J. Appl. Comput. Mech., xx(x) (2025) 1-26
DOI: 10.22055/jacm.2025.48654.5398

ISSN: 2383-4536
jacm.scu.ac.ir

Shahid Chamran University of Ahvaz

Journal of Applied and Computational Mechanics

Research Paper

Modelling, Control and Optimisation of an Electric Torque Vectoring System in a Formula SAE Race Car

Arthur Julve, Aydin Azizi

School of Engineering, Computing and Mathematics, Oxford Brookes University, OX3 0BP, Oxford, UK

Received July 30 2025; Revised November 20 2025; Accepted for publication November 25 2025.
✉ Corresponding author: A. Azizi (aydin.azizi@brookes.ac.uk)

Abstract. This research presents the design, development, and optimization of a torque vectoring system for a Formula Student vehicle, aimed at improving dynamic stability and handling performance. The study was guided by four primary objectives: constructing an accurate and realistic vehicle model, identifying an optimal control strategy for torque distribution, ensuring operational safety, and establishing a scalable framework for future development and knowledge transfer within the team. Vehicle dynamics were modelled in Simulink and supported by data from the VSM lap time simulation software to replicate real-world behaviour. Three control strategies for torque distribution were implemented and evaluated: a Proportional-Integral-Derivative (PID) controller, a Sliding Mode Controller (SMC), and a Fuzzy Logic Controller (FLC). Each controller was carefully tuned and tested, with performance assessed by comparing the vehicle's yaw acceleration response against a reference signal. Among the three, the Fuzzy Logic Controller demonstrated the best overall performance in terms of accuracy, responsiveness, and stability, making it the most suitable candidate for the application. The outcome of this research provides a validated control strategy that enhances the dynamic behaviour of Formula Student vehicles. The system's modular design allows for future expansion, including real-world testing, integration with braking strategies, and the adoption of advanced techniques such as Artificial Neural Networks for adaptive control.

Keywords: Formula SAE; PID; Fuzzy Logic; Sliding Mode; Control; Torque vectoring.

1. Introduction

1.1. Background

In Formula SAE, as in any motorsport competition, all research and development efforts are ultimately directed toward one clear goal: outperforming the competition. Teams devote extensive resources each season to introducing innovative technologies that improve performance and reliability.

Within this context, the present study supports Oxford Brookes Racing's continuous pursuit of competitiveness through the integration of **torque-vectoring** techniques into the car's vehicle-control system. Torque vectoring enables precise manipulation of the torque delivered to each wheel, optimizing dynamic behaviour and reducing lap times. It is a computer-controlled system that modulates torque between left, right, front, and rear wheels [1].

Torque vectoring enhances stability by maintaining the desired yaw rate through corners and allocating torque to the wheels with the greatest available grip. The concept combines vehicle dynamics, power electronics, control engineering, and data analysis, influencing longitudinal, lateral, yaw, roll, and pitch motions. During high-performance driving, one or more tyres may reach the friction limit [2], making precise torque distribution essential to avoid instability. Figure 1 illustrates the effect of torque vectoring on cornering dynamics.

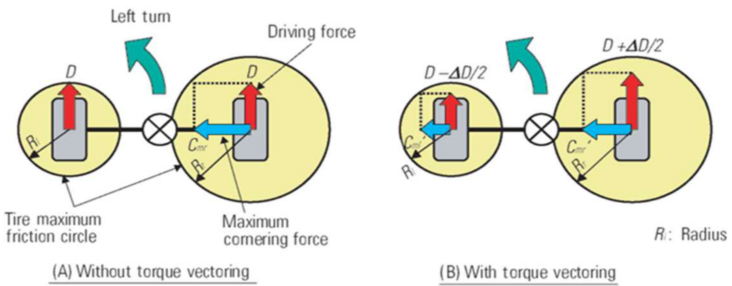

Fig. 1. Effect of torque vectoring [3].

Published online: December 03 2025

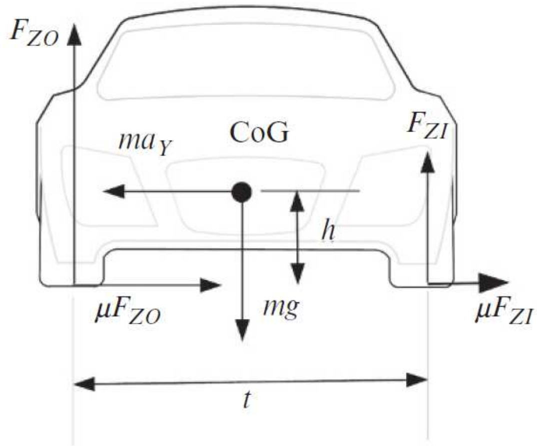

Fig. 2. Lateral weight transfer in a rigid chassis model [10].

Torque vectoring can be viewed as the evolution of the differential through electronic control. Historically, the influence of torque distribution on yaw moment and stability led to systems such as limited-slip differentials, Anti-lock Braking Systems (ABS), Traction Control Systems (TCS), and Electronic Stability Programs (ESP) [4]. Modern torque-vectoring control builds on these technologies to offer a more adaptable solution, particularly attractive for electric mobility.

While commercial implementations prioritise safety and comfort [5], motorsport applications focus on agility, dynamic response, and lap-time reduction. Properly designed torque-vectoring systems significantly enhance cornering stability and responsiveness, especially in electric vehicles with independently driven wheels.

1.2. Formulation of the problem

The development of an effective torque-vectoring system requires an understanding of vehicle-dynamics fundamentals. The yaw rate, representing the vehicle's rotation about its vertical axis, is a critical indicator of cornering behaviour [6]. Each corner has an ideal yaw rate; the closer the vehicle follows this target, the more stable and faster its cornering performance [7]. The yaw moment is determined by the longitudinal and lateral forces generated by each wheel, which depend on tyre characteristics and load distribution [8].

Tyre load is influenced by acceleration, suspension geometry, aerodynamic forces, and load transfers [9]. Figure 2 illustrates lateral load transfer in a rigid-chassis model, highlighting how dynamic load distribution affects control strategy design [10].

Realistic torque-vectoring design must capture both internal vehicle dynamics and external aerodynamic effects [11]. The degree of model fidelity directly determines control-system complexity.

Torque vectoring is applicable to multiple powertrain layouts including rear-, front-, and all-wheel-drive configurations [12]. In traditional combustion vehicles, true torque vectoring is complex, but electric powertrains with individual wheel motors simplify implementation and control precision [13].

The Oxford Brookes Racing electric vehicle features one motor per wheel, enabling direct yaw-moment control through independent torque modulation [14]. Selecting a suitable control strategy requires evaluating the advantages and limitations of available techniques to determine which offers the best balance between accuracy, robustness, and computational efficiency.

Recent studies have increasingly emphasized the importance of advanced control and modelling techniques for Formula Student vehicles. Mikuláš [15] introduced a model predictive torque vectoring control architecture for a Formula Student electric race car, demonstrating improved cornering performance through dynamic torque distribution. Similarly, Chatzikomis et al. [16] developed and validated a torque-vectoring control system for an autonomous electric racing vehicle, underscoring the potential of multi-motor architectures for stability and energy efficiency. These works provide valuable context for the present investigation, which extends similar control principles using a fuzzy logic framework optimized for Formula SAE applications.

1.3. Literature survey

A wide range of torque-vectoring control strategies have been proposed in the literature.

PID Control. PID controllers remain the most common due to simplicity and ease of tuning [17-20]. However, they struggle with nonlinearities and parameter uncertainties.

Sliding Mode Control (SMC). SMC offers robustness to modelling errors and disturbances by driving trajectories to a sliding surface [21-23]. Hameed et al. [24] introduced a backstepping-based quasi-SMC to reduce chattering and improve tracking under load perturbations.

Fuzzy Logic Control (FLC). FLC applies approximate reasoning to handle nonlinear or uncertain systems using rule-based inference [25-27]. Phan et al. [28] developed an adaptive fuzzy-PID controller for electric-vehicle cruise control that improved tracking accuracy and adaptability without detailed plant modelling.

Model Predictive Control (MPC). MPC predicts future states and optimises torque allocation while respecting constraints such as wheel slip and stability [29-32]. Its limitation is computational demand for real-time execution.

Artificial Neural Networks (ANNs). ANNs learn complex nonlinear patterns from data to predict optimal torque distributions [33, 34]. They achieve adaptive performance but require extensive training data and have limited interpretability.

Coordinated Control (RWS and TV). Coordinating multiple actuators such as rear-wheel steering (RWS) and torque vectoring (TV) improves stability and handling [35-37].

H-infinity Loop Shaping. $H\infty$ control minimises worst-case response and ensures stability under uncertainty [38-39].

Adaptive Second-Order Sliding Mode (ASOSM) with Backstepping. This hybrid method enhances robustness to matched and unmatched uncertainties [40, 41].

Further studies have validated these approaches using hardware- or software-in-the-loop setups. Lu et al. [39] implemented loop-shaping control on a four-motor EV achieving stable cornering. Vignati and Sabbioni [42] investigated coordinated RWS–TV control using nonlinear models. Zhu et al. [43] designed a fuzzy differential controller for electric buses using an unscented Kalman filter. Kim and Kim [31] proposed nonlinear MPC methods improving energy efficiency by 2–3 percent while maintaining stability. Dendaluce Jahnke et al. [44] examined neural-network control for embedded platforms, identifying hardware trade-offs. Validation practices commonly employ multi-phase simulation environments such as AVL VSM, CarSim, or IPG CarMaker [45]. These tools enable iterative model refinement before track testing [46].

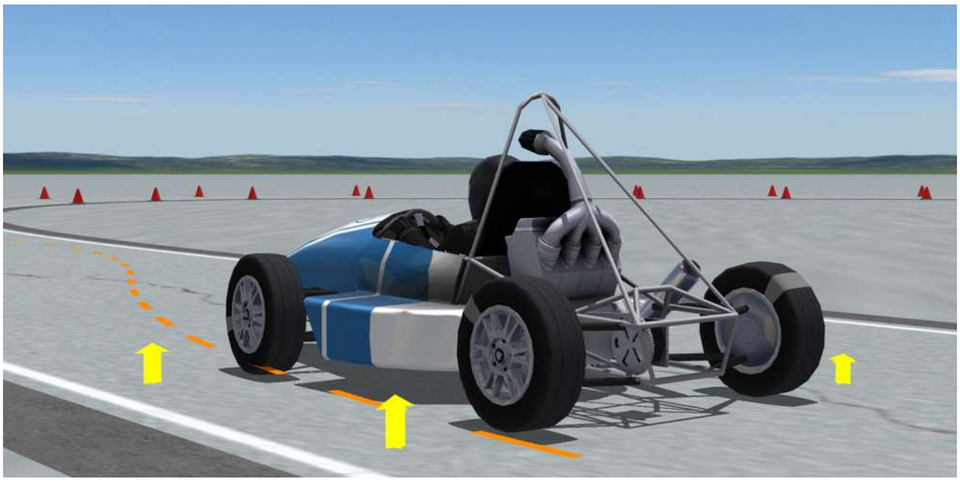

Fig. 3. FSAE car testing simulation [46].

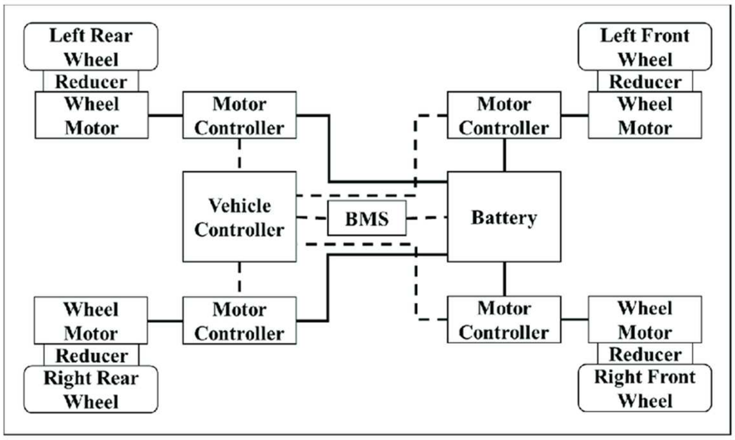

Fig. 4. Oxford Brookes Racing powertrain type [12].

Data acquisition and telemetry also play a key role in validating torque-vectoring performance. Typical FSAE workflows incorporate on-track data logging and sensor feedback for calibration and verification [47-49]. These practices underpin both simulation-in-the-loop and hardware-in-the-loop development stages.

In addition to these approaches, recent advancements in vehicle modeling and simulation have expanded the analytical capabilities of Formula SAE design teams. Regina et al. [50] presented a comprehensive Simscape Multibody model of an autonomous Formula SAE race car, illustrating the integration of mechanical and control subsystems within a unified simulation framework. Earlier, Govender and Barton [51] performed multibody optimization of Formula SAE vehicle dynamics, demonstrating the influence of suspension geometry and load transfer on performance. Balena et al. [52] further investigated dynamic handling characteristics and setup optimization using detailed multi-body simulation, reinforcing the value of simulation-based validation for race car development. These contributions collectively support the use of integrated simulation and control design methods, as adopted in this paper, to enhance both vehicle stability and performance in competitive Formula SAE contexts. Figure 3 illustrates a typical FSAE car testing and simulation workflow, showing how simulation, validation, and track evaluation interact.

1.4. Scope and contribution of this study

This research develops and evaluates three torque-vectoring controllers (PID, SMC, and FLC) for an electric Formula SAE race car using MATLAB/Simulink integrated with AVL VSM data. The Oxford Brookes Racing vehicle employs a four in-wheel-motor configuration, enabling independent torque control at each wheel [12]. Figure 4 shows the powertrain layout used in this study, which enables full control of longitudinal and yaw dynamics [12].

The main contributions of this work are summarized as follows:

1. Development of a high-fidelity vehicle dynamics model that accurately represents tire-load transfer, yaw-rate behaviour, and transient response.
2. Design, tuning, and performance evaluation of three control algorithms under identical conditions for fair comparison.
3. Quantitative analysis based on RMS yaw-rate error, response time, settling time, and robustness to torque disturbances.
4. Verification of lap-time improvement through optimized torque distribution using the FLC strategy.

The results demonstrate a 33 percent reduction in yaw-rate error and a 3.1 percent improvement in simulated lap time, confirming the effectiveness of the FLC-based torque vectoring system. The modular Simulink framework developed for this research can be extended for hardware-in-the-loop testing and real-vehicle validation.

Compared to existing studies, this work provides three distinctive contributions. First, while prior investigations examined torque-vectoring strategies individually, few have offered a direct, quantitative comparison among classical (PID), robust (SMC), and intelligent (FLC) controllers under a unified, validated Formula SAE environment. Second, this study uniquely integrates MATLAB/Simulink control design with AVL VSM validation, ensuring that the simulation model replicates real vehicle dynamics with high fidelity. Third, the focus on yaw-acceleration tracking and its link to lap-time improvement introduces a performance-based validation criterion rarely addressed in previous work. These contributions collectively advance the understanding of torque-vectoring control in high-performance electric vehicles and demonstrate the applicability of adaptive, knowledge-based control to real-time racing environments.

1.5. Organization of the paper

Following this introduction, Section 2 describes the methodology, including vehicle modeling, controller design, and validation. Section 3 presents the simulation results and the quantitative comparison of the three control strategies, including robustness and lap-time analysis. Section 4 summarizes the main findings, discusses the study's limitations, and suggests directions for future work.

2. Methodology

Vehicle dynamics represent a highly complex field of study, involving numerous variables and interdependent factors. Since this research is conducted by a Formula Student team, a number of simplifying assumptions must be made. While these assumptions reduce precision, they facilitate a clearer understanding of the system and enable faster resolution of the fundamental equations.

The first assumption involves calculating reference values for yaw rate and yaw acceleration. This setpoint, derived from the steering radius formula, corresponds to the widest turning radius and is assumed to be the most stable and fastest method for navigating any corner. While this is generally valid, other factors must also be considered. For instance, the driver plays a crucial role, some drivers may prefer a less stable car that they can rotate more quickly, potentially resulting in faster lap times. Furthermore, the widest turning radius may not always yield the fastest path through a series of corners. Although it might be optimal for a single corner, drivers often adopt strategies that sacrifice the ideal racing line across consecutive turns to maintain higher speeds leading into a long straight. Nonetheless, such considerations are excluded from this research for the sake of simplicity.

To simplify vehicle dynamics calculations, a second widely accepted assumption, known as the rigid chassis approach, is applied. This method treats the vehicle's mass as a single unit, ignoring the effects of the suspension system on tire loads and not distinguishing between sprung and unsprung mass. Although not entirely accurate, this approach significantly reduces computational complexity and development time, making the system easier to understand and implement.

A third assumption is made when distributing torque based on estimated tire load. In this study, torque is allocated under the assumption that grip is directly proportional to load. However, this relationship is inherently non-linear, and load is not the only factor influencing grip. Variables such as tire temperature, wear, and compound type also play significant roles, yet they are difficult to estimate without additional data. Therefore, a linear relationship is assumed for the purpose of designing the torque allocation strategy.

Lastly, the simulation conducted using the lap time simulation tool assumes ideal data acquisition. This means all necessary sensors are correctly installed, data is complete, and there is no signal noise or interference. In reality, such ideal conditions are unlikely, especially in a vehicle experiencing numerous internal movements and varied signal types across different terrains. However, addressing these complexities is beyond the scope of this research and is therefore omitted.

To create an effective plan and understand the rationale behind key decisions, it is essential to recognize the limitations of this research.

First, the mathematical assumptions used inevitably introduce limitations in accuracy. Many of the equations are approximations, and several component values are estimated. As a result, it is expected that the final validation results may slightly deviate from real-world measurements.

Another limitation is the restricted access to physical testing. Organizing a test day requires not only the readiness of the torque vectoring system but also the complete preparation of the vehicle, availability of a suitable test site, and sufficient personnel to run the event. Due to these logistical constraints, teams often combine several upgrades into a single testing session to maximize efficiency. Given the frequency of tests required, the development of the torque vectoring system cannot depend on waiting for ideal testing conditions. As a result, all testing for this research is conducted through lap time simulation tools.

Conducting this research on an existing or developing car introduces further limitations regarding components and procedures. First, the powertrain topology is restricted to four electric in-wheel motors, which limits the scope for testing torque vectoring across different configurations. Second, the control software must follow a fixed development path: it is designed entirely in Simulink and then translated into C code for implementation on the ECU. This workflow restricts the use of alternative programming languages that might be more effective for development and testing.

The schematic of the torque vectoring control system is composed of six distinct blocks. As shown in Fig. 5, the development of the feedback control system begins with the Input Parameters Block, which defines the key variables influencing vehicle yaw dynamics. Prior to simulation and control design, a comprehensive Vehicle Study is conducted to internalize the vehicle's behaviour, this involves reviewing component documentation, analysing vehicle dynamics, and consulting domain experts within the team.

The next stage, Vehicle Modelling, is implemented in Simulink and captures crucial elements such as longitudinal and lateral load transfer, steering ratio, and tire behaviour, thereby establishing a robust foundation for the rest of the system. After this, the Yaw Rate Reference Block estimates the desired yaw acceleration based on linear velocity, steering angle, and vehicle parameters, producing a yaw rate target for optimal cornering.

This signal then proceeds into the Closed-Loop System, which consists of both feedforward and feedback paths. In the Feedforward Path, a controller processes the yaw rate error, feeding into the Torque Distribution Block, where torque is allocated based on tire loads, throttle input, and steering angle to achieve the desired yaw response. Concurrently, the Feedback Path includes a Yaw Rate Recalculation Block, where actual yaw acceleration is derived from torque outputs and tire models. This recalculated value is compared with the setpoint to trigger corrective actions.

