuver with low adherence road and for a J-turn maneuver with high adherence road under tough conditions, and the results proved good performance and robustness in general. An LPV H_∞ controller for high-speed driving and evasive maneuvers was designed in [71] by Corno *et al.* Contrary to the feedforward approach, the design was based on the lateral error and look-ahead distance of the vehicle to ensure better robustness, moreover the actuator nonlinearities under low speeds were accounted for. The proposed controller was experimentally validated on an instrumented vehicle for different bends and double lane change maneuvers where sufficient and adequate performance for autonomous driving was achieved. Chaib *et al.* [72] compared H_∞ to self-adaptive, fuzzy and PID controllers for the lateral control task. The H_∞ controller was designed based on the loop shape procedure and tested against the others under variable road adhesion and under lateral wind disturbances. The simulation results favored the self-adaptive controller, however, the H_∞ along with fuzzy controller performed just as well while PID was found to be the worst. Yang *et al.* [73] compared H_∞ with MPC for trajectory tracking control using a 3DoF vehicle dynamics model. The two controllers were tested for curves and double lane change maneuvers in MATLAB-Carsim co-simulations. The results showed that MPC outperformed H_∞ in terms of accuracy and response time, but when tough conditions were introduced H_∞ was able to achieve higher longitudinal speeds compared to its rival. Wang *et al.* [74] developed a robust H_∞ controller that accounts for estimation and measurement delays and data dropouts. The authors formulated a general representation to include measurement and transmission delays and data dropouts. The design of the proposed controller was enhanced by including varying external disturbances. In addition, the uncertainties due to tire cornering stiffness coefficients were also included. Simulation results verified the effectiveness of the proposed approach, where good tracking was achieved even though the authors focused mainly on time invariant delays and data dropouts which can be time varying and unknown in real life.

3.1.4 Linear quadratic regulator

The LQR approach is an optimal control technique that achieves great results at minimum cost. The idea is similar to MPC where a quadratic cost function is

minimized resulting in an optimal feedback gain, this is achieved by solving the Riccati equation which is formulated based on the plant's state space model.

In lateral control applications, designing the LQR controller is a straight forward task. Considering the vehicle dynamics model given in equations (28,29), an error-based dynamic model can be deduced. Assuming e_1 is the lateral error (between trajectory and vehicle lateral position) and e_2 is the heading error (between vehicle and reference heading angles), the following relations hold true:

$$\begin{cases} \ddot{y} = \ddot{e}_1 - \dot{x}(\dot{\psi} - \dot{\psi}_{ref}) \\ \dot{y} = \dot{e}_1 - \dot{x}(\psi - \psi_{ref}) \\ \dot{\psi} = \dot{e}_2 - \dot{\psi}_{ref} \end{cases} \quad (81)$$

Combining equations (28) and (81) while considering $e_1, \dot{e}_1, e_2, \dot{e}_2$ as the new states with δ_f being the input, the error-based model can be defined by:

$$\begin{cases} \dot{X} = AX + BU + E\dot{\psi}_{ref} \\ Y = CX \end{cases} \quad (82)$$

where ψ_{ref} is the reference heading and the new matrices are given by:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -2\frac{C_f+C_r}{m} & 2\frac{C_f+C_r}{m\dot{x}} & -2\frac{C_f l_f - C_r l_r}{m\dot{x}} \\ 0 & 0 & 0 & 1 \\ 0 & -2\frac{C_f l_f - C_r l_r}{I_z \dot{x}} & 2\frac{C_f l_f - C_r l_r}{I_z} & -2\frac{C_f l_f^2 + C_r l_r^2}{I_z \dot{x}} \end{bmatrix},$$

$$B = \begin{bmatrix} 0 \\ 2\frac{C_f}{m} \\ 0 \\ 2\frac{C_f l_f}{I_z} \end{bmatrix}, \quad E = \begin{bmatrix} 0 \\ -2\frac{C_f l_f - C_r l_r}{m\dot{x}} \\ 0 \\ -2\frac{C_f l_f^2 + C_r l_r^2}{I_z \dot{x}} \end{bmatrix}.$$

The role of LQR optimal controller is to minimize the lateral error and the vehicle lateral acceleration. Similar to MPC, LQR solves an optimization problem constructed with the following objective function:

$$\text{argmin } J = \int_0^\infty X^T Q X + u^T R u dt \quad (83)$$

with Q being semi-positive definite diagonal weight matrix, and R being a constant matrix penalizing the control effort. As LQR is a state feedback control, the control signal is given in (66), where K is the feedback gain obtained by the variational method:

$$K = (R + B^T P B)^{-1} B^T P A \quad (84)$$

with matrix P being the solution of the Riccati equation:

$$A^T P + P A - P B R^{-1} B^T P + Q = 0 \quad (85)$$

Paper [75] developed an LQR-based lateral controller with road disturbance compensation, the road was modeled using clothoid cubic polynomials and a dynamic lateral model with look-ahead distance was adopted. The authors redesigned the state reference by adding an additional term to compensate for the winding road disturbance. Experimental tests showed superior performance of the LQR with disturbance compensation compared to regular LQR controller. Riofrio *et al.* [76] designed an LQR that handles lateral stability and rollover control taking into account the road bank angle which is estimated using a Kalman filter. Several tests were carried out in Trucksim where the proposed LQR controller was compared to a fuzzy controller. The results showed significant improvements in load transfer, yaw rate, roll and side-slip angles. Good vehicle behaviour was achieved in banked surfaces, and the designed controller was able to prevent over-steering effects often caused by rollover controllers. Paper [77] conducted a comparative study between LQR, MPC and Stanley controller for path tracking applications. the kinematic bicycle model was used and the controllers were tested in **CarMaker** simulator. LQR and MPC provided optimal results compared to Stanley. MPC outperformed LQR at the expense of higher computation loads. Articulated heavy duty vehicles have large sizes and experience important mass variations which result in poor maneuverability. In this regard, Barbosa *et al.* [78] improved a robust LQR (RLQR) previously developed in [79, 80] to take into account parametric variations. The proposed controller does not depend on offline parameter tuning, but rather uses an over-time vanishing penalty parameter to foresee smoothness and robustness and maintain optimality. The proposed controller was evaluated against robust H_∞ where the vehicle was subject to variable loads. The results showed smoother and more robust performance of RLQR, especially for high loads with mass uncertainties. Kim *et al.* [81] also worked on the lateral control of articulated vehicles, the authors designed an active steering controller based on LQR theory. The articulated vehicle was modelled by a 3DoF model where key parameters were identified with simulated annealing particle swarm optimization (SAPSO). The proposed LQR controller aims at minimizing yaw tracking error and side-slip angle simultaneously for both tractor and trailer. The performance of the proposed control strategy was validated through Trucksim simulations and the results were satisfactory at low and high speed maneuvers. Paper [82] addressed the lateral control of

autonomous vehicles using a Lyapunov-based method with an LQR-LMI tuning approach. The controller was developed with an LPV kinematic bicycle model, and the resolution of the LQR-LMI determined optimal control parameters. To further complement the tuning process, the authors added a pole placement constraint to guarantee maximum achievable performance. The proposed control technique achieved good autonomous guidance in virtual reality and real scenario tests.

