Автономная некоммерческая организация высшего образования
«Университет Иннополис»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)
по направлению подготовки
09.03.01 - «Информатика и вычислительная техника»

GRADUATION THESIS
(BACHELOR'S GRADUATION THESIS)
Field of Study
09.03.01 – «Computer Science»

Направленность (профиль) образовательной программы
«Информатика и вычислительная техника»
Area of Specialization / Academic Program Title:
«Computer Science»

Тема / Topic	**Управление боковым ускорением / Lateral Acceleration Control**

Работу выполнил / Thesis is executed by	**Зиннатуллин Нурислам Ильясович / Zinnatullin Nurislam Ilyasovich**	подпись / signature
Руководитель выпускной квалификационной работы / Supervisor of Graduation Thesis	**Лукманов Рустам Абубакирович / Lukmanov Rustam Abubakirovich**	подпись / signature

Иннополис, Innopolis, 2025

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Autonomous vehicles (AVs) are a transformative innovation in modern transportation, promising significant improvements in road safety, efficiency, and accessibility. A central challenge in the development of AVs is the design of reliable and robust control systems that allow vehicles to perceive the environment, plan trajectories, and execute safe maneuvers.

1.1.1 Layered vs end-to-end approaches

The control approaches for autonomous driving can be broadly divided into two paradigms: classical layered and end-to-end learning approaches [1]. Layered approach rely on a modular pipeline, where perception (what is around car), localization and mapping (where is car), planning (where to go) and control are handled separately. Each optimized with well-understood algorithms. In contrast, end-to-end is more radical, a "biology inspired" approach. Instead of pipeline, large deep learning model takes all raw data at once (such as camera images, LiDAR scans, GPS and other car descriptive data [2]) and outputs control commands for vehicle [3].

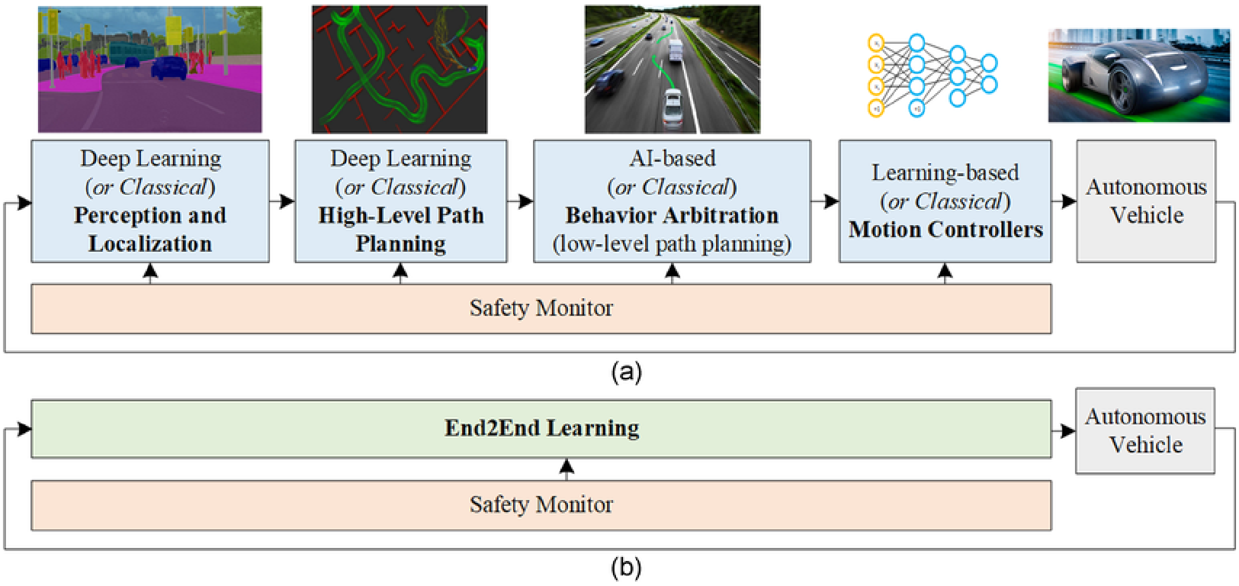

Fig. 1.1. Comparison of layered and end-to-end autonomous driving architectures. The figure illustrates two primary paradigms: (a) The classical layered approach, which separates tasks like perception, planning, and action into a sequential pipeline. (b) The end-to-end approach, where a single deep learning model directly maps raw sensor inputs to vehicle control commands. Adapted from [1].

Each paradigm has advantages and disadvantages. *End-to-end* method promise adaptability, reduced hand-engineering, and potentially better performance, but often suffer from interpretability issues, lack of robustness, and difficulties in handling rare or unseen driving scenarios. The *layered* method, on the other hand, offers interpretability, safety, modularity, and strong integration with control theory, but may require extensive system identification and careful tuning to deal with nonlinear vehicle dynamics.

Nowadays, self-driving regulators and safety accessors rely heavily on the ability to understand, explain, and verify system's behaviour. This is expected, since passengers' lives depend on the system. End-to-end approaches remain largely in the research domain because their lack of interpretability makes it extremely difficult to certify as unambiguously safe for full autonomy. On the other hand layered approach is used more frequently by self-driving industry and has better interptability and safety. With this in mind, this thesis will focus on the layered paradigm.

1.1.2 Layered architecture. Lateral acceleration control

Within the classical *layered* method, perception modules extract meaningful features (lane markings, objects, and free space), planning modules generate feasible trajectories, and control modules execute these trajectories through longitudinal and lateral control [4], [5]. This thesis specifically investigates the lateral control module, which determines steering actions to accurately follow planned trajectories, ensuring stability, safety, and passenger comfort.

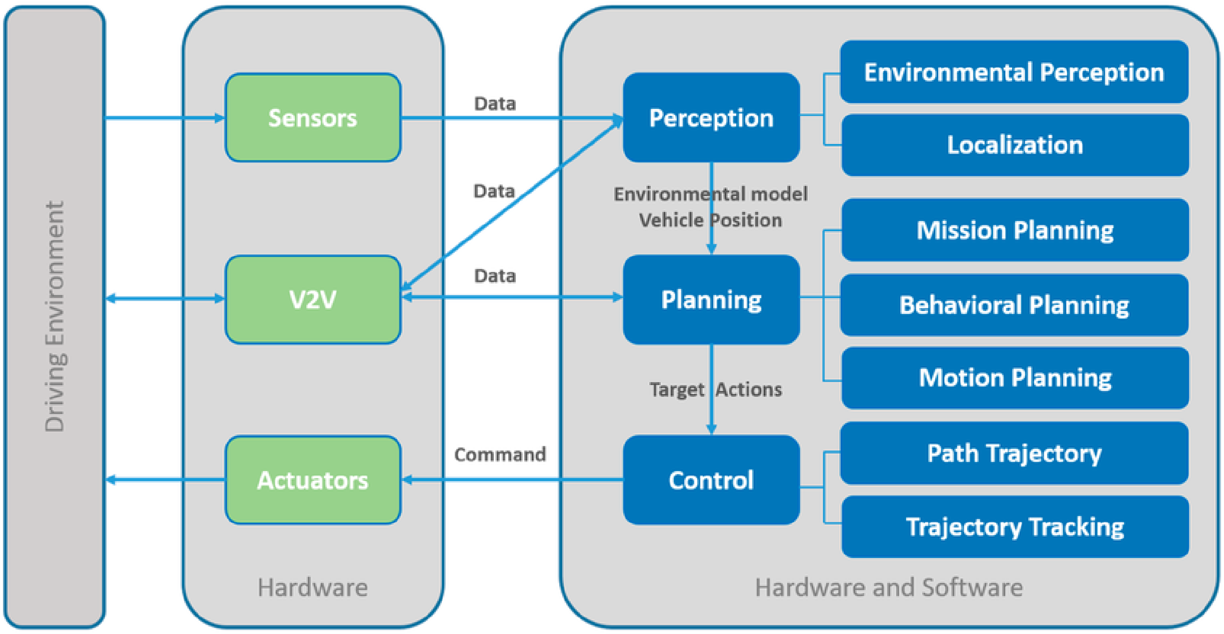

Fig. 1.2. Typical classical self-driving system. Adapted from [2]

