FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Advanced control system of a formula student prototype with two-independent single-wheel motors

Davide Pinto Teixeira



Mestrado em Engenharia Informática e Computação

Supervisor: Prof. Rui Esteves Araújo

July 21, 2026

© Davide Pinto Teixeira, 2026

Advanced control system of a formula student prototype with two-independent single-wheel motors

Davide Pinto Teixeira

Mestrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

President: Prof. Konstantinos Bletsas
Referee: Prof. Ricardo Pinto de Castro
Referee: Prof. Rui Esteves Araújo

July 21, 2026

Resumo

A presente dissertação de mestrado enquadra-se no âmbito das atividades de desenvolvimento e de aprendizagem da equipa de estudantes da Faculdade de Engenharia da Universidade do Porto da Formula Student para veículos de propulsão elétrica.

Na dissertação apresenta-se uma revisão da dinâmica de veículos terrestres, o desenvolvimento e a implementação de um sistema de controlo vetorial do binário propulsor aplicado ao veículo elétrico de Formula Student com dois motores independentes no eixo traseiro. A arquitetura de controlo baseou-se no algoritmo de proposto por Fujimoto et al., adaptado e sintonizado para as características específicas de um veículo de Formula Student (FS FEUP 03). Sobre esta estrutura implementou-se um sistema de controlo de tração baseado no deslizamento da roda. O sistema foi desenvolvido no Robot Operating System 2 (ROS2) em C++, com vista à sua futura integração no veículo físico, e validado utilizando simulações em malha fechada com o software CarMaker da empresa IPG Automotive.

Os resultados obtidos demonstram uma melhoria consistente do desempenho do veículo em relação à configuração de referência de diferencial aberto. O sistema de controlo vetorial das rodas propulsoras permitiu reduzir o erro de yaw rate em relação à sua referência e melhorar o comportamento em curva. Nos testes habituais de skidpad, verificaram-se reduções do tempo de volta entre 5% e 13%, com ganhos mais expressivos para os coeficientes de fricção mais baixos. Adicionalmente, a ativação do sistema de controlo de tração contribuiu para a diminuição do deslizamento das rodas traseiras, mitigando a patinagem em condições de baixa aderência. Os resultados confirmam a eficácia da arquitetura proposta na melhoria do comportamento do veículo, das velocidades médias atingidas e da redução do tempo por volta do veículo FS FEUP 03, constituindo uma base para a futura validação experimental em pista com o protótipo físico.

Palavras-chave: torque vectoring • controlo de tração • veículo eléctrico • dinâmica veicular • Formula Student • IPG CarMaker

Abstract

This master's dissertation was developed in the scope of the Formula Student activities carried out by the team of students from the Faculty of Engineering of the University of Porto, focusing on the development of an electric Formula Student race car.

The dissertation presents a review of ground vehicle dynamics, as well as the development and implementation of a torque vectoring control system for an electric Formula Student vehicle equipped with two independent rear-wheel motors. The control architecture was based on the algorithm proposed by Fujimoto et al., which was adapted and tuned to the specific characteristics of the Formula Student vehicle (FS FEUP 03). A wheel slip-based traction control system was implemented on top of this architecture. The system was developed in C++ using the Robot Operating System 2 (ROS 2), with the aim of its future integration into the physical vehicle, and was validated through closed-loop simulations using IPG Automotive's CarMaker software.

The results demonstrate a consistent improvement in the vehicle's performance compared to the baseline configuration of open differential. The torque vectoring system reduced the yaw rate tracking error and improved the vehicle's cornering behaviour. In the standard Formula Student skidpad tests, lap time reductions ranging from 5% to 13% were achieved, with more significant improvements observed under lower friction coefficients. Furthermore, the activation of the traction control system reduced rear-wheel slip, mitigating wheel spin under low-grip conditions. Overall, the results confirm the effectiveness of the proposed control architecture in improving vehicle handling, increasing average vehicle speed, and reducing lap times for the FS FEUP 03, providing a solid foundation for future experimental validation using the physical prototype.

Keywords: torque vectoring • traction control • electric vehicle • vehicle dynamics • Formula Student • IPG CarMaker

UN Sustainable Development Goals

The United Nations Sustainable Development Goals (SDGs) provide a global framework to achieve a better and more sustainable future for all. It includes 17 goals to address the world's most pressing challenges, including poverty, inequality, climate change, environmental degradation, peace, and justice.

This dissertation, through the design and testing of a torque vectoring controller for two independent single-wheel motors, aligns with the following goals: SDG 9 (*Industry, Innovation and Infrastructure*).

SDG 9 Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation.

SDG	Target	Contribution	Performance Indicators and Metrics
9	9.5	This dissertation develops and validates, in simulation, a torque vectoring and traction control architecture for a Formula Student electric vehicle. The proposed controller contributes to research on advanced vehicle dynamics control and provides a foundation for future implementation and experimental validation.	Reduction in yaw-rate tracking error, lap time improvement (%), increase in average vehicle speed (%), reduction in wheel slip ratio (%)

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Professor Rui Esteves Araújo, for his guidance and the high standards he demanded throughout this work, which ultimately contributed to improving its quality. I would also like to give a special thanks to the Vehicle Dynamics department for their support throughout the system design process, as well as for providing valuable data and software resources. A special acknowledgement goes to Duarte Mendes, João Abreu, and the department leader, Luísa Carmo.

As the journey that led me here began in 2023, I would like to express my gratitude to all the members of FS FEUP with whom I had the privilege of collaborating throughout the years. In particular, I would like to highlight the Autonomous Systems department, especially during the 2024/2025 academic year, when I had the honor of leading the department and working alongside an exceptional team.

As this path was not only shaped by those who helped me grow intellectually, I would like to extend my heartfelt gratitude to everyone who supported me emotionally throughout the process of writing this dissertation. From my parents and my grandmother, who always provided me with the right advice for every challenge I faced, to my brother, who helped me put my problems into perspective with his sense of humor.

Having the right friendships also proved to be a great advantage throughout this odyssey. I would like to give a special thanks to Matilde Bicho, Diogo Costa, and the members of "Condomínio", people who always made me feel at ease and brought me comfort whenever we talked. I would also like to thank my friend Leonor Vieira for accompanying me throughout my entire journey at FEUP during these five years, starting from the very first day.

Finally, I would like to thank Marcelo Couto, a person with whom I had the opportunity to develop both technical and non-technical skills. A special thank you for challenging me to make this the topic of my thesis.

Davide Pinto Teixeira

Declaration of honour

I declare that this dissertation is my own work and has not previously been submitted or assessed at this or any other institution.References to other authors (statements, ideas, thoughts), as well as the use of published works, strictly comply with the rules of attribution and are duly identified in the text and in the bibliographic references, in accordance with the applicable referencing standards. I am aware that the practice of plagiarism or self-plagiarism constitutes an academic offence, as established in the U.Porto Code of Ethics.
I also declare that I have used Artificial Intelligence tools to complete part(s) of this dissertation, and that this use and its results are duly noted and have been carefully checked for errors, inaccuracies, unfounded attributions or other forms of bias in content and arguments, as well as determining authorship.

Davide Pinto Teixeira



"Sic parvis magna."

Sir Francis Drake

Contents

1 Introduction **1**
- 1.1 Context and Motivation . . . 1
- 1.2 Formula Student . . . 2
- 1.3 Problem Statement . . . 3
- 1.4 Main Contributions . . . 4
- 1.5 Dissertation Structure . . . 5

2 Vehicle Dynamics, Drivetrain Modeling, and Control **7**
- 2.1 Vehicle Dynamics . . . 7
  - 2.1.1 Vehicle Axis System . . . 8
  - 2.1.2 Normal Force . . . 8
  - 2.1.3 Tire Model . . . 12
  - 2.1.4 Longitudinal Vehicle Dynamics . . . 15
  - 2.1.5 Lateral Dynamics . . . 17
- 2.2 Mechanical Differentials . . . 21
  - 2.2.1 Open Differential . . . 21
  - 2.2.2 Locking Differential . . . 21
  - 2.2.3 Limited-Slip Differential (LSD) . . . 22
- 2.3 Traction Control . . . 22
- 2.4 Torque Vectoring . . . 24
  - 2.4.1 Reference Generator . . . 26
  - 2.4.2 High Level Controller . . . 27
  - 2.4.3 Low Level Controller . . . 34

3 Torque Vectoring Implementation **39**
- 3.1 Architecture . . . 39
- 3.2 Simulation Environment . . . 44
  - 3.2.1 Motor . . . 45
  - 3.2.2 Tires . . . 46
  - 3.2.3 Driver Model . . . 47
  - 3.2.4 Simulation Environment Validation . . . 49
- 3.3 Algorithm . . . 52
  - 3.3.1 Reference Generator . . . 53
  - 3.3.2 High-Level Controller . . . 54
  - 3.3.3 Low-Level Controller . . . 56
  - 3.3.4 Traction Control . . . 56

4 Results and Discussion **59**
4.1 Open-Loop Maneuvers with Constant Steering 59
4.2 Skidpad 65
4.2.1 High-Friction Surface 67
4.2.2 Medium-Friction Surface 70
4.2.3 Low-Friction Surface 73
4.2.4 Summary of Results 78

5 Conclusion **79**
5.1 Limitations 80
5.2 Future Work 80

References **83**

A Motor Information **89**

B C++ Implementations **91**
B.1 Steady-State Yaw Rate Reference Generator 91
B.2 PID High Level Controller 92
B.3 Low Level Controller 95
B.4 Traction Control 95

C Formula Student Disciplines **97**
C.1 Overview of Competition Disciplines 97
C.2 Static Events 97
C.2.1 Static Events Points Distribution 98
C.2.2 Engineering Design 98
C.2.3 Cost and Manufacturing 99
C.2.4 Business Plan Presentation 99
C.3 Dynamic Events 99
C.3.1 Dynamic Events Points Distribution 99
C.3.2 Acceleration 99
C.3.3 Skidpad 99
C.3.4 Driverless Acceleration 100
C.3.5 Driverless Skidpad 100
C.3.6 Autocross 100
C.3.7 Endurance 100
C.3.8 Efficiency 100

List of Figures

1.1 FS FEUP 02 competing in the DV Autocross at Formula Student Germany 2025 @FSG - Andrae. . . . 3

2.1 Vehicle Axis System ISO 8855-2011, from [20]. . . . 8
2.2 Original sine curve of the magic formula, from [28]. . . . 13
2.3 Bristle model representation, from [31]. . . . 14
2.4 Representation of the slip angle α in a 4 wheel vehicle model, from [34]. . . . 18
2.5 Generic architecture of TC/ABS system, adapted from [45]. . . . 23
2.6 Illustration of yaw rate enhancement through differential longitudinal force generation. In the baseline/open case (a), the longitudinal tire forces are balanced. In the torque-vectoring case (b), a greater longitudinal force is applied at the outer wheel, producing an additional yaw moment that increases the vehicle yaw rate and improves cornering response. Arrow lengths indicate the relative magnitude of the longitudinal tire forces. . . . 24
2.7 Classification of yaw stability control types [51]. . . . 25
2.8 General torque vectoring control architecture for yaw moment control. . . . 26
2.9 Torque Vectoring Architecture with a PI Controller (FST Lisboa) [11]. . . . 28
2.10 Torque Vectoring control scheme of Fujimoto et al. [16]. . . . 29
2.11 Torque Vectoring control scheme (ETSEIB MotorSport) [13]. . . . 30

3.1 Torque vectoring system communication interfaces and architecture. . . . 40
3.2 Vehicle interfaces for the torque vectoring implementation. The ECU receives vehicle state information from the state estimator, which processes data from the IMU, wheel speed sensors, and motor resolvers. The ECU then computes the torque commands sent to the rear motors. . . . 40
3.3 UML component diagram of the torque vectoring communication architecture and its interface with the IPG CarMaker simulation environment. . . . 41
3.4 Simulink TCP/IP interface for transmitting driver steering and throttle inputs to the torque vectoring controller [67]. . . . 41
3.5 UML Class Diagram representing the architecture of the torque vectoring proposed solution. . . . 43
3.6 UML sequence diagram of the torque vectoring system execution flow. . . . 44
3.7 Wheel torque vs. vehicle velocity characteristic curve of the DTI F-MOT motor, adapted from [71]. . . . 46
3.8 Hoosier 43075 16x7.5-10 R20 tires mounted on the FS FEUP 02 prototype. . . . 47
3.9 Interface between the IPGDriver module and the vehicle model, showing the inputs received by the driver (top) and the control outputs it produces (bottom). Adapted from [72]. . . . 47

3.10 GG diagrams for different driver behaviours, from [17, 74]. 48
3.11 IPG CarMaker driver's longitudinal parameters, from [17]. 49
3.12 IPG CarMaker driver's lateral parameters, from [17]. 49
3.13 Effect of throttle on performance in the 75 m acceleration event: (left) lap time and average velocity; (right) velocity profiles for different throttle inputs. 50
3.14 ISO 4138 test results: (left) distance traveled for different throttle inputs; (right) yaw rate response over time. 51
3.15 Longitudinal velocity as a function of time for different throttle inputs. 51
3.16 Yaw rate response under left-wheel torque input. 52
3.17 Yaw rate response under right-wheel torque input. 52
3.18 Hierarchical yaw-rate tracking and torque distribution control architecture for the rear-wheel torque vectoring system. The upper-level controller generates the desired corrective yaw moment based on the yaw-rate tracking error using a PI controller, while the low-level controller allocates the required wheel torques to the left and right rear wheels. The resulting wheel torques are applied to the vehicle dynamics model, and the measured yaw response is fed back through a first-order vehicle yaw dynamics model for closed-loop stability enhancement. 53
3.19 Architecture of the High-Level Controller System. 55

4.1 Comparison between reference and actual yaw-rate responses for different vehicle velocities under a constant 10-degree steering input. 60
4.2 Comparison between actual sideslip angles for different vehicle velocities under a constant 10-degree steering input. 61
4.3 Comparison of vehicle trajectories for Torque Vectoring (TV) and non-Torque Vectoring (non-TV) architectures at a constant steering angle of 10° and longitudinal velocities of 1, 4, and 16 m/s. 62
4.4 Comparison of average and peak yaw rates as a function of vehicle velocity for the TV and No TV configurations under a constant 10-degree steering input. . . . 62
4.5 Comparison of yaw-rate tracking performance metrics as a function of vehicle velocity for the TV and No TV configurations under a constant 10-degree steering input. 63
4.6 Rear wheel longitudinal tire forces for the torque-vectoring configuration at $v = 1\,\mathrm{m/s}$, $v = 4\,\mathrm{m/s}$ and $v = 16\,\mathrm{m/s}$. 65
4.7 Formula Student Skidpad Track, from [76]. 66
4.8 Yaw-rate tracking comparison for $\mu = 0.9$. 67
4.9 Longitudinal velocity over time for TV and baseline implementations and $\mu = 0.9$. 68
4.10 Lateral acceleration comparison for $\mu = 0.9$. 68
4.11 Yaw Rate Tracking error over time for TV implementation adn $\mu = 0.9$. 69
4.12 TV implementation vehicle trajectory for $\mu = 0.9$. 70
4.13 Yaw-rate tracking comparison for $\mu = 0.6$. 71
4.14 Longitudinal velocity over time for TV and baseline implementations and $\mu = 0.6$. 71
4.15 Lateral acceleration comparison for $\mu = 0.6$. 72
4.16 Yaw-rate tracking error over time for TV implementation and $\mu = 0.6$. 73
4.17 TV implementation vehicle trajectory for $\mu = 0.6$. 73
4.18 Yaw-rate tracking comparison for $\mu = 0.3$. 74
4.19 Longitudinal velocity over time for TV and baseline implementations and $\mu = 0.3$. 75
4.20 Slip ratios of the rear wheels during the skidpad test. 75
4.21 Lateral acceleration comparison for $\mu = 0.3$. 76
4.22 Yaw-rate tracking error over time for TV implementation and $\mu = 0.3$. 77

4.23 TV implementation vehicle trajectory for $\mu = 0.3$. 77

A.1 DTI F-MOT motor datasheet [71]. 89

C.1 Formula Student points distribution by event. 98

List of Tables

2.1 Comparison of tire model categories and representative models. 15
2.2 Comparison of differential types and torque vectoring systems. 22
2.3 Torque Vectoring Control Architecture. 26
2.4 Comparison of vehicle performance with torque vectoring (TV) enabled and disabled [13]. 30
2.5 Comparison of high-level torque vectoring controllers. 35
2.6 Comparison of low-level torque allocation controllers. 38

3.1 FS FEUP 03 vehicle parameters. 45
3.2 Tuned parameters of the PI yaw moment controller. 56
3.3 Traction control parameters. 58

4.1 Tracking error comparison vs. velocity. 64
4.2 Yaw comparison vs. velocity. 64
4.3 Effect of torque vectoring on skidpad performance for $\mu = 0.9$. 68
4.4 Effect of torque vectoring on skidpad performance for $\mu = 0.6$. 72
4.5 Effect of torque vectoring on skidpad performance for $\mu = 0.3$. 76
4.6 Effect of torque vectoring on skidpad lap time. 78

C.1 Static events points distribution. 98
C.2 Dynamic events points distribution. 99