3.1.5 Other control methods

Other controllers include PID control which is often enhanced with neural networks or fuzzy logic systems to be adaptive to external disturbances and varying parameters [83–86]. Back-stepping control which is a recursive method that associates Lyapunov function with feedback control [87–90]. Geometry-based controllers like the pure pursuit [91–93] and Stanley controller [94–96]. Several other techniques are developed in the literature. For instance, Chen *et al.* [97] worked on autonomous parking, they introduced an adaptive pseudo-spectral method for solving time and energy optimal control for parking applications, which is handled as a nonlinear programming problem. The approach was compared to interior point and piecewise Gauss pseudo-spectral resolution methods. Simulations showed higher computation efficiency and better control accuracy. Li *et al.* [98] used the hierarchical control method to reduce tire slip energy and improve vehicle stability with time-varying tire cornering stiffness and tire slip energy torque distribution. The authors handled tire slip energy with a scheme of semi-empirical tire slip energy model used with the previously mentioned hierarchical control, which ensures active steering and yaw control. Simulation results proved that considering time-varying cornering stiffness and tire slip energy has significant advantages on the control performance. Paper [99] proposed a disturbance observer-based state-error port-controlled Hamiltonian (DOBC-SEPCH) control strategy to optimize energy consumption and enhance tracking performance of unmanned surface vehicles. A disturbance observer is introduced to estimate environmental perturbations coupled with an energy based controller, designed using SEPCH technique. The performance of the proposed approach was validated in simulations with stability insurance. Several other geometric, kinematic and dynamic controllers are reported in [18].

3.2 Controllers Based on Artificial Intelligence

Besides model-based controllers, model-free controllers provide an alternative solution when modelling becomes complex or inaccurate. Such type of control strategies are mostly data-driven, data coming from different sensors like camera, Lidar and IMU is exploited to

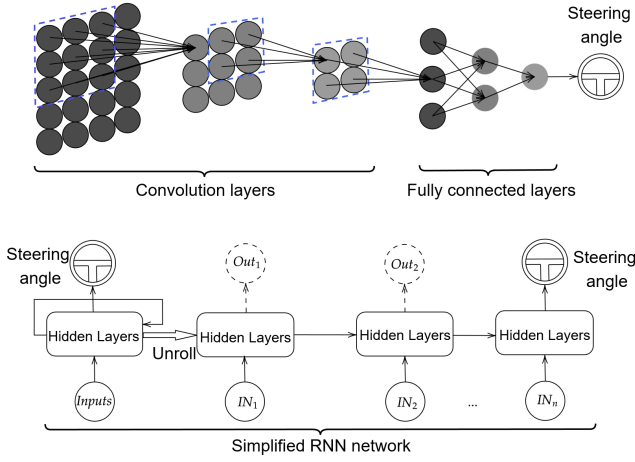


Fig. 10. Examples of neural networks.

predict control signals for the steering, acceleration or braking actions.

3.2.1 Supervised learning control

Neural networks have seen huge advancements recently [100], especially in the field of control. Autonomous driving applications are mostly based on fully connected neural networks [101], Convolutional Neural Network (CNN) [102] and Recurrent Neural Network (RNN) [103].

Fully connected networks are composed of multiple layers of neurons interconnected by synaptic weights, they are inspired from brain neural networks. Generally, they consist of one input layer, one output layer and one or more hidden layers which define the depth of the network. The training process of neural networks is mostly based on the gradient descent algorithm [104]. The forward pass propagates the inputs throughout the network layers, where each neuron normalizes its output by an activation function like the Sigmoid [105] or the ReLU [106]. The prediction error at the output layer is back propagated through the network to readjust its weights. CNNs differ from fully connected networks as they contain convolution layers whose neurons are connected to only some neurons of the next layer. Weights are shared between multiple connections, which greatly reduces network complexity in terms of trainable parameters leading to faster training. On the other hand, RNNs are able to retain memory. They contain feedback loops where layer outputs are not only sent to the next layer but also sent back to the previous layers. Common training problems in these types of neural networks include the vanishing and exploding gradients [107], making the task of hyper parameter tuning more difficult. A typical example of each network is illustrated in Fig. 10.

For instance, paper [108] used deep learning to control

vehicle lateral and longitudinal motion. The authors used CNN in two different models; the first model tries to handle both control tasks at once, and the second model consists of two CNN-based controllers for the steering and the speed control. Although different architectures with varying complexities were tested in the first model, none of them was able to handle the task and find patterns between input images and output control signals. However, using two separate CNN networks managed to achieve acceptable results where full autonomy was obtained on two different racing tracks. The authors used TORCS simulator for data collection and control tests. A complex CNN network was required for the steering task, while the speed control network was much simpler. Nevertheless, data collection and training are tedious tasks in such strategies. Jhung *et al.* [109] proposed end-to-end steering control with CNN-based closed-loop feedback. The authors divided the training process into supervised pre-training and reinforced closed-loop post-training where the latter improves the control performance. The proposed controller was tested in simulators and on a real autonomous car and was able to perform lateral control with acceptable errors. Paper [110] used CNNs and fully connected networks to develop control strategies for the coupled longitudinal and lateral control. The authors used a high fidelity vehicle model to generate a data-set for training, the inputs to the network are the trajectory and vehicle states and the outputs are the steering angle and the applied torque on the wheels. Evaluation tests showed that both control strategies were able to handle the combined longitudinal and lateral control tasks, but the CNN-based controller performed better than the one based on Multi-layer Perceptron (MLP). Rausch *et al.* [111] also used CNNs to develop a steering controller, the network comprised three convolution layers with one fully connected layer. Steering angle data and front facing camera footage were used to train the network to learn appropriate steering controls. The data was generated from simulations. In [112] an RNN was combined with CNN to predict steering controls from input images, the RNN adds temporal dependencies which enhanced the predictions and control performance by 35% compared to using only CNNs. These methods are known as supervised learning, since the data used to teach the network is labeled by an expert.

3.2.2 Reinforcement learning control

Reinforcement learning is an unsupervised approach where the network learns from trial and error, this is usually modeled as a Markov Decision Process (MDP) illustrated in Fig. 11. In a general way, an MDP is defined by a state space S , an action space A , a state transition probability P and a reward function R . The goal is to learn a policy $\pi(s_t, a_t)$ that links rewards and actions such that at each iteration, the agent observes

the set of states s_t and takes an action a_t from the action space A . Then the environment transitions according to probability model P . The agent then observes a new set of states s_{t+1} and receives a reward r_t for the performed action a_t . The policy π seeks to find the actions that maximize the rewards enabling the agent to learn by applying actions and receiving rewards or penalties.

One of the most widely used reinforcement learning algorithms is Q-learning [113], the latter is a value-based approach where the reward is quantified by a value function $V(s)$ and the objective of the algorithm is to map each action to a value for each state. Another type of algorithms is based on policy gradient, it parameterizes the policy instead of the value function to obtain the optimal policy. This is achieved through a loss function, whose gradient is estimated with regards to network parameters. Thus, the network parameters are adjusted according to the policy gradient. There are actor-critic algorithms as well, these use a hybrid approach combining value-based and policy gradient algorithms. Hence, one network estimates the reward for each action and another actor network estimates the optimal policy. Paper [114] combined deep learning and reinforcement learning to steer an autonomous vehicle, the authors used a multi-task learning CNN to learn track features and localise the vehicle from camera images, they employed policy gradient reinforcement learning to control the steering wheel on the virtual TORCS environment [115]. In [116] LIDAR data was used to develop and test reinforcement learning controllers based on policy gradient and Q-learning for obstacle avoidance maneuvers.