Existing literature presents a fragmented landscape of solutions for the lateral control module: model-based controllers like Model Predictive Control (MPC) offer high performance but are computationally expensive and sensitive to model inaccuracies; model-free controllers like PID are simple but lack optimality; and learning-based methods like Reinforcement Learning (RL) show great adaptive potential but face challenges in safety verification and sample efficiency [6], [7].

1.2 Purpose of the study

Despite the critical role of lateral control, existing solutions present a challenging trade-off between tracking accuracy, computational efficiency, and robustness. Furthermore, the literature lacks rigorous, side-by-side quantitative benchmarks of controllers from different paradigms (model-based, model-free, learning-based) within a unified testing environment. This makes it difficult for engineers to definitively assess the trade-offs when selecting a controller for a specific application.

This thesis aims to address this gap by systematically implementing and evaluating a representative suite of lateral controllers to provide a clear, data-driven analysis of their performance trade-offs.

In addition, this work should test the following hypotheses:

1. **Hybrid Superiority Hypothesis:** A hybrid architecture that combines model-based and learning-based methods can achieve superior robustness to unmodeled disturbances (e.g., sudden friction changes) compared to either approach in isolation.

2. **RL Performance Hypothesis:** A Reinforcement Learning (RL) agent, trained with a carefully shaped reward function that penalizes control effort

and jerk, can match the smoothness and comfort of a well-tuned classical controller while demonstrating superior adaptability in high-performance driving scenarios.

3. **Compact recursive model Hypothesis:** While large Transformer models are effective for processing multimodal data in end-to-end architectures, their computational cost can be prohibitive for a single control module. This hypothesis posits that a smaller, Tiny Recursive Model architecture [8] can offer a more efficient solution. We hypothesize that such a model can effectively learn vehicle dynamics for lateral control, achieving a superior trade-off between predictive accuracy and computational load, making it suitable for real-time embedded systems.

By investigating these hypotheses within a unified simulation framework (comma.ai simulator), this work seeks to provide actionable insights for the design and selection of next-generation lateral control systems.

1.3 Design approach

To test this hypothesis, a representative controller from each major category will be implemented and evaluated within the open-source comma.ai simulation environment. This unified framework allows for a controlled, comparative analysis of tracking accuracy, control smoothness, and computational load under identical conditions. Furthermore, the robustness of each controller will be tested by introducing unmodeled disturbances, such as simulated icy patches and sensor noise, to quantify their performance degradation.

1.4 Results

1.5 Significance

The significance of this thesis is twofold. First, it provides a much-needed quantitative benchmark for lateral control strategies, moving beyond qualitative discussions to offer concrete, data-driven insights into the performance trade-offs between classical, learning-based, and hybrid methods. This comparative analysis serves as a valuable guide for engineers and researchers in selecting and designing control systems that are best suited for their specific operational and computational constraints. Second, by proposing and evaluating novel hybrid and learning-based architectures, this work contributes to the development of more robust and intelligent controllers. The findings have direct implications for improving the safety and reliability of autonomous vehicles, particularly their ability to handle unforeseen and challenging driving conditions. By establishing a rigorous evaluation methodology within an open-source framework, this study also provides a reproducible baseline for future advancements in autonomous vehicle control.

1.6 Paper structure

This thesis is structured as follows: Chapter 2 provides a comprehensive literature review of model-based, model-free, and learning-based lateral control methods. Chapter 3 details the experimental methodology, including the simulation environment, the implementation of each controller, and the evaluation metrics. Chapter 4 presents and analyzes the comparative results. Finally, Chapter 5 discusses the implications of the findings, addresses the limitations of the study, and proposes directions for future research.

Chapter 2

Literature Review

In a layered autonomous driving stack, **lateral control** — often termed path-tracking or trajectory-tracking — is the module responsible for aligning the vehicle with a planned trajectory using steering commands. The controller's primary function is to minimize tracking errors while maintaining vehicle stability, safety, and passenger comfort.

Numerous control strategies have been developed to address the complex vehicle dynamics involved, each presenting inherent trade-offs in accuracy, robustness, and computational cost. Following the classification by Liu et al. [6] and Liu et al. [9], these approaches can be grouped into four primary categories:

- **Model-based methods**, which use explicit vehicle dynamics models to achieve high performance, at the cost of complexity and sensitivity to model accuracy.

- **Model-free methods**, which rely on feedback structures or adaptive rules for simpler implementation, but may lack formal guarantees of optimality.

- **Learning-based methods**, which leverage machine learning to learn complex behaviors but require significant data and present challenges in safety verification.

- **Hybrid approaches**, which combine techniques from the other categories to leverage their respective strengths and mitigate their weaknesses.

It is important to note that the boundaries between these categories are not always rigid. Certain controllers, most notably PID, can be considered ambiguous. The taxonomy used in this thesis prioritizes a controller's real-time operational principle over its design or tuning methodology. Therefore, a controller is classified based on whether it explicitly uses a predictive model of vehicle dynamics during its execution loop.

2.1 Model-based Control: The Physics Paradigm

Model-based methods utilize explicit physics models of the vehicle to predict its response to steering inputs. This approach allows for theoretical performance and stability guarantees, but creating accurate models is challenging due to complex, nonlinear vehicle dynamics that vary with speed, road conditions, and load [10]. The field is largely divided into two fundamental modeling paradigms:

- **Vehicle Kinematic-Based (VKB) models**, which describe the geometric motion of a vehicle without considering the forces that cause it.

- **Vehicle Dynamic-Based (VDB) models**, which incorporate the physical forces (e.g., tire friction, inertia) that govern vehicle movement.

Although more complex four-wheel models exist, they are generally avoided in real-time control applications due to their high computational demands.