Nomenclature

Angles and Kinematics

ψ	Yaw angle	rad
$\dot{\psi}$	Yaw rate	$\mathrm{rad\,s^{-1}}$
$\ddot{\psi}$	Yaw acceleration	$\mathrm{rad\,s^{-2}}$
β	Vehicle sideslip angle	rad
α	Tire slip angle	rad
δ	Steering angle	rad
δ_l	Steering angle of left wheel	rad
δ_r	Steering angle of right wheel	rad
δ_i	Steering angle of inner wheel	rad
δ_o	Steering angle of outer wheel	rad
γ	Camber angle	rad

Velocities and Accelerations

v_x	Longitudinal velocity	$\mathrm{m\,s^{-1}}$
v_y	Lateral velocity	$\mathrm{m\,s^{-1}}$
ω	Angular velocity	$\mathrm{rad\,s^{-1}}$
a_x	Longitudinal acceleration	$\mathrm{m\,s^{-2}}$
a_y	Lateral acceleration	$\mathrm{m\,s^{-2}}$

Vehicle Geometry and Mass Properties

m	Vehicle mass	kg
h_{CG}	Height of center of gravity	m
l_f	Distance from CG to front axle	m
l_r	Distance from CG to rear axle	m
L	Wheelbase	m
t	Track width	m
t_f	Front track width	m
t_r	Rear track width	m
I_{zz}	Yaw moment of inertia	$\mathrm{kg\,m^2}$
R_{wheel}	Wheel radius	m

Forces and Moments

F_x	Longitudinal force	N
F_y	Lateral force	N
F_z	Vertical force	N
M_z	Yaw moment	N m
$M_{z,ij}$	Self-aligning torque	N m
R_x	Rolling resistance force	N

Tire and Contact Parameters

σ_x	Longitudinal slip ratio	-
K_y	Cornering stiffness	$\mathrm{N\,rad^{-1}}$
z	Average bristle deflection	m
μ	Friction coefficient	-

Load Transfer and Roll

ΔW_x	Longitudinal load transfer	N
ΔW_y	Lateral load transfer	N

Aerodynamics

ρ	Air density	$\mathrm{kg\,m^{-3}}$
C_d	Aerodynamic drag coefficient	-
C_l	Aerodynamic lift coefficient	-
A_F	Frontal area	$\mathrm{m^2}$
F_{drag}	Aerodynamic drag force	N
F_{down}	Aerodynamic downforce	N

Powertrain and Control

G_r	Gear ratio	-
T	Torque	-
ΔT	Torque variation	N m
k_p	Proportional gain	-
k_i	Integral gain	-
k_d	Derivative gain	-

Constants

g	Gravitational acceleration	$\mathrm{m\,s^{-2}}$

List of Acronyms

ABS	Anti-lock Braking System
AWD	All-Wheel Drive
CG	Center of Gravity
CP	Center of Pressure
DOF	Degrees of Freedom
DV	Driverless
ECU	Electronic Control Unit
ESC	Electronic Stability Control
FL	Front Left
FR	Front Right
FS	Formula Student
FSG	Formula Student Germany
FS FEUP	Formula Student FEUP
FWD	Front Wheel Drive
LQR	Linear Quadratic Regulator
MDP	Markov Decision Process
MPC	Model Predictive Control
PID	Proportional-Integral-Derivative
RWD	Rear Wheel Drive
RL	Rear Left
ROS	Robot Operating System
RR	Rear Right
SAS	Steering Angle Sensor
TCS	Traction Control System
TV	Torque Vectoring
UKF	Unscented Kalman Filter
V2V	Vehicle-to-Vehicle Communication
VVM	Virtual Vehicle Model
WSS	Wheel Speed Sensor

Chapter 1

Introduction

The rapid growth of electric mobility has transformed the automotive industry, with global electric vehicle sales exceeding 20 million units in 2025 and accounting for approximately one-quarter of all new car sales worldwide [1]. This expansion has accelerated the development of new vehicle architectures and powertrain configurations that were previously unfeasible in conventional internal combustion vehicles.

Among the most significant innovations is the adoption of independent wheel motor architectures, typically implemented through two or four motor configurations. Unlike conventional drivetrains, where torque distribution is constrained by mechanical components such as differentials and transmissions, electric powertrains enable independent torque control at each wheel. This capability not only improves packaging flexibility and drivetrain efficiency but also provides new possibilities for vehicle dynamics control. The fast and precise torque response of electric motors, combined with their distributed installation, enables advanced control functions such as traction control, direct yaw control, and torque vectoring through software-based coordination of individual wheel torques [2, 3].

This transition reflects a broader shift within the automotive industry toward software-defined vehicles, in which vehicle behaviour is determined not only by mechanical design but also by increasingly sophisticated control algorithms. Consequently, advanced control strategies have become a key enabler of next-generation electric, providing new opportunities to enhance dynamic performance while maintaining efficiency and safety.

1.1 Context and Motivation

In the context of electric mobility, the field of vehicle dynamics has seen increasing interest in the development of advanced control systems aimed at improving vehicle stability, handling, and safety. Controllers such as yaw-rate controllers and slip-ratio controllers (e.g., Anti-lock Braking Systems) play a crucial role in maintaining desired vehicle behaviour under varying driving conditions by regulating tire forces and vehicle motion [4].

The importance of vehicle dynamics control extends beyond performance enhancement. Advanced control systems have been shown to significantly improve road safety by reducing the likelihood of vehicle instability and loss-of-control situations. For example, Electronic Stability Control (ESC) has been associated with substantial reductions in single-vehicle, head-on, and fatal multi-vehicle accidents by actively assisting the driver in maintaining vehicle stability [5].

By regulating wheel torques individually, it becomes possible to directly influence vehicle yaw motion and tire force distribution, enabling advanced control strategies such as torque vectoring. These systems can actively generate corrective yaw moments and adapt vehicle behaviour according to the driving situation, extending the capabilities of traditional stability control systems [6, 7].

Torque vectoring can provide several benefits for vehicle performance. Through appropriate torque distribution, vehicle behaviour during acceleration and cornering can be modified to improve agility, stability, and efficiency [8, 9]. In addition to enhancing dynamic performance, torque vectoring can also contribute to energy efficiency by minimizing wheel slip and improving the effectiveness of regenerative braking, thereby increasing energy recovery and potentially extending the vehicle driving range. As a result, torque vectoring offers benefits in both vehicle dynamics and energy consumption [10].

Within the Formula Student environment, advanced control systems have increasingly been adopted and optimized for high-performance electric prototypes. For instance, in [11], a yaw-rate controller based on both PID and LQR approaches was proposed and analysed, demonstrating the potential of active control strategies to improve the handling performance of electric racing vehicles.

1.2 Formula Student

Formula Student is an international engineering competition that challenges students from universities worldwide to design, manufacture, validate, and compete with a formula-style race car. In recent years, the competition has expanded to include autonomous vehicle challenges, which now account for approximately 15% of the total electric competition events [12].

FS FEUP is the Formula Student team from the Faculty of Engineering of the University of Porto (FEUP). Founded in 2021, the team developed its first electric prototype, the FS FEUP 01, which competed in 2023. Building upon this foundation, the team designed its first fully electric and autonomous vehicle during the 2023 to 2025 season, the FS FEUP 02 (Figure 1.1). This vehicle was subsequently manufactured, assembled, and tested in 2025, marking FS FEUP's participation in Formula Student Netherlands, Formula Student Portugal, and Formula Student Germany, the largest and most prestigious university engineering competition in the world.

The FS FEUP 02 was equipped with a single rear-mounted electric motor coupled to a conventional mechanical transmission and differential. However, this architecture does not allow independent control of the torque delivered to each rear wheel.

As part of the continuous evolution of the team's electric vehicle platform, the next-generation prototype, FS FEUP 03, is being developed with a dual-motor rear-wheel-drive architecture. By employing one motor per rear wheel, the vehicle will enable independent wheel torque control and provide the necessary hardware foundation for the implementation of advanced vehicle dynamics control strategies, such as torque vectoring and traction control.

The transition to this architecture is expected to enhance vehicle performance by improving traction, stability, and handling capabilities while creating new opportunities for the development and validation of advanced control algorithms. This evolution constitutes one of the primary motivations for the work presented in this dissertation.



Figure 1.1: FS FEUP 02 competing in the DV Autocross at Formula Student Germany 2025 @FSG - Andrae.

1.3 Problem Statement

The transition from conventional electric powertrain architectures to independently actuated wheel systems introduces new opportunities for vehicle dynamics control. However, the additional control authority provided by independent wheel actuation also increases the complexity of the control problem. To effectively exploit the capabilities of a dual-motor architecture, it is necessary to develop control strategies capable of regulating wheel torques in real time while maintaining vehicle stability, handling performance, and traction under varying operating conditions.

Within the FS FEUP project, the current vehicle platform relies on a single-motor powertrain coupled to a mechanical differential, which limits the ability to actively control the torque delivered to each driven wheel. The development of the next-generation FS FEUP 03 prototype, featuring two independently actuated rear wheels, creates the opportunity to implement advanced vehicle dynamics control functions such as torque vectoring and traction control. However, the design, implementation, and validation of these systems require a thorough understanding of vehicle dynamics, control theory, and the interaction between the vehicle and the tire-road interface.

Furthermore, the effectiveness of such control systems depends not only on the control algorithms themselves but also on their integration within the vehicle software architecture and their performance under realistic operating conditions. Consequently, there is a need to investigate suitable control strategies capable of improving vehicle dynamic behaviour while remaining computationally efficient and robust enough for real-time implementation in a Formula Student prototype.

Given the challenges associated with implementing torque vectoring control in independently driven electric vehicles, this work seeks to address the following research question:

> *How can a torque vectoring controller be designed and validated to improve the yaw stability and handling performance of a two-motor electric Formula Student vehicle?*

The main objective of this dissertation is to design, implement, and evaluate a torque vectoring controller for a two-motor Formula Student electric vehicle using a representative simulation environment.

Specific objectives include:

- Improve yaw stability and cornering performance compared to baseline open-loop or evenly distributed torque control strategies [11, 13, 8, 6, 14];
- Maintain controllability under variable friction conditions [15];
- Operate within the computational and hardware limitations of a Formula Student vehicle control unit;
- Evaluate the controller through simulation scenarios representative of Formula Student driving conditions.

1.4 Main Contributions

The main contributions of this dissertation are the following:

- **Integration of a peak-seeking traction control system.** A slip-ratio based traction control layer was developed and integrated into the hierarchical architecture, enabling combined yaw stability and longitudinal traction management without explicit road surface estimation.
- **Adaptation of Fujimoto's torque vectoring architecture to a Formula Student context.** The hierarchical yaw-rate tracking controller proposed by Fujimoto et al. [16] was re-parameterised and extended for the specific constraints of a Formula Student electric vehicle, including the DTI F-MOT motor characteristics, Formula Student track geometries, and the operating envelope of the FS FEUP 03 prototype.
- **Validation in a simulation environment.** The complete control architecture was validated using IPG CarMaker [17] with a vehicle model representative of the FS FEUP 03 prototype, including nonlinear tire dynamics, motor models, and a closed-loop driver model.

- **ROS-based implementation ready for real-vehicle deployment.** The controller was implemented as a ROS2 node, directly compatible with the autonomous system architecture of the FS FEUP 03 prototype, facilitating future deployment without significant software restructuring.

1.5 Dissertation Structure

This dissertation is organized into six chapters.

Chapter 1 introduces the motivation for the work and presents the research problem, objectives, hypothesis, and research question. The chapter also provides the context of electric mobility, vehicle dynamics control, and the Formula Student environment in which the work is developed.

Chapter 2 presents the theoretical foundations required for the development of the proposed control strategy and reviews the relevant state of the art in vehicle dynamics control. Topics such as vehicle dynamics, tire behaviour, yaw motion, differential systems, traction control, and torque vectoring for electric vehicles are introduced and discussed. Existing approaches are analysed and compared, highlighting their advantages, limitations, and applicability to Formula Student vehicles, thereby establishing the necessary background for the subsequent chapters.

Chapter 3 presents the methodology adopted throughout this work. It describes the development of the proposed torque vectoring controller, the simulation environment used for its evaluation, and the procedures employed to assess its performance under representative Formula Student operating conditions.

Chapter 4 presents and discusses the obtained results. The performance of the proposed controller is evaluated through a series of simulation experiments and compared with baseline approaches, assessing its impact on vehicle stability, handling performance, and overall behaviour.

Finally, Chapter 5 summarizes the main contributions of the dissertation, discusses the achieved objectives, and outlines possible directions for future work, particularly regarding the implementation and validation of the proposed controller on future FS FEUP vehicle platforms.

Chapter 2

Vehicle Dynamics, Drivetrain Modeling, and Control

This chapter reviews the fundamental concepts required for the development and analysis of torque vectoring control systems. Particular attention is given to vehicle dynamics, tire-road interaction, and torque distribution mechanisms, as these directly influence vehicle handling, stability, and performance. These concepts establish the theoretical foundation for the control strategies investigated in this dissertation.

2.1 Vehicle Dynamics

Vehicle Dynamics is the field of study concerned with how vehicles respond to various forces and inputs. It includes the analysis of vehicle motion by considering factors such as tire-road interaction, suspension dynamics, aerodynamics, and driver input [18].

Vehicle models are mathematical representations of a vehicle's physical behavior. They are employed to analyze and predict vehicle dynamics, performance, and control characteristics. This chapter provides an overview of different types of vehicle models, including tire models, longitudinal and lateral dynamics, and ackermann steering geometry.

Depending on the level of detail required, vehicle models can range from highly simplified to complex, nonlinear representations. Simplified models allow for a clearer understanding of fundamental behaviors, while detailed models provide precise insights necessary for advanced control and optimization tasks.

This chapter presents the key considerations involved in developing vehicle models, including the representation of tire slips, longitudinal and lateral forces, and the resulting moments acting on the vehicle. It also discusses how these factors influence overall vehicle motion and stability, providing a foundation for understanding the subsequent modeling approaches and analyses presented in later sections.

2.1.1 Vehicle Axis System

For the purpose of this study, the vehicle reference frame shown in Figure 2.1 has been selected. This coordinate system follows the ISO 8855-2011 [19] standard and will be used consistently in all subsequent vehicle dynamics analyses.

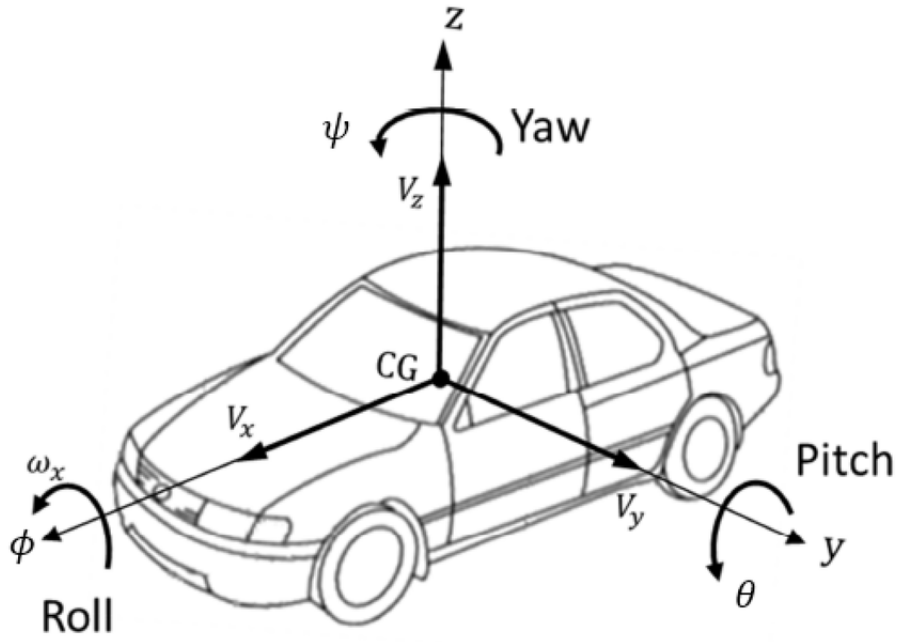


Figure 2.1: Vehicle Axis System ISO 8855-2011, from [20].

2.1.2 Normal Force

The normal force (F_z), represents the vertical load acting on a vehicle's tires. It is a fundamental variable in vehicle dynamics because it directly determines the interaction between the tires and the road. It has some importance in some calculations:

- **Tire forces:** The maximum longitudinal and lateral forces a tire can generate are dependent to the normal load. This relationship governs acceleration, braking, and cornering abilities [21].
- **Rolling resistance:** The opposing force resistive motion depends on the vertical load on each tire. Higher normal forces increase rolling resistance, affecting vehicle efficiency and energy consumption [4].
- **Weight transfer:** Changes in normal force during acceleration, braking, or cornering provide information on dynamic weight transfer and vehicle stability [22].
- **Aerodynamic effects:** Downforce generated by aerodynamic components modifies the normal load on the tires, improving grip and influencing the handling characteristics [4].