Paper [117] used reinforcement learning to perform lane change maneuvers, the authors designed a Q-function approximator with closed loop greedy policy to improve the computation efficiency. Evaluation tests showed that smooth and efficient control policies were achieved for lane-change maneuvers. The same authors developed a quadratic Q-network for autonomous driving in [118], the quadratic form achieved more optimal control actions that were tested in lane-change maneuvers as well. In [119], Deep Q-learning and Actor critic algorithms were used to learn optimal policies for lateral control. The developed controllers were tested in TORCS

while considering interactions with other vehicles. The actor critic algorithm, being a continuous deterministic approach, performed better by producing continuous control actions especially in curves where the Q-learning algorithm suffered from discontinuous control actions. Yu *et al.* [120] also designed deep Q-learning algorithms to steer the vehicle. The authors investigated the effect of several reward functions and experimented with different hyper-parameters like gradient update rules and double Q-learning.

3.2.3 Fuzzy logic control

Fuzzy logic systems seek to model human expertise by using linguistic variables, these variables conceptualize the fuzziness in human decision making. Such systems do not require high computational costs unlike conventional control methods, and they do not need any mathematical model which makes them suitable for systems with complex models. Human operational experience is then conceptualized into a fuzzy controller that consists of (**IF/THEN**) rules. These rules simulate the expert's "know how" as an algorithm governed by fuzzy sets and membership functions [121]. An example of a fuzzy rule for steering control would be; **if** "position error" is negative **then** "steering angle" is positive, where the position error is the input and the steering angle is the output. The downside to this control approach is the expertise needed to set the fuzzy rules. Fuzzy logic systems consist of four parts as shown in Fig. 12:

- The fuzzifier which converts the crisp inputs into fuzzy sets (fuzzification).
- The rule base that contains the fuzzy rules in the form of (**IF/THEN**) conditions.
- The inference module which determines the matching percentage between fuzzy inputs and rules.
- The defuzzifier which extracts output values from the fuzzy sets (defuzzification).

Authors of [122] used fuzzy logic control for obstacle avoidance maneuvers, the control task was divided into collision detection and then decision making. Each of the tasks was performed by a fuzzy controller, the obstacle distance and angle were chosen as inputs for the obstacle detection module and the outputs were steering maneuver and safe cruising speed. The inputs to the second module were set as tracking error and distance

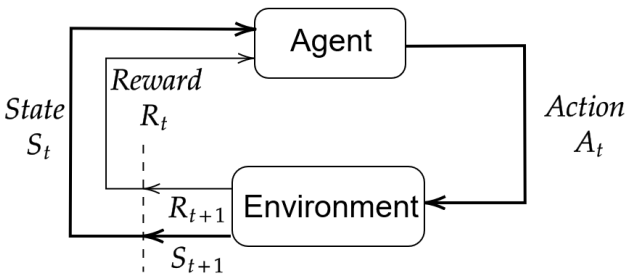


Fig. 11. Markov decision process.

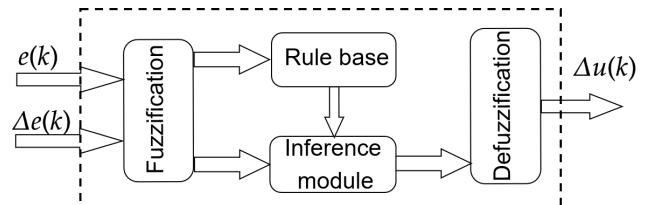


Fig. 12. Structure of fuzzy logic controller

to obstacle and the output defines the magnitude of the control signal. The results showed smooth performance but were limited to straight roads only. Paper [123] dealt with path tracking for vehicles with electromechanical wheel systems, they developed a fuzzy logic controller that handles steering and speed control. The kinematic bicycle model was used and the controller was simplified with only three fuzzy sets. However, the starting point and initial orientation were found to have a significant impact on controller performance. Paper [124] designed a fuzzy inference system to guide an autonomous vehicle under varying loads. The controller receives the tracking error and load as inputs and computes the speed for left and right wheels, the latter was subject to saturation. The controller was implemented on a microprocessor and tested for several scenarios. Shui *et al.* [125], presented a novel data-driven predictive control, the authors based their control approach on type 2 T-S fuzzy neural networks to achieve trajectory tracking of car-like mobile robots. The main advantage of such approach is that it is completely independent of mathematical models and simulation results showed that this approach can deal with the influence of uncertain factors, and obtain higher tracking accuracy. Korkmaz *et al.* [126] designed an autonomous vehicle control system based on fuzzy logic and deployed it into JavaScript racer game. The authors used a simple sliding histogram window approach to detect road lanes and used this information in fuzzy logic reference position and velocity generators. The fuzzy logic generators were based on a 7×7 rule base with triangular membership functions. The speed and position control was ensured by a PD controller given the reference values obtained from the fuzzy logic generators. Paper [127] introduced a fuzzy control algorithm for vehicles moving close to boundaries, the authors used a four motion cycle program according to five collision proximity levels. The latter are obtained from sensor data, Five linguistic variables with 25 fuzzy rules were used to develop the fuzzy controller. Evaluation tests showed that the developed fuzzy controller performed well and was able to overcome sensor imperfections. Van *et al.* [128] proposed a fuzzy logic controller with a convolution neural network for steering control. PilotNet [129] was modified and used for predicting steering angles which in turn were used with vehicle and steering angular velocities as inputs to the fuzzy logic system. The latter recommends optimal steering and velocity controls that are based on fuzzy rules inspired from human experience. Similarly, Chen *et al.* [130] used CMAC neural network [131] with fuzzy logic to control an autonomous vehicle. The neural network ensures the self-learning ability of the controller which eliminate errors, while the fuzzy logic module improves the control quality by suppressing disturbances and increasing robustness. The latter is formed by 56 fuzzy rules and takes the position and speed errors as inputs. The combined CMAC-fuzzy controller was tested in straight and curved roads, where its performance was superior

compared to PID control. Several research papers used fuzzy logic to improve classical controllers as well, especially PID control in [132–134], MPC control [37, 135, 136] and SMC [137–139]. Both model-based and model-free control methods have advantages and drawbacks. In many situations they can complement each other for a better altogether solution, an overall comparative summary is provided in table (1).

4 CONCLUSIONS

This article provided a technical review regarding the latest research on path tracking and lateral control techniques for autonomous driving. Several control formulation concepts were presented and discussed, limits and strengths of each control approach were highlighted and compared. However, this review does not claim to provide an exhaustive and thorough discussion on the subject. Based on the reviewed research, it has been found that vehicle steering systems are rarely considered in the control design, which makes this an attractive area for future developments. Furthermore, faulty sensors and sensing limits need to be considered in control design, which is rarely the case in the literature. The trend seems to shift more and more towards data-based methods and black-box like techniques over the classical control approach. However, the majority of published works tend to focus on control performance optimization and disregard constraints and safety conditions. Nevertheless, stability insurance is still a major barrier to the deployment of such approaches despite their good performance.