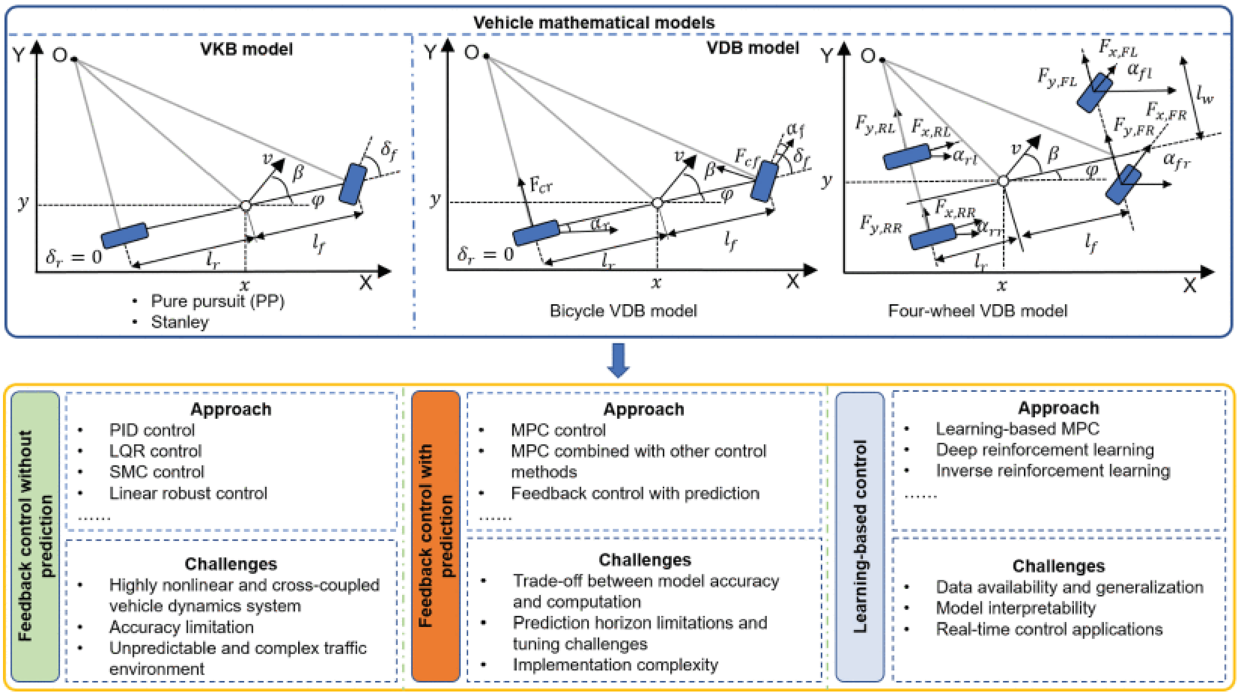

Fig. 2.1. A summary of a vehicle trajectory-tracking control system. Adapted from [6].

2.1.1 Kinematic and Dynamic Foundations of Lateral Control

The pursuit of precise lateral control begins with mathematically describing a vehicle's motion, a problem that branches into two distinct approaches with escalating complexity and realism.

Kinematic-Based Methods: Geometry in Motion

The simplest approach is the Vehicle Kinematic-Based (VKB) model, which treats the vehicle as a rigid body moving according to pure geometry, ignoring physical forces such as tire slip. This simplification leads to intuitive and computationally efficient controllers suitable for low-speed applications.

Pure Pursuit (PP), as described by Kapsalis et al. [11], exemplifies this geometric approach. It calculates the steering angle δ_f required to follow a circular arc toward a look-ahead point on the path, located at a distance L_d:

$$\delta_f = \arctan\left(\frac{2L\sin(\alpha)}{L_d}\right) \tag{2.1}$$

where L is the vehicle's wheelbase, and α is the angle between the vehicle's heading and the look-ahead point. A key limitation of Pure Pursuit is its focus on a future point, which causes it to neglect the vehicle's current heading error relative to the path.

The **Stanley** controller, introduced by Thrun et al. [12] and analyzed in works like AbdElmoniem et al. [13], addresses this limitation by explicitly incorporating both the lateral error e (cross-track error) and the heading error ψ_e. Its control law is defined as:

$$\delta_f = \psi_e + \arctan\left(\frac{ke}{v_x + k_s}\right) \tag{2.2}$$

where ψ_e is the heading error, e is the lateral error at the front axle, v_x is the longitudinal speed, k is a tunable gain, and k_s is a small softening constant to ensure numerical stability at low speeds. By combining both error terms, the Stanley controller generally achieves higher accuracy than Pure Pursuit but introduces the challenge of tuning its parameters for optimal performance. The performance is highly dependent on the gain k, which ideally should vary with vehicle speed, a key motivation for the development of more adaptive controllers.

Dynamic-Based Methods: Introducing Force and Physics

While kinematic models are sufficient for low-speed maneuvers, their core assumption of no tire slip breaks down at higher speeds or during aggressive cornering. This limitation necessitates a transition to Vehicle Dynamic-Based

(VDB) models, which incorporate the physics of tire forces and vehicle inertia. A common representation is the single-track (or "bicycle") model:

$$\ddot{y} = -\dot{\phi}\dot{x} + \frac{2}{m}(F_{yf}\cos\delta_f + F_{yr}) \tag{2.3}$$

$$\ddot{\phi} = \frac{2}{I_z}(l_f F_{yf} - l_r F_{yr}) \tag{2.4}$$

where y and ϕ are the lateral position and yaw angle, $\dot{x}$ is the longitudinal velocity, m is the vehicle mass, I_z is the yaw moment of inertia, l_f and l_r are the distances from the center of gravity to the front and rear axles, and F_{yf} and F_{yr} are the lateral tire forces. The control input is the front steering angle, δ_f. This more realistic model enables a new class of high-performance controllers but requires more sophisticated strategies to manage its complexity.

2.1.2 Advanced Model-Based Control Paradigms

The availability of a dynamic model opens the door to powerful optimal and robust control theories that can provide formal guarantees on performance and stability.

Optimal Control: The LQR Framework

For a linearized vehicle model, the Linear Quadratic Regulator (LQR) [14], [15] offers an elegant solution for optimal control. It computes the optimal steering command by minimizing a quadratic cost function over an infinite horizon:

$$J = \int_0^\infty (\mathbf{x}^T Q\mathbf{x} + \mathbf{u}^T R\mathbf{u})\, dt \tag{2.5}$$

where $\mathbf{x}$ is the system state vector (e.g., lateral error, heading error), $\mathbf{u}$ is the control input (steering angle), and $Q \succeq 0$ and $R \succ 0$ are symmetric weighting matrices. These matrices are tuned to balance tracking performance (penalizing state error $\mathbf{x}$) against control effort (penalizing input $\mathbf{u}$). The performance of LQR is intrinsically tied to the accuracy of the linearized model, and it can degrade when strong nonlinearities become dominant.

Robust Control: Handling Uncertainty with SMC and H_∞

When the model is imperfect or the vehicle encounters unexpected disturbances, robust control strategies become necessary.

Sliding Mode Control (SMC) [16], [17] is a nonlinear technique designed to be highly robust to model uncertainties and external disturbances. It forces the system's state trajectory onto a predefined "sliding surface" (where tracking error is zero) and maintains it there. A typical SMC law has a discontinuous term:

$$u = u_{eq} - K \cdot \text{sign}(s) \tag{2.6}$$

where u_{eq} is the equivalent control to keep the system on the surface, s is the sliding surface variable, and K is a gain. The main drawback of SMC is "chattering" — high-frequency control oscillations caused by the sign function. This is not only mechanically undesirable but can also excite unmodeled high-frequency dynamics. This has spurred research into mitigation strategies, such as replacing the sign function with a continuous saturation function inside a thin boundary layer around the sliding surface, or developing higher-order SMC.