Initial system validation is carried out using the VSM Event Simulation in AVL software, replicating a Formula Student endurance track to provide realistic vehicle data for integration in Simulink. With this data, Controller Design is executed, testing multiple strategies to evaluate closed-loop performance. For safety, a Failsafe Block is integrated to monitor the system for potentially hazardous conditions. If needed, torque vectoring is disabled, returning full control to the driver. Finally, System Validation is conducted to verify model accuracy, identify discrepancies, and refine the control strategy for reliable, robust performance. The functioning of each of these blocks is explained in depth in the following sections, along with their relevance to the overall system.

2.1. Input parameters

Although one of the final components developed in the process flowchart, the importation of data logged from the car's sensors is actually where the system begins. Due to time and resource constraints in Formula Student, physical testing is limited. Therefore, the decision was made to extract the required parameters using lap time simulation software.

For this purpose, AVL VSM was chosen. This software allows for the complete configuration of a car from scratch and its simulation on a track. Signals from all virtual sensors on the vehicle can be recorded and exported to an Excel file, which aligns with Simulink's capability to read Excel files using a Signal Builder Block. The implementation of this block is shown in Fig. 6.

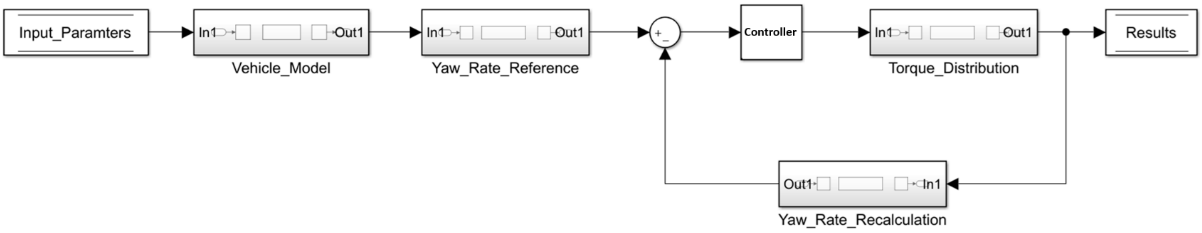

Fig. 5. Schematic of the torque vectoring control system.

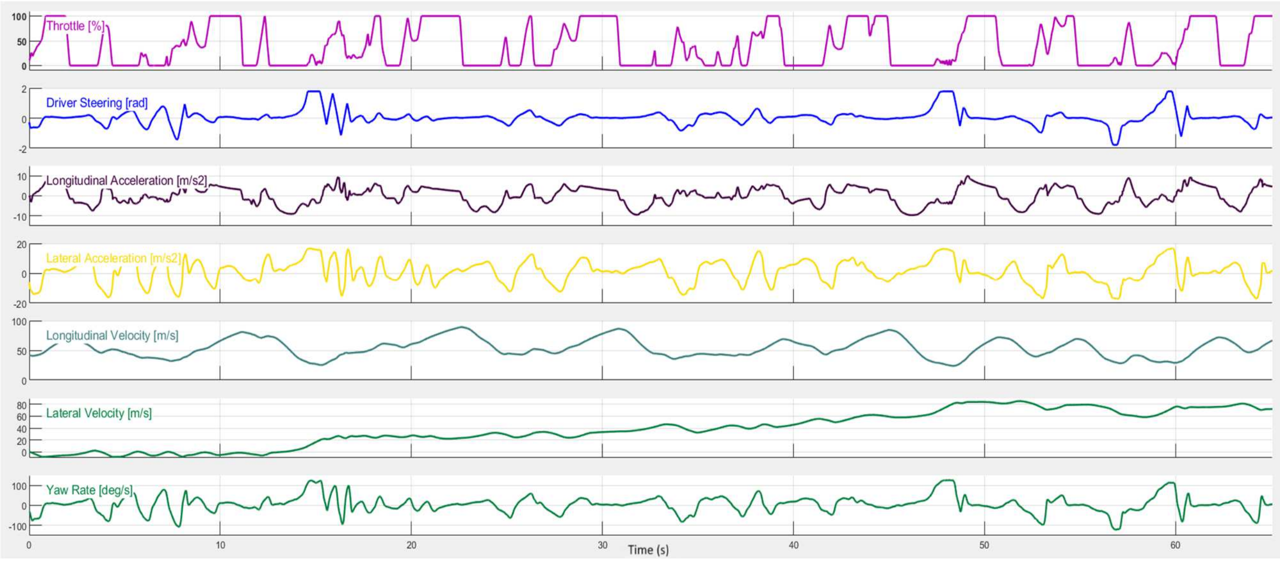

Fig. 6. Signal builder layout.

The track used for simulation was modelled based on the Formula Student UK competition track profile from the previous edition. Since this circuit is used for both the autocross and endurance events, it serves as a highly representative environment to evaluate vehicle dynamics and the impact of torque vectoring. After running the simulation, over 725 data channels can be accessed, each representing a specific measurement from the car. However, for this research, only a subset is retained, specifically, those relevant to torque vectoring and measurable with the sensors embedded in the current version of the single-seater.

2.2. Vehicle model

Some parameters required for the system are not directly measured by sensors and must be derived from the imported signals. This is where the vehicle model developed in Simulink plays a crucial role. It uses various vehicle parameters to estimate two main quantities: the vertical load on each tire at every moment and the steered angle of the front wheels. These values are later used in the control logic to determine the torque to be applied to each motor.

For load transfer calculations, the **rigid chassis assumption** is used treating the car's mass as a single unit, without differentiating between sprung and unsprung mass. The vertical load is calculated using Eqs. (1) and (2) for longitudinal and lateral load transfer [10]:

$$\text{Longitudinal Load Transfer} = \frac{h}{l} \cdot m \cdot a_x \tag{1}$$

$$\text{Lateral Load Transfer} = \frac{h}{t} \cdot m \cdot a_y \tag{2}$$

Longitudinal load transfer refers to weight shifting between the front and rear axles during acceleration or braking, caused by the moment generated around the centre of gravity. Lateral load transfer, on the other hand, describes weight shifting between the left and right sides during cornering. Both are proportional to the centre of gravity height (h), vehicle mass (m), and either longitudinal (a_x) or lateral (a_y) acceleration. These load transfers are then added to or subtracted from the static tire loads to determine the instantaneous vertical load on each wheel. This information is used by the torque vectoring system to allocate torque according to the available grip.

2.3. Yaw acceleration setpoint

With the relevant vehicle parameters defined, a yaw acceleration reference is calculated. Yaw rate refers to the speed at which the car rotates around its vertical (z) axis. This value is optimized to achieve neutral steering, meaning the smallest possible turning radius for a given velocity, which corresponds to optimal performance. This target behaviour can later be tuned based on driver preference. To align the desired yaw rate with the torque and forces exerted by the tires, the yaw rate is differentiated to obtain the corresponding angular acceleration.

The optimal rate of direction change is defined as the vehicle's longitudinal velocity divided by the corner radius. However, since the system cannot directly measure the corner radius unless explicitly provided, it is instead estimated based on driver input, specifically, the steering angle. The resulting Eq.3 accounts for the steering angle (δ), the understeer gradient (K), and the wheelbase (l), as outlined by [10]:

$$\dot{\gamma}_{desired} = \ tan\ \delta\ \cdot \frac{v_{CG}}{l \cdot (1 + K \cdot {v_{CG}}^2)} \tag{3}$$

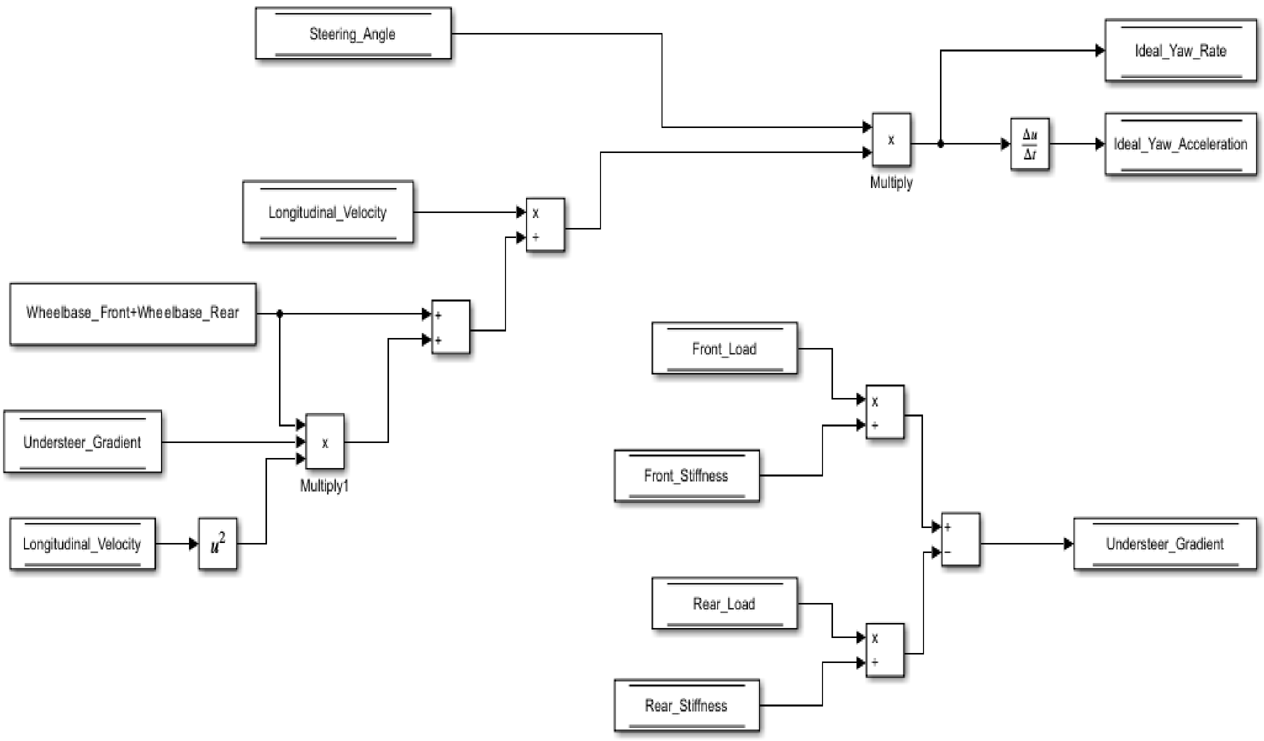

Fig. 7. Yaw acceleration reference block.

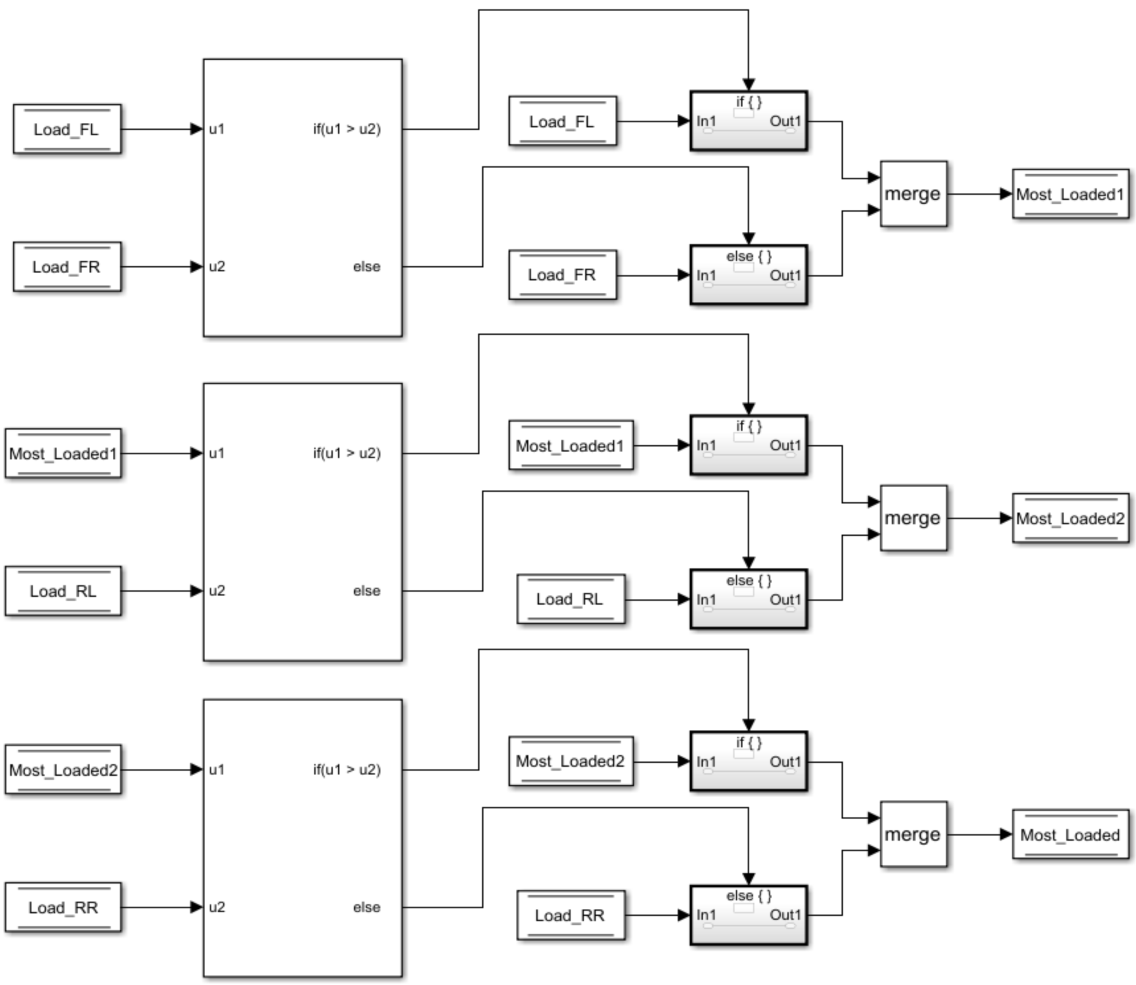

Fig. 8. Most loaded tyre identification.

The understeer gradient is a value derived from the axle loads and the cornering stiffness of the tires, as shown in Eq. (4). It quantifies how much the steering response deviates from the ideal Ackermann geometry [10]. A positive K indicates that the vehicle requires more steering input than an Ackermann-steered car to maintain the same cornering radius, or with the same steering input, it will follow a larger radius. Conversely, a negative K results in the opposite behaviour:

$$K = \left(\frac{F_{ZF}}{C_F} - \frac{F_{ZR}}{C_R}\right) \tag{4}$$

All these relationships are implemented as a separate block in Simulink, illustrated in Fig. 7. At each time step, the system records the steering angle, longitudinal velocity, and understeer gradient, and then calculates the ideal yaw rate accordingly.

2.4. Torque allocation strategy

The next step involves designing a torque allocation strategy, which defines the parameters used to distribute torque among the four wheels. The key inputs include the driver's throttle command, the maximum power allowed by both the engine and Formula Student (FS) regulations, and the vertical load at each corner of the car, representing the available grip at each tire.

In the initial stage, the system is configured as open-loop, with the yaw setpoint disregarded. The torque for each wheel is calculated using the same formula (Eq. (5)), which applies equally to all four wheels:

$$\text{Maximumtorque} \cdot \text{Throttle pedal value} \cdot \frac{\text{Tyreload}}{\text{Mostloadedtyre}} = \text{Requested torque} \tag{5}$$

The first term in the formula is the maximum allowable torque, which sets an upper limit that must not be exceeded. Since the other terms are expressed as percentages, the final torque output will not surpass this defined maximum. In Formula Student, this limit is set to 21 Nm per wheel, and this value is incorporated into the model.

The second term reflects the throttle input from the driver. It is normalized such that full throttle equals 1. If the value is less than 1, it indicates that full torque is not required, and only a proportional fraction of the maximum torque is applied.

Finally, the third term accounts for tire load. Each wheel's vertical load is compared to that of the most heavily loaded tire and expressed as a fraction. The most loaded tire receives a value of 1, allowing it to transmit the maximum torque. Other tires receive torque proportional to their respective loads. The most loaded wheel is identified through a series of comparisons, as illustrated in Fig. 8.

To enable individualized torque distribution, each in-wheel motor is assigned a unique torque value based on a proportional relationship among the maximum available torque, the driver's throttle input, and the grip potential of each wheel. This formulation allows dynamic torque adjustment in response to real-time driving conditions and is implemented as shown in the Simulink block diagram in Fig. 9.

The next phase in system development introduces a closed-loop control strategy that incorporates yaw dynamics. Specifically, the vehicle's actual yaw acceleration is continuously compared with the ideal yaw acceleration generated by a reference model. The resulting yaw error is processed by a controller and integrated into the torque allocation algorithm. Depending on each wheel's lateral position (left or right), the controller adjusts either the maximum torque or a residual torque value to correct the vehicle's trajectory. The mathematical expressions governing this torque distribution mechanism are provided in Eqs. (6) and (7):

$$\text{Controller Output} \cdot \ \text{Throttle pedal value} \cdot \frac{\text{Tyre load}}{\text{Most loaded tyre}} = \text{Requested Torque Right Wheels} \tag{6}$$

$$(\text{Maximum torque} - \text{Controller Output}) \cdot \ \text{Throttle pedal value} \cdot \frac{\text{Tyre load}}{\text{Most loaded tyre}} = \text{Requested Torque Left Wheels} \tag{7}$$

This enhancement necessitates a modification of the existing torque allocation block, as illustrated in the updated Simulink architecture in Fig. 10.

2.5. Yaw acceleration recalculation

As one of the final components of the system, the estimation of the actual yaw acceleration is performed. In a real-world scenario, this data would be provided by an onboard sensor. However, due to limited testing opportunities, a dedicated Simulink block has been developed to simulate this functionality. The formula used in the block is shown in Eq. (8) [7]:

$$\begin{aligned} I_z\dot{\psi} = l_f\left(cos\delta \cdot (F_{yFL} + F_{yFR}) + sin\delta \cdot (F_{xFL} + F_{xFR})\right) - l_r \cdot (F_{yRL} + F_{yRR}) + 0.5 \cdot t_f\left(cosdelta \cdot (F_{xFL} - F_{xFR}) - sindelta \cdot (F_{yFL} - F_{yFR})\right) + 0.5 \\ \cdot t_r \cdot (F_{xRL} - F_{xRR}) - M_{zFL} - M_{zFR} - M_{zRL} - M_{zRR} \end{aligned} \tag{8}$$

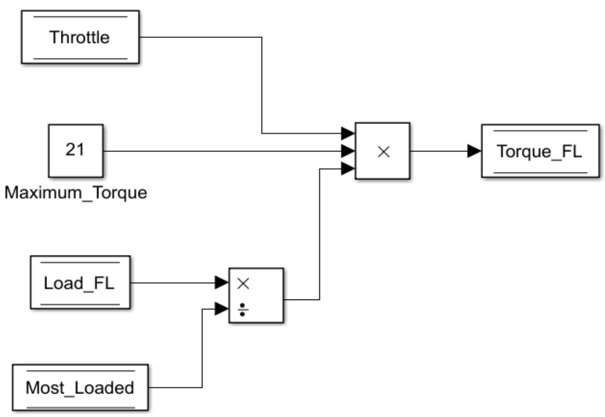

Fig. 9. Torque allocation for open loop.