Model predictive control has become one of the best control methods for the path tracking task due to its ability to guarantee state and input constraints. However, stability analysis for this type of control is still a challenge. H_∞ , SMC and back-stepping control makes a great choice for dealing with nonlinearities, parameter uncertainties and disturbances at the expense of complicated theoretical derivations and design approaches. LQR, PID and geometry-based controllers are suitable for low speed applications, simplified models with negligible disturbances and uncertainties. Except MPC control, most model-based controllers cannot impose constraints on system inputs, outputs and states. This is a very important quality for guaranteeing safety and comfort in automated guidance, especially for preserving stability conditions and enforcing actuator limits.

On the other hand, model-free controllers are a key solution when modelling becomes a daunting task. These methods become a very interesting approach when sufficient data is available, they can ensure multitasking without being affected by the nonlinear characteristics of the system. AI-based methods can also be used to enhance and improve model-based controllers

Table 1
Comparative summary of model-based and mode-free lateral controllers

Controller	Model type	Strengths	Weaknesses	Related work
MPC	Kinematic Dynamic Linear Nonlinear	Optimal results Handles constraints Predictive feature Handles MIMO systems	Online Optimization High computation burden Requires accurate model	[27, 28, 30, 31, 33] [35–41, 43] [132–134]
SMC	Kinematic Dynamic Linear Nonlinear	Strong robustness Suitable for nonlinear systems Can handle MIMO systems Suitable for switching systems Low computation burden	Chattering phenomenon Non continuous control actions Cannot handle constraints	[56–59, 61–66] [137–139]
LQR	Kinematic Dynamic Linear	Optimal results Handles MIMO systems Off-line optimization Easy to implement	Linearisation assumptions States must be measurable Requires accurate model Poor robustness	[75–82]
H_∞	Kinematic Dynamic Linear Nonlinear	Strong robustness Guaranteed system stability Can handle MIMO systems Suitable for nonlinear systems	Complex design and derivations Disturbances must be bounded Cannot impose constraints Impractical for large dimensions	[68–74]
LPV	Geometric Kinematic Dynamic Linear	Captures Non-linearities Feedback control Suitable for switching systems Low computation burden	Difficult design Requires instantiation Requires accurate model Limited stability	[31, 32, 36] [71, 82]
Back-stepping	Kinematic Dynamic Linear Nonlinear	Recursive feature Disturbance rejection Handles unmodeled parameters Extendable system dimension	States must be measurable Sensitive to parameter variation Complex design	[87–90]
Pure-pursuit Stanley	Geometric	Simple design Low computation burden	Affected by the preview distance Excludes vehicle dynamics Limited to low speeds Cannot handle varying velocity Unsuitable for reverse motion	[18, 91–96]
PID	–	Widely used in the industry Feedback control Simple design Easy implementation Requires only the error signal	Undershoot and overshoot Difficult gain tuning Limited performance Limited to single input systems Unsuitable for complex systems	[83–86] [37, 135, 136]
Deep learning	–	Data-driven Versatile application Model-free Handles MIMO systems Suitable for feed-forward control	Black-box system Stability is not guaranteed Requires training data Several hyper-parameters Long off-line training	[108–112]
Reinforcement learning	–	Continuous online learning No data required for training Can handle stochastic dynamics	Requires control policy Very long computations No stability insurance Discret control actions	[114, 116–120]
Fuzzy logic	–	Intuitive design Human like reasoning High precision Linguistic feature Fast operation Model-free	Stability is not guaranteed Human expertise required Prone to dimensionality curse Cannot handle MIMO systems Several tuning parameters	[122–124] [126–128, 130]

by learning complex models and control algorithms, the main downside of these methods is the black-box nature that cannot ensure stability and other physical properties.

In general, despite the huge advancements seen lately in the field of autonomous driving, several challenges such as steering systems, discontinuous data and

immeasurable parameters still persist and require extensive efforts to be resolved.

5 ACKNOWLEDGEMENTS

The authors would like to thank the University of Paris-Saclay for their financial support provided to conduct this research.

6 AUTHORS' CONTRIBUTIONS

Yassine Kebbati : Conceptualization, Methodology, Writing original draft, Reviewing and Editing; **Naima Ait-Oufroukh** : Review and Validation; **Dalil Ichalal** : Review and Supervision; **Vincent Vigneron** : Supervision, Review and Validation.

7 CONFLICTS of INTEREST

The authors declare having no conflict of interest for the publication of this article.

References

- [1] NHTSA, "Fatal Motor Vehicle Crashes: Overview," tech. rep., U.S. Department of Transportation, 2019.
- [2] R. Behringer, S. Sundareswaran, B. Gregory, R. Elsley, B. Addison, W. Guthmiller, R. Daily, and D. Bevely, "The darpa grand challenge - development of an autonomous vehicle," in *IEEE Intelligent Vehicles Symposium, 2004*, pp. 226–231, 2004.
- [3] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, et al., "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2446–2454, 2020.
- [4] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.
- [5] O. Amidi and C. E. Thorpe, "Integrated Mobile Robot Control," *Mobile Robots V*, vol. 1388, no. May, pp. 504–523, 1991.
- [6] T. Sebastian, S. Pascal, M. Mike, and et al, "Stanley: The Robot that Won the DARPA Grand Challenge," *Journal of Field Robotics*, vol. 33, no. 1, pp. 1–17, 2006.
- [7] A. Chebly, R. Talj, and A. Charara, "Coupled Longitudinal and Lateral Control for an Autonomous Vehicle Dynamics Modeled Using a Robotics Formalism," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 12526–12532, 2017.
- [8] A. Chebly, R. Talj, and A. Charara, "Coupled longitudinal/lateral controllers for autonomous vehicles navigation, with experimental validation," *Control Engineering Practice*, vol. 88, no. June 2018, pp. 79–96, 2019.
- [9] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.
- [10] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," *IEEE Intelligent Vehicles Symposium, Proceedings*, vol. 2015-Augus, no. Iv, pp. 1094–1099, 2015.
- [11] P. Polack, F. Altche, B. DAndrea-Novet, and A. De La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?," *IEEE Intelligent Vehicles Symposium, Proceedings*, no. Iv, pp. 812–818, 2017.
- [12] Y. Lin, J. Mcphee, and N. L. Azad, "Longitudinal Dynamic versus Kinematic Models for Car-Following Control Using Deep Reinforcement Learning," *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019*, pp. 1504–1510, 2019.
- [13] S. Dominguez, A. Ali, G. Garcia, and P. Martinet, "Comparison of lateral controllers for autonomous vehicle : Experimental results," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 1418–1423, 2016.
- [14] G. Indiveri, "Kinematic time-invariant control of a 2D nonholonomic vehicle," *Proceedings of the IEEE Conference on Decision and Control*, vol. 3, no. December, pp. 2112–2117, 1999.
- [15] S. Blažič, "Takagi-Sugeno vs. Lyapunov-based tracking control for a wheeled mobile robot," *WSEAS Transactions on Systems and Control*, vol. 5, no. 8, pp. 667–676, 2010.
- [16] R. Hasan-farag, J. J. Lee, J. J. Lee, C. March, and T. Shim, "Full Vehicle Dynamic Modeling for Chassis Controls," *Fisitacom*, vol. 221, no. 4, pp. 377–391, 2007.
- [17] J. V. Frasch, A. Gray, M. Zanon, H. J. Ferreau, S. Sager, F. Borrelli, and M. Diehl, "An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles," *2013 European Control Conference, ECC 2013*, pp. 4136–4141, 2013.
- [18] N. H. Amer, H. Zamzuri, K. Hudha, and Z. A. Kadir, "Modelling and control strategies in path tracking control for autonomous ground vehicles: a review of state of the art and challenges," *Journal of intelligent & robotic systems*, vol. 86, no. 2, pp. 225–254, 2017.
- [19] N. H. Amer, H. Zamzuri, K. Hudha, and Z. A. Kadir, "Modelling and Control Strategies in Path Tracking Control for Autonomous Ground Vehicles: A Review of State of the Art and Challenges," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 86, no. 2, pp. 225–254, 2017.
- [20] F. Althché, P. Polack, and A. de La Fortelle, "A simple dynamic model for aggressive, near-limits trajectory planning," in *2017 IEEE intelligent vehicles symposium (IV)*, pp. 141–147, IEEE, 2017.
- [21] H. B. Pacejka, "Vehicle System Dynamics : International Journal of Vehicle Mechanics and Mobility," *International Journal of Vehicle Mechanics and Mobility*, no. August 2012, pp. 37–41, 2008.
- [22] E. Alcalá, V. Puig, J. Quevedo, and T. Escobet, "Gain-scheduling LPV control for autonomous vehicles including friction force estimation and compensation mechanism," *IET Control Theory and Applications*, vol. 12, no. 12, pp. 1683–1693, 2018.
- [23] U. Eren, A. Prach, B. B. Koçer, S. V. Rakovic, E. Kayacan, and B. Açikmese, "Model predictive control in aerospace systems: Current state and opportunities," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 7, pp. 1541–1566, 2017.
- [24] T. P. Nascimento, C. E. Dórea, and L. M. G. Gonçalves, "Nonholonomic mobile robots' trajectory tracking model predictive control: A survey," *Robotica*, vol. 36, no. 5, pp. 676–696, 2018.
- [25] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.
- [26] S. J. Qin and T. A. Badgwell, "Process Control Dynamic," *Control Engineering Practice*, vol. 11, pp. 733–764, 2003.