H_∞ **control**, detailed in works such as Chang [18], offers a different framework for robustness, designing a controller that minimizes the worst-case gain from external disturbances to the tracking error. While it is a powerful theoretical tool for guaranteeing stability in the presence of a specified level of model uncertainty, its application often results in high-order, conservative controllers that are complex to design and implement in real-time systems, limiting its practical adoption compared to MPC or SMC.

The Pinnacle of Anticipation: Model Predictive Control (MPC)

Model Predictive Control (MPC) [19] represents a natural evolution of model-based control, directly addressing the limitations of reactive methods like LQR. At each time step, MPC solves a finite-horizon optimal control problem to compute a sequence of future control actions that minimizes a cost function, subject to system dynamics and constraints.

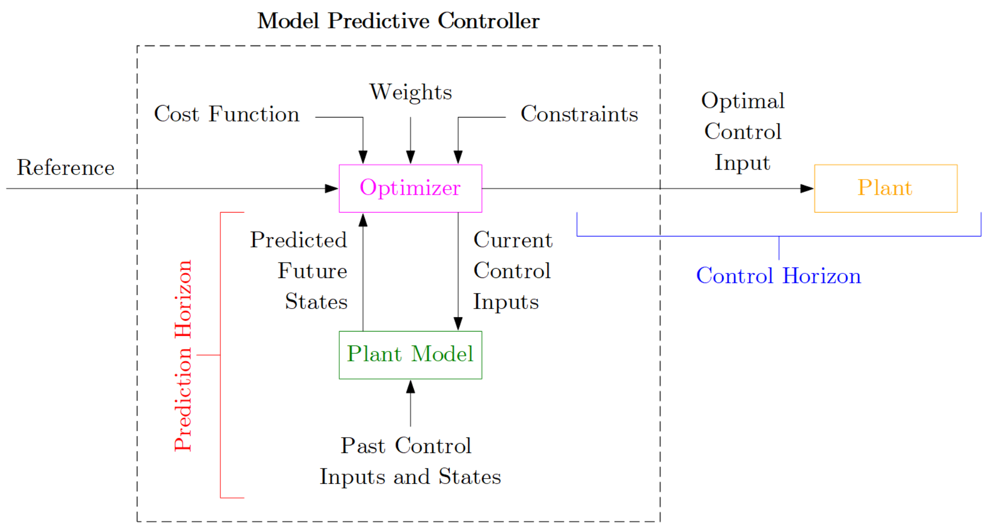

Fig. 2.2. A conceptual architecture of Model Predictive Control (MPC). Adapted from [7].

The core of MPC involves minimizing a cost function over a prediction horizon:

$$\min_{\mathbf{U}} J = \sum_{k=0}^{N_p-1} (\mathbf{x}_k^T Q \mathbf{x}_k + \mathbf{u}_k^T R \mathbf{u}_k) + \mathbf{x}_{N_p}^T P \mathbf{x}_{N_p} \tag{2.7}$$

subject to:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) \tag{2.8}$$

$$\mathbf{u}_{\min} \leq \mathbf{u}_k \leq \mathbf{u}_{\max} \tag{2.9}$$

$$\mathbf{x}_{\min} \leq \mathbf{x}_k \leq \mathbf{x}_{\max} \tag{2.10}$$

where $\mathbf{x}_k$ and $\mathbf{u}_k$ are the state and control input at step k, N_p is the prediction horizon, and Q, R, and P are weighting matrices. Only the first control action in the optimized sequence $\mathbf{U}$ is applied, and the process is repeated at the next time step.

This predictive capability allows MPC to anticipate future reference changes and systematically handle constraints (e.g., on steering angle or lateral acceleration). This is its greatest strength, but it comes at a high computational cost. This has led to a wide range of MPC variations, including nonlinear MPC (NMPC) for higher accuracy and explicit or real-time MPC algorithms designed to reduce the computational burden.

2.2 Model-free Control: Simplicity, Adaptation, and Intelligence

When developing an accurate vehicle model is impractical or the dynamics are highly time-varying, model-free methods offer a compelling alternative. These controllers bypass explicit physics modeling, instead relying on direct feedback, adaptive rules, and real-time data. This approach trades the theoretical guarantees of model-based methods for simplicity and direct applicability.

2.2.1 The Classic Workhorse: PID Control

The Proportional-Integral-Derivative (PID) controller is the cornerstone of model-free control [20], [21]. Its structure is intuitive: the Proportional term corrects current path deviation, the Integral term eliminates persistent offset (e.g., due to constant crosswind), and the Derivative term dampens oscillations by anticipating future error.

While some tuning methods like Ziegler-Nichols [22] use a simplified plant model, the controller's real-time operation is purely reactive to the error signal and does not use a predictive model of vehicle dynamics. It is therefore classified as model-free in this thesis. The primary challenge of PID control is its fixed gains, which often fail to provide consistent performance across varying speeds and road conditions, necessitating enhancements like gain-scheduling.

A discrete-time PID controller with a filtered derivative term can be expressed in the z-domain as:

$$U(z) = \left(K_p + K_i \frac{T_s}{1 - z^{-1}} + K_d \frac{N}{1 + NT_s \frac{1}{1-z^{-1}}} \right) E(z) \tag{2.11}$$

where K_p, K_i, and K_d are the controller gains, N is the derivative filter coefficient, T_s is the sampling time, and $E(z)$ is the z-transform of the tracking error.

2.2.2 Bridging the Gap with an "Ultra-Local" Model: Model-Free Control (MFC)

Model-Free Control (MFC), as proposed by Fliess and Join Fliess and Join [23], offers a more sophisticated data-driven approach. Instead of a detailed physical model, MFC approximates the complex system dynamics with an "ultra-local" model, typically a simple integrator, lumping all nonlinearities and disturbances into a single time-varying term F:

$$y^{(n)} = F + \alpha u \tag{2.12}$$

where $y^{(n)}$ is the n-th time derivative of the output y, u is the control input, α is a user-chosen scaling parameter, and F encapsulates all unknown dynamics and disturbances.

The key idea of MFC is to estimate F in real-time from recent input-output data, effectively learning and canceling the system's complex behavior on the fly. The control law then becomes an adaptive PID structure that compensates for the estimated dynamics:

$$u = \frac{1}{\alpha}\left(y_{ref}^{(n)} - \hat{F} + K_p e + K_d \dot{e}\right) \tag{2.13}$$

where $y_{ref}^{(n)}$ is the reference derivative, $\hat{F}$ is the estimate of F, and e is the tracking error. This creates a controller that is model-free in design yet adaptive in operation, though formal stability analysis can be challenging.

2.2.3 Adapting to a Key Dynamic: Speed-Adaptive MFC

The performance of the general MFC framework can be enhanced by incorporating domain knowledge. Recognizing that a vehicle's lateral dynamics are profoundly influenced by its longitudinal speed, the Speed-Adaptive MFC (SAMFC) proposed by Moreno-Gonzalez et al. [24] makes the model parameter

α a function of vehicle velocity v:

$$\alpha(v) = \max\left(\alpha_{min}, \alpha_0 + k(v - v_0)\right) \tag{2.14}$$

where α_{min} is a minimum gain, v_0 is a threshold speed, and k is an adaptation slope. This allows the controller's responsiveness to adapt seamlessly from low-speed urban driving to high-speed highway travel. The resulting control law represents a sophisticated evolution of PID, offering adaptability without a complex first-principles model.