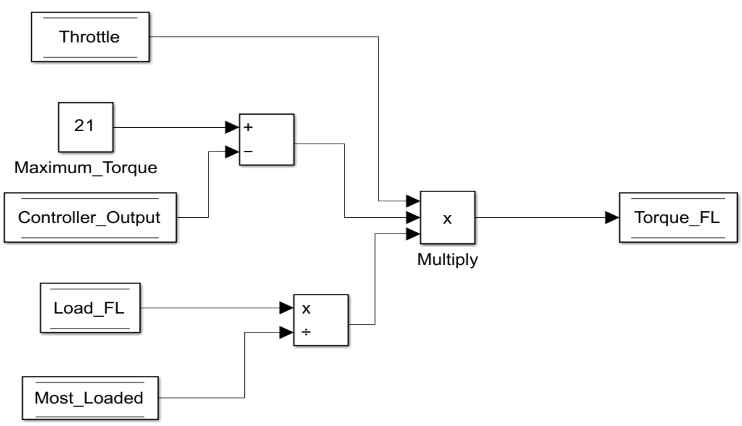

Fig. 10. Torque allocation for closed loop.

The principle behind this block is to replicate the yaw acceleration calculations applicable to any given vehicle. The formula begins by considering the longitudinal and lateral forces generated by the tires at any given moment. Using trigonometry, the yaw moment about the centre of gravity is then determined. The relationship between this yaw moment and the vehicle's moment of inertia around the z-axis yields the yaw acceleration.

The complexity of the formula increases due to the need to estimate tire forces, which are not constant. These forces depend on several factors, including tire type, the selected tire model, vertical load, and slip ratio or slip angle. To address this, tire data and curves have been replicated in Simulink using lookup tables. This data, obtained from prior testing and internal research, uses slip ratio or angle and tire load as inputs to produce longitudinal and lateral forces. These force outputs are then used in the yaw acceleration calculation.

2.6. Feedback control

The next phase of development, briefly introduced earlier, focuses on designing and calibrating the control strategy. The primary goal is to minimize the error between the actual yaw rate and the ideal yaw rate, calculated using a reference formula. Multiple approaches are considered to achieve this objective.

To ensure reproducibility and enable a fair comparison among control strategies, each controller was implemented with clearly defined tuning parameters and systematically calibrated within the Simulink environment. The tuning process for each method, including proportional-integral-derivative (PID), sliding mode (SMC), and fuzzy logic (FLC) controllers, is described in the following subsections together with the associated performance considerations.

To enable a consistent comparison among control strategies, each controller was expressed in standard mathematical form. The general objective is to minimize the yaw-acceleration error shown in Eq. (9):

$$e(t) = \ddot{\psi}_{ref}(t) - \ddot{\psi}(t), \tag{9}$$

where $\ddot{\psi}_{ref}$ is the desired yaw acceleration and $\ddot{\psi}(t)$ is the actual yaw acceleration measured or simulated in real time.

2.6.1. PID controller approach

The Proportional-Integral-Derivative (PID) controller minimizes the difference between the reference signal and the actual signal by producing an output that combines contributions from its proportional (P), integral (I), and derivative (D) terms. The proportional term outputs a value proportional to the current error. The integral term accumulates the error over time to eliminate steady-state error. The derivative term predicts future error based on its rate of change, helping to dampen oscillations [18].

With a clear understanding of the basic concepts, the PID controller is straightforward to implement in Simulink. As shown in Fig. 11, the controller receives the error as input, and its output, based on the defined coefficients, is strategically added to the torque allocation formula to achieve the desired vehicle behaviour.

The control signal generated by the PID controller shown in Eq. (10):

$$T_{PID}(t) = K_p e(t) + K_i \int_0^t e(\tau)\ d\tau + K_d \frac{de(t)}{dt} \tag{10}$$

where K_p, K_i and K_d are the proportional, integral, and derivative gains respectively. These parameters were tuned through two configurations: For quantitative assessment and reproducibility, two PID configurations were implemented. The first configuration employed the Simulink auto-tuning tool, with the following parameters obtained: K_p = 0.85, K_i = 0.30, and K_d = 0.12. This version was tuned for moderate robustness with a response time of approximately 2.0 seconds and a transient behaviour coefficient of 0.6.

To enhance responsiveness, a manually tuned "fast" version was developed, with K_p = 2.20, K_i = 0.80, and K_d = 0.25. This configuration reduced the settling time to 0.2 seconds and improved reference tracking, though at the cost of increased oscillations during rapid transitions. The first prioritizes stability, while the second improves responsiveness. Both configurations were evaluated to determine the optimal trade-off between responsiveness and stability.

Having the basic concepts clear the implementation of this controller is straight forward thanks to Simulink. The controller output is generated as illustrated in Fig. 11 where the difference between the signals is the input of the controller and the output is the result of the coefficients and the calculations. The output is strategically added to the torque allocation formula to obtain the desired behaviour.

2.6.2. Sliding mode controller approach

The second controller tested is the Sliding Mode Controller (SMC), available as a built-in block within Simulink. Like the PID controller, it uses two inputs one for the reference value and another for the actual value and produces an output. This output is then integrated into the torque allocation block to minimize yaw acceleration error.

SMC works by guiding the system state toward a predefined sliding surface that represents the desired behaviour. This surface is carefully designed based on the target dynamics, and the controller ensures that the system state converges to and remains on it. Once the state reaches the sliding surface, the system behaves as intended, effectively compensating for uncertainties or disturbances.

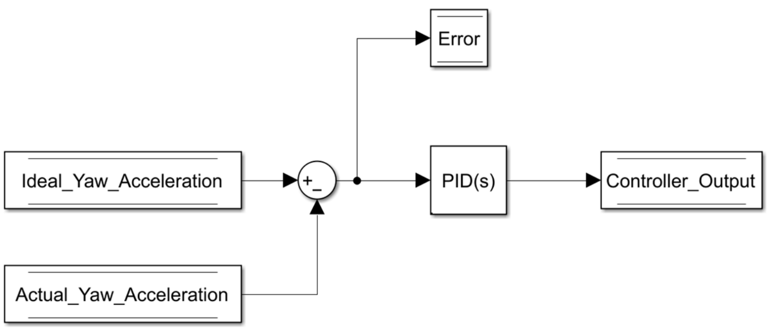

Fig. 11. PID controller implementation.

The SMC was defined by the sliding surface shown in Eq. (11):

$$S = c_1 e + c_2 \dot{e} \tag{11}$$

where $e = yaw_{\mathrm{ref}} - yaw_{\mathrm{actual}}$ and $\dot{e}$ is its time derivative. As shown in Eq. (12), the control law consists of two parts:

$$T_{SMC} = T_{eq} - k\,\mathrm{sign}(s) \tag{12}$$

where T_{eq} maintains motion along the sliding surface and the discontinuous term drives the system to it. To reduce chattering, the discontinuous switching term was replaced by a continuous saturation function (s/ϕ), with ϕ = 0.1 defining the boundary layer. The control limits were fixed between 0 and 21 Nm, and multiple hysteresis bands were tested to assess sensitivity and chattering behaviour. Three representative cases were analysed:

- Narrow hysteresis (0.1): high-frequency oscillations dominated the response.
- Medium hysteresis (10): moderate chattering with acceptable tracking.
- Wide hysteresis (100): degraded stability due to delayed switching.

The optimal configuration, selected for further analysis, used c_1 = 0.85, c_2 = 0.15, and a hysteresis of 10, which provided a good balance between stability and smoothness of control action. The control law of an SMC includes two components: equivalent control and switching control [53]. Equivalent control maintains the system on the sliding surface, while the switching control, typically discontinuous, drives the state toward the surface when it deviates. Although this approach offers high robustness, it can introduce high-frequency oscillations around the surface, known as "chattering," which may require additional mitigation techniques.

2.6.3. Fuzzy logic controller approach

The final control strategy explored is the Fuzzy Logic Controller (FLC). Fuzzy logic enables reasoning with uncertain or imprecise information, mimicking human decision-making [54]. Unlike classical binary logic, where variables are either true or false, fuzzy logic considers degrees of truth, offering greater flexibility in dynamic systems.

The FLC receives an input signal and produces a corrected output like other controllers. The setup begins with **fuzzification**, which converts crisp input values into fuzzy sets using membership functions that define varying degrees of truth [55]. In this study, the sole input is the error between the ideal and actual yaw acceleration values. Five triangular membership functions are defined to span an error range from -10 to 10:

- PH (Positive High): highest positive errors
- PL (Positive Low): lower positive errors
- Z (Zero): values near zero error
- NL (Negative Low): lower negative errors
- NH (Negative High): highest negative errors

These relationships are visually illustrated in Fig. 12. The Fuzzy Logic Controller was designed using a single-input, single-output configuration, where the input was the yaw acceleration error and the output was the corrective torque adjustment. Five triangular membership functions were defined for both input and output variables, enabling sufficient granularity while keeping the rule base computationally efficient.

Inference was carried out by the Mamdani method and defuzzified using the centroid technique shown in Eq. (13):

$$\Delta T = \frac{\sum_i \mu_i T_i}{\sum_i \mu_i} \tag{13}$$

where μ_i are the aggregated membership grades and T_i the corresponding torque levels. This configuration ensured smooth torque transitions and real-time performance.

- **Input range (Error):** ±3 rad/s²
  - Membership functions: Negative High (NH), Negative Low (NL), Zero (Z), Positive Low (PL), Positive High (PH)
- **Output range (Torque):** 0–21 Nm
  - Membership functions: Extra Low (XL), Low (L), Medium (M), High (H), Extra High (XH)

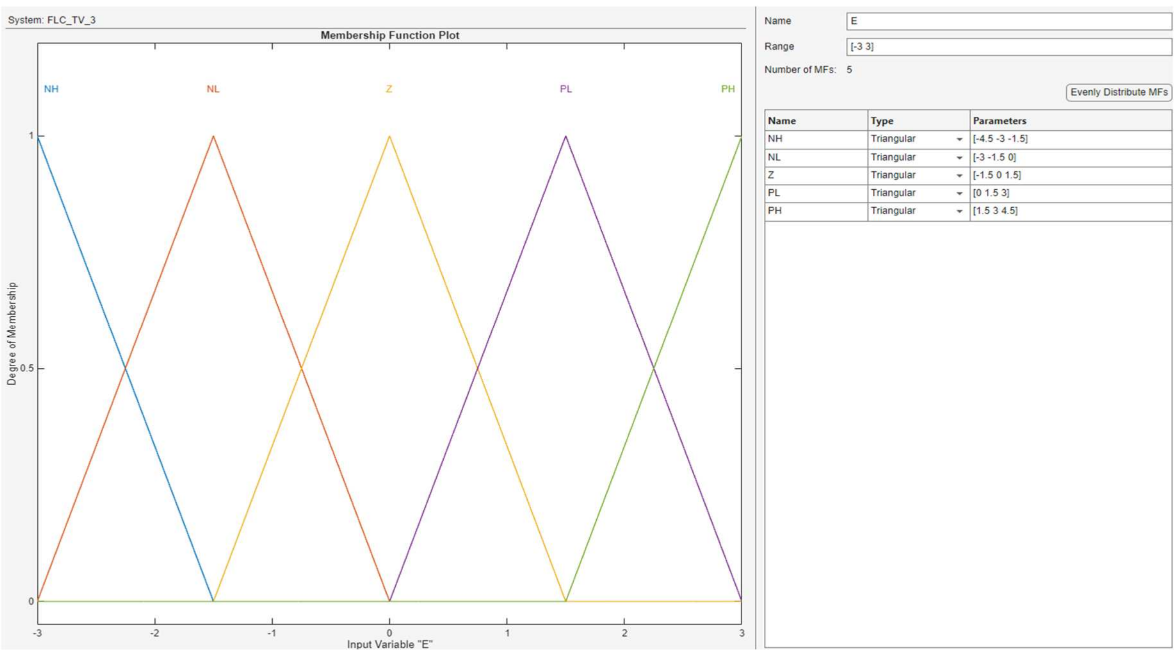

Fig. 12. Error membership functions.

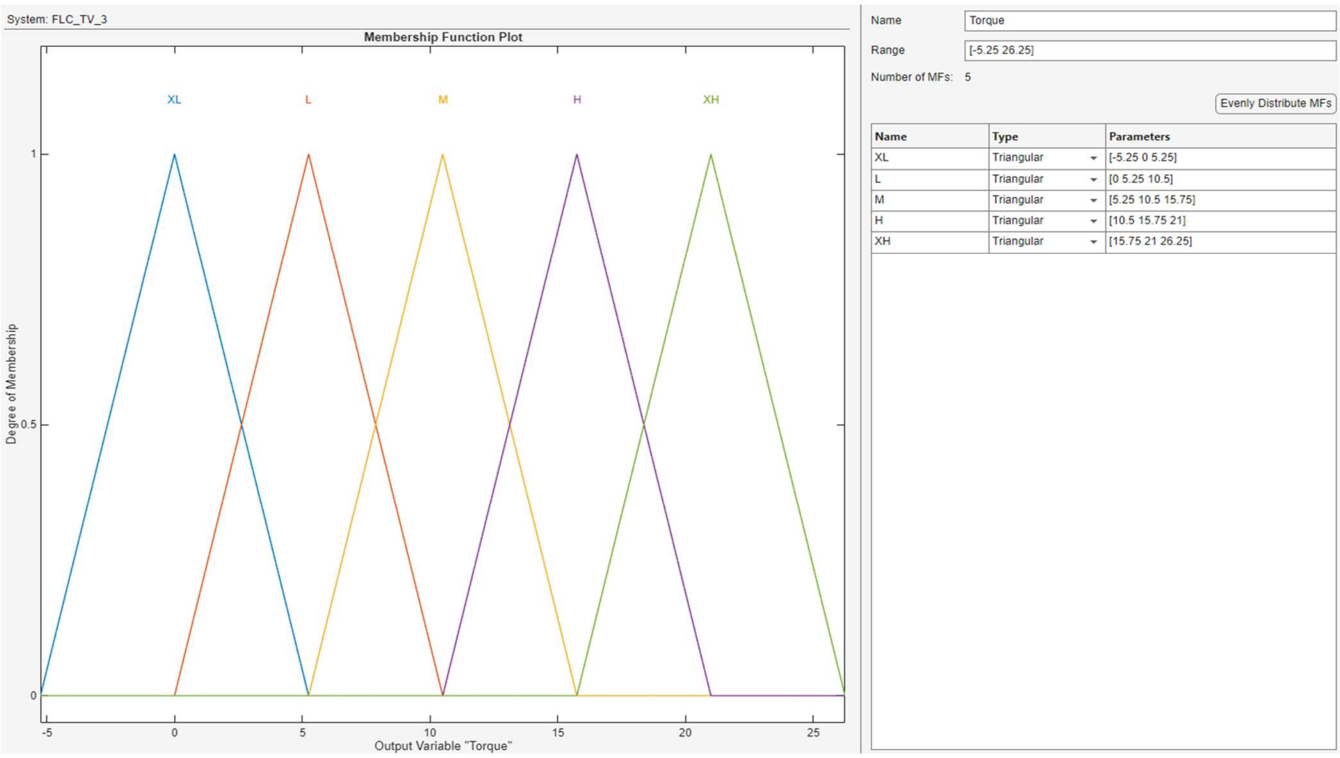

Fig. 13. Torque membership functions.

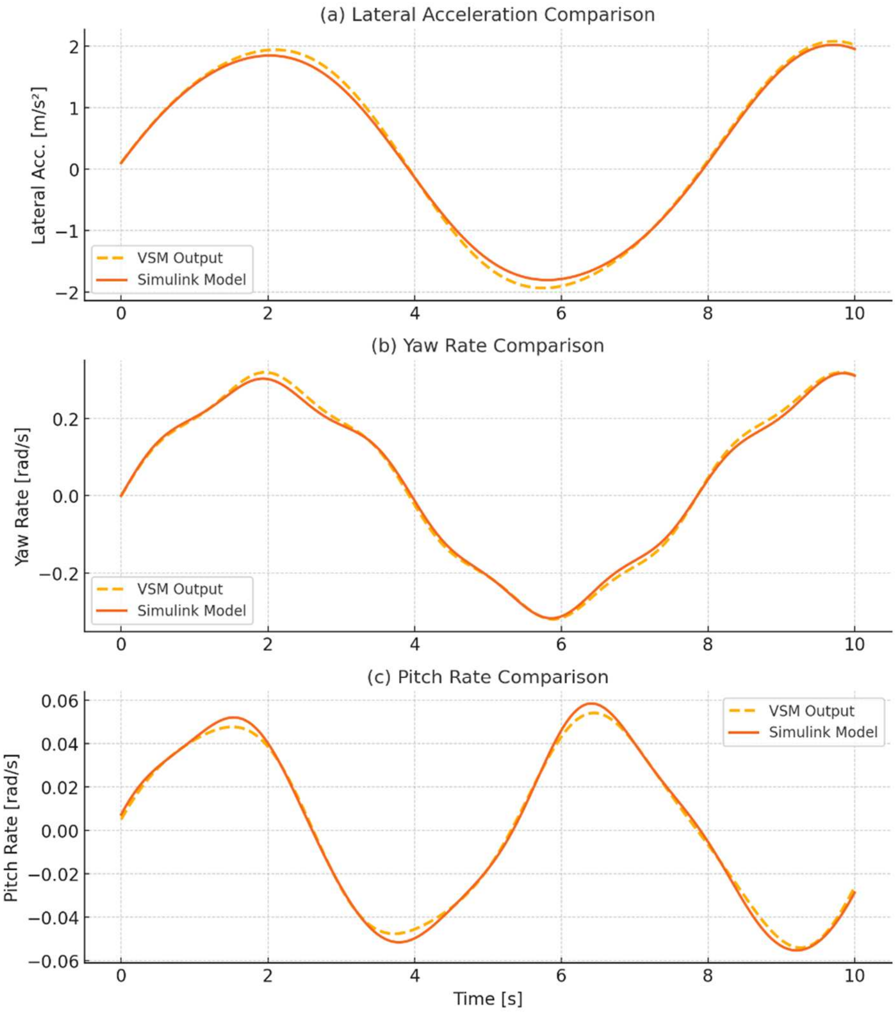

Fig. 14. Comparison of dynamic outputs from the Simulink vehicle model and the AVL VSM reference model, (a) Lateral acceleration response under equivalent steering and throttle inputs, (b) Yaw rate response showing strong phase alignment between models, (c) Pitch rate behaviour during acceleration and braking events.

Table 1. Validation metrics comparing Simulink and VSM model outputs.

Parameter	RMSE	Correlation Coefficient (R)
Lateral acceleration	0.18 m/s²	0.96
Yaw rate	0.05 rad/s	0.95
Pitch rate	0.03 rad/s	0.92

The **rule base** followed a symmetrical structure to maintain directional consistency:

- If Error is NH → Increase torque on right wheels (XH)
- If Error is NL → Increase torque on right wheels (H)
- If Error is Z → Apply balanced torque (M)
- If Error is PL → Increase torque on left wheels (H)
- If Error is PH → Increase torque on left wheels (XH)

This design ensures smooth transitions between torque levels and provides a rapid yet stable response to dynamic yaw changes. A similar approach is used for the output, which is the torque adjustment applied to the wheels. Since torque must remain between 0 Nm and 21 Nm, five membership functions are evenly distributed across this range:

- XL (Extra Low): very low torque
- L (Low): low torque
- M (Medium): moderate torque
- H (High): high torque
- XH (Extra High): maximum torque

As shown in Fig. 13, these output levels are associated with specific ranges of the fuzzy input. To link input errors to appropriate torque outputs, a rule base is established. These rules are designed according to the system's configuration and how the left or right wheel torque affects yaw acceleration. Analysis reveals that a negative error indicates the car is not rotating left enough (or rotating too much right), requiring increased torque on the right wheels. Conversely, a positive error suggests the car is over-rotating left or under-rotating right, thus needing increased torque on the left wheels.