Due to its influence on tire-road interactions and vehicle performance, the calculation of the normal force under different driving conditions is a prerequisite for modeling in vehicle dynamics. Formula 2.1 represents the 2 main variables to consider in the calculation of F_z - vehicle weight and aerodynamic downforce.

$$F_z = -mg + F_{down} \tag{2.1}$$

with

- m the mass of the vehicle,
- g the gravitational acceleration,
- F_{down} the downforce acting on the vehicle,

2.1.2.1 Aerodynamic Downforce

Aerodynamic downforce (F_{down}) is the force acting along the z-axis, generated by the vehicle's aerodynamic components such as wings, spoilers, and diffusers. This force is beneficial in vehicle dynamics because it increases the total normal load (F_z) without directly adding mass to the vehicle. An increase in F_z enhances the tire-road grip and, consequently, improves the vehicle's handling performance. Equation 2.2 describes how aerodynamic downforce is generated [22].

$$F_{down} = -\frac{1}{2}\rho C_l A_F v_x^2 \tag{2.2}$$

with

- ρ the mass density of air,
- C_l the aerodynamic lift coefficient,
- A_F the frontal area of the vehicle,
- v_x the longitudinal vehicle velocity.

Furthermore, the aerodynamic downforce can be distributed among the four wheels according to the longitudinal position of the aerodynamic center of pressure (CP), which determines the aerodynamic balance of the vehicle [23]. Assuming a symmetric left-right distribution, the wheel downforces are obtained from static equilibrium:

$$
\begin{aligned}
F_{\text{down,FR}} &= \frac{L - x_{CP}}{L} \cdot \frac{F_{\text{down}}}{2}, \\
F_{\text{down,FL}} &= \frac{L - x_{CP}}{L} \cdot \frac{F_{\text{down}}}{2}, \\
F_{\text{down,RR}} &= \frac{x_{CP}}{L} \cdot \frac{F_{\text{down}}}{2}, \\
F_{\text{down,RL}} &= \frac{x_{CP}}{L} \cdot \frac{F_{\text{down}}}{2}.
\end{aligned}
\tag{2.3}
$$

with

- L the wheelbase of the vehicle,
- x_{CP} the longitudinal position of the aerodynamic center of pressure measured from the front axle,
- F_{down} the total aerodynamic downforce acting on the vehicle.

2.1.2.2 Load Transfer

Besides aerodynamic influences, the distribution of normal loads on a tires of a vehicle another important factor is load transfer. The car's mass distribution changes because of the inertial forces acting on the center of gravity during dynamic maneuvers such as acceleration, braking, and cornering. As a result of this process, there tire loadings get changed, i.e. the vertical loads that each tire experiences, since the total weight of the vehicle is being redistributed among the tires [22].

Load transfer is an important concept of vehicle dynamics since it impacts that tire grip directly, handling balance, and overall stability. The effects of this phenomenon can be utilized for vehicle optimization under various driving conditions.

Longitudinal Load Transfer

During acceleration or braking, longitudinal load transfer occurs because the traction and braking forces act at the tire-road interface, while the vehicle's center of gravity is located above this point. This difference in force application creates a moment about the center of gravity, resulting in a shift of vertical load between the front and rear axles, as described in Equation 2.4 [22].

$$
\Delta W_x = \frac{F_x \cdot h_{CG}}{L} \tag{2.4}
$$

with

- F_x the total longitudinal force acting on the vehicle,
- h_{CG} the height of the center of gravity above the ground,
- L the wheelbase of the vehicle.

where h_{CG} is the height of the center of gravity and F_x is the longitudinal force (positive for acceleration, negative for braking). This load transfer affects the normal force on the front and rear axis.

Lateral Load Transfer

Analogous to longitudinal load transfer, during cornering, lateral load transfer occurs because the lateral tire forces are generated at the road surface while the vehicle's inertia acts at its center of gravity, creating an unbalanced moment (described in Equation 2.5) [22].

$$\Delta W_y = \frac{F_y \cdot h_{CG}}{t} \tag{2.5}$$

with

- F_y the total lateral force acting on the vehicle,
- h_{CG} the height of the center of gravity above the ground,
- t represent the track width of one of the axles (assuming $t_f = t_r$).

Equation 2.5 shows that an increase in lateral acceleration transfers load from the inner wheels to the outer wheels, increasing the available tire forces on the outside tires while reducing them on the inside tires.

F_z Distribution on 4-wheels

By considering the combined effects of static weight, dynamic load transfer, and aerodynamic downforce, the total normal forces acting on the front and rear tires can be determined as in equation 2.6:

$$\begin{aligned}
F_{z,FL} &= \frac{l_r}{L} mg - \Delta W_x - \Delta W_y + F_{down,FL} \\
F_{z,FR} &= \frac{l_r}{L} mg - \Delta W_x + \Delta W_y + F_{down,FR} \\
F_{z,RL} &= \frac{l_f}{L} mg + \Delta W_x - \Delta W_y + F_{down,RL} \\
F_{z,RR} &= \frac{l_f}{L} mg + \Delta W_x + \Delta W_y + F_{down,RR}
\end{aligned} \tag{2.6}$$

with

- l_r the distance from the center of gravity to the rear axle,
- l_f the distance from the center of gravity to the front axle,
- m the mass of the vehicle,

- g the gravitational acceleration,
- ΔW_x the longitudinal load transfer,
- ΔW_y the lateral load transfer,
- $F_{\text{down},i,j}$ the aerodynamic downforce acting on wheel i, j.

2.1.3 Tire Model

Tires are one of the main components in vehicle dynamics, as all vehicle movement can be described by the forces and moments generated at the tire-road interface. To capture these interactions, a tire model is required to represent the forces and moments produced by the tires.

A tire model is a mathematical representation of tire behavior that models how a tire generates forces for a specific set of inputs such as the angle of slip and the vertical load. Models the relationship between tire movement and the resulting longitudinal, lateral, and aligning forces, creating a vehicle dynamics simulation interface that includes vehicle design, performance evaluation, and safety studies [24].

Tire models can be classified into three main categories [25]:

- **Slip/Force maps:** Define a mapping function between slip and friction force, e.g. $s \mapsto F(s)$, typically obtained from experimental data. Some models may also include velocity, that is, $F(s,v)$. An example is the Pacejka tire model [21].
- **Lumped models:** Assume a point-like contact between a tire and a road. An example is the LuGre tire model [26].
- **Distributed models:** Consider the tire as a deformable body, taking into account the distribution of forces along the contact patch between the tire and the road.

2.1.3.1 Linear Tire Models

Linear tire models are representations of tire behavior that assume a linear function of the resulting force and the tire slip (longitudinal or lateral) and the resulting forces. Although they do not account for non-linear effects such as saturation at high slip angles, linear tire models provide a foundation for understanding vehicle responses and are used in preliminary simulations, analytical studies, and real-time control applications.

For example, in PC-Crash [27], the linear tire model assumes a linear increase up to a saturation point. Beyond this point, the lateral force remains constant, representing the maximum tire capacity, as shown in the equation 2.7.

$$F_y(\alpha) = \begin{cases} C_\alpha \alpha, & |\alpha| \leq \alpha_{\max} \\ F_{y,\max} \operatorname{sign}(\alpha), & |\alpha| > \alpha_{\max} \end{cases} \tag{2.7}$$

with

- C_α is the slip coefficient [27],
- α represents the slip angle,
- α_{max} is the saturation slip angle

2.1.3.2 Pacejka Tire Model

The Pacejka tire model, also known as the Magic Formula, is an empirical model that describes the relationship between tire forces and slip angles. It is characterized by a set of coefficients that can be adjusted to fit the experimental data. With this formula, it is possible to derive values such as F_x, F_y and M_z as a function of inputs such as the angle of slipping α and the longitudinal slipping σ_x [21] The general form of the Pacejka tire model is given by Equation 2.8.

$$y = D\sin[C\arctan\{Bx - E(Bx - \arctan(Bx))\}] \tag{2.8}$$

The parameters B, C, D, and E, with a graphical representation on Figure 2.2, have some physical meaning:

- B is the stiffness factor that influences the slope of the curve at the origin.
- C is the shape factor that affects the overall shape of the curve (sine function).
- D is the peak factor that determines the maximum value of the curve.
- E is the curvature factor, which adjusts the curvature of the curve near the peak.

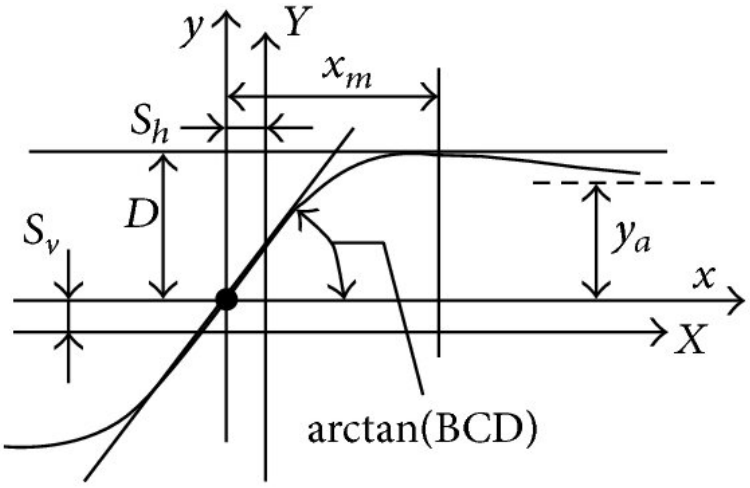


Figure 2.2: Original sine curve of the magic formula, from [28].

.

In this model, the value of the cornering stiffness is given by the product of B_y, C_y and D_y, which are the coefficients for calculating the value of F_y [21].

All these parameters vary with other variables, such as vertical load F_z. For example, corner stiffness increases with vertical load, which means that the tire generates more lateral force for a given slip angle when the load is higher, as shown in formula 2.9 [24].

$$K_y = B_y \cdot C_y \cdot D_y = a_3 \cdot \sin(2 \cdot \arctan(a_4 \cdot F_z)) \tag{2.9}$$

where
a_3 = maximum cornering stiffness (at camber angle (γ) = 0)
a_4 = load at maximum cornering stiffness

This model has some different variations, such as the one seen in [29], which includes the inflation pressure of the tire in the cornering stiffness K_y calculation by including scaling factors that depend quadratically on the inflation pressure. In [30], a thermal model is added, making the different parameters of the Pacejka model functions of the tire surface temperature. In [24], the magic formula is adapted in the cornering stiffness calculation to include the effects of pressure, tread depth, and temperature.

2.1.3.3 LuGre Tire Model

The LuGre tire model [26] represents friction through the dynamics of the bristles (Figure 2.3) at the contact patch with punctual tire-road friction contact [25], capturing static and dynamic effects. Then, the average deflection of the bristles is represented by the state variable z, which evolves according to the differential equation 2.10 [26].

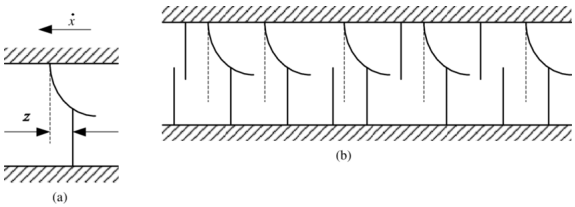


Figure 2.3: Bristle model representation, from [31].

$$\dot{z} = \frac{dz}{dt} = v - \frac{|v|}{g(v)} \cdot z \tag{2.10}$$

Where v is the relative velocity between the tire and the road, and $g(v)$ is a function that describes the behavior of the friction in steady state [25].

The friction force F is then given by equation 2.11 [26].

$$F = \sigma_0 \cdot z + \sigma_1 \cdot \dot{z} + \sigma_2 \cdot v \tag{2.11}$$

Where σ_0 is the stiffness of the normalized longitudinal lumped rubber, σ_1 is the damping of the normalized longitudinal lumped rubber and σ_2 is the normalized relative viscosity damping [25].

In Table 2.1, a summary of the characteristics of the tire models is presented.

Table 2.1: Comparison of tire model categories and representative models.

Model	**Type**	**Advantages**	**Limitations**
Linear Tire Model [27]	Analytical	Simple, computationally efficient, suitable for control design and preliminary vehicle dynamics analysis.	Cannot accurately represent tire saturation or highly nonlinear tire behaviour.
Pacejka (Magic Formula) [21]	Slip/Force Map (Empirical)	High accuracy, widely adopted in industry and motorsport simulations, capable of modelling nonlinear longitudinal and lateral tire forces.	Requires extensive experimental parameter identification; limited physical interpretation of parameters.
LuGre [26]	Lumped Dynamic Friction Model	Captures transient friction phenomena, stick-slip behaviour, Stribeck effect, and dynamic tire-road interaction.	More difficult parameter identification; less commonly used in full-vehicle simulations.

2.1.4 Longitudinal Vehicle Dynamics

Longitudinal vehicle dynamics is the study of a vehicle's movement in the line of its path which considers forces such as traction, braking, and aerodynamic drag. These forces are necessary to know to be able to measure the vehicle's capabilities, use the vehicle's energy in the most efficient way and make sure that the vehicle is safe under different driving conditions. This section presents the principles and the main characteristics of the longitudinal movement.

2.1.4.1 Forces and Resistances

The force balance in the longitudinal direction can be described by Newton's second law, as shown in equation 2.12[1] [4].

[1] The force $mg\sin(\theta)$ was neglected, in opposition to [4], as in formula student, there are no inclination

$$ma_x = F_{x,FL} + F_{x,FR} + F_{x,RL} + F_{x,RR} - F_{drag} - R_{x,FL} - R_{x,FR} - R_{x,RL} - R_{x,RR} \tag{2.12}$$

where

- $F_{x,FL}$ and $F_{x,FR}$ are the longitudinal tire force at the front tires
- $F_{x,RL}$ and $F_{x,RR}$ are the longitudinal tire force at the rear tires
- F_{drag} is the aerodynamic drag force
- $R_{x,FL}$ and $R_{x,FR}$ are the force due to rolling resistance at the front tires
- $R_{x,RL}$ and $R_{x,RR}$ are the force due to rolling resistance at the rear tires

Longitudinal Tire Forces

The longitudinal tire forces are the forces that come from the friction force that comes from the ground and the tires themselves.

As seen in section 2.1.3, the different tire models depend on a set of variables, the most important being the slip variable, which is defined in longitudinal tire force as the slip ratio. The slip ratio can be defined as the difference between the actual longitudinal velocity on the axle of the wheel v_x and the equivalent rotational velocity $R_{\text{wheel}} \cdot \omega_{\text{wheel}}$ and is defined in equation 2.13 [4].

$$\sigma_x = \begin{cases} \frac{R_{\text{wheel}} \cdot \omega_{\text{wheel}} - v_x}{v_x}, \text{during braking} \\ \frac{v_x - R_{\text{wheel}} \cdot \omega_{\text{wheel}}}{R_{\text{wheel}} \cdot \omega_{\text{wheel}}}, \text{during acceleration} \end{cases} \tag{2.13}$$

where

- R_{wheel} represents the wheel radius
- ω_{wheel} represents the wheel angular velocity
- v_x reprents the vehicle's longitudinal velocity

With that information, F_x can be defined with the Pacejka tire model (see section 2.1.3.2) to calculate the value F_x as seen in equation 2.14.

$$F_x(\sigma_x) = D_x \sin[C_x \arctan\{B_x \cdot \sigma_x - E_x(B_x \cdot \sigma_x - \arctan(B_x \cdot \sigma_x))\}] \tag{2.14}$$

Aerodynamic Drag Force

The aerodynamic drag force is a force that acts in the direction opposite to the direction of motion and can be calculated as seen in equation 2.15 [22].

$$F_{drag} = \frac{1}{2}\rho C_d A_F v_x^2 \tag{2.15}$$

where

- ρ is the mass density of air
- C_d is the aerodynamic drag coefficient
- A_F is the frontal area of the vehicle
- v_x is the longitudinal vehicle velocity

Rolling Resistance

As a tire rotates, the contact patch undergoes continuous deformation, with the stiffer road assumed rigid and the elastic tire material deflecting under the normal load. As new material enters the contact patch, it is compressed, reaches maximum deformation at the center, and recovers upon exiting; however, due to internal damping, the energy lost in deformation is not fully recovered. This energy loss gives rise to rolling resistance, a force opposing vehicle motion, and causes the distribution of the normal load within the contact patch to become non-symmetric. Unlike the static case, where the load is evenly distributed, in the rotating tire the viscous dissipation results in higher normal forces in the forward half of the contact patch compared to the rear [4]. In formula 2.16.

$$R_{x,FL} + R_{x,FR} + R_{x,RL} + R_{x,RR} = f(F_{z,FL} + F_{z,FR} + F_{z,RL} + F_{z,RR}) \tag{2.16}$$

where

- $F_{z,FL}$ and $F_{z,FR}$ are the normal forces in the front tires
- $F_{z,RL}$ and $F_{z,RR}$ are the normal force in the rear tires
- f is the rolling resistance coefficient

2.1.5 Lateral Dynamics

Lateral vehicle dynamics is the part of vehicle dynamics that describes the motion of the vehicle when cornering. Cornering comes as the output of the modulus of the velocity and the rotation

around the z axis, the yaw angle (ψ). The yaw rate ($\dot{\psi}$) comes as the rotational velocity around the z axis and can be written as in equation 2.17 [32].

$$\dot{\psi} = \frac{d\psi}{dt} = \frac{v}{R_{\text{corner}}} \tag{2.17}$$

where R_{corner} is the radius of the corner.