2.3 Learning-based Control: Adaptive Intelligence

Machine learning has demonstrated state-of-the-art performance in many domains, suggesting its potential to outperform classical controllers by learning complex, nonlinear vehicle dynamics directly from data. In the context of path-tracking, which is a regression problem of mapping vehicle state to a correct steering angle, supervised and reinforcement learning are the dominant paradigms.

Unsupervised learning is less common for direct control but can play an indirect role, for example, in learning a compact "world model" of the environment from unlabeled observational data, which can then be used for training a controller in simulation.

2.3.1 Imitation Learning (IL)

Imitation Learning (IL) or Behavioral Cloning is a form of supervised learning, trains a control policy to mimic expert (human) driving behavior [25]. The model learns a mapping from sensory inputs to steering commands based on a large dataset of human-driven trajectories.

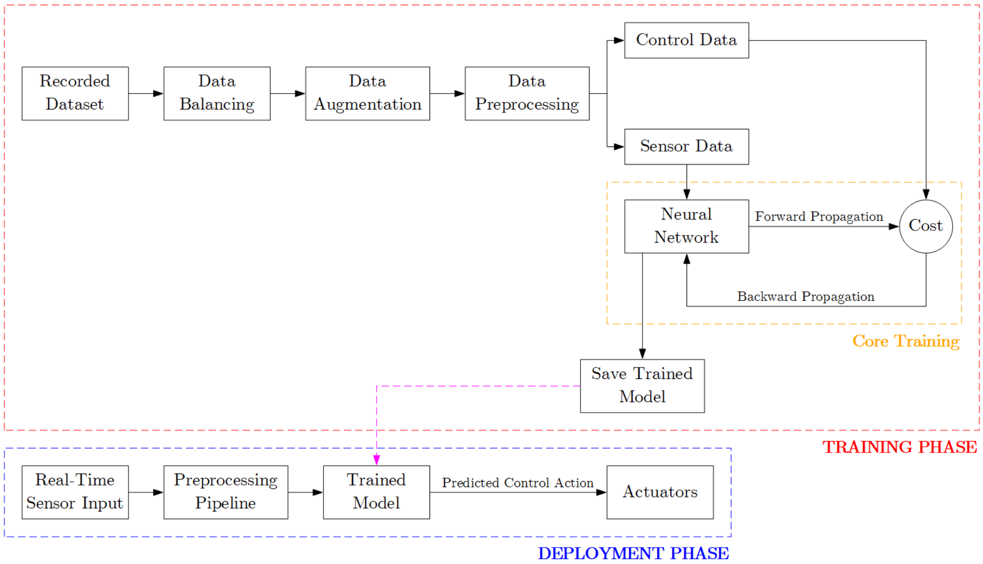

Fig. 2.3. A conceptual architecture for Imitation Learning (IL). Adapted from [7].

However, this approach has several drawbacks. The performance of the learned policy is fundamentally bounded by the skill of the human expert. Furthermore, the policy may fail catastrophically when it encounters states not seen in the training data (the "covariate shift" problem). This makes IL's reliability in rare or unseen events a major concern for safety-critical applications. Consequently, the quality of the dataset is paramount.

2.3.2 Reinforcement Learning (RL)

To overcome the limitations of IL, Reinforcement Learning (RL), as formalized by Sutton, Barto, et al. [26] and applied in works like Li et al. [27], offers a framework for an agent to learn an optimal policy through direct interaction with an environment. Instead of mimicking an expert, the agent learns by trial and error to maximize a cumulative reward signal. This requires the careful design of a reward function that incentivizes desired behaviors like accurate path tracking, passenger comfort, and safety.

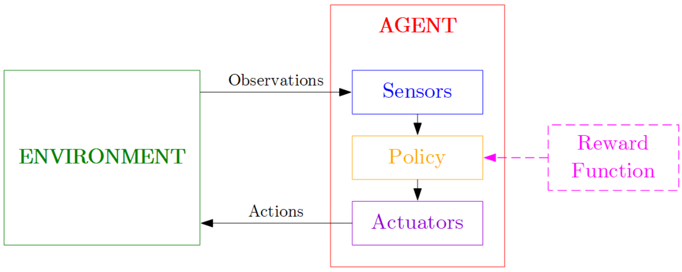

Fig. 2.4. A conceptual architecture for Reinforcement Learning (RL). Adapted from [7].

A key advantage of RL is its potential to discover novel strategies that exceed expert performance. However, RL is notoriously sample-inefficient, often requiring millions of interactions to learn a competent policy. This makes training on real vehicles infeasible, necessitating the use of high-fidelity simulators (e.g., CARLA) or learned world models. A common strategy to improve sample efficiency is to pre-train a policy using IL to acquire basic driving skills, then fine-tune it with RL to optimize performance and handle edge cases.

A recent survey by Chen et al. [28] analyzed the application of various deep RL algorithms to autonomous driving subproblems, as shown in Figure 2.5. For lateral control, Actor-Critic methods and their derivatives are particularly prominent.

The evolution of RL algorithms for control tasks can be understood as a progression of solutions addressing fundamental challenges in learning.

The foundational concept is the **Policy Gradient (PG)** method, explored by Kővári et al. [29], which directly optimizes a parameterized policy by adjusting its parameters in the direction of higher expected reward. While conceptually simple, basic PG methods often suffer from high variance in their gradient estimates,

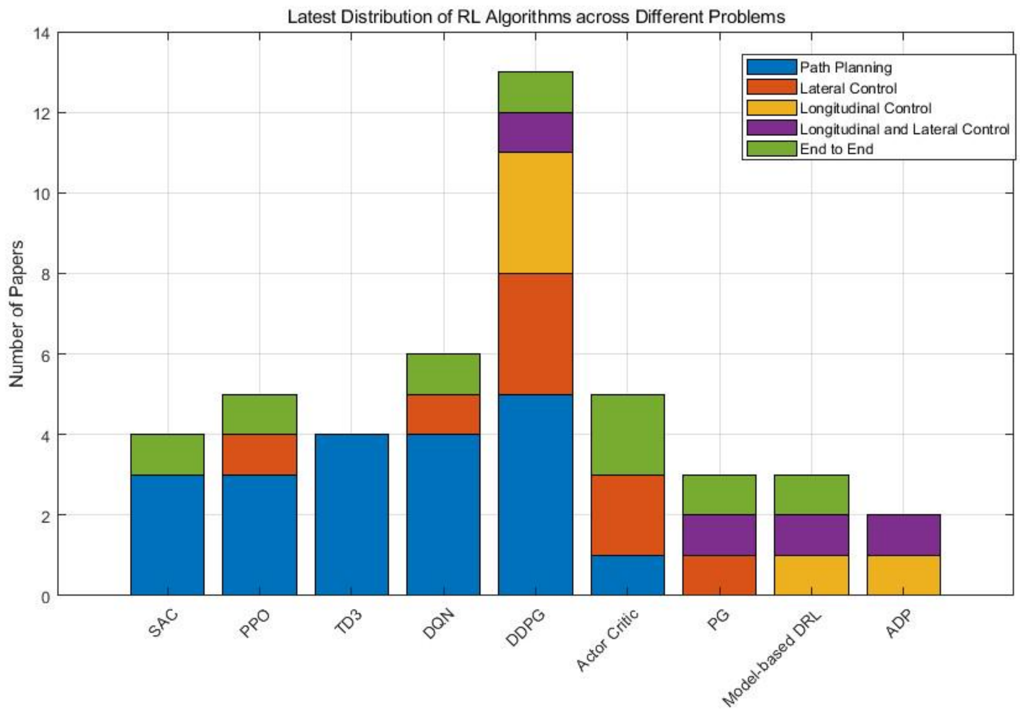

Fig. 2.5. Application of specific RL algorithms to subproblems of trajectory planning, lateral control, longitudinal control, and end-to-end control. Adapted from [28].

which can make training slow and unstable.