- [27] Q. Yao and Y. Tian, "A model predictive controller with longitudinal speed compensation for autonomous vehicle path tracking," *Applied Sciences (Switzerland)*, vol. 9, no. 22, 2019.
- [28] Y. Kebbati, V. Puig, N. Ait-Oufroukh, V. Vigneron, and D. Ichalal, "Optimized adaptive mpc for lateral control of autonomous vehicles," in *2021 9th International Conference on Control Mechatronics and Automation (ICCMA)*, pp. 1–9, IEEE, 2021.
- [29] Y. Kebbati and L. Baghli, "Design, modeling and control of a hybrid grid-connected photovoltaic-wind system for the region of adrar, algeria," *International Journal of Environmental Science and Technology*, pp. 1–28, 2022.
- [30] Y. Kebbati, N. Ait-Oufroukh, V. Vigneron, and D. Ichalal, "Neural network and anfis based auto-adaptive mpc for path tracking in autonomous vehicles," in *2021 18th International Conference on Networking, Sensing and Control (ICNSC)*, pp. 1–6, IEEE, 2021.
- [31] E. Alcalá, V. Puig, and J. Quevedo, "LPV-MPC Control for Autonomous Vehicles," *IFAC-PapersOnLine*, vol. 52, no. 28, pp. 106–113, 2019.
- [32] Y. Kebbati, N. Ait-Oufroukh, V. Vigneron, and D. Ichalal, "Coordinated pso-pid based longitudinal control and lpv-mpc based lateral control for autonomous vehicles," in *2022 20th European Control Conference (ECC)*, pp. 1–6, IEEE, 2022.
- [33] M. Rick, J. Clemens, L. Sommer, A. Folkers, K. Schill, and C. Büskens, "Autonomous Driving Based on Nonlinear Model Predictive Control and Multi-Sensor Fusion," *IFAC-PapersOnLine*, vol. 52, no. 8, pp. 458–473, 2019.
- [34] M. Knauer and C. Büskens, "From worhp to transworhp," in *Proceedings of the 5th international conference on astrodynamics tools and techniques*, 2012.
- [35] S. Li, Z. Li, B. Zhang, S. Zheng, X. Lu, and Z. Yu, "Path tracking for autonomous vehicles based on nonlinear model: Predictive control method," *SAE Technical Papers*, vol. 2019-April, no. April, pp. 1–7, 2019.
- [36] E. Alcalá, V. Puig, J. Quevedo, and U. Rosolia, "Autonomous racing using Linear Parameter Varying-Model Predictive Control (LPV-MPC)," *Control Engineering Practice*, vol. 95, no. November 2019, p. 104270, 2020.
- [37] H. Wang, B. Liu, X. Ping, and Q. An, "Path Tracking Control for Autonomous Vehicles Based on an Improved MPC," *IEEE Access*, vol. 7, pp. 161064–161073, 2019.
- [38] Y. Xu, B. Y. Chen, X. Shan, W. H. Jia, Z. F. Lu, and G. Xu, "Model predictive control for lane keeping system in autonomous vehicle," *2017 7th International Conference on Power Electronics Systems and Applications - Smart Mobility, Power Transfer and Security, PESA 2017*, vol. 2018-Janua, no. 1, pp. 1–5, 2018.
- [39] C. Sun, X. Zhang, Q. Zhou, and Y. Tian, "A Model Predictive Controller with Switched Tracking Error for Autonomous Vehicle Path Tracking," *IEEE Access*, vol. 7, no. c, pp. 53103–53114, 2019.
- [40] H. Guo, D. Cao, H. Chen, Z. Sun, and Y. Hu, "Model predictive path following control for autonomous cars considering a measurable disturbance: Implementation, testing, and verification," *Mechanical Systems and Signal Processing*, vol. 118, pp. 41–60, 2019.
- [41] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-Based Model Predictive Control for Autonomous Racing," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3363–3370, 2019.
- [42] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious Model Predictive Control Using Gaussian Process Regression," *IEEE Transactions on Control Systems Technology*, pp. 1–12, 2019.
- [43] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale RC cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 2015.
- [44] A. Zanelli, A. Domahidi, J. Jerez, and M. Morari, "FORCES NLP: an efficient implementation of interior-point methods for multistage nonlinear nonconvex programs," *International Journal of Control*, vol. 93, no. 1, pp. 13–29, 2020.
- [45] J. Kabzan, M. I. Valls, V. J. Reijgwart, H. F. Hendrikx, C. Ehmke, M. Prajapat, A. Bühler, N. Gosala, M. Gupta, R. Sivanesan, et al., "Amz driverless: The full autonomous racing system," *Journal of Field Robotics*, vol. 37, no. 7, pp. 1267–1294, 2020.
- [46] A. Tătlea-Codrean, T. Mariani, and S. Engell, "Design and simulation of a machine-learning and model predictive control approach to autonomous race driving for the f1/10 platform," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6031–6036, 2020.
- [47] R. Verschuere, S. De Bruyne, M. Zanon, J. V. Frasch, and M. Diehl, "Towards time-optimal race car driving using nonlinear mpc in real-time," in *53rd IEEE conference on decision and control*, pp. 2505–2510, IEEE, 2014.
- [48] R. Verschuere, M. Zanon, R. Quirynen, and M. Diehl, "Time-optimal race car driving using an online exact hessian based nonlinear mpc algorithm," in *2016 European control conference (ECC)*, pp. 141–147, IEEE, 2016.
- [49] D. Kloeser, T. Schoels, T. Sartor, A. Zanelli, G. Prison, and M. Diehl, "Nmpc for racing using a singularity-free path-parametric model with obstacle avoidance," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 14324–14329, 2020.
- [50] J. Betz, H. Zheng, A. Liniger, U. Rosolia, P. Karle, M. Behl, V. Krovi, and R. Mangharam, "Autonomous vehicles on the edge: A survey on autonomous vehicle racing," *IEEE Open Journal of Intelligent Transportation Systems*, 2022.
- [51] S. J. Gambhire, D. R. Kishore, P. S. Londhe, and S. N. Pawar, "Review of sliding mode based control techniques for control system applications," *International Journal of Dynamics and Control*, vol. 9, no. 1, pp. 363–378, 2021.
- [52] S. F. Ahmed, Y. Raza, H. F. Mahdi, W. M. Wan Muhamad, M. Kamran Joyo, A. Shah, and M. Y. Koonndhar, "Review on Sliding Mode Controller and Its Modified Types for Rehabilitation Robots," *ICETAS 2019 - 2019 6th IEEE International Conference on Engineering, Technologies and Applied Sciences*, 2019.
- [53] M. Mat-Noh, R. Mohd-Mokhtar, M. R. Arshad, Z. M. Zain, and Q. Khan, "Review of sliding mode control application in autonomous underwater vehicles," *Indian Journal of Geo-Marine Sciences*, vol. 48, no. 7, pp. 973–984, 2019.
- [54] X. Li, R. Lu, J. Bai, and A. Xue, "Robust integral sliding model controller design for system subject to packet loss," in *2017 36th Chinese Control Conference (CCC)*, pp. 3804–3808, IEEE, 2017.
- [55] S. Lv, N. Wang, Y. Wang, J. Yin, and M. J. Er, "Nonsingular terminal sliding mode based trajectory tracking control of an autonomous surface vehicle with finite-time convergence," in *International Symposium on Neural Networks*, pp. 83–92, Springer, 2017.

