Stability Control and Energy Management of an All-Wheel Drive Electric Vehicle Using Artificial Intelligence Methods

Reza Jafari

Supervisor: Dr Pouria Sarhadi

Co-supervisors: Dr Shady Khalil

Dr Amin Paykani

School of Physics, Engineering and Computer Science

University of Hertfordshire

This dissertation is submitted for the degree of

Doctor of Philosophy

March 2026

To my beloved father, whose memory continues to inspire me every day,
and to my mother, for her endless love, strength, and support.

Declaration

I hereby declare that, except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted, either in whole or in part, for consideration for any other degree or qualification at this or any other university. This dissertation is entirely my own work and contains nothing that is the outcome of work undertaken in collaboration with others, except as stated in the text and Acknowledgements. This dissertation contains fewer than 53,000 words, including appendices, bibliography, footnotes, tables, and equations, and includes fewer than 150 figures.

Reza Jafari

March 2026

Acknowledgements

I would like to express my deepest gratitude to my principal supervisor, **Dr Pouria Sarhadi**, for his continuous guidance, encouragement, and invaluable support throughout this research. I am deeply thankful for his patience, motivation, and the time he devoted to helping me overcome challenges during this PhD. This work would not have been possible without his mentorship and patience.

My sincere thanks also go to **Dr Shady Khalil** and **Dr Amin Paykani** for their constructive feedback, insightful discussions, and assistance during my doctoral studies.

I am especially grateful to **Dr Pedram Asef**, whose advice and support have had a profound impact not only on this PhD journey but also on my career. I owe him a great deal for his constant encouragement and inspiration.

I would also like to thank my internal and external examiners, **Dr Herfatmanesh** and **Prof. Fallah**, for their careful evaluation of this thesis and for their constructive comments during the viva examination, which have helped to further improve the quality of this work.

I would also like to thank **Dr Mahmoud Chizari** for his guidance and help during various stages of my research. I am also grateful to **Dr Peter Thomas**, **Dr Vahid Hosseini**, and **Dr Robert Rayner** for their support and guidance during my doctoral studies.

I would like to thank **IPG Automotive** for providing the CarMaker software license used for vehicle dynamics simulations and verification in this research.

Finally, I would like to express my heartfelt appreciation to my beloved family, my friends, and everyone who supported me during this journey. Their love, patience, and belief in me gave me the strength to complete this work.

Abstract

This thesis presents the development of reinforcement learning (RL)-based control frameworks for torque vectoring and energy optimization in all-wheel-drive (AWD) electric vehicles (EVs). The proposed model-free controllers aim to enhance vehicle dynamic stability and energy efficiency simultaneously under nonlinear and uncertain driving conditions. A detailed vehicle model with seven degrees of freedom (7 DOF) and a Pacejka tyre formulation is employed to capture realistic vehicle behaviour. Three deterministic actor–critic algorithms, i.e., deep deterministic policy gradient (DDPG), twin delayed deep deterministic policy gradient (TD3), and curriculum learning-enhanced TD3 (CL TD3), are designed and compared against conventional model-based controllers. The RL agents are trained using a multi-objective reward function that incorporates yaw stability, sideslip angle regulation, tire slip limitation, and energy efficiency based on real electric machine maps. The simulation and verification processes are carried out using the IPG CarMaker environment, which provides a high-fidelity virtual test platform for realistic vehicle dynamics assessment. Simulation results demonstrate that the proposed TD3 and CL TD3 controllers achieve the best trade-off between dynamic stability and energy efficiency, with faster convergence and improved robustness across varying driving conditions. The findings confirm that RL-based control strategies provide an effective and computationally efficient alternative to traditional model-based approaches for real-time torque vectoring and energy optimization in EVs without the requirement for an explicit model of the vehicle.

Table of contents

List of figures

List of tables

Nomenclature

AFS	Active Front Steering.
AI	Artificial Intelligence.
ASOSM	Adaptive Backstepping Second-Order Sliding Mode Control.
AWD	All-Wheel Drive.
CEM	Cross-Entropy Method.
CIM	Command Interpreter Module.
CL	Curriculum Learning.
COG	Centre of Gravity.
DDPG	Deep Deterministic Policy Gradient.
DDPI	Deep Deterministic Policy Iteration.
DDQN	Deep Q-Learning Network.
DMEPPO	Deep Maximum Entropy Proximal Policy Optimization.
DOF	Degrees of Freedom.
DQN	Deep Q-Network.
DYC	Direct Yaw Control.
EV	Electric Vehicle.
FC	Fully Connected (neural network layer).
FTO	Fibonacci tree optimisation.
GAIL	Generative Adversarial Imitation Learning.
HIL	Hardware-in-the-Loop.
ICE	Internal Combustion Engine.
KA	Knowledge-Assisted.
LMI	Linear Matrix Inequalities
LPV	Linear Parameter Varying.
LQR	Linear Quadratic Regulator.

LSQP	Linear Quadratic Regulator with Sequential Quadratic Programming.
MDP	Markov Decision Process.
MEEW	Mechanical Elastic Electric Wheels.
MIMO	Multiple-Input Multiple-Output.
MRAC	Model Reference Adaptive Control.
MPC	Model Predictive Control.
NN	Neural Network.
OU	Ornstein–Uhlenbeck (noise process).
PMSM	Permanent Magnet Synchronous Machine.
REDQ	Randomised Ensemble Double Q-Learning.
RL	Reinforcement Learning.
RWS	Rear-Wheel Steering.
SAC	Soft Actor-Critic.
SARSA	State–Action–Reward–State–Action.
SMC	Sliding Mode Controller.
SQP	Sequential Quadratic Programming.
SSQP	Sliding Mode Controller with Sequential Quadratic Programming.
TD3	Twin Delayed Deep Deterministic Policy Gradient.

a_t	Action vector at time step t.
a_y	Lateral acceleration.
β	Vehicle sideslip angle.
β_e	Sideslip angle error.
β_{des}	Desired vehicle sideslip angle.
β_{max}	Maximum sideslip angle.
$F_{x,t}$	Total longitudinal force.
F_x	Longitudinal tyre force.
F_y	Lateral tyre force.
F_z	Vertical tyre force.
$F_{x,ij}$	Longitudinal force at wheel ij.
$F_{y,ij}$	Lateral force at wheel ij.
$F_{z,ij}$	Vertical force at wheel ij.
γ	Discount factor in reinforcement learning.
η	Electric machine efficiency.
J	Objective or cost function.
L	Wheelbase length.
m	Vehicle mass.
M_z	Corrective yaw moment.
μ	Tyre-road friction coefficient.
ω_{ij}	Wheel angular velocity at wheel ij.
$\pi(s\|\theta^\mu)$	Deterministic policy function parameterised by θ^μ.
P_{tot}	Total power consumption.
$Q(s,a\|\theta^Q)$	Action–value function parameterised by θ^Q.
Q	State weighting matrix in cost function.
R	Input weighting matrix in cost function.
r_t	Instantaneous reward at time step t.
R_i	Reward components of terms i.
s_t	State vector at time step t.
s'	Next state in reinforcement learning.
$s_{x,ij}$	Longitudinal slip ratio of wheel ij.
S	Sliding surface in SMC control law.

T_{ij}	Torque applied at wheel ij.
ΔT_{ij}	Torque correction term.
v_x	Longitudinal velocity.
v_y	Lateral velocity.
v_e	Longitudinal velocity error.
V	Vehicle speed magnitude.
w_i	Weighting coefficients in reward function.
Y_e	Lateral deviation from reference trajectory.
X	Longitudinal vehicle position.
Y	Lateral vehicle position.
λ_{ij}	Tyre slip ratio at wheel ij.
κ_{ij}	Longitudinal slip ratio at wheel ij.
α_f	Front tyre slip angle.
α_r	Rear tyre slip angle.
α_π	Actor learn rate.
α_Q	Critic learn rate.
α_{ij}	Slip angle of tyre ij.
δ	Steering angle.
δ_f	Front wheel steering angle.
δ_r	Rear wheel steering angle.
ψ	Vehicle yaw angle.
ψ_{des}	Desired yaw angle.
$\dot{\psi}$	Yaw rate.
$\dot{\psi}_e$	Yaw rate error.
$\dot{\beta}$	Sideslip rate.
t	Time.
Δt	Sampling time.
k	Discrete time index.
N	Prediction horizon length.
A	State transition matrix.
B	Input matrix.
P	Terminal weight matrix in MPC.

i, j	Wheel indices (fl, fr, rl, rr).
R_w	Effective wheel radius.
I_z	Vehicle yaw moment of inertia.
I_ω	Wheel rotational inertia.
g	Gravitational acceleration.
h_g	Height of vehicle centre of gravity.
ℓ_f	Distance from centre of gravity to front axle.
ℓ_r	Distance from centre of gravity to rear axle.

Chapter 1

Introduction

1.1 Electric Vehicles

The growing adoption of electric vehicles (EVs) is a major step in the modern transportation sector. EVs offer higher energy efficiency and reduce greenhouse gas emissions, air pollution, and dependence on fossil fuels, supporting the global shift towards cleaner and more sustainable mobility. This shift from conventional internal combustion engine (ICE) vehicles to electrified powertrains is driven by technological advancements, supportive government policies, and rising demand for sustainable transport solutions [1, 2]. However, the development and operation of EVs introduce several technical challenges that must be addressed to ensure reliable, efficient, and safe performance. Among these, vehicle dynamic stability and its energy management are of primary importance.

1.2 Vehicle Dynamics and Energy Management

The dynamic performance and energy efficiency of EVs are strongly influenced by their control strategies, particularly in all-wheel drive (AWD) configurations equipped with independently controlled in-wheel electric machines. EVs with the AWD powertrain architecture offer the capability to generate and control torque at each wheel individually. This feature provides significant potential for both stability control and energy optimisation through a torque vectoring scheme to allocate optimal torques to the individual wheels.

Torque vectoring enables active control of the vehicle trajectory and enhances cornering capability, stability, and driver confidence, particularly during high-speed or low-friction manoeuvres. Conventional stability control systems rely on hydraulic braking or simplified control laws, which often result in slow response and reduced efficiency. In contrast, using

torque vectoring in electrified powertrains provides faster response, smoother actuation, and higher control precision. In addition to stability enhancement, effective torque vectoring directly influences the overall energy consumption of the vehicle. By optimising the distribution of torques among the four wheels according to road and vehicle conditions, power losses can be minimised and efficiency can be improved. The next section presents the motivations behind this research, highlighting the necessity for an intelligent control framework capable of addressing both energy optimisation and vehicle stability within a single control architecture.

1.3 Motivations

The growing development and increasing adoption of EVs have introduced both new opportunities and unique challenges to the field of automotive engineering and vehicle control systems. Additionally, the integration of electric machines, as the heart of EVs, in AWD architectures allows precise torque vectoring capabilities by distributing torques among the wheels of the vehicle. Exploiting this capability, however, requires control structures that can cope with the objectives in vehicle dynamics, yaw stability control, overall vehicle safety, and the energy consumption of the electric machines [3–6]. Studies report that well-designed control strategies can decrease the slip-related power consumption by 17% [7], or enhance the vehicle efficiency by 10% [8].

Maintaining vehicle stability is a fundamental requirement for ensuring safety and handling performance. Under dynamic manoeuvres, such as cornering or sudden lane changes, the vehicle may experience instability in the form of understeer or oversteer, which can lead to loss of control if not properly managed. Achieving stable behaviour under different road and driving conditions requires practical control systems to ensure vehicle dynamics enhancement in real time. Therefore, stability control remains a key area of focus in this research, both as an independent objective and as part of the integrated energy management framework. The implementation of an integrated control system that manages both energy optimisation and dynamic stability remains a challenge. Traditional control approaches are often designed to address these aspects separately and are limited in handling nonlinear and coupled vehicle behaviours under changing driving conditions. Moreover, model-based controllers depend heavily on accurate system models and predefined parameters, which can limit adaptability to real-world uncertainties.

Recent advancements in artificial intelligence (AI), particularly reinforcement learning (RL), have provided new opportunities to overcome these limitations. RL-based control methods can learn optimal control policies directly through interaction with the vehicle

model or simulation environment, without relying on an explicit mathematical model. Such approaches can adapt to variations in road friction, vehicle speed, and driver input while maintaining efficient and stable performance. These capabilities make RL a strong candidate for integrated control of vehicle dynamics and energy management in AWD EVs. The motivation for this research arises from the need to develop an intelligent control framework that ensures stability and also combines energy management and vehicle dynamics control in a unified structure.

1.4 Research Contributions

This research advances the field of EV dynamic control through the design and verification of deep RL-based frameworks for stability enhancement and energy optimisation in AWD EVs. The main contributions are summarised as follows:

- **Establishment of a Hierarchical Framework for Intelligent Torque Vectoring Control**
  A detailed investigation of hierarchical control approaches for vehicle stability and energy optimisation is carried out. The study presents a hierarchical structure for the implementation of intelligent model-free RL techniques. The analysis also highlights the limitations of traditional methods in managing nonlinear, time-varying vehicle dynamics and adapting to uncertain road conditions. It also emphasises the advantages of learning-based algorithms in achieving real-time adaptability and multi-objective control. The outcomes of this work define the motivation for developing an integrated intelligent control framework that can address both energy management and vehicle dynamics within a unified torque vectoring structure. This work has been published in the journal publication:

  Jafari, R., A., Refaat, S.S., Paykani, Asef, P., and Sarhadi, P. *Reinforcement Learning for Torque Vectoring in Electric Vehicles: A Review of Stability and Energy Optimization Methods. IEEE Open Journal of Vehicular Technology* [9].

- **An Adaptive Reinforcement Learning-Based Yaw Stability Controller**
  A hierarchical control structure is developed with an RL-based torque allocator to enhance the yaw stability and dynamic performance of an AWD EV. Furthermore, an adaptive yaw stability control framework is developed using an RL approach to enhance vehicle stability and manoeuvrability under varying road and driving conditions. The proposed controller adopts a hierarchical architecture consisting of a high-level RL-based torque vectoring controller and a low-level torque allocation module. The

high-level controller employs an RL algorithm, integrated with a curriculum learning strategy, to progressively learn optimal control policies through interaction with the vehicle environment. The introduction of curriculum learning enables the agent to train across tasks of increasing complexity, improving convergence stability and adaptability to diverse conditions such as variations in steering, velocity, and road friction.

Unlike conventional model-based controllers, the proposed model-free RL controller eliminates dependency on precise vehicle models and manual parameter tuning, offering enhanced robustness against uncertainties in system dynamics. The developed control algorithm is implemented on a detailed nonlinear vehicle model in MATLAB/Simulink and verified using IPG CarMaker under different tests. The results demonstrate significant improvements in yaw rate tracking, sideslip angle reduction, and overall dynamic stability compared with traditional model-based controllers.

The hierarchical controller with RL-based torque allocator is presented in the conference paper:

Jafari, R., Sarhadi, P., Paykani, A., Refaat, S.S., and Asef, P. *Optimal Torque Allocation for All-Wheel-Drive Electric Vehicles Using a Reinforcement Learning Algorithm.* In 2024 *13th Mediterranean Conference on Embedded Computing (MECO)*, pp. 1-5. IEEE, 2024 [10].

The findings of that study on using a model-free RL-based algorithm with curriculum learning are extended and explained in the journal publication:

Jafari, R., Sarhadi, P., Paykani, A., Refaat, S.S., and Asef, P. *A TD3-Based Reinforcement Learning Algorithm With Curriculum Learning for Adaptive Yaw Control in All-Wheel-Drive Electric Vehicles. IEEE Access*, 2025 [11].

- **Integration of Energy Optimisation and Stability Control Using Deep Reinforcement Learning**
  A unified control framework based on RL is proposed to achieve simultaneous energy optimisation and stability control of AWD EVs. The proposed approach integrates both objectives within a single decision-making structure, enabling the control agent to optimise the overall power distribution among the four wheels while maintaining vehicle stability under varying driving conditions.

  The framework employs a multi-objective reward function that combines terms related to yaw rate tracking, sideslip angle regulation, and power loss minimisation. Real efficiency maps of the electric machines are incorporated into the training process to ensure realistic modelling of energy consumption. The control agent learns to

generate optimal corrective yaw moments and torque commands through continuous interaction with the vehicle environment, without the need for explicit analytical models or predefined optimisation procedures. The results demonstrate that the RL-based controller significantly improves energy efficiency and yaw stability compared with conventional optimisation-based methods. This work has been published in:

Jafari, R., Sarhadi, P., Paykani, A., Refaat, S.S., and Asef, P. *Integrated Energy Optimisation and Stability Control Using Deep Reinforcement Learning for an All-Wheel-Drive Electric Vehicle. IEEE Open Journal of Vehicular Technology*, 2025 [9].

1.5 Thesis Outline

This thesis is organised into several chapters, each dedicated to critical aspects of this research. The outline below provides a roadmap of the content covered in each chapter.

Chapter 1 introduced the research motivation and contributions of this study. It began with an overview of EVs and highlighted the challenges in achieving both vehicle stability and energy efficiency. Chapter 2 reviews the fundamentals of yaw control and torque vectoring in EVs. It outlines hierarchical control structures, surveys conventional model-based strategies for stability and energy optimisation, and introduces RL as a practical data-driven alternative. The chapter concludes by identifying research gaps that motivate the proposed RL-based integrated control framework.

Chapter 3 presents the development of RL-based torque vectoring controllers for AWD EVs. It introduces the hierarchical control framework and details the design and implementation of RL-based controllers. The algorithms are trained to optimise wheel torque distribution for yaw stability and vehicle handling without requiring explicit system models. Simulation results demonstrate that the proposed RL-based controllers achieve superior yaw rate tracking, sideslip regulation, and lateral stability compared with conventional methods.

Chapter 4 presents an integrated RL–based framework for torque vectoring and energy optimisation in AWD EVs. It introduces three actor–critic algorithms implemented within a hierarchical control architecture to achieve simultaneous yaw stability and energy efficiency. The chapter details the multi-objective reward design, network architecture, and training strategy that enable the controllers to learn energy-aware torque distribution directly from interaction with the vehicle environment. Extensive simulations and IPG CarMaker tests overall demonstrate that the proposed RL controllers outperform conventional model-based methods in dynamic stability, energy consumption, and computational efficiency.

Chapter 5 concludes the thesis by summarising the key findings, contributions, and limitations of the proposed RL-based control frameworks for torque vectoring and energy optimisation. It highlights the successful integration of RL-based algorithms within a hierarchical architecture that achieves simultaneous improvements in stability, manoeuvrability, and energy efficiency. Finally, some directions for future research are outlined, setting the foundation for the next works on intelligent and energy-efficient vehicle control systems.

Chapter 2

Background and Literature Review

2.1 Introduction

The transition towards renewable energy has increased the demand for electrified transportation, with EVs playing a central role in reducing emissions and improving energy efficiency [12, 13]. Compared with conventional vehicles, electric drivetrains convert around 77% of grid energy to wheel power, whereas gasoline vehicles are capable of converting just 12–30% to power at wheels, leading to lower energy use per mile [14, 15]. In addition, EVs provide advanced energy-aware control strategies aimed at improving both vehicle dynamics and overall efficiency [16]. The availability of multiple electric machines and independent actuators enables practical torque vectoring, which refers to the active control of individual wheel torques to generate a desired yaw moment and enhance vehicle stability [17–20]. At the same time, the ability to adjust wheel torques allows the controller to reduce energy consumption through the allocation of optimal torque values to the vehicle [21–24].

Conventional torque vectoring and energy management methods, in most cases, rely on rule-based or model-based control techniques. Whereas these methods are well-understood and widely deployed, they typically depend on accurate modeling of vehicle dynamics and environmental conditions [25, 26]. As a result, their performance can degrade under uncertainty, nonlinearity, or unmodeled disturbances. Moreover, predefined control laws often lack the flexibility to adapt to rapidly changing conditions such as variable road friction, unbalanced loading, or aggressive driving scenarios. These limitations have motivated the investigation of model-free and data-driven control strategies, such as RL as a promising framework for learning control policies through interaction with the environment [27–31]. As EVs continue to evolve with increasingly complex dynamics and control requirements, there is a growing need to explore advanced strategies that can offer greater adaptability and performance.

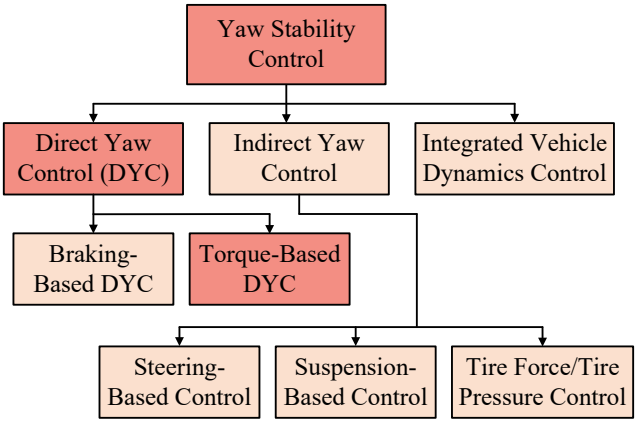


Fig. 2.1 A classification of yaw stability control approaches.

2.2 Fundamentals of Yaw Control in Vehicle Dynamics

Yaw stability control, which refers to the ability of the vehicle to maintain controlled rotation about its vertical axis, is particularly critical in high-performance or low-traction conditions [32–35]. This section presents a classification and comparison of different yaw stability control methods, with a focus on those that lead toward direct yaw control (DYC) via torque vectoring, which is the main scope of this work. Fig. 2.1 presents the classification of yaw stability control approaches, with the subcategories relevant to the scope of this study emphasised using a darker shade. Yaw stability control strategies in EVs can be generally classified into the following subclasses based on how they influence the yaw moment to maintain lateral stability:

- Direct yaw control (DYC)
- Indirect yaw control
- Integrated vehicle dynamics control

Each method differs in terms of control authority, complexity, and response speed. Among these, DYC has received particular attention due to its effectiveness in high-speed and low-adhesion scenarios, especially when implemented through torque vectoring. The following paragraphs describe each category in detail, beginning with DYC.

In a DYC, corrective yaw moments are directly applied through differential braking or torque distribution as demonstrated in Fig. 2.2 to maintain vehicle stability [36–38]. The arrows represent the direction and type of forces involved. In braking-based DYC, yaw moments are generated by applying a braking force to the vehicle [39, 40] that helps the vehicle to maintain its stability, as depicted in Fig. 2.2(a). However, limitations in efficiency

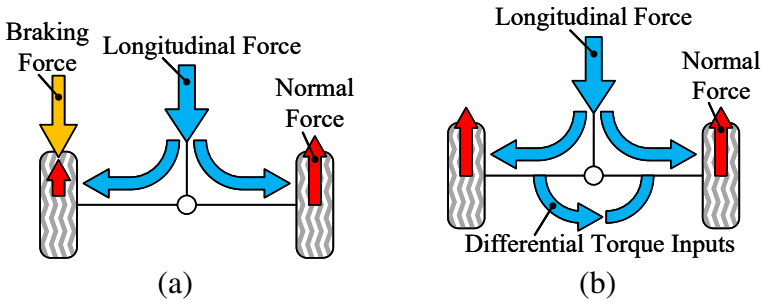


Fig. 2.2 Direct yaw control approaches: (a) Braking-based, and (b) Torque-based methods.

and durability make this approach less suitable for long-term or performance-oriented applications in EVs [41–43]. The other subcategory of DYC is the torque-based approach, commonly referred to as torque vectoring, which distributes drive torque across wheels, as shown in Fig. 2.2(b), to generate yaw moments more efficiently than the braking-based method. By increasing torque on one side and reducing it on the other, a differential driving force is created, producing a yaw moment without braking [44, 45]. This method allows for smoother, continuous control without braking losses, improving energy efficiency and reducing mechanical wear. The review in this chapter focuses primarily on torque-based DYC methods, emphasizing their design, integration with energy optimisation schemes, and the recent advancements [46–48].

Indirect yaw control strategies improve vehicle stability by modifying vehicle parameters that influence yaw dynamics, rather than directly generating yaw moments. These approaches generally provide slower response and lower effectiveness under extreme conditions compared with DYC. Indirect methods include steering-based, suspension-based, and tyre force or tyre pressure control strategies. Steering-based systems, such as active front steering (AFS) and rear-wheel steering (RWS), can improve manoeuvrability but add mechanical complexity and might limit effectiveness in certain conditions [49–52]. Suspension-based control adjusts damping and stiffness to influence vehicle behaviour, although it requires accurate state estimation and increases system complexity [53–56]. Tyre force or pressure control can also affect yaw stability, but it offers limited control authority and requires precise monitoring [57–59]. Integrated vehicle dynamics control coordinates multiple subsystems, such as braking, steering, and suspension, to improve overall stability and handling. By combining their capabilities, integrated approaches can simultaneously address multiple performance metrics and enhance robustness. However, integrated control systems are often complex to implement, require high computational effort, and present challenges in real-time coordination and controller tuning [60–64].

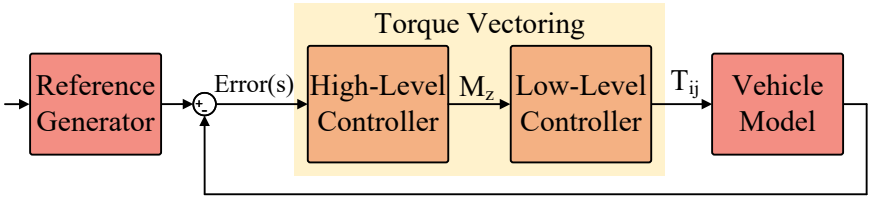


Fig. 2.3 A schematic of a simple hierarchical torque vectoring algorithm.

2.3 Torque Vectoring With Hierarchical Structure

Torque vectoring is a fundamental control strategy in EVs that allows precise adjustment of wheel torques to generate the desired yaw moment. By actively distributing torque across wheels or axles, torque vectoring enhances cornering response, reduces understeer and oversteer tendencies, and maintains lateral stability even under aggressive manoeuvres or adverse road conditions. This capability is especially important for applications where rapid and accurate control over vehicle behaviour is required [65–68].

To manage the complexity of coordinating multiple actuators and objectives, torque vectoring strategies are often implemented using hierarchical control architectures [69–71], as shown in Fig. 2.3. A typical hierarchical framework consists of a high-level controller that determines the desired yaw moment or vehicle behaviour based on objectives, such as path tracking or stability margins. This is followed by a low-level controller or torque allocator, which distributes the total required yaw moment among the individual wheels or axles while satisfying actuator limitations and optimizing secondary objectives like energy efficiency [72]. Hierarchical structures also contain a model of the vehicle to capture its dynamic behaviour and a reference generator model to compute desired trajectories or target states based on driver inputs and operating conditions [73–75].

The main objective of torque vectoring is to improve lateral stability and manoeuvrability by regulating the yaw moment. Depending on the control design, the focus can be on yaw rate tracking, sideslip angle regulation, or a combination of both to ensure accurate path following and stable behaviour under demanding conditions [76–82]. In EVs, where independent wheel torque control is available, energy efficiency can also be incorporated into the control objectives [83–87]. Several studies have proposed integrated frameworks that balance stability and energy consumption, demonstrating measurable energy savings while maintaining yaw stability [88, 89].A torque allocation strategy aimed at enhancing vehicle stability and reducing energy consumption was proposed by Zhi et al. [90]. The developed method improved the average motor efficiency by 2.4% to 2.9% across various driving cycles and increased the driving range by 4% to 5.5% compared to a conventional torque distribution approach. However, these model-based approaches often require high-fidelity vehicle models

and significant computational resources, which limits their applicability in real-time control under dynamic or uncertain conditions. In addition, conventional control methods frequently struggle to balance multiple conflicting objectives within a unified framework, particularly in nonlinear and time-varying driving environments [91–93].

2.4 Conventional Torque Vectoring Strategies For Stability Enhancement And Energy Optimisation

Torque vectoring strategies can be broadly categorised into three main categories:

- Model-based controller
- Model-free non-RL controller
- Model-free RL-based controller

Model-based approaches rely on explicit vehicle dynamic models and control theory to calculate the required yaw moment or wheel torques. These include methods such as linear quadratic regulator (LQR), MPC, SMC, and adaptive control. In contrast, model-free non-RL strategies use strategies that do not explicitly incorporate system models but are often limited in adaptability and performance optimisation. Model-free RL-based controllers, on the other hand, learn optimal torque vectoring policies through direct interaction with the environment and have obtained increasing attention for their effectiveness in simultaneously achieving stability and energy optimisation in EVs. Fig. 2.4 illustrates the classification of different torque vectoring strategies. The blocks shown in the darker shade highlight the subclasses that fall within the scope of RL-based torque vectoring strategies. To provide a structured understanding of conventional torque vectoring strategies and to establish a foundation for

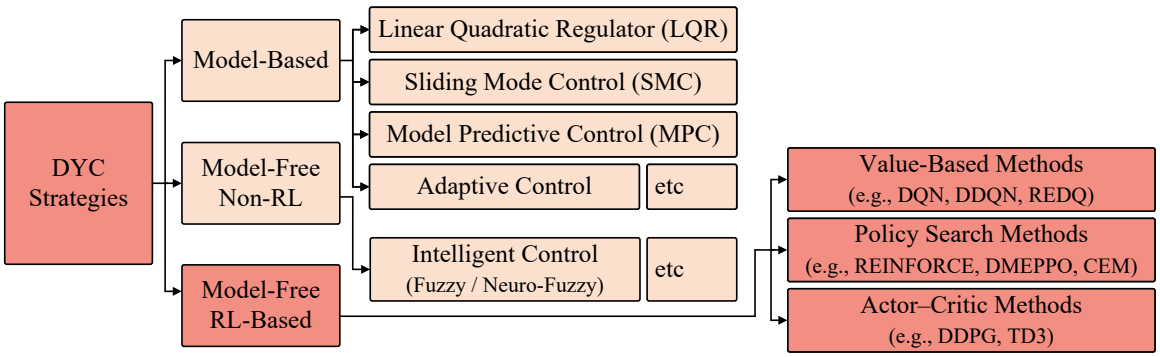


Fig. 2.4 Classification of direct yaw control strategies.

comparison with RL-based methods, a review of commonly used model-based hierarchical control techniques and model-free non-RL methods is presented. Each algorithm is introduced with a brief overview, followed by representative applications for EV yaw stability and energy optimisation.

One of the most commonly applied model-based strategies in vehicle yaw control is the LQR, which offers a balance between simplicity, optimality, and computational efficiency. The objective is to track a desired yaw rate and/or sideslip angle by applying corrective yaw moments M_z. Even though LQR assumes linear dynamics and constant vehicle parameters, its simplicity and real-time feasibility make it a common baseline in the design of model-based yaw control systems [94–96]. A representative example of LQR-based torque vectoring was presented in the work by Antunes et al. [97], which implemented and experimentally validated PI and LQR-based torque vectoring controllers for an electric Formula Student vehicle. Simulation and real-vehicle tests demonstrated that the PI controller improved yaw tracking and reduced lap time by 7.6%, whereas the LQR controller achieved enhanced robustness to variations in tyre–road friction and lateral velocity estimation. However, the reliance on linear time-invariant dynamics and sensitivity to parameter variations limit the adaptability of LQR in the nonlinear and uncertain conditions typical of real-world EV operation. The nonlinear model diagram of the proposed torque vectoring scheme is shown in Fig. 2.5 for simulation and testing. The model included detailed dynamics and tyre forces but relied on linear approximations for control, which limited the robustness under nonlinear and uncertain conditions.

SMC is a robust nonlinear control technique commonly applied in vehicle dynamics due to its insensitivity to model uncertainties and external disturbances [98, 99]. SMC aims to force key vehicle states to follow desired reference values by driving the system states onto a predefined sliding surface and maintaining motion along it. Unlike linear controllers, SMC maintains control performance under parameter variations, which makes it particularly suitable for torque vectoring applications. The core of SMC design lies in the definition of a

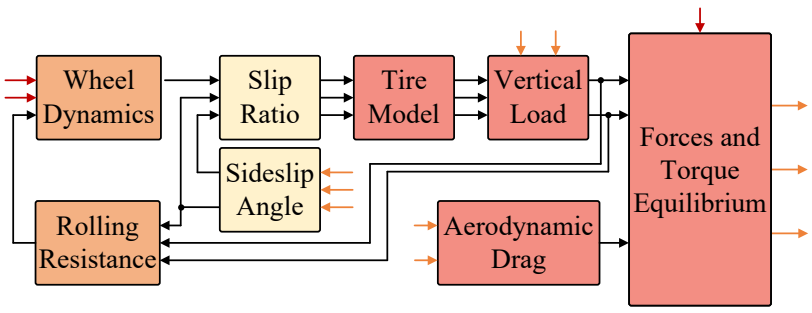


Fig. 2.5 Nonlinear model diagram of torque vectoring by Antunes et al. [97].

sliding surface [100, 101], typically expressed as:

$$s(x) = \dot{e} + \lambda e \tag{2.1}$$

where $e = \dot{\psi} - \dot{\psi}_{des}$ is the tracking error between the actual yaw rate $\dot{\psi}$ and its desired value $\dot{\psi}_{des}$, and $\lambda > 0$ is a positive constant that shapes the convergence rate of the error dynamics. The control input is then formulated as:

$$u = u_{eq} - k \cdot \text{sign}(s) \tag{2.2}$$

where u_{eq} is the equivalent control that maintains $s = 0$, and the term $-k \cdot \text{sign}(s)$ ensures finite-time convergence to the sliding surface. To reduce high-frequency chattering, the discontinuous sign function is often replaced with a saturation function:

$$\text{sign}(s) \rightarrow \text{sat}\left(\frac{s}{\phi}\right) \tag{2.3}$$

where ϕ defines the thickness of the boundary layer used to smooth the control action. In vehicle control, SMC enables real-time yaw stabilization by distributing the yaw moment across wheels, though it often causes chattering—oscillatory behaviour, which can lead to mechanical wear or instability in actuator systems. This issue can be typically mitigated using higher-order SMC or smooth approximations of the sign function, albeit with increased design complexity [102].

Building upon these fundamental principles, SMC has been used in vehicle dynamics problems. Xu et al. [103] presented a hierarchical torque vectoring framework for in-wheel motor EVs, combining LQR and high-order SMC at the high level for economy and stability control under parameter uncertainties. The schematic of the proposed robust optimal controller is illustrated in Fig. 2.6. This hierarchical configuration demonstrates how feedback-based stability control and optimisation-based energy management can be coordinated within a unified structure. Simulation results verified that the proposed strategy improved tracking precision, smoothed control signals, and reduced power loss compared with standard LQR-based control, although its performance still depends on accurate modeling of vehicle parameters and assumed ideal measurement availability.

MPC is an advanced control strategy that uses a dynamic model to predict the future behaviour of a system over a finite time horizon. In DYC applications, MPC computes the optimal sequence of control inputs by solving a constrained optimisation problem at each time step, based on the current state of the vehicle [104–106]. The standard MPC formulation

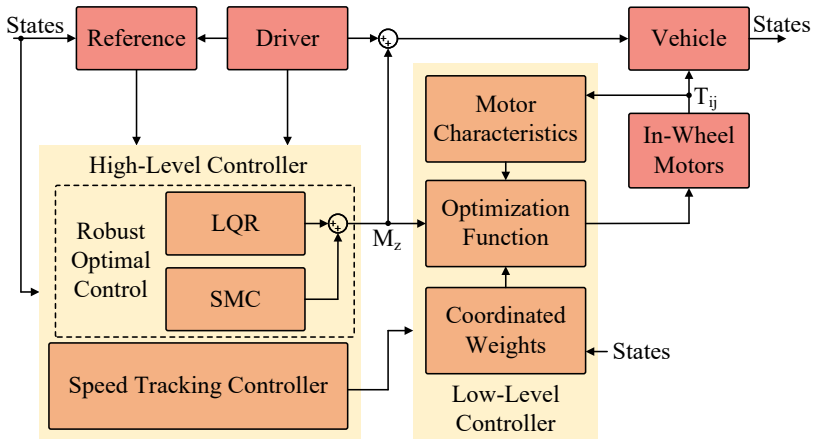


Fig. 2.6 Robust optimal control for stability and energy optimisation by Xu et al. [103].

solves the following quadratic optimisation problem:

$$\min_{U} \sum_{k=0}^{N-1} \left(x_k^\top Q x_k + u_k^\top R u_k \right) \tag{2.4}$$

subject to:

$$x_{k+1} = A x_k + B u_k \tag{2.5}$$

$$x_k \in \mathcal{X}, \quad u_k \in \mathcal{U} \tag{2.6}$$

where x_k and u_k denote the state and control input at prediction step k, Q and R are positive semi-definite weighting matrices, N is the prediction horizon, and $\mathcal{X}, \mathcal{U}$ define the admissible sets for states and inputs. MPC is well-suited for high-performance applications due to its constraint-handling capabilities, but it relies on accurate linear models and may degrade under nonlinear conditions or large disturbances [107–109].

NMPC extends MPC by using a full nonlinear vehicle model within the prediction horizon. This allows for capturing the effects of nonlinear tyre forces, load transfer, and saturation, which are critical under aggressive driving manoeuvres or low-adhesion surfaces [110–112]. NMPC can be employed to compute more accurate corrective yaw moments and improve robustness across a wider range of operating conditions [113, 114]. The general NMPC problem is formulated as:

$$\min_{U} \int_0^T \left(x(t)^\top Q x(t) + u(t)^\top R u(t) \right) dt \tag{2.7}$$

subject to:

$$\dot{x}(t) = f(x(t), u(t)) \tag{2.8}$$

$$x(t) \in \mathcal{X}, \quad u(t) \in \mathcal{U} \tag{2.9}$$

where $f(x, u)$ represents the nonlinear dynamics, and the constraints $\mathcal{X}$ and $\mathcal{U}$ may include actuator limits, tyre saturation bounds, and safety-critical regions. Whereas NMPC provides better accuracy and constraint satisfaction than linear MPC, it comes with a significant computational cost. Solving nonlinear optimisation problems online may require tailored solvers or reduced-order models to meet real-time deadlines in embedded systems.

Deng et al. [115] proposed a hierarchical torque vectoring strategy for EVs with mechanical elastic electric wheels (MEEW), targeting both stability and energy efficiency. As shown in Fig. 2.7, the proposed control architecture integrates multiple coordinated subsystems. The algorithm exploits a parameter identification module that determines the MEEW tyre model using a Fibonacci optimisation approach to capture nonlinear tyre characteristics. The high-level controller combines a linear parameter varying (LPV) LQR and an NMPC function to compute the required additional yaw moment and longitudinal force, balancing handling precision and robustness under varying road conditions. These control signals are delivered to the low-level controller, which allocates torque among the four in-wheel motors through rule-based and coordinated optimisation methods. The coordinated control mechanism dynamically adjusts the weighting between stability and energy-saving objectives via a coordination weight adjustment module, ensuring that the control emphasis shifts adaptively with vehicle dynamics. Simulation and HIL tests demonstrated that the method improved handling by up to 94% and reduced energy consumption by up to 35.1% under various driving cycles.

Dalboni et al. [116] developed an NMPC-based integrated control framework for torque vectoring and anti-roll moment distribution in four-wheel-drive EVs, demonstrated in Fig. 2.8. The torque-vectoring and anti-roll moment distribution module forms the core of the control system, simultaneously regulating the distribution of wheel torques and the front-to-rear anti-roll moments. It receives feedback from the vehicle model and coordinates its outputs with the brake-blending and suspension-force distribution blocks. The vehicle model, implemented in VSM, provided high-fidelity dynamic feedback for closed-loop evaluation. Simulation results demonstrated improved yaw tracking, reduced power losses, and maintained vehicle stability compared to rule-based and PI controllers.

Another paradigm in dynamic yaw control uses adaptive controllers that update system models online to adjust controller parameters, directly or indirectly [117–119]. Whereas some

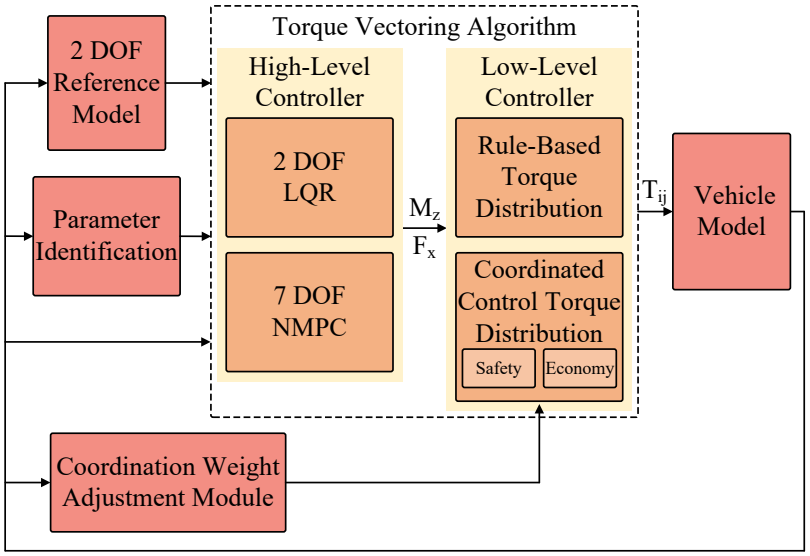


Fig. 2.7 Torque vectoring algorithm for stability and economy enhancement by Deng et al. [115].

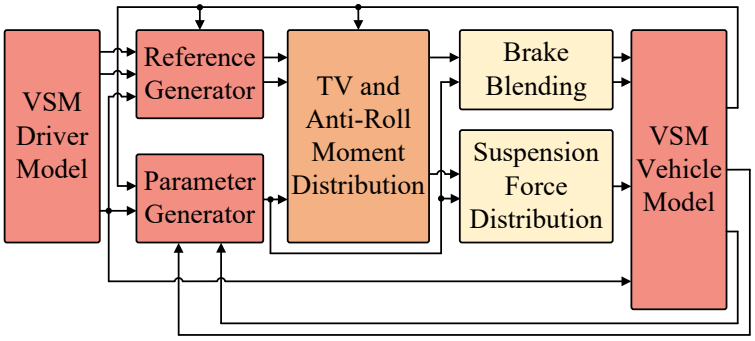


Fig. 2.8 Schematic of NMPC for integrated energy-efficient torque vectoring and anti-roll moment distribution by Dalboni et al. [116].

target higher-level chassis controllers like AFS and DYC for dynamic stability, their online adaptation methods can be of interest. For example, Fig. 2.9 shows a hierarchical control strategy combining model reference adaptive control (MRAC) with an optimisation-based actuator allocation from [120]. The high-level adaptive controller generated desired yaw moment and steering actions, and a low-level controller allocated those commands between braking and steering actuators for efficient yaw moment distribution. The high-level multiple-input multiple-output (MIMO) MRAC generated the desired yaw moment and steering angle, adapting to uncertainties in vehicle mass and tyre–road friction. At the low level, the yaw moment was distributed between AFS and DYC via constrained optimisation, respecting tyre

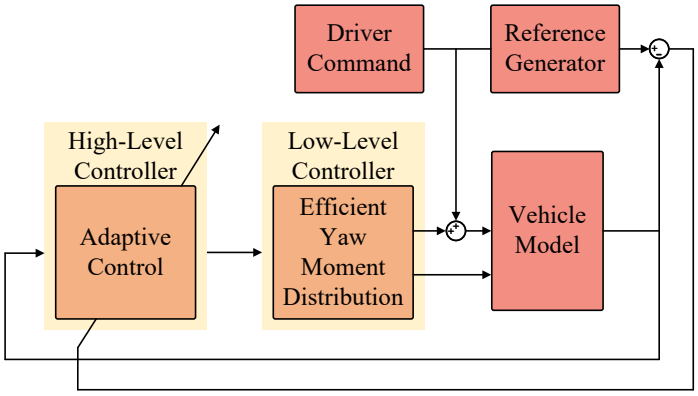


Fig. 2.9 Integrated chassis control structure by Ahmadian et al. [120].

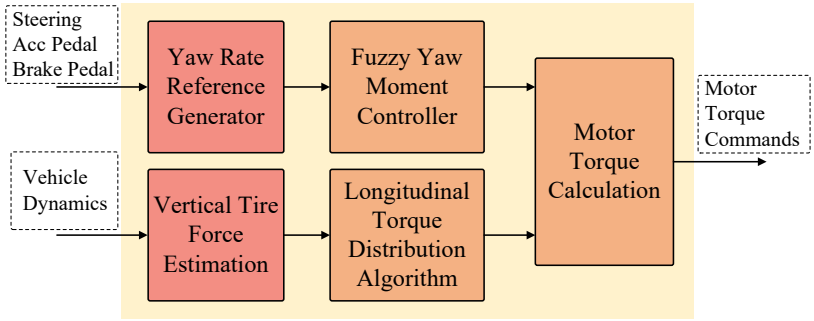


Fig. 2.10 Schematic of the model-free non-RL torque vectoring system by Parra et al. [8].

force and stability limits. Simulation results demonstrated the ability of the controller to improve yaw rate tracking, reduce sideslip angle, and maintain lateral stability.

An example of a model-free non-RL torque vectoring approach was presented by Parra et al. [8] for per-wheel motor EVs, combining a fuzzy logic-based yaw controller and an ANFIS-based vertical tyre force estimator, as depicted in Fig. 2.10. The block diagram shows a fuzzy yaw controller and a vertical tyre force estimator jointly contributed to determining the motor torque commands, without relying on a complete vehicle model. The system, implemented on a real-time platform and validated in high-fidelity simulations, improved yaw tracking, reduced slip ratio and sideslip. The intelligent torque vectoring system achieved better path tracking, reduced control effort, and lowered mechanical energy consumption by approximately 10% compared to a PID baseline. This study highlighted the effectiveness of model-free, non-RL control in over-actuated EVs.