To address this high variance, **Actor-Critic (AC)** methods were developed [30]. This architecture introduces a second neural network, the "Critic," whose sole purpose is to estimate the value of the "Actor's" actions. This value-based feedback provides a lower-variance learning signal, allowing the Actor to update its policy more stably and efficiently than with PG alone.

Concurrently, a different family of value-based methods emerged, epitomized by the **Deep Q-Network (DQN)** from Mnih et al. [31]. Instead of learning a policy directly, DQN learns to approximate the optimal action-value function, $Q^*(s,a)$, which predicts the total future reward for taking any action from a given state. The policy is then simply to choose the action with the highest predicted Q-value. However, DQN possesses a critical flaw for continuous control applications: it can only operate in discrete action spaces. Applying it to a task like steering requires crudely discretizing the output (e.g., into angles like {-5, 0, 5} degrees). This forced discretization is a major drawback, as it can lead to jerky control and suffers from the "curse of dimensionality" if fine-grained precision is needed.

This fundamental limitation of DQN was the primary motivation for developing algorithms suited for continuous domains. The **Deep Deterministic Policy Gradient (DDPG)**, developed by Lillicrap et al. [32] and applied in works like He et al. [33], directly addresses this by combining the AC structure with insights from DQN. Its actor network outputs a precise, continuous action, making it a natural fit for vehicle control. While powerful, DDPG is notoriously sensitive to hyperparameter tuning and can be unstable during training.

Seeking to improve upon this instability, **Proximal Policy Optimization (PPO)** was introduced by Schulman et al. [34]. As a state-of-the-art AC algorithm, PPO enhances training stability by using a "clipped" surrogate objective function. This mechanism prevents the policy from changing too drastically in a single update by constraining the size of the policy update, effectively creating a trust region for each gradient step. This approach directly mitigates the risk of destructive, large updates that can plague DDPG, leading to more reliable and robust learning. This has made PPO a popular and effective choice for a wide range of robotics and control problems, including autonomous driving.

2.4 State-of-the-Art and Hybrid Approaches

Beyond the foundational paradigms, recent research has pushed the boundaries of lateral control by creating sophisticated hybrid systems and leveraging novel deep learning architectures. These state-of-the-art methods aim to synthesize the strengths of different approaches, creating controllers that are more adaptive, robust, and intelligent.

2.4.1 Hybrid Model-Based/Model-Free and Learning-Based Control

Recognizing the complementary nature of model-based, model-free, and learning-based methods, hybrid architectures have emerged as a highly promising research direction. These systems combine the predictive power and constraint-handling of model-based controllers or the simplicity of model-free controllers with the adaptability of machine learning to overcome the limitations of each approach in isolation.

A common strategy is to use a learned model to augment a traditional controller. For model-based approaches like MPC, this can involve learning the residual between a simplified dynamics model and the true system and incorporating that estimate into the prediction step [35]. A more general approach, applicable to both model-based and model-free controllers, is to use a neural network to adapt key parameters online. For example, a neural network can tune MPC cost weights (Q, R) based on context (road curvature, speed) [36], or it can perform sophisticated gain scheduling for an adaptive PID controller, adjusting its gains based on complex vehicle states [21].

These architectures preserve the core structure and relative interpretability of the classical controller while improving tracking and robustness to unmodeled dynamics. Although, their drawbacks include increased implementation and runtime complexity, dependence on the fidelity of the learned components, and potentially weakened formal guarantees when learned elements operate in a closed loop.

2.4.2 Advanced Reinforcement Learning Frameworks

While the core RL algorithms provide the foundation, state-of-the-art applications focus on overcoming their practical limitations, namely sample efficiency and the "sim-to-real" gap. To accelerate learning, many modern frameworks pre-train policies using Imitation Learning on expert datasets before fine-tuning with RL [37]. This provides the agent with a competent baseline, significantly reducing the exploration time needed to discover a viable policy.

Furthermore, to enforce safety some constraints can be applied during learning (e.g., via Lagrangian methods or safety “shields”) [38].

Both previous methods address sample-efficiency and safety concerns but can be overly conservative, sensitive to dataset coverage, and still struggle with distributional shift when deployed on a real vehicle.

2.4.3 Learning-Based Architectures for Control

The success of large-scale models like the Transformer [39] (’Transfuser’ variation [40]) in end-to-end autonomous driving, where they excel at fusing complex sensor data (camera, LiDAR, and radar), raises the question of their applicability to the control layer itself. The capacity of these architectures to model intricate, long-range dependencies suggests they could potentially learn highly effective, nonlinear control policies directly from vehicle state data.

However, a significant barrier to this direct application is their computational expense. Standard Transformer models are resource-intensive, and their inference latency is often too high for the demands of a real-time, high-frequency control loop. This makes them impractical for a dedicated low-level controller where rapid and deterministic execution is paramount.

This trade-off motivates a key research direction of this thesis: the investigation of more computationally efficient alternatives that retain a high capacity for learning. A promising approach lies in adapting emerging lightweight architectures, such as Hierarchical Reasoning and Tiny Recursive Models [8], [41], which have demonstrated strong performance on complex reasoning tasks (ARC-AGI) without the massive parameter counts of larger models.

Therefore, this thesis will test the hypothesis that such compact yet powerful architectures can be adapted for the lateral control problem. The goal is to determine if they can offer the adaptive benefits of a sophisticated learning model while adhering to the strict computational constraints of a real-time embedded system, potentially representing a new frontier for intelligent vehicle control.

2.4.4 Online and Continual Learning

To ensure robustness in an ever-changing world, controllers must be able to adapt to novel conditions not seen during initial training. Online and continual learning frameworks address this by enabling the control system to learn from new data streams in real-time [42]. Using techniques like mini-batch updates with efficient optimizers, a vehicle's neural network-based components (e.g., a perception model or a learned dynamics model) can be continuously refined during operation. This reduces the system's bias toward its initial training data and allows it to adapt to gradual changes (e.g., tire wear) or sudden environmental shifts (e.g., changing weather), maintaining consistent performance over the vehicle's lifetime.

2.5 Summary and Research Gap

This review has surveyed the primary paradigms for lateral vehicle control: model-based methods like MPC that promise optimality at a high computational cost; model-free methods like PID that offer simplicity but lack performance guarantees; and learning-based methods like RL that show immense adaptive potential but face challenges in safety verification. State-of-the-art research indicates a trend toward hybrid systems that seek to combine these strengths.