2.1.5.1 Tire Lateral Force Generation

The lateral force is calculated based on the tire behaviour [33] and can be expressed as equation 2.18 [21].

$$F_y = D_y \sin\Big[C_y \arctan\Big(B_y\alpha - E_y\big(B_y\alpha - \arctan(B_y\alpha)\big)\Big)\Big] \tag{2.18}$$

where α is the slip angle, and B_y, C_y, D_y, and E_y are the Magic Formula tire model parameters. This nonlinear relationship captures how the lateral force increases with slip angle, reaches a maximum, and then decreases as the tire saturates.

Slip Angle Calculation

The slip angle α is defined as the angle between the direction in which the wheel is pointing and the actual velocity vector of the wheel center (figure 2.4). It is a necessary information in order to create cornering force [11].

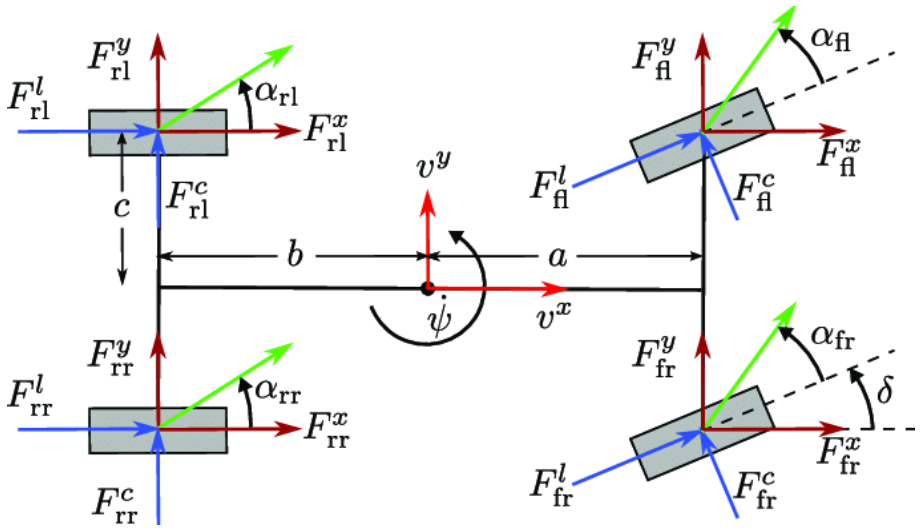


Figure 2.4: Representation of the slip angle α in a 4 wheel vehicle model, from [34].

The vehicle slip angles can be defined as in equations 2.19 [35].

$$
\begin{aligned}
\alpha_{FL} &= \arctan\left(\frac{v_y + \dot{\psi}\cdot l_f}{v_x - \dot{\psi}\cdot t_f}\right) - \delta_l \\
\alpha_{FR} &= \arctan\left(\frac{v_y + \dot{\psi}\cdot l_f}{v_x + \dot{\psi}\cdot t_f}\right) - \delta_r \\
\alpha_{RL} &= \arctan\left(\frac{v_y - \dot{\psi}\cdot l_r}{v_x - \dot{\psi}\cdot t_r}\right) \\
\alpha_{RR} &= \arctan\left(\frac{v_y - \dot{\psi}\cdot l_r}{v_x + \dot{\psi}\cdot t_r}\right)
\end{aligned}
\tag{2.19}
$$

where:

- δ_l and δ_r are the steering angles at the front wheels.
- v_x and v_y are the longitudinal and lateral components of the vehicle velocity at the center of gravity,
- $\dot{\psi}$ is the yaw rate of the vehicle,
- l_f and l_r are the distances from the center of gravity to the front and rear axles, respectively.
- t_f and t_r are the width of the front and rear axle, respectively.

These equations show that the slip angle depends not only on the steering input δ, but also on the lateral motion of the vehicle body (v_y) and the yaw rate $\dot{\psi}$. In practice, accurate slip angle estimation is critical for vehicle stability control systems, as it directly influences the lateral force generation capability of the tires.

Combined Slip Behaviour

The generation of tire forces is limited by the available friction at the tire-road interface. As a consequence, longitudinal and lateral tire forces cannot be considered completely independent. When a tire generates longitudinal force through acceleration or braking, its ability to simultaneously generate lateral force is reduced, and vice versa [21].

This interaction is commonly represented through friction circle or friction ellipse concepts and is referred to as combined slip behaviour [36]. Combined slip effects become particularly relevant in high-performance vehicles and advanced vehicle dynamics control systems, such as traction control and torque vectoring, where longitudinal tire forces are actively manipulated to influence vehicle behaviour.

Yaw Moment and Vehicle Stability

When a vehicle negotiates a curve, unequal lateral tire forces between the front and rear axles generate a yaw moment M_z about the vertical axis of the vehicle. The yaw moment depends on

several factors such as longitudinal force F_x and F_y self-aligning torques $M_{z,FL}$, $M_{z,FR}$, $M_{z,RL}$ and $M_{z,RR}$. Formula 2.20 represents the yaw moment formula considering variables [37].

$$\begin{aligned} M_z = I_{zz}\ddot{\psi} = l_f\Big(&\cos\delta\left(F_{yFL}+F_{yFR}\right)+\sin\delta\left(F_{xFL}+F_{xFR}\right)\Big) \\ &- l_r\left(F_{yRL}+F_{yRR}\right) \\ &+ \frac{t_f}{2}\Big(\cos\delta\left(F_{xFL}-F_{xFR}\right)-\sin\delta\left(F_{yFL}-F_{yFR}\right)\Big) \\ &+ \frac{t_r}{2}\left(F_{xRL}-F_{xRR}\right) \\ &- M_{z,FL}-M_{z,FR}-M_{z,RL}-M_{z,RR} \end{aligned} \tag{2.20}$$

where I_{zz} is the vehicle yaw moment of inertia, $\ddot{\psi}$ is the yaw acceleration, l_f and l_r are the distances from the center of gravity to the front and rear axles, respectively, t_f and t_r are the front and rear track widths, respectively, δ is the front wheel steering angle (assuming Ackermann effects are neglected), F_{xij} and F_{yij} are the longitudinal and lateral tire forces, respectively, acting on wheel ij, and $M_{z,ij}$ is the self-aligning moment generated by tire ij.

If the yaw moment grows uncontrollably, the vehicle can spin or lose control. Vehicle designers carefully tune suspension and tire characteristics to maintain yaw stability over the expected operating range.

As shown in Equation 2.20, the yaw moment depends on the difference between the longitudinal forces F_x. This variable can be manipulated through yaw moment control systems to improve cornering performance.

2.1.5.2 Bicycle Model

The bicycle model, also known as the single-track model, is one of the most widely used representations of vehicle lateral dynamics. It simplifies the vehicle by combining the left and right wheels of each axle into a single equivalent wheel, reducing the complexity of the system while preserving the dominant lateral and yaw dynamics [38].

Under the assumptions of small slip angles, small steering angles, linear tire behaviour, and constant longitudinal velocity, the vehicle motion can be described by two state variables: the vehicle sideslip angle (β) and the yaw rate ($\dot{\psi}$). The resulting equations of motion are obtained by applying Newton's second law in the lateral direction and the rotational equation of motion about the vertical axis passing through the centre of gravity [38].

The bicycle model provides a useful framework for analysing vehicle handling and stability characteristics. Despite its simplicity, it captures the fundamental relationship between steering input, tire forces, sideslip angle, and yaw response, making it one of the most widely used models in vehicle dynamics studies.

$$\begin{cases} mv\left(\frac{d\beta}{dt}+\dot{\psi}\right) = -K_{y,F}\left(\beta+\frac{l_f}{v}\dot{\psi}-\delta\right)-K_{y,R}\left(\beta-\frac{l_r}{v}\dot{\psi}\right) \\ I_{zz}\frac{d\dot{\psi}}{dt} = -K_{y,F}\left(\beta+\frac{l_f}{v}\dot{\psi}-\delta\right)l_f+K_{y,R}\left(\beta-\frac{l_r}{v}\dot{\psi}\right)l_r \end{cases} \tag{2.21}$$

where m is the vehicle mass, v is the longitudinal velocity, β is the vehicle sideslip angle, ψ is the yaw angle, δ is the steering angle, $K_{y,F}$ and $K_{y,R}$ are the front and rear cornering stiffnesses, l_f and l_r are the distances from the centre of gravity to the front and rear axles, respectively, and I_{zz} is the vehicle yaw moment of inertia.

2.2 Mechanical Differentials

Mechanical differentials are drivetrain components responsible for distributing torque between the driven wheels while allowing differences in wheel rotational speed. Their primary function is to accommodate the kinematic requirements of cornering, where the inner and outer wheels must travel different distances and therefore rotate at different speeds. Beyond this basic function, the way torque is distributed between the wheels has a significant influence on vehicle traction, handling, and stability [39].

From a vehicle dynamics perspective, differentials can be interpreted as torque distribution systems. Modern torque vectoring controllers seek to actively manipulate wheel torques to achieve vehicle dynamics objectives such as improved traction and yaw stability. In this sense, torque vectoring can be viewed as an evolution of traditional mechanical differentials, replacing passive mechanical torque distribution with actively controlled torque allocation.

2.2.1 Open Differential

An open differential [40] represents the simplest and most common differential configuration. It distributes equal torque to both driven wheels while allowing unrestricted differences in wheel speed. This behaviour satisfies the kinematic requirements of cornering and provides smooth vehicle operation under normal driving conditions.

The main limitation of the open differential arises when the available traction differs significantly between the driven wheels. Because the transmitted torque is limited by the wheel with the lowest available grip, a loss of traction on one wheel reduces the torque that can be transmitted to both wheels. Consequently, a large portion of the available driving torque may be wasted through wheel spin [40].

Despite these limitations, the open differential serves as an important reference case in vehicle dynamics studies. In the absence of additional control mechanisms, it represents the passive torque distribution behaviour against which torque vectoring systems are commonly compared [41].

2.2.2 Locking Differential

A locking differential represents the opposite extreme of torque distribution. When locked, both wheels are forced to rotate at the same speed regardless of differences in tire grip or cornering conditions. This configuration maximizes traction but compromises handling performance due to the inability of the wheels to satisfy the natural speed differences required during cornering [42].

2.2.3 Limited-Slip Differential (LSD)

A limited-slip differential (LSD) [43] can be considered an intermediate solution between open and locking differentials. Rather than enforcing equal wheel speeds or allowing unrestricted speed differences, an LSD permits a controlled amount of slip while redistributing torque toward the wheel with greater available traction.

From a vehicle dynamics perspective, the LSD introduces a passive form of torque vectoring. By modifying the torque distribution between the driven wheels according to mechanical conditions within the differential, it can influence the yaw behaviour of the vehicle and improve corner-exit performance. However, unlike modern torque vectoring systems, the torque distribution characteristics are determined by the mechanical design of the differential and cannot be actively adapted to changing driving conditions [4].

The M-Diff Formula Student V2 Adjustable [44] is a limited-slip differential designed specifically for Formula Student and FSAE race cars. It is the differential used by the FS FEUP 02 prototype, which was the last prototype to use a single motor with a differential transmission.

In Table 2.2, a summary of the characteristics of the differentials is presented.

Table 2.2: Comparison of differential types and torque vectoring systems.

System	**Wheel Speed Difference**	**Torque Distribution**	**Traction Performance**	**Control Authority**
Open Differential [40]	Free	Equal torque to both wheels	Low on split-μ surfaces	None
Locking Differential [42]	Not allowed when locked	Equal wheel speed	Very high	None
Limited-Slip Differential [43]	Limited	Passive torque redistribution	Improved	Limited

2.3 Traction Control

Traction Control Systems (TCS) regulate wheel slip to maximize tire-road force utilization while maintaining vehicle stability. Similar to Anti-lock Braking Systems (ABS), traction controllers aim to keep the tire operating near the peak of the friction-slip characteristic, thereby maximizing longitudinal performance and preserving vehicle controllability. In high-performance electric vehicles equipped with independently actuated in-wheel motors, traction control benefits from fast torque response and wheel-level actuation, enabling more advanced control strategies than those available in conventional drivetrains [45, 4].

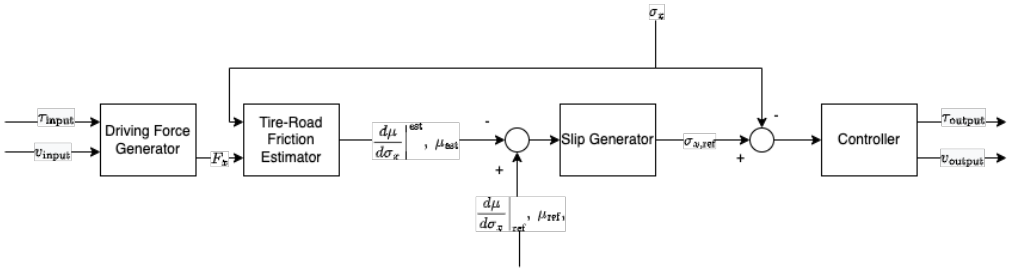


Figure 2.5: Generic architecture of TC/ABS system, adapted from [45].

Figure 2.5 shows an outline of how traction control systems are usually implemented. Typically, they consist of several modules, including a tire-road friction estimator that uses wheel slip and wheel force to estimate the tire-road friction coefficient, μ. This estimated value is compared with a reference value to generate the reference slip, σ_x. The controller is then responsible for calculating the actual wheel torque to apply in order to achieve the desired slip.

Among the various slip-regulation approaches proposed in the literature, adaptive Sliding Mode Control (SMC) remains one of the most robust solutions due to its ability to handle nonlinear tire dynamics, uncertain vehicle parameters, and varying road conditions [46]. Kachroo and Tomizuka proposed an adaptive sliding mode traction controller combined with a sliding-mode parameter estimation scheme capable of estimating the local slope of the tire friction characteristic and updating the target slip ratio toward the maximum-traction region without requiring prior knowledge of the road surface [46]. Simulation results showed satisfactory slip regulation under significant parameter uncertainties and demonstrated that the traction controller remained stable under severe external disturbances that caused passive PID and tractionless sliding-mode controllers to become unstable, highlighting its robustness to model uncertainties and disturbances [46].

More recent research has extended traction control beyond individual wheel-slip regulation toward coordinated force-allocation strategies for four-wheel independently driven electric vehicles. Fujimoto et al. proposed a driving-force distribution framework that combines traction control with optimization-based allocation of wheel forces, allowing available traction to be redistributed among wheels according to local adhesion conditions [47]. By exploiting actuator redundancy, the method maintains higher total longitudinal force on transient low-friction surfaces while simultaneously reducing undesired yaw moments under split-friction conditions. Both simulation and experimental results showed that the proposed approach preserved a larger total driving force than conventional Driving Force Control (DFC) and improved directional stability by suppressing yaw moment generation when individual wheels encountered low-adhesion regions [47].

Another traction control strategy relies on estimating parameters of the tire-road friction characteristic from wheel-slip measurements. Following the approach proposed by Gustafsson [48], the method exploits the approximately linear relationship between friction coefficient and slip at low slip levels. An adaptive estimator is used to identify the local slope of the friction-slip characteristic, $k = d\mu/d\sigma_x$, together with a slip offset parameter. Changes in the estimated slope are then

used as an indicator of variations in available tire-road friction, enabling adaptation to changing surface conditions without requiring dedicated friction sensors or detailed road models.

In [49], road-surface friction is estimated from wheel-speed, wheel-acceleration, and engine-output information during traction events. The estimated friction coefficient is then used to determine desired slip values for engine and brake control. By adapting these slip targets to the current road condition, the controller regulates wheel slip toward a region of high traction while maintaining vehicle stability and drivability under varying surface conditions.

2.4 Torque Vectoring

Torque vectoring is a vehicle control strategy that improves handling, stability, and cornering performance by actively distributing driving torque between individual wheels (Figure 2.6). By creating differential torques, the system generates a controllable yaw moment, allowing precise regulation of the vehicle's rotation about its vertical axis and enhancing responsiveness during maneuvers [50].



(a) Baseline cornering condition, assuming equal torque distribution.

(b) Increased yaw moment through differential longitudinal force application.

Figure 2.6: Illustration of yaw rate enhancement through differential longitudinal force generation. In the baseline/open case (a), the longitudinal tire forces are balanced. In the torque-vectoring case (b), a greater longitudinal force is applied at the outer wheel, producing an additional yaw moment that increases the vehicle yaw rate and improves cornering response. Arrow lengths indicate the relative magnitude of the longitudinal tire forces.

Torque vectoring can be classified under Yaw Stability Control, specifically as a form of Direct Yaw Control (DYC). Yaw Stability Control itself can be further subdivided into several types and subtypes, as illustrated in Figure 2.7 [51].