Conventional torque vectoring methods, though effective, are limited by their dependence on accurate models, fixed parameters, and predefined control laws, which can lead to degraded performance under nonlinear dynamics, changing road conditions, or modeling inaccuracies. Furthermore, tuning model parameters and weighting matrices often requires

Table 2.1 Comparison of conventional torque vectoring algorithms reviewed in this study

Reference	Control Method	Control Objectives	Key Findings
Antunes et al. [97]	PI, LQR	Yaw rate tracking	PI controller improved yaw tracking and reduced lap time; LQR showed promising results despite limited logging data
Xu et al. [103]	LQR + SMC	Stability control and energy optimisation	Improved yaw stability and energy efficiency; reduced torque fluctuation and power loss (simulation-based)
Deng et al. [115]	LPV LQR + NMPC	Stability and energy efficiency	Stability improved by 94% and energy use reduced by 35%; validated via HIL and driving cycles
Dalboni et al. [116]	NMPC	Yaw rate tracking, energy efficiency, and stability	Torque vectoring and anti-roll moment distribution improved stability and energy efficiency; outperformed PI and rule-based controllers
Ahmadian et al. [120]	MRAC + optimisation-based allocation	Yaw rate tracking, sideslip regulation, stability under uncertainty	Adaptive controller improved yaw tracking and maintained lateral stability; robust to mass and tyre–road friction variations
Parra et al. [8]	Fuzzy Logic + ANFIS	Yaw rate and sideslip control, energy efficiency	Improved yaw tracking, 10% energy saving, reduced slip and enhanced path tracking; real-time capable

expert knowledge and offline calibration. These limitations reduce the adaptability and scalability of model-based and model-free non-RL approaches. In contrast, model-free RL-based algorithms offer greater flexibility and learning capability by directly interacting with the environment and adapting policies based on performance feedback. Moreover, RL-based structures outperform model-free non-RL methods in terms of optimality over time, flexibility in multi-objective optimisation, and better performance under uncertainty. RL-based schemes provide a more straightforward framework for handling multi-criteria design optimisation compared to traditional approaches, especially when the objective is optimizing the energy efficiency and improving stability at the same time. A summary of the conventional torque vectoring strategies discussed in this section is presented in Table 2.1, summarising their control methods, objectives, and key findings.

2.5 Overview of Reinforcement Learning for Vehicle Dynamics Application

RL is a framework where an agent learns to make decisions by interacting with an environment and receiving rewards as feedback. The goal is to learn a policy that maximises cumulative rewards over time [121, 122]. At each step, the agent observes a state, selects an action, and transitions to a new state while receiving a scalar reward [123]. In control applications, including vehicle dynamics and torque vectoring, RL offers a model-free alternative to traditional model-based control strategies. It removes the need for explicit modeling of system dynamics and can adapt to complex multi-variable interactions. As such, RL has been widely explored for applications where the system model is difficult to derive or where classical control methods struggle with high-dimensional or time-varying environments.

2.5.1 Reinforcement Learning Framework and Operational Principles

RL is defined by a set of core components that describe how an agent interacts with an environment and improves its control policy through experience [124, 125]. The main elements include the agent, environment, state, action, reward, policy, and value function. In torque vectoring applications, the agent represents the control algorithm, while the environment corresponds to the vehicle and its operating conditions. The state typically includes variables such as yaw rate, sideslip angle, vehicle velocities, steering input, and wheel speeds. Actions correspond to control commands, such as corrective yaw moment or individual wheel torques. The reward function quantifies the effectiveness of the action. The policy defines the mapping from states to actions, and value functions estimate expected future rewards to guide learning.

The operational process of RL is defined by an iterative loop through which the agent learns to make better decisions based on continuous interaction with the environment [126]. Fig. 2.11 illustrates the core RL loop. At each timestep, the agent observes the current state, selects an action according to its policy, and receives a reward and next state from the environment. Using this transition data, the agent updates its policy and value functions to improve long-term performance. This process is commonly organised into episodes representing complete driving scenarios [127, 128]. RL problems are formally described using a Markov decision process (MDP), defined by the tuple (s, a, p, r, γ), which specifies states, actions, transition probabilities, rewards, and a discount factor [129, 130]. Bellman equations provide the recursive relationships for state-value $V(s)$ and action-value functions $Q(s,a)$ and form the theoretical basis for deriving optimal policies [131, 132]. These

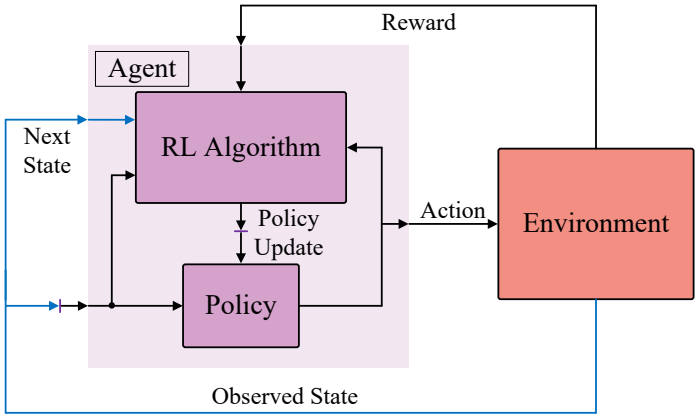


Fig. 2.11 A simple process of reinforcement learning.

principles provide the foundation for implementing RL-based control strategies in vehicle dynamics and torque vectoring.

2.5.2 Deep Reinforcement Learning

Deep RL combines reinforcement learning with deep neural networks to solve decision-making problems in environments with high-dimensional and continuous state-action spaces [133, 134]. Traditional RL methods rely on tabular representations or linear approximators, which are not suitable for complex systems such as vehicle dynamics control. Deep RL overcomes this by using deep neural networks to approximate value functions, policies, or environment models, enabling agents to learn directly from raw state variables, such as yaw rate, lateral velocity, or tyre slip ratios. Neural networks consist of multiple layers of interconnected units (neurons) that perform non-linear transformations on input data. This structure allows them to extract hierarchical features and generalise across a wide range of inputs. Fig. 2.12 shows a basic architecture of artificial neural networks. In deep RL, these networks are used to represent either the action-value function, the policy, or both, depending on the algorithm. This capability is especially useful in torque vectoring applications, where the relationship between control inputs and vehicle responses is nonlinear and difficult to model explicitly. The main advantage of deep RL is its ability to scale to high-dimensional vehicle states and learn control strategies without requiring an accurate model of the vehicle dynamics. A conceptual illustration of a policy gradient–based deep RL framework is presented in Fig. 2.13. The agent observes the current state, and a neural-network-based policy generates an action that is applied to the environment. The environment then returns the next state and an associated reward. The reward signal is used to update the parameters of

the policy network, enabling gradual improvement of the control policy through interaction with the environment.

2.5.3 Deep Reinforcement Learning Techniques

RL algorithms can be grouped into three main categories of value-based methods, policy search methods, and actor–critic methods [135]. Value-based methods aim to estimate the expected cumulative reward for each action without directly learning a policy [136, 137]. Typical examples include state–action–reward–state–action (SARSA), deep Q-network (DQN), Q-learning, and randomised ensemble double Q-learning (REDQ). These methods are generally suitable for discrete action spaces and do not maintain an explicit policy representation. Although effective and relatively simple, their scalability is limited in continuous control problems. Recent work has applied value-based methods, such as improved double deep Q-learning network (DDQN) architecture, to vehicle control tasks including trajectory tracking under varying speeds [138]. Policy search methods directly optimise the decision-making policy of the agent [139]. Some approaches, such as the cross-entropy method (CEM) and deep maximum entropy proximal policy optimisation (DMEPPO), are gradient-free and rely on sampling and selection. Others, such as REINFORCE with Monte Carlo target, fall under the class of policy gradient methods, which use the gradient of expected return to update the policy parameters. These methods are more suitable for continuous and high-dimensional action spaces but may suffer from high variance and data inefficiency [140].

Actor-critic methods combine both approaches by using one model (the critic) to estimate value functions and another (the actor) to update the policy [141]. Algorithms such as deep deterministic policy gradient (DDPG), twin delayed deep deterministic policy gradient (TD3), AlphaZero, and deep deterministic policy iteration (DDPI) follow this structure

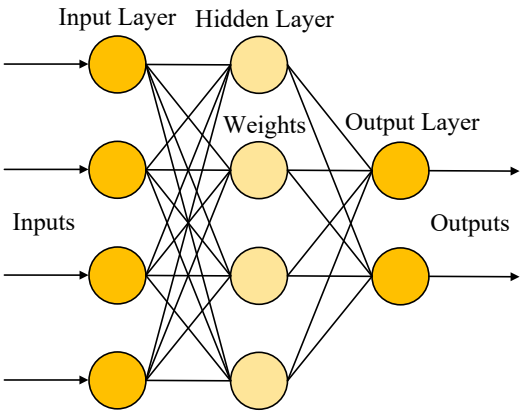


Fig. 2.12 A basic architecture of artificial neural networks.

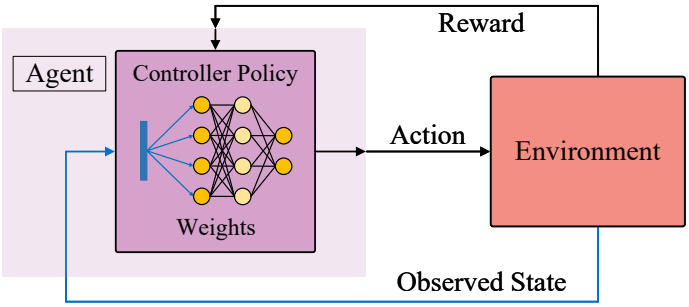


Fig. 2.13 Conceptual illustration of a policy gradient-based deep reinforcement learning framework.

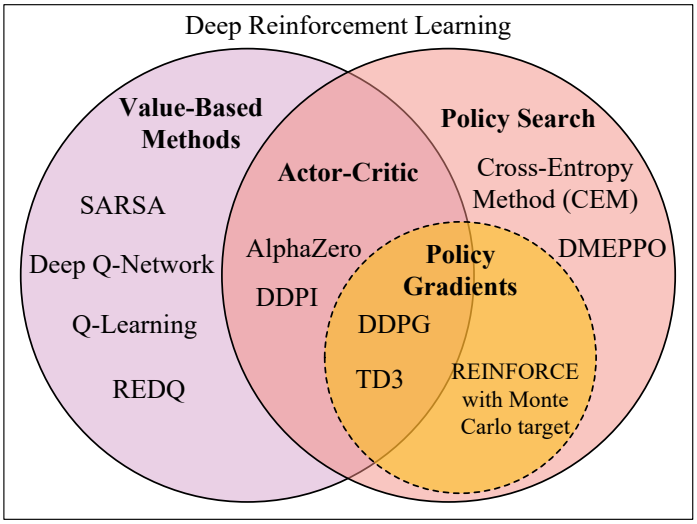


Fig. 2.14 Categorization of reinforcement learning techniques.

and are well suited to continuous control tasks such as torque vectoring in EVs. Recent studies have demonstrated the effectiveness of actor–critic frameworks in vehicle applications, including traction and speed control under varying driving conditions [142]. Due to their ability to handle continuous actions and high-dimensional dynamics, actor–critic methods are widely adopted in advanced vehicle control problems. Fig. 2.14 presents a Venn diagram summarizing this taxonomy, including representative methods as examples.

In addition to this taxonomy, RL algorithms can also be distinguished as on-policy or off-policy, depending on the relationship between the behavioural policy and the target policy. The behavioural policy is used to generate data through interaction with the environment, whereas the target policy is the policy being updated during learning. In on-policy methods, these two policies are identical, meaning that the agent improves the same policy it uses for data collection. This often leads to stable learning but may reduce data efficiency. In contrast, off-policy methods allow the target policy to differ from the behavioural policy, enabling

the reuse of previously collected data and improving sample efficiency. However, off-policy learning may require additional mechanisms to maintain stability.

2.5.4 Curriculum Learning in Reinforcement Learning

Curriculum learning is a training strategy in which a learning agent is exposed to tasks in a progressive order, starting from simpler scenarios and gradually increasing the level of difficulty [143, 144]. This approach allows the model to first acquire fundamental behaviours in easier tasks before adapting to more challenging conditions [145–149]. Such a progressive learning process can improve training stability, accelerate convergence, and enhance the generalisation capability of RL agents. Curriculum learning has attracted increasing attention in RL research, particularly in applications involving complex decision-making and control tasks.

The effectiveness of curriculum learning in improving the training performance of RL agents was investigated in [150]. The authors applied curriculum learning together with reward shaping to train a PPO-based agent in the VizDoom environment. The results showed that when the agent was trained directly in complex environments, the learning process became unstable and the agent struggled to achieve consistent performance. However, by introducing curriculum learning, where the training began with simpler tasks and gradually increased in difficulty, the agent was able to acquire fundamental skills before facing more challenging scenarios. This improved the learning efficiency and stability of the RL agent, leading to higher cumulative rewards and better task completion performance. Curriculum learning has also been applied to RL problems involving sparse reward structures. Curriculum learning was integrated with a PPO-based reinforcement learning framework to train an autonomous driving agent in the CARLA simulator in [151]. The agent was first trained in simplified driving scenarios and was then gradually exposed to more challenging conditions, including varying traffic density and environmental complexity. This progressive training strategy enabled the agent to incrementally acquire driving skills while maintaining training stability, highlighting the potential of curriculum learning for developing robust RL-based autonomous driving policies.

Curriculum learning has also been explored to improve the efficiency and robustness of RL in autonomous navigation tasks. In [152], a curriculum-based training framework was integrated with a SAC algorithm to enable mobile robots to navigate safely in unknown and dynamic environments. The proposed approach organised the training process through task-level curriculum scheduling and introduced a curriculum-based energy prioritisation mechanism for experience replay. This strategy allowed the agent to learn navigation behaviours progressively according to its current capability. Experimental results demon-

strated that the curriculum-based approach improved sampling efficiency, enhanced obstacle avoidance performance, and produced more stable navigation behaviour compared with conventional reinforcement learning training methods. By exploiting curriculum learning, RL agents can develop robust control policies in environments that would otherwise be difficult to learn from directly. These advantages make curriculum learning particularly suitable for control applications involving complex and dynamic systems. Consequently, curriculum learning provides a promising training strategy for RL-based controllers, which motivates its integration within the RL framework adopted in this research.

2.6 Reinforcement Learning-Based Torque Vectoring Strategies for Stability Enhancement and Energy Optimisation

RL has been increasingly applied to control problems in EVs [153, 154], particularly where system dynamics are nonlinear and difficult to model accurately. Classical torque vectoring strategies typically compute corrective yaw moments using simplified models and fixed-gain controllers, but are often sensitive to parameter variations and external disturbances. RL provides an alternative by learning torque distribution policies through interaction with the environment, enabling improved tracking performance and adaptability across varying driving conditions. In this section, a range of RL-based approaches are surveyed in the literature, highlighting their design principles, objectives, and achieved performance. These implementations serve as concrete evidence of the adaptability and effectiveness of RL in vehicle control scenarios.

REDQ is an off-policy deep reinforcement learning algorithm designed to improve learning stability, sample efficiency, and control performance in continuous action spaces. REDQ builds upon the soft actor-critic (SAC) framework by incorporating an ensemble of Q-value networks and randomised target computation, which collectively reduce overestimation bias and improve convergence. The core idea is to maintain multiple critics and use a subset of them at each update step to compute the target Q-value. The REDQ architecture includes one actor network and an ensemble of N critic networks. During training, a random subset of M critics is selected from the ensemble to compute the target value, which is then used to update each critic. The actor is updated using stochastic policy gradients derived from the minimum Q-value over the sampled subset. This randomised ensemble strategy enhances robustness and prevents the critic from overfitting to biased Q-value estimates, which is a common issue in deep value-based methods [155, 156].

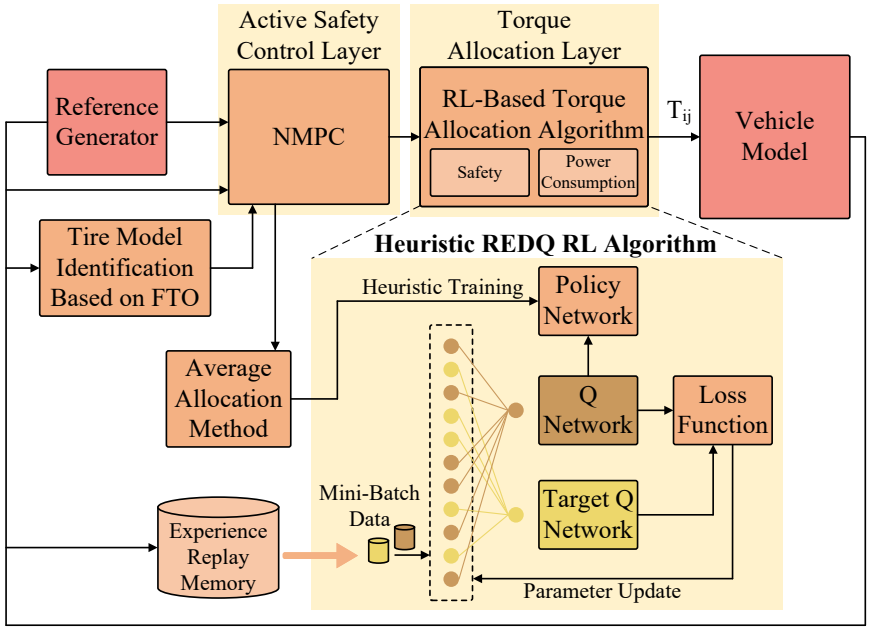


Fig. 2.15 Schematic of the RL-based torque vectoring by Deng et al.$_a$ [157].

A hierarchical torque vectoring control strategy for four in-wheel-motor EVs was proposed by Deng et al.$_a$ [157], which simultaneously addressed vehicle stability and energy consumption using deep RL. The control architecture consisted of two layers. The high-level controller employed NMPC to generate the reference longitudinal force and corrective yaw moment, while the low-level controller distributed the four-wheel torques using an RL-based strategy. The torque allocation layer was implemented using a heuristic REDQ algorithm, which learned optimal torque commands based on the current driving states, phase-plane stability indicators, and vehicle dynamic responses. In addition, a tyre model was identified using a Fibonacci tree optimisation (FTO) algorithm to improve force estimation accuracy. A schematic representation of the RL-based torque vectoring framework is shown in Fig. 2.15. The novelty of this work lay in the integration of deep RL as the torque allocator in an over-actuated EV. Rather than relying on fixed or rule-based mappings, the REDQ algorithm dynamically computed optimal wheel torques using an ensemble of Q-networks. The state space included yaw rate error, stability index from phase portrait analysis, velocity, and its deviation, while actions were continuous torques at each wheel. The reward function balanced four competing criteria, i.e., vehicle safety (tracking yaw and sideslip), motor power consumption, driver comfort (steering workload), and trajectory accuracy. The safety terms were prioritised through weighted design to ensure critical stability was preserved under learning policies. The use of REDQ improved learning stability, reduced overestimation, and enhanced sample efficiency by combining multiple Q-functions. A heuristic decay mecha-

nism transitioned exploration from rule-based to policy-based torque selection, accelerating convergence and safety during early training. Simulations confirmed that the RL agent converged after 580 episodes and generalised well across conditions. Simulations across various manoeuvres demonstrated that the proposed controller achieved lower sideslip error, reduced driver input workload, and significant energy savings (up to 11%) compared to a conventional model-based approach. The controller also minimised yaw moment and torque variation, which were critical for motor durability and passenger comfort. Although the torque allocation layer employed deep RL, the overall framework still relied on an NMPC-based high-level controller and predefined reference models, indicating that the system was not fully independent of explicit vehicle dynamics modeling.

Deng et. al.$_b$ [158] introduced a fault-tolerant control architecture for four in-wheel motor drive (4IWMD) EVs, targeting both stability and energy efficiency. The control system was structured hierarchically. The high-level consisted of an NMPC that computed the optimal longitudinal force and yaw moment. The low-level controller executed torque distribution through a quadratic optimisation problem, where the cost function dynamically balanced vehicle stability and motor power consumption. What distinguished this study was the application of a REDQ deep RL algorithm. The RL component adaptively adjusted a weighting factor in the cost function used by the low-level optimiser. This enabled the controller to handle a wide range of vehicle states and actuator fault conditions without sacrificing real-time performance. Simulation results using CarSim verified the proposed method across various driving scenarios, demonstrating reduced energy consumption and improved vehicle stability, even in the presence of actuator faults. The hierarchical control architecture of the proposed fault-tolerant scheme is shown in Fig. 2.16. The integration of deep RL, specifically the REDQ algorithm, enabled the control system to adapt in real time to changes in vehicle dynamics and fault conditions. Unlike conventional optimisation strategies with fixed weighting parameters, the RL approach dynamically adjusted the trade-off between vehicle stability and energy consumption by tuning a scalar weighting factor in the cost function. The RL agent was trained using a reward function that incorporated vehicle stability, motor efficiency, and driver comfort metrics. This formulation allowed the controller to maintain effective performance across various scenarios, including stability–energy trade-off analysis, training under random motor fault conditions, acceleration on split-friction surfaces with motor faults, and double lane change manoeuvres with motor faults. The RL output modified the optimisation cost function while preserving closed-loop system stability. Compared with fuzzy logic-based tuning methods, the RL-based approach demonstrated improved generalisation across different operating conditions and achieved lower energy consumption while maintaining or enhancing vehicle handling performance.

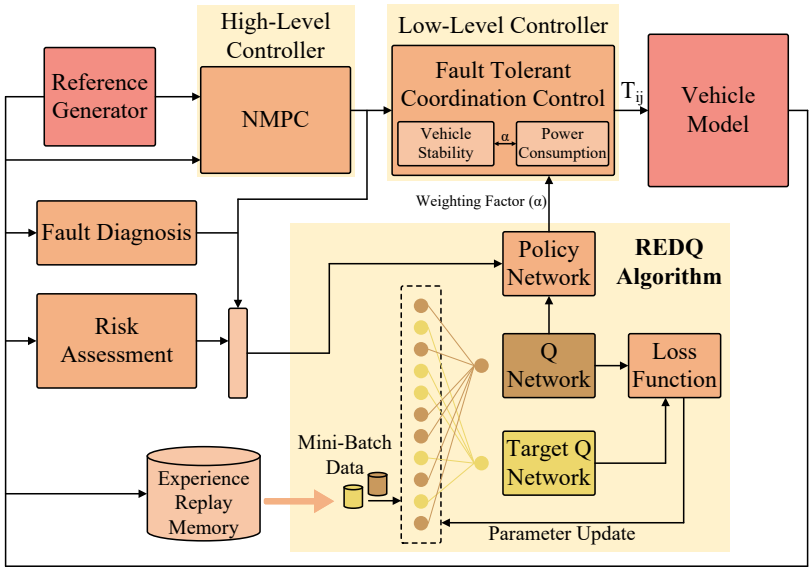


Fig. 2.16 Hierarchical control architecture of the fault-tolerant scheme by Deng et. al.$_b$ [158].

DDPG is a model-free, off-policy RL algorithm designed for continuous action spaces. It builds on the actor-critic architecture by combining value-based and policy-based methods. The critic learns an approximation of the action-value function $Q(s,a)$, while the actor learns a deterministic policy $\mu(s)$ that maps states directly to actions. Unlike stochastic policy gradient methods, DDPG uses deterministic gradients, which reduces variance and improves learning stability in control applications [159]. DDPG operates using two neural networks for each component, i.e. an actor network and a critic network, along with their corresponding target networks. The critic network is trained by minimizing the Bellman error, using target Q-values computed from the target networks. The actor is updated by applying the chain rule to the Q-value function of the critic, effectively moving the policy parameters in the direction that increases expected return. The use of target networks and experience replay helps stabilise learning by reducing the correlation between samples [160, 161].

The key update rules in DDPG are derived from the deterministic policy gradient theorem. The critic is updated to minimise the loss:

$$L = \mathbb{E}\left[(Q(s,a) - y)^2\right] \tag{2.10}$$

where the target $y = r + \gamma Q'(s', \mu'(s'))$ is computed using the target networks. The actor is updated by maximizing the estimate of the Q-value of the critic:

$$\nabla_\theta J \approx \mathbb{E}\left[\nabla_a Q(s,a) \nabla_\theta \mu(s)\right] \tag{2.11}$$

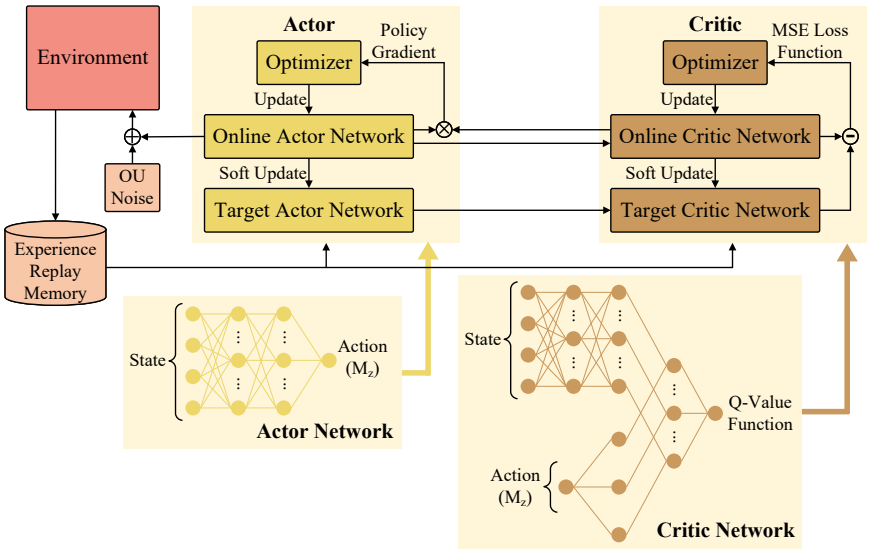


Fig. 2.17 Model-free torque vectoring strategy for distributed drive EVs using a DDPG algorithm by Wei et al.$_a$ [162].

These updates enable the actor to improve the policy based on the feedback of the critic, while the critic learns to predict returns under the current policy. This structure makes DDPG well-suited for continuous control tasks like torque vectoring, where actions such as torque split must be adjusted in a fine-grained manner. Due to its effectiveness in high-dimensional and nonlinear control systems, DDPG has been adopted in several studies focused on vehicle dynamics control. In the following examples, the recent applications of DDPG in torque vectoring are reviewed, emphasizing their system models, control objectives, and performance outcomes.

Wei et al.$_a$ [162] introduced a model-free torque vectoring strategy for distributed drive EVs using a DDPG algorithm, as shown in Fig. 2.17. The proposed system eliminated the need for nonlinear tyre models and analytically derived controllers. The torque vectoring problem was framed as an MDP, where the observed states included yaw rate, yaw rate error, and its integral, sideslip angle, and its integral, while the continuous action was the external yaw moment. The actor-critic framework enabled deterministic policy generation through neural network approximation of value and policy functions. Reference yaw rate and sideslip angle were calculated based on the linear vehicle model and were used as the supervisory signals. The low-level torque allocation followed a rule-based strategy based on vertical load transfer to distribute the demanded yaw moment to four in-wheel motors. This work demonstrated the potential of RL for torque vectoring control in EVs, particularly in handling complex and nonlinear vehicle dynamics where conventional model-based approaches may become inaccurate or computationally demanding. By employing the DDPG algorithm,

the controller learned a deterministic policy that directly mapped the vehicle state to the required corrective yaw moment without relying on explicit tyre or road friction models. The state representation included both instantaneous errors and their integral terms, enabling improved long-term stability tracking. During training, an experience replay buffer and soft target updates were used to stabilise the learning process, while exploration was guided by an Ornstein–Uhlenbeck (OU) noise process. The reward function incorporated tracking accuracy, penalties for excessive yaw moments, and terminal stability constraints, encouraging the controller to balance responsiveness with robustness. Simulation results showed that the trained DDPG agent generalised well to different steering scenarios, indicating that a single learned policy can adapt to varying and potentially unsafe driving conditions. This adaptability, together with reduced reliance on detailed system modelling, highlighted the potential of RL as a promising framework for future real-time torque vectoring control in safety-critical EV applications. Simulation experiments using CarSim-Simulink under extreme manoeuvres of double lane change and snake lane change on low-adhesion roads demonstrated that the RL-based controller significantly improved tracking accuracy and lateral stability compared to LQR and MPC baselines. The proposed RL-based approach resulted in smaller lateral deviation, reduced sideslip angle, and faster convergence to stability in phase-plane plots. In both double lane change and snake lane change cases, it maintained the yaw rate within safety bounds more effectively, improving active safety. The reward function explicitly penalised stability loss and large yaw moment deviation, guiding learning toward robust control. The proposed RL-based controller achieved a maximum lateral displacement error of 0.236 m, compared to 0.495 m for LQR and 0.402 m for MPC.

Dai et al. [163] proposed a knowledge-assisted (KA) DDPG algorithm for yaw rate and longitudinal speed tracking in skid-steering vehicles, as demonstrated in Fig. 2.18. By integrating criteria action and guiding reward mechanisms, the method accelerated learning, improved policy robustness, and reduced training time. Simulations showed improved tracking precision and faster torque adaptation compared to the baseline controller.

Taherian et al. [164] proposed an RL-based adaptive tuning strategy for torque vectoring in EVs, as shown in Fig. 2.19. The core concept was to use the DDPG algorithm to dynamically adjust the weighting matrix in a model-based torque vectoring controller. The control objective was to stabilise the vehicle under varying road friction and speed by minimizing the error between actual and desired forces at the center of gravity (COG). The desired forces were derived from a command interpreter module (CIM), and corrective longitudinal tyre forces were computed based on a quadratic objective function involving COG force tracking and control effort. Unlike other deep RL-based torque allocation methods that directly map states to wheel torques, this study applied DDPG to adaptively tune internal controller parameters.

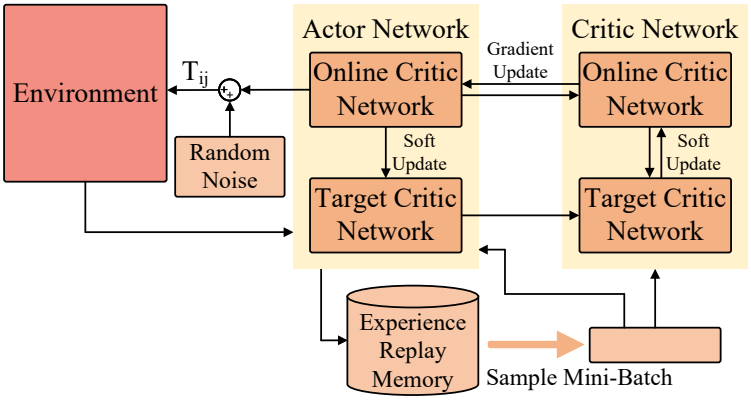


Fig. 2.18 Schematic of the driving torque distribution strategy with knowledge-assisted RL by Dai et al. [163].

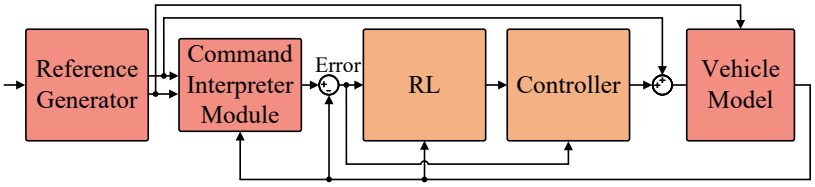


Fig. 2.19 Schematic of RL-based adaptive torque vectoring controller by Taherian et al. [164].

Simulation results verified the DDPG tuning strategy against a genetic algorithm and manual tuning under both trained and unseen driving conditions. The DDPG-tuned controller showed superior stability, smoother torque application, and lower force error. The performance index and error trends also converged more rapidly, confirming the efficacy of adaptive learning over static tuning approaches. The key innovation lay in the real-time adaptability of the learned tuning policy. During training, DDPG learned to optimise control effort dynamically to maintain stability, even in friction-speed regimes unseen during training. Despite the success, the framework only adjusted longitudinal weights, omitting lateral tyre force effects. Still, this work confirmed that actor-critic RL can reliably generalise adaptive control strategies across nonlinear domains and environmental uncertainties in torque vectoring applications.

TD3 is an actor-critic RL algorithm designed to improve the stability and performance of DDPG in continuous control tasks. TD3 addresses several key limitations of DDPG, including overestimation bias in Q-value estimates and instability due to closely coupled actor and critic updates. It introduces three main modifications of clipped double Q-learning, delayed policy updates, and target policy smoothing [165, 166]. TD3 uses two critic networks to estimate the action-value function. During training, the minimum of the two Q-value estimates is used to reduce overestimation bias, similar to the principle of Double Q-learning. This clipped Q-value is then used to compute the target for critic updates. The actor network,

which learns a deterministic policy $\mu(s)$, is updated less frequently than the critics to avoid destabilizing the learning process. This delayed update improves convergence and reduces sensitivity to noise in early training. To further stabilise training, TD3 applies target policy smoothing by adding clipped noise to the target action used in the critic target calculation. This helps regularise the Q-function by preventing sharp peaks in the value estimates. The critic loss is computed as:

$$L = \mathbb{E}\left[(Q(s,a) - y)^2\right] \tag{2.12}$$

where $\mathbb{E}[\cdot]$ is the expectation over the state–action distribution, and the target is given by:

$$y = r + \gamma \min_{i=1,2} Q_i'(s', \mu'(s') + \epsilon) \tag{2.13}$$

and $\epsilon \sim \text{clip}(\mathcal{N}(0,\sigma), -c, c)$ is the clipped noise added for policy smoothing. The actor is updated to maximise the expected Q-value estimated by one of the critics:

$$\nabla_\theta J = \mathbb{E}\left[\nabla_a Q_1(s,a) \nabla_\theta \mu(s)\right] \tag{2.14}$$

where Q_1 is typically used for policy evaluation. TD3 has been shown to outperform DDPG in several continuous control benchmarks and is well-suited for vehicle dynamics applications such as torque vectoring. Its improved stability and lower variance in value estimation make it a strong candidate for learning in high-dimensional systems where precise torque control and real-time adaptation are required [167, 168].

Wei et al.$_b$ [169] proposed a direct torque distribution strategy for distributed drive EVs using a twin delayed deep deterministic policy gradient (TD3-DDPG) reinforcement learning algorithm, as illustrated in Fig. 2.20. The authors formulated torque vectoring as an MDP that explicitly considered both safety and energy consumption through a custom reward function. Unlike traditional model-based torque vectoring approaches that compute external yaw moments, this method directly outputed motor torques from the policy network, thereby bypassing the intermediate yaw moment generation step. The proposed structure integrated actor-critic networks for continuous control, with clipped double Q-learning to reduce overestimation, delayed policy updates to improve training stability, and target policy smoothing. This study exemplified the role of deep RL in addressing nonlinearities and uncertainties in vehicle dynamics without relying on explicit modeling. The torque distribution problem was cast as a high-dimensional continuous control task, where TD3-DDPG efficiently mapped vehicle states (yaw rate, sideslip angle, velocity, etc.) to torque actions for four wheels. The RL agent was trained using a custom-designed reward that balanced lateral stability with motor energy efficiency, enabling the policy to generalise

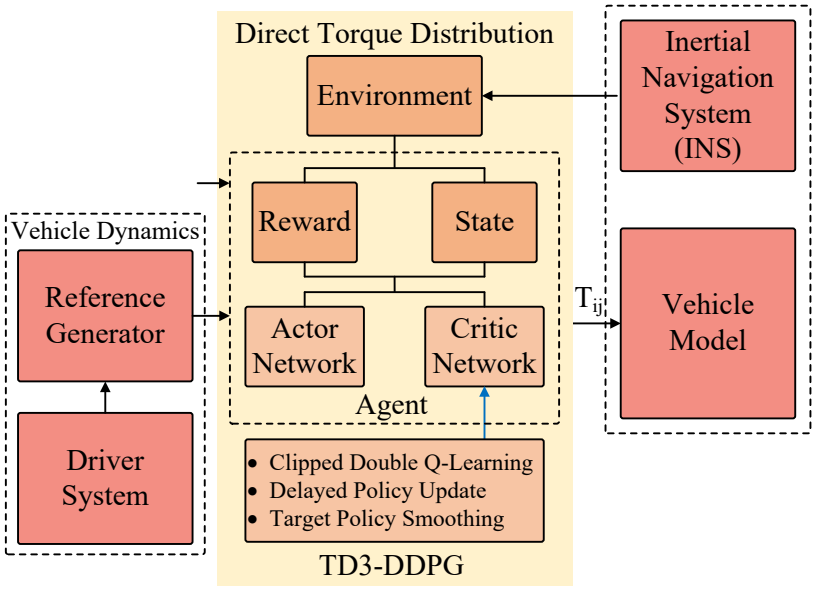


Fig. 2.20 Framework of TD3-DDPG direct torque distribution strategy by Wei et al.$_b$ [169].

across various road conditions and vehicle manoeuvres. The results demonstrated improved tracking, faster transient response, and reduced energy loss by 5.25% to 10.51%. A HIL experiment further confirmed its real-time applicability, supporting its practical viability.

In summary, model-free RL has demonstrated strong potential in addressing the challenges of torque vectoring and energy optimisation in EVs by enabling adaptive, model-free control under diverse conditions. The reviewed studies highlight the ability of RL to outperform conventional methods in both stability and efficiency metrics. A comparative overview of these RL-based torque vectoring strategies is provided in Table 2.2. In addition, Table 2.3 summarizes the reported implementation and training details of the reviewed RL-based controllers, including their configuration settings, network structures, simulation environments, and explicitly stated hyperparameters. REDQ-based methods generally prioritise learning stability and sample efficiency, making them well-suited for hierarchical control architectures where energy efficiency and fault tolerance are critical. In contrast, DDPG variants offer more direct control policies, often replacing or tuning model-based controllers, and are valued for their flexibility and fast inference in real-time applications. TD3-based approaches further improve robustness by reducing Q-value overestimation and enhancing policy stability, especially when combined with curriculum learning for improved generalization. Across the reviewed works, actor-critic frameworks have shown superior adaptability to varying road conditions and vehicle dynamics, while hierarchical and hybrid structures allow RL to complement rather than replace model-based control logic. These distinctions provide valuable guidance for selecting an appropriate RL strategy based on the

specific goals of stability, energy optimisation, learning efficiency, and fault resilience in EV control systems.

2.7 Conclusion

This chapter presents a comprehensive review of torque vectoring algorithms for vehicle dynamic control and energy optimisation in EVs using RL-based approaches. Conventional torque vectoring controllers have been surveyed for yaw stability and energy management. Model-based schemes often rely on simplified models and fixed control rules, which limit their adaptability to nonlinear dynamics and uncertain driving conditions. In contrast, model-free RL offers a data-driven alternative that allows controllers to learn directly from interaction with the environment, making it suitable for complex tasks with limited modeling fidelity. The review then focuses on some deep RL techniques used in torque vectoring problems, including DDPG, REDQ, and TD3, which are capable of handling continuous action spaces and high-dimensional inputs. A set of representative studies demonstrated how RL can be integrated into hierarchical control frameworks for torque allocation, energy efficiency, and fault tolerance. These works show that RL-based controllers can outperform traditional methods in terms of tracking performance, adaptability, and energy savings. In conclusion, model-free RL represents a promising direction for future EV control systems, especially in scenarios where classical models struggle to capture complex, nonlinear, and time-varying dynamics. Despite the progress achieved, there remains a need for an integrated and adaptive torque vectoring framework that simultaneously enhances yaw stability and energy efficiency under varying driving conditions. This gap motivates the development of the RL-based control strategy proposed in the subsequent chapters.

Table 2.2 Comparison of RL-based torque vectoring algorithms

Reference	Control Method	Control Objectives	Key Findings	Remarks
Deng et al.$_a$ [157]	NMPC + Heuristic REDQ-RL	Stability, energy efficiency, reduced driver workload	RL-based method outperformed MPC and LQR; reduced energy use, improved driver effort and stability	Integrated REDQ with hierarchical control; RL improved adaptability; added training complexity
Deng et al.$_b$ [158]	NMPC + deep RL (REDQ)	Yaw rate tracking, sideslip minimization, energy efficiency, fault tolerance	Effective hierarchical fault-tolerant structure that balanced stability and energy consumption under motor faults	REDQ deep RL dynamically managed tradeoffs; verified with CarSim; robust to actuator faults
Wei et al.$_a$ [162]	DDPG-based RL	Yaw stability, sideslip regulation, active safety under extreme conditions	Improved yaw rate tracking, reduced sideslip, and showed superior performance over MPC and LQR under critical manoeuvres	Model-free approach removed dependence on the model; verified via CarSim-Simulink simulation
Dai et al. [163]	Knowledge-Assisted DDPG (KA DDPG)	Yaw rate and longitudinal speed tracking	KA DDPG accelerated learning and improved control accuracy compared to baseline torque controller	Targeted at skid-steering platforms; promising for over-actuated EVs
Taherian et al. [164]	DDPG-tuned torque vectoring controller	Yaw rate and force tracking, stability under low tyre-road friction and varying speed	DDPG achieved lowest COG force error and best performance index compared to genetic and manual tuning	RL only tuned longitudinal weights; real-time capable but limited lateral control; simulation verified
Wei et al.$_b$ [169]	TD3-DDPG	Yaw rate and sideslip tracking; energy minimization	Achieved superior lateral stability, 5.25–10.51% energy savings, validated via HIL	No need for model; real-time capable with improved robustness

Table 2.3 Implementation and training details of RL-based torque vectoring algorithms

Reference	Training Configuration	RL Algorithm and Network Description	Simulation Environment / Real-Time Feasibility	Key Hyperparameters and Remarks
Deng et al.$_a$ [157]	800 episodes, sampling time 0.02s	RL uses an ensemble of N Q-networks; feed-forward networks; target networks updated via Polyak averaging; policy with entropy term	Inference latency not reported	Heuristic action selection probability with $\tanh(\cdot)$ decay; training platform CarSim–Simulink–Python co-simulation
Deng et al.$_b$ [158]	Fault factor k_{ij}=0.5 applied at 7s; training up to 700 episodes	Deep neural networks represent the Q-value $Q_\theta(s_t, a_t)$ and policy $\pi_\phi(s_t)$; N double deep-Q networks; layer sizes or activations not reported	Training implemented on CarSim–Simulink–Python; Inference latency or control-loop rate not reported	Discount factor $\gamma \in [0,1]$; entropy coefficient ξ; fault-tolerant weighting $\alpha_0 = 0.2$
Wei et al.$_a$ [162]	Convergence after 700 episodes	Actor–critic networks adopted; online and target networks updated by soft update factor τ =1e-5; specific layer sizes and activation functions not reported	CarSim–Simulink; training and inference executed on a 16-core Intel(R) Xeon(R) Platinum 8269CY CPU @ 2.5GHz; control-loop or latency values not reported	Discount factor $\gamma = 0.99$; experience buffer 1e6; mini-batch 64
Dai et al. [163]	Training 5000 episodes, step size 0.1	Actor and target actor networks fully connected $4\times512\times512\times4$ layers; Critic and target critic networks: $8\times512\times512\times1$ layers; activations not specified	PyCharm IDE with Python 3.7 on Intel Core i5 computer; Inference latency or control-loop rate not reported	Actor learning rate 3e-4; critic learning rate 1e-3; memory capacity 1e6; batch size 512; discount factor 0.999; soft-update rate 0.01; exploration noise 0.1; random seed = 2
Taherian et al. [164]	Training 650 episodes with 15s length; sample time 0.5	Actor–critic structure with two hidden layers of 100 neurons each; ReLU activation for hidden layers, tanh for actor output, and linear for critic output	MATLAB/Simulink Reinforcement Learning Toolbox; Inference latency or control-loop rate not reported	Discount factor 0.99; critic learning rate 1e-3; actor learning rate 1e-4; target-network update 1e-3; mini-batch size 70; variance 30; variance decay 1e-3; experience buffer 1e6
Wei et al.$_b$ [169]	Maximum episode 2000 with sample time 0.02	Actor network: fully connected layers with $10\times10\times4$ neurons (ReLU, tanh, scaling); Critic network fully connected layers with $10\times10\times10\times1$ neurons (ReLU)	Simulation platform CarSim–Simulink co-simulation; training time consumed 120min on 16-core Intel(R) Xeon(R) Platinum 8269CY CPU @ 2.5 GHz	Critic learn rate 1e-3; actor learn rate 1e-4; soft update factor 1e-5; target smooth factor 1e-3; discount factor 0.99; mini-batch size 64; experience buffer 1e6

Chapter 3

Reinforcement Learning-Based Vehicle Dynamics Control

3.1 Introduction

In the context of vehicle dynamics, RL provides a data-driven method to generate control actions that enhance yaw stability, controllability, and manoeuvrability while adapting to changing surface conditions and driving scenarios. This chapter presents the development of RL-based control frameworks for torque vectoring and yaw stability enhancement in an AWD EV platform. The proposed approach aims to establish a foundation for intelligent, model-free vehicle dynamic control capable of operating effectively in nonlinear and uncertain environments.

3.2 Reinforcement Learning Framework for Vehicle Dynamics Control

The RL framework developed in this study forms the basis for a model-free control strategy aimed at improving yaw stability and overall vehicle handling in an AWD EV. A key advantage of this design is that it removes the need for an accurate analytical model of the vehicle. Instead, the learning agent interacts directly with a simulation environment that represents the dynamic response of the vehicle. Through repeated interaction, the agent observes the consequences of its actions and gradually improves its policy to achieve the desired balance between stability, responsiveness, and energy efficiency.

Among several RL algorithms, the DDPG algorithm is chosen because it is well-suited to control problems with continuous action spaces, such as torque vectoring and yaw control.

Unlike value-based methods that operate on discrete actions, DDPG generates continuous commands directly, allowing smooth control responses without discretisation. Moreover, as an off-policy algorithm, it can learn efficiently from previously collected experience, improving training stability and data efficiency. The following section provides an explanation of the DDPG algorithm, its network structure, and its application to vehicle dynamics control within the proposed framework.

3.3 Deep Deterministic Policy Gradient (DDPG)

The DDPG algorithm is a model-free, off-policy RL approach designed to handle continuous state and action spaces. It combines the advantages of deterministic policy gradient methods with the representational power of deep neural networks. DDPG is particularly suitable for vehicle dynamics control problems such as torque vectoring, where the control actions are continuous rather than discrete.

3.3.1 Overview of the Algorithm

In the DDPG framework, the control system is formulated as a Markov Decision Process characterised by the tuple (s, a, r, p, γ), where s represents the set of states, a is the set of possible actions, r is the reward function, p is the transition probability between states, and γ is the discount factor. The objective of the learning agent is to determine a policy π that maximises the expected return G_t, defined as:

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{3.1}$$

where r_t is the immediate reward at time step t.

Unlike stochastic policy gradient methods, DDPG employs a deterministic policy $\pi(s|\theta^\mu)$ that directly maps each state s to a specific action a, as expressed by:

$$a = \pi(s|\theta^\mu) \tag{3.2}$$

where θ^μ denotes the parameters of the actor network. The goal of the algorithm is to find the optimal policy parameters that maximise the expected return.