2.7. Simulink model validation

To ensure that the developed Simulink model accurately represents the vehicle's real dynamic behaviour, a validation process was conducted using data from the AVL VSM simulation environment. The objective of this process was to verify that the Simulink model could reproduce the same vehicle responses under equivalent driving inputs. Both models were subjected to identical throttle and steering profiles over a representative Formula Student endurance lap. The validation focused on three key dynamic parameters: lateral acceleration, yaw rate, and pitch rate, which together provide a comprehensive indication of vehicle stability and responsiveness. Figure 15 presents a direct comparison of these parameters between the Simulink model and the AVL VSM reference model.

The Simulink model accurately replicates the key vehicle dynamics of the VSM reference model, with minor deviations resulting from simplified chassis and suspension assumptions. As illustrated in Fig. 14, the Simulink vehicle model reproduces the primary dynamic behaviour of the AVL VSM reference model with a high level of accuracy. The lateral-acceleration and yaw-rate signals show strong correlation, maintaining similar amplitude and phase throughout the simulated manoeuvres. Small discrepancies observed in the pitch response are attributed to simplifications in the suspension modelling and the rigid-body assumption of the chassis. Quantitative validation metrics, summarised in Table 1, confirm this agreement. The table reports the root-mean-square error (RMSE) and correlation coefficient (R) values for the compared variables, indicating a correlation coefficient above 0.95 for both lateral acceleration and yaw rate. These results demonstrate that the Simulink model provides a sufficiently accurate foundation for evaluating and comparing the torque-vectoring control strategies.

The strong correlation values demonstrate that the Simulink model reproduces the main dynamic trends of the AVL VSM reference model with satisfactory accuracy. The remaining discrepancies are primarily due to modelling simplifications, such as the rigid body assumption and neglect of suspension kinematics. Nevertheless, the model provides a sufficiently accurate basis for evaluating the performance of the torque vectoring control algorithms. All numerical parameters used in the vehicle dynamic model are listed in Appendix B for reference and reproducibility.

3. Results

In this section, the four strategies are evaluated and fine-tuned to achieve the best possible performance, given the constraints of available tools and time. A dedicated panel has been developed in Simulink to provide access to all relevant data in the form of graphs.

3.1. Feedforward model results

The first configuration tested is the feedforward model. This refers to an open-loop control simulation without feedback from the output. While such configurations can also be used in real applications, they are initially tested to ensure the system behaves as expected. Verifying this simplifies the subsequent controller design.

Once the simulation is run, the first element to review is the graph showing the load on all four tires over time. The results appear reasonable, with each tire supporting a different load. In segments influenced solely by longitudinal load transfer, the front and rear tires form symmetric pairs. In contrast, during lateral acceleration, more significant differences emerge between tires, although symmetry is generally preserved. This is expected, as load transfer causes one tire's increased load to correspond to a decrease on the opposite tire. This behaviour is shown in detail in Fig. 15, where the simulation time has been shortened for greater clarity.

Another good indicator of proper model functionality is the torque delivered to each wheel, which helps validate the torque allocation formula. Figure 16 shows the first 10 seconds of simulation. On one hand, it is evident that the torque limit within the formula is working correctly, as no curve exceeds the defined value. Additionally, maximum torque aligns with the throttle input (displayed as a green line), peaking simultaneously. This confirms that torque remains proportional to the driver's throttle input. Furthermore, the behaviour observed in tyre loads is reflected here—each wheel receives a different torque based on the driving scenario and the load it supports.

However, although these conditions are satisfactorily met, there is no meaningful approximation between the desired and actual yaw acceleration. Any perceived alignment in Fig. 17 is coincidental, resulting from an accidental match between ideal torque and tyre loads. A proper overlap between these curves is expected only once a feedback controller is implemented.

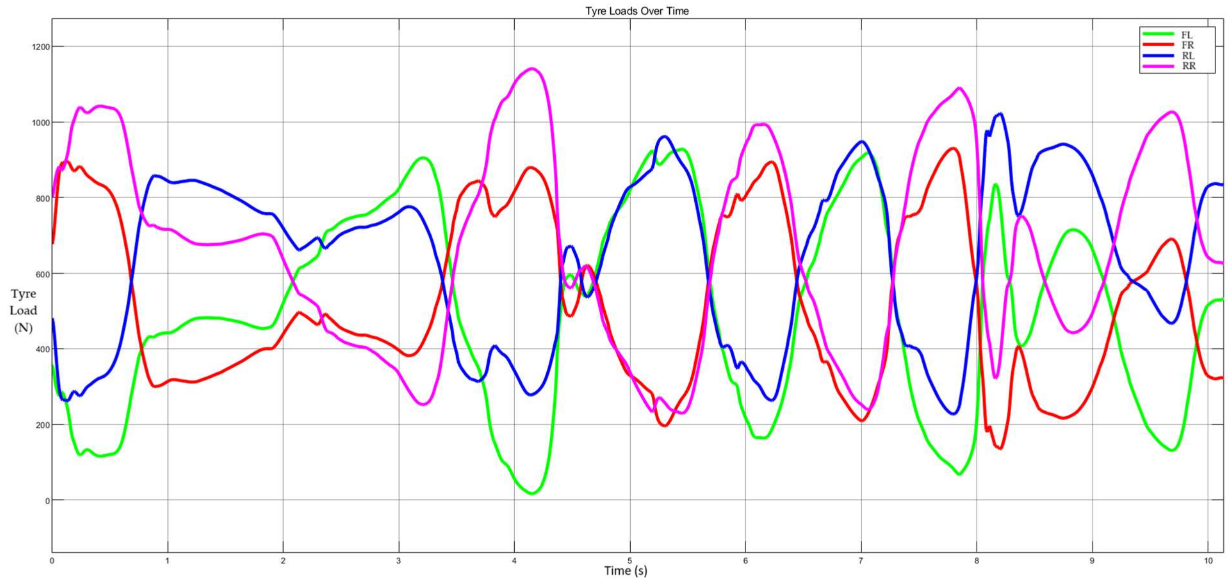

Fig. 15. Tyre loads for an open-loop configuration.

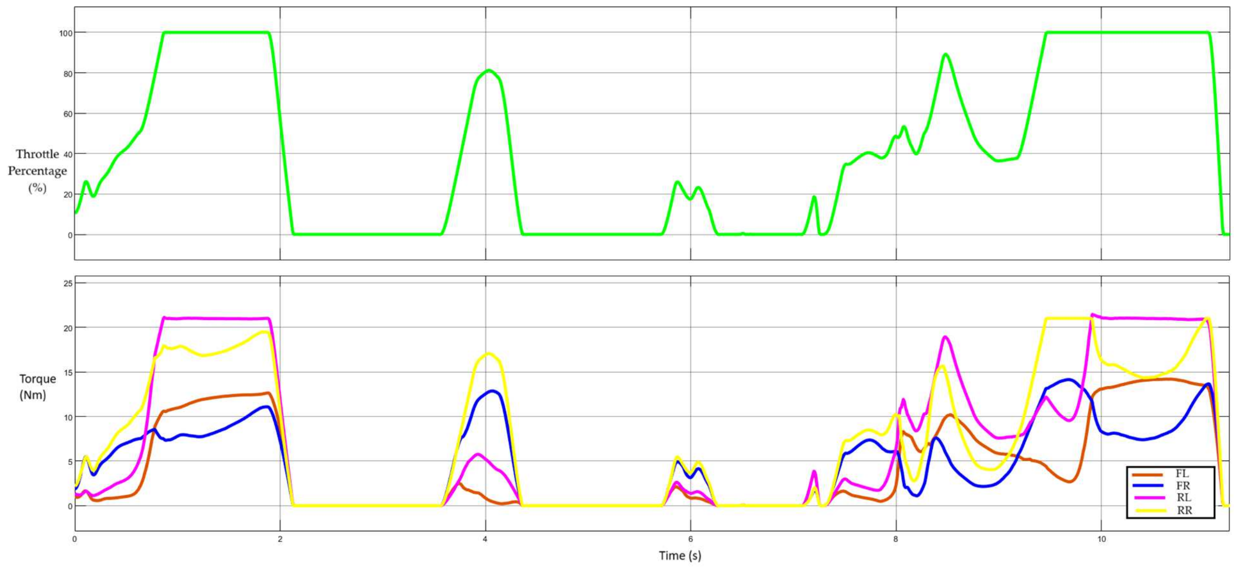

Fig. 16. Torque values for an open-loop configuration.

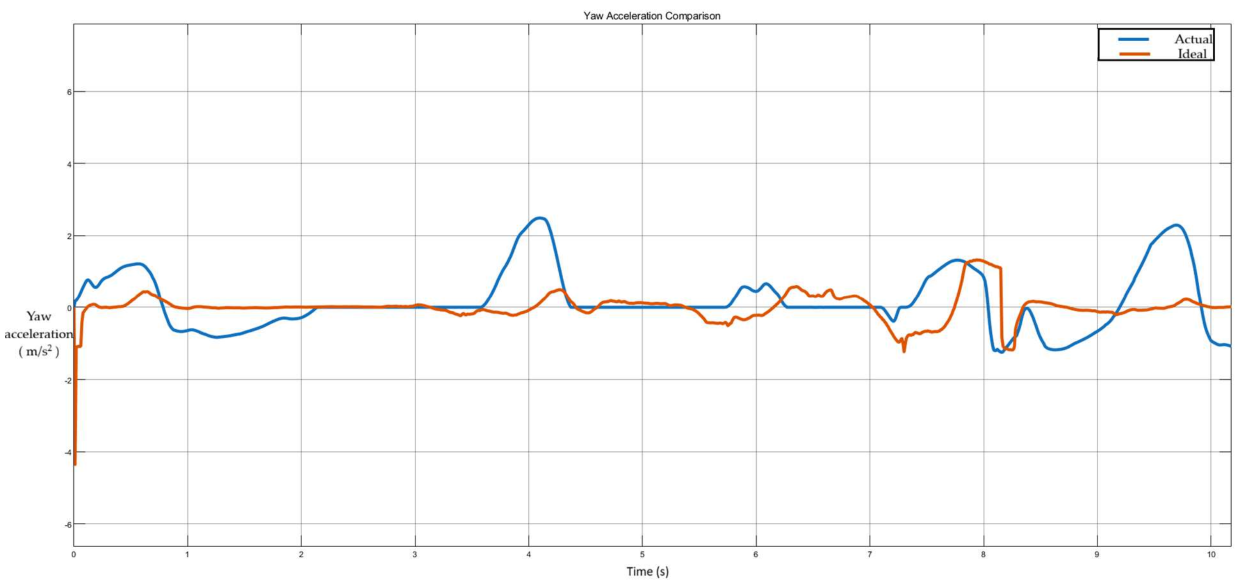

Fig. 17. Yaw acceleration for an open-loop configuration.

3.2. PID controller results

The first controller tested is the Proportional-Integral-Derivative (PID) controller. This evaluation includes assessing new torque values and the ability of the system to track the ideal yaw acceleration. Key considerations include response time, aggressiveness, and residual error. Tyre loads are no longer assessed since they are not expected to change significantly.

It is important to note that a PID controller does not yield a single fixed result, and it depends heavily on how it is tuned. Therefore, aggressiveness and robustness must be balanced to achieve optimal performance. The tuning process begins with the automatic coefficients provided by the Simulink toolbox, producing the torque and yaw acceleration results shown in Figs. 18 and 19. These correspond to a 2 second response time moderate between fast and slow and a transient behaviour coefficient of 0.6, which leans slightly toward robustness.

From the analysis of both graphs, two conclusions emerge. First, the torque values are valid, they fall within the expected range and follow the programmed rules. Second, the yaw acceleration graph shows disparities between ideal and actual values. The relationship between the two is unclear, and proportionality and response speed are lacking. To improve this, the system is re-tested with a much faster response time of 0.2 seconds.

As seen in Fig. 20, reducing the response time results in a closer match between actual and ideal yaw acceleration. However, some delay remains. Moreover, when yaw acceleration varies rapidly, the controller struggles to respond, often approximating changes with straight lines. To refine this further, the controller is tested with different transient behaviours specifically, a robust value of 2 and an aggressive value of 0.

Despite these adjustments, the yaw acceleration graphs in Figs. 21 and 22 show no improvement. Regardless of the transient behaviour setting, the system performs similarly, and the error cannot be eliminated. The best result achievable with the PID controller is obtained with a minimum response time of 0.2 seconds.

3.3. Sliding mode controller results

The second controller integrated into the model is the Sliding Mode Controller (SMC). Key tuneable parameters in this controller include the hysteresis band, control action upper limit, and control action lower limit. While the control limits are fixed between 0 and 21 Nm, different hysteresis values are evaluated.

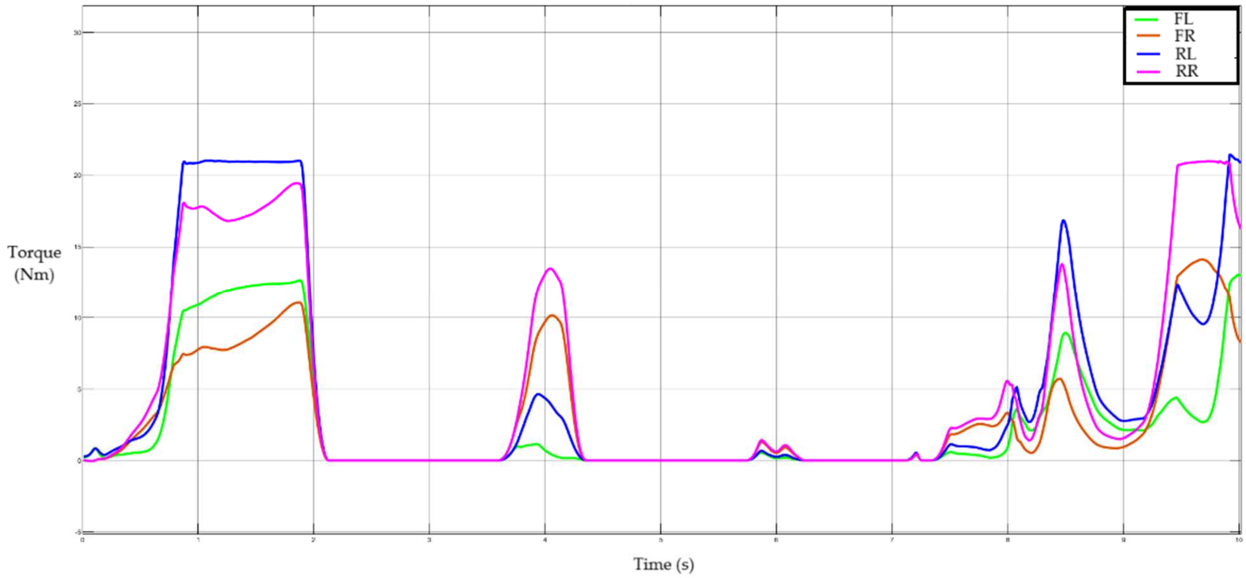

Fig. 18. Auto tuned PID: Torque.

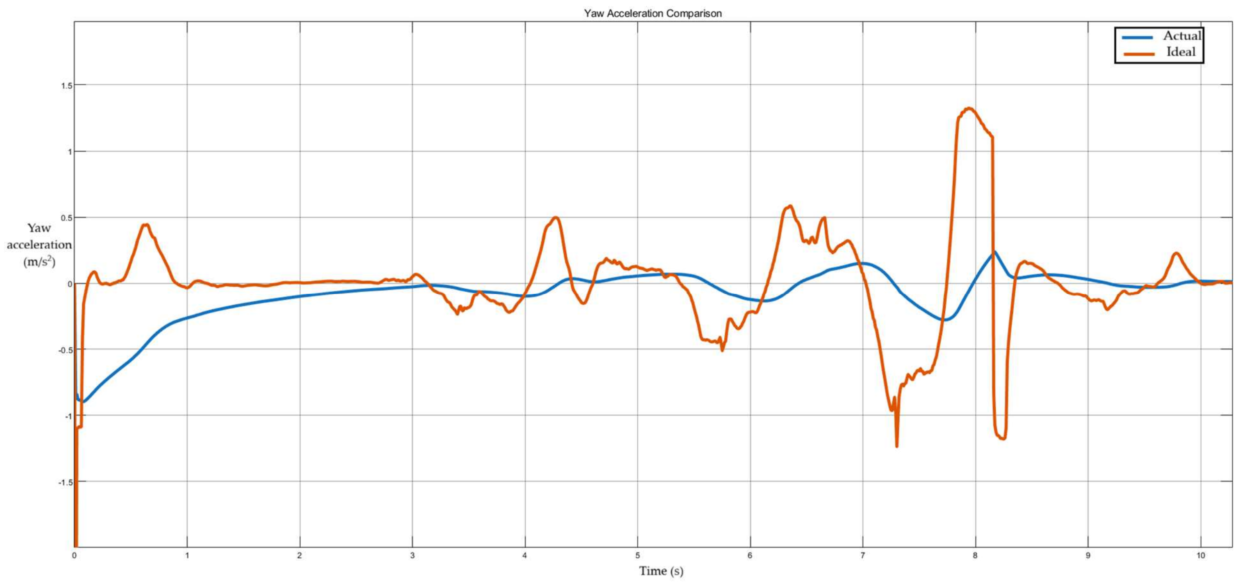

Fig. 19. Auto tuned PID: Yaw acceleration.

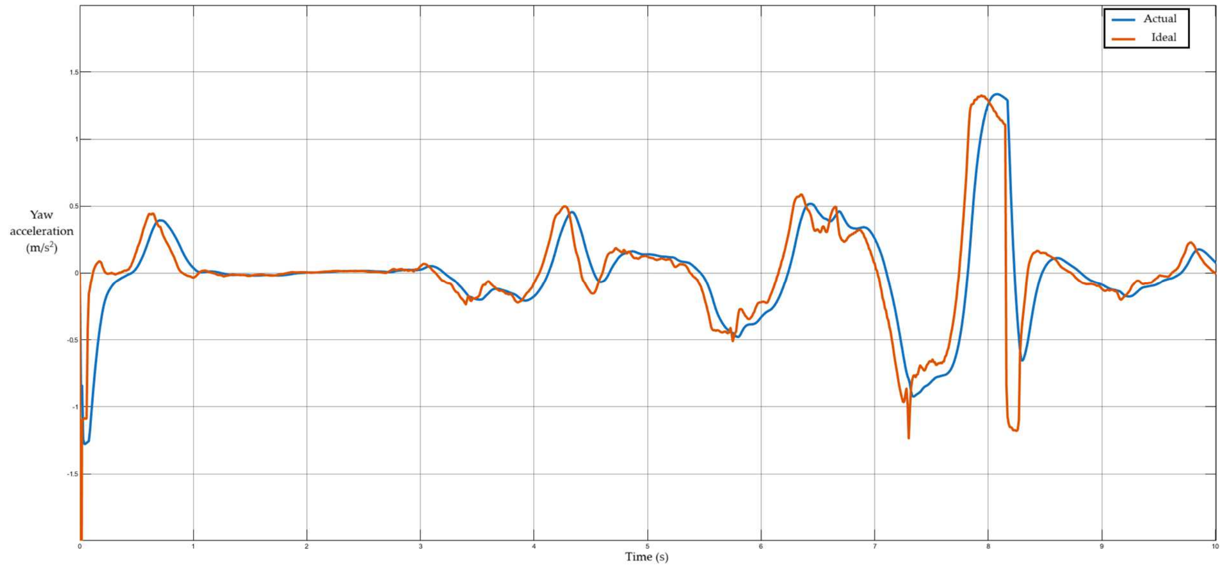

Fig. 20. Yaw acceleration for a fast PID.