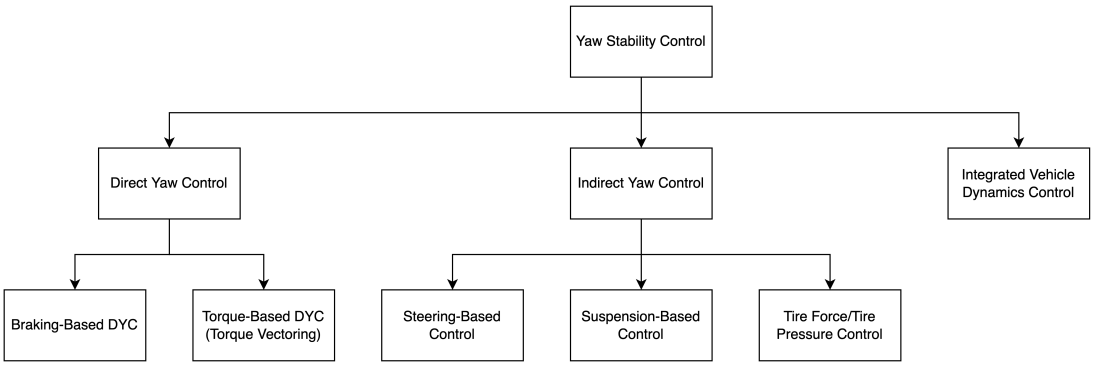


Figure 2.7: Classification of yaw stability control types [51].

In Direct Yaw Control (DYC), the control is achieved through the yaw moment generated by the difference in longitudinal forces between the outer and inner wheels. Brake-based DYC applies this control via selective braking, while torque vectoring accomplishes it through targeted torque distribution.

Indirect Yaw Control modifies vehicle parameters (such as steering, suspension, or tire pressure) to achieve the desired yaw behavior.

Integrated Vehicle Dynamics Control coordinates multiple vehicle subsystems to optimize overall stability and handling, ensuring that the car responds in a balanced and controlled manner under various driving conditions [51].

Torque vectoring architectures (Figure 2.8) can be conceptually divided into several stages. As described in [52], the torque vectoring control scheme is typically composed of three main components:

- **Reference Generator:** Processes the driver inputs and vehicle states (e.g., velocity) to compute the reference values required by the controller, such as the desired yaw rate ($\dot{\psi}$).
- **High-Level Controller:** Uses the reference values and current vehicle states to determine the required yaw moment (M_z) to be applied to the vehicle, as well as the total driving torque (τ_T).
- **Low-Level Controller:** Converts the total driving torque (τ_T) and yaw moment demand (M_z) into individual wheel torque commands.

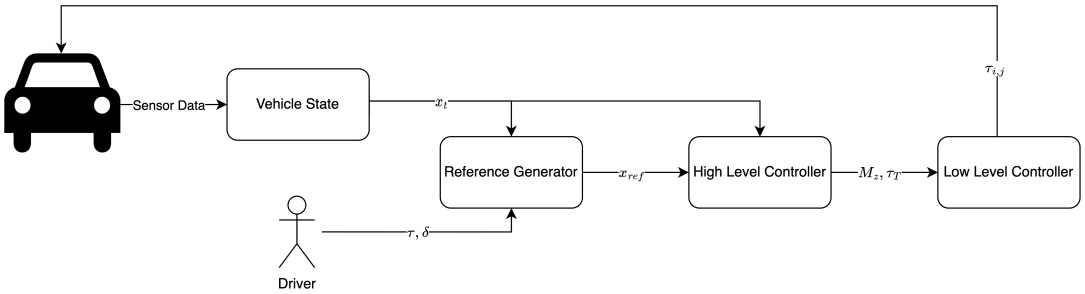


Figure 2.8: General torque vectoring control architecture for yaw moment control.

Table 2.3 summarizes all the information and the outputs of each part of any torque vectoring system [50].

Table 2.3: Torque Vectoring Control Architecture.

Reference Generator	**High Level Controller**	**Low Level Controller**
Sideslip angle (β) Yaw rate ($\dot{\psi}$)	Yaw Moment (M_z) Total Torque (T_T)	Torque Distribution

2.4.1 Reference Generator

As said, the reference generator is responsible for creating the set of reference variables that represent the state of our vehicle that we want to reach. Usually, the variables that represent the reference generator are the sideslip angle (β) and/or the yaw rate ($\dot{\psi}$) [50].

2.4.1.1 Yaw Rate Reference Generation

In [50], two main approaches are described for the yaw rate ($\dot{\psi}$) control. The first one, results from a linearization of the steady-state ($\ddot{\psi} = 0$) equations of a single-track vehicle model [8, 13, 53]. Figure 2.22 describes the resulting formula of the simplification.

$$\dot{\psi}_{\text{ref}} = \frac{v}{L(1 + K_u v^2)} \delta \tag{2.22}$$

where $\dot{\psi}_{\text{ref}}$ is the reference yaw rate, v is the vehicle longitudinal velocity, L is the wheelbase, and δ is the steering angle.

In this formula, K_u is defined as the understeering coefficient (or Stability factor) that models the vehicle's cornering response and can be defined as in equation 2.23 [54, 53].

$$K_u = \frac{m}{L^2} \left(\frac{l_r}{K_{y,F}} - \frac{l_f}{K_{y,R}} \right) \tag{2.23}$$

where m is the vehicle mass, L is the wheelbase, l_f and l_r are the distances from the center of gravity to the front and rear axles, respectively, and $K_{y,F}$ and $K_{y,R}$ are the front and rear axle cornering stiffnesses, respectively.

Several equivalent formulations exist for computing the yaw-rate reference, all of which lead to the same result, as discussed in [55, 11]. Even thought formula 2.23 allows us to calculate anallitically with the parameters of our vehicle, this value can be used to manipulate the vehcile's cornering response. Authors such as [8] and [13] also limits the yaw rate with the maximum lateral acceleration of the vehicle.

$$|\dot{\psi}_{\max}| = \left| \frac{a_{y,\max}}{v_x} \right| \tag{2.24}$$

$$\dot{\psi}_{\text{ref}} = \begin{cases} \dot{\psi}_{\text{des}}, & \text{if } |\dot{\psi}_{\text{des}}| < |\dot{\psi}_{\max}| \\ \dot{\psi}_{\text{des}}, & \text{otherwise} \end{cases} \tag{2.25}$$

Another way to calculate the reference yaw rate ($\dot{\psi}$) is presented in [50]:

$$\dot{\psi}_{\text{ref}} = \begin{cases} \dfrac{v}{L(1+K_u v^2)}\,\delta = \alpha\,\delta, & \text{if } \delta \leq \delta^* \\ \dot{\psi}_{\max} + (\dot{\psi}^* - \dot{\psi}_{\max}) e^{\left(\frac{-\alpha(\delta-\delta^*)}{\dot{\psi}_{\max}-\dot{\psi}^*}\right)}, & \text{if } \delta > \delta^* \end{cases} \tag{2.26}$$

where $\dot{\psi}^*$ represents the yaw rate value that defines the boundary between the linear and saturation regions, and $\delta^* = K_u a_y$ is the corresponding steering angle at the transition point.

2.4.1.2 Sideslip Angle Reference Generation

For the reference sideslip angle, [50] presents a formulation that uses the previously calculated reference yaw rate ($\dot{\psi}$ref) to determine the target sideslip angle (βref), which is then used by the high-level controller to compute the desired yaw moment (M_z):

$$\beta_{\text{ref}} = \left(\frac{l_r}{v} - \frac{m \cdot l_f \cdot v}{L \cdot K_{y,R}} \right) \dot{\psi}_{\text{ref}} \tag{2.27}$$

where β_{ref} is the reference sideslip angle, v is the vehicle longitudinal velocity, m is the vehicle mass, L is the wheelbase, l_f and l_r are the distances from the center of gravity to the front and rear axles, respectively, $K_{y,R}$ is the rear axle cornering stiffness, and $\dot{\psi}_{\text{ref}}$ is the reference yaw rate.

Another approach, proposed in [10], uses the current sideslip angle (β) and the maximum sideslip angle ($\beta_{\max}$) to generate the next reference value.

$$\beta_{\text{ref}} = \beta_{\max} \tanh\left(\frac{\beta}{\beta_{\max}} \right) \tag{2.28}$$

2.4.2 High Level Controller

The high-level controller is the part of the Torque Vectoring algorithm that is responsible for creating the vehicle dynamics inputs required by the vehicle in order to convert its current state (mainly sideslip angle (β) and/or yaw rate ($\dot{\psi}$)) into the target values generated by the Reference Generator.

In [50], the different controllers are divided into different cattegories: PID Controllers, Optimal Controllers (such as LQR, LQV and LPV), Sliding Mode Controllers, Model Predictive Controllers and Fuzzy Controllers.

2.4.2.1 PID Controllers

In the Torque Vectoring context, the Proportional-Integral-Derivative (PID) controller is used as a feedback controller for yaw rate ($\dot{\psi}$) or sideslip angle (β) control. The main advantage of using a PID controller is the reduced number of parameters to tune in order to achieve optimal performance.

The PID control law can be expressed as [56]:

$$M_z = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt} \tag{2.29}$$

In the work of Antunes et al. [11], a torque vectoring control architecture based on a PI algorithm is proposed for yaw rate control. As seen in Figure 2.9, the controller uses a lookup table to determine the proportional (k_P) and integral (k_I) gains as functions of the vehicle's longitudinal velocity (v_{CG}), allowing the controller to adapt its response to different operating conditions. The desired yaw rate ($\dot{\psi}_{ref}$) is generated from the steering input (δ) and vehicle velocity, and is then compared with the measured yaw rate ($\dot{\psi}$) to compute the yaw rate error ($\dot{\psi}_{error}$). This error serves as the input to the PI controller, which outputs a torque difference command (ΔT) to achieve the desired yaw behavior. A linear bicycle model is used to estimate the actual yaw rate of the vehicle, providing feedback for the control loop and enabling accurate tracking of the reference yaw rate across varying dynamic conditions.

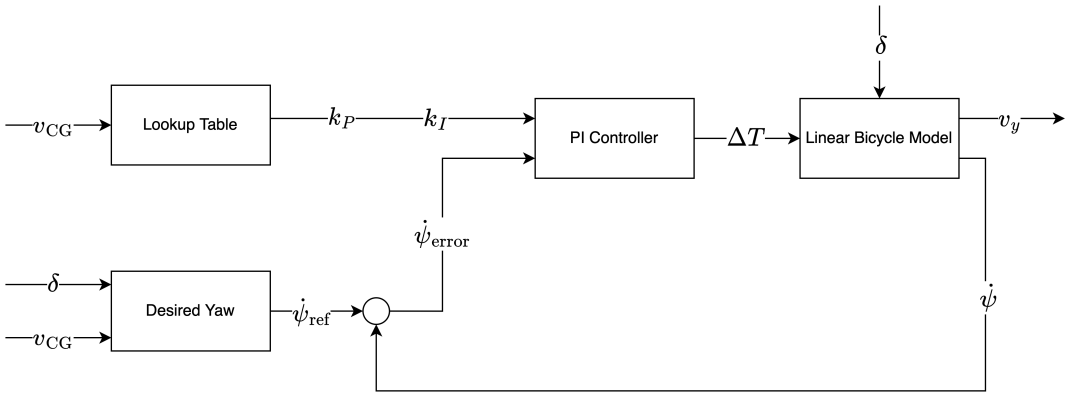


Figure 2.9: Torque Vectoring Architecture with a PI Controller (FST Lisboa) [11].

Unlike other torque vectoring control strategies that distinguish between high-level and low-level control layers, the approach by Antunes et al. [11] does not explicitly separate these stages. Instead, the controller directly computes the torque difference (ΔT) as a control function of the yaw rate error ($\dot{\psi}_{error}$). This simplifies the overall control architecture by eliminating the need for

a dedicated torque allocation or distribution layer, while still ensuring that the generated torque difference effectively compensates for deviations between the desired and actual yaw rates.

The implementation of torque vectoring was shown to enhance vehicle performance, achieving an average increase of 5% performance gain in skidpad lap times in a Formula Student event [12].

In the torque vectoring control strategy proposed by Fujimoto et al. [16], the control architecture combines a reference generator, a PI controller, and a low-level torque distribution stage to regulate the vehicle's yaw dynamics, as seen in Figure 2.10. The reference generator computes the desired yaw rate (ψ_r) based on the driver's steering input (δ), which is then compared with the measured yaw rate (ψ) to obtain the yaw rate error (e_ψ). This error is processed by a PI controller to generate a desired corrective yaw moment ($M_{z,PI}$). Unlike conventional PI-based schemes, Fujimoto introduces an additional compensating term that accounts for the actual yaw moment, derived from the product of the vehicle's yaw rate (ψ) and yaw moment of inertia (I_{zz}). This term ensures that the total commanded yaw moment (M_z) reflects both the controller's corrective action and the dynamic contribution of the vehicle's rotational inertia. The resulting yaw moment is then processed by a low-level controller, which translates the desired total torque (T_{TOT}) and the required yaw moment into individual wheel torque commands. This hierarchical control approach enhances yaw stability and steering responsiveness by providing accurate moment generation while compensating for vehicle inertia effects.

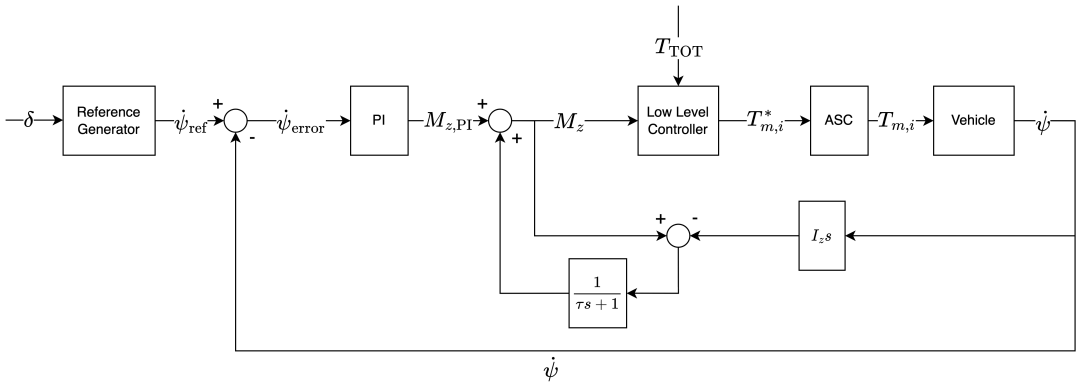


Figure 2.10: Torque Vectoring control scheme of Fujimoto et al. [16].

Experimental validation demonstrated the effectiveness of the controller in mitigating yaw-rate disturbances under varying road conditions. On dry asphalt, the controller reduced the yaw-rate response by approximately 33% compared to the uncontrolled case, while also decreasing the corrective steering effort required from the driver following the disturbance. Similar benefits were observed on snowy surfaces, where the yaw-rate response was reduced by 25% [16].

The torque vectoring control strategy presented in [13] (Figure 2.11) uses a control architecture centered around a reference generator, which determines the desired yaw rate (ψ_{ref}) from the driver's steering input (δ). The difference between this reference yaw rate and the actual measured yaw rate (ψ) computes the yaw rate error (ψ_{error}), which is processed by a PI controller to

produce a preliminary corrective yaw moment ($M_{z,pre}$). To improve dynamic response and compensate for the moments generated by tire lateral forces, a feed-forward term ($M_{z,Fy}$) proportional to the steering input (δ) is subtracted from this preliminary moment. The resulting total corrective moment ($M_{z,TV}$) is then converted into a corrective yaw rate ($\dot{\psi}_{z,TV}$) using the gain $\frac{\dot{\psi}}{M_{z,TV}}$. This corrective yaw rate is subsequently combined with the direct yaw rate contribution from steering, obtained through the gain $\frac{\dot{\psi}}{\delta}$, before being applied to the vehicle model to generate the actual yaw rate ($\dot{\psi}$), thus closing the feedback loop. This control scheme enhances vehicle yaw stability and agility by integrating closed-loop feedback for accurate yaw tracking with a feed-forward path that anticipates steering-induced yaw dynamics.

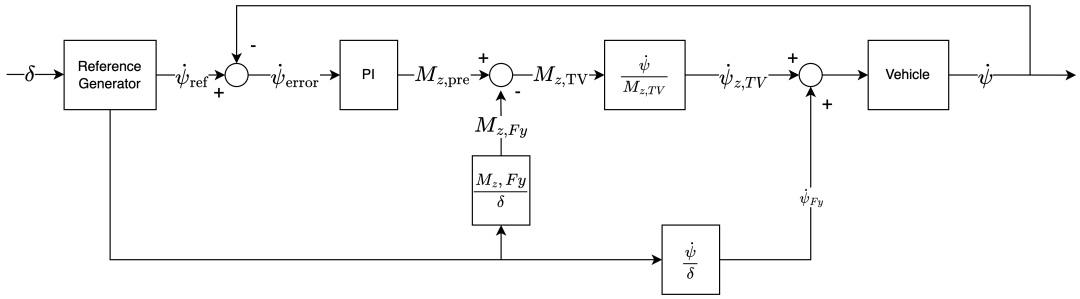


Figure 2.11: Torque Vectoring control scheme (ETSEIB MotorSport) [13].

For this implementation, the results obtained for the Skidpad event [12], are reported in Table 2.4, comparing vehicle performance with torque vectoring enabled and disabled.

Table 2.4: Comparison of vehicle performance with torque vectoring (TV) enabled and disabled [13].