3.3.2 Actor–Critic Structure

DDPG follows an actor–critic architecture consisting of two primary neural networks:

- Actor network: generates continuous control actions based on the observed state. It represents the policy function $\pi(s|\theta^{\mu})$.
- Critic network: evaluates the quality of the selected actions by estimating the action-value function $Q(s, a|\theta^{Q})$.

The critic network guides the actor by providing feedback on how good each action is, enabling the actor to adjust its parameters to maximise the predicted Q-value.

3.3.3 Q-Value Estimation and Policy Update

The critic network is trained to minimise the difference between the predicted Q-value and a target value based on the Bellman equation:

$$Q(s_t, a_t) = r_t + \gamma Q'(s_{t+1}, \pi'(s_{t+1}|\theta^{\mu'})|\theta^{Q'}) \tag{3.3}$$

where Q' and π' denote the target critic and target actor networks, which are delayed copies of the main networks. These target networks stabilise learning by providing slowly changing targets. The critic loss function is defined as:

$$L = \frac{1}{N}\sum_{i=1}^{N}\left(Q(s_i, a_i|\theta^{Q}) - y_i\right)^2 \tag{3.4}$$

where $y_i = r_i + \gamma Q'(s_{i+1}, \pi'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$ represents the target Q-value for each sample in a mini-batch, and N denotes the number of samples in the mini-batch used for training.

The actor network is updated using the deterministic policy gradient, which aims to maximise the estimated Q-value of the critic:

$$\nabla_{\theta^{\mu}} J \approx \frac{1}{N}\sum_{i=1}^{N} \nabla_{\theta^{\mu}} \pi(s_i|\theta^{\mu}) \nabla_a Q(s_i, a|\theta^{Q})\big|_{a=\pi(s_i|\theta^{\mu})} \tag{3.5}$$

This gradient determines how the actor parameters should be adjusted to increase the expected return predicted by the critic.

3.3.4 Exploration Strategy

Since DDPG is a deterministic algorithm, it does not inherently include stochastic exploration. To promote exploration during training, a noise term is added to the output actions of the actor network. The Ornstein–Uhlenbeck (OU) process is typically employed to generate

temporally correlated noise suitable for physical control problems:

$$\mathcal{N}_{t+1} = \mathcal{N}_t + \theta(\mu - \mathcal{N}_t)\Delta t + \sigma\sqrt{\Delta t}\,\mathcal{W}_t \tag{3.6}$$

where $\mathcal{N}_t$ is the noise at time t, μ is the mean, σ is the standard deviation, θ controls the rate of mean reversion, and $\mathcal{W}_t$ is a Wiener process. This noise helps the agent to explore different actions during training, improving convergence and preventing premature policy saturation.

3.3.5 Stabilisation Techniques

To improve training stability, DDPG adopts several mechanisms:

- Target networks: Soft updates are used for the target actor and critic networks to avoid divergence. The parameters are updated using:

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta' \tag{3.7}$$

  where τ is a small constant.

- Experience replay: Past experiences (s_t, a_t, r_t, s_{t+1}) are stored in a replay buffer and sampled randomly to break temporal correlations and improve data efficiency.

The overall DDPG learning process can be summarised in Algorithm 1. The algorithm describes the initialisation of the networks, interaction with the environment, experience replay mechanism, and parameter updates of the actor and critic networks.

3.3.6 Application to Vehicle Dynamics Control

In the context of vehicle dynamics control, the DDPG algorithm serves as the learning-based low-level controller responsible for distributing wheel torques in an all-wheel-drive electric vehicle. The actor network generates continuous torque commands for each wheel, while the critic network evaluates the quality of these actions in terms of stability and performance. By interacting with the vehicle model, the DDPG agent learns to minimise yaw rate and sideslip errors while maintaining optimal traction. This allows the control system to adapt to varying road conditions and driving scenarios without requiring an explicit vehicle model, offering superior adaptability and robustness compared with conventional model-based methods.

Algorithm 1 DDPG Training Procedure

Initialise actor network $\pi_{\theta^\mu}(s)$
Initialise critic network $Q_{\theta^Q}(s,a)$
Initialise target networks with same parameters:
    $\theta^{\mu'} \leftarrow \theta^\mu$
    $\theta^{Q'} \leftarrow \theta^Q$
Initialise replay buffer $\mathcal{R}$
for each episode $e = 1, 2, \ldots, N_{episodes}$ **do**
    Reset environment
    Observe initial state s_0
    for each time step t **do**
        Select action:
            $a_t = \pi(s_t|\theta^\mu) + \mathcal{N}_t$
        Execute action a_t
        Observe reward r_t and next state s_{t+1}
        Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{R}$
        Sample random mini-batch from $\mathcal{R}$
        Compute target Q-value:
            $y_t = r_t + \gamma Q_{\theta^{Q'}}(s_{t+1}, \pi_{\theta^{\mu'}}(s_{t+1}))$
        Update critic network by minimising loss between y_t and $Q_{\theta^Q}(s_t, a_t)$
        Update actor network using policy gradient
        Update target networks using soft update:
            $\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$
            $\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$
    end for
end for

3.3.7 Reinforcement Learning for Hierarchical Control Structure

A hierarchical control structure is adopted to separate vehicle-level yaw stability control from actuator-level torque allocation. The high-level controller generates the corrective yaw moment command M_z based on vehicle lateral dynamics, while the low-level controller distributes the individual wheel torques to realise the requested yaw moment within actuator limits. The controller interacts with a nonlinear vehicle model that captures the dynamic behaviour of the vehicle. The detailed mathematical formulation of the vehicle models, tyre model, and conventional control framework is provided in Appendix A. The parameters of a midsize vehicle used in this study are provided in Table 3.1.

To further verify the dynamic behavior and overall performance of the proposed control structure under realistic scenarios, IPG CarMaker is exploited as a high-fidelity vehicle dynamics simulation environment. The controller is implemented on a realistic EV model within IPG CarMaker to evaluate its capability. The built-in vehicle configuration for torque vectoring is selected as a representative AWD high-performance platform. This reference

Table 3.1 Vehicle parameters used in this study

Parameter	Unit	Value
Total vehicle mass	kg	1411
Yaw moment of inertia	kg·m^2	2031
Distance from COG to front axle	m	1.04
Distance from COG to rear axle	m	1.56
Track width	m	1.48
Wheel radius	m	0.30
Tyre cornering stiffness	N/rad	45 000
Wheel rotational inertia	kg·m^2	1.46

vehicle includes a torque vectoring powertrain layout and enables dynamic testing under various manoeuvres. Different views of the demo vehicle model under a test track in IPG CarMaker are depicted in Fig. 3.1. The verification is conducted in a simulation framework, where the MATLAB/Simulink-based controller is implemented to control and optimize the dynamic energy of the AWD vehicle in IPG CarMaker.

Accordingly, different RL algorithms are used at the two control levels in this work due to differences in the action space and learning requirements. At the low level, the action vector consists of four wheel torques, which requires continuous multi-dimensional control. The RL-based low-level controller is described in 3.4, and the RL-based high-level controller is presented in Section 3.5. To ensure fair evaluation, the comparisons are performed within the same control level using consistent interfaces. The low-level torque allocator is compared with a conventional torque allocation method under the same yaw moment command and actuator constraints. Similarly, the high-level controller is compared with a conventional LQR yaw moment controller, where both controllers generate M_z under identical manoeuvre conditions.

(a) (b) (c)

Fig. 3.1 Test demo vehicle and environment in IPG CarMaker: (a) Front-side view, (b) Left-side profile view, (c) First-person driver perspective.

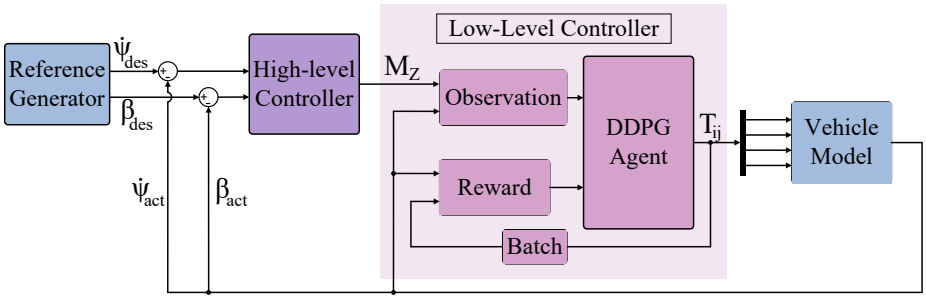


Fig. 3.2 DDPG-based torque allocation algorithm.

3.4 DDPG-Based Optimal Torque Allocation

This section presents the implementation of the proposed RL-based torque vectoring approach for an AWD EV. A hierarchical control structure is adopted, in which a high-level controller generates the corrective yaw moment and a low-level RL-based controller distributes the wheel torques accordingly. The objective of the system is to improve vehicle stability, manoeuvrability, and overall control performance under driving conditions.

3.4.1 Control Framework

The overall control structure is composed of two levels [10]. The high-level employs an LQR controller to compute the optimal corrective yaw moment, while the low-level controller, based on the DDPG algorithm, learns to allocate optimal torques among the four wheels of the vehicle. The DDPG agent interacts with the vehicle environment, learning through experience to achieve an optimal trade-off between yaw stability and control effort. A schematic of the control structure is shown in Fig. 3.2.

3.4.2 Low-Level Controller Based on DDPG

The low-level controller employs a DDPG algorithm to allocate the individual wheel torques according to the corrective yaw moment command (M_z) and the current dynamic states of the vehicle. The agent interacts with the vehicle environment, observes its response, and learns an optimal mapping between the vehicle states and the torque allocation commands. Through continuous interaction and adaptation, the policy is refined to enhance overall vehicle stability and control performance.

The state vector is designed to capture all relevant dynamic information required for decision-making. This comprehensive representation ensures that the actor network receives sufficient input information to determine effective torque distribution. The observation vector

is expressed as:

$$S_t = \{\dot{\psi}, \dot{\psi}_e, \int \dot{\psi}_e, \dot{\beta}, \dot{\beta}_e, \int \dot{\beta}_e, v_x, v_e, \int v_e, \omega_{ij}, s_{x,ij}, F_{x,total}, M_z\} \tag{3.8}$$

The action vector represents the control outputs of the agent, corresponding to the torque values applied to each of the four wheels. Four actions are known as the outputs of the DDPG agent:

$$a_t = \{T_{fl}, T_{fr}, T_{rl}, T_{rr}\} \tag{3.9}$$

The reward function is a fundamental element of the DDPG algorithm, as it quantifies the quality of each action and guides the learning process. It is defined to improve vehicle stability, minimise yaw rate and sideslip errors, and penalise excessive control effort. The reward function is expressed as:

$$r_t(s_t, a_t) = R_1 + R_2 + R_3 + R_4 + R_5 \tag{3.10}$$

where R_1 is a boolean term to guide the agent to make the yaw rate error less than a specific value:

$$R_1 = \begin{cases} -w_1 & \text{if} \quad \left|\dot{\psi}_e\right| \geq c_1 \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

where w_1 is 4, w_2 is 10, w_3 is 2, and w_4 is 1. R_2, R_3 and R_4 are defined as:

$$R_2 = -w_2 * (\dot{\psi}_e)^2 \tag{3.12}$$

$$R_3 = -w_3 * (\beta_e)^2 \tag{3.13}$$

$$R_4 = -w_4 * (v_e)^2 \tag{3.14}$$

R_5 is defined to penalise the excessive control effort taken by the agent:

$$R_5 = -\Delta u(t)^T * w_5 * \Delta u(t) \tag{3.15}$$

where $\Delta u(t)$ is $[\Delta T_{fl}, \Delta T_{fr}, \Delta T_{rl}, \Delta T_{rr}]$ and w_5 is $diag(0.1, 0.1, 0.1, 0.1)_{4*4}$

3.4.3 Simulation Results

The performance of the proposed DDPG-based torque allocation algorithm is evaluated in this section. The main objective of the simulation is to examine the capability of the DDPG agent in managing torque distribution among the four wheels of the AWD EV to ensure

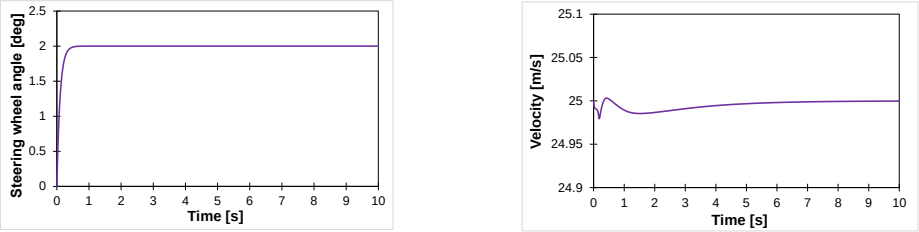
Table 3.2 DDPG Hyperparameters for DDPG-based torque allocation controller

Parameter	Value
Critic learn rate	1e-3
Actor learn rate	0.9e-4
Noise model	Ornstein Uhlenbeck (OU)
Noise standard deviation	0.6
Variance decay rate	1e-5
Discount factor	0.99
Target smooth factor	1e-3
Mini batch size	64

dynamic stability. The control system dynamically adjusts the applied torque based on the operating conditions of the vehicle to maintain stable and predictable handling. The proposed RL framework provides an effective solution for real-time control applications, as the DDPG algorithm can operate in continuous action spaces and learn optimal control policies through direct interaction with the vehicle model. The agent, trained using an actor–critic structure, continuously refines its policy to achieve the desired vehicle stability and yaw behaviour. A detailed analysis of the simulation results is presented to illustrate the performance of the DDPG-based controller in regulating vehicle dynamics.

The hyperparameter settings used for the DDPG algorithm in this study are summarised in Table 3.2. Each hyperparameter has been carefully selected through empirical tuning and supported by findings from the literature to achieve reliable and efficient learning for torque allocation and stability control in the AWD EV. The learning rate for the critic network is set to $1e-3$, defining how rapidly the critic updates its parameters in response to the error gradients. A balanced learning rate ensures accurate estimation of the Q-values while avoiding oscillations or slow convergence. The actor network uses a learning rate of $0.9e-4$, allowing for gradual and stable policy updates. A smaller learning rate for the actor helps maintain smooth learning behaviour and prevents instability in policy refinement.

The Ornstein–Uhlenbeck (OU) process is adopted as the noise model to generate temporally correlated noise, which is beneficial for exploration in continuous action spaces. The standard deviation of the noise is set to 0.6, determining the magnitude of exploration during training. A higher standard deviation promotes exploration, whereas a lower value focuses on exploiting learned policies. The variance decay rate of $1e-5$ gradually reduces the exploration noise as training progresses, ensuring extensive exploration in the early stages and greater policy stability in later iterations. The discount factor γ is set to 0.99, prioritising long-term rewards during policy evaluation. This encourages the agent to learn strategies that maximise cumulative reward rather than focusing on short-term gains. The target smoothing



(a) (b)

Fig. 3.3 Performance analysis of the low-level DDPG-based controller under circular turning manoeuvre: (a) Steering wheel angle, (b) Vehicle velocity.

coefficient is set to $1e-3$ to softly update the target networks, which enhances training stability by slowly blending the target and online network parameters. The mini-batch size of 64 specifies the number of experiences sampled from the replay buffer at each update step. Using mini-batches reduces correlation between consecutive samples and contributes to more stable learning by averaging over multiple experiences.

The performance of the proposed controller is evaluated under a circular turning manoeuvre to analyse its behaviour in response to potential oversteering or understeering. Fig. 3.3a shows the steering wheel angle input during the simulation. The steering angle increases to 2 degrees and remains constant for the remainder of the 10-second test, representing a steady circular path. The simulation is performed at a constant vehicle velocity of 25 m/s (90 km/h) on a low-friction surface with a road adhesion coefficient of 0.2. The vehicle velocity profile is shown in Fig. 3.3b. Starting slightly below 25 m/s, the velocity exhibits a brief transient oscillation before stabilising. This initial fluctuation reflects the transient response of the system to the steering input, while the rapid stabilisation demonstrates the ability of the controller to effectively manage the vehicle dynamics in real time.

To investigate the performance of the proposed DDPG-based controller, the results are compared with the conventional model-based controller. The conventional controller exploits the LQR algorithm for the high-level controller, and the torque distribution is achieved using the conventional torque allocation method. The proposed approach takes advantage of an intelligent DDPG-based controller to optimally split torque among the four wheels of the vehicle. The objective of this section is to evaluate how effectively the DDPG agent can manage torque allocation for an AWD EV, ensuring dynamic stability. The simulation results are shown in Fig. 3.4.

Fig. 3.4a illustrates the yaw rate over time for both the DDPG-based controller and the conventional controller, compared to the desired yaw rate. The desired yaw rate reaches a certain value within the first second and then stabilises. Both the DDPG and conventional controllers follow the desired yaw rate closely, with the DDPG controller demonstrating a

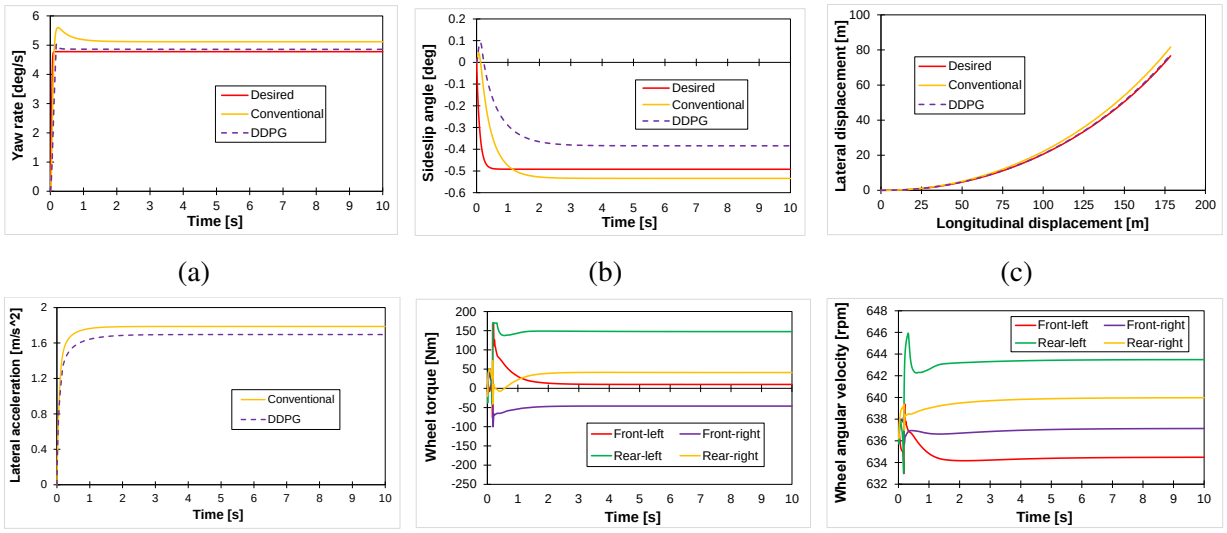


Fig. 3.4 Simulation results of the low-level DDPG-based controller under circular turning manoeuvre: (a) Yaw rate, (b) Sideslip angle, (c) Trajectory, (d) Lateral acceleration, (e) Wheel torques, (d) Wheel angular velocities.

slightly smoother response. The conventional controller shows a minor overshoot before stabilising, indicating a slightly more aggressive initial response. The DDPG controller, on the other hand, aligns with the desired yaw rate more gradually, minimising the overshoot and ensuring a smoother transition. Whereas the error between the DDPG and desired yaw rate is 0.08 deg/s after stabilisation, the same error for between the conventional and desired yaw rate is 0.34 deg/s. A comparison is accomplished in Fig. 3.4b among the desired sideslip angle, the sideslip angle from the conventional controller, and the sideslip angle from the proposed low-level DDPG-based algorithm. The conventional controller quickly reduces the sideslip angle to approximately -0.53 deg, while the sideslip angle from the DDPG algorithm has a minimum value of around -0.38 deg. This indicates that both controllers have the capability to minimise the sideslip angle.

Fig. 3.4c presents the trajectory of the AWD EV under the circular turn scenario. The desired trajectory shows a consistent lateral displacement as the vehicle moves forward. Both controllers follow the desired trajectory closely, with the DDPG controller showing a slightly better alignment with the desired path. The trajectory of the conventional controller diverges more at higher longitudinal displacements, indicating that the DDPG controller is more effective in maintaining the desired path, especially over longer distances. The lateral acceleration obtained from the conventional and DDPG controllers is compared in Fig. 3.4d over time. Fig. 3.4e shows the torque distribution among the four wheels over time based

on the proposed intelligent torque allocation method. The DDPG controller distributes the torque across the wheels to guarantee the stability of the vehicle. This helps in maintaining better manoeuvrability and stability, and enhancing the overall driving experience. Also, the angular velocity of the wheels is demonstrated in Fig. 3.4f for the proposed DDPG-based algorithm, while the controller manages to ensure the dynamic stability of the vehicle via an optimal torque allocation approach.

The DDPG controller demonstrates smoother responses in yaw rate, side-slip angle, and lateral acceleration, along with balanced torque distribution and wheel angular velocity. The DDPG-based method possesses the ability to continuously learn and adapt to new driving conditions, and improve its performance over time. Furthermore, handling the nonlinearities and real-time optimisation are guaranteed. The need for detailed system modelling is also reduced as the DDPG relies on interaction with the environment. These improvements contribute to enhanced vehicle stability and decreased oversteer and understeer.

3.5 Twin Delayed Deep Deterministic Policy Gradient (TD3)

The TD3 algorithm is an improved variant of the DDPG algorithm designed to address the issues of overestimation bias and training instability often observed in deterministic actor–critic methods. TD3 enhances the learning process through three main mechanisms: clipped double Q-learning, delayed policy updates, and target policy smoothing. These improvements make TD3 more stable and sample efficient, particularly in continuous control problems such as vehicle torque vectoring and yaw stability control.

3.5.1 Overview of the Algorithm

The TD3 algorithm builds upon the same actor–critic framework as DDPG, in which the environment is modelled as a Markov Decision Process defined by (S, A, R, P, γ). The agent aims to learn a deterministic policy $\pi(s|\theta^{\mu})$ that maximises the expected return:

$$J = \mathbb{E}_{s_t \sim P}\left[Q(s_t, \pi(s_t|\theta^{\mu}))\right] \tag{3.16}$$

where $Q(s, a|\theta^{Q})$ is the action–value function parameterised by θ^{Q}.

The key modification in TD3 is the introduction of two critic networks, Q_1 and Q_2, which independently estimate the expected return for the same state–action pair. By taking the minimum of these two estimates during target computation, TD3 effectively reduces overoptimistic value predictions and improves learning stability.

3.5.2 Actor–Critic Structure

Similar to DDPG, TD3 employs an actor–critic architecture consisting of:

- Actor network: represents the deterministic policy $\pi(s|\theta^{\mu})$ that outputs continuous control actions.
- Twin critic networks: $Q_1(s,a|\theta^{Q_1})$ and $Q_2(s,a|\theta^{Q_2})$, which evaluate the quality of the actor's chosen actions.

The twin critics provide a more reliable estimate of the expected return by reducing value overestimation through the clipped double Q-learning mechanism. The actor network is trained to maximise the critic's evaluation of its actions, while the critics are trained to minimise the prediction error against the target Q-value.

3.5.3 Clipped Double Q-Learning

The target Q-value in TD3 is computed using both critic networks as:

$$y = r_t + \gamma \min_{i=1,2} Q_i'(s_{t+1}, \pi'(s_{t+1}|\theta^{\mu'})|\theta^{Q_i'}) \tag{3.17}$$

where Q_i' and π' represent the target critic and target actor networks, respectively. By using the minimum of the two predicted values, TD3 mitigates overestimation bias, which can otherwise lead to unstable or divergent learning behaviour. Each critic network is trained by minimising its respective loss function:

$$L_i = \frac{1}{N}\sum_{j=1}^{N}\left(Q_i(s_j, a_j|\theta^{Q_i}) - y_j\right)^2 \tag{3.18}$$

3.5.4 Delayed Policy Update

TD3 introduces a delay in the policy and target network updates relative to the critic updates. For every policy update, the critic networks are updated multiple times. This approach allows the critics to provide more accurate and stable Q-value estimates before the actor is updated. To reduce overestimation bias and ensure consistency with clipped double Q-learning, the minimum of the two critic networks is used when computing the policy gradient. The actor parameters are therefore updated according to:

$$\nabla_{\theta^{\mu}} J \approx \frac{1}{N}\sum_{j=1}^{N} \nabla_a \left(\min_{i=1,2} Q_i(s,a|\theta^{Q_i})\right)\Big|_{s=s_j, a=\pi(s_j)} \nabla_{\theta^{\mu}} \pi(s|\theta^{\mu})\Big|_{s_j} \tag{3.19}$$

3.5.5 Target Policy Smoothing

To further improve robustness, TD3 adds a small, clipped noise term to the target policy during critic target computation:

$$a' = \pi'(s_{t+1}|\theta^{\mu'}) + \epsilon, \quad \epsilon \sim \text{clip}(\mathcal{N}(0, \sigma), -c, c) \tag{3.20}$$

This smoothing operation prevents the policy from exploiting sharp Q-value peaks, resulting in smoother and more consistent learning behaviour. The noise ϵ is typically drawn from a Gaussian distribution with small variance and is clipped within a fixed range to ensure bounded perturbations.

3.5.6 Exploration Strategy

TD3 adopts a similar exploration strategy to DDPG by injecting temporally correlated noise into the action space during training. The Ornstein–Uhlenbeck (OU) process or a Gaussian noise model is typically used to ensure exploration around the deterministic policy output:

$$a_t = \pi(s_t|\theta^{\mu}) + \mathcal{N}_t \tag{3.21}$$

where $\mathcal{N}_t$ denotes the exploration noise. This mechanism ensures sufficient policy exploration while maintaining control smoothness.

The overall TD3 learning process is summarised in Algorithm 2. The algorithm describes the initialisation of the actor and twin critic networks, experience replay, delayed policy updates, and soft target updates.

3.5.7 Application to Vehicle Dynamics Control

For vehicle dynamics control, the TD3 algorithm serves as the learning agent that generates continuous torque commands for each wheel of the AWD EV. The actor network outputs action(s) corresponding to the front-left, front-right, rear-left, and rear-right wheels. The twin critic networks evaluate the stability and performance associated with each action, providing reliable feedback to guide the updates of the actor.

The use of TD3 provides enhanced stability during training compared with DDPG due to the reduced value overestimation and delayed actor updates. This enables the controller to converge towards a more accurate and robust torque allocation policy. As a result, the vehicle maintains improved yaw stability and sideslip regulation under nonlinear and low-adhesion conditions. The enhanced sample efficiency of the algorithm also contributes to faster

Algorithm 2 TD3 Training Procedure

Initialise actor network $\pi_{\theta^\mu}(s)$
Initialise twin critic networks $Q_{\theta_1}(s,a)$ and $Q_{\theta_2}(s,a)$
Initialise target networks with same parameters:
    $\theta'_1 \leftarrow \theta_1$
    $\theta'_2 \leftarrow \theta_2$
    $\theta^{\mu'} \leftarrow \theta^\mu$
Initialise replay buffer $\mathcal{R}$
for each episode $e = 1, 2, \ldots, N_{episodes}$ **do**
    Reset environment
    Observe initial state s_0
    for each time step t **do**
        Select action:
            $a_t = \pi(s_t|\theta^\mu) + \mathcal{N}_t$
        Execute action a_t
        Observe reward r_t and next state s_{t+1}
        Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{R}$
        Sample random mini-batch from $\mathcal{R}$
        Compute target Q-value using clipped double Q-learning:
            $y_t = r_t + \gamma \min_{i=1,2} Q_{\theta'_i}(s_{t+1}, \pi_{\theta^{\mu'}}(s_{t+1}))$
        Update both critic networks by minimising loss
        if time step t mod $d = 0$ **then**
            Update actor network using policy gradient
            Update target networks using soft update:
                $\theta'_i \leftarrow \tau\theta_i + (1-\tau)\theta'_i$ for $i = 1, 2$
                $\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$
        end if
    end for
end for

convergence, making it suitable for real-time control applications when integrated with high-fidelity simulation environments.

3.6 Curriculum Learning for Adaptive Torque Vectoring

The high-level controller within the proposed framework is developed based on a TD3 algorithm, incorporating curriculum learning to progressively introduce task complexity during the training process. Curriculum learning is a training strategy in which a learning agent is exposed to training tasks in a progressive order, starting from simpler scenarios and gradually increasing the level of difficulty. The original concept of curriculum learning was first introduced in [170], inspired by the way humans acquire complex skills through structured learning stages. In RL, this approach enables the model to first capture fundamental

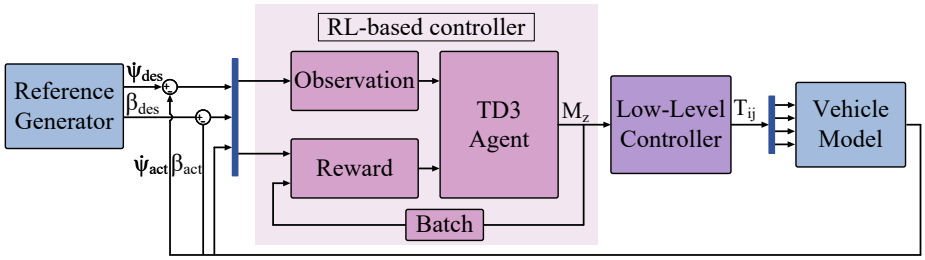


Fig. 3.5 A schematic of the proposed TD3-based dynamic controller.

patterns before encountering more complex examples, which can improve convergence speed and lead to better generalisation performance. By gradually expanding the difficulty of the training distribution, curriculum learning can guide the optimisation process towards more stable and effective solutions. Curriculum learning is exploited in this research to progressively train the TD3-based controller across tasks with increasing complexity in steering input, vehicle speed, and road friction. The proposed TD3-based controller in this work is implemented within a hierarchical control framework, as illustrated in Fig. 3.5.

3.6.1 TD3 Formulation with Curriculum Learning

To improve training efficiency and enhance policy convergence, a curriculum learning strategy is incorporated into the TD3 framework. The training process is divided into a sequence of progressively challenging tasks, defined as $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3\}$, where each task modifies specific parameters of the training environment. The curriculum is implemented over three consecutive stages, with transitions between tasks occurring after a predefined number of training episodes:

$$\mathcal{T}(e) = \begin{cases} \mathcal{T}_1, & 0 \le e < E_1 \\ \mathcal{T}_2, & E_1 \le e < E_2 \\ \mathcal{T}_3, & E_2 \le e < E_3 \end{cases} \tag{3.22}$$

where $\mathcal{T}(e)$ represents the task assigned to episode e, and E_1, E_2, and E_3 denote the transition points between successive stages.

The curriculum structure enables a gradual increase in task complexity, allowing the agent to first master basic control behaviours before progressing to more complex dynamic scenarios. Fixed transition points are adopted to balance computational efficiency and training stability. This non-automated scheduling approach provides greater interpretability and avoids abrupt environmental changes that could disrupt convergence. Manual adjustment of hyperparameters in each stage ensures precise control over the learning dynamics, which

is particularly beneficial in torque vectoring problems where system responses are highly sensitive to control inputs. During each task $\mathcal{T}_i$, both the actor and critic networks are updated using task-specific reward functions and hyperparameters. The TD3 target update for each task is formulated as:

$$y_{\mathcal{T}} = r_{\mathcal{T}} + \gamma \cdot \min\left(Q_1'(s', \pi'(s', \Lambda_{\mathcal{T}}) + \epsilon_{\mathcal{T}}),\ Q_2'(s', \pi'(s', \Lambda_{\mathcal{T}}) + \epsilon_{\mathcal{T}})\right) \tag{3.23}$$

where $r_{\mathcal{T}}$ denotes the task-specific reward, $\Lambda_{\mathcal{T}}$ represents the hyperparameters associated with task $\mathcal{T}$, and $\epsilon_{\mathcal{T}}$ is the exploration noise applied during target updates.

Each task modifies the reward structure, environment dynamics, and network hyperparameters, enabling the policy to evolve progressively across the curriculum:

$$\pi_1 = \arg\max_a Q_1^{\mathcal{T}_1}(s, a; \Lambda_1) \tag{3.24}$$

$$\pi_2 = \arg\max_a Q_1^{\mathcal{T}_2}(s, a; \Lambda_2) \tag{3.25}$$

$$\pi_3 = \arg\max_a Q_1^{\mathcal{T}_3}(s, a; \Lambda_3) \tag{3.26}$$

This formulation preserves learning continuity while allowing task-specific adaptation. The TD3 loss function for each stage is defined as:

$$\mathcal{L}_{TD3}^{\mathcal{T}} = \frac{1}{N} \sum (y_{\mathcal{T}} - Q_{\mathcal{T}}(s, a; \Lambda_{\mathcal{T}}))^2 \tag{3.27}$$

and the transition between tasks follows:

$$\mathcal{L}_{TD3}^{\mathcal{T}+1} = \mathcal{L}_{TD3}^{\mathcal{T}} + \Delta_{\mathcal{T}} \tag{3.28}$$

where $\Delta_{\mathcal{T}}$ represents the task-dependent adaptation required for transitioning to the next stage. The final policy is obtained as:

$$\pi^* = \lim_{e \to E_{\mathcal{T}}} \pi_e \tag{3.29}$$

where $E_{\mathcal{T}}$ represents the episode threshold for the task. The optimal policy is reached at the end of training:

$$\pi^* = \pi_3 \tag{3.30}$$

This curriculum-enhanced TD3 framework enables structured and stable policy development through progressive learning stages. By refining the policy incrementally under task-specific conditions, the agent achieves improved convergence, robustness, and generalisation for complex vehicle dynamics and torque vectoring control tasks. The approach utilized to accomplish this is outlined in Algorithm 3.

Algorithm 3 TD3-Based Adaptive Yaw Stability Control with Curriculum Learning

Initialise vehicle dynamics model in MATLAB/Simulink
Initialise state space S, action space A, and hyperparameters
Initialise TD3 agent:
    - Two critic networks $Q_{\theta_1}(s,a)$ and $Q_{\theta_2}(s,a)$
    - Actor network $\pi_\phi(s)$
    - Target networks $Q_{\theta'_1}(s,a)$, $Q_{\theta'_2}(s,a)$, $\pi_{\phi'}(s)$
    - Experience replay buffer $\mathcal{D}$
Define Curriculum Learning Strategy:
    Task 1: Train agent with varying steering inputs (fixed velocity, fixed road friction)
    Task 2: Introduce varying velocity conditions
    Task 3: Introduce varying road friction μ
for each task $\mathcal{T}_k$ in {Task 1, Task 2, Task 3} **do**
    for each episode $e = 1, 2, \ldots, N_{episodes}$ **do**
        Reset environment with task-specific variations
        Observe initial state s_0
        for each time step t **do**
            Select action using actor policy:
                $a_t = \pi_\phi(s_t) + \text{OU noise}$
            Apply action a_t and observe $(s_{t+1}, r_t, done)$
            Store experience $(s_t, a_t, r_t, s_{t+1}, done)$ in $\mathcal{D}$
            Sample mini-batch $\mathcal{B}$ from $\mathcal{D}$
            Compute target Q-value:
                $\hat{y} = r + \gamma \min_{i=1,2} Q_{\theta'_i}(s', \pi_{\phi'}(s') + \text{clipped noise})$
            Update critics:
                $\theta_i \leftarrow \theta_i - \alpha_c \nabla_{\theta_i} J(\theta_i)$
            if update step is due **then**
                Compute policy gradient $\nabla_\phi J(\phi)$ and update actor:
                    $\phi \leftarrow \phi - \alpha_a \nabla_\phi J(\phi)$
                Update target networks:
                    $\theta'_i \leftarrow \tau\theta_i + (1-\tau)\theta'_i$
                    $\phi' \leftarrow \tau\phi + (1-\tau)\phi'$
            end if
        end for
    end for
    Evaluate performance and transition to next task
end for
Deploy trained TD3 agent in different scenarios and evaluate performance

3.6.2 Curriculum Learning Framework for Progressive Training of the TD3-Based Controller

In this work, curriculum learning is incorporated into the TD3 training process through three sequential tasks, each designed to address a specific aspect of vehicle dynamics. This progressive approach improves learning stability, accelerates convergence, and enhances the ability of the controller to generalise across varying operating conditions.

- Task 1: The first stage focuses on training the agent to regulate its output actions in response to different steering inputs, where the steering wheel angle is randomly varied.
- Task 2: The second stage introduces variations in vehicle speed between 10 and 25 m/s to expose the agent to a range of longitudinal motion conditions, ensuring that the learned torque vectoring policy remains effective across different velocities.
- Task 3: The final stage randomises the road friction coefficient (μ) between 0.3 and 1.0 to simulate various driving surfaces such as dry asphalt, wet roads, and snowy conditions, enabling the agent to adapt its control policy to changing tyre–road interactions.

The selected parameter ranges for steering angle, velocity, and friction coefficient reflect realistic operating limits of road vehicles. The chosen friction range of 0.3 to 1.0 in Task 3 ensures consistency with the practical operating conditions of AWD EVs. Combining friction values below 0.3 with higher speeds (up to 25 m/s) represents an extreme condition where conventional torque vectoring methods are no longer effective. The lower bound of 0.3 therefore captures demanding yet feasible conditions, such as snow-covered surfaces, while preserving the controllability and relevance of torque-based yaw stability control. This configuration allows the agent to learn optimal torque allocation policies within a realistic and physically valid domain.

By gradually increasing task complexity, curriculum learning improves the efficiency of the TD3-based training process. The hyperparameter settings are summarised in Table 3.3. Each parameter is tuned to achieve stable and effective performance across all curriculum stages. Tasks 1 and 2 each include 200 training episodes, whereas Task 3 is extended to 400 episodes to provide sufficient training time for the agent to adapt to more complex environmental dynamics. The experience buffer size is set to $1e6$, ensuring a diverse set of transitions for stable learning. A mini-batch size of 64 is used to provide a balance between computational efficiency and gradient stability, allowing the agent to generalise effectively from sampled experiences. The gradient thresholds for both actor and critic networks are set to 1 to prevent instability caused by excessively large gradient updates, which can occur in early training stages.

Table 3.3 Hyperparameter settings for the proposed curriculum learning algorithm

Parameter	Task 1	Task 2	Task 3
General settings			
Experience buffer length		1e6	
Mini-batch size		64	
Actor gradient threshold		1	
Critic gradient threshold		1	
Target policy standard deviation		$\sqrt{0.3}$	
Discount factor		0.995	
Noise model		Ornstein–Uhlenbeck (OU)	
Task-specific settings			
Number of training episodes	200	200	400
Actor learning rate	1e-4	1e-5	1e-5
Critic learning rate	1e-3	1e-4	1e-4
Noise mean attraction constant	0.15	0.15	0.20
Noise standard deviation	46	40	32

In the TD3 algorithm, the target policy standard deviation determines the amount of noise added to the target actions during critic target computation. This mechanism, referred to as target policy smoothing, mitigates overestimation of Q-values and enhances robustness. The target policy standard deviation is set to $\sqrt{0.3}$ in this work to prevent sharp spikes in estimated values. The discount factor is fixed at 0.995 to ensure that both short-term and long-term rewards are appropriately balanced in the optimisation process. The actor and critic learning rates are adjusted across the curriculum stages. In the initial stage, higher learning rates are applied to enable faster convergence and encourage broader exploration. As training progresses, these rates are gradually reduced to ensure smoother updates and improve stability in the later stages. Excessive learning rates can lead to unstable policy updates, whereas overly small values may slow down learning. The adopted configuration strikes a balance between adaptability and convergence consistency across all training phases.

Exploration during training follows the Ornstein–Uhlenbeck (OU) process to generate temporally correlated noise suitable for continuous control problems such as torque vectoring. The noise standard deviation is set to 46 in Task 1 to promote broad exploration and reduced progressively to 40 and 32 in Tasks 2 and 3, respectively. This gradual reduction enables the agent to transition from exploration to fine-tuning as learning progresses. The noise mean attraction constant in the OU process, which governs how rapidly the noise reverts to its mean, is maintained at 0.15 in the first two tasks and increased to 0.2 in Task 3 to improve adaptability during the most challenging phase.

The criteria for transitioning between tasks and the associated hyperparameters are manually defined for each stage of training. Higher exploration noise and learning rates are used in early stages to promote faster policy discovery, whereas reduced values in later tasks encourage convergence and refinement. This structured, progressively tuned learning framework enhances the robustness and stability of the TD3-based controller, enabling it to achieve optimal torque vectoring and improved dynamic stability across a wide range of driving conditions in AWD EVs.

A sensitivity analysis is conducted on key hyperparameters to determine appropriate parameter settings for the proposed controller. As an illustrative example, the actor learning rate α_π and critic learning rate α_Q are analysed to evaluate their influence on controller performance. The analysis is performed for the TD3 agent under a circular turning scenario at a velocity of 15 m/s and a tyre–road friction coefficient of 0.3. Since the actor–critic structure and training procedure are consistent across all agents, the results can be generalised to other RL-based controllers in this study. Table 3.4 summarises the outcomes across performance indicators, including maximum of sideslip angle ($max|\beta|$), maximum of sideslip angle rate ($max|\dot{\beta}|$), integral of sideslip angle ($\int|\beta|$), integral of sideslip angle rate ($\int|\dot{\beta}|$), integral of yaw rate error ($\int|\dot{\psi_e}|$), and maximum lateral displacement from the desired path ($max|Y_e|$). The baseline configuration is set to $\alpha_\pi = 1e-5$ and $\alpha_Q = 1e-4$.

To evaluate the relative influence of the learning rates on the controller performance, a normalised Sensitivity Weight Index (SWI) is defined, while keeping all other hyperparameters unchanged:

$$SWI_{\alpha_i} = \frac{(\Delta C_{SWI}/C_{SWI,nom})}{(\Delta\alpha_i/\alpha_{i,nom})} \times 100 \tag{3.31}$$

where $C_{SWI,nom}$ is the nominal value of the performance criterion obtained using the baseline learning rates, ΔC_{SWI} represents the variation in the criterion resulting from a change in the learning rate, $\alpha_{i,nom}$ is the nominal learning rate, and $\Delta\alpha_i$ denotes its variation. A negative SWI indicates that changing the learning rate degrades performance, while a positive SWI shows improvement in Table 3.4.

The results indicate that increasing the actor learning rate to $1e{-}4$ or $1e{-}3$ generally leads to degraded stability and tracking performance. This behaviour is reflected by the increase in several stability-related metrics. Although the degradation is moderate, the negative SWI values observed for most metrics confirm that higher actor learning rates reduce control performance. A similar trend is observed when the critic learning rate is decreased to $1e{-}5$, which results in the largest SWI magnitudes across several metrics, particularly $\int|\dot{\beta}|$ and $max|Y_e|$. Increasing the critic learning rate to $1e{-}3$ produces smaller variations in the performance indicators, suggesting a relatively moderate influence compared with the

Table 3.4 Sensitivity analysis of learning rates on stability performance metrics

Actor learning rate (α_π)	Critic learning rate (α_Q)		$max\|\beta\|$ [deg]	$max\|\dot{\beta}\|$ [deg/s]	$\int\|\beta\|$ [deg·s]	$\int\|\dot{\beta}\|$ [deg]	$\int\|\dot{\psi}_e\|$ [deg/s]	$max\|Y_e\|$ [m]
1e-5	1e-4	Value	0.458	2.895	3.504	1.272	5.838	1.022
		SWI	–	–	–	–	–	–
1e-4	1e-4	Value	0.581	2.907	3.747	1.778	5.360	1.345
		SWI	-2.983	-0.046	-0.770	-4.419	+0.909	-3.511
1e-3	1e-4	Value	0.563	3.068	4.808	2.062	7.323	1.662
		SWI	-0.231	-0.060	-0.375	-0.627	-0.256	-0.632
1e-4	**1e-5**	Value	0.541	3.075	4.170	1.712	6.010	1.498
		SWI	-20.135	-6.908	-21.118	-38.434	-3.273	-51.750
1e-4	**1e-3**	Value	0.489	2.926	3.314	1.281	5.902	1.073
		SWI	-0.752	-0.118	+0.602	-0.078	-0.121	-0.554

other cases. Overall, the results demonstrate that the baseline configuration ($\alpha_\pi = 1e{-}5$ and $\alpha_Q = 1e{-}4$) provides the most balanced trade-off between stability performance and training robustness.

3.7 TD3-Based Direct Yaw Control

The TD3 algorithm provides a model-free control approach in which an intelligent agent learns an optimal control policy through interaction with the environment [11]. The TD3 architecture utilises separate neural networks for the actor and the two critic components to mitigate overestimation bias and enhance learning robustness. The network architectures for both components are illustrated in Fig. 3.6. The actor network receives the state vector as input and processes it through a sequence of fully connected (FC) layers comprising 128, 64, 32, and 32 neurons. Each layer is followed by a ReLU activation function to introduce nonlinearity and support efficient gradient propagation. The output layer of the actor employs a hyperbolic tangent (Tanh) activation function followed by a scaling operation to constrain the action outputs within the physically allowable torque limits.

Each of the critic networks take both the state and action vectors as inputs. These inputs are first processed independently through fully connected layers and then concatenated before passing through subsequent layers for feature fusion. The final output of each critic network provides a scalar Q-value representing the estimated return of the given state–action

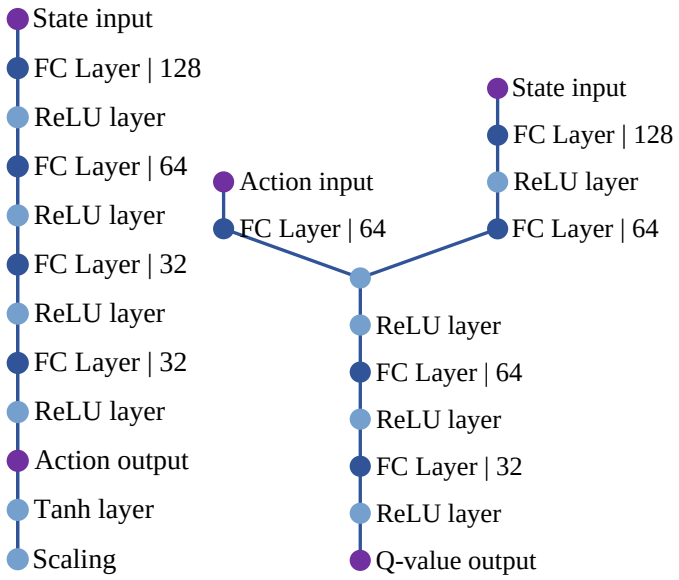


Fig. 3.6 Architecture of neural networks of: (a) Actor, and (b) Critics.

pair. During training, the TD3 algorithm updates the critic networks to accurately estimate the expected return, while the actor network is optimised to select actions that maximise this Q-value. This dual-critic structure, combined with delayed policy updates and target smoothing, enables the TD3 controller to achieve stable learning and reliable convergence for direct yaw control in all-wheel-drive electric vehicles. The following subsections detail the implementation of TD3 for vehicle dynamic control and the training process.