Controller	T1				T2				T3				Mean (T1, T2, T3)			
setup	IAE	MLE	M_ϵ	M_ζ	IAE	MLE	M_ϵ	M_ζ	IAE	MLE	M_ϵ	M_ζ	IAE	MLE	M_ϵ	M_ζ
LQR-1	0.239	0.893	0.010	0.322	0.146	0.603	0.076	0.286	0.128	0.516	0.231	0.152	0.171	0.671	0.106	0.253
LQR-2	0.239	0.909	0.009	0.290	0.146	0.594	0.070	0.278	0.126	0.548	0.127	**0.000**	0.170	0.684	0.069	0.189
LQR-3	0.249	0.909	0.005	0.169	0.178	0.611	0.063	0.293	0.224	0.750	0.093	0.061	0.217	0.757	0.054	0.174
LQR[a]	0.242	0.904	**0.008**	0.260	0.157	0.603	**0.070**	0.286	0.159	0.605	0.150	**0.071**	0.186	0.704	**0.076**	0.206
MFC-1	0.184	0.660	0.070	**0.000**	0.074	0.261	0.157	0.296	0.138	0.315	0.151	0.108	0.132	0.412	0.126	0.135
MFC-2	0.229	0.779	0.016	0.102	0.118	0.329	0.101	0.374	0.220	0.456	0.121	0.089	0.189	0.521	0.079	0.188
MFC-3	0.294	1.057	0.026	0.135	0.223	0.513	0.090	0.297	0.380	0.697	0.145	0.571	0.299	0.756	0.087	0.334
MFC[a]	0.236	0.832	0.037	0.079	0.138	0.368	0.116	0.322	0.246	**0.489**	**0.139**	0.256	0.207	0.563	0.097	0.219
SAMFC-1	0.093	0.372	0.077	0.355	0.085	0.284	0.100	0.232	0.175	0.560	0.356	0.214	0.118	0.405	0.178	0.267
SAMFC-2	0.077	**0.251**	0.108	0.569	0.066	0.301	0.088	0.380	0.118	0.844	0.276	0.407	0.087	0.465	0.157	0.452
SAMFC-3	0.112	0.375	0.048	0.287	0.144	0.520	0.061	0.522	0.315	1.209	0.334	0.365	0.190	0.701	0.148	0.391
SAMFC[a]	**0.094**	**0.333**	0.078	0.404	**0.098**	0.368	0.083	0.378	0.203	0.871	0.322	0.329	0.132	0.524	0.161	0.370
PID-1	0.235	0.801	0.003	0.084	0.083	0.314	0.112	0.451	0.132	0.373	0.250	0.239	0.150	0.496	0.122	0.258
PID-2	0.227	0.773	0.038	0.434	0.153	0.361	0.138	0.849	0.262	0.480	0.144	0.745	0.214	0.538	0.107	0.676
PID-3	0.433	1.211	**0.000**	**0.000**	0.245	0.657	**0.012**	0.013	0.445	1.209	0.083	**0.000**	0.374	1.026	**0.032**	0.004
PID[a]	0.298	0.928	0.014	0.173	0.160	0.444	0.087	0.438	0.280	0.687	0.159	0.328	0.246	0.687	0.087	0.313
NLMPC-1	**0.063**	0.300	0.082	0.134	**0.050**	**0.214**	0.165	**0.000**	**0.051**	**0.274**	0.419	0.569	**0.055**	**0.263**	0.222	0.234
NLMPC-2	0.106	0.459	0.075	**0.000**	0.075	0.301	0.135	**0.000**	0.056	0.374	**0.075**	**0.000**	0.079	0.378	0.095	**0.000**
NLMPC-3	0.127	0.542	**0.000**	**0.000**	0.157	0.501	0.052	**0.000**	0.242	0.959	0.145	**0.000**	0.175	0.667	0.066	**0.000**
NLMPC[a]	0.099	0.434	0.052	**0.045**	0.094	**0.339**	0.117	**0.000**	**0.116**	0.536	0.213	0.190	**0.103**	**0.436**	0.128	**0.078**

[a] Mean values of the 3 setups for each controller.

Fig. 2.6. Fig. 2.6. A comparative evaluation of lateral controllers across three distinct test tracks (T1, T2, T3). Performance is quantified using four metrics: tracking accuracy, measured by Integral Absolute Error (IAE) and Maximum Lateral Error (MLE); stability margin (M_ϵ); and passenger discomfort (M_ζ). For all metrics, lower values indicate superior performance. Adapted from [43].

.

However, a significant gap persists in the literature. Existing survey works tend to be either quantitative but narrow in scope, comparing a few classical controllers on specific tracks (e.g., [43], as shown in Figure 2.6), or they are broad but qualitative, summarizing trade-offs without side-by-side numerical data (e.g., [7], as shown in Figure 2.7). A unified, quantitative comparison that spans all

Comparison Metric	Bang-Bang Control	PID Control	Geometric Control	Model Predictive Control	Imitation Learning Based Control	Reinforcement Learning Based Control
Principle of Operation	Error Driven (FSM)	Error Driven (Correcitve)	Model Based (Corrective)	Model Based (Optimal)	Supervised Learning	Reinforcement Learning
Tracking	Poor	Good	Very Good	Excellent	Good	-
Robustness	Poor	Very Good	Very Good	Excellent	Good	Good
Stability	Poor	Very Good	Very Good	Excellent	Good	Good
Reliability	Very Low	Very High	Very High	Extremely High	Low	Low
Technical Complexity	Very Low	Low	Low	Very High	Low	High
Computational Overhead	Very Low	Low	Low	Very High	High	High
Constraint Satisfaction	Poor	Good	Good	Excellent	-	-
Technical Maturity	Very High	Very High	High	Moderate	Low	Very Low

Fig. 2.7. A qualitative summary of control strategies for autonomous vehicles. Adapted from [7].

major paradigms under identical conditions is missing.

This lack of unified, quantitative comparison makes critical engineering decisions difficult. For instance, what is the precise trade-off in tracking error and computational load between a PID and an MPC controller? How does an RL agent's robustness to sudden disturbances, like an icy patch, compare quantitatively to a classical SMC controller?

This thesis this research will provide the empirical data needed to test the hypotheses outlined in Chapter 1 and create a data-driven analysis of the performance landscape for lateral control.

Bibliography cited

[1] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, “A survey of deep learning techniques for autonomous driving,” *Journal of field robotics*, vol. 37, no. 3, pp. 362–386, 2020. DOI: 10.1002/rob.21918

[2] F. Rosique, P. J. Navarro, C. Fernández, and A. Padilla, “A systematic review of perception system and simulators for autonomous vehicles research,” *Sensors*, vol. 19, no. 3, p. 648, 2019, ISSN: 1424-8220. DOI: 10.3390/s19030648 [Online]. Available: https://www.mdpi.com/1424-8220/19/3/648

[3] L. Chen, P. Wu, K. Chitta, B. Jaeger, A. Geiger, and H. Li, “End-to-end autonomous driving: Challenges and frontiers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 10 164–10 183, 2024. DOI: 10.1109/TPAMI.2024.3435937

[4] J. Zhao et al., “Autonomous driving system: A comprehensive survey,” *Expert Systems with Applications*, vol. 242, p. 122 836, 2024, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2023.122836 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423033389

[5] M. Reda, A. Onsy, A. Y. Haikal, and A. Ghanbari, "Path planning algorithms in the autonomous driving system: A comprehensive review," *Robotics and Autonomous Systems*, vol. 174, p. 104 630, 2024.