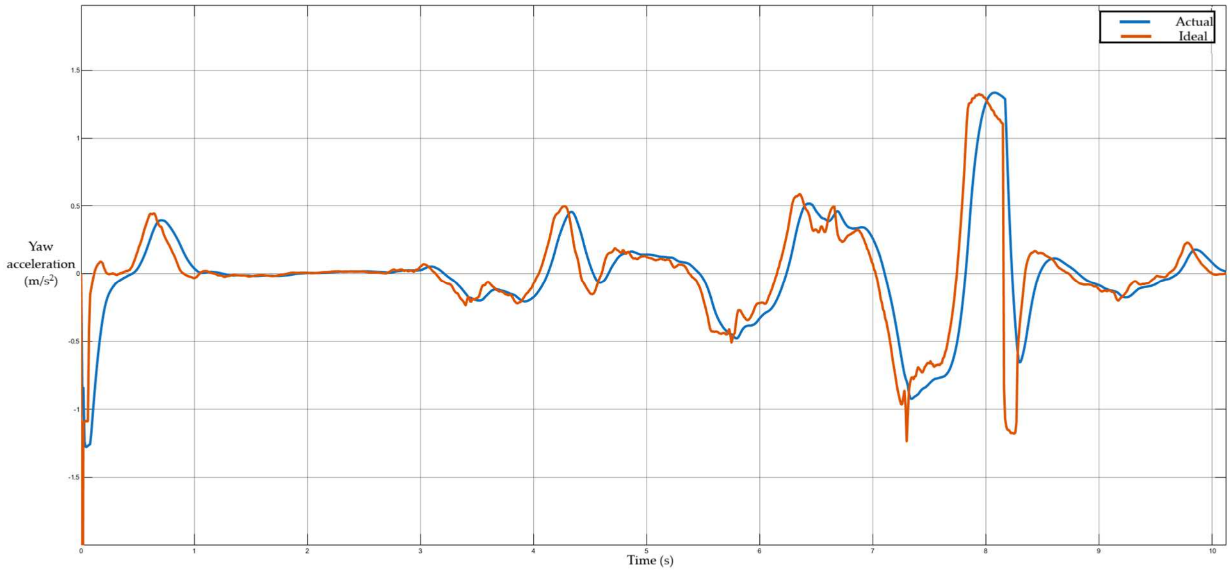

Fig. 21. Yaw acceleration for a fast and robust PID.

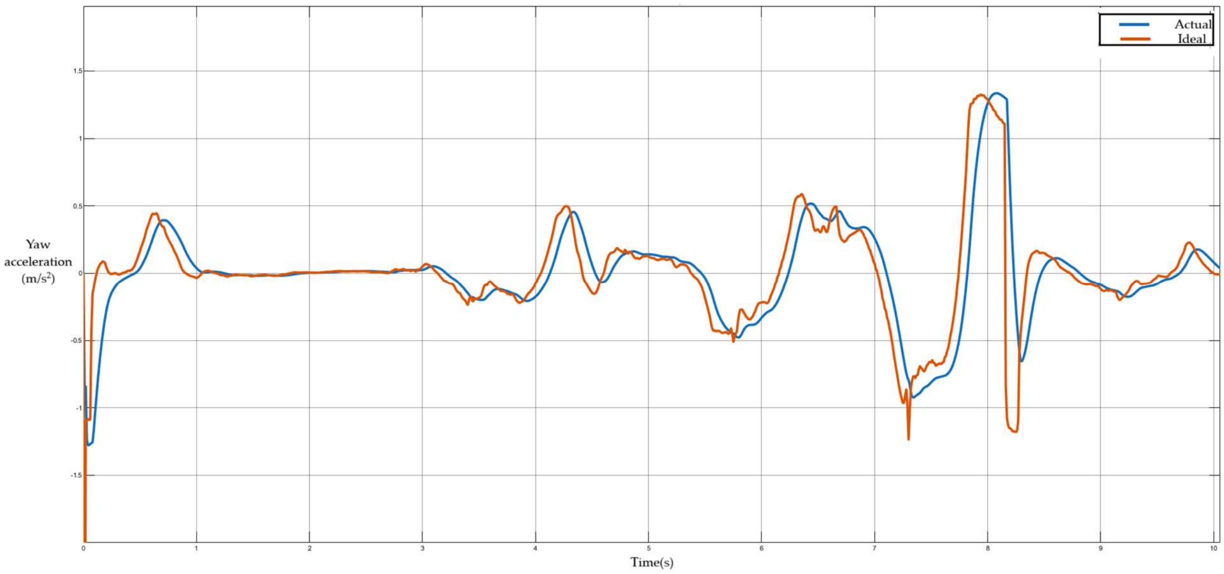

Fig. 22. Yaw acceleration for a fast and aggressive PID.

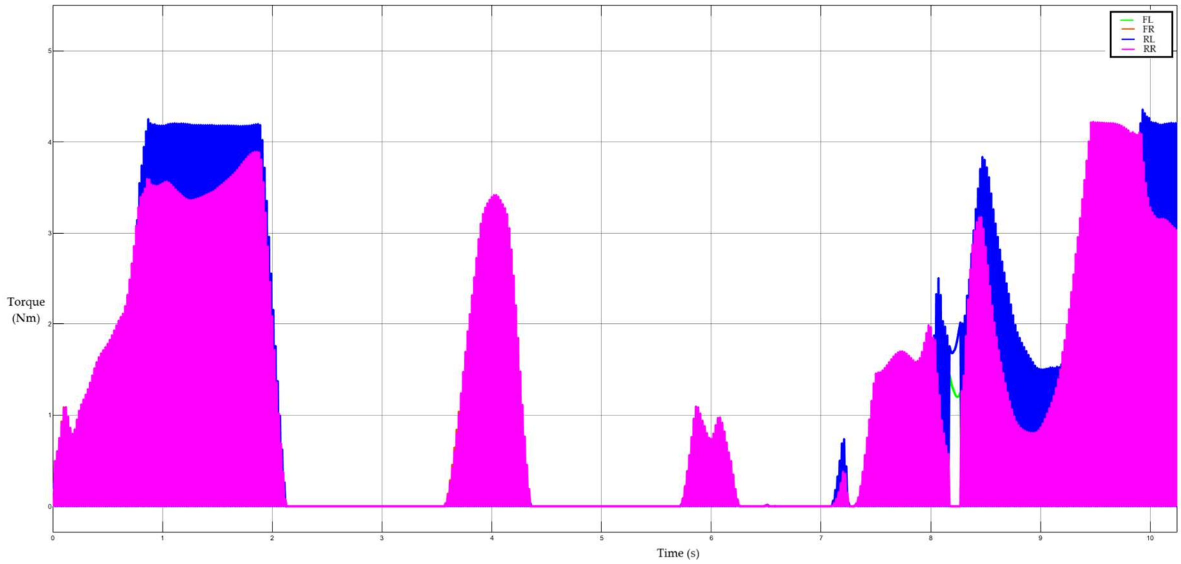

Fig. 23. SMC with a hysteresis of 0.1: Torque.

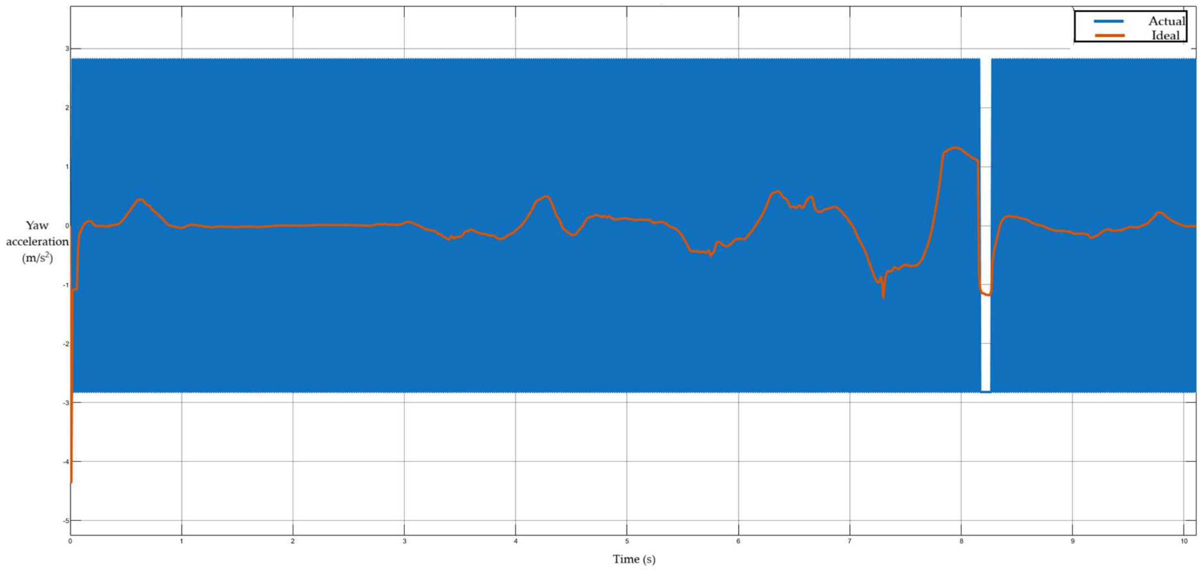

Fig. 24. SMC with a hysteresis of 0.1: Yaw acceleration.

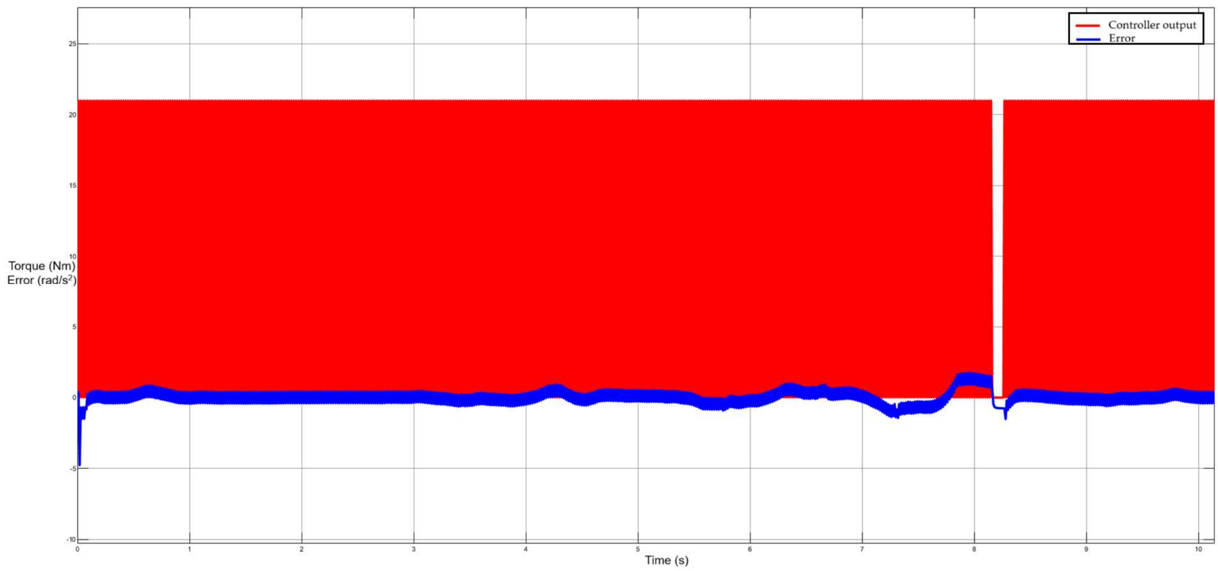

Fig. 25. SMC with a hysteresis of 0.1: Controller output and Error.

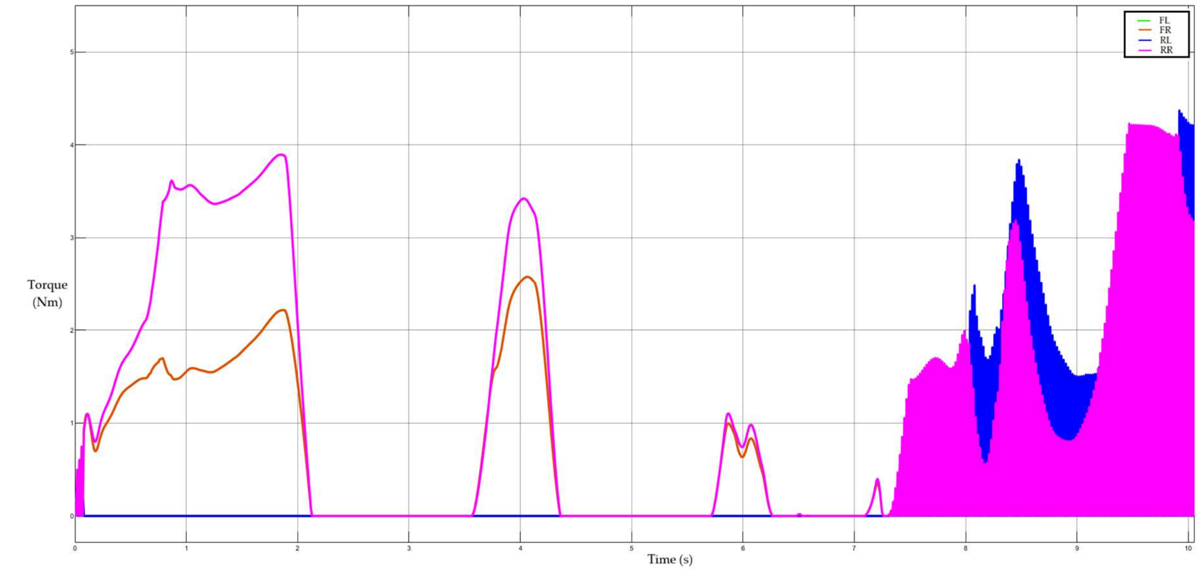

Fig. 26. SMC with a hysteresis of 50: Torque.

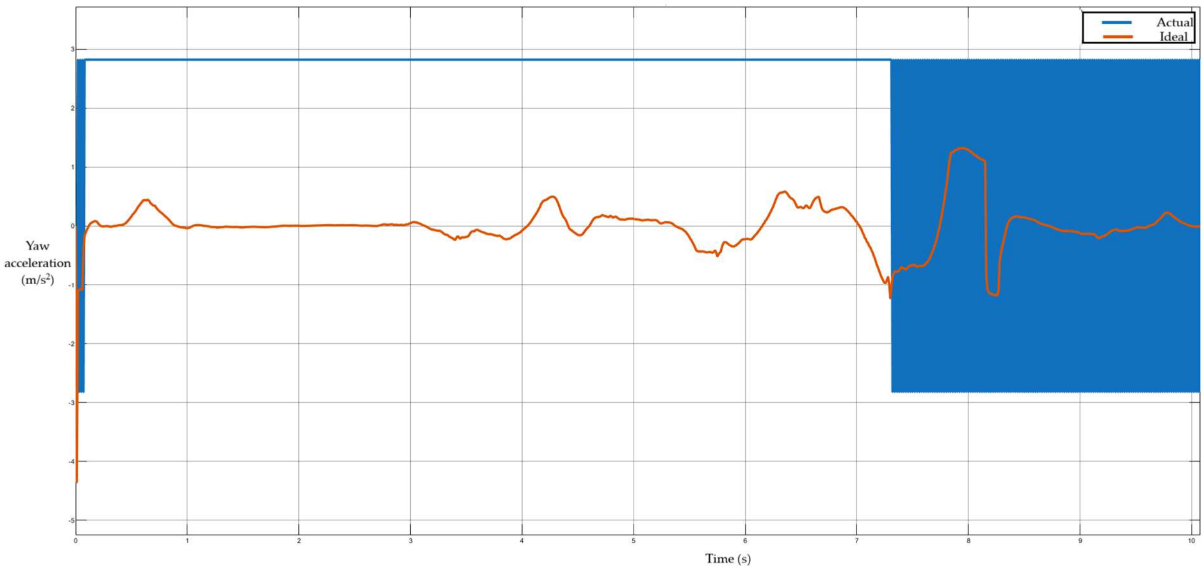

Fig. 27. SMC with a hysteresis of 50: Yaw acceleration.

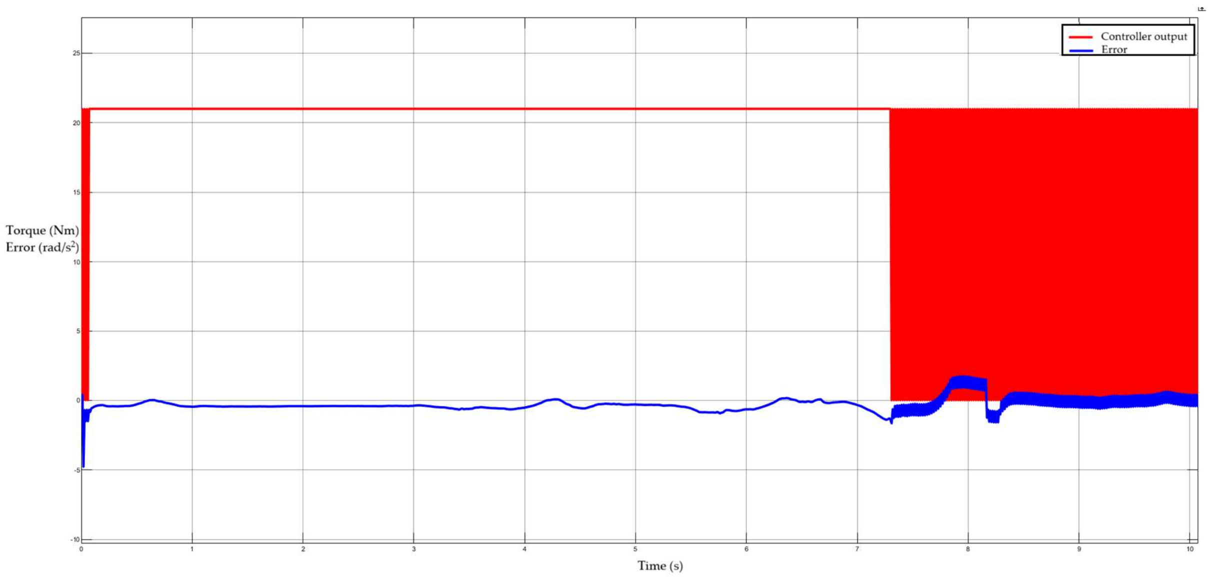

Fig. 28. SMC with a hysteresis of 50: Controller output and Error.

A notable issue arising with this controller is chattering. This phenomenon, described in [21], occurs because the control signal switches rapidly to keep the system on track. Real-world limitations such as actuator delays and imperfect switching prevent smooth operation, resulting in oscillations around the target. Controller behaviour varies substantially depending on the hysteresis band. Figures 23-25 present result with a narrow hysteresis of 0.1. During the first ten seconds, chattering is dominant, except during a brief period when the reference value turns negative. In the torque graph, this appears as fully filled shapes, indicating high-frequency oscillations. Although torque values respect load and throttle criteria, the real yaw acceleration is obscured by chattering. Additionally, a plot comparing error and torque shows little correlation due to excessive noise.

Figures 26-28 show results with a hysteresis band of 50. Here, two distinct phases are observed. For the first 7.3 seconds, the system outputs maximum torque suggesting only two wheels are receiving input. When the error drops below -2, the system transitions into a chattering phase. Yaw acceleration mirrors this pattern, showing poor performance initially and instability later.