- [56] R. Talj, G. Tagne, and A. Charara, "Immersion and invariance control for lateral dynamics of autonomous vehicles, with experimental validation," *2013 European Control Conference, ECC 2013*, no. Iv, pp. 968–973, 2013.
- [57] G. Tagne, R. Talj, and A. Charara, "Design and Comparison of Robust Nonlinear Controllers for the Lateral Dynamics of Intelligent Vehicles," *Control Engineering Practice*, vol. 40, pp. 81–92, 2015.
- [58] Y. Wu, L. Wang, J. Zhang, and F. Li, "Path Following Control of Autonomous Ground Vehicle Based on Nonsingular Terminal Sliding Mode and Active Disturbance Rejection Control," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 6379–6390, 2019.
- [59] K. Akermi, S. Chouraqui, and B. Boudaa, "Novel SMC control design for path following of autonomous vehicles with uncertainties and mismatched disturbances," *International Journal of Dynamics and Control*, vol. 8, no. 1, pp. 254–268, 2020.
- [60] J. Yang, S. Li, and X. Yu, "Sliding-mode control for systems with mismatched uncertainties via a disturbance observer," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 1, pp. 160–169, 2013.
- [61] Y. Yang and Y. Yan, "Attitude regulation for unmanned quadrotors using adaptive fuzzy gain-scheduling sliding mode control," *Aerospace Science and Technology*, vol. 54, pp. 208–217, 2016.
- [62] F. Gao, X. Hu, S. E. Li, K. Li, and Q. Sun, "Distributed Adaptive Sliding Mode Control of Vehicular Platoon with Uncertain Interaction Topology," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 9, pp. 2–3, 2018.
- [63] A. Norouzi, R. Kazemi, and S. Azadi, "Vehicle lateral control in the presence of uncertainty for lane change maneuver using adaptive sliding mode control with fuzzy boundary layer," *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, vol. 232, no. 1, pp. 12–28, 2018.
- [64] P. Wang, S. Gao, L. Li, S. Cheng, and L. Zhao, "Automatic steering control strategy for unmanned vehicles based on robust backstepping sliding mode control theory," *IEEE Access*, vol. 7, pp. 64984–64992, 2019.
- [65] R. Alika, E. M. Mellouli, and E. H. Tissir, "Optimization of Higher-Order Sliding Mode Control Parameter using Particle Swarm Optimization for Lateral Dynamics of Autonomous Vehicles," *2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology, IRASET 2020*, 2020.
- [66] R. Talj, G. Tagne, and A. Charara, "Immersion and invariance control for lateral dynamics of autonomous vehicles, with experimental validation," *2013 European Control Conference, ECC 2013*, no. Iv, pp. 968–973, 2013.
- [67] X.-H. Chang, *Robust output feedback H-infinity control and filtering for uncertain linear systems*, vol. 7. Springer Science & Business, 2014.
- [68] X. He, Y. Liu, K. Yang, J. Wu, and X. Ji, "Robust coordination control of AFS and ARS for autonomous vehicle path tracking and stability," *Proceedings of 2018 IEEE International Conference on Mechatronics and Automation, ICMA 2018*, no. Lmi, pp. 924–929, 2018.
- [69] H. Sun, C. Zhang, G. An, Q. Chen, and C. Liu, "Fuzzy-model-based H_∞ dynamic output feedback control with feedforward for autonomous vehicle path tracking," *2017 International Conference on Fuzzy Theory and Its Applications, iFUZZY 2017*, vol. 2017-November, pp. 1–6, 2018.
- [70] C. Hu, H. Jing, R. Wang, F. Yan, and M. Chadli, "Robust H_∞ output-feedback control for path following of autonomous ground vehicles," *Mechanical Systems and Signal Processing*, vol. 70–71, pp. 414–427, 2016.
- [71] M. Corno, G. Panzani, F. Roselli, M. Giorelli, D. Azzolini, and S. M. Savaresi, "An LPV Approach to Autonomous Vehicle Path Tracking in the Presence of Steering Actuation Nonlinearities," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 4, pp. 1766–1774, 2020.
- [72] S. Chaib, M. S. Netto, and S. Mammar, " H_∞ , adaptive, PID and fuzzy control: A comparison of controllers for vehicle lane keeping," *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 139–144, 2004.
- [73] K. Yang, X. Tang, Y. Qin, Y. Huang, H. Wang, and H. Pu, "Comparative Study of Trajectory Tracking Control for Automated Vehicles via Model Predictive Control and Robust H-infinity State Feedback Control," *Chinese Journal of Mechanical Engineering*, vol. 34, no. 1, 2021.
- [74] R. Wang, H. Jing, C. Hu, F. Yan, and N. Chen, "Robust h_∞ path following control for autonomous ground vehicles with delay and data dropout," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 2042–2050, 2016.
- [75] W. Y. Choi, S.-H. Lee, and C. C. Chung, "Vehicular Lateral Tracking Control with Winding Road Disturbance Compensation," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 15699–15704, 2020.
- [76] A. Riofrio, S. Sanz, M. J. L. Boada, and B. L. Boada, "A LQR-based controller with estimation of road bank for improving vehicle lateral and rollover stability via active suspension," *Sensors (Switzerland)*, vol. 17, no. 10, 2017.
- [77] K. V. Vivek, M. A. Sheta, and V. Gumtapure, "A comparative study of stanley, lqr and mpc controllers for path tracking application (adas/ad)," *Proceedings - 2019 IEEE International Conference on Intelligent Systems and Green Technology, ICISGT 2019*, pp. 67–71, 2019.
- [78] F. M. Barbosa, L. B. Marcos, M. M. da Silva, M. H. Terra, and V. G. Junior, "Robust path-following control for articulated heavy-duty vehicles," *Control Engineering Practice*, vol. 85, pp. 246–256, 2019.
- [79] M. H. Terra, J. P. Cerri, and J. Y. Ishihara, "Optimal robust linear quadratic regulator for systems subject to uncertainties," *IEEE Transactions on Automatic Control*, vol. 59, no. 9, pp. 2586–2591, 2014.
- [80] J. P. Cerri, M. H. Terra, and J. Y. Ishihara, "Recursive robust regulator for discrete-time state-space systems," *Proceedings of the American Control Conference*, no. 3, pp. 3077–3082, 2009.
- [81] K. il Kim, H. Guan, B. Wang, R. Guo, and F. Liang, "Active steering control strategy for articulated vehicles," *Frontiers of Information Technology and Electronic Engineering*, vol. 17, no. 6, pp. 576–586, 2016.
- [82] E. Alcala, V. Puig, J. Quevedo, T. Escobet, and R. Comasolivas, "Autonomous vehicle control using a kinematic Lyapunov-based technique with LQR-LMI tuning," *Control Engineering Practice*, vol. 73, no. May 2017, pp. 1–12, 2018.
- [83] Y. Kebbati, N. Ait-Oufroukh, V. Vigneron, D. Ichalal, and D. Gruyer, "Optimized self-adaptive pid speed control for autonomous vehicles," in *2021 26th International Conference on Automation and Computing (ICAC)*, pp. 1–6, IEEE, 2021.
- [84] X. Han, X. Zhang, Y. Du, and G. Cheng, "Design of autonomous vehicle controller based on BP-PID," *IOP*