3.7.1 Observation and Action Space Definition

The observation (state) space provides the TD3 controller with real-time information about the dynamic behaviour of the vehicle, enabling it to evaluate current performance relative to the desired reference states. These observation signals capture the essential vehicle dynamics required for decision-making and control adaptation. The complete set of observation signals is defined as:

$$S_t = \{\dot{\psi}_e, \int \dot{\psi}_e, \beta, \beta_e, v_x, v_e, \omega_{ij}, \lambda_{ij}\} \tag{3.32}$$

where $\dot{\psi}_e$ and β_e represent the yaw rate and sideslip angle errors, v_x and v_e denote the longitudinal velocity and its error, ω_{ij} is the wheel angular velocity, and λ_{ij} is the slip ratio of each wheel. This formulation ensures that the TD3 agent has access to both vehicle-level and wheel-level dynamic states for accurate torque vectoring and stability control.

The action space is defined by the corrective yaw moment (M_z), which serves as the control command generated by the TD3 agent. The objective is to maintain stability and enhance manoeuvrability based on the observed vehicle dynamics:

$$a_t = \{M_z\} \tag{3.33}$$

The agent learns to determine the optimal corrective yaw moment M_z that maximises the cumulative reward, thereby achieving balanced and stable vehicle behaviour under varying driving and road conditions.

3.7.2 Reward Function Design for Adaptive Control

The reward function is a key component of the TD3 training process, as it defines the learning objectives and guides the agent toward achieving stable and efficient control behaviour. In this study, the reward function is designed to improve the yaw stability and overall dynamic performance of an AWD EV through adaptive control. The total reward at each time step $r_t(s_t, a_t)$ is composed of five components, each targeting a specific control objective:

$$r_t(s_t, a_t) = R_1 + R_2 + R_3 + R_4 + R_5 \tag{3.34}$$

The individual reward terms are defined as:

$$R_1 = -w_1(\dot{\psi}_e)^2 \tag{3.35}$$

$$R_2 = \begin{cases} -w_2, & \text{if } |\dot{\psi}_e| > |c_2| \\ 0, & \text{otherwise} \end{cases} \tag{3.36}$$

$$R_3 = \begin{cases} -w_3, & \text{if } |\beta| > |c_3| \\ 0, & \text{otherwise} \end{cases} \tag{3.37}$$

$$R_4 = \begin{cases} -w_4, & \text{if simulation terminates} \\ 0, & \text{otherwise} \end{cases} \tag{3.38}$$

$$R_5 = -\Delta u(t)^T w_5 \Delta u(t) \tag{3.39}$$

The first term (R_1) penalises deviations between the actual and desired yaw rate, encouraging the agent to maintain the desired vehicle orientation. The second and third terms (R_2 and R_3) constrain the yaw rate error and sideslip angle within safe limits, preventing

excessive yaw motion or lateral instability. The threshold values c_2 and c_3 are selected based on the driving scenario to ensure realistic stability margins. For example, during training under a single lane-change manoeuvre with a longitudinal velocity of 15 m/s and a tyre–road friction coefficient of 0.3, c_2 and c_3 are set to 0.0009 and 0.1, respectively. Overly restrictive thresholds could limit manoeuvrability, whereas excessively loose limits may lead to instability.

The fourth component (R_4) introduces a penalty when the simulation terminates prematurely, which occurs if the vehicle velocity becomes negative ($v_x < 0$) or if the yaw rate error exceeds 0.1 rad/s. This discourages unsafe or unstable behaviours during training and promotes robust control policies. The final term (R_5) penalises abrupt changes in the control input $\Delta u(t)$ to maintain smooth torque transitions and improve driving comfort. Here, $\Delta u(t) = [\Delta T_{fl}, \Delta T_{fr}, \Delta T_{rl}, \Delta T_{rr}]$, with $\Delta T_{ij} = T_{ij}(k) - T_{ij}(k-1)$ representing the torque increment at each wheel.

The weighting coefficients w_1–w_5 determine the relative importance of each objective and are selected as 50, 10, 5, 50, and 0.01, respectively. These values are tuned empirically to ensure balanced learning between yaw stability, smooth control effort, and overall manoeuvrability.

To analyse the contribution of each reward component and justify the selected weights, an ablation study is performed by removing one term at a time (w_1 to w_4). The performance of the trained agent is then evaluated using six criteria, summarised in Table 3.5, where the worst-performing metrics are bolded. The results show that removing any reward component leads to a noticeable degradation in controller performance compared with the unablated configuration. In particular, removing the yaw rate tracking term w_1 produces the largest deterioration across all evaluation metrics, indicating that accurate yaw rate regulation plays a critical role in maintaining vehicle stability. Eliminating the constraint terms w_2 and w_3 also increases the sideslip angle and yaw rate error, demonstrating the importance of enforcing stability limits during training. In contrast, removing the termination penalty w_4 results in comparatively smaller performance degradation, suggesting that this term mainly contributes to improving training robustness. Overall, the ablation results confirm that each reward component contributes to the stability and tracking performance of the controller, thereby justifying the selected reward structure and weight configuration.

3.7.3 Learning Performance

The learning performance of the proposed TD3-based reinforcement learning agent is evaluated under two training configurations of without curriculum learning and with the integration of curriculum learning. The preogression of episode rewards over training episodes

Table 3.5 Ablation study of reward function based on different evaluation criteria

Ablation term	$max\|\beta\|$ [deg]	$max\|\dot{\beta}\|$ [deg/s]	$\int \|\beta\|$ [deg·s]	$\int \|\dot{\beta}\|$ [deg]	$\int \|\dot{\psi}_e\|$ [deg]	$max\|Y_e\|$ [m]
Unablated	0.458	2.895	3.504	1.272	5.838	1.022
w_1	**0.731**	**3.684**	**5.142**	**2.061**	**8.944**	**1.836**
w_2	0.612	3.487	4.328	1.745	7.214	1.402
w_3	0.684	3.214	4.917	1.631	6.988	1.523
w_4	0.541	3.046	3.892	1.442	6.215	1.188

is shown in Fig. 3.7 and Fig. 3.8, respectively. These figures illustrate the convergence behaviour, reward progression, and learning stability of the TD3 agent under both conditions.

As shown in Fig. 3.7, the TD3 agent trained without curriculum learning exhibits significant fluctuations in episode rewards during the early training stages, which corresponds to the exploration phase typical in RL. Although the episode rewards gradually improve and stabilise, the learning curve remains relatively noisy, suggesting limited stability during training. The final average reward indicates that the agent successfully learns a usable policy; nevertheless, convergence is slower and less consistent.

In contrast, Fig. 3.8 presents the results of the TD3 agent trained with curriculum learning, consisting of three progressively challenging tasks, i.e., task 1 (Fig. 3.8a), task 2 (Fig. 3.8b), and task 3 (Fig. 3.8c). In task 1, the episode rewards initially fluctuate due to exploration but gradually stabilise as the agent learns to regulate torque distribution effectively. The cumulative penalty decreases from an initial value of −145,045 in the first episode to an average of −59,081 by the end of training, corresponding to a 59.2% reduction. For comparison, the TD3 algorithm without curriculum learning achieves a smaller improvement of 48.5% over the same training duration.

During task 2, the episode rewards continue to increase steadily, indicating the development of a more refined control policy capable of managing vehicle stability under variable speed conditions. In task 3, where the agent encounters varying road friction coefficients, the rewards further converge toward an optimal policy. The final average reward achieved by the TD3 agent with curriculum learning is −23,431, representing a 23.0% reduction in cumulative penalty compared with the corresponding value obtained without curriculum learning. These results demonstrate that curriculum learning accelerates convergence, enhances learning stability, and improves the overall policy performance of the TD3 controller.

To support effective training, the TD3 agent with curriculum learning employs adaptive hyperparameter configurations across tasks. A higher exploration noise standard deviation and learning rate are applied in task 1 to promote diverse action exploration and rapid early-stage

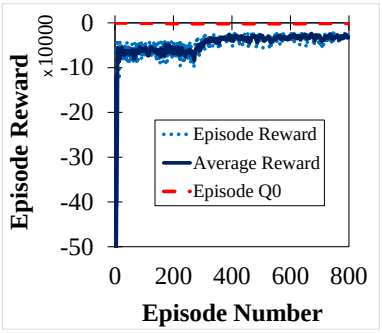
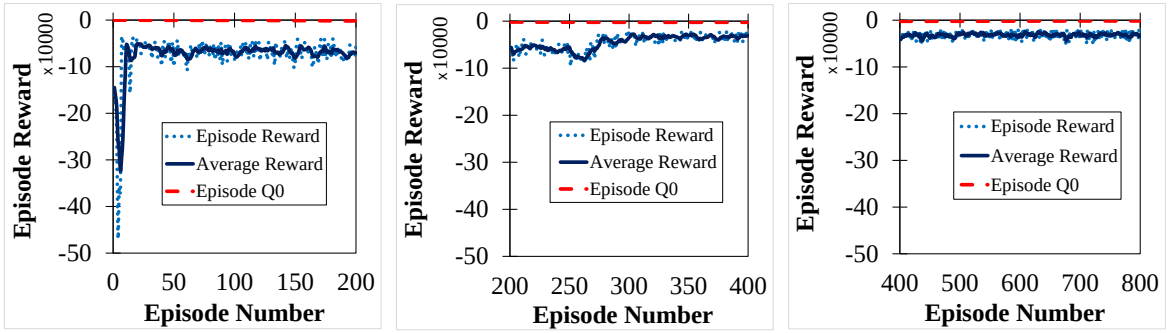


Fig. 3.7 Episode rewards for TD3 agent training without curriculum learning



(a) (b) (c)

Fig. 3.8 Episode rewards for TD3 agent training with curriculum learning: (a) Task 1, (b) Task 2, and (c) Task 3.

learning. As the agent progresses to tasks 2 and 3, these parameters are gradually reduced to stabilise training and refine policy convergence. Similarly, the noise mean attraction constant, which governs the rate at which exploration noise reverts to its mean, is set to a lower value in the first two tasks and increased in the final stage to favour consistent control behaviour. These adaptive adjustments ensure a balanced trade-off between exploration and exploitation, resulting in improved convergence robustness and better generalisation of the learned control policy.

3.7.4 Effectiveness of Curriculum Learning in TD3 Training

This section demonstrates the impact of integrating curriculum learning into the TD3 framework for vehicle dynamic control. As discussed earlier, the proposed curriculum learning structure divides the training process into three tasks of increasing complexity. This gradual progression allows the agent to first stabilise its control policy under simple conditions before adapting to more demanding scenarios. In contrast, a standard TD3 agent must learn across all variations from the outset, often leading to slower convergence and less stable

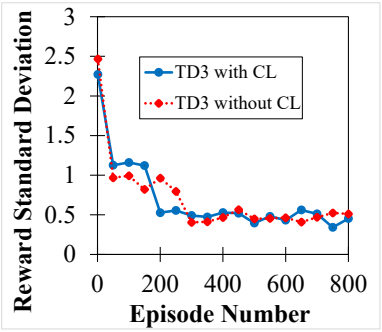


Fig. 3.9 Standard deviation of episode rewards for TD3 agents with and without curriculum learning.

learning. Furthermore, curriculum learning helps mitigate overestimation bias by decoupling the training stages and progressively increasing the state-space complexity. This ensures that the critic network learns accurate Q-value approximations early in training, reducing the likelihood of unstable or exaggerated estimates when exposed to complex environments.

The standard deviation of the episode rewards over every 50 training episodes is presented in Fig. 3.9 for both TD3 agents with and without curriculum learning. This metric quantifies the variability of the reward signal, providing insight into the consistency of the learning process. Both agents begin with high reward variability due to random exploration. In the first episode, the reward standard deviation is 2.27 for the curriculum-based TD3 agent and 2.46 for the baseline TD3 agent. By episode 300, the standard deviation decreases to 0.49 and 0.40, respectively. These results indicate that curriculum learning leads to a smoother and more stable learning progression, allowing the agent to retain moderate exploratory behaviour without destabilising training. Beyond episode 300, both agents maintain low reward variability, signifying convergence.

To further quantify convergence speed, three thresholds corresponding to 50%, 60%, and 70% reductions in the initial penalty are defined, as shown in Fig. 3.10. These thresholds represent the number of episodes required for the average episode reward to improve by a given percentage. The TD3 agent trained with curriculum learning consistently reaches each convergence threshold in fewer episodes than the baseline agent. At the 50% reduction level, the curriculum-based agent converges within 13 episodes, compared with 21 episodes for the baseline. At 60% reduction, convergence occurs at episode 211 versus 305, and at 70%, at episode 292 compared with 328. These results clearly demonstrate that curriculum learning accelerates policy convergence and enhances sample efficiency.

The earlier convergence achieved by the curriculum-based agent highlights its ability to learn effective control policies with significantly fewer training interactions, which is particularly valuable when computational resources or training data are limited. By enabling

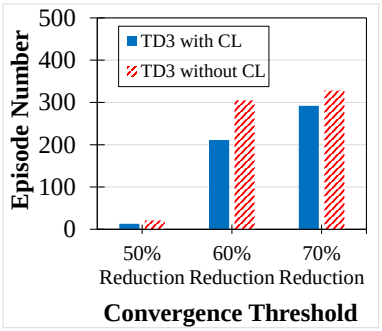


Fig. 3.10 Comparison of convergence speed for TD3 agents with and without curriculum learning.

more structured and progressive policy refinement, curriculum learning improves the training efficiency, stability, and generalisation of the TD3 controller. This makes it more suitable for real-time applications such as torque vectoring and yaw stability control, where reliable and adaptive performance under varying conditions is essential.

3.8 Results and Discussions

This section presents an evaluation of the proposed TD3-based torque vectoring controller under various driving scenarios. The performance of the controller is analyzed through simulation results obtained from MATLAB/Simulink and verified using IPG CarMaker. In particular, the performance of the proposed TD3 with curriculum learning and conventional LQR controllers are compared in Table 3.6 based on seven key performance metrics, including maximum of corrective yaw moment ($max|M_z|$), maximum of sideslip angle ($max|\beta|$), maximum of sideslip angle rate ($max|\dot{\beta}|$), integral of sideslip angle ($\int|\beta|$), integral of sideslip angle rate ($\int|\dot{\beta}|$), integral of yaw rate error ($\int|\dot{\psi_e}|$), and maximum lateral displacement from the desired path ($max|Y_e|$).

Table 3.6 Performance comparison of TD3 and LQR controllers based on different evaluation criteria

Manoeuvre	v [m/s]	μ	Controller	$max\|M_z\|$ [Nm]	$max\|\beta\|$ [deg]	$max\|\dot{\beta}\|$ [deg/s]	$\int\|\beta\|$ [deg·s]	$\int\|\dot{\beta}\|$ [deg]	$\int\|\dot{\psi}_e\|$ [deg/s]	$max\|Y_e\|$ [m]
Circular turning	25	0.5	LQR	501.0	2.454	5.277	21.440	2.900	9.780	7.665
			TD3	502.8	2.215	5.123	15.950	3.586	5.137	3.369
			Difference		9.7% ↓	2.9% ↓	25.6% ↓	23.6% ↑	47.4% ↓	56.0% ↓
Circular turning	30	0.7	LQR	611.6	3.271	7.074	29.500	3.620	5.505	4.302
			TD3	615.4	3.015	6.803	22.340	4.566	8.318	3.633
			Difference		7.8% ↓	3.8% ↓	24.2% ↓	26.1% ↑	51.0% ↑	15.5% ↓
Circular turning	35	0.9	LQR	678.1	3.986	8.727	36.120	4.335	7.106	3.822
			TD3	673.3	3.706	8.284	27.620	5.511	14.920	5.345
			Difference		7.0% ↓	5.0% ↓	23.5% ↓	27.1% ↑	109.9% ↑	39.8% ↑
Single-lane change	25	0.5	LQR	437.5	1.759	3.563	3.374	6.999	3.466	0.672
			TD3	436.4	1.524	3.179	2.897	6.042	3.355	0.580
			Difference		13.3% ↓	10.7% ↓	14.1% ↓	13.6% ↓	3.2% ↓	13.7% ↓
Single-lane change	30	0.7	LQR	534.4	2.207	4.395	4.261	8.745	2.830	6.438
			TD3	536.6	2.095	4.386	3.987	8.285	3.368	1.023
			Difference		5.0% ↓	0.2% ↓	6.4% ↓	5.2% ↓	19.0% ↑	84.1% ↓
Single-lane change	35	0.9	LQR	606.9	2.594	5.422	4.917	10.290	4.072	1.348
			TD3	608.0	2.561	5.258	5.069	10.150	3.268	0.848
			Difference		1.2% ↓	3.0% ↓	3.1% ↑	1.3% ↓	19.7% ↓	37.0% ↓
Double-lane change	25	0.5	LQR	369.4	1.516	3.405	3.220	6.869	6.311	0.740
			TD3	368.9	1.252	2.954	2.712	5.902	5.286	0.309
			Difference		17.4% ↓	13.2% ↓	15.7% ↓	14.0% ↓	16.2% ↓	58.2% ↓
Double-lane change	30	0.7	LQR	460.9	2.277	5.259	4.627	10.360	8.065	1.085
			TD3	461.9	1.910	4.802	3.980	9.188	6.955	0.355
			Difference		16.1% ↓	8.6% ↓	13.9% ↓	11.3% ↓	13.7% ↓	67.2% ↓
Double-lane change	35	0.9	LQR	561.2	3.128	7.303	6.168	14.280	9.625	1.761
			TD3	557.0	2.631	6.915	5.448	12.910	8.831	1.043
			Difference		15.8% ↓	5.3% ↓	11.6% ↓	9.6% ↓	8.2% ↓	40.7% ↓

To make a fair comparison, the maximum amplitudes of actions are set to be in the same range for both the proposed TD3 and conventional LQR controllers. Also, the tuning process for the LQR controller is carried out systematically using standard control design practices for each scenario. Specifically, the state-space model of the vehicle dynamics is linearized around a nominal operating point, and the state weighting matrix Q and control weighting matrix R are selected based on performance trade-offs between performance and control effort. The matrices are initially chosen based on prior literature and further refined for each scenario through iterative simulations to achieve balanced behavior across different manoeuvres. This tuning process ensures that the LQR controller operates at its best achievable performance while maintaining a comparable control effort to the TD3 agent, and provides a fair baseline to highlight the advantages of the model-free TD3 agent compared to the LQR controller, particularly in terms of adaptability and performance generalization. The comparison is carried out across a range of vehicle velocities (25 m/s, 30 m/s, and 35 m/s) and tyre-road friction coefficients (0.5, 0.7, and 0.9), under circular turning, single-lane change, and double-lane change manoeuvres. The test conditions are selected to be challenging in order to thoroughly evaluate the performance of controllers under demanding scenarios. The table also reports the percentage change for each criterion, in which downward arrows indicate a reduction in TD3 relative to LQR, whereas upward arrows indicate an increase.

In terms of maximum sideslip angle ($max|\beta|$), the proposed TD3-based controller generally achieves smaller peak values than the LQR baseline across most scenarios. For example, under the circular turning scenario at 25 m/s with $\mu = 0.5$, the LQR controller achieves $max|\beta| = 2.454$ deg, whereas TD3 reduces it to 2.215 deg, corresponding to a 9.7% reduction. Similar improvements are observed in the single-lane and double-lane change manoeuvres, where reductions of up to 17.4% are achieved. The maximum rate of sideslip angle ($max|\dot{\beta}|$) is also generally reduced by the TD3 controller. For instance, reductions of 13.2% and 8.6% are observed in the double-lane change manoeuvres at 25 m/s and 30 m/s, respectively. This indicates smoother transient responses during aggressive manoeuvres. Similarly, the integral of sideslip angle ($\int |\beta|$) is consistently lower for the TD3 controller in most scenarios. In circular turning manoeuvres, reductions of 25.6%, 24.2%, and 23.5% are observed at speeds of 25 m/s, 30 m/s, and 35 m/s, respectively.

The TD3 controller also reduces the integral of sideslip rate ($\int |\dot{\beta}|$) in most lane-change manoeuvres, with reductions of up to 14.0% in the double-lane change scenario at 25 m/s. However, under certain circular turning conditions, the LQR controller achieves slightly lower values, indicating that both controllers can perform competitively depending on the manoeuvre. The integral of yaw rate error ($\int |\dot{\psi}_e|$) is used to evaluate tracking performance. In most lane-change scenarios, the TD3 controller provides lower values than the LQR

baseline, with improvements of up to 19.7% in the single-lane change scenario at 35 m/s. Finally, the maximum lateral deviation from the desired path ($max|Y_e|$) is reduced in most manoeuvres. For example, in the double-lane change scenario at 30 m/s with $\mu = 0.7$, the TD3 controller reduces the lateral deviation by 67.2% compared to the LQR controller. These results demonstrate that the proposed TD3-based controller can improve vehicle stability and path tracking performance under demanding driving conditions. To further examine the robustness of the proposed controller under parameter uncertainties, a Monte Carlo analysis is also performed. The detailed setup of the Monte Carlo simulations, including the uncertain parameters and their distributions, is provided in Appendix B.

3.8.1 Circular Turning Manoeuvre

To further evaluate the performance of the proposed TD3-based torque vectoring controller in maintaining yaw stability, a circular turning manoeuvre is conducted at a constant longitudinal velocity of 25 m/s (90 km/h) with a tyre-road friction coefficient of 0.5. The results are compared with a conventional LQR-based controller, as shown in Fig. 3.11.

The trajectory of the vehicle under circular turning is shown in Fig. 3.11a, where both controllers closely follow the desired path. However, the TD3-based approach exhibits tighter lane adherence, indicating improved lateral tracking. The maximum lateral deviation from the desired path is 7.665 m for the conventional LQR controller, whereas the same criterion decreases to 3.369 m for the proposed TD3 controller. The velocity plot in Fig. 3.11b shows that the vehicle maintains the speed that is set by the driver. The yaw rate, shown in Fig. 3.11c, highlights the superior tracking accuracy of the TD3 controller compared to the conventional method. The TD3-based approach effectively reduces overshoot and oscillations, bringing the yaw rate closer to the desired reference. The maximum deviation of the yaw rate for the proposed TD3 controller is 5.137 deg/s, while this is 9.780 deg/s for the conventional controller. The sideslip angle in Fig. 3.11d indicates the maximum sideslip angle is 2.215 deg for the proposed controller, which is 9.7% less than the baseline LQR with 2.454 deg. This indicates enhanced lateral stability and better utilization of available tyre grip without approaching saturation. Fig. 3.11e illustrates the lateral acceleration, where the TD3-based controller offers a controlled and more stable response.

Also, the steering wheel angle is depicted in Fig. 3.11f to simulate the circular turning manoeuvre. Wheel angular velocities and torques are demonstrated in Fig. 3.11g and Fig. 3.11h, respectively. The TD3-based controller dynamically adjusts torque distribution, maintaining optimal traction while minimizing instability. Torque values are optimally distributed among the four wheels of the vehicle to maintain stability under this scenario. These results confirm that the TD3-based RL controller enhances yaw stability by reducing

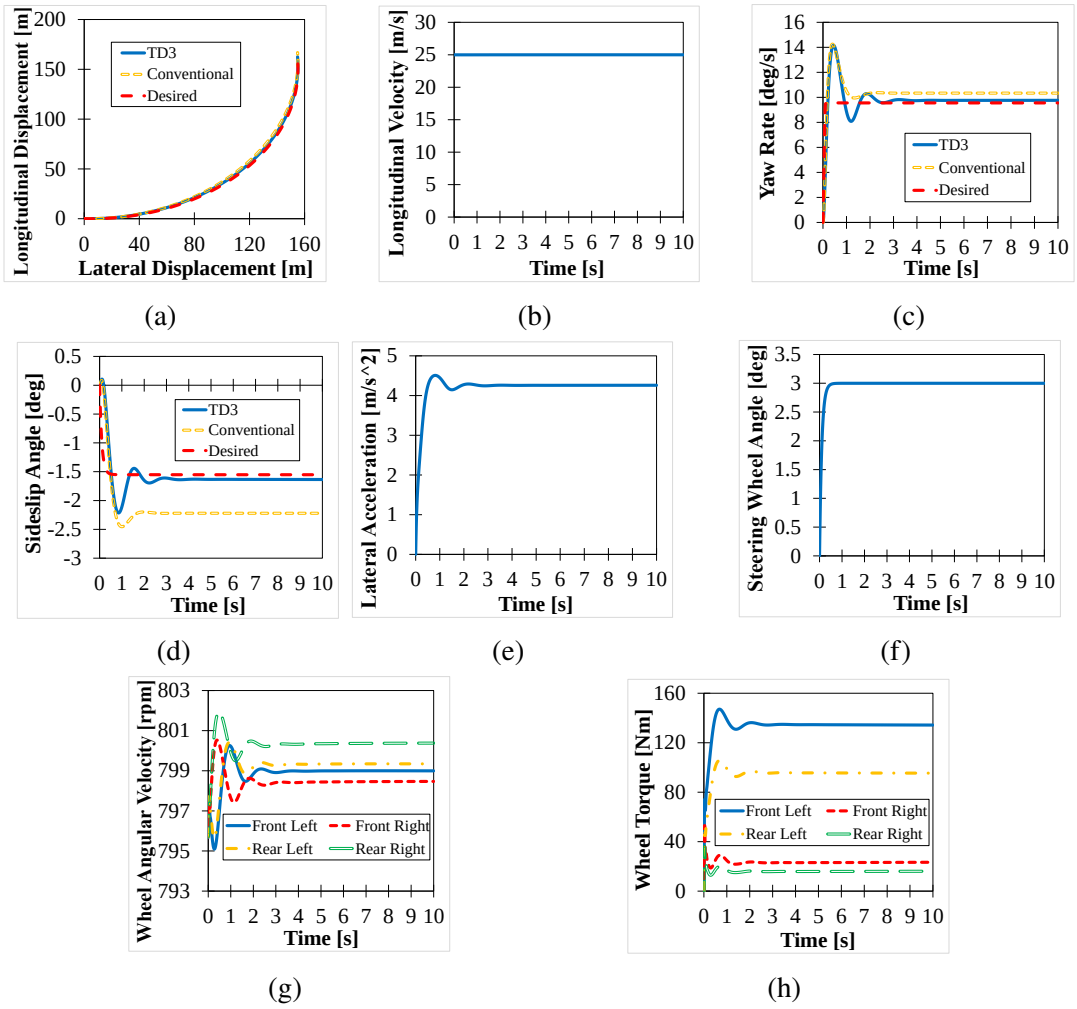


Fig. 3.11 Simulation results for circular turning manoeuvre: (a) Vehicle Trajectory, (b) Vehicle velocity, (c) Yaw rate, (d) Sideslip angle, (e) Lateral acceleration, (f) Steering wheel angle, (g) Wheel angular velocities, (h) Applied wheel torques.

yaw rate error, minimizing sideslip angle, and ensuring smoother torque allocation compared to the conventional LQR controller.

3.8.2 Single-Lane Change Manoeuvre

The single-lane change manoeuvre is also conducted to further evaluate the performance of the proposed TD3-based torque vectoring controller. The test is performed at a constant velocity of 15 m/s (54 km/h) on a snowy road surface with a low friction coefficient of 0.3. This scenario presents a challenging stability control task, as low friction reduces the available lateral tyre forces and increases the likelihood of oversteer or understeer. The simulation results are plotted in Fig. 3.12.

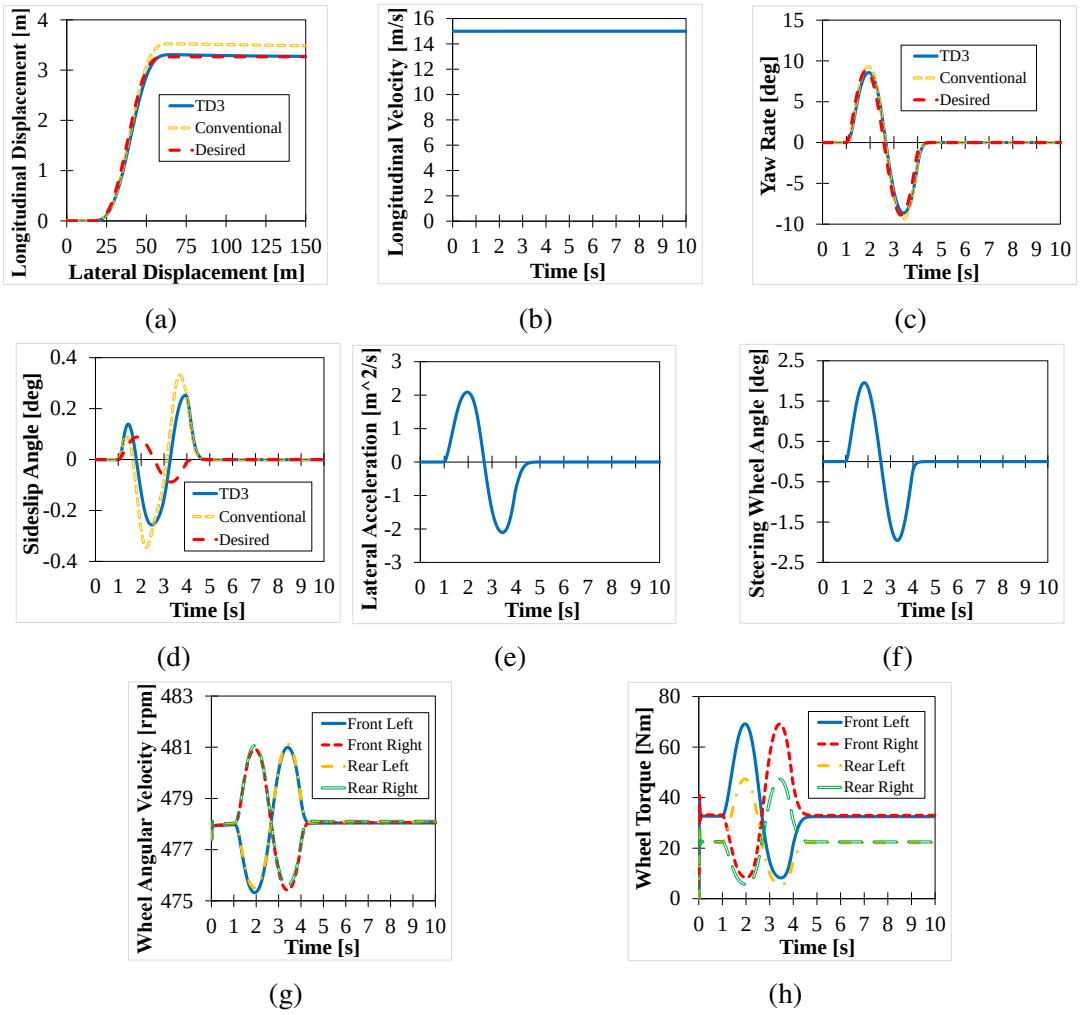


Fig. 3.12 Simulation results for the single-lane change manoeuvre: (a) Vehicle Trajectory, (b) Vehicle velocity, (c) Yaw rate, (d) Sideslip angle, (e) Lateral acceleration, (f) Steering wheel angle, (g) Wheel angular velocities, (h) Applied wheel torques.

The vehicle trajectory in Fig. 3.12a demonstrates that both controllers can track the desired path; however, the TD3-based approach achieves a smoother transition with reduced lateral deviation compared to the conventional method. This suggests that the TD3 agent is able to better regulate lateral tyre forces despite low traction. The velocity is plotted in Fig. 3.12b, which shows that the controller can achieve the desired velocity set by the driver. The yaw rate in Fig. 3.12c and sideslip angle in Fig. 3.12d indicate that the proposed controller can mitigate oscillations and maintain stability, closely aligning with the desired values, whereas the conventional controller exhibits higher fluctuations. The maximum amplitude of the sideslip angle reaches 0.256 deg for the TD3-based controller, which is reduced by 25.8% compared to the conventional LQR controller with a maximum amplitude of 0.345 deg.

Notably, the TD3 controller manages this without access to a vehicle model, instead relying solely on learned state–action trajectories, showcasing its generalization capability under unseen road conditions.

The maximum lateral acceleration amplitude for the TD3-based controller reaches 2.105 m^2/s, as can be seen in Fig. 3.12e. Fig. 3.12f shows the steering wheel angle for the single lane change scenario. The wheel angular velocities and applied wheel torques in Fig. 3.12g and Fig. 3.12h illustrate the real-time adaptation of the proposed controller in redistributing torque among the four wheels to maintain stability. Wheel angular velocities are within the range of approximately 475 rpm to 481 rpm. The minimum and maximum amplitudes of wheel torques are roughly 5 Nm and 70 Nm. These results confirm the effectiveness of the proposed controller in improving yaw stability and overall handling performance in critical driving scenarios. The TD3-based controller not only achieves more stable dynamics under the low-friction condition, but it also demonstrates superior performance in diminishing the lateral deviation from the desired trajectory and minimizing the sideslip angle.

3.8.3 Double-Lane Change Manoeuvre

A double-lane change manoeuvre is conducted at a longitudinal velocity of 20 m/s (72 km/h) with a tyre-road friction coefficient of 0.4. This scenario represents a dynamic and transient condition, requiring quick directional changes and precise control of lateral dynamics. The simulation results in Fig. 3.13 are compared against those obtained from a conventional LQR controller.

The trajectory plot in Fig. 3.13a illustrates that while both controllers are capable of completing the double-lane change, the TD3-based controller maintains a tighter lane trajectory, especially at the turning points. This tighter lateral path adherence is a result of better regulation of the lateral motion of the vehicle and indicates more precise path-following capability. The maximum lateral deviation is 0.391 m for the TD3-based controller, which is reduced by 29.3% compared to the conventional counterpart. The velocity profile is shown in Fig. 3.13b, in which the vehicle maintains the velocity set by the driver at 20 m/s. Fig. 3.13c compares the yaw rates for TD3 and LQR against the desired plot. The integral of yaw rate error is decreased by 6.8% for the TD3 controller, thus promoting smoother yaw dynamics. The sideslip angle in Fig. 3.13d confirms that the TD3 approach maintains smaller angular deviations during lane transitions, evidencing greater stability in sudden steering events. The maximum absolute value of sideslip angle is reduced by 14.1% for the proposed controller compared to the baseline counterpart. Moreover, the maximum sideslip angle rate for LQR

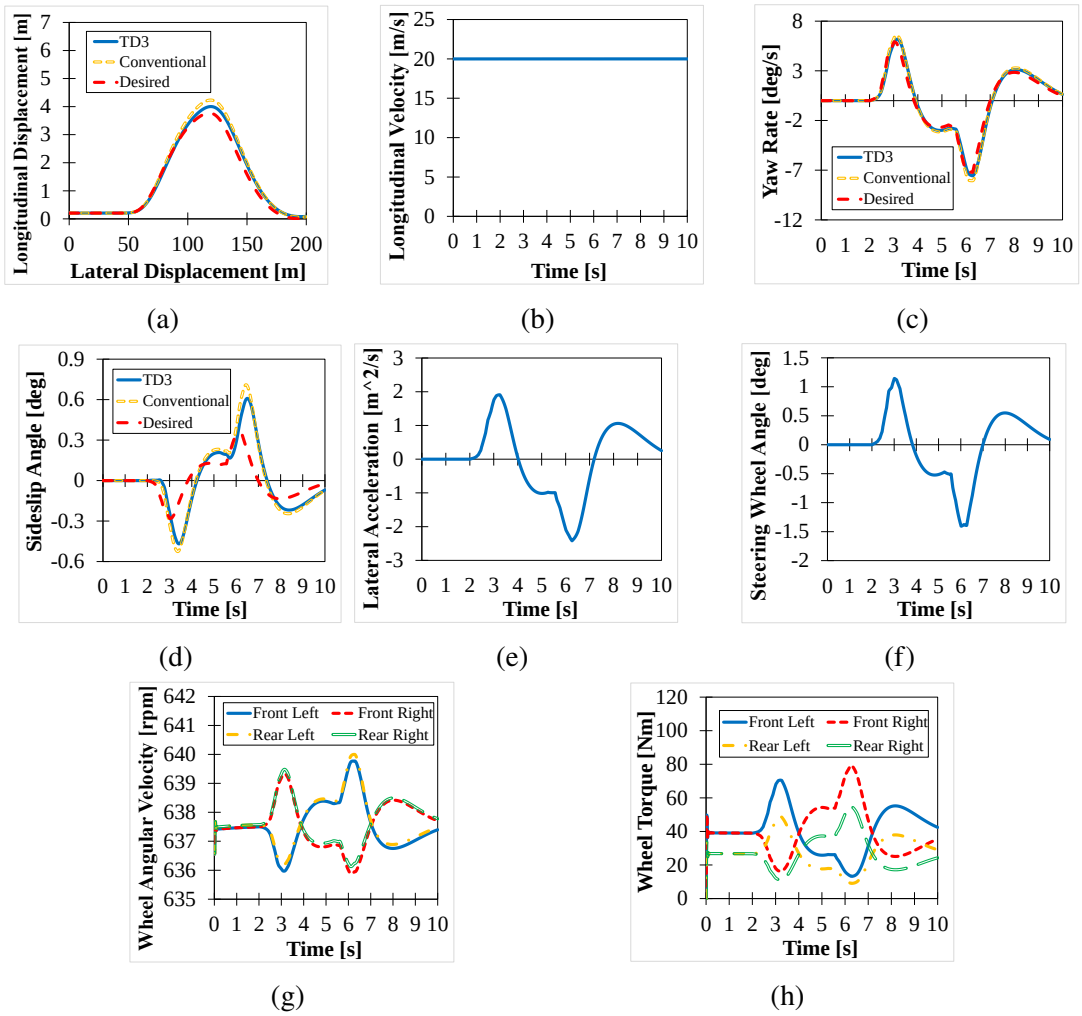
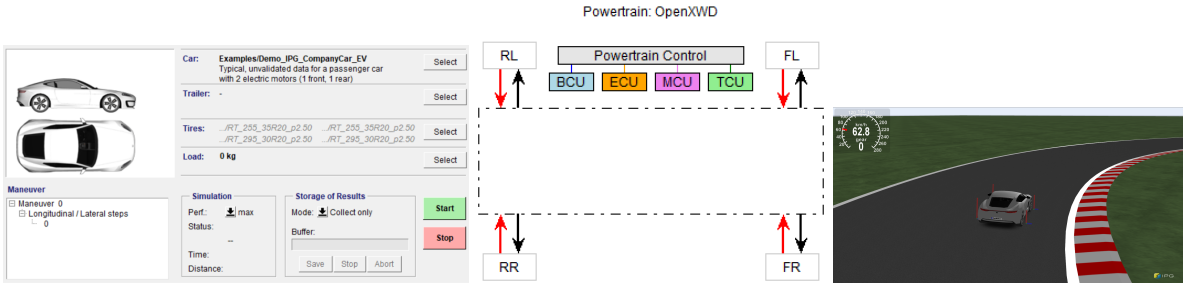


Fig. 3.13 Simulation results for the double-lane change manoeuvre: (a) Vehicle Trajectory, (b) Vehicle velocity, (c) Yaw rate, (d) Sideslip angle, (e) Lateral acceleration, (f) Steering wheel angle, (g) Wheel angular velocities, (h) Applied wheel torques.

and TD3 controllers are 1.096 deg/s and 0.952 deg/s, respectively, showing a reduction of 13.1% when the TD3 controller is used.

Fig. 3.13e presents the lateral acceleration with a controlled response in vehicle dynamics and stability. This confirms the smoother transient behavior with less oscillation during the lane entry and exit phases. The steering wheel angle in Fig. 3.13f illustrates the input profile of the driver for the double-lane change. Lastly, Fig. 3.13g and Fig. 3.13h compare the wheel angular velocities and applied wheel torques, respectively. The TD3 controller actively redistributes torque among the four wheels to mitigate oversteer or understeer tendencies, particularly at the lane-change transitions. Overall, these results verify that the TD3-based



(a) (b) (c)

Fig. 3.14 Verification in IPG CarMaker: (a) CarMaker GUI, (b) OpenXWD powertrain, and (c) Performance under a cornering manoeuvre.

controller enhances lateral stability, limits abrupt sideslip fluctuations, and achieves a more precise lane tracking under demanding double-lane change manoeuvres.

3.8.4 ISO 4138 Circular Turning Test in IPG CarMaker

To further verify the performance of the proposed TD3-based torque vectoring controller in a realistic environment, additional tests can be conducted using IPG CarMaker. The IPG CarMaker environment offers a higher level of fidelity by incorporating more realistic vehicle dynamics and environmental interactions. The main graphical user interface (GUI) of the CarMaker is shown in Fig. 3.14a. For this aim, the controller is designed in MATLAB/Simulink and implemented in IPG CarMaker. The OpenXWD powertrain model with Stand Alone configuration is selected in CarMaker to apply torques via an external user-defined driveline, as shown in Fig. 3.14b.

A circular turning test based on ISO 4138 standards is conducted in IPG CarMaker, and the speed control is selected for the longitudinal dynamics with a speed of 20 m/s (90 km/h). The results are illustrated in Fig. 3.15. The vehicle trajectory, depicted in Fig. 3.15a, shows that the vehicle can follow a circular path with a 100 m radius while maintaining stable lateral positioning. The longitudinal velocity is shown in Fig. 3.15b, which indicates that the controller effectively sustains the target speed of 20 m/s.

Fig. 3.15c compares the yaw rate responses for both the proposed TD3-based controller and the conventional LQR controller. The results indicate that both controllers exhibit similar performance. The differences become more evident in the sideslip dynamics. Sideslip angles are compared in Fig. 3.15d for the proposed TD3-based and baseline controllers. The peak sideslip angle ($max|\beta|$) is 0.439 deg for the TD3-based controller compared to 0.430 deg for the LQR controller, which is slightly higher for TD3. However, the TD3 controller demonstrates better overall stability, as evidenced by a 37.2% lower integral of the

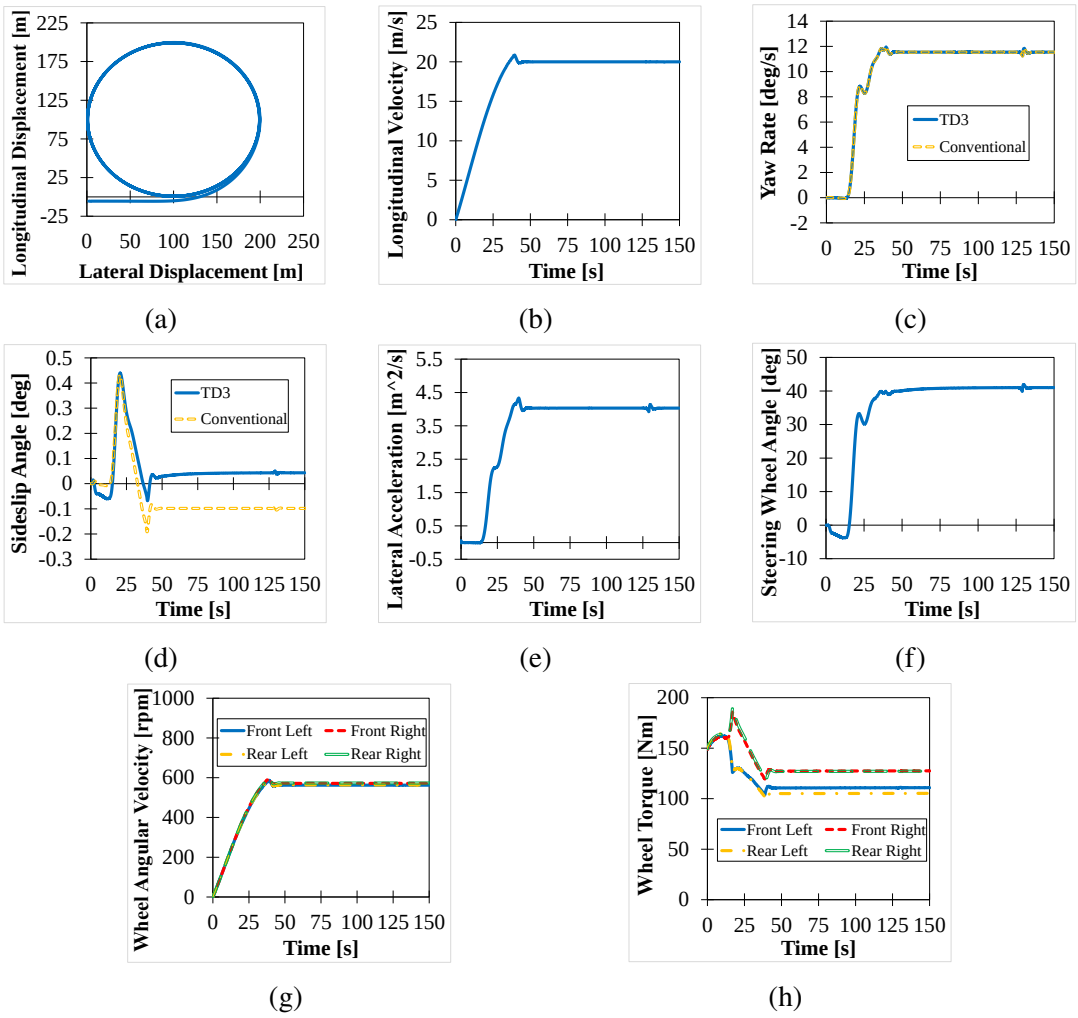


Fig. 3.15 Simulation results for the Circular Turning Test in IPG CarMaker: (a) Vehicle Trajectory, (b) Vehicle velocity, (c) Yaw rate, (d) Sideslip angle, (e) Lateral acceleration, (f) Steering wheel angle, (g) Wheel angular velocities, (h) Applied wheel torques.

absolute sideslip angle ($\int |\beta|$), indicating that it can more effectively minimize the cumulative lateral deviation over time. The peak sideslip angles are similar between the TD3-based controller and the LQR controller because both controllers aim to maintain lateral stability by minimizing sideslip, and they operate based on the same reference trajectory. The reference yaw rate and desired sideslip angle are derived using steady-state assumptions, which both controllers attempt to track closely. Additionally, in the evaluated scenario, the operating conditions are relatively nominal, with moderate speed and friction values, where both controllers are effective and converge toward similar corrective strategies. However, the key difference lies in how each controller handles sideslip over time. The TD3 controller demonstrates better cumulative performance, indicating smoother and more stable control

throughout the entire manoeuvre. Even marginal reductions in sideslip angle are meaningful under near-limit conditions. These small differences can prevent a vehicle from critical lateral stability thresholds, especially on low-friction surfaces.