Variable	**TV Active**	**TV Disabled**
Peak yaw rate	1.43	1.3
Average yaw rate	1.29	1.19
Left turn time (s)	4.84	5.24
Right turn time (s)	4.85	5.2
Total time (s)	4.845	5.22

2.4.2.2 Optimal Controllers

Optimal controllers are the controllers that can be considered as model-based. In specific, the model correspondes to a virtual vehicle model of the system that is able to predict how the vehicle will handle the inputs the algorithm is giving to the vehicle. In other words, how the yaw moment that is being generated by the high-level controller will behave on the vehicle itself, to predict the change in the yaw rate, for instance.

In Mangia et al. [10], a Linear Quadratic Regulator (LQR) is employed to compute the corrective yaw moment M_z. The vehicle's lateral dynamics are expressed in state-space form with

sideslip angle β and yaw rate $\dot{\psi}$ as states, and the controller minimizes a quadratic cost function (Equation 2.30) that penalizes both tracking error and control effort. The weighting matrices Q and R are normalized by the maximum allowable values of β, $\dot{\psi}$, and M_z, providing an intuitive tuning approach. The main advantage of LQR in this context is its ability to simultaneously regulate both states optimally, at the cost of depending on a linearised bicycle model that may degrade at high lateral accelerations.

$$J = \int_0^\infty \left(e^{\mathrm{T}} Q\, e + M_{z,\mathrm{SSC}}\, R\, M_{z,\mathrm{SSC}} \right) dt, \tag{2.30}$$

Experimental results showed that the proposed torque-vectoring controller was capable of reducing vehicle power losses while maintaining handling performance comparable to the baseline vehicle. In the Normal driving mode, power losses were reduced by up to 4.28% at high lateral accelerations, demonstrating that the LQR-based yaw moment control can improve vehicle efficiency without significantly compromising dynamic behaviour. These results highlight the potential of optimal control approaches to balance vehicle stability objectives with energy consumption considerations in electric vehicles [10].

In [11], an LQR controller is proposed using lateral velocity v_y and yaw rate $\dot{\psi}$ as state variables, with the torque difference ΔT as control input (Equation 2.31). Since v_y cannot be directly measured, it is estimated via a Kalman Filter or approximated from longitudinal velocity under steady-state conditions. Compared to a PI controller, which can become unstable under low friction or sliding conditions, the LQR provides a more robust solution by minimizing a cost function that balances yaw rate tracking error and control effort, resulting in near-neutral steer behaviour and improved stability.

$$\Delta T = K_R \dot{\psi} + K_v v_y \tag{2.31}$$

In [57], a gain-scheduled Robust Linear Quadratic Regulator (RLQR) is proposed for direct yaw moment control of a fully electric vehicle with individually controlled motors. The controller extends a conventional gain-scheduled LQR by incorporating three control terms: the LQR feedback action (u_{LQ}), the feedforward action (u_{FF}) based on the reference trajectory, and an additional robust feedback term (u_{RB}) to enhance robustness against unmodelled dynamics and parameter uncertainties. All control gains are scheduled as functions of the longitudinal velocity.

The results obtained from the skidpad tests, comparing the passive vehicle dynamics with the LQR and robust LQR (RLQR) controllers, show a reduction in the RMSE of the yaw rate tracking error of 44% and 66% for the LQR and RLQR controllers, respectively, relative to the baseline vehicle [57].

2.4.2.3 Model Predictive Control

Model Predictive Control (MPC) has gained increasing attention in torque vectoring applications due to its ability to handle multi-input multi-output systems with constraints. MPC uses an internal predictive model to forecast the vehicle's future behavior over a defined prediction horizon,

optimizing a cost function that balances yaw rate tracking accuracy and control effort, subject to actuator and stability constraints.

In [58], the authors propose a Model Predictive Control (MPC)-based torque vectoring strategy aimed at improving cornering performance by exploiting the natural understeer characteristics of the vehicle. Unlike conventional yaw-rate tracking controllers, the proposed approach does not require real-time road friction estimation or the computation of a friction-dependent reference yaw rate. Instead, the controller regulates the difference between the front and rear wheel slip angles, maintaining the vehicle close to a neutral steer condition while preserving stability.

The controller is based on a linear bicycle model describing the vehicle's lateral dynamics, where an additional yaw moment generated through torque vectoring acts as the control input. Using predictions of the vehicle states over a finite horizon, the MPC optimizes the corrective yaw moment while respecting constraints on tire slip angles and actuator limits. By directly controlling the vehicle's understeer behaviour, the strategy seeks to maximise cornering performance without approaching unstable operating conditions.

Equation 2.32 represents the state vector x(t), output vector y(t), and control input u(t), respectively.

$$\begin{aligned} x(t) &= \begin{bmatrix} \alpha_f - \alpha_r \\ \alpha_r \end{bmatrix}, \\ y(t) &= \begin{bmatrix} \dot{\psi} \\ a_y \end{bmatrix}, \\ u(t) &= M_z \end{aligned} \tag{2.32}$$

A major advantage of this approach is its reduced dependence on accurate tire-road friction estimation, which is often a significant source of uncertainty in vehicle dynamics controllers. Furthermore, the predictive nature of MPC allows vehicle stability and performance constraints to be handled explicitly within the optimization process. Experimental and simulation results demonstrated improved lateral stability and cornering performance, with the controller successfully maintaining the desired understeer margin while operating near the vehicle's handling limits, showing an improvement of between 7% and 9% in the maximum lateral acceleration reached [58].

Mikuláš et al. [59] proposed a torque vectoring control strategy for a Formula Student electric racing car equipped with four independently driven wheels. The study compared two advanced control approaches: a Linear Time-Varying Model Predictive Controller (LTV-MPC) and a Nonlinear Model Predictive Controller (NMPC). A nonlinear two-track vehicle model combined with a Pacejka tire model was used to represent vehicle dynamics, while an Unscented Kalman Filter (UKF) was employed for state estimation.

Equation 2.33 represents the state vector and the control input of the MPC. This implementation has the particularity of directly controlling the wheel torque as the control input instead of the yaw moment, thereby integrating the high-level and low-level controllers.

$$\begin{aligned} x(t) &= [v_x, v_y, \dot{\psi}] \\ u(t) &= [T_{FR}, T_{FL}, T_{RR}, T_{RL}] \end{aligned} \tag{2.33}$$

The controllers were evaluated through simulation scenarios, including high-speed U-turns and steering step response tests. Their performance was compared with a conventional equal-torque distribution strategy. The results demonstrated that both MPC-based torque vectoring approaches significantly improved vehicle stability, prevented spin-out situations, and enabled the vehicle to negotiate tighter cornering radii at high speeds [59].

2.4.2.4 Deep Reinforcement Learning

Another approach that is emerging for the high-level controller is the deep reinforcement learning technique. Reinforcement learning is a machine learning technique widely used in robotic systems and sufficiently complex systems to create a black-box model that maps the environment and its variables into a set of actions, which can be either discrete or continuous. This process is described as a Markov Decision Process (MDP).

In [60], the author formally defines the MDP as a five-tuple (S, A, P, R, λ), where S represents the state space of the vehicle (e.g., yaw rate ($\dot{\psi}$), sideslip angle β, etc.). A is the action space (in the scope of the high-level controller, the yaw moment M_z). P is the probability density function that defines the mapping $P : S \times A \times S \rightarrow [0, 1]$, which depicts the likelihood of transitioning from a current state in the state space (S) to a subsequent state after the agent executes a specific action (A). R is the reward function $R : S \times A \rightarrow \mathbb{R}$ that, in this case, is composed of the immediate reward for reference tracking and a penalty term targeting state deviation, control effort, and potential loss of vehicle control. λ denotes the discount coefficient for weighting future rewards.

$$s_t = \left(\beta, \dot{\psi}_{\text{ref}} - \dot{\psi}, \int \beta, \int (\dot{\psi}_{\text{ref}} - \dot{\psi}), \dot{\psi}\right) \tag{2.34}$$

By adopting a Deep Deterministic Policy Gradient (DDPG) algorithm within an actor-critic architecture, this strategy approximates complex, nonlinear vehicle dynamics without the need for an empirical tire force model. The reward function (R) is specifically engineered to prioritize active safety, combining immediate rewards for precise tracking of the sideslip angle and yaw rate with penalty terms that discourage excessive control effort or loss of vehicle control. Numerical simulations conducted on the Carsim/Simulink platform demonstrate that this approach significantly enhances lateral stability and path-tracking accuracy on slippery roads, effectively outperforming traditional linear control methods such as LQR and MPC [60].

Another approach is presented in [61], where the authors implement a TD3-based reinforcement learning controller to compute the yaw moment command using a model-free approach. The TD3 agent takes as input a set of measured and derived vehicle states (Equation 2.35). Based on these observations, the agent outputs a continuous yaw moment command. The learning process

is formulated as a Markov decision process, where the control policy is optimized through interaction with a nonlinear vehicle model, without requiring prior knowledge of vehicle dynamics or tire characteristics.

$$s_t = \left(\dot{\psi}_e, \int \dot{\psi}_e, \beta, \beta_e, v_x, v_e, \omega_{ij}, \sigma_{x,ij} \right) \tag{2.35}$$

To improve learning stability and robustness, the implementation adopts the core features of the TD3 algorithm, including twin critic networks, delayed actor updates to stabilize policy learning, and target policy smoothing to prevent abrupt changes in the control action. In addition, the authors integrate a curriculum learning strategy during training, in which the agent is gradually exposed to increasing levels of task complexity by varying steering input, vehicle speed, and road friction conditions. The trained TD3 policy outputs a single yaw moment command, which is subsequently passed to a low-level torque allocator for wheel torque distribution. This implementation enables adaptive yaw control under varying operating conditions while maintaining stable and smooth torque vectoring behavior [61].

Table 2.5 summarizes the main high-level control approaches used in torque vectoring systems. PID-based controllers remain attractive for Formula Student applications due to their simplicity and low computational requirements. More advanced approaches, such as LQR and MPC, generally provide superior yaw-rate tracking and stability performance at the expense of increased modelling requirements and computational complexity. Consequently, the selection of a high-level controller depends on the trade-off between performance, robustness, implementation complexity, and available computational resources.

Considering the trade-off between control performance, implementation complexity, and computational requirements, a PID-based controller was selected for this work, following the approach proposed by Fujimoto et al. [16].

2.4.3 Low Level Controller

The Low-Level Controller in a torque vectoring system is responsible for distributing torque to the individual wheels, ensuring that the desired outputs from the High-Level Controller, specifically the yaw moment (M_z) and total torque (τ_{TOT}), are achieved.

Equation 2.36 relates the torque differential required between the left and right driven wheels to the desired yaw moment [10]:

$$\Delta T_{\text{LR}} = \frac{M_z \cdot R_{\text{wheel}}}{t_r} \tag{2.36}$$

In this expression, M_z is the additional yaw moment to be generated, t_r is the full rear track (distance between the two driven wheels), and R_{wheel} is the wheel radius. By commanding a torque difference ΔT according to this relation, the vehicle can produce the target yaw moment through longitudinal force asymmetry, thus improving cornering agility and lap-time performance without relying solely on lateral force modulation.

Table 2.5: Comparison of high-level torque vectoring controllers.

Controller	Principle	Advantages	Disadvantages
PID / PI [11, 16, 13]	Uses yaw-rate or sideslip error feedback to generate a corrective yaw moment.	Simple implementation; low computational cost; easy tuning; suitable for real-time embedded systems; widely adopted in Formula Student applications.	Performance degrades under highly nonlinear vehicle dynamics; limited robustness to parameter variations and changing road conditions.
LQR [10, 11]	Minimizes a quadratic cost function that balances tracking performance and control effort.	Excellent yaw-rate tracking; systematic design methodology; good stability properties; low online computational requirements.	Requires an accurate vehicle model; performance may deteriorate under significant model uncertainties and nonlinear operating conditions.
RLQR [57]	Extends the LQR framework by incorporating robustness against parameter uncertainties and model variations.	Improved robustness to modelling errors and changing operating conditions; maintains good tracking performance; enhanced stability margins.	More complex design and tuning process; potentially higher control effort compared to standard LQR.
MPC [58, 59]	Predicts future vehicle behaviour over a finite horizon and optimizes control actions while considering system constraints.	Handles actuator and vehicle constraints explicitly; excellent tracking performance; suitable for multivariable control problems.	High computational complexity; requires an accurate prediction model; more difficult to implement and tune.
Reinforcement Learning [60, 61]	Learns a control policy through interaction with the environment by maximizing a cumulative reward function.	Can handle highly nonlinear vehicle dynamics; adapts to complex driving scenarios; does not require an explicit analytical vehicle model; capable of improving performance through training.	Requires large amounts of training data; high computational cost during training; difficult to guarantee stability and safety; limited interpretability compared to model-based controllers.

In [10], a torque distribution strategy for four-wheel drive vehicles is proposed, which can be directly applied in low-level torque vectoring control. The total driving torque T_{tot} is split between the left and right wheels on each axle, while allowing an adjustable torque bias ΔT_{LR} between the left and right sides to improve handling and stability. The resulting wheel torques are computed as:

$$\begin{cases} T_{11} = \left(\dfrac{T_{\text{tot}}}{2} + \Delta T_{LR}\right)\sigma_L, \\ T_{21} = \left(\dfrac{T_{\text{tot}}}{2} + \Delta T_{LR}\right)(1-\sigma_L), \\ T_{12} = \left(\dfrac{T_{\text{tot}}}{2} - \Delta T_{LR}\right)\sigma_R, \\ T_{22} = \left(\dfrac{T_{\text{tot}}}{2} - \Delta T_{LR}\right)(1-\sigma_R), \end{cases} \tag{2.37}$$

where:

- T_{ij} denotes the torque applied to wheel i on axle j (1 = front, 2 = rear, left/right according to σ_L and σ_R),
- σ_L and σ_R are the left-right torque split coefficients for the front and rear axles, respectively,
- ΔT_{LR} represents the left-right torque bias applied to enhance yaw response or traction.

This formulation allows a low-level controller to adjust individual wheel torques dynamically based on vehicle states (e.g., slip, steering angle, or yaw rate), making it suitable for torque vectoring applications in high-performance or stability-focused control strategies.

To explicitly define the torque allocation for a rear-wheel-drive configuration, the total driving torque is assumed to act only on the rear axle. In this case, the desired yaw moment is generated by imposing a torque difference between the rear left and rear right wheels, while maintaining the total rear axle torque equal to τ_{TOT}.

The rear wheel torques are therefore computed as in equation 2.38:

$$\begin{cases} T_{\text{RL}} = \dfrac{T_{\text{TOT}}}{2} - \Delta T_{\text{LR}}, \\ T_{\text{RR}} = \dfrac{T_{\text{TOT}}}{2} + \Delta T_{\text{LR}}, \end{cases} \tag{2.38}$$

This formulation guarantees that the total longitudinal force demand is preserved while enabling precise yaw moment control through differential torque actuation. As a result, the rear axle becomes the primary actuator for yaw control, making this approach particularly suitable for rear-wheel-drive performance vehicles and simplified torque vectoring implementations.

2.4.3.1 Deep Reinforcement Learning

Similarly to the High-Level Controller, some approaches were created in order to introduce reinforcement learning techniques for torque allocation to ensure the right yaw moment is applied.

For instance in [62], a reinforcement learning approach with a Deep Deterministic Policy Gradient (DDPG) agent (to account for nonlinear interactions within the all-wheel-drive system) is utilized for the low-level controller with the state variables defined as:

$$S_t = \left(\dot{\psi}, \dot{\psi}_e, \int \dot{\psi}, \dot{\beta}, \dot{\beta}_e, \int \dot{\beta}_e, v_x, v_e, \omega_{ij}, \sigma_{x,ij}, F_x, M_z\right) \tag{2.39}$$

The actions are defined as the individual torque values for the front and rear axles:

$$a_t = (T_{FL}, T_{FR}, T_{RL}, T_{RR}) \tag{2.40}$$

The reward function for this low-level controller consists of five primary components: a positive reward is granted when the yaw rate error is kept within an extremely strict threshold, while separate penalty terms target squared errors in the yaw rate, sideslip angle, and longitudinal velocity. Additionally, the function includes a penalty for excessive control effort, which prevents sudden or extreme torque fluctuations that could destabilize the vehicle [62].

By outputting these continuous torque values, the agent learns to optimally split the required yaw moment among the four wheels through trial and error. This allocation enables the vehicle to dynamically adapt to varying road conditions, such as slippery or snowy surfaces, thereby improving lateral stability and maneuverability.

The reward function is also useful for imposing constraints on the system in order to achieve the desired outcome. In [63], Deng *et al.* propose a REDQ-based reinforcement learning torque vectoring algorithm with a reward function that accounts for vehicle and tire stability, drive system power consumption, and driver load, thus incorporating energy-saving considerations. In [64], de Carvalho Pinheiro and Carello present a reward function that uses the cumulative sideslip angle, yaw rate penalty factor, battery state of charge variation (ΔSOC), battery temperature variation and rotor and stator temperature variation.