Conference Series: Earth and Environmental Science, vol. 234, p. 012097, mar 2019.

- [85] X. Dong, H. Pei, and M. Gan, "Autonomous vehicle lateral control based on fractional-order pid," in *2021 IEEE 5th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, vol. 5, pp. 830–835, 2021.
- [86] G. Han, W. Fu, W. Wang, and Z. Wu, "The lateral tracking control for the intelligent vehicle based on adaptive pid neural network," *Sensors*, vol. 17, no. 6, 2017.
- [87] L. Khasawneh and M. Das, "Lateral trajectory tracking control using backstepping method for autonomous vehicles," in *2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1013–1016, 2021.
- [88] C. M. Kang, W. Kim, and H. Baek, "Cascade backstepping control with augmented observer for lateral control of vehicle," *IEEE Access*, vol. 9, pp. 45367–45376, 2021.
- [89] Y. W. Jeong, J. S. Kim, C. M. Kang, and C. C. Chung, "Backstepping control design for the second-order electric power steering system using augmented observer," in *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pp. 79–84, 2017.
- [90] L. Nehaoua and L. Nouvelière, "Backstepping based approach for the combined longitudinal-lateral vehicle control," in *2012 IEEE Intelligent Vehicles Symposium*, pp. 395–400, 2012.
- [91] D. S. Lal, A. Vivek, and G. Selvaraj, "Lateral control of an autonomous vehicle based on pure pursuit algorithm," in *2017 International Conference on Technological Advancements in Power and Energy (TAP Energy)*, pp. 1–8, 2017.
- [92] H. Wang, X. Chen, Y. Chen, B. Li, and Z. Miao, "Trajectory tracking and speed control of cleaning vehicle based on improved pure pursuit algorithm," in *2019 Chinese Control Conference (CCC)*, pp. 4348–4353, 2019.
- [93] D. Kapsalis, O. Sename, V. Milanés, and J. J. Martínez Molina, "Design and Experimental Validation of an LPV Pure Pursuit Automatic Steering Controller," in *CTS 2021 - 16th IFAC Symposium on Control in Transportation Systems*, (Lille (virtual), France), June 2021.
- [94] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo, and S. Thrun, "Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing," in *2007 American Control Conference*, pp. 2296–2301, 2007.
- [95] N. H. Amer, K. Hudha, H. Zamzuri, V. R. Aparow, A. F. Z. Abidin, Z. Abd Kadir, and M. Murrad, "Adaptive modified stanley controller with fuzzy supervisory system for trajectory tracking of an autonomous armoured vehicle," *Robotics and Autonomous Systems*, vol. 105, pp. 94–111, 2018.
- [96] A. AbdElmoniem, A. Osama, M. Abdelaziz, and S. A. Maged, "A path-tracking algorithm using predictive stanley lateral controller," *International Journal of Advanced Robotic Systems*, vol. 17, no. 6, p. 1729881420974852, 2020.
- [97] X. Chen, H. Mai, Z. Zhang, and F. Gu, "A novel adaptive pseudospectral method for the optimal control problem of automatic car parking," *Asian Journal of Control*, vol. 24, no. 3, pp. 1363–1377, 2022.
- [98] H. Li, B. Zhao, W. Yu, N. Xu, and K. Guo, "Design and application of a tire slip energy control scheme based on optimal guaranteed cost theory for four-in-wheel drive electric vehicles," *Asian Journal of Control*, vol. 24, no. 3, pp. 1333–1350, 2022.
- [99] C. Lv, H. Yu, N. Zhao, J. Chi, H. Liu, and L. Li, "Robust state-error port-controlled hamiltonian trajectory tracking control for unmanned surface vehicle with disturbance uncertainties," *Asian Journal of Control*, vol. 24, no. 1, pp. 320–332, 2022.
- [100] S. Kuutti, S. Fallah, R. Bowden, and P. Barber, "Deep learning for autonomous vehicle control: Algorithms, state-of-the-art, and future prospects," *Synthesis Lectures on Advances in Automotive Technology*, vol. 3, no. 4, pp. 1–80, 2019.
- [101] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (Y. W. Teh and M. Titterton, eds.), vol. 9 of *Proceedings of Machine Learning Research*, (Chia Laguna Resort, Sardinia, Italy), pp. 249–256, PMLR, 13–15 May 2010.
- [102] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.
- [103] L. Medsker and L. C. Jain, *Recurrent neural networks: design and applications*. CRC press, 1999.
- [104] R. J. Williams, "Learning representations by back-propagating errors david e. rumelhart, geoffrey e. hinton, and," *Cognitive Modeling*, p. 213, 2002.
- [105] A. Menon, K. Mehrotra, C. K. Mohan, and S. Ranka, "Characterization of a class of sigmoid functions with applications to neural networks," *Neural Networks*, vol. 9, no. 5, pp. 819–835, 1996.
- [106] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10*, (Madison, WI, USA), p. 807–814, Omnipress, 2010.
- [107] J. F. Kolen and S. C. Kremer, "Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies," *A Field Guide to Dynamical Recurrent Networks*, 2010.
- [108] S. Sharma, G. Tewolde, and J. Kwon, "Lateral and longitudinal motion control of autonomous vehicles using deep learning," *IEEE International Conference on Electro Information Technology*, vol. 2019-May, pp. 460–464, 2019.
- [109] J. Jhung, I. Bae, J. Moon, T. Kim, J. Kim, and S. Kim, "End-to-End Steering Controller with CNN-based Closed-loop Feedback for Autonomous Vehicles," *IEEE Intelligent Vehicles Symposium, Proceedings*, vol. 2018-June, no. Iv, pp. 617–622, 2018.
- [110] G. Devineau, P. Polack, F. Altche, and F. Moutarde, "Coupled Longitudinal and Lateral Control of a Vehicle using Deep Learning," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 2018-Novem, pp. 642–649, 2018.
- [111] V. Rausch, A. Hansen, E. Solowjow, C. Liu, E. Kreuzer, and J. K. Hedrick, "Learning a deep neural net policy for end-to-end control of autonomous vehicles," in *2017 American Control Conference (ACC)*, pp. 4914–4919, 2017.
- [112] H. M. Eraqi, M. N. Moustafa, and J. Honer, "End-to-End Deep Learning for Steering Autonomous Vehicles Considering Temporal Dependencies," *arXiv preprint arXiv:1710.03804*, no. Nips, pp. 1–8, 2017.
- [113] C. J. Watkins and P. DAYAN, *Q-learning*, vol. 8. Boston: Kluwer Academic Publishers, 1992.