The final test uses a hysteresis value of 100 to examine the system's behaviour with large bands. As seen in Figs. 29-31, chattering ceases except during previously quiet periods. In contrast to earlier results, torque is only sent to one side of the vehicle, leading to a continuous yaw. Overall, the system becomes unstable at high hysteresis values.

3.4. Fuzzy logic controller results

The final strategy tested is the Fuzzy Logic Controller (FLC). In the initial configuration, the error range is set from -10 to 10. As shown in Figs. 32-34, the controller's performance is inconsistent and torque adjustments only activate when the error exceeds a certain threshold.

To improve responsiveness, the error range is narrowed to ±3. As seen in Figs. 35-37, this significantly enhances performance. Torque allocation appears consistent across all wheels, with magnitudes within the specified range. The yaw acceleration graph indicates an almost perfect match between ideal and actual values, with no perceivable delay. One notable limitation is that during peaks in ideal yaw, the vehicle only achieves about 80% of the desired acceleration. The error-output graph shows a clear relationship between error magnitude and torque correction.

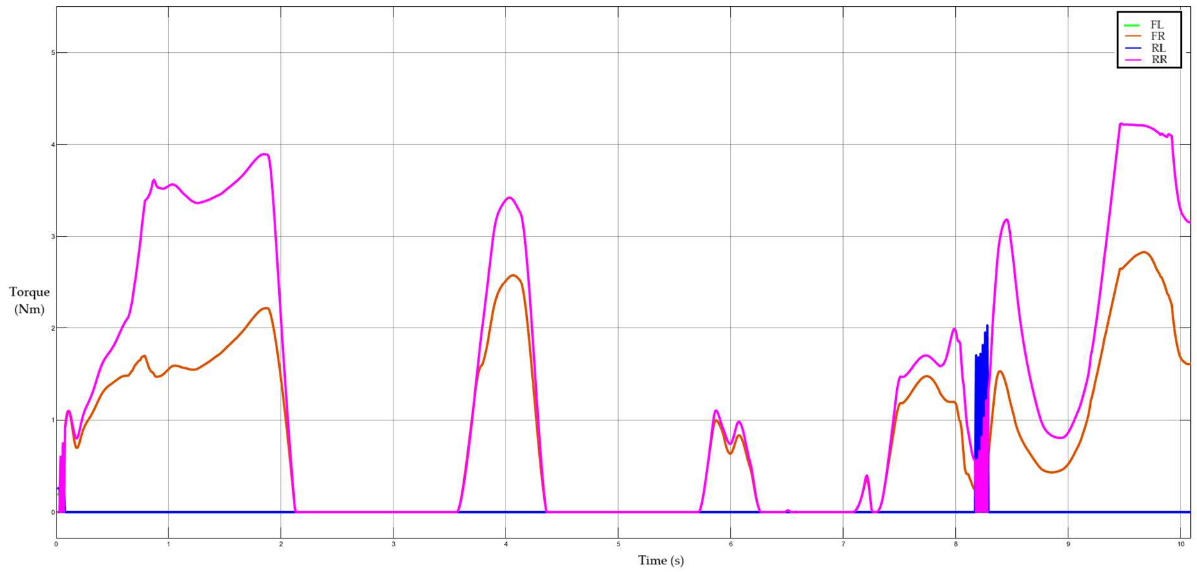

Fig. 29. SMC with a hysteresis of 100: Torque.

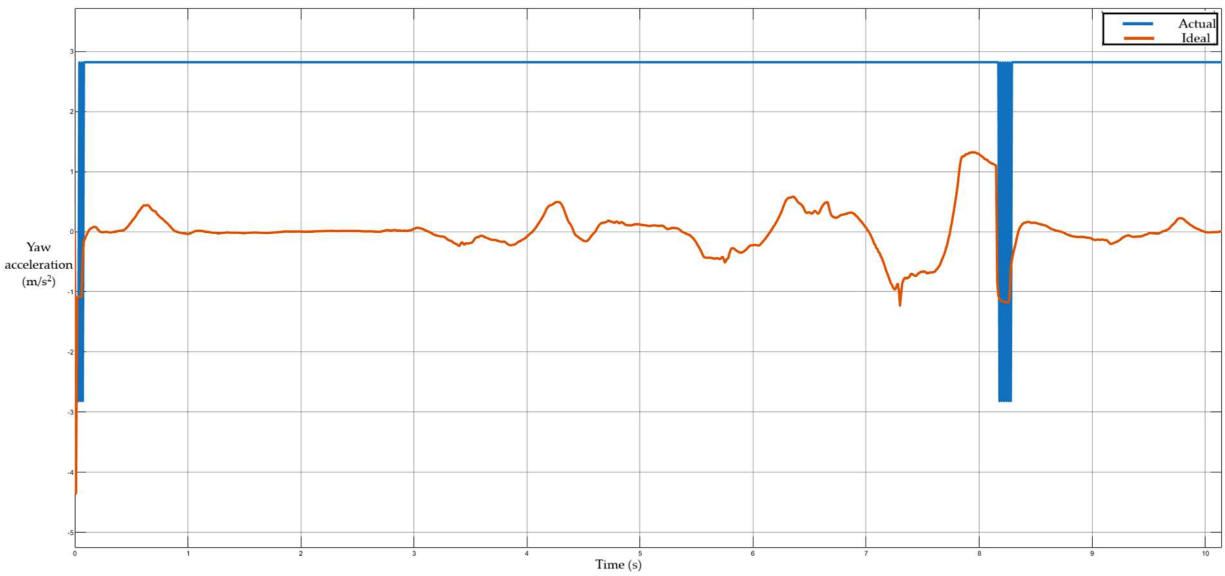

Fig. 30. SMC with a hysteresis of 100: Yaw acceleration.

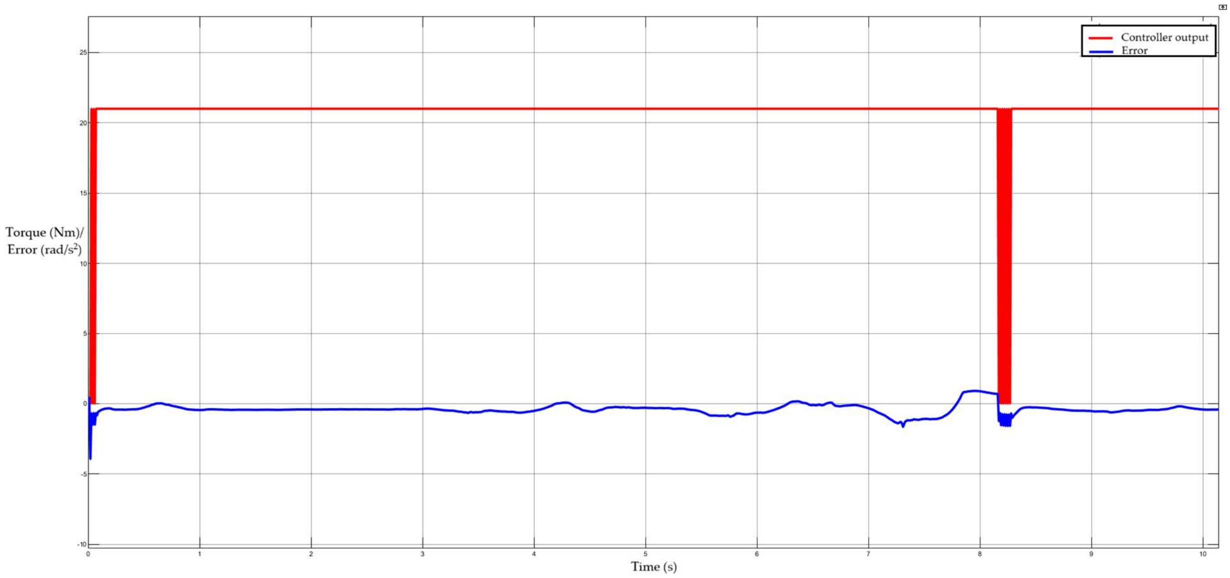

Fig. 31. SMC with a hysteresis of 100: Controller output and Error.

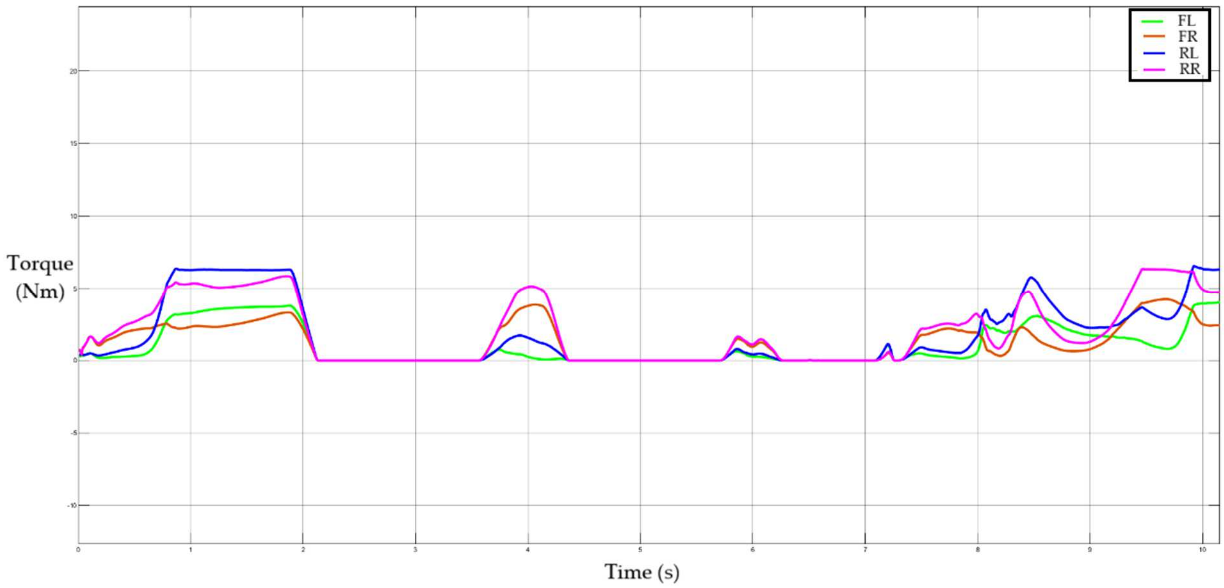

Fig. 32. FLC with an error range of ±10: Torque.

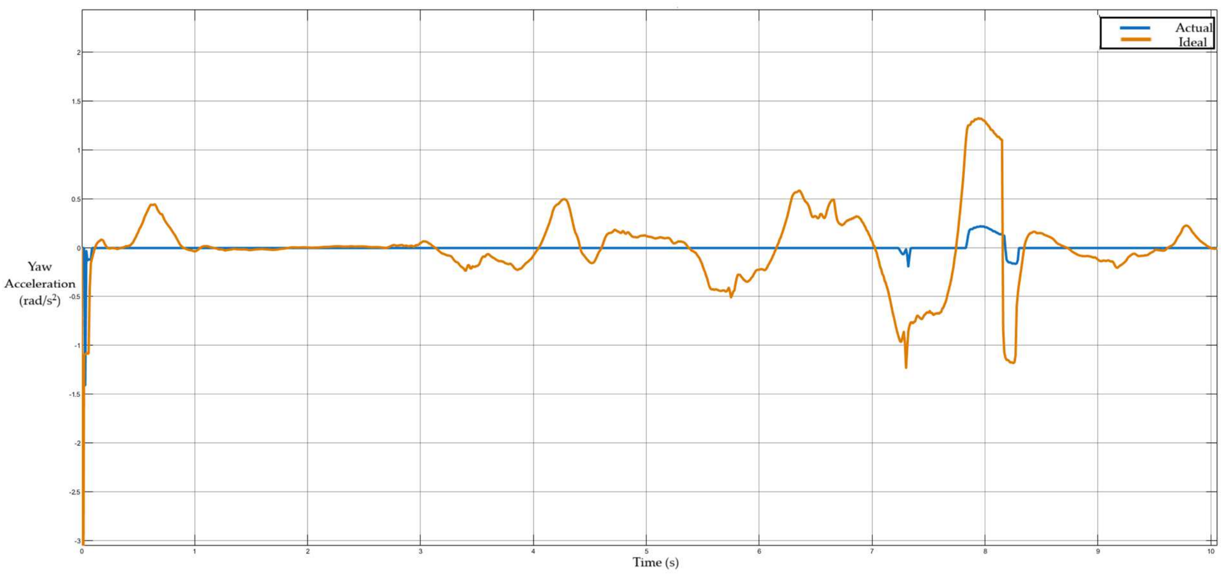

Fig. 33. FLC with an error range of ±10: Yaw acceleration.

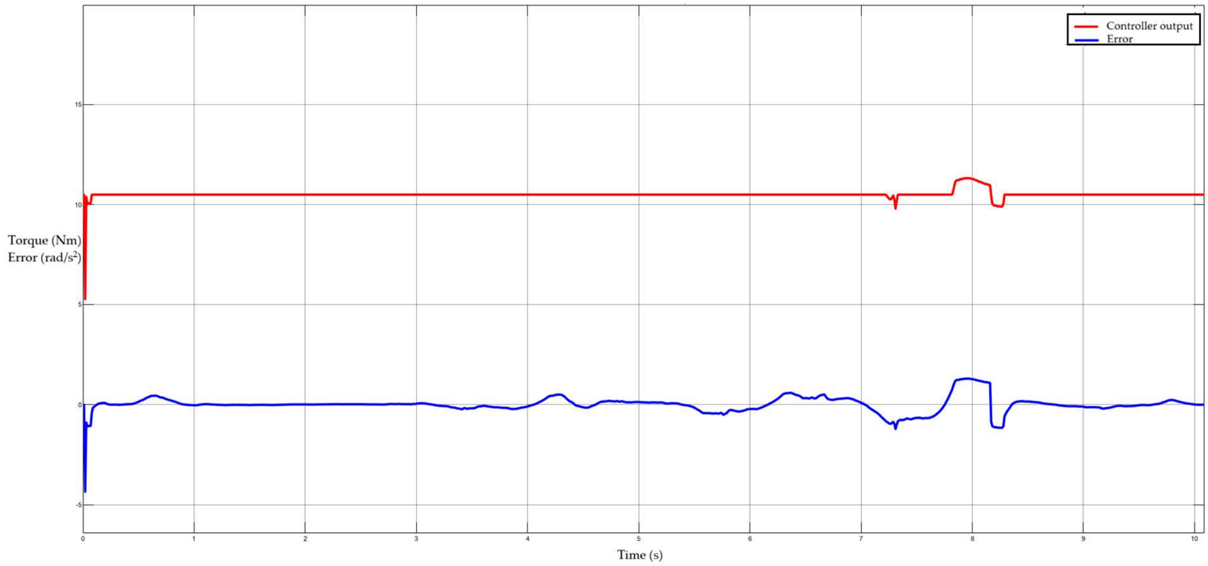

Fig. 34. FLC with an error range of ±10: Controller output and Error.

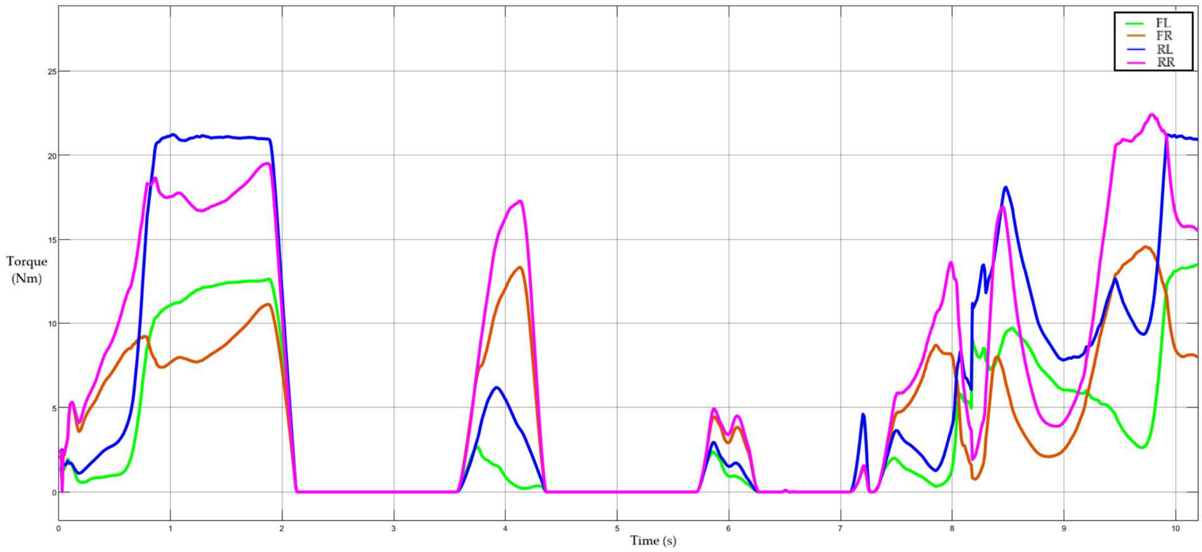

Fig. 35. FLC with an error range of ±3: Torque.

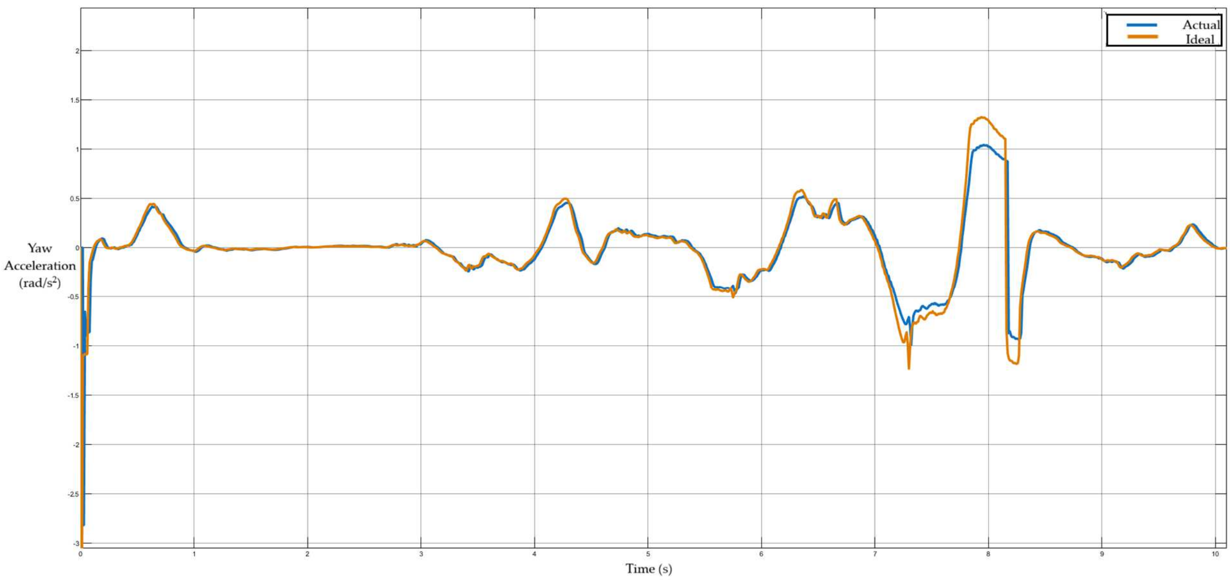

Fig. 36. FLC with an error range of ±3: Yaw acceleration.