The lateral acceleration is presented in Fig. 3.15e with a maximum value of 4.335 m^2/s. Fig. 3.15f shows the steering wheel angle input to follow the trajectory. The wheel angular velocities and the applied torques, illustrated in Figs. 3.15g and 3.15h, respectively, highlight how the TD3-based method adaptively distributes torque among the four wheels to preserve stability and manage understeer or oversteer tendencies. The proposed controller allocates torques with the peaks of approximately 185 Nm and 190 Nm to the front right and rear right wheels to obtain stability when the vehicle starts turning around the circular path. On the other hand, the torques applied to front left and rear left wheels drop to around 126 Nm and 127 Nm, respectively. This torque distribution highlights the effectiveness of the proposed torque vectoring strategy in maintaining vehicle stability during turning manoeuvres. In conclusion, the ISO 4138 test conducted in IPG CarMaker verifies the effectiveness of the proposed TD3-based controller in a high-fidelity environment, demonstrating its capability to maintain stable cornering dynamics.

3.8.5 Hockenheim Driving Test in IPG CarMaker

To further assess the adaptability and performance of the proposed TD3-based torque vectoring controller under dynamically changing road curvature and steering input, a simulation is conducted on the Hockenheim driving circuit using IPG CarMaker. The simulation results for the Hockenheim driving test using IPG CarMaker are depicted in Fig. 3.16 at a longitudinal velocity of 25 m/s (90 km/h) under dry road conditions.

The trajectory of the vehicle, shown in Fig. 3.16a, demonstrates the ability of the vehicle to follow the Hockenheim driving circuit without losing dynamic control. The velocity profile in Fig. 3.16b indicates a stable speed at 25 m/s (90 km/h) with speed control in longitudinal dynamics. Fig. 3.16c compares the yaw rate of the AWD EV for the proposed TD3-based controller with the desired yaw rate. This shows the effective performance of both controllers to mitigate excessive yaw and prevent instability. The similarity in yaw rate responses between the proposed TD3-based controller and the baseline LQR is attributed to the fact that both controllers are designed to track the same reference yaw rate, which is generated using a common vehicle model and steady-state assumptions. Since yaw rate is a primary target in yaw stability control, both controllers focus on minimizing yaw rate error, leading to similar overall tracking performance under standard conditions. However, the key advantage of the TD3-based controller lies in its adaptability and robustness under varying and challenging conditions, such as low-friction surfaces and higher-speed manoeuvres, where it maintains

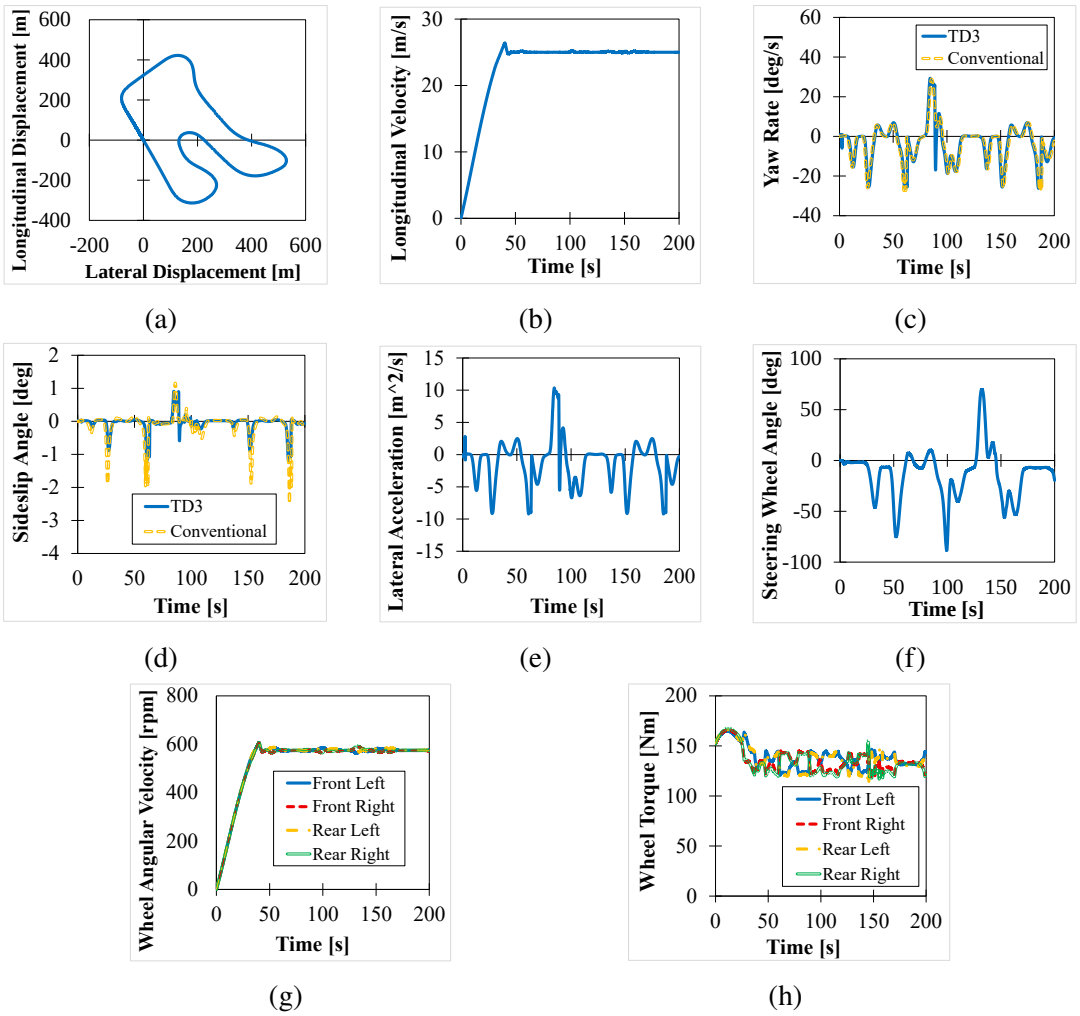


Fig. 3.16 Simulation results for the Hockenheim driving test using IPG CarMaker: (a) Vehicle Trajectory, (b) Vehicle velocity, (c) Yaw rate, (d) Sideslip angle, (e) Lateral acceleration, (f) Steering wheel angle, (g) Wheel angular velocities, (h) Applied wheel torques.

consistent performance without requiring model-based tuning or recalibration. Fig. 3.16d shows the sideslip angle of the TD3-based and LQR controllers. The maximum absolute value of the sideslip angle for the proposed TD3 is reduced by 52.4% compared to the baseline LQR. The lateral acceleration in Fig. 3.16e shows some variations, which correspond to the steering wheel angle input in Fig. 3.16f, required to navigate the turns of the track. Wheel angular velocities are shown in Fig. 3.16g reaching to a maximum value of approximately 580 rpm. Also, the torque distribution in Fig. 3.16h shows the optimal torque allocation by the proposed controller in favor of ensuring vehicle stability. The variations in applied torques across the four wheels demonstrate the real-time adaptation of the controller to maintain stability while following the curvatures in the test track. These results verify the effectiveness

of the TD3-based torque vectoring approach in enhancing vehicle stability, highlighting significant improvements in lateral control, sideslip angle minimization, and overall dynamic performance. The performance of the proposed controller is similar to the conventional LQR approach in IPG CarMaker-based tests. Moreover, its model-free nature enables the TD3 algorithm to adaptively sustain vehicle stability under a wide range of conditions, whereas the LQR controller relies on an accurate system model and parameter tuning to perform effectively across various scenarios.

3.9 Conclusion

This chapter presents the development and verification of RL-based torque vectoring controllers for an AWD EV using DDPG, TD3, and TD3 enhanced with curriculum learning. The controllers are trained to improve yaw stability and vehicle dynamics by reducing yaw rate error, limiting sideslip, and producing smooth torque distribution without relying on an explicit vehicle model. The simulation results show that the RL-based controllers improve stability and path tracking compared with a conventional LQR controller across circular turning, single-lane change, and double-lane change manoeuvres. In the circular turning test at 15 m/s and $\mu = 0.3$, the TD3 controller reduced the maximum sideslip angle by 40.6% and reduced the maximum lateral path error by 70.4%. In the same manoeuvre at 20 m/s and $\mu = 0.4$, the integral of yaw rate error was reduced by 47.5%. In the double-lane change test at 25 m/s and $\mu = 0.5$, the TD3 controller reduced the maximum lateral deviation by 58.2%. These results confirm that the proposed RL controllers achieve improved lateral stability and better tracking under low-friction and high-demand conditions.

TD3 provides more stable learning than DDPG due to clipped double Q-learning, delayed policy updates, and target policy smoothing. Curriculum learning further improves training behaviour by progressively increasing task difficulty. Compared with TD3 without curriculum learning, the curriculum-based TD3 achieved faster convergence and improved final performance, with the final average episode penalty reduced by 23.0%. The controller performance is also verified in a high-fidelity environment using IPG CarMaker, confirming that the learned policy can be transferred from MATLAB/Simulink simulation to a more realistic vehicle simulation platform. Overall, the key finding of this chapter is that TD3, especially when combined with curriculum learning, provides an effective model-free torque vectoring solution that improves yaw stability, reduces sideslip, and reduces path tracking error compared with a tuned LQR baseline. The next chapter extends this RL framework to integrate vehicle stability control with energy optimisation within a unified control structure.

Chapter 4

Reinforcement Learning-Based Energy Optimization and Vehicle Dynamics Control

4.1 Introduction

This chapter presents the proposed RL-based framework for integrated energy optimisation and vehicle dynamics control in an AWD EV. Building upon the torque vectoring concepts and hierarchical control architecture previously introduced, RL is employed to learn optimal control policies that simultaneously address vehicle stability and energy efficiency. Specifically, actor–critic algorithms are integrated within the hierarchical control structure to generate corrective yaw moments and allocate wheel torques while considering both dynamic performance and electric machine efficiency. The formulation of the control problem, the design of the reward function, and the implementation of the RL training framework are described in the following sections.

4.2 Control Framework and Optimization Objectives

This study develops an RL-based torque vectoring control framework for integrated vehicle stability control and energy optimisation in AWD EVs. Three actor–critic algorithms, namely DDPG, TD3, and CL-TD3, are employed to generate the corrective yaw moment in the high-level controller, which is then translated into individual wheel torques through a low-level allocation strategy. The RL agent is trained in a simulated environment using a multi-objective

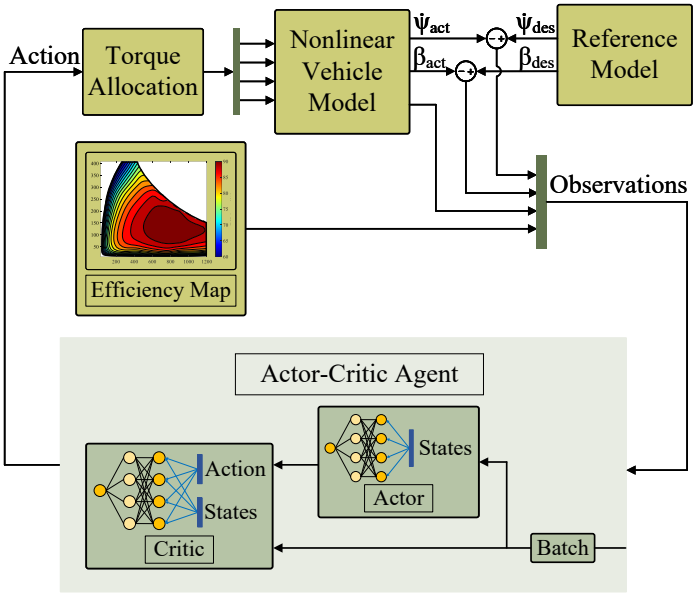


Fig. 4.1 A schematic of the actor-critic RL-based control framework for energy optimisation and vehicle dynamics control.

reward function that accounts for vehicle stability and energy efficiency. The overall control architecture is shown in Fig. 4.1 [9].

4.2.1 Vehicle Dynamics Enhancement

Three RL-based agents are implemented in this study to generate the corrective yaw moment in the high-level controller, i.e., DDPG, TD3, and CL-TD3. These agents share the same observation and action spaces but differ in their learning mechanisms [166, 165]. Their detailed formulations are presented in the rest of this chapter. To provide meaningful performance benchmarks, two conventional model-based controllers are also developed, i.e., LQR and SMC approaches, each combined with sequential quadratic programming (SQP) for torque distribution and energy optimization. SQP is a gradient-based nonlinear optimization method widely used for solving constrained optimization problems. It operates by iteratively solving a sequence of quadratic programming subproblems that approximate the original nonlinear problem while satisfying system constraints. In this study, SQP is employed in the low-level controller to optimally distribute wheel torques while considering actuator limits and motor efficiency characteristics. These benchmark controllers, referred to as LSQP and SSQP, enable a fair comparison between model-based and model-free control paradigms. The model-based high-level controllers generate the corrective yaw moment. The computed yaw

moment is then transmitted to the low-level SQP-based torque allocator, which distributes the required torque among the four wheels while considering motor efficiency constraints.

The SMC law that determines the corrective yaw moment performs based on a sliding surface defined as a weighted combination of yaw rate and sideslip angle errors, expressed as:

$$S = \dot{\psi}_e - \zeta \beta_e \tag{4.1}$$

where ζ is a weighting coefficient that governs the relative influence of sideslip on the control action. The control law is designed to drive $S \rightarrow 0$ using a discontinuous switching term, ensuring robustness to model uncertainties and external disturbances. Since SMC does not inherently account for energy optimization, SQP is integrated to minimize power losses while maintaining dynamic stability. Consequently, the hybrid SSQP controller combines the robustness of SMC with the optimization capability of SQP, serving as a strong model-based reference for comparison with the proposed RL-based methods [101].

4.2.2 Energy-Efficient Control Strategy

Energy management is a critical factor in determining the overall performance and driving range of EVs [86, 85]. The proposed control framework aims to enhance vehicle dynamics while simultaneously optimizing energy consumption through both RL-based and model-based approaches. To achieve this, the real efficiency characteristics of the electric machines are incorporated directly into the control framework, allowing the controller to consider the energy cost of torque generation during operation, as illustrated in Fig. 4.2.

Permanent magnet synchronous machines (PMSMs) are employed in this study due to their high power density, compact design, and excellent suitability for in-wheel applications. Each wheel of the AWD EV is driven by an outer-rotor PMSM with interior magnets, providing fast torque response and high efficiency across a broad operating range. The motoring and regenerative efficiency maps are integrated into the control architecture to guide both model-based and RL agents toward energy-optimal torque allocation, ensuring a balanced trade-off between dynamic stability and power efficiency. For the RL-based controllers, energy optimisation is incorporated through a reward component that penalises power loss and encourages operation in high-efficiency torque–speed regions. The power loss characteristics of the PMSM used in this study are shown in Fig. 4.3, illustrating the regions of elevated loss across the torque–speed plane.

The model-based benchmark methods, i.e., LSQP and SSQP, are employed for comparison. In these benchmarks, energy optimization is performed by introducing a correction term, ΔT_{ij}, to the reference torque T_{ij}^{ref} for each wheel, where $ij \in \{fl, fr, rl, rr\}$ represents the front-left,

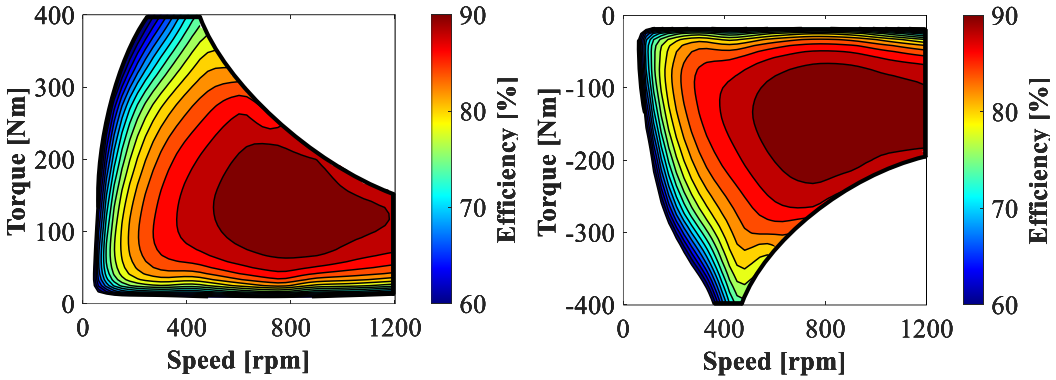


Fig. 4.2 Efficiency map of a PMSM: (a) Motoring Efficiency, (b) Regenerative braking efficiency.

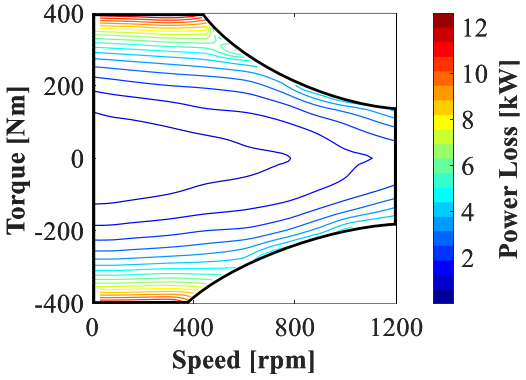


Fig. 4.3 Power loss distribution of the PMSM used in this study.

front-right, rear-left, and rear-right wheels, respectively. The objective is to minimize the total power loss across all four in-wheel electric machines while maintaining the desired yaw moment for stability. The optimization problem is formulated as:

$$\min_{\Delta T_{ij}} \sum_{ij} \frac{(T_{ij}^{\text{ref}} + \Delta T_{ij}) \cdot \omega_{ij}}{\eta(T_{ij}^{\text{ref}} + \Delta T_{ij}, \omega_{ij})} \tag{4.2}$$

where ω_{ij} is the angular velocity of each wheel, and $\eta(T, \omega)$ is the machine efficiency obtained from a calibrated two-dimensional lookup table as a function of torque and angular velocity. The numerator represents the mechanical power delivered to each wheel, while the denominator accounts for efficiency, penalizing torque allocations that fall within low-efficiency regions. The optimization variables ΔT_{ij} are bounded within actuator torque limits. The optimizer receives the current wheel speeds ω_{ij} and reference torques T_{ij}^{ref} as inputs

and outputs the optimal torque corrections. The resulting optimized torque commands are expressed as:

$$T_{ij} = T_{ij}^{\text{ref}} + \Delta T_{ij} \tag{4.3}$$

This hierarchical strategy ensures that vehicle stability, maintained by the high-level LQR or SMC controllers, is complemented by improved energy efficiency through the low-level optimization process.

4.3 Reinforcement Learning for Vehicle Dynamics Control and Energy Optimization

The RL-based structures in this study perform based on the actor-critic architecture, i.e., an actor that maps the observed state to a deterministic control action, and a critic that estimates the action-value $Q(s_t, a_t)$. During training, the critic evaluates how well a sampled action reduces a long-horizon cost, and the actor updates its parameters via the deterministic policy gradient to maximize that estimate.

Deep Deterministic Policy Gradient (DDPG)

The DDPG algorithm is a model-free, off-policy reinforcement learning approach designed for continuous control problems [166]. It adopts an actor–critic structure, where the actor network $\mu(s|\theta^{\mu})$ outputs deterministic control actions, and the critic network $Q(s, a|\theta^{Q})$ estimates the corresponding action-value function. The critic is trained using the temporal-difference (TD) error derived from the Bellman equation:

$$L(\theta^{Q}) = \mathbb{E}_{(s,a,r,s',d)}\left[\left(Q(s,a|\theta^{Q}) - y\right)^2\right], \quad y = r + (1-d)\,\gamma\, Q'\left(s', \mu'(s'|\theta^{\mu'})|\theta^{Q'}\right) \tag{4.4}$$

where r is the immediate reward, s' is the next state, and $d \in \{0, 1\}$ is the terminal indicator, with $d = 1$ when the episode terminates and $d = 0$ otherwise. The target networks (μ', Q') are updated softly for training stability. The actor parameters are updated through the deterministic policy gradient:

$$\nabla_{\theta^{\mu}} J = \mathbb{E}_s\left[\nabla_a Q(s,a|\theta^{Q})\big|_{a=\mu(s)} \nabla_{\theta^{\mu}} \mu(s|\theta^{\mu})\right] \tag{4.5}$$

To improve learning robustness, DDPG employs an experience replay buffer to decorrelate training samples, and Ornstein–Uhlenbeck (OU) noise is added to the actor output during training to promote exploration in continuous action spaces. In this work, DDPG is used to

Algorithm 4 DDPG for torque vectoring with energy optimization

Initialise actor $\mu(s\,|\,\theta^{\mu})$, critic $Q(s,a\,|\,\theta^{Q})$
Initialise target networks μ' and Q' with $\theta^{\mu'} \leftarrow \theta^{\mu}$, $\theta^{Q'} \leftarrow \theta^{Q}$
Initialise replay buffer $\mathcal{D}$
for each episode **do**
    Reset energy-aware vehicle dynamics plant, observe state s
    for each time step **do**
        Select action $a = \mu(s;\theta^{\mu}) + \mathcal{N}_t$, apply to plant
        Observe next state s', reward r, terminal flag *done*; store (s,a,r,s',done) in $\mathcal{D}$
        Sample mini-batch $\{(s_i,a_i,r_i,s'_i,d_i)\}_{i=1}^{B} \sim \mathcal{D}$
        $y_i = r_i \;+\; (1-d_i)\,\gamma\, Q'(s'_i, \mu'(s'_i;\theta^{\mu'});\theta^{Q'})$
        Update critic: $L = \frac{1}{B}\sum_i (y_i - Q(s_i,a_i;\theta^{Q}))^2$; $\theta^{Q} \leftarrow \theta^{Q} - \eta_Q \nabla_{\theta^Q} L$
        Update actor: $\theta^{\mu} \leftarrow \theta^{\mu} + \eta_{\mu} \frac{1}{B}\sum_i \nabla_a Q(s_i,a;\theta^{Q})|_{a=\mu(s_i)} \nabla_{\theta^{\mu}} \mu(s_i;\theta^{\mu})$
        Soft-update targets: $\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'}$, $\theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'}$
        $s \leftarrow s'$; **if** *done* **then break**
    end for
end for
Deploy $\mu(s;\theta^{\mu})$ on unseen manoeuvres

generate the corrective yaw moment for integrated stability and energy-aware torque vectoring. The training procedure for the DDPG-based torque vectoring with energy optimization is summarized in Algorithm 4.

Twin Delayed Deep Deterministic Policy Gradient (TD3)

The TD3 algorithm extends the DDPG framework by introducing three key mechanisms to address overestimation bias and instability in value function approximation, i.e., twin critics, target-policy smoothing, and delayed actor updates [165]. Two independent critic networks, $Q_1(s,a|\theta^{Q_1})$ and $Q_2(s,a|\theta^{Q_2})$, are trained simultaneously, and the minimum of their target estimates is used to mitigate optimistic value predictions. During target computation, the action is perturbed with clipped Gaussian noise, $\tilde{a} = \mu'(s') + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0,\sigma^2)$, to prevent exploitation of sharp Q-value peaks and to encourage smoother policy learning. The actor network is updated only after a fixed number of critic updates, allowing more accurate value estimates before policy improvement.

The critic target in TD3 is expressed as:

$$y_i = r_i + (1-d_i)\,\gamma \min_{j\in\{1,2\}} Q'_j\left(s'_i, \tilde{a}_i \,|\, \theta^{Q'_j}\right) \tag{4.6}$$

Algorithm 5 TD3 for torque vectoring with energy optimization

Initialise actor $\mu(s|\theta^{\mu})$, critics Q_1, Q_2, and their targets
Initialise replay buffer $\mathcal{D}$
for each episode **do**
  Reset energy-aware vehicle dynamics plant with random (m, v, μ); observe s
  for each time step **do**
    $a = \mu(s) + \mathcal{N}_{\mathrm{OU}}$; apply to plant, observe (s', r, d)
    Store (s, a, r, s', d) in $\mathcal{D}$; sample mini-batch $\{\cdot\}_{i=1}^{B}$
    Critic targets:
      $\tilde{a}_i = \mu'(s'_i) + \mathrm{clip}(\varepsilon_i)$
      $y_i = r_i + (1 - d_i)\gamma \min\big(Q'_1(s'_i, \tilde{a}_i), Q'_2(s'_i, \tilde{a}_i)\big)$
    Update both critics with $L = \frac{1}{B}\sum (y_i - Q_j(s_i, a_i))^2$
    if step mod $d = 0$ **then**
      Update actor:
        $\theta^{\mu} \leftarrow \theta^{\mu} + \eta_{\mu} \frac{1}{B} \sum \nabla_a Q_1(s_i, a)|_{a=\mu(s_i)} \nabla_{\theta^{\mu}} \mu(s_i)$
      Soft-update all targets:
        $\theta^{(\cdot)'} \leftarrow \tau\theta^{(\cdot)} + (1 - \tau)\theta^{(\cdot)'}$
    end if
    $s \leftarrow s'$; **if** $d = 1$ **then** break
  end for
end for
Deploy trained actor $\mu(\cdot|\theta^{\mu})$ on unseen manoeuvres

where $d_i \in \{0, 1\}$ is the terminal indicator for sample i, ensuring that value bootstrapping is disabled when a terminal transition is reached, and each critic minimizes the loss:

$$L_j = \frac{1}{B} \sum_{i=1}^{B} \big(y_i - Q_j(s_i, a_i|\theta^{Q_j})\big)^2 \tag{4.7}$$

The actor update follows the deterministic policy gradient but is performed on the delayed schedule. Experience replay, soft target updates, and Ornstein–Uhlenbeck (OU) exploration noise are inherited from DDPG. TD3 is employed to improve the robustness of learning under combined stability and energy objectives. Its dual-critic structure and delayed policy updates provide more reliable value estimates for the torque vectoring problem. The overall training procedure is summarized in Algorithm 5. A comparison between DDPG and TD3 algorithms is carried out in Table 4.1.

Table 4.1 Comparison of actor–critic DDPG and TD3 algorithms

Characteristic	DDPG	TD3
Model type	Model-free	Model-free
Policy type	Deterministic	Deterministic
Learning type	Actor–critic	Actor–critic
Value function	Q-value	Q-value (two critics)
Exploration method	OU noise	OU noise + target policy smoothing
Action space	Continuous	Continuous
Policy gradient	Yes	Yes
On-policy / off-policy	Off-policy	Off-policy
Replay buffer	Yes	Yes
Sample efficiency	Moderate	High
Stability mechanism	Target networks	Twin critics, delayed updates, smoothing

TD3 Enhanced with Curriculum Learning (CL TD3)

Curriculum learning is integrated into the TD3 framework to enhance sample efficiency and accelerate convergence in complex, high-dimensional environments [143]. The key idea is to expose the agent to progressively more challenging training conditions, allowing it to master simpler tasks before addressing more difficult ones. In this work, the CL TD3 framework is organised into three sequential tasks, i.e., (i) random variation of the steering wheel angle, (ii) randomisation of the vehicle velocity between 10 and 25 m/s, and (iii) introduction of varying tyre–road friction coefficients ranging from 0.3 to 1.0. While the underlying learning algorithm remains similar to TD3, the curriculum-based training process improves policy robustness, stabilises learning, and promotes more structured exploration, leading to faster and more consistent convergence. The algorithm for implementing CL TD3 for torque vectoring with energy optimization is presented in Algorithm 6.

4.3.1 Hyperparameter Settings

To ensure a fair comparison among the RL algorithms (DDPG, TD3, and CL TD3), similar environment settings and shared hyperparameters are adopted. The list of hyperparameters is provided in Table 4.2. All agents are trained for 900 episodes under the same simulation conditions. Since training is performed offline, the computational cost does not affect real-time deployment. During operation, the controllers only require a forward pass of the actor network at each control step, making them suitable for onboard implementation.

Algorithm 6 CL TD3 for torque vectoring with energy optimisation

Initialise actor μ, critics Q_1, Q_2, targets, replay buffer $\mathcal{D}$
Define curriculum: tasks $\mathcal{T} = \{\text{Task 1}, \text{Task 2}, \text{Task 3}\}$ and return thresholds $\{\rho_1, \rho_2\}$
for task $k = 1$ to $|\mathcal{T}|$ **do**
    for episode = 1 to $N_{\text{ep},k}$ **do**
        Reset 7-DOF plant with parameters drawn from task $\mathcal{T}[k]$; observe s
        for each time step **do**
            $a = \mu(s) + \mathcal{N}_{\text{OU}}$; apply to plant, observe (s', r, d)
            Store (s, a, r, s', d) in $\mathcal{D}$; sample mini-batch
            Compute target with twin critics and clipped noise; update critics
            if step mod $d = 0$ **then**
                update actor and soft-update targets
            end if
            $s \leftarrow s'$; **if** $d = 1$ **then break**
        end for
    end for
    if mean return $\geq \rho_k$ **then**
        move to next task
    end if
end for
Deploy trained actor $\mu(\cdot|\theta^{\mu})$ on unseen manoeuvres

Shared training parameters are selected to stabilise learning and ensure sufficient exploration. The experience buffer size is set to 1e6, and the mini-batch size is 64 for all algorithms. Gaussian noise with a standard deviation of $\sqrt{0.3}$ is added to the target action for policy smoothing in TD3 and CL TD3. A discount factor of 0.995 is used to prioritise long-term rewards while maintaining responsiveness to immediate control performance. The Ornstein–Uhlenbeck (OU) noise process is employed for exploration during training. OU noise introduces temporally correlated perturbations suitable for systems with inertia, such as vehicle dynamics. The related parameters, including noise standard deviation, decay rate, and mean attraction constant, are tuned to balance exploration and exploitation throughout training. Both TD3 and CL TD3 use a target update frequency of 2 and a target smooth factor of 0.005. The policy update frequency, defining the number of critic updates before an actor update, is also set to 2 for these algorithms.

To regulate the learning pace, the actor and critic learning rates, α_π and α_Q, are set to 1e-5 and 1e-4, respectively. The critic learns directly from reward signals and therefore requires a

Table 4.2 Hyperparameter settings for RL-based algorithms

Description	DDPG	TD3	CL TD3		
			General settings		
Experience buffer size	1e6	1e6	1e6		
Mini-batch size	64	64	64		
Actor gradient threshold	1	1	1		
Critic gradient threshold	1	1	1		
Target policy standard deviation	–	$\sqrt{0.3}$	$\sqrt{0.3}$		
Discount factor	0.995	0.995	0.995		
Noise model	OU	OU	OU		
Target update frequency	–	2	2		
Target smooth factor	–	0.005	0.005		
Policy update frequency	–	2	2		
			Task-specific settings		
Maximum number of episodes	900	900	200	300	400
Actor learning rate	1e-5	1e-5	1e-4	1e-5	1e-5
Critic learning rate	1e-4	1e-4	1e-3	1e-4	1e-4
Noise standard deviation	65	65	72	68	65
Noise standard deviation decay rate	1e-5	1e-5	1e-4	1e-5	1e-5
Noise mean attraction constant	0.2	0.2	0.15	0.15	0.2

higher learning rate to adapt faster to the environment, while the actor updates its policy more conservatively to maintain stability. For CL TD3, higher learning rates are applied during task 1, followed by gradual reductions as training progresses. Gradient thresholds of 1 are applied to both actor and critic networks to prevent exploding gradients and stabilise learning.

A sensitivity analysis is conducted on the actor learning rate α_π and critic learning rate α_Q to evaluate their influence on controller performance. The analysis is performed for the TD3 agent under a single-lane change scenario at a velocity of 15 m/s and a tyre–road friction coefficient of 0.4. Table 4.3 summarises the outcomes across nine stability- and energy-related performance indicators (maximum absolute sideslip angle ($\max|\beta|$), maximum absolute sideslip rate ($\max|\dot{\beta}|$), integral of absolute sideslip angle ($\int|\beta|$), integral of absolute sideslip rate ($\int|\dot{\beta}|$), integral of absolute yaw rate error ($\int|\dot{\psi}_e|$), maximum absolute lateral deviation from the desired path ($\max|Y_e|$), maximum absolute tyre slip ratio ($\max|\lambda_{ij}|$), average efficiency of the four electric machines (η_{avg}), and total power consumption (P_{tot})). The baseline configuration is set to $\alpha_\pi = 1e-5$ and $\alpha_Q = 1e-4$. Also, the normalised sensitivity weight index (SWI) given in Eq. (3.31) is used to investigate the impact of each learning rate on the performance metrics, with all other hyperparameters kept constant.

The results show that increasing the actor learning rate to $1e-4$ or $1e-3$ leads to reduced yaw stability and tracking accuracy, as evidenced by higher values of $\int |\dot{\psi}_e|$ and $\max |Y_e|$. Conversely, lowering the critic learning rate to $1e-5$ produces unstable training behaviour, reflected in large SWI magnitudes across stability metrics. Increasing the critic learning rate to $1e-3$ also results in moderate performance deterioration. These findings confirm that the baseline configuration provides the best trade-off between training stability, dynamic performance, and energy efficiency.

4.3.2 Actor and Critic Neural Network Architecture

Fig. 4.4 shows the neural network architectures employed for the actor and critic in the DDPG, TD3, and CL TD3 frameworks. The actor network receives the observation vector as input and processes it through four fully connected (FC) layers with 256, 128, 64, and 32 neurons, respectively, each followed by a ReLU activation function. The output layer applies a Tanh activation to bound the action within the range $[-1, 1]$, followed by a scaling layer to map the output to the physical action limits.

The critic network takes both the observation and the corresponding action as inputs. The observation is processed through two FC layers with 256 and 128 neurons, respectively, each activated by ReLU, while the action passes through a separate FC layer with 128 neurons. The extracted feature vectors are then concatenated and passed through two additional FC layers with 64 and 32 neurons, respectively, using ReLU activations. The final output represents the estimated Q-value. This modular structure enables efficient feature extraction for both policy generation and value estimation, providing a balance between representational capacity and computational efficiency.

4.3.3 State, Action, and Reward Definitions

The choice of state variables is crucial for enabling the RL agent to make accurate and effective control decisions. In this study, the state vector is defined as:

$$s_t = \{\dot{\psi}_e, \int \dot{\psi}_e, \beta, \beta_e, v_x, v_e, a_y, \omega_{ij}, \lambda_{ij}, F_{x,t}\} \tag{4.8}$$

where $\dot{\psi}_e$ and $\int \dot{\psi}_e$ represent the yaw rate error and its integral, capturing both instantaneous and accumulated deviations from the desired yaw response. The sideslip angle β and its error β_e quantify lateral stability.

Table 4.3 Sensitivity analysis of learning rates on stability and energy-related performance metrics

Actor learning rate (α_π)	Critic learning rate (α_Q)		$max\|\beta\|$ [deg]	$max\|\dot{\beta}\|$ [deg/s]	$\int\|\beta\|$ [deg·s]	$\int\|\dot{\beta}\|$ [deg]	$\int\|\dot{\psi}_e\|$ [deg/s]	$max\|Y_e\|$ [m]	$max\|\lambda_{ij}\|$ [%]	η_{avg} [%]	P_{tot} [kJ]
1e-5	1e-4	Value	0.205	0.594	0.392	1.095	2.979	0.274	0.502	79.66	210.0
		SWI	–	–	–	–	–	–	–	–	–
1e-4	1e-4	Value	0.223	0.576	0.451	1.155	3.231	0.513	0.524	75.14	218.10
		SWI	–0.975	+0.336	–1.672	–0.608	–0.939	–9.691	–0.486	–0.630	–0.428
1e-3	1e-4	Value	0.369	0.865	0.752	1.651	4.982	1.207	0.534	74.82	212.8
		SWI	–0.808	–0.460	–0.927	–0.512	–0.679	–3.439	–0.064	–0.061	–0.013
1e-5	**1e-5**	Value	0.356	0.847	0.658	1.653	4.184	0.489	0.616	79.06	212.5
		SWI	–81.842	–47.325	–75.396	–56.621	–44.944	–87.185	–25.232	–0.836	–1.322
1e-5	**1e-3**	Value	0.217	0.565	0.453	1.148	3.209	0.290	0.541	76.71	214.8
		SWI	–0.650	+0.542	–1.729	–0.537	–0.857	–0.648	–0.863	–0.411	–0.253

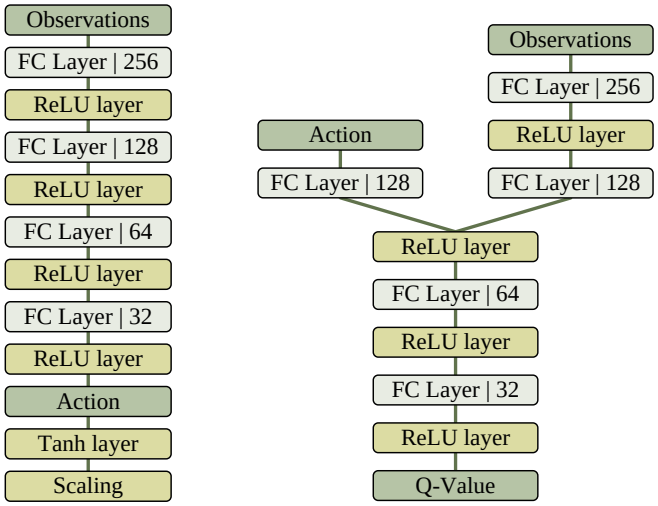


Fig. 4.4 Neural network architectures of: (a) Actor, (b) Critic.

The longitudinal velocity v_x and its tracking error v_e describe forward motion control. Lateral acceleration a_y represents cornering demand and load transfer. Wheel-level dynamics are reflected through the angular velocity ω_{ij} and slip ratio λ_{ij}, which are essential for assessing traction and tyre–road interaction. The total longitudinal force $F_{x,t}$ denotes the overall propulsion demand. This comprehensive state representation provides the agent with complete feedback on the dynamic behaviour of the vehicle, enabling effective torque allocation. The corrective yaw moment generated by the agent serves as the action applied to the environment.

The TD3 agent is trained to simultaneously achieve yaw stability, minimize sideslip angle, limit tyre saturation, and optimize energy efficiency. The total reward at each timestep is defined as a weighted sum of multiple terms:

$$r_t(s_t, a_t) = R_1 + R_2 + R_3 + R_4 + R_5 + R_6 + R_7 \tag{4.9}$$

The yaw rate-related reward components are:

$$R_1 = -w_1 \cdot (\dot{\psi}_e)^2 \tag{4.10}$$

$$R_2 = \begin{cases} -w_2, & \text{if } |\dot{\psi}_e| > |c_2| \\ 0, & \text{otherwise} \end{cases} \tag{4.11}$$

R_1 penalizes the squared yaw rate error, encouraging the agent to follow the desired yaw rate trajectory, while R_2 introduces a fixed penalty if the yaw rate error exceeds a critical limit c_2. The associated weights w_1 and w_2 are set to 30 and 15, respectively.
The sideslip-related reward terms are:

$$R_3 = \begin{cases} -w_3, & \text{if } |\beta| > |\beta_{max}| + c_3 \\ 0, & \text{otherwise} \end{cases} \tag{4.12}$$

$$R_4 = -w_4 \cdot \tanh(k|\beta|^2) \tag{4.13}$$

R_3 applies a discrete penalty when the sideslip angle exceeds a safety threshold β_{max}, enforcing lateral stability. R_4 complements it by providing a smooth, continuous penalty that discourages large sideslip angles without causing discontinuities. Both weights, w_3 and w_4, are set to 5, and k determines the steepness of the penalty curve.
The tyre slip and energy-related rewards are:

$$R_5 = -w_5 \cdot \sum_{ij} \left(\max(0, |\lambda_{ij} - \lambda_{\text{peak}}|)\right)^2 \tag{4.14}$$

$$R_6 = -w_6 \cdot \sum_{ij} P_{ij} = -w_6 \cdot \sum_{ij} \frac{T_{ij}\omega_{ij}}{\eta(T_{ij}, \omega_{ij})} \tag{4.15}$$

R_5 penalizes tyre slip ratios exceeding the optimal grip threshold λ_{peak}, encouraging the tyres to operate below the saturation point. R_6 penalizes total power losses based on motor efficiency, guiding the agent toward energy-efficient torque–speed combinations. The weights w_5 and w_6 are 10 and 20, respectively.
To avoid abrupt torque changes, the final term penalizes control effort variations:

$$R_7 = -\Delta u(t)^T \cdot w_7 \cdot \Delta u(t) \tag{4.16}$$

where $\Delta u(t) = [\Delta T_{fl}, \Delta T_{fr}, \Delta T_{rl}, \Delta T_{rr}]$ represents the torque increments with $\Delta T_{ij} = T_{ij}(k) - T_{ij}(k-1)$, and $w_7 = 0.01$. This term ensures smooth control action, reducing torque oscillations and improving drivability. The multi-objective reward design reflects the trade-off between vehicle stability and energy efficiency.

To evaluate the contribution of each reward term and justify the chosen weights, an ablation study is performed in which each term (w_1 to w_6) is removed individually. The performance of the trained agent is evaluated across nine criteria, summarised in Table 4.4, where the worst-performing metrics are bolded. Fig. 4.5 visualises the normalised criteria using a radar plot. The baseline reward (unablated) achieves the most balanced performance

Table 4.4 Ablation study of reward function based on different evaluation criteria

Ablation term	$max\|\beta\|$ [deg]	$max\|\dot{\beta}\|$ [deg/s]	$\int\|\beta\|$ [deg·s]	$\int\|\dot{\beta}\|$ [deg]	$\int\|\dot{\psi}_e\|$ [deg]	$max\|Y_e\|$ [m]	$max\|\lambda_{ij}\|$ [%]	η_{avg} [%]	P_{tot} [kJ]
Unablated	0.205	0.594	0.392	1.095	2.979	0.274	0.502	79.66	210.0
w_1	**0.496**	**1.041**	**1.075**	**1.941**	**7.426**	**4.309**	0.241	86.33	205.3
w_2	0.241	0.635	0.500	1.235	3.415	0.425	0.355	81.56	203.7
w_3	0.268	0.670	0.519	1.311	3.435	0.514	0.336	83.28	208.2
w_4	0.276	0.705	0.517	1.365	3.386	0.298	0.304	80.52	209.0
w_5	0.202	0.604	0.320	1.106	3.079	0.295	**0.865**	78.70	209.7
w_6	0.196	0.570	0.347	0.973	2.832	0.242	0.481	**72.41**	**222.9**

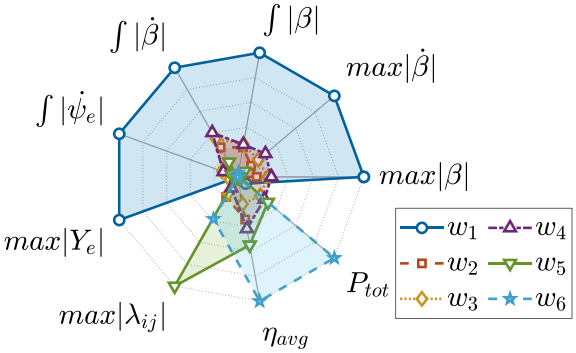


Fig. 4.5 Radar plot of normalized performance metrics under reward ablation analysis.

across stability and efficiency objectives. Removing w_1 (yaw rate term) leads to significant degradation in lateral stability, with higher sideslip and yaw rate errors. Excluding R_2, R_3, or R_4 also reduces stability performance. Removing w_5 (tyre slip penalty) increases $\max|\lambda_{ij}|$ to 0.865%, indicating weaker traction control. Eliminating w_6 (energy term) slightly improves stability but worsens efficiency, with η_{avg} dropping to 72.41% and P_{tot} rising to 222.9 kJ. These results confirm that each term contributes to the overall control objectives and that the selected weights yield a well-balanced performance.

4.4 Simulation-Based Evaluation and Comparative Analysis

The effectiveness of the proposed RL-based torque vectoring strategies is evaluated through simulation studies under multiple driving manoeuvres. Each manoeuvre is designed to test different aspects of vehicle dynamics and energy management. The simulation environment incorporates varying road surface conditions and low-friction scenarios to emulate real-

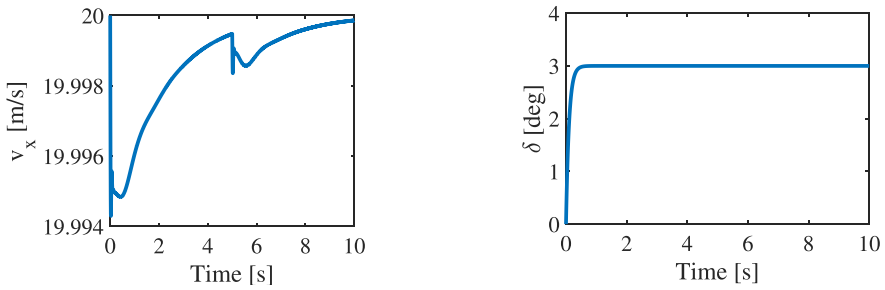


Fig. 4.6 Inputs set by the driver for circular turning manoeuvre: (a) Longitudinal velocity, (b) Steering wheel angle.

world driving uncertainties. Three RL-based controllers of DDPG, TD3, and CL TD3 are benchmarked against two model-based baselines, namely LSQP and SSQP. To ensure a fair comparison, the magnitude of control actions is constrained to maintain similar torque levels across all controllers. Additionally, real-time feasibility is assessed by measuring the average inference time per control step for each control strategy.