Another approach is presented by Wei *et al.* [65], where a torque-allocation DRL framework is employed to configure both the high-level and low-level controllers. The state space consists of reference signals generated by the reference generator (including their integrals and tracking errors), as well as the vehicle state vector, such as yaw rate, sideslip angle, and longitudinal velocity. By embedding vehicle stability and energy efficiency objectives into the reward function, the framework enables coordinated torque distribution without relying on an explicit tire or vehicle model.

Table 2.6 summarizes the main low-level torque allocation strategies identified in the literature.

Table 2.6: Comparison of low-level torque allocation controllers.

Controller	**Principle**	**Advantages**	**Disadvantages**
Rule-Based Allocation [10]	Distributes wheel torques using predefined equations, lookup tables, or heuristic rules based on the desired yaw moment and total torque demand.	Simple implementation; low computational cost; easy to validate and interpret; suitable for real-time embedded systems.	Limited adaptability to changing vehicle dynamics and road conditions; performance depends on manual tuning; may not achieve optimal torque distribution.
Reinforcement Learning (RL) [62, 63, 65]	Learns a torque allocation policy through interaction with the environment by maximizing a reward function related to vehicle performance and stability.	Can adapt to complex and nonlinear vehicle dynamics; capable of learning near-optimal allocation strategies; does not require explicit analytical allocation rules.	Requires extensive training data and computational resources; difficult to guarantee stability and safety; limited interpretability and validation compared to rule-based approaches.

Chapter 3

Torque Vectoring Implementation

This work proposes a torque vectoring and traction control architecture for a Formula Student electric vehicle equipped with two independent rear-wheel motors. By exploiting the over-actuated nature of the drivetrain, the proposed system directly influences the vehicle yaw dynamics through differential torque control, eliminating the need for a mechanical differential. In addition, simplified traction control model are integrated into the architecture to enhance vehicle stability and improve the reliability and robustness of the obtained results under varying driving conditions.

3.1 Architecture

The first step in the architecture design of the torque vectoring (TV) system is the definition of the system requirements and the interfaces with which the algorithm must communicate. Figure 3.1 illustrates these interfaces and the overall communication architecture.

The TV system has two main requirements regarding the environment in which it operates. First, it must be capable of interfacing seamlessly with the real vehicle hardware, represented in blue in Figure 3.1. This includes communication with the Autonomous System, which provides the autonomous throttle command and the vehicle state information, as well as communication through the CAN bus, which supplies the driver input signals and other vehicle data.

Second, the TV system must also support operation within a simulation environment, represented in red. As this work focuses heavily on development and validation through simulation, integration with IPG CarMaker [17] is essential. The simulation environment enables the validation of modifications and to evaluate controller performance while also providing access to variables that are difficult or impossible to measure directly on the real vehicle, such as tire forces.

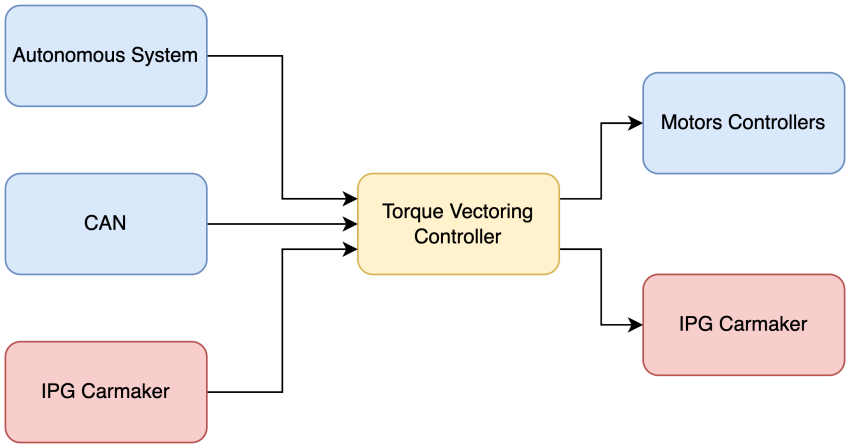


Figure 3.1: Torque vectoring system communication interfaces and architecture.

Figure 3.2 represents how the torque vectoring algorithm will be deployed into the vehicle ECU in the future, when it is implemented in the FS FEUP 03 prototype, which will be built in 2027. The required vehicle states will be obtained from the existing state estimation system developed by the team (Unscented Kalman Filter with the state vector of equation 3.1), which fuses measurements from the IMU, motor resolvers, and wheel speed sensors. Based on these estimated states, the ECU will calculate the appropriate torque commands for the rear motors.

$$s = (v_x, v_y, \dot{\psi}, a_x, a_y, \delta, \omega_{FL}, \omega_{FR}, \omega_{RL}, \omega_{RR}) \tag{3.1}$$

where v_x and v_y are the longitudinal and lateral vehicle velocities, $\dot{\psi}$ is the yaw rate, a_x and a_y are the longitudinal and lateral accelerations, δ is the steering angle, and ω_{FL}, ω_{FR}, ω_{RL}, and ω_{RR} represent the angular velocities of the wheels.

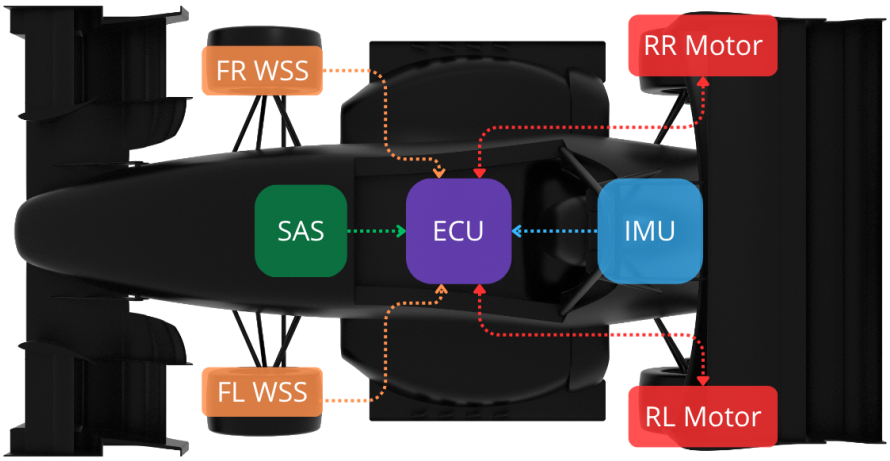


Figure 3.2: Vehicle interfaces for the torque vectoring implementation. The ECU receives vehicle state information from the state estimator, which processes data from the IMU, wheel speed sensors, and motor resolvers. The ECU then computes the torque commands sent to the rear motors.

Going deeper into the implementation of the simulation interfaces, one of the main requirements of the developed system is that the torque vectoring algorithm can later be deployed on a real Formula Student vehicle. Since the autonomous system of the vehicle already relies on ROS2 [66] for communication and state estimation, ROS2 was also selected as the middleware for the development of the torque vectoring controller. This choice simplifies the future integration of the controller with the real vehicle architecture while also providing a modular and scalable communication framework.

To enable communication between ROS and the simulation environment, an intermediate interface was developed. Figure 3.3 illustrates the communication architecture between the different software components and the simulator.

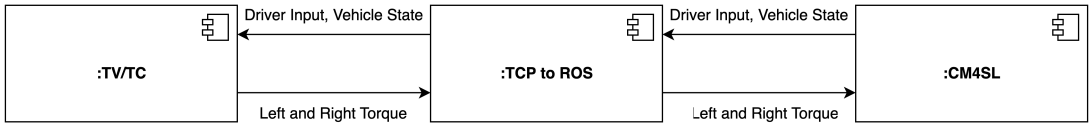


Figure 3.3: UML component diagram of the torque vectoring communication architecture and its interface with the IPG CarMaker simulation environment.

The CM4SL (CarMaker for Simulink [67]) environment exchanges data through TCP sockets, as seen in Figure 3.4. The intermediate interface, named TCP to ROS, receives the information transmitted by CM4SL and converts it into ROS topics containing the relevant simulation data, such as the vehicle state and driver inputs. The torque vectoring and traction control module subscribes to these topics, performs the required computations, and publishes the computed wheel torque commands to additional ROS topics.

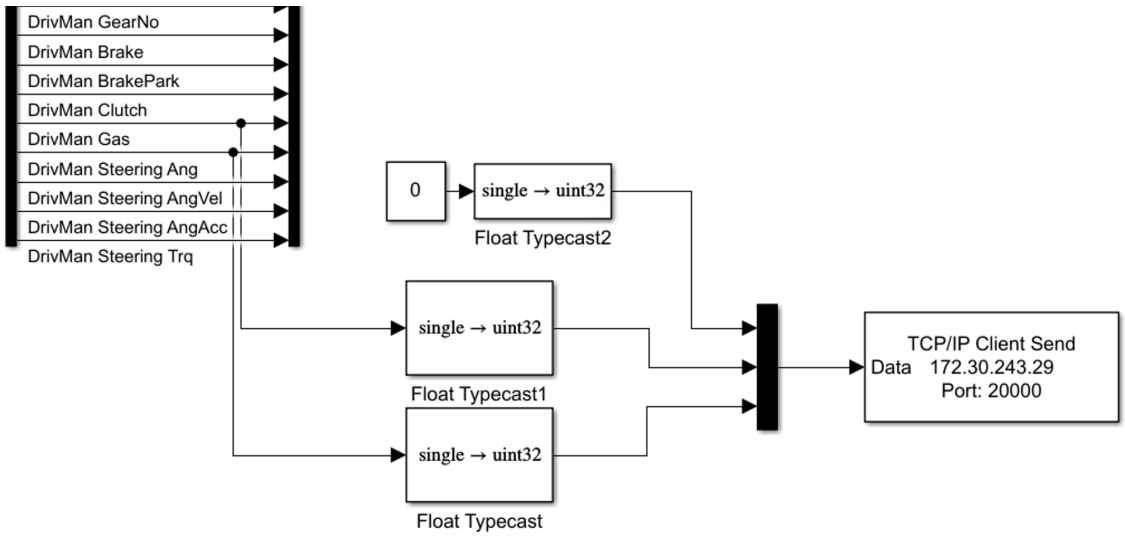


Figure 3.4: Simulink TCP/IP interface for transmitting driver steering and throttle inputs to the torque vectoring controller [67].

These output topics are then received by the TCP to ROS interface and transmitted back to CarMaker through another TCP socket connection. This architecture guarantees bidirectional communication between the developed torque vectoring controller and the external simulation environment, enabling efficient testing, validation, and debugging of the algorithm before deployment on the real vehicle.

The control strategy is designed according to a hierarchical and modular architecture, which enhances flexibility, reusability, and ease of future extensions. The overall structure is composed of four main functional layers.

- **Reference Generator**, which computes the desired vehicle dynamic targets (yaw rate or side-slip angle) based on the current vehicle state and driving conditions;
- **High-Level Yaw Controller**, responsible for determining the corrective yaw moment and the total traction torque required to track the reference signals;
- **Low-Level Torque Allocation Layer**, which distributes the requested total torque and yaw moment into individual wheel torque commands, while enforcing actuator limits and safety constraints.
- **Traction Control Layer**, which recalculates the individual torque in order to control the slip ration in that specific wheel.

Each layer is implemented through well-defined interfaces to guarantee modularity and interchangeability. This architecture follows the SOLID design principles, particularly the Dependency Inversion Principle, where high-level modules depend on abstractions rather than concrete implementations. Such an approach allows different control strategies, reference generation methods, torque allocation algorithms, or traction control techniques to be evaluated and replaced without modifying the overall system structure (e.g. making the upgrade from 2 to 4 motors) [68].

Figure 3.5 illustrates the class diagram of the proposed torque vectoring architecture. The TV Node represents the ROS2 [66] control node responsible for handling sensor subscriptions, executing the torque vectoring logic, and publishing the resulting wheel torque commands. The node interacts with the Torque Vectoring core module, which orchestrates the control flow among the different control layers.

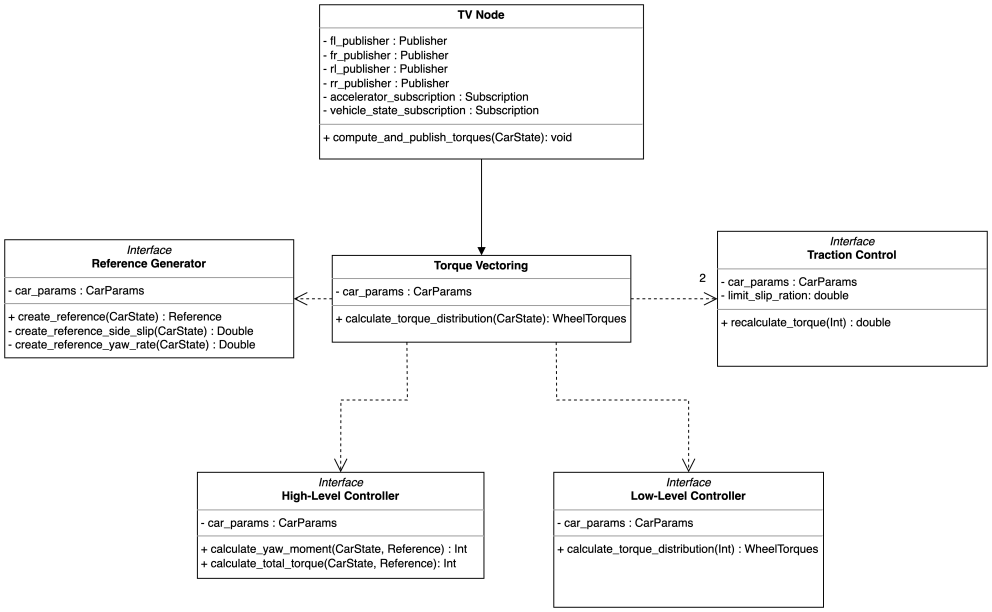


Figure 3.5: UML Class Diagram representing the architecture of the torque vectoring proposed solution.

This layered approach ensures a clear separation of responsibilities between reference generation, control computation, and actuator-level execution, thereby improving code maintainability, scalability, and experimental flexibility.

The execution flow of the torque vectoring system follows the hierarchical control architecture previously presented. During each control cycle, the Torque Vectoring module first requests the desired vehicle dynamic references from the Reference Generator, including the target yaw rate and sideslip angle.

These references, together with the current vehicle state, are then used by the High-Level Controller to compute the corrective yaw moment and the total traction torque required to achieve the desired vehicle behavior. Subsequently, the Low-Level Controller distributes the requested torque among the individual driven wheels while considering actuator and system constraints.

Finally, the Traction Control modules independently evaluate the slip ratio of each wheel and, if necessary, recalculate the wheel torque commands to maintain traction and avoid excessive wheel slip. This sequential interaction between the different control layers is illustrated in Figure 3.6.

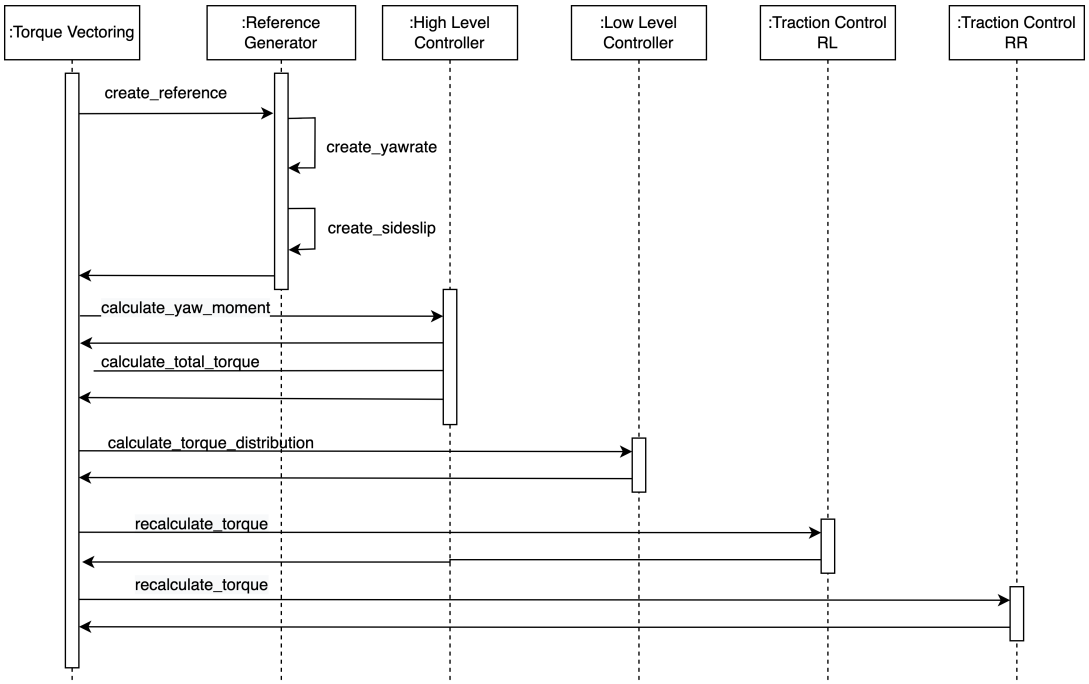


Figure 3.6: UML sequence diagram of the torque vectoring system execution flow.