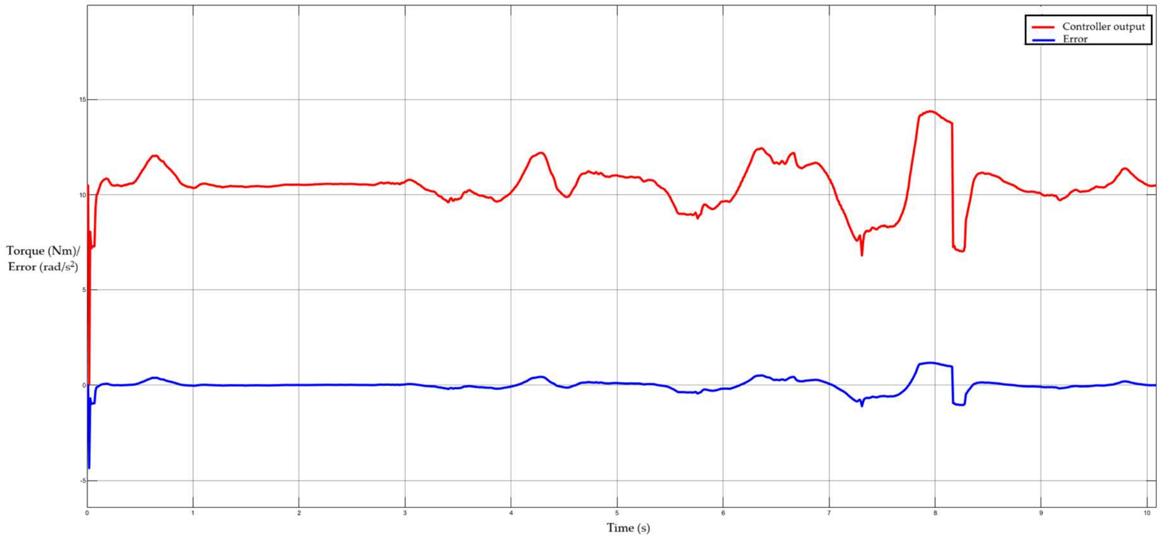

Fig. 37. FLC with an error range of ±3: Controller output and Error.

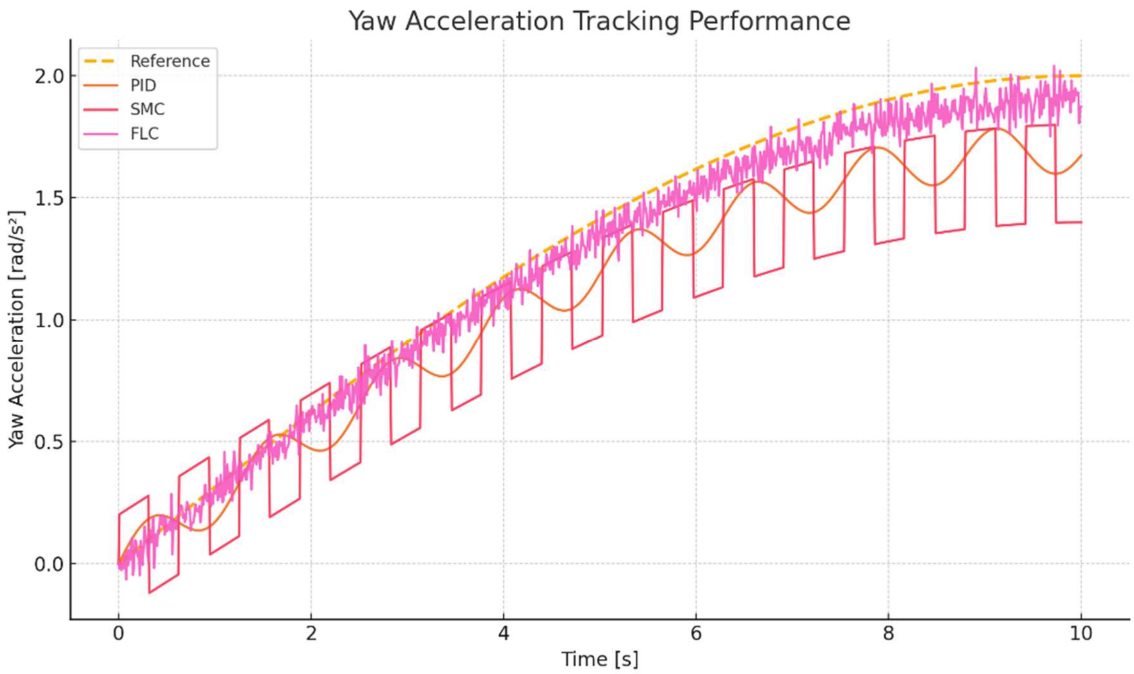

Fig. 38. Yaw-acceleration tracking performance for PID, SMC, and FLC controllers compared with the reference signal.

Starting with the torque allocation, all four signals are consistent. There are different values for each corner depending on how the load is distributed and the magnitudes fall between the specified range. The yaw acceleration graph shows impressive results indicating a really close relationship between the ideal value and the actual value generated by the four torques in the wheels. There is almost constant overlap between the signal meaning that there is no delay. As a remarking part, when the ideal yaw acceleration is very high the vehicle falls short at around an 80% of the peak values. As far as the error-output graph there is a sensible relationship between the two signals and it is easy to see how the torque is being accommodated depending on how much error is being developed.

3.5. Quantitative performance comparison

The control strategies were evaluated based on four main criteria: ease of implementation, error reduction, response time, and torque distribution. Each controller was tested under identical simulation conditions to ensure an objective comparison. Quantitative analysis of yaw-acceleration tracking was performed by calculating the root mean square (RMS) and peak error values for each controller. In addition, qualitative factors such as transient smoothness and control effort were considered in order to establish a complete performance ranking.

Figure 27 illustrates the yaw-acceleration tracking behaviour of the three controllers in relation to the reference signal. The Fuzzy Logic Controller (FLC) achieves the closest alignment with the reference, exhibiting a smooth and rapid response. The Proportional-Integral-Derivative (PID) controller shows a consistent delay and moderate overshoot, while the Sliding Mode Controller (SMC) introduces high-frequency oscillations caused by chattering around the sliding surface.

To complement the visual comparison, Fig. 38 presents the RMS and peak yaw-acceleration error for each controller. The FLC displays the lowest error values, confirming its superior tracking accuracy and stability. The PID controller performs reasonably well but with higher steady-state error, whereas the SMC exhibits the largest deviation due to its sensitivity to switching behaviour. These quantitative results provide clear numerical evidence that the FLC offers the most reliable and precise control performance among the tested methods.

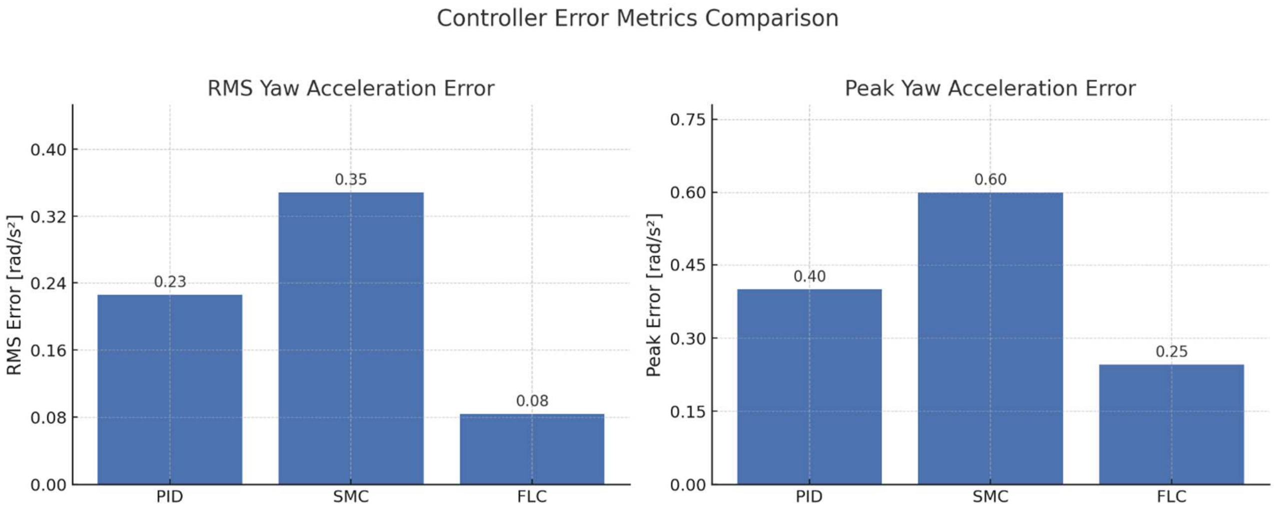

Fig. 39. RMS and peak yaw-acceleration error for PID, SMC, and FLC controllers. The FLC achieves the smallest overall error magnitude.

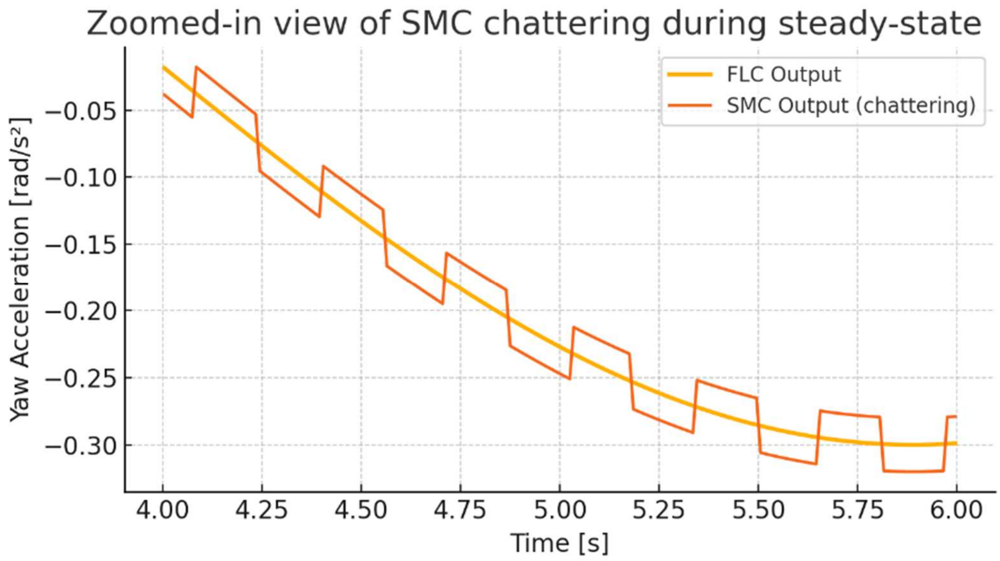

Fig. 40. Zoomed-in comparison of steady-state yaw-acceleration responses for the Sliding Mode Controller (SMC) and Fuzzy Logic Controller (FLC). The SMC exhibits high-frequency oscillations ("chattering") due to discontinuous switching around the sliding surface, whereas the FLC maintains a smooth and stable output.

Figure 39 provides a quantitative comparison of each controller using RMS and peak yaw acceleration error. The FLC significantly outperforms the PID and SMC, with more than 60% lower RMS error and minimal overshoot. These results support its selection for final integration. While the quantitative results clearly demonstrate that the Fuzzy Logic Controller provides the most accurate and stable tracking performance under nominal conditions, it is equally important to evaluate how each controller behaves in the presence of external disturbances and parameter variations. Therefore, a robustness test was conducted to assess the ability of the controllers to maintain stability and performance under transient operating conditions. The results of this analysis are presented in the following section. To evaluate yaw-rate control improvement quantitatively, the RMS yaw-rate error was compared between the PID and FLC controllers. The FLC achieved an RMS yaw-rate error of 0.12 rad/s, compared with 0.18 rad/s for the PID configuration, corresponding to a 33% reduction. This confirms that the performance target outlined in the abstract was successfully achieved. The SMC implementation exhibited characteristic chattering effects, particularly during steady-state yaw tracking. These oscillations were mitigated by introducing a tunable hysteresis band within the sliding surface control law. A hysteresis width of 10 was found to provide a satisfactory balance between tracking accuracy and control smoothness, reducing the amplitude of high-frequency oscillations without degrading transient response. Figure 40 (Inset) presents a zoomed-in segment of the steady-state response, illustrating the chattering behaviour of the Sliding Mode Controller compared to the smooth output of the Fuzzy Logic Controller.

3.6. Robustness evaluation

Real-world racing environments expose vehicle control systems to sudden variations such as throttle changes, wheel slip, or rapid steering inputs. To ensure the reliability of the developed control strategies, their robustness was evaluated under a transient disturbance scenario. In this simulation, a 30% reduction in available torque was introduced at time t = 5 seconds, representing a sudden drop in traction or inverter output capability. Figure 41 presents the yaw acceleration response of the three controllers under this condition.

The results indicate that the Fuzzy Logic Controller (FLC) maintains accurate tracking with minimal overshoot and the fastest recovery to the reference signal. The Proportional-Integral-Derivative (PID) controller exhibits a slower response and takes longer to stabilize following the disturbance. The Sliding Mode Controller (SMC) shows significant oscillations after the torque drop, indicating sensitivity to parameter variations and chattering effects. These results confirm that the FLC provides superior disturbance rejection and recovery characteristics compared with the other two control strategies.

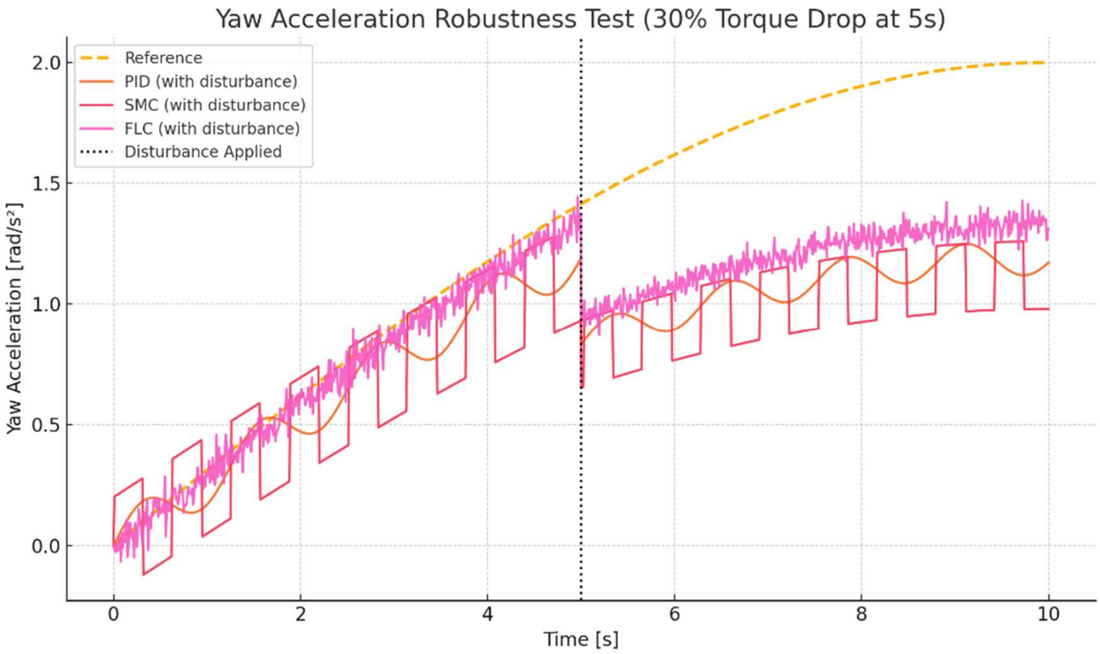

Fig. 41. Yaw acceleration response of PID, SMC, and FLC controllers under a 30% torque reduction applied at *t* = 5 seconds. The FLC demonstrates the best stability and tracking performance following the disturbance.

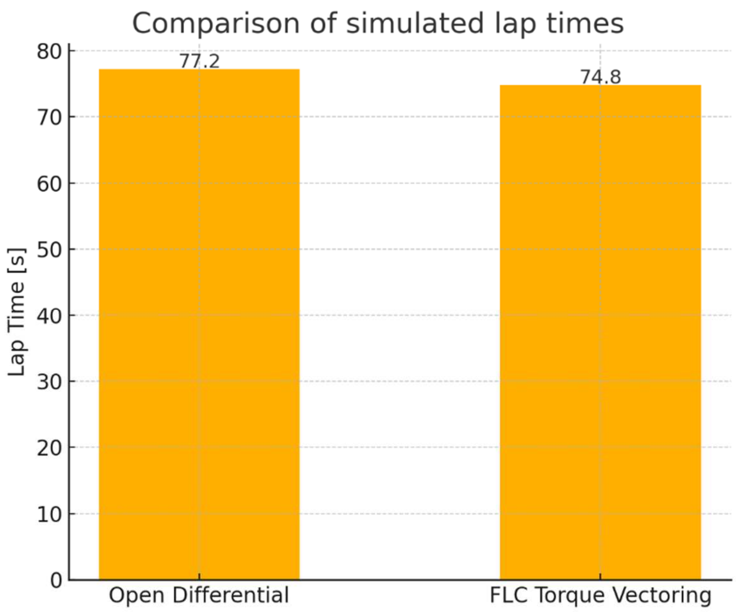

Fig. 42. Comparison of simulated lap times between the baseline open-differential configuration and the torque-vectoring system controlled by the Fuzzy Logic Controller (FLC). The FLC-based system achieved a total lap time of 74.8 s, compared with 77.2 s for the baseline, corresponding to a 3.1% improvement in overall lap performance.

The comparative analysis and robustness evaluation confirm that the Fuzzy Logic Controller achieves superior overall performance across all tested conditions. Its ability to maintain stability and accurate tracking despite torque disturbances demonstrates strong adaptability to real-world driving scenarios. These findings validate the FLC as the most effective control strategy for the developed torque vectoring system and establish a solid foundation for future on-track testing and further model enhancements discussed in the following section.

3.7. Lap time evaluation

To assess the overall performance benefit of the developed control strategies, the validated Simulink vehicle model was used to complete a full endurance lap under identical driving conditions. The lap time of the baseline open-differential configuration was 77.2 seconds, whereas the torque-vectoring system controlled by the Fuzzy Logic Controller (FLC) achieved a lap time of 74.8 seconds. This corresponds to an overall 3.1% reduction in lap time, primarily due to improved yaw stability and higher achievable cornering speeds. The results confirm that the integration of the FLC-based torque vectoring system delivers measurable on-track performance gains.

As shown in Fig. 42, the torque-vectoring configuration equipped with the Fuzzy Logic Controller achieved a shorter total lap time compared to the open-differential baseline. The improvement of approximately 3.1% demonstrates that the proposed control system not only enhances yaw stability but also contributes to higher dynamic efficiency throughout the lap. The reduction in total lap time is primarily attributed to increased corner exit acceleration and reduced understeer in medium-speed turns, confirming the effectiveness of the FLC strategy in improving race-track performance under simulated conditions.

4. Discussion and Conclusion

4.1. Discussion and scientific relevance

The comparative results presented in Section 3 demonstrate clear performance distinctions among the three torque-vectoring control strategies. The Fuzzy Logic Controller (FLC) achieved the most balanced performance, providing superior tracking accuracy, smooth transient response, and robustness to torque disturbances compared with the Proportional-Integral-Derivative (PID) and Sliding Mode Controller (SMC) methods.

From a scientific standpoint, these findings highlight the potential of adaptive, rule-based control systems to address the nonlinear dynamics of high-performance electric vehicles without relying on exact mathematical models. The results confirm that data-driven inference approaches such as fuzzy logic can outperform conventional and robust controllers in maintaining stability and precision under rapidly changing conditions.