The LSQP and SSQP baselines rely on an iterative, offline SQP optimization, resulting in higher computational demands. Their average inference times per control step are 0.0465 ms and 0.0480 ms, respectively. In contrast, the RL-based controllers operate fully online during deployment, requiring only a forward pass through the trained actor network, which significantly reduces computational cost. The DDPG controller achieves an average inference time of 0.0292 ms per control step, while TD3 and CL TD3 require 0.0299 ms and 0.0307 ms, respectively. These results correspond to computational cost reductions of 37.2%, 35.6%, and 33.9% compared to the LSQP controller, and 39.1%, 37.7%, and 36.0% relative to the SSQP controller, for DDPG, TD3, and CL TD3, respectively. The findings confirm that the proposed RL-based controllers not only enhance stability and efficiency but also meet the real-time computational requirements for practical torque vectoring applications in AWD EVs.

4.4.1 Circular Turning Manoeuvre

The circular turning manoeuvre is simulated at a vehicle velocity of 20 m/s (72 km/h), as depicted in Fig. 4.6a. The vehicle travels in a circular path with a steering wheel angle shown in Fig. 4.6b. During the first 5 seconds of the test, the road surface exhibits a low friction coefficient $\mu = 0.3$, which then increases to $\mu = 0.5$ for the remaining duration. This transition from a road with a snowy surface to wet asphalt introduces a disturbance that requires adaptive torque distribution to maintain yaw stability and energy efficiency. The

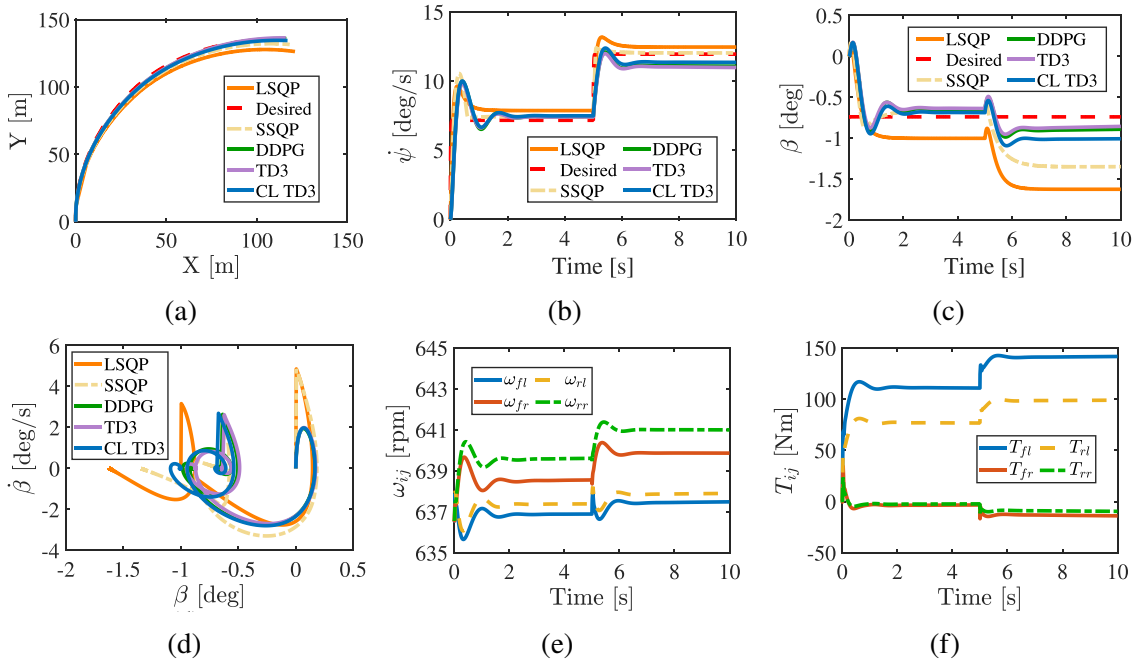
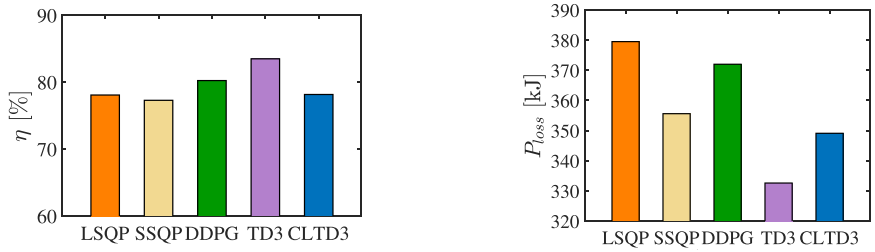


Fig. 4.7 Simulation results under circular turning manoeuvre: (a) Vehicle trajectory, (b) Yaw rate, (c) Sideslip angle, (d) Sideslip angle versus sideslip angle rate, (e) Wheel angular velocities of CL TD3, (f) Applied wheel torques of CL TD3.

simulation results are presented in Fig. 4.7 and Fig. 4.8 for the vehicle dynamics and energy optimization, respectively.

All controllers maintain tyre slip ratios below 2%, indicating stable traction behaviour. Among them, CL TD3 achieves the lowest maximum slip ratio of 1.067%. The vehicle trajectory under the circular turning manoeuvre is illustrated in Fig. 4.7a for different controllers. The TD3-based controller achieves the minimum lateral deviation from the desired path ($\max|Y_e|$), measured at 1.485 m, which represents a reduction of approximately 74.2% and 51.7% compared to the LSQP and SSQP controllers, respectively. Other RL-based controllers, i.e., DDPG and CL TD3, also show better performance than the model-based controller, each reduced $\max|Y_e|$ by 71.5% and 64.1% compared to the LSQP controller, and 46.7% and 32.7% compared to SSQP, respectively. The yaw rates of LSQP, SSQP, DDPG, TD3, and CL TD3 are compared in Fig. 4.7b. The sideslip angles are presented in Fig. 4.7c, and the sideslip angles versus the sideslip angle rates are shown in Fig. 4.7d. TD3 achieves the best performance among all controllers in terms of the maximum of absolute sideslip angle ($max|\beta|$), the maximum of absolute sideslip angle rate ($max|\dot{\beta}|$), and the integral of absolute sideslip angle ($\int|\beta|$). Finally, the angular velocities and the applied torques to wheels for the CL TD3 are shown in Fig. 4.7e and Fig. 4.7f, respectively. The CL TD3 agent generates an optimal corrective yaw moment and allocates optimal torques to maintain the

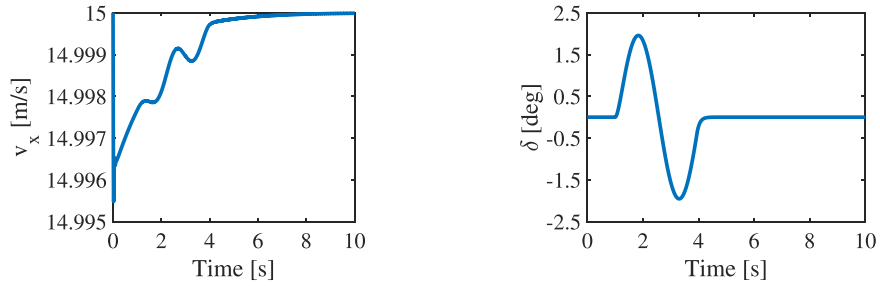


(a) (b)

Fig. 4.8 Energy optimization results under circular turning manoeuvre: (a) Average efficiency, (b) Total power consumption.

yaw stability of the AWD EV under the circular turning manoeuvre. As can be seen, the torques applied to the wheels are adjusted when the vehicle enters a road with a different friction coefficient to meet the stability and manoeuvrability requirements.

The average efficiencies of the four electric machines and the total power consumed are reported in Fig. 4.8a and Fig. 4.8b, respectively. Among the controllers, the TD3-based algorithm achieves the highest average efficiency of 83.50%, compared to 78.09% and 77.30% obtained by the LSQP and SSQP controllers, respectively, with the SQP-based energy optimization. In terms of energy consumption, TD3 and CL TD3 outperform the other strategies, reducing total power usage by 12.3% and 8.0% compared to LSQP, and by 6.4% and 1.8% compared to SSQP, respectively. The power consumption of CL TD3 for the front-left, front-right, rear-left, and rear-right electric machines is 138.9 kJ, 54.2 kJ, 103.1 kJ, and 52.8 kJ, respectively. This distribution reflects the torque allocation strategy under circular turning, where higher torques are assigned to the front-left and rear-left wheels to maintain vehicle stability, resulting in correspondingly higher power consumption by their associated electric machines.



(a) (b)

Fig. 4.9 Inputs set by the driver for single-lane change manoeuvre: (a) Longitudinal velocity, (b) Steering wheel angle.

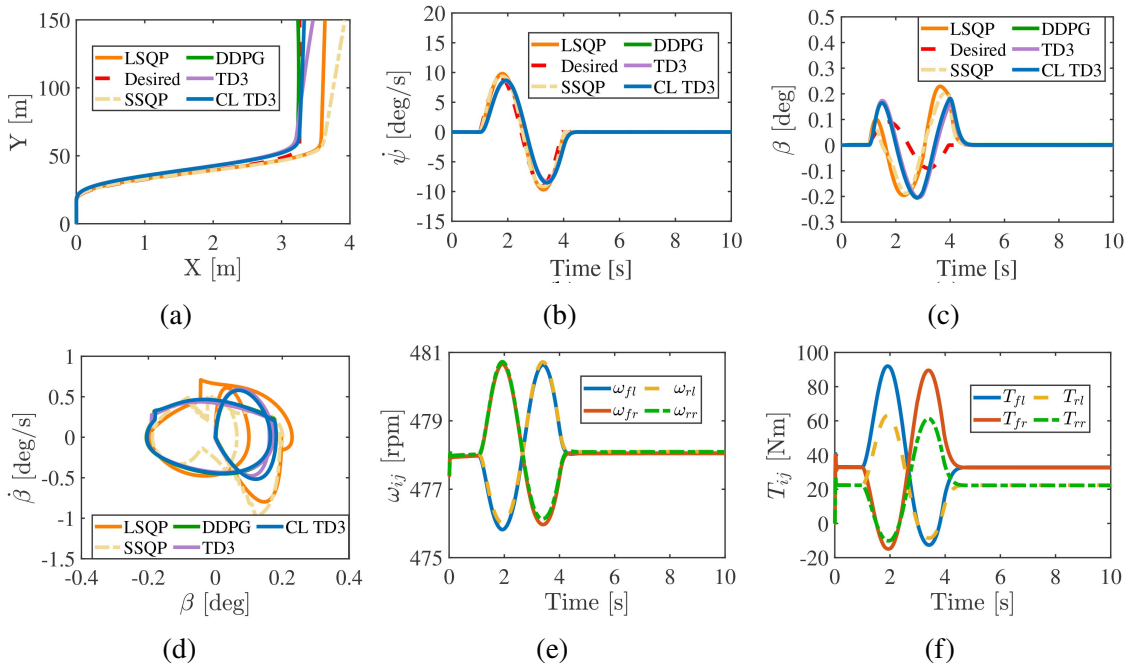
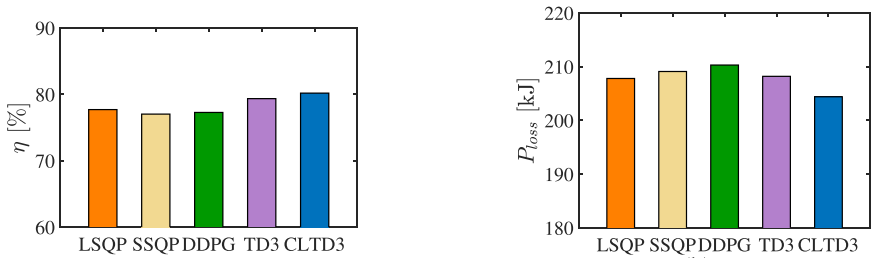


Fig. 4.10 Simulation results under single-lane change manoeuvre: (a) Vehicle Trajectory, (b) Yaw rate, (c) Sideslip angle, (d) Sideslip angle versus sideslip angle rate, (e) Wheel angular velocities of CL TD3, (f) Applied wheel torques of CL TD3.

4.4.2 Single-Lane Change Manoeuvre

To further study the performance of the proposed RL-based controllers against the model-based LSQP and SSQP, a single-lane change manoeuvre is executed at a velocity of 15 m/s (54 km/h). Inputs set by the driver to model the single-lane change manoeuvre are shown in Fig. 4.9. Fig. 4.9a illustrates the longitudinal velocity of the vehicle, showing that it successfully tracks the desired velocity set by the driver. The steering wheel angle is also demonstrated in Fig. 4.9b. The road surface has a friction coefficient of $\mu = 0.6$ from $t = 0$ to $t = 3$ seconds and then reduces to $\mu = 0.4$ to simulate sudden surface degradation. This manoeuvre requires fast transient control response and precise torque vectoring to follow the desired trajectory while mitigating instability. The vehicle response under the single-lane change manoeuvre is illustrated in Fig. 4.10, where the performance of the proposed RL-based controllers (DDPG, TD3, and CL TD3) is compared with the LSQP and SSQP methods.

One of the objectives of the reward function in RL-based algorithms is the minimization of the slip ratio of tyres. Under the single-lane change scenario, all controllers offer slip ratios less than 1%, showing the outstanding performance of the controllers in traction with



(a) (b)

Fig. 4.11 Energy optimization results under single-lane change manoeuvre: (a) Average efficiency, (b) Total power consumption.

minimal slip. The maximum slip ratio of tyres is 0.503% for LSQP controller, 0.564% for SSQP, 0.472% for DDPG, 0.374% for TD3, and 0.464% for CL TD3.

Fig. 4.10a shows the trajectory of the vehicle in the longitudinal-lateral plane. Among all methods, the CL TD3-based controller achieves the smallest lateral deviation from the desired path, demonstrating precise path-tracking capability. The maximum of absolute lateral deviation from the desired path ($max|Y_e|$) is 0.205 m for the CL TD3-based controller, which is decreased by 44.6%, 68.9%, 7.6%, and 11.6%, compared to LSQP, SSQP, DDPG, and TD3 controllers, respectively. Fig. 4.10b presents the yaw rate responses of different controllers under the single-lane change scenario. As depicted, all controllers try to follow the desired yaw rate with a small error. The sideslip angle versus the sideslip angle rate in Fig. 4.10c highlights the capability of each controller in managing lateral stability. The maximum of absolute sideslip angles ($max|\beta|$) is 0.229 deg for LSQP, 0.201 deg for SSQP, 0.206 deg for DDPG, 0.204 deg for TD3, and again 0.204 deg for CL TD3 controllers. Fig. 4.10d plots the sideslip angle versus the sideslip angle rate. Among all controllers, the RL-based algorithms achieve lower values of $max|\dot{\beta}|$ compared to LSQP and SSQP. In particular, CL TD3 offers the minimum value of 0.579 deg/s, which is decreased by 27.4% compared to LSQP, 40.3% compared to SSQP, 2.0% compared to DDPG, and 2.8% compared to TD3. Fig. 4.10e shows the angular velocities of four wheels of the vehicle with the CL TD3 under the single-change scenario. The wheel angular velocities remain within a range, varying between 475 rpm and 481 rpm. Finally, the wheel torques in Fig. 4.10f reveal how torque is redistributed in real-time to achieve the desired yaw moment for the CL TD3. The maximum torques applied to front left, front right, rear left, and rear right wheels are 92.0 Nm, 89.3 Nm, 62.9 Nm, and 61.3 Nm, respectively.

The energy optimization results for the single-lane change manoeuvre are shown in Fig. 4.11. The average efficiencies are provided in Fig. 4.11a, where the CL TD3 achieves the highest average efficiency of 80.18%, followed by TD3 at 79.35%. Fig. 4.11b presents

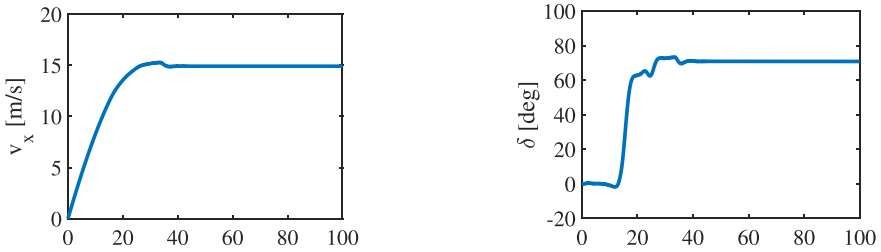


Fig. 4.12 Inputs set by the driver for constant radius circular manoeuvre in IPG CarMaker: (a) Longitudinal velocity, (b) Steering wheel angle.

the total energy consumption of the electric machines. Among the compared controllers, CL TD3 obtains the lowest total power consumption at 204.4 kJ. In comparison, the LSQP, SSQP, DDPG, and TD3 controllers result in energy consumption levels of 207.8 kJ, 209.1 kJ, 210.3 kJ, and 208.2 kJ, respectively.

4.4.3 Constant Radius Circular Manoeuvre (ISO 4138:2021) in IPG CarMaker

To further evaluate the performance of the proposed RL approach under a high-fidelity environment, IPG CarMaker is employed as a vehicle dynamics simulator. For this aim, the CL TD3 agent is tested in IPG CarMaker as the proposed controller with the strongest trade-off between stability and energy optimization, fastest convergence, and improved training stability. The other RL-based methods demonstrate similar performance trends, and therefore CL TD3 is selected as a representative algorithm for this extended verification, and its performance is benchmarked against model-based LSQP and SSQP controllers.

In this scenario, the vehicle is commanded to follow a circular trajectory with a radius of 42 m at a steady longitudinal speed of 15 m/s, as shown in Fig. 4.12. This manoeuvre places high demands on lateral stability, requiring precise torque vectoring to follow the desired trajectory. The results are shown in Fig. 4.13. The CL TD3 controller can maintain yaw stability and minimize sideslip throughout the manoeuvre. CL TD3 achieves the best performance in reducing the maximum of slip ratio $max|\lambda_{ij}|$, with reductions of 57.0% and 15.9% compared to LSQP and SSQP, respectively. CL TD3 reduces the maximum of sideslip angle $max|\beta|$ by 22.2% compared to SSQP. Although LSQP achieves slightly lower $max|\beta|$, CL TD3 offers better overall performance by combining stability with improved energy efficiency. The energy optimization results are demonstrated in Fig. 4.14. The proposed CL

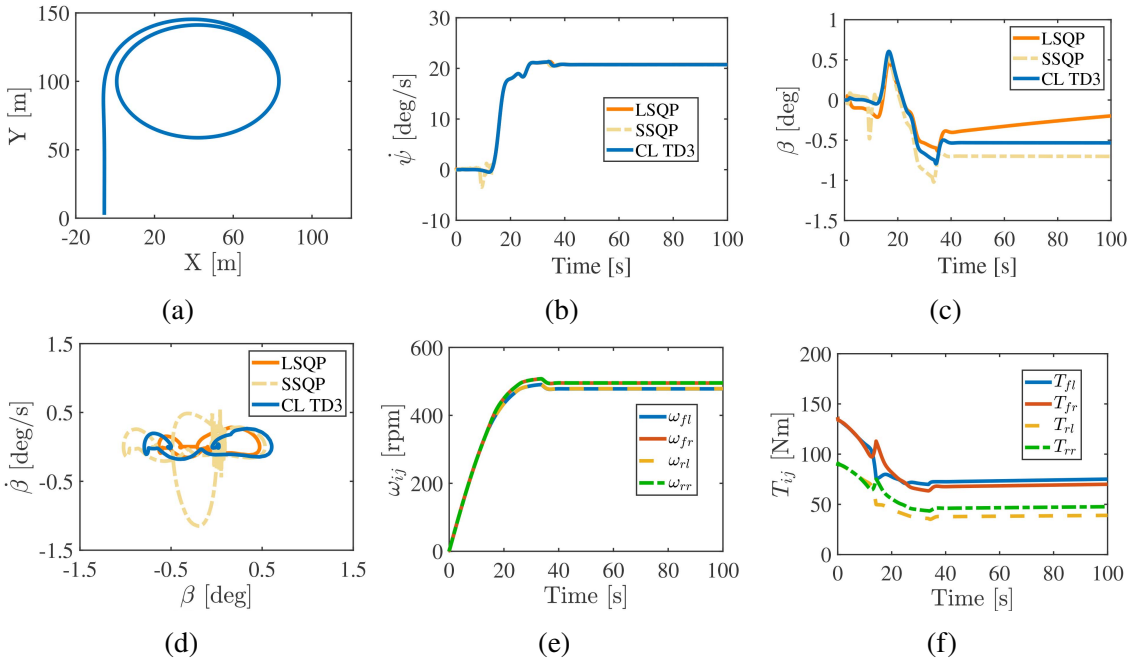


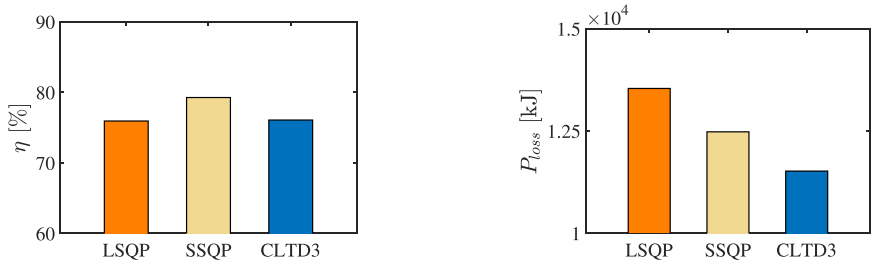
Fig. 4.13 Simulation results under constant radius circular manoeuvre in IPG CarMaker: (a) Vehicle Trajectory, (b) Yaw rate, (c) Sideslip angle, (d) Sideslip angle versus sideslip angle rate, (e) Wheel angular velocities of CL TD3, (f) Applied wheel torques of CL TD3.

TD3 reduces power consumption by 14.9% compared to LSQP and by 7.6% compared to SSQP.

4.4.4 Expressway (S-Curve) Manoeuvre in IPG CarMaker

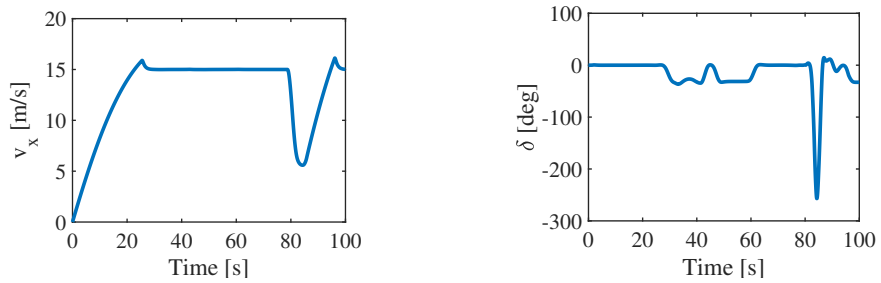
To assess the performance under real-road conditions, an expressway S-curve trajectory is selected in IPG CarMaker. The vehicle is driven at a longitudinal velocity of 15 m/s (54 km/h), representative of typical highway operation. For longitudinal dynamics, the IPGDriver is used to maintain the target velocity during the manoeuvre. The commanded longitudinal velocity and steering angle are shown in Fig. 4.15.

The vehicle response is shown in Fig. 4.16. The trajectory plot in Fig. 4.16a confirms that the vehicle successfully follows the allocated expressway S-curve route, maintaining path tracking throughout the manoeuvre. The yaw-rate for LSQP, SSQP, and CL TD3 are shown in Fig. 4.16b. The plots of sideslip angle and sideslip angle versus sideslip angle rate are depicted in Fig. 4.16c and Fig. 4.16d, respectively. The integral of absolute sideslip angle ($\int |\beta|$) is 58.81 deg.s for LSQP, 26.25 deg.s for SSQP, and 24.88 deg.s for CL TD3. In this regard, CL TD3 achieves the best performance, with a reduction of 57.6% compared to LSQP and 5.2% compared to SSQP. Fig. 4.16e presents the wheel angular velocities, and



(a) (b)

Fig. 4.14 Energy optimization results under constant radius circular manoeuvre in IPG CarMaker: (a) Average efficiency, (b) Total power consumption.



(a) (b)

Fig. 4.15 Inputs set by the driver for expressway manoeuvre in IPG CarMaker: (a) Longitudinal velocity, (b) Steering wheel angle.

Fig. 4.16f illustrates the applied wheel torques of CL TD3. The slip ratio is 2.649% for LSQP, 2.241% for SSQP, and 1.642% for CL TD3, with CL TD3 achieving the best performance in minimizing slip ratio.

Energy results are provided in Fig. 4.17. A higher average machine efficiency and a lower total power loss are obtained with CL-TD3 relative to LSQP and SSQP over the same manoeuvre duration, demonstrating that the learned policy can exploit the efficiency map while preserving stability during fast direction changes. CL TD3 reduces the power consumption by 12.2% and 15.1% compared to LSQP and SSQP, respectively, while maintaining stability. This confirms the improved performance of the proposed CL TD3 under critical manoeuvres.

4.4.5 Results and Discussion

The performance and effectiveness of the RL-based controllers are assessed against the model-based LSQP and SSQP controllers based on nine previously mentioned evaluation metrics under different velocities and tyre-road friction coefficients. The results are tabulated in Table 4.5, comparing the model-based LSQP and SSQP controllers with the DDPG, TD3,

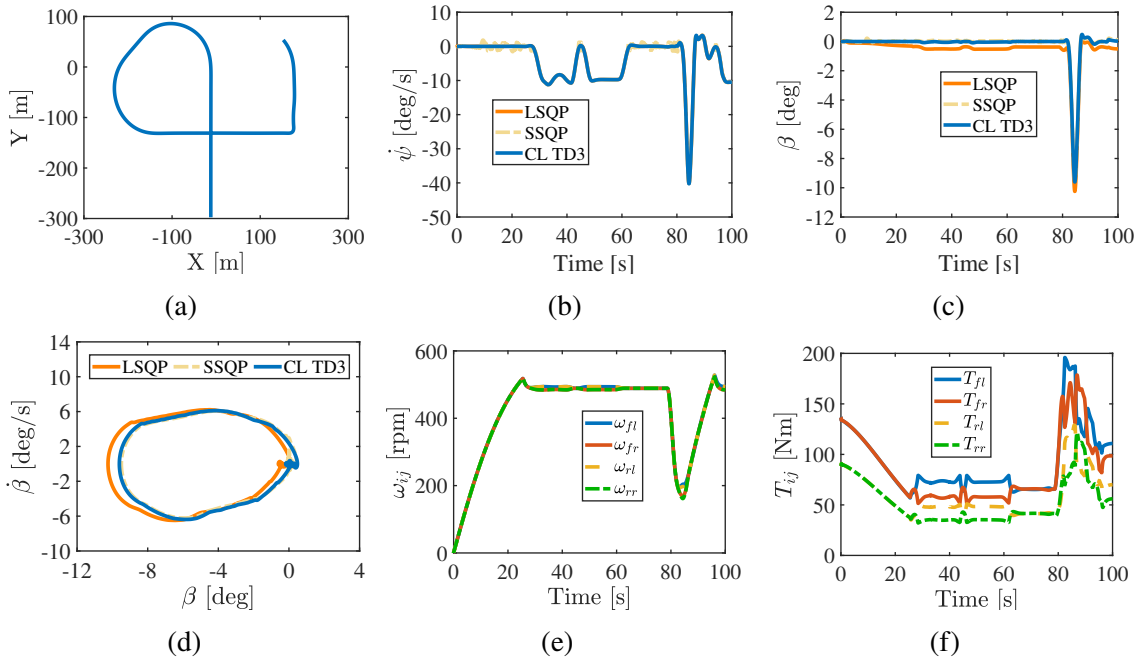


Fig. 4.16 Simulation results under expressway manoeuvre in IPG CarMaker: (a) Vehicle Trajectory, (b) Yaw rate, (c) Sideslip angle, (d) Sideslip angle versus sideslip angle rate, (e) Wheel angular velocities of CL TD3, (f) Applied wheel torques of CL TD3.

and CL TD3 algorithms. The best-performing values in each category are shown in green for clarity. The simulations are collected for circular turning and single-lane change manoeuvres at velocities of 15 m/s, 20 m/s, and 25 m/s, and tyre-friction coefficients of 0.4, 0.5, and 0.6.

Under the circular turning manoeuvre at a velocity of 15 m/s and a friction coefficient of 0.4, the TD3 and CL TD3 controllers demonstrate superior performance compared to DDPG across most evaluation metrics. Although the LSQP and SSQP achieve better performance in terms of minimizing yaw rate error and lateral deviation from the desired path, both TD3 and CL TD3 outperform them in key dynamic metrics such as sideslip angle and slip ratio. At 20 m/s and $\mu = 0.5$, TD3 and CL TD3 also demonstrate improvements in sideslip angle metrics relative to LSQP, SSQP, and DDPG. CL TD3 achieves the lowest maximum slip ratio of 0.717, representing reductions of 35.1%, 26.7%, 9.2%, and 15.8% compared to LSQP, SSQP, DDPG, and TD3, respectively. At 25 m/s and $\mu = 0.6$, TD3 exhibits the lowest values for $max|\beta|$, $max|\dot{\beta}|$, $\int |\beta|$, and $\int |\dot{\beta}|$. Despite this, SSQP achieves the maximum average efficiency, and LSQP obtains the minimum energy consumption under this condition. This observation highlights the key strength of the RL-based controllers to prioritize stability objectives in high-speed scenarios where maintaining vehicle control is critical. In this instance, the agent adaptively sacrifices minor energy efficiency to preserve dynamic stability, demonstrating the context-awareness of policy in decision-making capabilities.

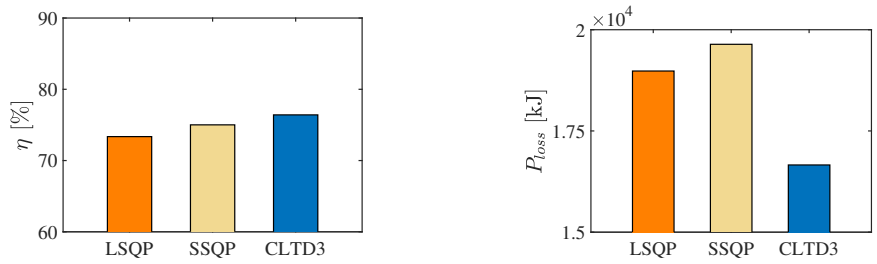


Fig. 4.17 Energy optimization results under expressway manoeuvre in IPG CarMaker: (a) Average efficiency, (b) Total power consumption.

Under the single-lane change manoeuvres, the RL-based controllers exhibit favorable performance across a range of velocities and friction conditions. At 15 m/s and $\mu = 0.4$, CL TD3 achieves the best overall stability metrics, including the lowest values for $max|\beta|$, $max|\dot{\beta}|$, $\int |\beta|$, $\int |\dot{\beta}|$, and $max|Y_e|$, while maintaining competitive energy consumption. At 20 m/s and $\mu = 0.5$, both TD3 and CL TD3 offer strong trade-offs between stability and efficiency, with CL TD3 achieving the lowest $\int |\beta|$ and highest average efficiency of 84.83%. At 25 m/s and $\mu = 0.6$, CL TD3 continues to perform well in stability-related metrics while also achieving the lowest total power consumption of 396.9 kJ. These results further highlight the adaptability of the RL-based controllers, which dynamically balance stability and energy efficiency depending on the driving context.

Whereas the proposed RL-based controllers generally deliver strong performance across most metrics, certain inconsistencies can be observed in Table 4.5 depending on the operating conditions. These variations are primarily the result of context-dependent trade-offs between stability and energy objectives, which are inherent in the multi-objective control problem. Generally, under high-speed or low-friction scenarios, the RL agents tend to prioritize vehicle stability at the expense of energy efficiency, while still maintaining energy-related performance comparable to LSQP and SSQP. This adaptive behavior reflects the flexibility of the learned policies to optimize control actions according to the environment, confirming the practicality of RL-based approaches under varying driving scenarios. The use of MATLAB/Simulink and IPG CarMaker enables comprehensive evaluation of vehicle dynamics and energy performance under realistic driving conditions.

Table 4.5 Performance comparison of controllers under different velocities and tyre–road friction coefficients

Manoeuvre	v [m/s]	μ	Controller	$max\|\beta\|$ [deg]	$max\|\dot{\beta}\|$ [deg/s]	$\int\|\beta\|$ [deg·s]	$\int\|\dot{\beta}\|$ [deg]	$\int\|\dot{\psi}_e\|$ [deg/s]	$max\|Y_e\|$ [m]	$max\|\lambda_{ij}\|$ [%]	η_{avg} [%]	P_{tot} [kJ]
Circular turning	15	0.4	LSQP	0.693	8.201	6.622	1.338	4.335	0.885	1.035	83.66	334.6
			SSQP	0.559	7.218	5.339	1.481	3.388	1.576	1.526	80.27	319.4
			DDPG	0.388	3.137	1.162	1.018	15.240	5.894	0.749	83.86	314.5
			TD3	0.389	3.140	0.725	1.005	17.220	6.463	0.811	85.42	309.1
			CL TD3	0.378	3.112	1.699	1.007	12.910	5.236	0.681	84.03	292.1
Circular turning	20	0.5	LSQP	0.512	9.498	4.938	1.224	6.165	1.248	1.105	83.33	295.8
			SSQP	0.772	7.112	6.840	1.584	7.794	1.751	0.979	84.09	326.5
			DDPG	0.404	3.228	0.894	0.975	14.520	4.958	0.790	84.67	326.5
			TD3	0.406	3.231	0.495	0.966	16.760	5.507	0.852	86.02	338.0
			CL TD3	0.395	3.199	1.406	0.970	11.730	4.279	0.717	85.04	301.2
Circular turning	25	0.6	LSQP	1.917	7.803	18.360	2.316	8.318	1.042	1.684	88.91	389.2
			SSQP	1.887	6.511	16.260	2.026	8.072	1.095	1.609	89.64	409.3
			DDPG	1.055	3.120	8.686	1.698	22.330	8.769	1.333	87.58	459.9
			TD3	1.014	3.054	8.223	1.674	23.860	9.215	1.387	88.31	480.5
			CL TD3	1.176	3.203	10.130	1.747	17.870	7.282	1.204	85.16	404.8
Lane change	15	0.4	LSQP	0.238	0.798	0.449	1.132	2.495	0.266	0.373	77.40	208.4
			SSQP	0.205	0.871	0.391	1.113	1.935	0.294	0.403	79.94	212.3
			DDPG	0.206	0.588	0.406	1.110	3.078	0.320	0.472	77.23	208.1
			TD3	0.205	0.594	0.392	1.095	2.979	0.274	0.502	79.66	210.0
			CL TD3	0.204	0.576	0.380	1.073	2.943	0.243	0.463	80.14	207.6
Lane change	20	0.5	LSQP	1.017	2.075	1.823	4.084	2.712	0.418	0.650	82.48	289.6
			SSQP	0.964	1.732	1.859	3.923	2.938	0.751	0.669	81.11	283.5
			DDPG	0.679	1.412	1.300	2.789	3.525	1.047	0.728	82.14	292.7
			TD3	0.634	1.448	1.323	2.887	3.351	0.524	0.774	84.63	295.4
			CL TD3	0.696	1.303	1.181	2.616	3.335	0.565	0.714	84.83	291.4
Lane change	25	0.6	LSQP	1.706	3.291	3.168	6.788	2.642	0.502	0.986	85.19	398.6
			SSQP	1.715	3.031	3.357	6.900	3.612	0.848	1.062	85.49	397.6
			DDPG	1.227	2.554	2.361	4.874	3.239	2.085	1.086	84.33	399.9
			TD3	1.260	2.667	2.431	5.098	3.041	0.801	1.141	86.95	404.1
			CL TD3	1.148	2.414	2.167	4.616	2.949	1.002	1.066	86.69	396.9

4.5 Conclusion

This chapter presents an integrated RL-based control framework for torque vectoring and energy optimisation in AWD EVs. The proposed multi-objective control architecture employs three actor–critic algorithms, namely DDPG, TD3, and curriculum learning-enhanced TD3 (CL TD3), to improve vehicle stability, manoeuvrability, and energy efficiency simultaneously. The RL agents learn optimal control policies directly through interaction with the vehicle environment, removing the need for explicit system modelling or manual rule-based tuning. The reward function integrates dynamic and energy-related objectives, including yaw rate error, sideslip angle, tyre slip ratio, and electric machine efficiency, enabling the controller to balance vehicle stability and energy consumption in real time. Simulation results demonstrate that the RL-based controllers outperform the model-based LSQP and SSQP benchmarks across several stability and energy-related performance metrics. In particular, the TD3 and CL TD3 algorithms achieve improved yaw rate tracking, reduced sideslip and slip ratios, and enhanced motor efficiency with lower total power consumption. Furthermore, the computational analysis confirms the real-time feasibility of the proposed RL controllers, as they require significantly lower inference time than SQP-based optimisation methods.

Chapter 5

Conclusion and Future Work

5.1 Overview

This thesis presents the development of advanced RL-based control frameworks for torque vectoring and energy optimization in AWD EVs. The primary objective is to enhance vehicle dynamic stability while improving energy efficiency through intelligent, model-free control strategies capable of operating under nonlinear and uncertain conditions. A comprehensive nonlinear vehicle model is employed to represent the real dynamic behaviour of an AWD EV. The model incorporates a detailed tyre model based on the Pacejka Magic Formula to accurately capture tyre–road interactions. The benchmark control architectures are designed using conventional model-based controllers combined with Sequential Quadratic Programming (SQP) for energy optimization. These methods serve as reliable baselines for evaluating the performance of the proposed RL-based approaches.

Overall, the proposed RL-based control strategies demonstrate a strong capability to enhance vehicle stability and energy efficiency simultaneously. The simulation results show that the RL-based controllers consistently reduce sideslip angle, slip ratio, and lateral path deviation compared with conventional model-based LSQP and SSQP controllers across a wide range of vehicle speeds and tyre–road friction conditions. In addition, the TD3 and CL TD3 algorithms improve average electric machine efficiency while reducing total power consumption in several manoeuvres. The results also confirm the real-time feasibility of the proposed controllers, as the RL-based strategies require significantly lower computational effort than optimisation-based approaches during deployment. These findings demonstrate that reinforcement learning provides an effective and scalable alternative to conventional model-based torque vectoring strategies for AWD EVs operating under nonlinear and uncertain driving conditions.

5.2 Summary of Key Contributions and Limitations

The primary contributions of this research are summarised in this section. A hierarchical control architecture is designed to separate the control task into two layers. The high-level controller generates the corrective yaw moment required for stability, while the low-level allocator distributes the torque among the four wheels. This structure ensures modularity and allows the seamless integration of learning-based and optimisation-based strategies.

A model-free RL control strategy is developed for torque vectoring and yaw stability enhancement. Three RL algorithms of DDPG, TD3, and CL TD3 are implemented and compared against model-based benchmark controllers, including LSQP and SSQP. The DDPG algorithm demonstrates the feasibility of applying actor–critic learning to vehicle dynamics. The TD3 algorithm improves training stability and performance by introducing twin critics, target-policy smoothing, and delayed policy updates. The curriculum learning-enhanced TD3 further improves convergence and policy robustness by gradually increasing task complexity during training. Simulation results confirm that RL-based controllers outperform conventional model-based methods across multiple stability metrics.

The RL framework is extended to incorporate energy efficiency as a primary control objective. The developed reward function includes both dynamic and energy-related components, such as yaw rate error, sideslip angle, tyre slip ratios, and electric machine efficiency. This design enables the controller to balance stability and power consumption objectives dynamically. The framework integrates electric machine efficiency maps into the control process, allowing real-time torque allocation in high-efficiency operating regions. Simulation results in MATLAB/Simulink and IPG CarMaker demonstrate that the proposed approach achieves reductions in energy consumption while improving stability performance. Extensive simulations under various manoeuvres are conducted to evaluate controller performance under various road adhesion conditions and vehicle velocities. The results verify that RL-based methods deliver smoother control responses, lower sideslip angles, and improved trajectory tracking compared with model-based controllers. The computational analysis confirms the real-time feasibility of the proposed methods, well within the requirements for onboard vehicle control systems. The verification of the CL TD3 controller in IPG CarMaker confirms its adaptability and robustness in high-fidelity environments.

Even though the proposed frameworks achieve promising results, some limitations remain. The RL agents are trained in simulation environments, and although verified in high-fidelity software, transfer to physical hardware introduces modelling discrepancies and sensor uncertainties. The learning process relies on extensive offline training, which may require further optimisation to meet industrial time constraints. The framework currently assumes ideal communication between controllers and actuators, excluding potential delays

or noise that may occur in practical systems. The obtained results demonstrate that the RL-based torque vectoring framework can simultaneously improve vehicle stability and energy efficiency compared with conventional optimisation-based controllers. In particular, the TD3 and CL TD3 algorithms consistently achieve lower sideslip angles, reduced tyre slip ratios, and improved path tracking across different manoeuvres and road conditions. Furthermore, the integration of electric machine efficiency maps within the reward function enables the RL agents to allocate wheel torques in energy-efficient operating regions without compromising vehicle stability. These findings confirm that reinforcement learning provides an effective multi-objective control strategy capable of balancing dynamic performance and energy consumption in AWD electric vehicles.

5.3 Future Work

Building on the outcomes of this research, several directions can be explored to further develop the proposed RL-based torque vectoring framework and bring it closer to practical deployment. The next stages of research mainly involve experimental validation, safety-oriented learning strategies, and integration with higher-level vehicle automation systems.

5.3.1 Hardware-in-the-Loop (HIL) and Real-Time Implementation

A key next step is the implementation of the proposed RL controllers in a hardware-in-the-loop (HIL) environment. While the present study demonstrates the effectiveness of the approach through high-fidelity simulations, experimental validation is necessary to evaluate the controller under realistic embedded computing conditions. An HIL platform allows the control algorithms to be executed on automotive-grade processors while interacting with a real-time vehicle dynamics simulator. Such a setup enables the investigation of practical implementation aspects including computational latency, actuator saturation, communication delays between control units, and sensor noise. In addition, it provides an opportunity to evaluate the robustness of the learned policies when subject to real-time constraints and limited computational resources. The insights gained from HIL testing will help identify potential modifications required for embedded implementation and will represent an important intermediate step toward full vehicle testing.

5.3.2 Safe and Explainable Reinforcement Learning

Although RL provides powerful capabilities for handling nonlinear and uncertain vehicle dynamics, safety guarantees remain an important requirement for real-world automotive

applications. Future work will therefore focus on developing safety-aware RL strategies that explicitly incorporate system constraints during learning and deployment.

One possible direction is the integration of constrained RL formulations, where safety limits related to vehicle stability, tyre forces, or actuator bounds are enforced during policy optimisation. Alternatively, safety filters or control barrier functions can be used as supervisory layers to ensure that the control actions generated by the RL agent remain within safe operational limits. These mechanisms can enhance the reliability of RL-based controllers while preserving their adaptability. Another important aspect concerns the interpretability of RL policies. Explainable RL techniques could be employed to better understand how the learned controller makes decisions under different driving conditions. Improving the transparency of the learning process is particularly valuable for safety-critical automotive systems, as it facilitates system verification, validation, and certification.

5.3.3 Integration with Motion Planning and Perception

Another promising direction is the integration of the proposed torque vectoring framework with higher-level motion planning and perception modules. In modern intelligent vehicles, vehicle control operates within a hierarchical architecture that includes perception, decision-making, trajectory planning, and low-level actuation. Extending the RL-based controller to interact with these higher-level layers would enable more coordinated and anticipatory vehicle behaviour.

For example, information obtained from perception systems such as road curvature, lane boundaries, traffic conditions, or surface characteristics could be incorporated into the control policy. This additional context would allow the controller to proactively adjust torque distribution before critical manoeuvres occur. Similarly, coupling the torque vectoring controller with trajectory planning algorithms could enable joint optimisation of path tracking, vehicle stability, and energy efficiency. Such an integrated architecture would move the proposed framework closer to real-world autonomous driving systems, where control strategies must operate in a dynamic environment while simultaneously satisfying safety, efficiency, and performance requirements.

5.4 Concluding Remarks

This thesis demonstrates that RL provides a powerful framework for integrated vehicle dynamics control and energy optimisation in AWD EVs. By learning control policies directly from interaction with the environment, the proposed RL-based torque vectoring strategies

are able to handle nonlinear vehicle dynamics and varying road conditions without requiring highly accurate mathematical models.

The results obtained in MATLAB/Simulink and IPG CarMaker confirm that the TD3 and CL TD3 algorithms achieve improved stability performance, reduced tyre slip ratios, and better path tracking compared with conventional model-based controllers. At the same time, the integration of electric machine efficiency maps enables the controller to reduce power consumption and operate motors within more efficient regions. The computational analysis further demonstrates that RL-based controllers are suitable for real-time implementation due to their low inference time during deployment. Generally, this research highlights the potential of reinforcement learning as an enabling technology for intelligent vehicle control systems capable of simultaneously addressing multiple objectives such as stability, energy efficiency, and adaptability. The developed framework provides a foundation for future research on AI-driven vehicle control architectures and contributes to the advancement of safer and more energy-efficient electric mobility systems.

References

[1] J. Pasha, B. Li, Z. Elmi, A. M. Fathollahi-Fard, Y.-y. Lau, A. Roshani, T. Kawasaki, and M. A. Dulebenets, "Electric vehicle scheduling: State of the art, critical challenges, and future research opportunities," *Journal of industrial information integration*, vol. 38, p. 100561, 2024.

[2] A. J. Niri, G. A. Poelzer, S. E. Zhang, J. Rosenkranz, M. Pettersson, and Y. Ghorbani, "Sustainability challenges throughout the electric vehicle battery value chain," *Renewable and Sustainable Energy Reviews*, vol. 191, p. 114176, 2024.

[3] X. Hu, H. Chen, X. Gong, Y. Hu, and P. Wang, "Embedded model predictive control for torque distribution optimization of electric vehicles," *IEEE/ASME Transactions on Mechatronics*, 2024.

[4] N. Amer, M. Dalboni, P. Georgiev, C. Caponio, D. Tavernini, P. Gruber, M. Dhaens, and A. Sorniotti, "Integrated torque-vectoring and anti-roll moment distribution strategies based on optimal control: influence of model complexity and road curvature preview," *Vehicle system dynamics*, vol. 62, no. 10, pp. 2533–2566, 2024.