[6] W. Liu et al., "A systematic survey of control techniques and applications in connected and automated vehicles," *IEEE Internet of Things Journal*, vol. 10, no. 24, pp. 21 892–21 916, 2023. DOI: 10.1109/JIOT.2023.3307002

[7] C. V. Samak, T. V. Samak, and S. Kandhasamy, "Control strategies for autonomous vehicles," in *Autonomous driving and advanced driver-assistance systems (ADAS)*, CRC Press, 2021, pp. 37–86. DOI: 10.48550/arXiv.2011.08729

[8] A. Jolicoeur-Martineau, "Less is more: Recursive reasoning with tiny networks," *arXiv preprint arXiv:2510.04871*, 2025. DOI: https://doi.org/10.48550/arXiv.2510.04871

[9] Q. Liu, X. Li, S. Yuan, and Z. Li, "Decision-making technology for autonomous vehicles: Learning-based methods, applications and future outlook," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2021, pp. 30–37. DOI: 10.1109/ITSC48978.2021.9564580

[10] A. Boyali, S. Mita, and V. John, "A tutorial on autonomous vehicle steering controller design, simulation and implementation," *arXiv preprint arXiv:1803.03758*, 2018. DOI: 10.48550/arXiv.1803.03758

[11] D. Kapsalis, O. Sename, V. Milanes, and J. J. M. Molina, "Design and experimental validation of an lpv pure pursuit automatic steering controller,"

IFAC-PapersOnLine, vol. 54, no. 2, pp. 63–68, 2021. DOI: 10.1016/j.ifacol.2021.06.010

[12] S. Thrun et al., "Stanley: The robot that won the darpa grand challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006. DOI: 10.1002/rob.20147

[13] 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. 1 729 881 420 974 852, 2020. DOI: 10.1177/1729881420974852

[14] M. T. Peterson, T. Goel, and J. C. Gerdes, "Exploiting linear structure for precision control of highly nonlinear vehicle dynamics," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1852–1862, 2022. DOI: 10.1109/TIV.2022.3171734

[15] Z. Wang, K. Sun, S. Ma, L. Sun, W. Gao, and Z. Dong, "Improved linear quadratic regulator lateral path tracking approach based on a real-time updated algorithm with fuzzy control and cosine similarity for autonomous vehicles," *Electronics*, vol. 11, no. 22, p. 3703, 2022. DOI: 10.3390/electronics11223703

[16] S. Gambhire, D. R. Kishore, P. Londhe, and S. 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. DOI: 10.1007/s40435-020-00638-7

[17] 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. DOI: 10.1007/s40435-018-0478-z

[18] X.-H. Chang, *Robust output feedback H-infinity control and filtering for uncertain linear systems*. Springer Science & Business, 2014, vol. 7.

[19] S. Yu, M. Hirche, Y. Huang, H. Chen, and F. Allgöwer, "Model predictive control for autonomous ground vehicles: A review," *Autonomous Intelligent Systems*, vol. 1, no. 1, p. 4, Aug. 2021, ISSN: 2730-616X. DOI: 10.1007/s43684-021-00005-z [Online]. Available: https://doi.org/10.1007/s43684-021-00005-z

[20] 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)*, IEEE, vol. 5, 2021, pp. 830–835. DOI: 10.1109/ITNEC52019.2021.9586818

[21] 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, p. 1244, 2017. DOI: 10.3390/s17061244

[22] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Transactions of the American society of mechanical engineers*, vol. 64, no. 8, pp. 759–765, 1942. DOI: 10.1115/1.4019264

[23] M. Fliess and C. Join, "Model-free control," *International journal of control*, vol. 86, no. 12, pp. 2228–2252, 2013. DOI: 10.1080/00207179.2013.810345

[24] M. Moreno-Gonzalez, A. Artuñedo, J. Villagra, C. Join, and M. Fliess, "Speed-adaptive model-free lateral control for automated cars," *IFAC-*

PapersOnLine, vol. 55, no. 34, pp. 84–89, 2022. DOI: 10.1016/j.ifacol.2022.11.312

[25] S. Sharma, G. Tewolde, and J. Kwon, "Lateral and longitudinal motion control of autonomous vehicles using deep learning," in *2019 IEEE International Conference on Electro Information Technology (EIT)*, IEEE, 2019, pp. 1–5. DOI: 10.1109/EIT.2019.8833873

[26] R. S. Sutton, A. G. Barto, et al., *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1.

[27] 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. DOI: 10.1109/MCI.2019.2901089

[28] Y. Chen, C. Ji, Y. Cai, T. Yan, and B. Su, "Deep reinforcement learning in autonomous car path planning and control: A survey," *arXiv preprint arXiv:2404.00340*, 2024. DOI: 10.48550/arXiv.2404.00340

[29] B. Kővári, F. Hegedüs, and T. Bécsi, "Design of a reinforcement learning-based lane keeping planning agent for automated vehicles," *Applied Sciences*, vol. 10, no. 20, p. 7171, 2020. DOI: 10.3390/app10207171

[30] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," *Advances in neural information processing systems*, vol. 12, 1999.

[31] V. Mnih et al., "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015. DOI: 10.1038/nature14236

[32] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015. DOI: 10.48550/arXiv.1509.02971

[33] R. He, H. Lv, S. Zhang, D. Zhang, and H. Zhang, “Lane following method based on improved ddpg algorithm,” *Sensors*, vol. 21, no. 14, p. 4827, 2021. DOI: 10.3390/s21144827

[34] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017. DOI: 10.48550/arXiv.1707.06347

[35] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, “Learning-based model predictive control: Toward safe learning in control,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 269–296, 2020. DOI: 10.1146/annurev-control-090419-075625

[36] M. Mohamed-Seghir, A. Krama, S. S. Refaat, M. Trabelsi, and H. Abu-Rub, “Artificial intelligence-based weighting factor autotuning for model predictive control of grid-tied packed u-cell inverter,” *Energies*, vol. 13, no. 12, p. 3107, 2020. DOI: 10.3390/en13123107

[37] H. Lu, “Integrating imitation learning with human driving data into reinforcement learning to improve training efficiency for autonomous driving,” *arXiv preprint arXiv:2111.11673*, 2021. DOI: 10.48550/arXiv.2111.11673

[38] J. Achiam, D. Held, A. Tamar, and P. Abbeel, “Constrained policy optimization,” in *International conference on machine learning*, PMLR, 2017, pp. 22–31.

[39] A. Vaswani et al., “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[40] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger, “Transfuser: Imitation with transformer-based sensor fusion for autonomous driving,”

IEEE transactions on pattern analysis and machine intelligence, vol. 45, no. 11, pp. 12 878–12 895, 2022. DOI: 10.1109/TPAMI.2022.3200245

[41] G. Wang et al., "Hierarchical reasoning model," *arXiv preprint arXiv:2506.21734*, 2025. DOI: 10.48550/arXiv.2506.21734

[42] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural networks*, vol. 113, pp. 54–71, 2019. DOI: 10.1016/j.neunet.2019.01.012

[43] A. Artuñedo, M. Moreno-Gonzalez, and J. Villagra, "Lateral control for autonomous vehicles: A comparative evaluation," *Annual reviews in control*, vol. 57, p. 100 910, 2024. DOI: 10.1016/j.arcontrol.2023.100910