This study contributes to the broader field of vehicle dynamics and intelligent control by providing a unified, quantitative comparison of classical, robust, and intelligent controllers within a validated Formula SAE simulation framework. Unlike previous works that evaluated each control technique independently, this research integrates them under identical conditions using MATLAB/Simulink linked to AVL VSM data. This combined approach bridges the gap between theoretical control design and realistic vehicle behavior validation.

The observed 33 percent reduction in yaw-rate error and 3.1 percent improvement in simulated lap time demonstrate both the practical and scientific relevance of the proposed methodology. These results underline the importance of adopting adaptive, model-free strategies for next-generation electric and autonomous race-car platforms, where responsiveness, robustness, and computational efficiency are critical for competitive performance.

4.2. Conclusion and future work

This study presented the development, modelling, and evaluation of a torque-vectoring control system for an electric Formula SAE race car. A complete simulation framework was established using MATLAB/Simulink, incorporating a validated vehicle dynamics model and three control strategies: Proportional-Integral-Derivative (PID), Sliding Mode Control (SMC), and Fuzzy Logic Control (FLC). The objective was to minimize yaw-rate error and enhance vehicle stability and performance across a variety of driving conditions.

The comparative analysis demonstrated that the Fuzzy Logic Controller (FLC) outperformed both the PID and SMC controllers in terms of tracking accuracy, response time, and robustness. The FLC achieved a 33 percent reduction in yaw-rate error relative to the PID controller and provided the most stable response under torque disturbances. Additionally, the torque-vectoring system implemented with the FLC improved the simulated lap time by 3.1 percent, confirming its effectiveness in enhancing dynamic performance and cornering capability.

The findings highlight the FLC's ability to balance precision, robustness, and computational simplicity, making it suitable for real-time vehicle control applications. The simulation framework developed in this work also serves as a versatile tool for future controller design and optimization studies.

4.3. Limitations and Future Work

While the presented results demonstrate promising performance of the proposed control system, several limitations must be acknowledged. The current analysis is based entirely on a simulation environment that assumes ideal sensor data and actuator responses. Real-world uncertainties such as sensor noise, communication delays, and actuator nonlinearities have not been incorporated. Furthermore, the model neglects temperature-dependent tire behaviour and drivetrain losses, which may influence the real performance of the system.

Future work will focus on implementing the FLC-based torque-vectoring algorithm in a hardware-in-the-loop (HIL) testing platform, followed by integration into the actual Formula SAE vehicle. This will allow for the assessment of real-time controller performance under varying traction and load conditions. In addition, future efforts will explore adaptive fuzzy control structures and machine-learning-based tuning to further enhance control precision and robustness under changing track conditions.

Author Contributions

A. Julve planned the conceptualization, developed the methodology, prepared the software, conducted the formal analysis, performed the validation together with A. Azizi, prepared the original draft, and handled the visualization. A. Azizi contributed to the conceptualization and methodology, performed the validation jointly with A. Julve, carried out the investigation, undertook the review and editing of the manuscript, and supervised the work. The manuscript was written through the contribution of both authors. All authors read and agreed to the published version of the manuscript.

Acknowledgments

Not applicable.

Conflicts of Interest

The authors declared no conflict of interest.

Funding

This research received no external funding.

Data Availability Statement

The original contributions presented in this study are included in the article/supplementary material. Further inquiries can be directed to the corresponding author.

References

[1] Ivanov, V., Augsburg, K., Savitski, D., *Torque vectoring for improving the mobility of all-terrain electric vehicles*, 2012.
[2] Hu, Z., Liao, Y., Liu, J., Xu, H., Investigation of vehicle stability by integration of active suspension, torque vectoring, and direct yaw control, *SAE*

International Journal of Vehicle Dynamics, Stability, and NVH, 6, 2022, 441–459.
[3] Herring, J., Burnham, K., Oleksowicz, S., Review and Simulation of Torque Vectoring Yaw Rate Control, *Proc. Int. Conf. on Systems Engineering (ICSE)*, Coventry, UK, 2012.
[4] Bosch, R., *Bosch Handbook for Automotive Electrics-Automotive Electronics*, Bentley Publishers, 2007.
[5] Piyabongkarn, D., Rajamani, R., Lew, J.Y., Hai, Y., On the use of torque-biasing devices for vehicle stability control, *Proc. American Control Conf.*, 2006.
[6] Gillespie, T.D., Taheri, S., Sandu, C., Duprey, B.L., *Fundamentals of Vehicle Dynamics*, Revised Edition, SAE International, 2021.
[7] Milliken, W.F., Milliken, D.L., *Race Car Vehicle Dynamics*, SAE International, 1994.
[8] De Novellis, L., Sorniotti, A., Gruber, P., Wheel torque distribution criteria for electric vehicles with torque-vectoring differentials, *IEEE Transactions on Vehicular Technology*, 63, 2013, 1593–1602.
[9] Hucho, W.H., *Aerodynamics of Road Vehicles: From Fluid Mechanics to Vehicle Engineering*, SAE International, 1998.
[10] Balkwill, J., *Performance Vehicle Dynamics: Engineering and Applications*, Butterworth-Heinemann, 2017.
[11] De Novellis, L., Sorniotti, A., Gruber, P., Shead, L., Ivanov, V., Hoepping, K., Torque vectoring for electric vehicles with individually controlled motors: State-of-the-art and future developments, *World Electric Vehicle Journal*, 5, 2012, 617–628.
[12] Nemeth, T., Bubert, A., Becker, J.N., De Doncker, R.W., Sauer, D.U., A simulation platform for optimization of electric vehicles with modular drivetrain topologies, *IEEE Transactions on Transportation Electrification*, 4, 2018, 888–900.
[13] Parra, A., Zubizarreta, A., Pérez, J., A comparative study of the effect of Intelligent Control based Torque Vectoring Systems on EVs with different powertrain architectures, *IEEE Intelligent Transportation Systems Conf.*, 2019, 480–485.
[14] Nowlan, D., *The Dynamics of the Race Car*, ChassisSim, Sydney, 2010.
[15] Mikuláš, E., Gulan, M., G. Takács, G., Model predictive torque vectoring control for a formula student electric racing car, *Proc. European Control Conf.*, 2018, 581–588.
[16] Chatzikomis, C., Sorniotti, A., Gruber, P., Bastin, M., Shah, R.M., Orlov, Y., Torque-vectoring control for an autonomous and driverless electric racing vehicle with multiple motors, *SAE International Journal of Vehicle Dynamics, Stability, and NVH*, 1, 2017, 338–351.
[17] Azizi, A., Computer-Based Analysis of the Stochastic Stability of Mechanical Structures Driven by White and Colored Noise, *Sustainability*, 10, 2018, 3419.
[18] Antunes, J., Antunes, A., Outeiro, P., Cardeira, C., Oliveira, P., Testing of a torque vectoring controller for a Formula Student prototype, *Robotics and Autonomous Systems*, 113, 2019, 56–62.
[19] de Carvalho Pinheiro, H., Punta, E., Carello, M., Ferraris, A., Airale, A.G., Torque vectoring in hybrid vehicles with in-wheel electric motors: comparing SMC and PID control, *IEEE Int. Conf. on Environment and Electrical Engineering (EEEIC/I&CPS)*, 2021, 1–6.
[20] Asperti, M., Vignati, M., Sabbioni, E., On torque vectoring control: review and comparison of state-of-the-art approaches, *Machines*, 12, 2024, 160.
[21] Zhang, L., Ding, H., Shi, J., Huang, Y., Chen, H., Guo, K., Li, Q., An Adaptive Backstepping Sliding Mode Controller to Improve Vehicle Maneuverability and Stability via Torque Vectoring Control, *IEEE Transactions on Vehicular Technology*, 69, 2020, 2598–2612.
[22] Ammar, A., Bourek, A., Benakcha, A., Robust SVM-direct torque control of induction motor based on sliding mode controller and sliding mode observer, *Frontiers in Energy*, 14, 2020, 836–849.
[23] Zhao, J., Li, R., Yang, K., Liang, Z., Pi, D., Yang, Z., Wong, P.K., Discrete Fast Terminal Sliding Mode Control for Yaw Stability and Energy-Saving of FWID-EVs With Input Delay and Lumped Disturbances, *IEEE Transactions on Intelligent Vehicles*, 2024.
[24] Hameed, A.H., Al-Samarraie, S.A., Humaidi, A.J., Saeed, N., Backstepping-Based Quasi-Sliding Mode Control and Observation for Electric Vehicle Systems, *World Electric Vehicle Journal*, 15, 2024, 419.
[25] Parra, A., Zubizarreta, A., Pérez, J., An energy efficient intelligent torque vectoring approach based on fuzzy logic controller and neural network tire forces estimator, *Neural Computing and Applications*, 33, 2021, 9171–9184.
[26] Oubelaid, A., Albalawi, F., Rekioua, T., Ghoneim, S.S., Taib, N., Abdelwahab, S.A.M., Intelligent torque allocation based coordinated switching strategy for comfort enhancement of hybrid electric vehicles, *IEEE Access*, 10, 2022, 58097–58115.
[27] Herath, P., Rajasuriya, H., Dasanayake, N., Perera, S., Subasinghe, L., Gamage, J., Fuzzy-Based Vehicle Yaw Stability Control System with Torque Vectoring and Active Steering, *2024 Moratuwa Engineering Research Conference (MERCon)*, 2024, 700–705.
[28] Van Du Phan, V.Q.P., Dang, T.S., Trinh, N.H., Nguyen, P.N., Luong, N.M., Nguyen, B.U., Phan, Q.C., Design of Adaptive Fuzzy-PID for Adaptive Cruise Control of Electric Vehicle Using DC Motor, *Proc. Int. Conf. on Sustainability and Emerging Technologies for Smart Manufacturing*, Springer, 2024.
[29] Dalboni, M., Tavernini, D., Montanaro, U., Soldati, A., Concari, C., Dhaens, M., Sorniotti, A., Nonlinear Model Predictive Control for Integrated Energy-Efficient Torque-Vectoring and Anti-Roll Moment Distribution, *EEE/ASME Transactions on Mechatronics*, 26, 2021, 1212–1224.
[30] Cao, M., Hu, C., Wang, R., Wang, J., Chen, N., 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*, 235, 2021, 903–919.
[31] Kim, S.H., Kim, K.K.K., Model Predictive Control for Energy-Efficient Yaw-Stabilizing Torque Vectoring in Electric Vehicles with Four In-Wheel Motors, *IEEE Access*, 11, 2023, 37665–37680.
[32] Kang, Q., Meng, D., Jiang, Y., Zhao, C., Chu, H., Gao, B., Torque Vectoring Control of Distributed Drive Electric Vehicles Using Fast Iterative Model Predictive Control, *IFAC-PapersOnLine*, 58, 2024, 106–111.
[33] Taherian, S., Kuutti, S., Visca, M., Fallah, S., Self-Adaptive Torque Vectoring Controller Using Reinforcement Learning, *IEEE Intelligent Transportation Systems Conf.*, 2021, 172–179.
[34] Chen, H., Lv, C., RHONN-Modeling-Based Predictive Safety Assessment and Torque Vectoring for Holistic Stabilization of Electrified Vehicles, *IEEE/ASME Transactions on Mechatronics*, 28, 2022, 450–460.
[35] Favilli, A., Pugi, L., Berzi, L., Pierini, M., Coordinated Steering and Torque Vectoring Lateral Stability Sliding Mode Control Applied to an Electric In-Wheel Motors Vehicle, *IEEE EEEIC/I&CPS Europe Conf.*, 2021, 1–6.
[36] Jneid, M.S., Harth, P., Coordinate Torque Vectoring Control for Enhancing Handling and Stability of All-Wheel-Drive Electric Vehicles, *IEEE Int. Conf. on Cognitive Mobility*, 2023, 310–329.
[37] Sayssouk, W., Orjuela, R., Roos, C., Basset, M., 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*, 47, 2025, 1375-1386.
[38] Senofieni, R., Corno, M., Savaresi, S.M., Data-Driven Feedforward Compensation Tuning in Torque Vectoring Control, *IFAC-PapersOnLine*, 56, 2023, 115–120.
[39] Lu, Q., Sorniotti, A., Gruber, P., Theunissen, J., De Smet, J., H∞ Loop Shaping for the Torque-Vectoring Control of Electric Vehicles, *Mechatronics*, 35, 2016, 32–43.
[40] Wang, F., Wang, J.M., Wang, K., Zong, A., Hua, C., Adaptive Backstepping Sliding Mode Control of Uncertain Semi-Strict Nonlinear Systems and Application to Permanent Magnet Synchronous Motor, *Journal of Systems Science and Complexity*, 34, 2021, 552–571.
[41] Wang, G., Wang, D., Lin, H., Wang, J., Yi, X., A DC Error Suppression Adaptive Second-Order Backstepping Observer for Sensorless Control of PMSM, *IEEE Transactions on Power Electronics*, 39, 2024, 6664-6676.
[42] Vignati, M., Sabbioni, E., A Cooperative Control Strategy for Yaw Rate and Sideslip Angle Control Combining Torque Vectoring with Rear Wheel Steering, *Vehicle System Dynamics*, 60, 2022, 1668–1701.
[43] Zhu, S., Huang, X., Jiang, D., Wu, Z., Low-Cost Electric Bus Stability Enhancement Scheme Based on Fuzzy Torque Vectoring Differentials: Design and Hardware-in-the-Loop Test, *IFAC-PapersOnLine*, 54, 2021, 500–507.
[44] Dendaluce Jahnke, M., Cosco, F., Novickis, R., Pérez Rastelli, J., Gomez-Garay, V., Efficient Neural Network Implementations on Parallel Embedded Platforms, *Electronics*, 8, 2019, 250.
[45] Hong, C.J., Aparow, V.R., System Configuration of Human-in-the-Loop Simulation for Level 3 Autonomous Vehicle Using IPG CarMaker, *IEEE Int. Conf. on Internet of Things and Intelligence Systems*, 2021.
[46] Choo, C.L.W., *Real-Time Decision Making in Motorsports: Analytics for Improving Professional Car Race Strategy*, Doctoral Dissertation, Massachusetts Institute of Technology, USA, 2015.
[47] Lopez, G., Seaber, A., The Theory and Practice of Race Vehicle Data Acquisition and Analysis in Motor Sports Engineering Education, *Annual Conference & Exposition*, Jun 14, 2009, 14-1257.
[48] Sergers, J., *Analysis Techniques for Racecar Data Acquisition*, SAE International, 2014.
[49] Kegelman, J.C., Harbott, L.K., Gerdes, J.C., Insights into Vehicle Trajectories at the Handling Limits, *Vehicle System Dynamics*, 55, 2017, 191–207.
[50] La Regina, R., Russo, R.P., Pappalardo, C.M., Guida, D., Modeling an Autonomous Formula SAE Racing Car Using Simscape Multibody, *New Technologies, Development and Applications*, Springer, 2025.
[51] Govender, V., Barton, D.C., Multibody System Simulation and Optimisation of the Driving Dynamics of a Formula SAE Race Car, *SAE Technical Paper*, 2009.

[52] Balena, M., Mantriota, G., Reina, G., Dynamic Handling Characterization and Set-Up Optimization for a Formula SAE Race Car, *Machines*, 9, 2021, 126.
[53] Young, K.D., Utkin, V.I., Ozguner, U., A Control Engineer's Guide to Sliding Mode Control, *IEEE Transactions on Control Systems Technology*, 7, 1999, 328–342.
[54] Bai, Y., Wang, D., Fundamentals of fuzzy logic control—fuzzy sets, fuzzy rules and defuzzifications, *Advanced Fuzzy Logic Technologies in Industrial Applications*, Springer, 2006, 17–36.
[55] Uzunsoy, E., A brief review on fuzzy logic used in vehicle dynamics control, *Journal of Innovative Science and Engineering*, 2, 2018, 1-7.

Appendix A: Nomenclature

This section defines the symbols, parameters, and acronyms used throughout the paper.

Symbol	Description	Units
τ	Applied torque	N·m
τ_L, τ_R	Left and right wheel torques	N·m
τ_{ref}	Reference torque from controller	N·m
ω	Angular velocity of wheel	rad/s
ω_{ref}	Reference angular velocity	rad/s
$\dot{\omega}$	Angular acceleration of wheel	rad/s²
a_y	Lateral acceleration	m/s²
a_x	Longitudinal acceleration	m/s²
ψ	Yaw angle	rad
$\dot{\psi}$	Yaw rate	rad/s
$\ddot{\psi}$	Yaw acceleration	rad/s²
δ	Steering angle	rad
m	Vehicle mass	kg
I_z	Yaw moment of inertia	kg·m²
F_x, F_y	Longitudinal and lateral tire forces	N
F_z	Vertical tire load	N
v_x, v_y	Longitudinal and lateral velocities	m/s
L_f, L_r	Distance from center of gravity to front and rear axles	m
$C_{\alpha f}, C_{\alpha r}$	Cornering stiffness of front and rear tires	N/rad
μ	Tire-road friction coefficient	–
e_ψ	Yaw-rate error	rad/s
$e_{\dot{\psi}}$	Yaw-acceleration error	rad/s²
K_p, K_i, K_d	PID controller proportional, integral, and derivative gains	–
λ	Sliding surface coefficient (SMC)	–
η	Switching gain (SMC)	–
MF	Membership function (FLC)	–
RMSE	Root mean square of error	–
T_s	Settling time	s
t_p	Peak time	s
%OS	Percentage overshoot	%

Acronym	Description
PID	Proportional-Integral-Derivative Controller
SMC	Sliding Mode Controller
FLC	Fuzzy Logic Controller
TV	Torque Vectoring
VSM	Vehicle Simulation Model (AVL)
HIL	Hardware-in-the-Loop
FSAE	Formula Student / Formula SAE

Appendix B: Vehicle Model Parameters

This appendix presents the numerical values of the physical parameters used in the vehicle dynamic model. These include geometric, mass, tire, and aerodynamic properties relevant to the Formula SAE electric vehicle simulated in this study.

Parameter	Symbol	Value	Unit	Description
Vehicle mass	m	240	kg	Total vehicle mass
Yaw inertia	I_z	120	kg·m²	Moment of inertia about vertical axis
Wheelbase	L	1.55	m	Distance between front and rear axles
Front track width	T_f	1.20	m	Distance between front wheels
Rear track width	T_r	1.10	m	Distance between rear wheels
Front cornering stiffness	$C_{\alpha f}$	45,000	N/rad	Front tire cornering stiffness

Rear cornering stiffness	$C_{\alpha r}$	48,000	N/rad	Rear tire cornering stiffness
Tire radius	R_t	0.23	m	Effective rolling radius of tire
Center of gravity height	h_{CG}	0.28	m	Height of vehicle center of gravity
Aerodynamic drag coefficient	C_d	0.90	–	Dimensionless drag coefficient
Aerodynamic lift coefficient	C_l	1.30	–	Lift/downforce coefficient
Frontal area	A_f	1.10	m^2	Effective frontal area
Air density	ρ	1.225	kg/m^3	Ambient air density at sea level

Note: The parameter values are representative of a typical Formula SAE electric race car configuration and were used for dynamic simulation and controller evaluation in this study.

ORCID iD

Aydin Azizi https://orcid.org/0000-0001-6772-743X

How to cite this article: Julve A., Azizi A. Modelling, Control and Optimisation of an Electric Torque Vectoring System in a Formula SAE Race Car, *J. Appl. Comput. Mech.*, xx(x), 2025, 1–26. https://doi.org/10.22055/jacm.2025.48654.5398