[5] E. Elghanam, A. Abdelfatah, M. S. Hassan, and A. OSMAN, "Optimization techniques in electric vehicle charging scheduling, routing and spatio-temporal demand coordination: a systematic review," *IEEE Open Journal of Vehicular Technology*, 2024.

[6] C. H. Lee, W. Hua, T. Long, C. Jiang, and L. V. Iyer, "A critical review of emerging technologies for electric and hybrid vehicles," *IEEE Open Journal of Vehicular Technology*, vol. 2, pp. 471–485, 2021.

[7] J. Wang, S. Lv, N. Sun, S. Gao, W. Sun, and Z. Zhou, "Torque vectoring control of RWID electric vehicle for reducing driving-wheel slippage energy dissipation in cornering," *Energies*, vol. 14, no. 23, p. 8143, 2021.

[8] A. Parra, A. Zubizarreta, J. Pérez, and M. Dendaluce, "Intelligent torque vectoring approach for electric vehicles with per-wheel motors," *Complexity*, vol. 2018, no. 1, p. 7030184, 2018.

[9] R. Jafari, P. Sarhadi, A. Paykani, S. S. Refaat, and P. Asef, "Integrated energy optimization and stability control using deep reinforcement learning for an all-wheel-drive electric vehicle," *IEEE Open Journal of Vehicular Technology*, 2025.

[10] ——, "Optimal torque allocation for all-wheel-drive electric vehicles using a reinforcement learning algorithm," in *2024 13th Mediterranean Conference on Embedded Computing (MECO)*. IEEE, 2024, pp. 1–5.

[11] ——, "A td3-based reinforcement learning algorithm with curriculum learning for adaptive yaw stability control in all-wheel-drive electric vehicles," *IEEE Access*, 2025.

[12] D. De Simone, M. S. Carmeli, S. D'Arco, L. Piegari, and P. Tricoli, "Design and control of hybrid electric light rail vehicles with open ended winding machines," *IEEE Open Journal of Vehicular Technology*, vol. 5, pp. 695–703, 2024.

[13] U. Fesli and M. B. Ozdemir, "Electric vehicles: A comprehensive review of technologies, integration, adoption, and optimization," *IEEE Access*, 2024.

[14] R. Zaino, V. Ahmed, A. M. Alhammadi, and M. Alghoush, "Electric vehicle adoption: A comprehensive systematic review of technological, environmental, organizational and policy impacts," *World Electric Vehicle Journal*, vol. 15, no. 8, p. 375, 2024.

[15] M. Rashid, T. Elfouly, and N. Chen, "A comprehensive survey of electric vehicle charging demand forecasting techniques," *IEEE Open Journal of Vehicular Technology*, 2024.

[16] Y. Jiang, J. Guo, D. Xie, Z. Hou, and J. Deng, "Remaining driving range prediction of electric vehicles based on personalized driving behavior in complex traffic scenarios," *IEEE Open Journal of Vehicular Technology*, 2025.

[17] M. Aripin, Y. Md Sam, K. A. Danapalasingam, K. Peng, N. Hamzah, and M. Ismail, "A review of active yaw control system for vehicle handling and stability enhancement," *International journal of vehicular technology*, vol. 2014, no. 1, p. 437515, 2014.

[18] Z. Liang, J. Zhao, Z. Dong, Y. Wang, and Z. Ding, "Torque vectoring and rear-wheel-steering control for vehicle's uncertain slips on soft and slope terrain using sliding mode algorithm," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 3805–3815, 2020.

[19] V. Mazzilli, S. De Pinto, L. Pascali, M. Contrino, F. Bottiglione, G. Mantriota, P. Gruber, and A. Sorniotti, "Integrated chassis control: Classification, analysis and future trends," *Annual Reviews in Control*, vol. 51, pp. 172–205, 2021.

[20] C. Lin, B. Li, E. Siampis, S. Longo, and E. Velenis, "Predictive path-tracking control of an autonomous electric vehicle with various multi-actuation topologies," *Sensors*, vol. 24, no. 5, p. 1566, 2024.

[21] X. Hu, P. Wang, Y. Hu, and H. Chen, "A stability-guaranteed and energy-conserving torque distribution strategy for electric vehicles under extreme conditions," *Applied Energy*, vol. 259, p. 114162, 2020.

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

[23] G. De Filippis, B. Lenzo, A. Sorniotti, P. Gruber, and W. De Nijs, "Energy-efficient torque-vectoring control of electric vehicles with multiple drivetrains," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 4702–4715, 2018.

[24] Z. Li and Y. Wang, "Energy-efficiency optimization and control for electric vehicle platooning with regenerating braking," *IET Intelligent Transport Systems*, vol. 18, no. 2, pp. 203–217, 2024.

[25] H.-W. Kim, A. Amarnathvarma, E. Kim, M.-H. Hwang, K. Kim, H. Kim, I. Choi, and H.-R. Cha, "A novel torque matching strategy for dual motor-based all-wheel-driving electric vehicles," *Energies*, vol. 15, no. 8, p. 2717, 2022.

[26] M. Dalboni, G. Martins, D. Tavernini, U. Montanaro, A. Soldati, C. Concari, M. Dhaens, and A. Sorniotti, "On the energy efficiency potential of multi-actuated electric vehicles," *IEEE Transactions on Vehicular Technology*, 2024.

[27] J. Wu, Z. Song, and C. Lv, "Deep reinforcement learning-based energy-efficient decision-making for autonomous electric vehicle in dynamic traffic environments," *IEEE Transactions on Transportation Electrification*, vol. 10, no. 1, pp. 875–887, 2023.

[28] C. Liu, Z. Wang, Z. Liu, and K. Huang, "Multi-agent reinforcement learning framework for addressing demand-supply imbalance of shared autonomous electric vehicle," *Transportation Research Part E: Logistics and Transportation Review*, vol. 197, p. 104062, 2025.

[29] M. Delamou, A. Naeem, H. Arslan, and E. M. Amhoud, "Joint adaptive OFDM and reinforcement learning design for autonomous vehicles: Leveraging age of updates," *IEEE Open Journal of Vehicular Technology*, 2025.

[30] Y. Wang, J. Wu, H. He, Z. Wei, and F. Sun, "Data-driven energy management for electric vehicles using offline reinforcement learning," *Nature Communications*, vol. 16, no. 1, p. 2835, 2025.

[31] C. Laflamme, J. Doppler, B. Palvolgyi, S. Dominka, Z. J. Viharos, and S. Haeussler, "Explainable reinforcement learning for powertrain control engineering," *Engineering Applications of Artificial Intelligence*, vol. 146, p. 110135, 2025.

[32] C. Liu, H. Liu, L. Han, W. Wang, and C. Guo, "Multi-level coordinated yaw stability control based on sliding mode predictive control for distributed drive electric vehicles under extreme conditions," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 1, pp. 280–296, 2022.

[33] I.-G. Jang, S.-H. You, S.-H. Hwang, and W. Cho, "Lateral stability control of a 4-wheel independent drive electric vehicle using the yaw moment contour line concept," *IEEE Access*, vol. 9, pp. 136 892–136 904, 2021.

[34] S. Zhu, J. Lu, L. Zhu, H. Chen, J. Gao, and W. Xie, "Coordinated control of differential drive-assist steering and direct yaw moment control for distributed-drive electric vehicles," *Electronics*, vol. 13, no. 18, p. 3711, 2024.

[35] Z. Zhu, X. Tang, Y. Qin, Y. Huang, and E. Hashemi, "A survey of lateral stability criterion and control application for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 10, pp. 10 382–10 399, 2023.

[36] X. Sun, Y. Wang, Z. Quan, Y. Cai, L. Chen, and S. Bei, "DYC design for autonomous distributed drive electric vehicle considering tire nonlinear mechanical characteristics in the PWA form," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 10, pp. 11 030–11 046, 2023.

[37] A. Medina, G. Bistue, and A. Rubio, "Comparison of typical controllers for direct yaw moment control applied on an electric race car," *Vehicles*, vol. 3, no. 1, pp. 127–144, 2021.

[38] J. Liang, J. Feng, Y. Lu, G. Yin, W. Zhuang, and X. Mao, "A direct yaw moment control framework through robust TS fuzzy approach considering vehicle stability margin," *IEEE/ASME Transactions on Mechatronics*, vol. 29, no. 1, pp. 166–178, 2023.

[39] J. Hu, Y. Liu, F. Xiao, Z. Lin, and C. Deng, "Coordinated control of active suspension and dyc for four-wheel independent drive electric vehicles based on stability," *Applied Sciences*, vol. 12, no. 22, p. 11768, 2022.

[40] F. Tarhini, R. Talj, and M. Doumiati, "Multi-objective control architecture for an autonomous in-wheel driven electric vehicle," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 11 470–11 476, 2023.

[41] ——, "Driving towards energy efficiency: A novel torque allocation strategy for in-wheel electric vehicles," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2023, pp. 1022–1029.

[42] ——, "Dual-level control architectures for over-actuated autonomous vehicle's stability, path-tracking, and energy economy," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 287–303, 2023.

[43] I. Kobayashi, J. Kuroda, D. Uchino, K. Ogawa, K. Ikeda, T. Kato, A. Endo, M. H. B. Peeie, T. Narita, and H. Kato, "Research on yaw moment control system for race cars using drive and brake torques," *Vehicles*, vol. 5, no. 2, pp. 515–534, 2023.

[44] J. Feng, J. Liang, Y. Lu, W. Zhuang, D. Pi, G. Yin, L. Xu, P. Peng, and C. Zhou, "An integrated control framework for torque vectoring and active suspension system," *Chinese Journal of Mechanical Engineering*, vol. 37, no. 1, p. 10, 2024.

[45] R. Hajiloo, A. Khajepour, A. Kasaiezadeh, S.-K. Chen, and B. Litkouhi, "A model predictive control of electronic limited slip differential and differential braking for improving vehicle yaw stability," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 2, pp. 797–808, 2022.

[46] M. Vignati, E. Sabbioni, and F. Cheli, "A torque vectoring control for enhancing vehicle performance in drifting," *Electronics*, vol. 7, no. 12, p. 394, 2018.

[47] M. Asperti, M. Vignati, and E. Sabbioni, "On torque vectoring control: review and comparison of state-of-the-art approaches," *Machines*, vol. 12, no. 3, p. 160, 2024.

[48] J. Liang, F. Wang, J. Feng, M. Zhao, R. Fang, D. Pi, and G. Yin, "A hierarchical control of independently driven electric vehicles considering handling stability and energy conservation," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 738–751, 2023.

[49] N. Ahmadian, A. Khosravi, and P. Sarhadi, "Driver assistant yaw stability control via integration of AFS and DYC," *Vehicle system dynamics*, vol. 60, no. 5, pp. 1742–1762, 2022.

[50] Z. Zhang, X.-j. Ma, C.-g. Liu, and S.-g. Wei, "Dual-steering mode based on direct yaw moment control for multi-wheel hub motor driven vehicles: Theoretical design and experimental assessment," *Defence Technology*, vol. 18, no. 1, pp. 49–61, 2022.

[51] G. Wang, Y. Liu, S. Li, Y. Tian, N. Zhang, and G. Cui, "New integrated vehicle stability control of active front steering and electronic stability control considering tire force reserve capability," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 3, pp. 2181–2195, 2021.

[52] H. Liu, C. Liu, L. Han, and C. Xiang, "Handling and stability integrated control of AFS and DYC for distributed drive electric vehicles based on risk assessment and prediction," *IEEE transactions on intelligent transportation systems*, vol. 23, no. 12, pp. 23 148–23 163, 2022.

[53] I. Ahmad, X. Ge, and Q.-L. Han, "Decentralized dynamic event-triggered communication and active suspension control of in-wheel motor driven electric vehicles with dynamic damping," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 5, pp. 971–986, 2021.

[54] F. Jia, H. Jing, Z. Liu, and M. Gu, "Cooperative control of yaw and roll motion for in-wheel motor vehicle with semi-active suspension," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 236, no. 1, pp. 3–15, 2022.

[55] W. Cho, J. Suh, and S.-H. You, "Integrated motion control using a semi-active damper system to improve yaw-roll-pitch motion of a vehicle," *IEEE Access*, vol. 9, pp. 52 464–52 473, 2021.

[56] B. Tan, B. Zhang, N. Zhang, Y. Chen, and A. Qin, "Integrated control of electronic stability program and active suspension system using a priority-weighting mechanism," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 237, no. 12, pp. 2857–2871, 2023.

[57] M. Doumiati, A. C. Victorino, A. Charara, and D. Lechner, "Onboard real-time estimation of vehicle lateral tire–road forces and sideslip angle," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 4, pp. 601–614, 2010.

[58] K. Nam, S. Oh, H. Fujimoto, and Y. Hori, "Estimation of sideslip and roll angles of electric vehicles using lateral tire force sensors through rls and kalman filter approaches," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 3, pp. 988–1000, 2012.

[59] K. Nam, H. Fujimoto, and Y. Hori, "Advanced motion control of electric vehicles based on robust lateral tire force control via active front steering," *IEEE/ASME Transactions on mechatronics*, vol. 19, no. 1, pp. 289–299, 2012.

[60] J. Liang, Y. Lu, D. Pi, G. Yin, W. Zhuang, F. Wang, J. Feng, and C. Zhou, "A decentralized cooperative control framework for active steering and active suspension: Multi-agent approach," *IEEE Transactions on Transportation Electrification*, vol. 8, no. 1, pp. 1414–1429, 2021.

[61] C. Jing, H. Shu, R. Shu, and Y. Song, "Integrated control of electric vehicles based on active front steering and model predictive control," *Control Engineering Practice*, vol. 121, p. 105066, 2022.

[62] J. Wang, X. Zhang, Y. Dong, S. Liu, and L. Zhang, "Roll stability control of in-wheel motors drive electric vehicle on potholed roads," *Control Engineering Practice*, vol. 157, p. 106247, 2025.

[63] J. Liang, Y. Lu, F. Wang, G. Yin, X. Zhu, and Y. Li, "A robust dynamic game-based control framework for integrated torque vectoring and active front-wheel steering system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 7, pp. 7328–7341, 2023.

[64] J. Dong, J. Li, Q. Gao, J. Hu, and Z. Liu, "Optimal coordinated control of active steering and direct yaw moment for distributed-driven electric vehicles," *Control Engineering Practice*, vol. 134, p. 105486, 2023.

[65] S. Gao, J. Wang, C. Guan, Z. Zhou, and Z. Liu, "Dynamic characteristic modeling and RBF-SMC based torque control of a novel torque vectoring drive-axle for electric vehicles," *IEEE Transactions on Vehicular Technology*, 2024.

[66] M. S. Jneid and P. Harth, "Integrated torque vectoring control using vehicle yaw rate and sideslip angle for improving steering and stability of all off-wheel-motor drive electric vehicles," *Acta Polytech. Hung*, vol. 21, pp. 87–106, 2024.

[67] N. Ahmadian, A. Khosravi, and P. Sarhadi, "Integrated model reference adaptive control to coordinate active front steering and direct yaw moment control," *ISA transactions*, vol. 106, pp. 85–96, 2020.

[68] S. Ding, L. Liu, and W. X. Zheng, "Sliding mode direct yaw-moment control design for in-wheel electric vehicles," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6752–6762, 2017.

[69] A. Tahouni, M. Mirzaei, and B. Najjari, "Novel constrained nonlinear control of vehicle dynamics using integrated active torque vectoring and electronic stability control," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 9564–9572, 2019.

[70] E. Morera-Torres, C. Ocampo-Martinez, and F. D. Bianchi, "Experimental modelling and optimal torque vectoring control for 4WD vehicles," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 4922–4932, 2022.

[71] M. Tristano, B. Lenzo, X. Xu, B. Forrier, T. D'hondt, E. Risaliti, and E. Wilhelm, "Hardware-in-the-loop real-time implementation of a vehicle stability control through individual wheel torques," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 4, pp. 4683–4693, 2024.

[72] A. Mangia, B. Lenzo, and E. Sabbioni, "An integrated torque-vectoring control framework for electric vehicles featuring multiple handling and energy-efficiency modes selectable by the driver," *Meccanica*, vol. 56, no. 5, pp. 991–1010, 2021.

[73] M. Rokonuzzaman, N. Mohajer, S. Nahavandi, and S. Mohamed, "Model predictive control with learned vehicle dynamics for autonomous vehicle path tracking," *IEEE Access*, vol. 9, pp. 128 233–128 249, 2021.

[74] P. Fang, Y. Cai, L. Chen, H. Wang, Y. Li, M. A. Sotelo, and Z. Li, "A high-performance neural network vehicle dynamics model for trajectory tracking control," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 237, no. 7, pp. 1695–1709, 2023.

[75] E. Katsuyama, M. Yamakado, and M. Abe, "A state-of-the-art review: toward a novel vehicle dynamics control concept taking the driveline of electric vehicles into account as promising control actuators," *Vehicle System Dynamics*, vol. 59, no. 7, pp. 976–1025, 2021.

[76] L. S. Sawaqed and I. H. Rabbaa, "Fuzzy yaw rate and sideslip angle direct yaw moment control for student electric racing vehicle with independent motors," *World Electric Vehicle Journal*, vol. 13, no. 7, p. 109, 2022.

[77] M. Vignati and E. Sabbioni, "A cooperative control strategy for yaw rate and sideslip angle control combining torque vectoring with rear wheel steering," *Vehicle system dynamics*, vol. 60, no. 5, pp. 1668–1701, 2022.

[78] H. Cha, E. Joa, K. Park, J. Park, and K. Yi, "Torque vectoring control of a hybrid drive vehicle to enhance vehicle agility and stability," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 237, no. 13, pp. 3165–3185, 2023.

[79] S. Yahagi and M. Suzuki, "Intelligent PI control based on the ultra-local model and kalman filter for vehicle yaw-rate control," *SICE Journal of Control, Measurement, and System Integration*, vol. 16, no. 1, pp. 38–47, 2023.

[80] C. Guan, J. Wang, T. Zheng, Q. Wang, W. Sun, and H. Wang, "Parameter optimization and multi-mode operation of a novel dual-motor coupling torque vectoring drive system for electric vehicles," *Applied Energy*, vol. 389, p. 125744, 2025.

[81] L. M. Castellanos Molina, R. Manca, S. Hegde, N. Amati, and A. Tonoli, "Predictive handling limits monitoring and agility improvement with torque vectoring on a rear in-wheel drive electric vehicle," *Vehicle System Dynamics*, vol. 62, no. 9, pp. 2185–2209, 2024.

[82] M. Švec, Š. Ileš, and J. Matuško, "Predictive direct yaw moment control based on the koopman operator," *IEEE transactions on control systems technology*, vol. 31, no. 6, pp. 2912–2919, 2023.

[83] P. Sun, A. Stensson Trigell, L. Drugge, and J. Jerrelind, "Energy-efficient direct yaw moment control for in-wheel motor electric vehicles utilising motor efficiency maps," *Energies*, vol. 13, no. 3, p. 593, 2020.

[84] R. Achdad, A. Rabhi, and J. Bosche, "Energy-efficient torque distribution strategy for four wheel drive electric vehicles based on traffic zone," *IFAC-PapersOnLine*, vol. 58, no. 10, pp. 1–6, 2024.

[85] F. Tarhini, R. Talj, and M. Doumiati, "Holistic adaptive energy-efficient mpc architecture for multi-objective control in over-actuated autonomous vehicles," *Control Engineering Practice*, vol. 164, p. 106464, 2025.

[86] ——, "On analytical modeling for fast multi-objective torque allocation in over-actuated iwm vehicles," *IEEE/CAA Journal of Automatica Sinica*, vol. 12, pp. 1–20, 2025.

[87] Y. Liang, W. Zhao, J. Wu, K. Xu, X. Zhou, Z. Luan, and C. Wang, "Energy-efficient driving for distributed electric vehicles considering wheel loss energy: A distributed strategy based on multi-agent architecture," *Applied Energy*, vol. 384, p. 125462, 2025.

[88] A. Parra, D. Tavernini, P. Gruber, A. Sorniotti, A. Zubizarreta, and J. Pérez, "On nonlinear model predictive control for energy-efficient torque-vectoring," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 1, pp. 173–188, 2020.

[89] X. Hu, H. Chen, Z. Li, and P. Wang, "An energy-saving torque vectoring control strategy for electric vehicles considering handling stability under extreme conditions," *IEEE transactions on vehicular technology*, vol. 69, no. 10, pp. 10 787–10 796, 2020.

[90] J. Zhi, X. Wang, Q. Shi, Z. Yao, and Y. Qi, "Torque allocation strategy based on economy and stability for electric vehicle considering controllability after motors failure," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 237, no. 12, pp. 2759–2779, 2023.

[91] A. Wong, D. Kasinathan, A. Khajepour, S.-K. Chen, and B. Litkouhi, "Integrated torque vectoring and power management framework for electric vehicles," *Control Engineering Practice*, vol. 48, pp. 22–36, 2016.

[92] C.-T. Chung and H.-Y. Tsai, "Conceptual design and energy efficiency evaluation for a novel torque vectoring differential applied to front-wheel-drive electric vehicles," *Applied Sciences*, vol. 13, no. 20, p. 11434, 2023.

[93] W. Chen, J. Peng, Y. Ma, H. He, T. Ren, and C. Wang, "Eco-driving framework for hybrid electric vehicles in multi-lane scenarios by using deep reinforcement learning methods," *Green Energy and Intelligent Transportation*, p. 100309, 2025.

[94] Z. Han, N. Xu, H. Chen, Y. Huang, and B. Zhao, "Energy-efficient control of electric vehicles based on linear quadratic regulator and phase plane analysis," *Applied Energy*, vol. 213, pp. 639–657, 2018.

[95] H. Peng, W. Wang, C. Xiang, L. Li, and X. Wang, "Torque coordinated control of four in-wheel motor independent-drive vehicles with consideration of the safety and economy," *IEEE transactions on vehicular technology*, vol. 68, no. 10, pp. 9604–9618, 2019.

[96] S. Taherian, K. Halder, S. Dixit, and S. Fallah, "Autonomous collision avoidance using mpc with LQR-based weight transformation," *Sensors*, vol. 21, no. 13, p. 4296, 2021.

[97] J. Antunes, A. Antunes, P. Outeiro, C. Cardeira, and P. Oliveira, "Testing of a torque vectoring controller for a formula student prototype," *Robotics and Autonomous Systems*, vol. 113, pp. 56–62, 2019.

[98] H. Wang, J. Han, and H. Zhang, "Lateral stability analysis of 4WID electric vehicle based on sliding mode control and optimal distribution torque strategy," in *Actuators*, vol. 11, no. 9. MDPI, 2022, p. 244.

[99] L. Zhang, H. Ding, J. Shi, Y. Huang, H. Chen, K. Guo, and Q. Li, "An adaptive backstepping sliding mode controller to improve vehicle maneuverability and stability via torque vectoring control," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 2598–2612, 2020.

[100] X. Sun, Y. Wang, Y. Cai, P. K. Wong, L. Chen, and S. Bei, "Nonsingular terminal sliding mode-based direct yaw moment control for four-wheel independently actuated autonomous vehicles," *IEEE Transactions on Transportation Electrification*, vol. 9, no. 2, pp. 2568–2582, 2022.

[101] M. Majidi and A. N. Asiabar, "Stability enhancement of in-wheel motor drive electric vehicle using adaptive sliding mode control." *International Journal of Advanced Design & Manufacturing Technology*, vol. 15, no. 2, 2022.

[102] L. Zhang, H. Chen, Y. Huang, P. Wang, and K. Guo, "Human-centered torque vectoring control for distributed drive electric vehicle considering driving characteristics," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 8, pp. 7386–7399, 2021.

[103] T. Xu, Y. Zhao, H. Deng, S. Guo, D. Li, and F. Lin, "Integrated optimal control of distributed in-wheel motor drive electric vehicle in consideration of the stability and economy," *Energy*, vol. 282, p. 128990, 2023.

[104] Y. Ren, L. Zheng, and A. Khajepour, "Integrated model predictive and torque vectoring control for path tracking of 4-wheel-driven autonomous vehicles," *IET Intelligent Transport Systems*, vol. 13, no. 1, pp. 98–107, 2019.

[105] M. Cao, C. Hu, R. Wang, J. Wang, and N. Chen, "Compensatory model predictive control for post-impact trajectory tracking via active front steering and differential torque vectoring," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 235, no. 4, pp. 903–919, 2021.

[106] Y. Su, D. Liang, and P. Kou, "MPC-based torque distribution for planar motion of four-wheel independently driven electric vehicles: considering motor models and iron losses," *CES Transactions on Electrical Machines and Systems*, vol. 7, no. 1, pp. 45–53, 2023.

[107] K. Han, G. Park, G. S. Sankar, K. Nam, and S. B. Choi, "Model predictive control framework for improving vehicle cornering performance using handling characteristics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 3014–3024, 2020.

[108] C. Lin, S. Liang, J. Chen, and X. Gao, "A multi-objective optimal torque distribution strategy for four in-wheel-motor drive electric vehicles," *IEEE Access*, vol. 7, pp. 64 627–64 640, 2019.

[109] B. Zhao, N. Xu, H. Chen, K. Guo, and Y. Huang, "Design and experimental evaluations on energy-efficient control for 4WIMD-EVs considering tire slip energy," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14 631–14 644, 2020.

[110] H. Liu, L. Zhang, P. Wang, and H. Chen, "A real-time NMPC strategy for electric vehicle stability improvement combining torque vectoring with rear-wheel steering," *IEEE Transactions on Transportation Electrification*, vol. 8, no. 3, pp. 3825–3835, 2022.

[111] N. Guo, B. Lenzo, X. Zhang, Y. Zou, R. Zhai, and T. Zhang, "A real-time nonlinear model predictive controller for yaw motion optimization of distributed drive electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 4935–4946, 2020.

[112] C. Caponio, G. Tavolo, D. Tavernini, A. Hartavi, J. Ahmadi, B. Lenzo, G. Reina, G. Mantriota, P. Perlo, and A. Sorniotti, "Nonlinear model predictive yaw moment control through electric axle and friction brake torque distribution," *IEEE Access*, 2025.

[113] E. Siampis, E. Velenis, S. Gariuolo, and S. Longo, "A real-time nonlinear model predictive control strategy for stabilization of an electric vehicle at the limits of handling," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 6, pp. 1982–1994, 2017.

[114] S. H. Kim and K.-K. K. Kim, "Model predictive control for energy-efficient yaw-stabilizing torque vectoring in electric vehicles with four in-wheel motors," *IEEE Access*, vol. 11, pp. 37 665–37 680, 2023.

[115] H. Deng, Y. Zhao, S. Feng, Q. Wang, C. Zhang, and F. Lin, "Torque vectoring algorithm based on mechanical elastic electric wheels with consideration of the stability and economy," *Energy*, vol. 219, p. 119643, 2021.

[116] M. Dalboni, D. Tavernini, U. Montanaro, A. Soldati, C. Concari, M. Dhaens, and A. Sorniotti, "Nonlinear model predictive control for integrated energy-efficient torque-vectoring and anti-roll moment distribution," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 3, pp. 1212–1224, 2021.

[117] D.-H. Kim, C.-J. Kim, S.-H. Kim, J.-Y. Choi, and C.-S. Han, "Development of adaptive direct yaw-moment control method for electric vehicle based on identification of yaw-rate model," in *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2011, pp. 1098–1103.

[118] L. De Novellis, A. Sorniotti, P. Gruber, and A. Pennycott, "Comparison of feedback control techniques for torque-vectoring control of fully electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 8, pp. 3612–3623, 2014.

[119] N. Ahmadian, A. Khosravi, and P. Sarhadi, "Integrated model reference adaptive control to coordinate active front steering and direct yaw moment control," *ISA transactions*, vol. 106, pp. 85–96, 2020.

[120] ——, "Adaptive yaw stability control by coordination of active steering and braking with an optimized lower-level controller," *International Journal of Adaptive Control and Signal Processing*, vol. 34, no. 9, pp. 1242–1258, 2020.

[121] Z. Zhao, C. K. Lee, X. Yan, and H. Wang, "Reinforcement learning for electric vehicle charging scheduling: A systematic review," *Transportation Research Part E: Logistics and Transportation Review*, vol. 190, p. 103698, 2024.

[122] W. Huang, K. Huang, and M. Huang, "Improved deep reinforcement learning decision for integrated control of distributed drive electric vehicle dynamics," in *Journal of Physics: Conference Series*, vol. 2483, no. 1. IOP Publishing, 2023, p. 012004.

[123] C. Tang, B. Abbatematteo, J. Hu, R. Chandra, R. Martín-Martín, and P. Stone, "Deep reinforcement learning for robotics: A survey of real-world successes," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 27, 2025, pp. 28 694–28 698.

[124] P. Jamjuntr, C. Techawatcharapaikul, and P. Suanpang, "Adaptive multi-agent reinforcement learning for optimizing dynamic electric vehicle charging networks in thailand," *World Electric Vehicle Journal*, vol. 15, no. 10, p. 453, 2024.

[125] O. A. Yildiz, I. Saricicek, and A. Yazici, "A reinforcement learning-based solution for the capacitated electric vehicle routing problem from the last-mile delivery perspective," *APPLIED SCIENCES-BASEL*, vol. 15, no. 3, 2025.

[126] A. K. Shakya, G. Pillai, and S. Chakrabarty, "Reinforcement learning algorithms: A brief survey," *Expert Systems with Applications*, vol. 231, p. 120495, 2023.

[127] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel *et al.*, "Mastering atari, go, chess and shogi by planning with a learned model," *Nature*, vol. 588, no. 7839, pp. 604–609, 2020.

[128] C. Allen, N. Parikh, O. Gottesman, and G. Konidaris, "Learning markov state abstractions for deep reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8229–8241, 2021.

[129] B. Wang, Y. Yan, and J. Fan, "Sample-efficient reinforcement learning for linearly-parameterized mdps with a generative model," *Advances in neural information processing systems*, vol. 34, pp. 23 009–23 022, 2021.

[130] C. Shi, R. Wan, R. Song, W. Lu, and L. Leng, "Does the markov decision process fit the data: Testing for the markov property in sequential decision making," in *International Conference on Machine Learning*. PMLR, 2020, pp. 8807–8817.

[131] 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.

[132] X. Jin, H. Lv, Y. Tao, J. Lu, J. Lv, and N. V. Opinat Ikiela, "Deep reinforcement learning-based active disturbance rejection control for trajectory tracking of autonomous ground electric vehicles," *Machines*, vol. 13, no. 6, p. 523, 2025.

[133] P. Cai, X. Mei, L. Tai, Y. Sun, and M. Liu, "High-speed autonomous drifting with deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1247–1254, 2020.

[134] D. Liu, P. Zeng, S. Cui, and C. Song, "Deep reinforcement learning for charging scheduling of electric vehicles considering distribution network voltage stability," *Sensors*, vol. 23, no. 3, p. 1618, 2023.

[135] A. Najar and M. Chetouani, "Reinforcement learning with human advice: a survey," *Frontiers in Robotics and AI*, vol. 8, p. 584075, 2021.

[136] X. Ma, Y. Yang, H. Hu, Q. Liu, J. Yang, C. Zhang, Q. Zhao, and B. Liang, "Offline reinforcement learning with value-based episodic memory," *arXiv preprint arXiv:2110.09796*, 2021.

[137] H. Sun, L. Han, R. Yang, X. Ma, J. Guo, and B. Zhou, "Exploit reward shifting in value-based deep-rl: Optimistic curiosity-based exploration and conservative exploitation via linear reward shaping," *Advances in neural information processing systems*, vol. 35, pp. 37 719–37 734, 2022.

[138] X. Lin, J. Huang, B. Zhang, B. Zhou, and Z. Chen, "A velocity adaptive steering control strategy of autonomous vehicle based on double deep q-learning network with varied agents," *Engineering Applications of Artificial Intelligence*, vol. 139, p. 109655, 2025.

[139] O. Sigaud, "Combining evolution and deep reinforcement learning for policy search: A survey," *ACM Transactions on Evolutionary Learning*, vol. 3, no. 3, pp. 1–20, 2023.

[140] G. Xu, X. He, M. Chen, H. Miao, H. Pang, J. Wu, P. Diao, and W. Wang, "Hierarchical speed control for autonomous electric vehicle through deep reinforcement learning and robust control," *IET Control Theory & Applications*, vol. 16, no. 1, pp. 112–124, 2022.

[141] A. Zanette, M. J. Wainwright, and E. Brunskill, "Provable benefits of actor-critic methods for offline reinforcement learning," *Advances in neural information processing systems*, vol. 34, pp. 13 626–13 640, 2021.

[142] M. F. Drechsler, T. A. Fiorentin, and H. Göllinger, "Actor-critic traction control based on reinforcement learning with open-loop training," *Modelling and Simulation in Engineering*, vol. 2021, no. 1, p. 4641450, 2021.

[143] X. Wang, Y. Chen, and W. Zhu, "A survey on curriculum learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 9, pp. 4555–4576, 2021.

[144] P. Klink, H. Yang, C. D'Eramo, J. Peters, and J. Pajarinen, "Curriculum reinforcement learning via constrained optimal transport," in *International Conference on Machine Learning*. PMLR, 2022, pp. 11 341–11 358.

[145] X. Zhang, G. Xiong, Y. Ai, K. Liu, and L. Chen, "Vehicle dynamic dispatching using curriculum-driven reinforcement learning," *Mechanical Systems and Signal Processing*, vol. 204, p. 110698, 2023.

[146] K. Yu, M. Fu, X. Tian, S. Yang, and Y. Yang, "Curriculum reinforcement learning-based drifting along a general path for autonomous vehicles," *Robotica*, vol. 42, no. 10, pp. 3263–3280, 2024.

[147] S. Li, W. Hu, D. Cao, Z. Zhang, Q. Huang, Z. Chen, and F. Blaabjerg, "EV charging strategy considering transformer lifetime via evolutionary curriculum learning-based multiagent deep reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 13, no. 4, pp. 2774–2787, 2022.

[148] G.-P. Antonio and C. Maria-Dolores, "Multi-agent deep reinforcement learning to manage connected autonomous vehicles at tomorrow's intersections," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 7, pp. 7033–7043, 2022.

[149] R. Gutiérrez-Moreno, R. Barea, E. López-Guillén, J. Araluce, and L. M. Bergasa, "Reinforcement learning-based autonomous driving at intersections in CARLA simulator," *Sensors*, vol. 22, no. 21, p. 8373, 2022.

[150] A. Khan, Muhammad, and M. Naeem, "Optimizing reinforcement learning agents in games using curriculum learning and reward shaping," *Computer Animation and Virtual Worlds*, vol. 36, no. 1, p. e70008, 2025.

[151] L. Anzalone, P. Barra, S. Barra, A. Castiglione, and M. Nappi, "An end-to-end curriculum learning approach for autonomous driving scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 19 817–19 826, 2022.

[152] Y. Yin, Z. Chen, G. Liu, J. Yin, and J. Guo, "Autonomous navigation of mobile robots in unknown environments using off-policy reinforcement learning with curriculum learning," *Expert Systems with Applications*, vol. 247, p. 123202, 2024.

[153] A. Lelkó, B. Németh, and P. Gáspár, "Reinforcement learning-based robust vehicle control for autonomous vehicle trajectory tracking," *Engineering Proceedings*, vol. 79, no. 1, p. 30, 2024.

[154] G. Ananganó-Alvarado, I. Umaña-Morel, and B. Keith-Norambuena, "Reinforcement learning in electric vehicle energy management: a comprehensive open-access review of methods, challenges, and future innovations," *Frontiers in Future Transportation*, vol. 6, p. 1555250, 2025.

[155] X. Chen, C. Wang, Z. Zhou, and K. Ross, "Randomized ensembled double q-learning: Learning fast without a model," *arXiv preprint arXiv:2101.05982*, 2021.

[156] Y. Song, P. N. Suganthan, W. Pedrycz, J. Ou, Y. He, Y. Chen, and Y. Wu, "Ensemble reinforcement learning: A survey," *Applied Soft Computing*, vol. 149, p. 110975, 2023.

[157] H. Deng, Y. Zhao, F. Lin, and Q. Wang, "Deep reinforcement learning-based torque vectoring control considering economy and safety," *Machines*, vol. 11, no. 4, p. 459, 2023.

[158] H. Deng, Y. Zhao, A.-T. Nguyen, and C. Huang, "Fault-tolerant predictive control with deep-reinforcement-learning-based torque distribution for four in-wheel motor drive electric vehicles," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 2, pp. 668–680, 2023.

[159] R. Siraskar, "Reinforcement learning for control of valves," *Machine Learning with Applications*, vol. 4, p. 100030, 2021.

[160] J. Kim, S. Park, J. Kim, and J. Yoo, "A deep reinforcement learning strategy for surrounding vehicles-based lane-keeping control," *Sensors*, vol. 23, no. 24, p. 9843, 2023.

[161] E. H. Sumiea, S. J. Abdulkadir, H. S. Alhussian, S. M. Al-Selwi, A. Alqushaibi, M. G. Ragab, and S. M. Fati, "Deep deterministic policy gradient algorithm: A systematic review," *Heliyon*, 2024.

[162] H. Wei, W. Zhao, Q. Ai, Y. Zhang, and T. Huang, "Deep reinforcement learning based active safety control for distributed drive electric vehicles," *IET Intelligent Transport Systems*, vol. 16, no. 6, pp. 813–824, 2022.

[163] H. Dai, P. Chen, and H. Yang, "Driving torque distribution strategy of skid-steering vehicles with knowledge-assisted reinforcement learning," *Applied Sciences*, vol. 12, no. 10, p. 5171, 2022.

[164] S. Taherian, S. Kuutti, M. Visca, and S. Fallah, "Self-adaptive torque vectoring controller using reinforcement learning," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 172–179.

[165] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*. PMLR, 2018, pp. 1587–1596.

[166] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[167] J. Zhou, S. Xue, Y. Xue, Y. Liao, J. Liu, and W. Zhao, "A novel energy management strategy of hybrid electric vehicle via an improved td3 deep reinforcement learning," *Energy*, vol. 224, p. 120118, 2021.

[168] S. Liu, "An evaluation of DDPG, TD3, SAC, and PPO: deep reinforcement learning algorithms for controlling continuous system," in *2023 International Conference on Data Science, Advanced Algorithm and Intelligent Computing (DAI 2023)*. Atlantis Press, 2024, pp. 15–24.

[169] H. Wei, N. Zhang, J. Liang, Q. Ai, W. Zhao, T. Huang, and Y. Zhang, "Deep reinforcement learning based direct torque control strategy for distributed drive electric vehicles considering active safety and energy saving performance," *Energy*, vol. 238, p. 121725, 2022.

[170] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.

[171] W. Sayssouk, R. Orjuela, C. Roos, and M. Basset, "Coordinated torque vectoring and direct yaw control based on kalman filter control allocation for a four in-wheel electric vehicle," *Transactions of the Institute of Measurement and Control*, vol. 47, no. 7, pp. 1375–1386, 2025.

[172] Y. Zou, N. Guo, and X. Zhang, "An integrated control strategy of path following and lateral motion stabilization for autonomous distributed drive electric vehicles," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 235, no. 4, pp. 1164–1179, 2021.

[173] N. Guo, X. Zhang, Y. Zou, B. Lenzo, G. Du, and T. Zhang, "A supervisory control strategy of distributed drive electric vehicles for coordinating handling, lateral stability, and energy efficiency," *IEEE transactions on transportation electrification*, vol. 7, no. 4, pp. 2488–2504, 2021.

[174] P. K. Wong and D. Ao, "A novel event-triggered torque vectoring control for improving lateral stability and communication resource consumption of electric vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 2046–2060, 2023.

[175] L. Zhai, R. Hou, T. Sun, and S. Kavuma, "Continuous steering stability control based on an energy-saving torque distribution algorithm for a four in-wheel-motor independent-drive electric vehicle," *Energies*, vol. 11, no. 2, p. 350, 2018.

[176] J. Liang, J. Feng, Z. Fang, Y. Lu, G. Yin, X. Mao, J. Wu, and F. Wang, "An energy-oriented torque-vector control framework for distributed drive electric vehicles," *IEEE Transactions on Transportation Electrification*, vol. 9, no. 3, pp. 4014–4031, 2023.

[177] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[178] M. Doumiati, O. Sename, L. Dugard, J.-J. Martinez-Molina, P. Gaspar, and Z. Szabo, "Integrated vehicle dynamics control via coordination of active front steering and rear braking," *European Journal of Control*, vol. 19, no. 2, pp. 121–143, 2013.

[179] Y. Shen, Y. Zhao, H. Deng, F. Lin, and H. Shen, "Coordinated control of stability and economy of distributed drive electric vehicle based on lyapunov adaptive theory," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 238, no. 6, pp. 1535–1549, 2024.

[180] S. Ding, L. Liu, and W. X. Zheng, "Sliding mode direct yaw-moment control design for in-wheel electric vehicles," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6752–6762, 2017.

[181] S. Gao, J. Wang, C. Guan, Z. Zhou, and Z. Liu, "Wheel torque distribution control strategy for electric vehicles dynamic performance with an electric torque vectoring drive axle," *IEEE Transactions on Transportation Electrification*, vol. 10, no. 1, pp. 1692–1705, 2023.

[182] S. Li, G. Wang, B. Zhang, Z. Yu, and G. Cui, "Vehicle yaw stability control at the handling limits based on model predictive control," *International Journal of Automotive Technology*, vol. 21, pp. 361–370, 2020.

[183] Q. Kang, D. Meng, Y. Jiang, C. Zhao, H. Chu, and B. Gao, "Torque vectoring control of distributed drive electric vehicles using fast iterative model predictive control," *IFAC-PapersOnLine*, vol. 58, no. 29, pp. 106–111, 2024.

[184] A. Parra, D. Tavernini, P. Gruber, A. Sorniotti, A. Zubizarreta, and J. Pérez, "On pre-emptive vehicle stability control," *Vehicle system dynamics*, vol. 60, no. 6, pp. 2098–2123, 2022.

[185] A. N. Asiabar and R. Kazemi, "A direct yaw moment controller for a four in-wheel motor drive electric vehicle using adaptive sliding mode control," *Proceedings of the institution of mechanical engineers, part K: journal of multi-body dynamics*, vol. 233, no. 3, pp. 549–567, 2019.

[186] H. Pacejka, *Tire and vehicle dynamics*. Elsevier, 2005.

[187] E. Siampis, E. Velenis, and S. Longo, "Model predictive torque vectoring control for electric vehicles near the limits of handling," in *2015 European Control Conference (ECC)*. IEEE, 2015, pp. 2553–2558.

[188] Y. Park, J. Gim, and C. Ahn, "Post-impact stabilization during lane change maneuver," *Electronics*, vol. 12, no. 22, p. 4712, 2023.

[189] O. Ammari, K. E. Majdoub, F. Giri, and R. Baz, "Modeling and control design for half electric vehicle with wheel bldc actuator and Pacejka's tire," *Computers and Electrical Engineering*, vol. 116, p. 109163, 2024.

[190] Y. Lian, G. Chen, and P. Liu, "Study of yaw moment control strategy of four wheel independent drive electric vehicle," *Automotive Innovation*, pp. 1–12, 2025.

[191] E. Siampis, E. Velenis, and S. Longo, "Rear wheel torque vectoring model predictive control with velocity regulation for electric vehicles," *Vehicle System Dynamics*, vol. 53, no. 11, pp. 1555–1579, 2015.

[192] E. Siampis, M. Massaro, and E. Velenis, "Electric rear axle torque vectoring for combined yaw stability and velocity control near the limit of handling," in *52nd IEEE conference on Decision and Control*. IEEE, 2013, pp. 1552–1557.

[193] L. Su, Z. Wang, and C. Chen, "Torque vectoring control system for distributed drive electric bus under complicated driving conditions," *Assembly Automation*, vol. 42, no. 1, pp. 1–18, 2022.

Appendix A

Vehicle Dynamic Control Modelling

A.1 Introduction

Vehicle dynamics control plays a central role in modern automotive engineering and operates in parallel with energy management to enhance safety, stability, and performance. Advances in sensing, computation, and actuation have enabled the implementation of increasingly sophisticated control algorithms capable of regulating the vehicle's dynamic behaviour in real time [98]. The primary objective of vehicle dynamics control is to maintain stable and predictable handling across a wide range of driving conditions. This is achieved by generating corrective yaw moments to prevent undesirable behaviours such as skidding, oversteer, or understeer. On high-friction surfaces, vehicles are more prone to oversteer, whereas on low-friction surfaces or at high speeds, understeer is more likely to occur, as illustrated in Fig. A.1. These behaviours highlight the need for an effective stability control system that can mitigate both understeer and oversteer while preserving consistent handling characteristics.

In AWD EVs equipped with in-wheel electric machines, torque vectoring provides an effective means of controlling yaw motion and lateral dynamics. By distributing the drive torques independently among the four wheels, the system can generate a desired yaw moment without relying on additional braking intervention. This approach not only enhances stability and manoeuvrability but also complements energy management objectives through the optimal use of available electric machine torque.

A.2 Torque Vectoring

Torque vectoring refers to the active distribution of drive torque between the axles and the left and right wheels to generate a corrective yaw moment and maintain balanced tyre force

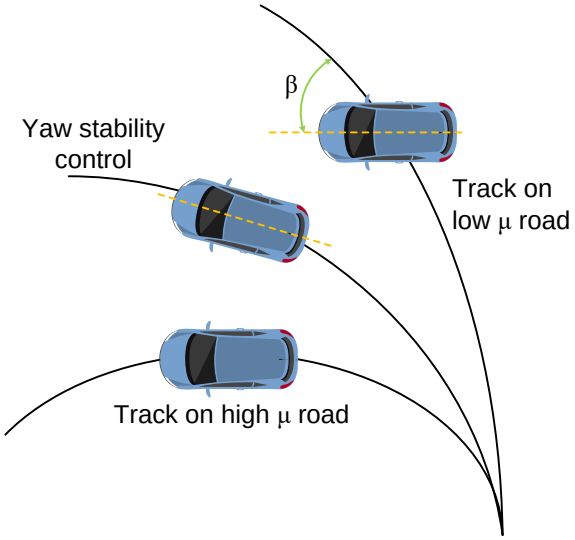


Fig. A.1 Oversteer and understeer.

utilisation. By directly shaping this yaw moment, torque vectoring enables the vehicle to follow the desired trajectory more accurately, particularly during cornering or on surfaces with uneven or low tyre–road friction [47]. In comparison with passive differentials, electric powertrains equipped with in-wheel electric machines provide faster and more precise torque control, offering higher bandwidth and finer resolution in response to dynamic driving conditions.