- [114] D. Li, D. Zhao, Q. Zhang, and Y. Chen, "Reinforcement Learning and Deep Learning Based Lateral Control for Autonomous Driving [Application Notes]," *IEEE Computational Intelligence Magazine*, vol. 14, no. 2, pp. 83–98, 2019.
- [115] W. Bernhard, E. Eric, G. Christophe, D. Christos, C. Rémi, and S. Andrew, "TORCS, The Open Racing Car Simulator." [Online]. Available: <http://www.torcs.org>, 2014.
- [116] L. Manuelli and P. Florence, "Reinforcement learning for autonomous driving obstacle avoidance using lidar," *Massachusetts Institute of Technology, Tech. Rep.*, 2015.
- [117] P. Wang, C. Y. Chan, and A. De La Fortelle, "A Reinforcement Learning Based Approach for Automated Lane Change Maneuvers," *IEEE Intelligent Vehicles Symposium, Proceedings*, vol. 2018-June, no. Iv, pp. 1379–1384, 2018.
- [118] P. Wang, H. Li, and C.-Y. Chan, "Quadratic q-network for learning continuous control for autonomous vehicles," *arXiv preprint arXiv:1912.00074*, 2019.
- [119] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, "End-to-end deep reinforcement learning for lane keeping assist," *arXiv preprint arXiv:1612.04340*, 2016.
- [120] A. Yu, R. Palefsky-Smith, and R. Bedi, "Deep reinforcement learning for simulated autonomous vehicle control," *Course Project Reports: Winter*, pp. 1–7, 2016.
- [121] I. S. Emmanuel, "Fuzzy logic-based control for autonomous vehicle: A survey," *International Journal of Education and Management Engineering*, vol. 7, pp. 41–49, 2017.
- [122] H. Elsayed, B. A. Abdullah, and G. Aly, "Fuzzy logic based collision avoidance system for autonomous navigation vehicle," in *2018 13th International Conference on Computer Engineering and Systems (ICCES)*, pp. 469–474, 2018.
- [123] S. Allou, Y. Zennir, and A. Belmeguenai, "Fuzzy logic controller for autonomous vehicle path tracking," in *2017 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, pp. 328–333, 2017.
- [124] H. Q. T. Ngo and A. S. Tran, "Using fuzzy logic scheme for automated guided vehicle to track following path under various load," in *2018 4th International Conference on Green Technology and Sustainable Development (GTSD)*, pp. 312–316, 2018.
- [125] Y. Shui, T. Zhao, S. Dian, Y. Hu, R. Guo, and S. Li, "Data-driven generalized predictive control for car-like mobile robots using interval type-2 t-s fuzzy neural network," *Asian Journal of Control*, vol. 24, no. 3, pp. 1391–1405, 2022.
- [126] U. B. Etlik, B. Korkmaz, A. Beke, and T. Kumbasar, "A fuzzy logic-based autonomous car control system for the javascript racer game," *Transactions of the Institute of Measurement and Control*, vol. 43, no. 5, pp. 1028–1038, 2021.
- [127] V. Stoian, "A control algorithm for autonomous electric vehicles by fuzzy logic," *Advanced Engineering Forum*, vol. 27, pp. 103–110, 04 2018.
- [128] N. Dinh Van and G.-W. Kim, "Fuzzy logic and deep steering control based recommendation system for self-driving car," in *2018 18th International Conference on Control, Automation and Systems (ICCAS)*, pp. 1107–1110, 2018.
- [129] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al., "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.
- [130] J. Chen, W. Han, P. Zhao, Z. Wei, and X. Liu, "The study on path tracking control method based on fuzzy-cmac for autonomous vehicle in rural environment," in *2020 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, pp. 1–4, 2020.
- [131] J. S. Albus, "A new approach to manipulator control: The cerebellar model articulation controller," *Transaction of the ASME Journal of Dynamic Systems*, vol. 97, no. 3, pp. 270–277, 1975.
- [132] Y. Han, Q. Zhu, and Y. Xiao, "Data-driven control of autonomous vehicle using recurrent fuzzy neural network combined with pid method," in *2018 37th Chinese Control Conference (CCC)*, pp. 5239–5244, 2018.
- [133] D. Tiep, K. Lee, D.-Y. Im, B. Kwak, and Y.-J. Ryoo, "Design of fuzzy-pid controller for path tracking of mobile robot with differential drive," *INTERNATIONAL JOURNAL of FUZZY LOGIC and INTELLIGENT SYSTEMS*, vol. 18, pp. 220–228, 09 2018.
- [134] X. Chen, Q. Bao, and B. Zhang, "Research on 4wis electric vehicle path tracking control based on adaptive fuzzy pid algorithm," in *2019 Chinese Control Conference (CCC)*, pp. 6753–6760, 2019.
- [135] A. S. M. Bakibillah, M. Kamal, C. P. E. Tan, T. Hayakawa, and J.-i. Imura, "Fuzzy-tuned model predictive control for dynamic eco-driving on hilly roads," *Applied Soft Computing*, vol. 99, 11 2020.
- [136] M. Khooban, N. Vafamand, T. Dragicevic, F. Blaabjerg, and T. Niknam, "Model predictive control based on t-s fuzzy model for electrical vehicles delayed model," *IET Electric Power Applications*, vol. 11, 10 2016.
- [137] M. Shirzadeh, M. H. Shojaeefard, A. Amirkhani, and H. Behroozi, "Adaptive fuzzy nonlinear sliding-mode controller for a car-like robot," in *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*, pp. 686–691, 2019.
- [138] C.-L. Hwang, C.-C. Yang, and J. Y. Hung, "Path tracking of an autonomous ground vehicle with different payloads by hierarchical improved fuzzy dynamic sliding-mode control," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 2, pp. 899–914, 2018.
- [139] A. Norouzi, R. Kazemi, and S. Azadi, "Vehicle lateral control in the presence of uncertainty for lane change maneuver using adaptive sliding mode control with fuzzy boundary layer," *Proceedings of the Institution of Mechanical Engineers Part I Journal of Systems and Control Engineering*, vol. 232, pp. 12–28, 10 2017.