Torque vectoring has a significant role in vehicle dynamics control and is closely related to energy management. While energy management focuses on minimising energy consumption, vehicle dynamics control ensures stability, handling, and responsiveness across a range of manoeuvres. In an integrated control framework, torque vectoring can be coordinated with energy management strategies to achieve both efficiency and stability objectives simultaneously. By modulating individual wheel torques in real time, torque vectoring effectively mitigates understeer and oversteer tendencies, particularly at high lateral acceleration or on low-friction surfaces. This results in improved vehicle safety, enhanced cornering performance, and greater driver confidence under diverse operating conditions [47].

A.3 Hierarchical Dynamic Control Configuration

With the increasing level of control available over individual wheel torques and the transition of the automotive industry towards full electrification, torque vectoring provides an effective means of managing torque distribution in AWD EVs [171–173]. To achieve enhanced stability

performance, a hierarchical dynamic control configuration is adopted in this research, as explain in previous Chapter. The general structure of a conventional hierarchical torque vectoring control system is illustrated in Fig. 2.3. As shown, the system is composed of four main components:

1. Reference generator
2. High-level controller
3. Low-level controller
4. Vehicle model

The reference generator defines the desired dynamic states of the vehicle based on driver commands and operating conditions. These reference values in this work include the desired yaw rate ($\dot{\psi}_{des}$) and the desired sideslip angle (β_{des}). They are the target indicators for vehicle performance, providing the benchmark against which the actual vehicle response is compared.

The high-level controller maintains vehicle stability by generating the corrective yaw moment (M_z) required to minimise the deviation between the reference and actual values. It processes the errors to determine the additional yaw moment necessary for stability. This controller is responsible for ensuring that the vehicle follows the intended trajectory, particularly during aggressive manoeuvres or under adverse road conditions. The low-level controller receives the corrective yaw moment command from the high-level controller and determines the optimal torque to be applied to each of the four wheels. It ensures coordinated distribution of torques in a way that satisfies both stability and actuator constraints. Finally, the vehicle model represents the dynamic behaviour of the vehicle, and provides a virtual platform for control design and verification, enabling accurate analysis of how different control inputs affect vehicle responses.

A.4 Vehicle Dynamic Model

The effectiveness of the control strategy in vehicle dynamics relies strongly on the accuracy and fidelity of the underlying vehicle dynamic model. A reliable model is essential for capturing the nonlinear behaviour of the vehicle, enabling the controller to predict system responses and apply corrective actions effectively. This section presents the dynamic models that form the basis of the proposed hierarchical control architecture. The subsequent subsections describe different modelling approaches used in this research [174–176].

A.4.1 Kinematic Model of Lateral Vehicle Motion

The kinematic model provides a simplified mathematical representation of vehicle motion by describing the geometric relationships of the vehicle's trajectory while neglecting the forces acting on it. The main assumption in this model is the omission of tyre slip angles at both the front and rear wheels, which is valid under low-speed operating conditions where lateral tyre forces are small [177]. Based on the geometric relationships illustrated in Fig. A.2, the equations of motion for the kinematic model are derived.

In this model, the front left and right wheels are represented by a single equivalent wheel located at point A, and the rear wheels are represented by a single equivalent wheel at point B. The vehicle's COG is denoted by point C. As shown in Fig. A.2, δ_f and δ_r are the front and rear steering angles, respectively. The distances from the COG to the front and rear axles are ℓ_f and ℓ_r, and the total wheelbase of the vehicle is defined as $L = \ell_f + \ell_r$. The vehicle motion is assumed to occur within a single plane. The global coordinates of the COG are represented by X and Y, while ψ denotes the yaw angle, which defines the orientation of the vehicle with respect to the global X-axis. The vehicle velocity at the COG is denoted by V, and the angle between V and the vehicle's longitudinal axis is defined as the sideslip angle β.

The equations of motion for the kinematic model are expressed as:

$$\dot{X} = V\cos(\psi + \beta) \tag{A.1}$$

$$\dot{Y} = V\sin(\psi + \beta) \tag{A.2}$$

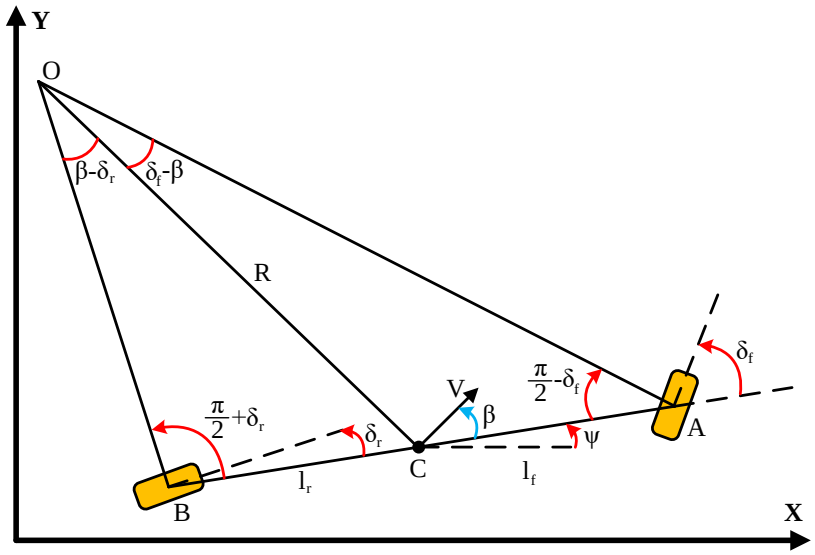


Fig. A.2 Schematic representation of vehicle kinematic model.

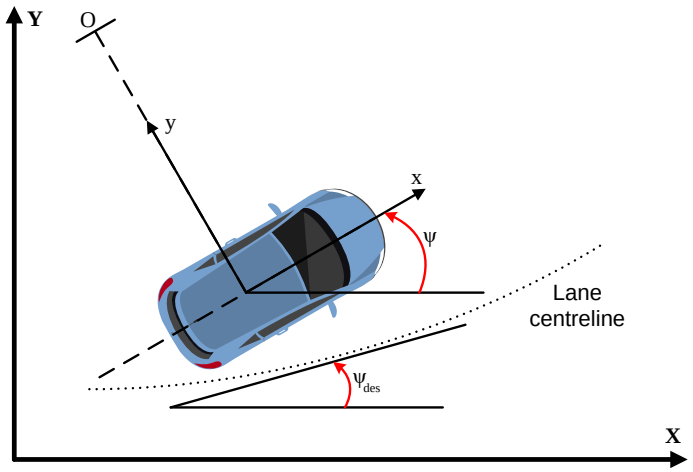


Fig. A.3 Schematic of the lateral vehicle dynamics.

$$\dot{\psi} = \frac{V\cos(\beta)}{\ell_f + \ell_r}\left[\tan(\delta_f) - \tan(\delta_r)\right] \tag{A.3}$$

The inputs of the kinematic model are δ_f, δ_r, and V. The sideslip angle β can be determined using the following relationship:

$$\beta = \tan^{-1}\left(\frac{\ell_f \tan(\delta_r) + \ell_r \tan(\delta_f)}{\ell_f + \ell_r}\right) \tag{A.4}$$

A.4.2 Bicycle Model of Lateral Vehicle Dynamics

At higher vehicle speeds, the velocity vector at each wheel does not align with the wheel orientation, which makes the kinematic model inadequate for capturing the lateral dynamics of the vehicle. To address this limitation, the bicycle model is introduced as a simplified dynamic representation of vehicle motion with two DOF, i.e., the lateral motion y and the yaw motion ψ [178, 179]. The schematic of the model is shown in Fig. A.3.

Newton's second law is applied along the lateral (y) direction, neglecting the effect of road bank angle:

$$ma_y = F_{yf} + F_{yr} \tag{A.5}$$

where F_{yf} and F_{yr} denote the lateral tyre forces at the front and rear wheels, respectively, and m is the vehicle mass. The lateral acceleration at the COG of the vehicle is expressed as:

$$a_y = \ddot{y} + \dot{\psi}V_x \tag{A.6}$$

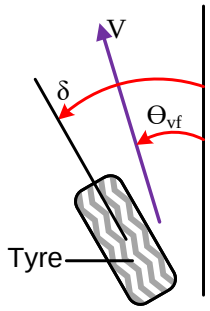


Fig. A.4 Definition of the tyre slip angle.

where $\ddot{y}$ is the lateral acceleration and $\dot{\psi}V_x$ represents the centripetal acceleration. Substituting this into the lateral force balance yields:

$$m(\ddot{y} + \dot{\psi}V_x) = F_{yf} + F_{yr} \tag{A.7}$$

The moment balance about the vertical (z) axis is given by:

$$I_z\ddot{\psi} = \ell_f F_{yf} - \ell_r F_{yr} \tag{A.8}$$

where I_z is the yaw moment of inertia, and ℓ_f and ℓ_r are the distances from the COG to the front and rear axles, respectively.

For small tyre slip angles, the lateral forces are proportional to the slip angles:

$$F_{yf} = 2C_{\alpha f}\alpha_f, \quad F_{yr} = 2C_{\alpha r}\alpha_r \tag{A.9}$$

where $C_{\alpha f}$ and $C_{\alpha r}$ are the cornering stiffness coefficients of the front and rear tyres, and α_f and α_r are the corresponding slip angles. The slip angle of a tyre is the angle between the wheel orientation and the direction of the wheel's velocity vector, as shown in Fig. A.4.

The slip angles for the front and rear wheels can be expressed as:

$$\alpha_f = \delta - \theta_{Vf}, \quad \alpha_r = -\theta_{Vr} \tag{A.10}$$

where δ is the front wheel steering angle, and θ_{Vf} and θ_{Vr} are the velocity angles of the front and rear tyres, respectively. These velocity angles are calculated as:

$$\theta_{Vf} = \frac{V_y + \ell_f\dot{\psi}}{V_x}, \quad \theta_{Vr} = \frac{V_y - \ell_r\dot{\psi}}{V_x} \tag{A.11}$$

Substituting these expressions into the lateral force equations gives:

$$F_{yf} = 2C_{\alpha f}(\delta - \theta_{Vf}), \quad F_{yr} = -2C_{\alpha r}\theta_{Vr} \tag{A.12}$$

Combining the lateral and yaw motion equations, the state-space representation of the bicycle model can be written as:

$$\frac{d}{dt}\begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2(C_{\alpha f}+C_{\alpha r})}{mV_x} & 0 & -V_x - \frac{2(C_{\alpha f}\ell_f - C_{\alpha r}\ell_r)}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2(\ell_f C_{\alpha f} - \ell_r C_{\alpha r})}{I_z V_x} & 0 & -\frac{2(\ell_f^2 C_{\alpha f} + \ell_r^2 C_{\alpha r})}{I_z V_x} \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2\ell_f C_{\alpha f}}{I_z} \end{bmatrix} \delta \tag{A.13}$$

Alternatively, the model can be expressed in terms of the yaw rate ($\dot{\psi}$) and sideslip angle (β), where β is defined as the angle between the longitudinal axis of the vehicle and the velocity vector at the COG. The front and rear slip angles are then given by [178, 180, 181]:

$$\alpha_f = \delta - \beta - \frac{\ell_f \dot{\psi}}{V_x}, \quad \alpha_r = -\beta + \frac{\ell_r \dot{\psi}}{V_x} \tag{A.14}$$

Substituting these expressions into the dynamic equations gives:

$$\dot{\beta} = -\dot{\psi} + \frac{C_{\alpha f}}{mV_x}\left(\delta - \beta - \frac{\ell_f \dot{\psi}}{V_x}\right) + \frac{C_{\alpha r}}{mV_x}\left(-\beta + \frac{\ell_r \dot{\psi}}{V_x}\right) + \frac{g \sin\phi}{V_x} \tag{A.15}$$

$$\ddot{\psi} = \frac{\ell_f C_{\alpha f}}{I_z}\left(\delta - \beta - \frac{\ell_f \dot{\psi}}{V_x}\right) - \frac{\ell_r C_{\alpha r}}{I_z}\left(-\beta + \frac{\ell_r \dot{\psi}}{V_x}\right) \tag{A.16}$$

Thus, the 2-DOF bicycle model can be expressed in the compact state-space form [40]:

$$\begin{bmatrix} \dot{\beta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} -\frac{C_{\alpha f}+C_{\alpha r}}{mV_x} & -\left(1 + \frac{\ell_f C_{\alpha f} - \ell_r C_{\alpha r}}{mV_x^2}\right) \\ -\frac{\ell_f C_{\alpha f} - \ell_r C_{\alpha r}}{I_z} & -\frac{\ell_f^2 C_{\alpha f} + \ell_r^2 C_{\alpha r}}{I_z V_x} \end{bmatrix} \begin{bmatrix} \beta \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \frac{C_{\alpha f}}{mV_x} \\ \frac{\ell_f C_{\alpha f}}{I_z} \end{bmatrix} \delta \tag{A.17}$$

A.4.3 Reference Generation Model

In vehicle dynamics control, generating reference signals is a key step in ensuring that the vehicle follows the desired behaviour accurately. The reference signals for yaw rate and sideslip angle define the target performance that the control system aims to achieve. These references are commonly derived from a steady-state analysis of the bicycle model [182, 183], as illustrated in Fig. A.5.

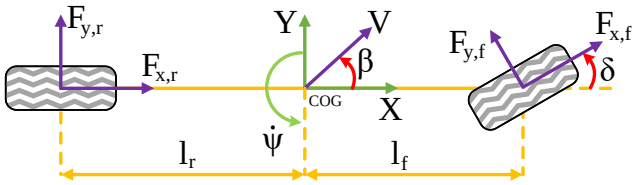


Fig. A.5 Bicycle model with two degrees of freedom.

Under steady-state cornering conditions, the required steering angle for a vehicle travelling on a circular path of radius R can be expressed as:

$$\delta_{ss} = \frac{\ell_f + \ell_r}{R} + K_{US}\frac{v_x^2}{R} \tag{A.18}$$

where K_{US} is the understeer gradient, defined as:

$$K_{US} = \frac{m(\ell_r C_{\alpha r} - \ell_f C_{\alpha f})}{2C_{\alpha f} C_{\alpha r} L} \tag{A.19}$$

Equation (A.18) can be rearranged to obtain the turning radius:

$$\frac{1}{R} = \frac{\delta_{ss}}{L + K_{US} v_x^2} \tag{A.20}$$

Accordingly, the desired yaw rate can be expressed as:

$$\dot{\psi}_{des} = \frac{V}{R} = \frac{V}{L + K_{US} v_x^2}\delta \tag{A.21}$$

Similarly, the desired sideslip angle can be derived from the steady-state yaw angle error during cornering:

$$\beta_{des} = \frac{\ell_r}{R} - \frac{\ell_f}{2C_{\alpha r} L}\frac{m v_x^2}{R} \tag{A.22}$$

Rewriting β_{des} in terms of the steering angle and longitudinal velocity gives:

$$\beta_{des} = \frac{1}{L + K_{US} v_x^2}\left(\ell_r - \frac{\ell_f}{L}\frac{m v_x^2}{2C_{\alpha r}}\right)\delta_f \tag{A.23}$$

In practice, achieving the desired yaw rate and sideslip angle is constrained by safety, vehicle dynamics limitations, and comfort considerations. Excessive reference values may cause the controller to exceed actuator limits or compromise stability. The upper limit of the

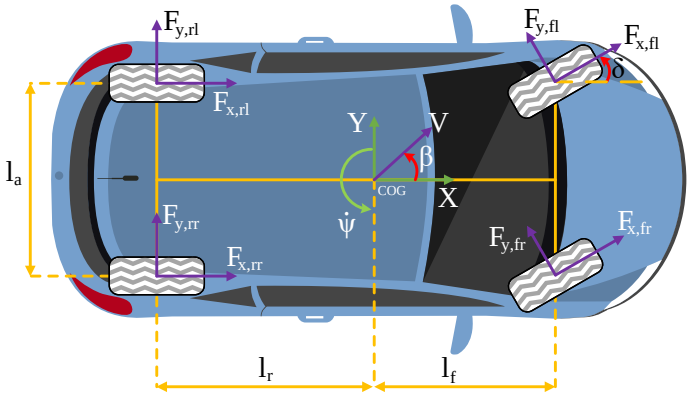


Fig. A.6 Nonlinear vehicle model with seven degrees of freedom.

desired yaw rate is restricted by the available tyre–road friction coefficient, expressed as:

$$\dot{\psi}_{upper\ bound} = \frac{0.85\mu g}{v_x} \tag{A.24}$$

where μ is the tyre–road friction coefficient and g is the gravitational acceleration.

At high sideslip angles, tyres deviate from their linear region and approach the adhesion limit. To prevent excessive sideslip and maintain controllability, the nominal sideslip angle is limited by the empirical relationship:

$$\beta_{nominal} = \tan^{-1}(0.02\mu g) \tag{A.25}$$

A.4.4 Nonlinear Model with Seven Degrees of Freedom

The nonlinear model with seven DOF provides a detailed and comprehensive representation of the dynamic behaviour of the vehicle by capturing the interactions between the forces and moments acting on the vehicle during stability control analysis. The seven DOF include the longitudinal velocity (v_x), lateral velocity (v_y), yaw rate ($\dot{\psi}$), and the rotational speeds of the four wheels ($\dot{\omega}_{ij}$), where $ij \in \{fl, fr, rl, rr\}$, corresponding to the front-left, front-right, rear-left, and rear-right wheels [184, 185]. The schematic of the 7-DOF vehicle model is shown in Fig. A.6.

The translational and rotational equations of motion are formulated using Newton's second law for the planar motion of the vehicle as follows:

$$m(\dot{v}_x - \dot{\psi} v_y) = (F_{x,fl} + F_{x,fr})\cos\delta - (F_{y,fl} + F_{y,fr})\sin\delta + F_{x,rl} + F_{x,rr} \tag{A.26}$$

$$m(\dot{v}_y + \dot{\psi} v_x) = (F_{y,fl} + F_{y,fr})\cos\delta + (F_{x,fl} + F_{x,fr})\sin\delta + F_{y,rl} + F_{y,rr} \tag{A.27}$$

$$I_z\ddot{\psi} = \ell_f(F_{x,fl} + F_{x,fr})\sin\delta + \ell_f(F_{y,fl} + F_{y,fr})\cos\delta - \ell_r(F_{y,rl} + F_{y,rr}) + \frac{\ell_w}{2}\left[(F_{x,fr} - F_{x,fl})\cos\delta + (F_{x,rr} - F_{x,rl})\right] + \frac{\ell_w}{2}(F_{y,fl} - F_{y,fr})\sin\delta \tag{A.28}$$

$$I_\omega\dot{\omega}_{ij} = T_{ij} - F_{x,ij}R_w \tag{A.29}$$

In these equations, $F_{x,ij}$ and $F_{y,ij}$ denote the longitudinal and lateral tyre forces acting on each wheel, respectively. The parameter m is the vehicle mass, δ is the steering angle of the front wheels, and R_w is the effective tyre radius. The term I_z represents the vehicle yaw moment of inertia about the vertical axis, while ℓ_f and ℓ_r denote the longitudinal distances from the vehicle COG to the front and rear axles, respectively. The track width is represented by ℓ_w, I_ω is the wheel rotational inertia, and T_{ij} is the driving or braking torque applied to each wheel. The 7-DOF vehicle model provides a realistic representation of vehicle behaviour and allows accurate evaluation of control algorithms under various driving conditions. By capturing longitudinal, lateral, and yaw dynamics along with individual wheel motions, the model serves as a high-fidelity platform for analysing and verifying the proposed control strategies.

The tyre model is also important in modelling the dynamic behaviour of the vehicle, as it captures the nonlinear interactions between the tyre and the road surface. An accurate tyre model is essential for describing the generation of longitudinal and lateral forces, which directly influence vehicle stability, handling, and performance. Among the different tyre models developed, the Pacejka Magic Formula, proposed by Hans B. Pacejka [186], is one of the most widely used in vehicle dynamics and control applications. The Magic Formula provides an empirical representation of the nonlinear relationship between tyre forces and slip quantities, offering good accuracy under different speeds, vertical loads, and road conditions [187]. This model is commonly employed for predicting tyre behaviour and for the design and optimisation of advanced vehicle control systems [188, 189].

The general form of the Magic Formula is given as [190, 191]:

$$y(x) = D\sin\{C\arctan[B_1(x + S_h)(1 - E) + E\arctan(B_1(x + S_h))]\} + S_v \tag{A.30}$$

where y represents the longitudinal or lateral force acting on the tyre, and x is either the slip ratio or the slip angle. The coefficients B_1, C, D, and E are the stiffness, shape, peak, and curvature factors, respectively. S_h and S_v are the horizontal and vertical shifts.

The longitudinal force generated by each tyre, denoted as $F_{x,ij}$, is expressed as:

$$F_{x,ij}(\lambda_{ij}) = D\sin\left\{C\arctan\left[B_1(\lambda_{ij} + S_h)(1 - E) + E\arctan\left(B_1(\lambda_{ij} + S_h)\right)\right]\right\} + S_v \tag{A.31}$$

The characteristic parameters of the Magic Formula are determined empirically using the following relationships:

$$\begin{aligned} &C = b_0, \quad D = \mu(b_1 F_{z,ij}^2 + b_2 F_{z,ij}), \quad B_1 = \frac{b_3 F_{z,ij}^2 + b_4 F_{z,ij}}{CDe^{b_5 F_{z,ij}}}, \\ &E = b_6 F_{z,ij}^2 + b_7 F_{z,ij} + b_8, \quad S_h = b_9 F_{z,ij} + b_{10}, \quad S_v = 0 \end{aligned} \tag{A.32}$$

where b_0–b_{10} are fitting coefficients obtained from experimental data, and $F_{z,ij}$ denotes the vertical load on each tyre.

The slip ratio of each tyre is defined as:

$$\lambda_{ij} = \frac{v_{ij} - \omega_{ij} R_\omega}{v_{ij}}, \quad i \in \{f, r\},\ j \in \{l, r\} \tag{A.33}$$

where v_{ij} is the longitudinal velocity of the wheel centre, ω_{ij} is the wheel angular velocity, and R_ω is the effective tyre radius. The reference longitudinal velocities of the wheels are given by:

$$\begin{aligned} v_{fl} &= (v_x - \dot{\psi}\frac{\ell_w}{2})\cos\delta + (v_y + \dot{\psi}\ell_f)\sin\delta, \quad v_{rl} = v_x - \dot{\psi}\frac{\ell_w}{2}, \\ v_{fr} &= (v_x + \dot{\psi}\frac{\ell_w}{2})\cos\delta + (v_y + \dot{\psi}\ell_f)\sin\delta, \quad v_{rr} = v_x + \dot{\psi}\frac{\ell_w}{2} \end{aligned} \tag{A.34}$$

The vertical loads of wheels can be calculated:

$$F_{z,fl} = \frac{m}{\ell_f + \ell_r}\left(\frac{g\ell_r}{2} - \frac{\dot{v}_x h_g}{2} - \frac{\dot{v}_y h_g \ell_r}{\ell_w}\right) \tag{A.35}$$

$$F_{z,fr} = \frac{m}{\ell_f + \ell_r}\left(\frac{g\ell_r}{2} - \frac{\dot{v}_x h_g}{2} + \frac{\dot{v}_y h_g \ell_r}{\ell_w}\right) \tag{A.36}$$

$$F_{z,rl} = \frac{m}{\ell_f + \ell_r}\left(\frac{g\ell_f}{2} + \frac{\dot{v}_x h_g}{2} - \frac{\dot{v}_y h_g \ell_f}{\ell_w}\right) \tag{A.37}$$

$$F_{z,rr} = \frac{m}{\ell_f + \ell_r}\left(\frac{g\ell_f}{2} + \frac{\dot{v}_x h_g}{2} + \frac{\dot{v}_y h_g \ell_f}{\ell_w}\right) \tag{A.38}$$

where h_g is the height of the COG. Similarly, the lateral tyre force $F_{y,ij}$ is calculated using:

$$F_{y,ij}(\alpha_{ij}) = D\sin\left\{C\arctan\left[B_1(\alpha_{ij} + S_h)(1-E) + E\arctan\left(B_1(\alpha_{ij} + S_h)\right)\right]\right\} + S_v \tag{A.39}$$

with the corresponding parameters defined as:

$$\begin{aligned} &C = a_0, \quad D = \mu(a_1 F_{z,ij}^2 + a_2 F_{z,ij}), \quad B_1 = \frac{a_3 \sin[2\arctan(F_{z,ij}/a_4)]}{CD}, \\ &E = a_6 F_{z,ij} + a_7, \quad S_h = a_9 F_{z,ij} + a_{10}, \quad S_v = a_{12} F_{z,ij} + a_{13} \end{aligned} \tag{A.40}$$

The slip angles for each wheel are calculated as:

$$\begin{aligned} \alpha_{fl} &= \arctan\left(\frac{v_y + \omega_z \ell_f}{v_x - \omega_z(\ell_w/2)}\right) - \delta, \quad \alpha_{rl} = \arctan\left(\frac{v_y - \omega_z \ell_r}{v_x - \omega_z(\ell_w/2)}\right), \\ \alpha_{fr} &= \arctan\left(\frac{v_y + \omega_z \ell_f}{v_x + \omega_z(\ell_w/2)}\right) - \delta, \quad \alpha_{rr} = \arctan\left(\frac{v_y - \omega_z \ell_r}{v_x + \omega_z(\ell_w/2)}\right) \end{aligned} \tag{A.41}$$

The longitudinal and lateral forces must satisfy the tyre–road adhesion limits, represented by the adhesion ellipse constraint:

$$\begin{aligned} F_{x,ij} &= \frac{|\sigma_{x,ij}|}{\sigma_{ij}} y(x), \quad F_{y,ij} = \frac{|\sigma_{y,ij}|}{\sigma_{ij}} y(x), \\ \sigma_{ij} &= \sqrt{\sigma_{x,ij}^2 + \sigma_{y,ij}^2}, \quad \sigma_{x,ij} = \frac{\lambda_{ij}}{1+\lambda_{ij}}, \quad \sigma_{y,ij} = \frac{\tan\alpha_{ij}}{1+\lambda_{ij}} \end{aligned} \tag{A.42}$$

This constraint ensures that the combined longitudinal and lateral forces do not exceed the maximum friction capability of the tyre–road interface.

Fig. A.7 shows the nonlinear behavior of the longitudinal and lateral tire forces for different tire-road friction coefficients ($\mu = \{0.3, 0.5, 0.7, 0.9\}$), which represent conditions ranging from low-adhesion surfaces to dry asphalt. Fig. A.7(a) illustrates the longitudinal tire force F_x as a function of the longitudinal slip ratio, and Fig. A.7(b) demonstrates the lateral force (F_y) versus slip angle. These plots confirm that the MF-based tire model replicates the expected nonlinear characteristics of tire behavior and aligns with physical trends reported in the literature. The tire forces increase with greater friction levels μ. As the slip ratio or slip angle grows, the forces rise gradually and then level off, showing the typical grip and slip phases of tire behavior. This model captures how the forces depend on both slip ratio and slip angle, making it reliable for simulating various driving situations like accelerating, braking, and turning on different road surfaces.

A.5 Vehicle Dynamic Control

Vehicle dynamic control systems are designed to improve the stability, handling, and overall safety of a vehicle under various driving conditions. Torque vectoring is one of the most

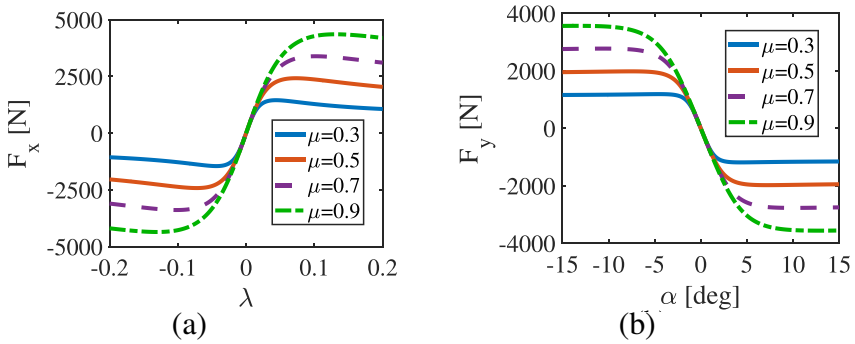


Fig. A.7 Tire model characteristics for various friction levels: (a) Longitudinal force versus slip ratio, (b) Lateral force versus slip angle.

effective approaches for achieving these objectives. By actively distributing the driving and braking torques among the wheels, torque vectoring enables direct control of the yaw moment of the vehicle and lateral dynamics. Conventional torque vectoring control architectures typically consist of two main layers of high-level and low-level controllers. The high-level controller calculates the required corrective yaw moment based on the difference between the desired and actual vehicle states, and the low-level controller allocates the corresponding torques among the four wheels to generate the commanded yaw moment in an optimal manner.

A.5.1 High-Level Controller

The high-level controller determines the vehicle response and generates the corresponding control commands to achieve it. In this work, an LQR is employed as the high-level controller. The LQR aims to minimise a quadratic cost function that balances the trade-off between state deviations and control effort, thereby ensuring optimal performance in terms of stability and control precision [95]. As expressed previously in Eq. (A.17), the state-space representation of the vehicle dynamics with 2 DOF can be written as:

$$\begin{bmatrix} \dot{\beta}_d \\ \ddot{\psi}_d \end{bmatrix} = A \begin{bmatrix} \beta_d \\ \dot{\psi}_d \end{bmatrix} + B\delta \tag{A.43}$$

where δ is the steering angle and v_x is the vehicle longitudinal speed. To account for stability control, an additional term representing the corrective yaw moment is included in the actual system equation:

$$\begin{bmatrix} \dot{\beta} \\ \ddot{\psi} \end{bmatrix} = A \begin{bmatrix} \beta \\ \dot{\psi} \end{bmatrix} + B\delta + B^0 M_z \tag{A.44}$$

where $B^0 = [0 \quad 1/I_z]^T$, and I_z is the yaw moment of inertia. By subtracting Eq. (A.43) from Eq. (A.44), the dynamic relationship between the corrective yaw moment and the deviation of vehicle motion states can be obtained as:

$$\begin{bmatrix} \Delta\dot{\beta} \\ \Delta\ddot{\psi} \end{bmatrix} = A \begin{bmatrix} \Delta\beta \\ \Delta\dot{\psi} \end{bmatrix} + B^0 M_z \tag{A.45}$$

where $\Delta\dot{\psi}$ and $\Delta\beta$ represent the errors between the desired and actual yaw rate and sideslip angle, respectively. The LQR controller minimises the following cost function:

$$J = \int_0^\infty \left(x^T Q x + u^T R u \right) dt \tag{A.46}$$

where x is the state vector, u is the control input, Q is a positive semi-definite weighting matrix that penalises state deviations, and R is a positive definite weighting matrix that penalises control effort. Increasing the elements of Q gives higher priority to minimising state errors, whereas increasing R penalises excessive control effort. The optimal control law that minimises the cost function is expressed as:

$$u = Kx \tag{A.47}$$

where K is the optimal feedback gain matrix, calculated as:

$$K = -R^{-1} B^T P \tag{A.48}$$

The matrix P is obtained by solving the continuous-time Algebraic Riccati Equation [192]:

$$-PA - A^T P + PBR^{-1}B^T P - Q = 0 \tag{A.49}$$

Finally, the corrective yaw moment command generated by the high-level controller is computed as:

$$M_z = Kx(t) = k_1(\beta - \beta_{des}) + k_2(\dot{\psi} - \dot{\psi}_{des}) \tag{A.50}$$

where k_1 and k_2 are the LQR feedback gains associated with the sideslip angle and yaw rate errors, respectively. This formulation ensures that the vehicle maintains stability and accurately follows the desired trajectory under various driving conditions.

A.5.2 Low-Level Controller

The low-level controller is responsible for distributing the torques among the four wheels based on the corrective yaw moment generated by the high-level controller. In this research, a vertical load dynamic distribution algorithm is employed to allocate the torque to maintain stability and balance [193, 184]. The individual wheel torques are calculated using the corrective yaw moment obtained from the high-level controller as follows:

$$T_{fl} = \frac{R_w}{1+\frac{1}{\rho_L}}\left(\frac{F_{x,t}}{2} - \frac{M_z}{\ell_a}\right) \tag{A.51}$$

$$T_{fr} = \frac{R_w}{1+\frac{1}{\rho_R}}\left(\frac{F_{x,t}}{2} + \frac{M_z}{\ell_a}\right) \tag{A.52}$$

$$T_{rl} = \frac{R_w}{1+\rho_L}\left(\frac{F_{x,t}}{2} - \frac{M_z}{\ell_a}\right) \tag{A.53}$$

$$T_{rr} = \frac{R_w}{1+\rho_R}\left(\frac{F_{x,t}}{2} + \frac{M_z}{\ell_a}\right) \tag{A.54}$$

where T_{ij} is the torque applied to each wheel ($ij \in \{fl, fr, rl, rr\}$), M_z is the corrective yaw moment, ℓ_a is the axle tread, $F_{x,t}$ is the total longitudinal demand force, and R_w is the effective tyre radius. The parameters ρ_L and ρ_R represent the load distribution ratios between the front and rear wheels on the left and right sides of the vehicle, respectively, and are expressed as:

$$\rho_L = \frac{F_{z,fl}}{F_{z,rl}}, \quad \rho_R = \frac{F_{z,fr}}{F_{z,rr}} \tag{A.55}$$

where $F_{z,ij}$ denotes the vertical tyre forces, obtained from the 7-DOF vehicle model.

This hierarchical approach allows the low-level controller to ensure optimal torque allocation while satisfying the yaw moment demand generated by the high-level controller. It enables effective coordination between longitudinal and lateral dynamics, thereby improving overall vehicle stability and control performance.

Appendix B

Monte Carlo Robustness Analysis of the TD3-Based Controller

This appendix presents a Monte Carlo based robustness analysis of the proposed TD3-based torque vectoring controller. The purpose of this analysis is to evaluate the sensitivity of the controller performance to uncertainties in vehicle parameters and operating conditions. In practical vehicle systems, parameters may deviate from their nominal values due to modelling inaccuracies, manufacturing tolerances, payload variations, and environmental factors such as road surface conditions. Consequently, a controller designed using nominal parameters must be evaluated under parameter variations to ensure that it can maintain stable and reliable vehicle behaviour in real-world operation.

Monte Carlo simulation is a widely adopted stochastic analysis method for evaluating system robustness under uncertainty. In this approach, uncertain parameters are modelled as random variables and multiple simulations are performed using randomly generated parameter sets. The resulting system responses are then statistically analysed to assess the stability and performance of the controller across a range of possible operating conditions. The Monte Carlo estimate of a performance metric can be expressed as

$$\hat{J} = \frac{1}{N} \sum_{i=1}^{N} J(\theta_i) \tag{B.1}$$

where N represents the number of simulation runs, θ_i denotes the randomly sampled parameter vector for the i-th simulation, and $J(\theta_i)$ represents the performance metric obtained from that simulation. As the number of simulations increases, the statistical estimates converge toward the true distribution of the system response.

Table B.1 Uncertain parameters used in the Monte Carlo robustness analysis

Parameter	Symbol	Nominal	Min of range	Max of range	Distribution type
Vehicle mass	m	1411 kg	1270 kg	1552 kg	Truncated normal
Yaw moment of inertia	J	2031.4 kg m^2	1828 kg m^2	2234 kg m^2	Truncated normal
Front axle distance	ℓ_f	1.04 m	0.988 m	1.092 m	Uniform
Rear axle distance	ℓ_r	1.56 m	1.482 m	1.638 m	Uniform
Track width parameter	ℓ_w	1.48 m	1.406 m	1.554 m	Uniform
Effective wheel radius	R_w	0.30 m	0.285 m	0.315 m	Uniform
Front cornering stiffness	C_{af}	45000 N/rad	36000 N/rad	54000 N/rad	Truncated normal
Rear cornering stiffness	C_{ar}	45000 N/rad	36000 N/rad	54000 N/rad	Truncated normal
Height of COG	h_g	0.54 m	0.486 m	0.594 m	Truncated normal
Wheel inertia	J_w	1.46 kg m^2	1.314 kg m^2	1.606 kg m^2	Truncated normal
Vehicle speed	v	20 m/s	15 m/s	25 m/s	Uniform
Tyre–road friction coefficient	μ	0.5	0.3	0.9	Uniform

In this study, a total of 1000 Monte Carlo simulations are performed to obtain statistically meaningful results. During each run, several vehicle parameters and environmental variables are randomly sampled within predefined uncertainty ranges around their nominal values. These parameters include the vehicle mass, yaw moment of inertia, geometric dimensions, tyre cornering stiffness, aerodynamic drag coefficient, rolling resistance coefficient, vehicle speed, and tyre–road friction coefficient. The uncertainty ranges are selected to represent realistic variations typically encountered in vehicle dynamics modelling and real-world vehicle operation. Table B.1 summarises the parameters considered in the robustness analysis, including their nominal values, uncertainty bounds, and assumed probability distributions.

For each simulation run, the nonlinear vehicle model is executed using a randomly generated parameter set, while the TD3-based controller remains unchanged. The controller performance is then evaluated using the same metrics defined in Chapter 3.8. These metrics include the maximum sideslip angle ($max|\beta|$), maximum sideslip rate ($max|\dot{\beta}|$), integral of sideslip angle ($\int|\beta|$), integral of sideslip rate ($\int|\dot{\beta}|$), integral of yaw rate tracking error ($\int|\dot{\psi}_e|$), and maximum lateral deviation from the desired path ($max|Y_e|$). These metrics collectively quantify vehicle stability, transient behaviour, and path-following performance.

Fig. B.1 illustrates the distribution of the performance metrics obtained from the 1000 Monte Carlo simulations. Each point in the plots corresponds to one simulation run with randomly sampled parameters. The dispersion of the points represents the sensitivity of the controller performance to parameter variations. Despite the introduced uncertainties, all performance metrics remain within bounded ranges, indicating that the TD3-based controller maintains stable and consistent vehicle behaviour.

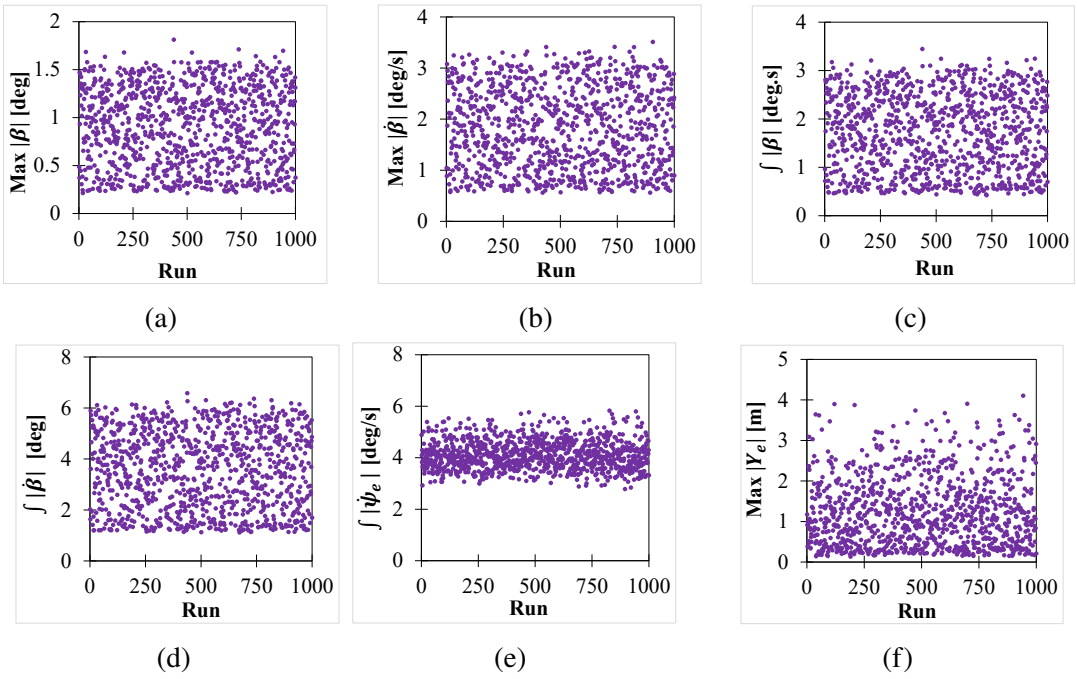
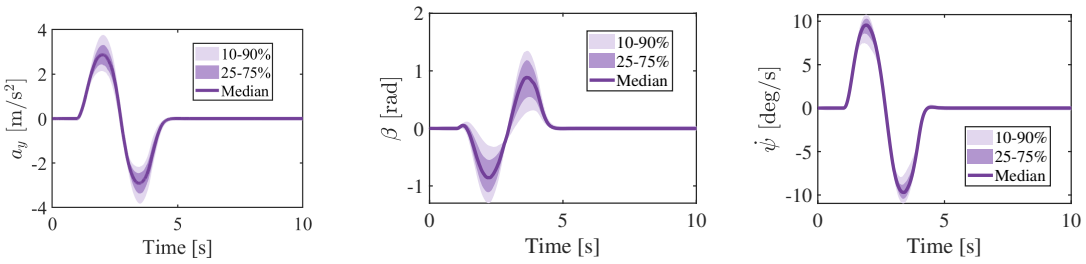


Fig. B.1 Monte Carlo simulation results under parameter uncertainty: (a) maximum sideslip angle ($max|\beta|$), (b) maximum sideslip rate ($max|\dot{\beta}|$), (c) integral of sideslip angle ($\int |\beta|$), (d) integral of sideslip rate ($\int |\dot{\beta}|$), (e) integral of yaw rate error ($\int |\dot{\psi}_e|$), and (f) maximum lateral deviation from the desired path ($max|Y_e|$).

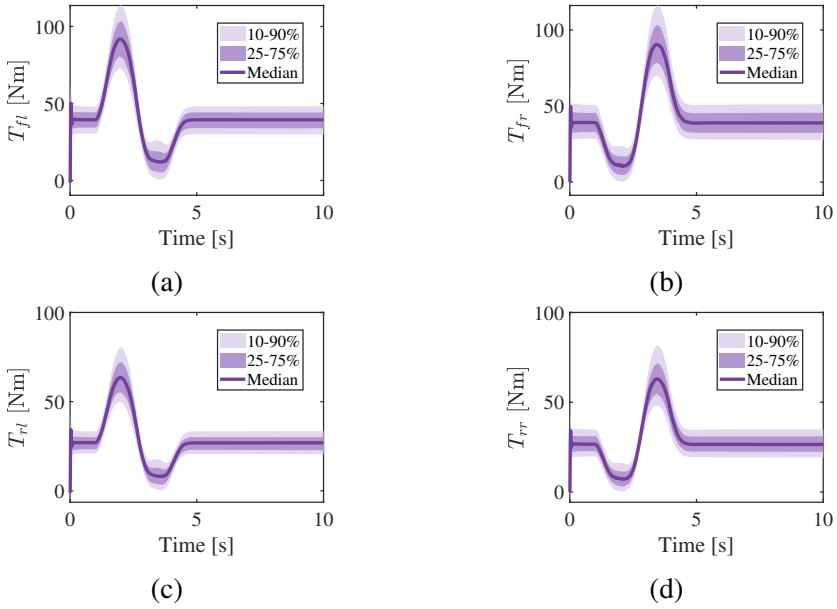
The maximum sideslip angle shown in Fig. B.1a remains confined within a relatively narrow range across all simulations, demonstrating that excessive lateral slip is effectively prevented even when vehicle parameters vary. Similarly, the maximum sideslip rate presented in Fig. B.1b exhibits limited variation, indicating that the controller preserves smooth transient dynamics under uncertain conditions. The integral-based metrics shown in Fig. B.1c, Fig. B.1d, and Fig. B.1e provide additional insight into the cumulative dynamic behaviour of the vehicle. The distributions of $\int |\beta|$ and $\int |\dot{\beta}|$ confirm that the overall lateral motion remains consistently bounded throughout the simulations. In addition, the integral of yaw rate error $\int |\dot{\psi}_e|$ indicates the controller maintains accurate yaw rate tracking despite the presence of parameter uncertainties. Fig. B.1f presents the maximum lateral deviation from the desired path across all simulation runs. Most simulations result in relatively small lateral errors, suggesting that the controller preserves reliable path-following performance under varying vehicle dynamics and road conditions. The overall spread of the results indicates that the TD3-based controller is not overly sensitive to moderate variations in vehicle parameters.

In addition to the distribution-based analysis, dispersion plots are used to illustrate the time-domain variability of key vehicle states across the Monte Carlo simulations. These plots



(a) (b) (c)

Fig. B.2 Dispersion of vehicle dynamic responses under Monte Carlo simulations: (a) lateral acceleration, (b) sideslip angle, and (c) yaw rate.



(a) (b) (c) (d)

Fig. B.3 Dispersion of wheel torque responses under Monte Carlo simulations: (a) front-left torque, (b) front-right torque, (c) rear-left torque, and (d) rear-right torque.

visualise the statistical spread of the system responses at each time instant using percentile bands. Specifically, the median response together with the 25–75% and 10–90% percentile intervals are presented to illustrate both typical and extreme behaviour. Fig. B.2 shows the dispersion of the vehicle dynamic responses, including the lateral acceleration a_y, sideslip angle β, and yaw rate $\dot{\psi}$. The median curve represents the central tendency of the responses obtained from the simulations, and the shaded regions indicate the variability across different parameter realisations. The relatively narrow percentile bands demonstrate that the controller maintains consistent dynamic responses even when system parameters vary. Furthermore, Fig. B.3 presents the dispersion of the wheel torques generated by the torque vectoring controller. Although slight variations appear due to differences in the vehicle parameters

and operating conditions, the overall torque distribution pattern remains similar across the simulations. This behaviour indicates that the controller adapts the torque allocation in a stable and predictable manner, ensuring balanced torque distribution while maintaining vehicle stability.

Overall, the Monte Carlo analysis demonstrates that the proposed TD3-based torque vectoring controller exhibits robust performance across a wide range of parameter uncertainties and operating conditions. Both the statistical distributions of the performance metrics and the dispersion of time-domain responses confirm that the controller maintains stable vehicle dynamics, reliable path tracking, and consistent torque allocation behaviour even when the vehicle model deviates from its nominal configuration.