3.2 Simulation Environment

The validation of the proposed torque vectoring strategy is carried out through a structured experimental process with simulation-based testing. This approach enables systematic performance assessment under controlled and repeatable conditions, while reducing risks during early development phases.

Hardware-in-the-loop or on-vehicle validation was not feasible within the project timeline, as the FS FEUP 03 prototype was still under construction during the development of this work. The use of IPG CarMaker with a model parameterised from CAD data and TTC [69] tire characterisation data represents the industry-standard approach for this stage of development, and forms the basis of the evaluation presented in this chapter.

A simulation environment was adapted to reproduce the dynamic behaviour of the FS FEUP 03 electric prototype. The simulation framework was implemented using IPG CarMaker [17] and includes:

- A nonlinear vehicle dynamics model with 19 DoF [70] incorporating load transfer effects, aerodynamic forces, and nonlinear tire behavior;
- Independent rear-wheel electric motor models, including torque saturation limits;
- A road model capable of reproducing different track layouts and surface conditions;

- A driver model responsible for generating steering, throttle, and braking inputs during simulation scenarios.

The adopted modeling approach allows the reproduction of combined longitudinal and lateral vehicle dynamics and enables the interaction between the torque vectoring controller and the simulated vehicle. The use of IPG CarMaker supports the execution of standardized driving maneuvers. This environment facilitates safe exploration of controller limits and parameter tuning prior to real-world deployment.

The vehicle parameters used to characterise the FS FEUP 03 prototype in the simulation environment are presented in Table 3.1. The inertial and geometric parameters were obtained from the FS FEUP 03 CAD model, while the cornering stiffness values were derived from raw tire characterisation data provided by the Tire Test Consortium (TTC) [69], a standardised dataset widely used in Formula Student vehicle dynamics development.

Table 3.1: FS FEUP 03 vehicle parameters.

Parameter	**Symbol**	**Value**	**Unit**
Total mass	m	270.05	kg
Front axle distance (to CoG)	l_f	0.873	m
Rear axle distance (to CoG)	l_r	0.717	m
Wheelbase	L	1.590	m
Front track width	t_f	1.2	m
Rear track width	t_r	1.2	m
Front cornering stiffness	$K_{y,f}$	483	$\mathrm{N\,rad^{-1}}$
Rear cornering stiffness	$K_{y,r}$	530	$\mathrm{N\,rad^{-1}}$
Wheel radius	R_{wheel}	0.206	m
Maximum motor speed	$n_{m,\max}$	20000 [71]	rpm
Maximum motor power	$P_{m,\max}$	35 [71]	kW
Maximum lateral acceleration	$a_{y,\max}$	15.70	$\mathrm{m\,s^{-2}}$
Max motor torque [71]	$T_{m,\max}$	31.6	Nm
Gear ratio	G_r	13	-
Maximum brake torque (at wheel)	$T_{brake,\max}$	684	Nm
Yaw moment of inertia	I_z	95.0	$\mathrm{kg\,m^2}$
Drag Coefficient	C_D	1.2	-
Lift Coefficient	C_L	-2.9	-
Frontal Aera	A_F	1.1	$\mathrm{m^2}$

3.2.1 Motor

The vehicle is equipped with the DTI Formula Student (F-MOT) motor (more information on Appendix A) and the DTI Formula Student (F-SIC) inverter. To better represent the traction capability of the drivetrain, the motor torque-speed characteristic was converted into the equivalent wheel torque as a function of vehicle speed, assuming no longitudinal slip, and drivetrain losses. As shown in Figure 3.7, the drivetrain provides a nearly constant wheel torque of approximately

410 Nm up to a vehicle speed of around 18 m/s. Beyond this point, the available wheel torque decreases progressively as the motor enters the constant-power operating region, reaching approximately 235 Nm at the maximum vehicle speed of approximately 33 m/s.

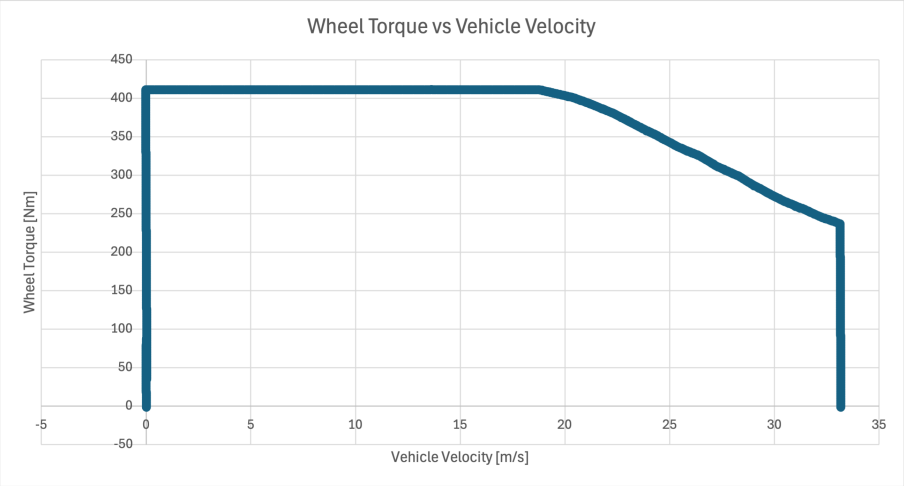


Figure 3.7: Wheel torque vs. vehicle velocity characteristic curve of the DTI F-MOT motor, adapted from [71].

For the purposes of this work, the torque vectoring controller assumes instantaneous torque delivery, meaning that any torque request is fulfilled immediately with no delay or rise time. This assumption is reflected in the simulation environment, where the motor model produces the demanded torque instantly, allowing the control strategy to be evaluated independently of actuator dynamics. Nevertheless, a validation layer is implemented to ensure that the requested torque does not exceed the motor's available torque at a given operating speed. In particular, when operating in the constant-power region above the base speed, torque commands are limited according to the motor torque-speed characteristic, preventing the controller from demanding unrealistically high torque values that cannot be physically delivered by the powertrain:

$$T_{\max,\text{wheel}}(v) = \begin{cases} 410.8, & v \lesssim 17.6 \text{ m/s} \\ \dfrac{35 \times 10^3 \times R_{\text{wheel}}}{v}, & v \gtrsim 17.6 \text{ m/s} \end{cases} \tag{3.2}$$

$$T_{\text{cmd,wheel}}(v, T_{\text{request}}) = \begin{cases} T_{\text{request}}, & |T_{\text{request}}| \leq T_{\max}(v) \\ \text{sign}(T_{\text{request}})\, T_{\max}(v), & |T_{\text{request}}| > T_{\max}(v) \end{cases} \tag{3.3}$$

3.2.2 Tires

The tire model used is the Hoosier 43075 16x7.5-10 R20 (as shown in Figure 3.8) the same compound employed in the FS FEUP 02 prototype, ensuring continuity between the physical vehicle and the simulation environment. The tire behavior in the simulation is represented using the Pacejka Magic Formula tire model [21].



Figure 3.8: Hoosier 43075 16x7.5-10 R20 tires mounted on the FS FEUP 02 prototype.

3.2.3 Driver Model

The driver model, and especially its characterized aggressiveness, represents an important control variable in the evaluation of vehicle control systems. Knowledge of human behaviour while driving a car reveals a pivotal aspect in achieving a closed-loop simulation. As described in the IPGDriver User Manual [72], the Carmaker's driver module receives the intended path, vehicle position, speed and acceleration, and steering wheel torque as inputs. In response, it outputs the accelerator pedal position, the force on the brake pedal, and the steering wheel angle or torque, as illustrated in Figure 3.9. Since the vehicle under study is electric, clutch, gear shifter and handbrake outputs are not considered.

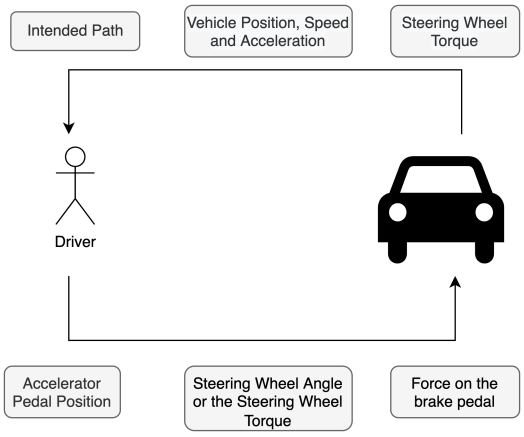


Figure 3.9: Interface between the IPGDriver module and the vehicle model, showing the inputs received by the driver (top) and the control outputs it produces (bottom). Adapted from [72].

In [73], the authors divide driver models into two categories. The simplest approach consists of using classical control theory to model human driving behaviour, particularly through PID-based

approaches and fuzzy control techniques. The second approach aims to model the actual cognitive behaviour of the driver in order to generate the control actions.

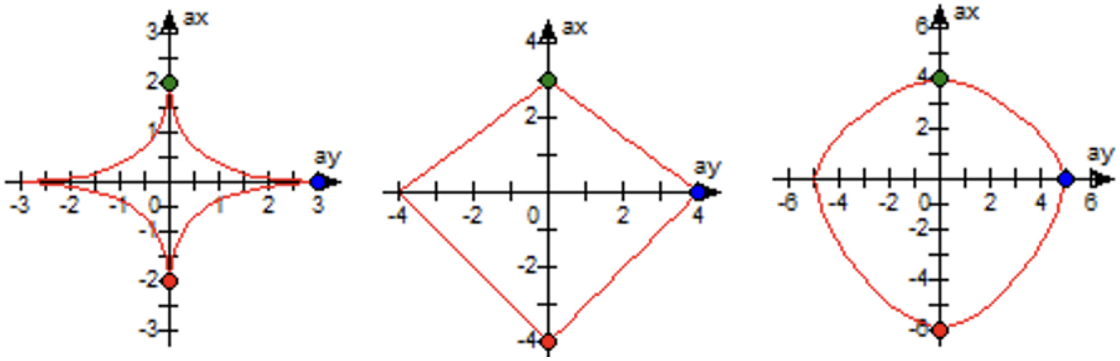
This approach uses two almost independent continuous PID controllers: one for longitudinal control and another for lateral control. The coupling between both controllers occurs mainly in critical driving situations.

The longitudinal controller uses the desired and actual longitudinal accelerations to calculate the throttle input, which then passes through a risk corrector together with the lateral and longitudinal accelerations to prevent unsafe conditions.

The lateral controller mainly uses three components: the lateral displacement, denoted by Δy, which is processed through a PD controller; the Ackermann steering angle δ_A; and the direction error $\Delta\psi$ (defined as the angle between the vehicle velocity vector and the road direction), both processed through a first-order lag. These three components are then combined through a weighted sum together with a counter-steering constraint, mainly used to maintain vehicle stability during transitions between different curve conditions [73].

In CarMaker, the aggressiveness of the driver is characterized through the achievable combination of longitudinal and lateral accelerations, commonly represented by GG diagrams (Figure 3.10). These diagrams define the limits of vehicle operation by describing the maximum acceleration and deceleration capabilities for different lateral acceleration levels, thus reflecting the driving style of the simulated driver. Defensive drivers typically operate with lower acceleration limits, while aggressive drivers are able to exploit a larger portion of the vehicle dynamic capabilities [17].

The driver behaviour model implemented in CarMaker is based on PID control principles, although with a significantly more advanced structure. Instead of reacting only to the current driving conditions, the model predicts a future preview point using Taylor polynomial approximations. Moreover, the model incorporates psychological studies and experimental data collected from real test drivers in order to reproduce human driving behaviour as realistically as possible [74, 72].



(a) Defensive driver. (b) Normal driver. (c) Aggressive driver.

Figure 3.10: GG diagrams for different driver behaviours, from [17, 74].

IPG CarMaker also allows the modelling of specific parameters related to both longitudinal and lateral driver behaviour. For longitudinal control, parameters such as cruising speed and longitudinal reaction time can be adjusted. Additional parameters associated with declutching and

gear shifting are also available, although they are not relevant in the present context due to the electric vehicle configuration. Figure 3.11 presents the longitudinal parameters available in IPG CarMaker together with the values corresponding to the default aggressive driver model.

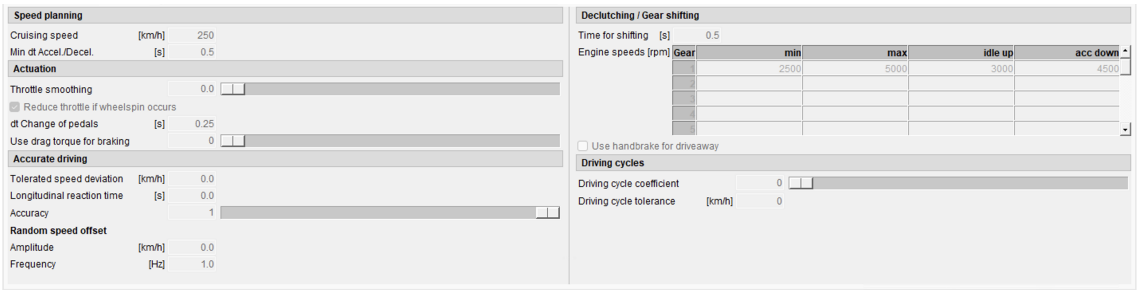


Figure 3.11: IPG CarMaker driver's longitudinal parameters, from [17].

Similarly, the lateral behaviour can also be modelled through a set of configurable parameters. These include the corner-cutting coefficient, maximum steering wheel velocity, steering wheel acceleration, steering torque, and torque rate. Figure 3.12 shows the lateral parameters available in IPG CarMaker and the corresponding values for the default aggressive driver model.

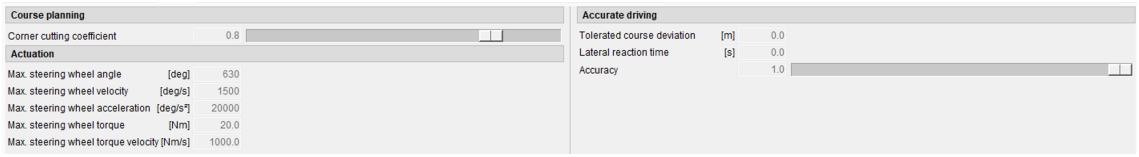


Figure 3.12: IPG CarMaker driver's lateral parameters, from [17].

In addition to the standard drivers, IPG CarMaker also provides a Race Driver mode, which aims to achieve the maximum possible velocity on a given track by using a slip-based controller that operates near the optimal tire slip limits [17].

Outside the scope of this work, CarMaker also allows the modelling of traffic-related parameters, particularly those associated with vehicle-following behaviour, such as time gap and following distance.

3.2.4 Simulation Environment Validation

To validate the communication between the ROS environment and the Simulink environment for CarMaker, a set of representative vehicle dynamics tests were conducted. These tests aim to verify that the system produces physically consistent results across longitudinal and lateral scenarios. The validation includes: (i) a Formula Student acceleration event (75 m) [12], (ii) the ISO 4138 steady-state circular test [75], and (iii) a single-wheel torque input test.

The simulations were executed using the IPG CarMaker closed-loop driver model, responsible for generating steering, throttle, and braking commands while tracking a predefined trajectory and target vehicle behavior. However, in the implemented scenarios, the throttle command was overwritten by a predefined input profile, meaning that the driver model did not influence the vehicle

acceleration. The driver model still provided the remaining control actions through feedback interaction with the vehicle dynamics model, affecting the resulting yaw response, lateral acceleration, tire slip behavior, and overall vehicle stability.

3.2.4.1 Formula Student Acceleration Test

The acceleration test consists of a straight-line run over 75 meters, reproducing the Formula Student acceleration event. Different throttle inputs were applied to evaluate longitudinal vehicle performance.

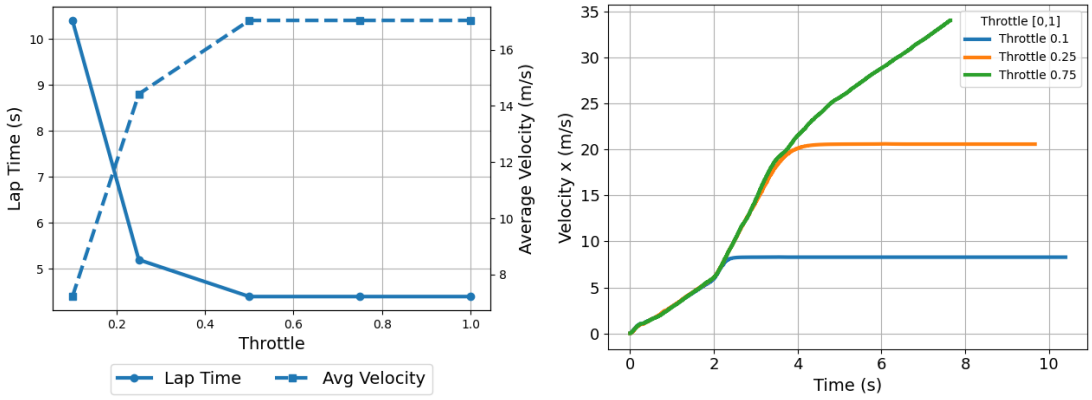


Figure 3.13: Effect of throttle on performance in the 75 m acceleration event: (left) lap time and average velocity; (right) velocity profiles for different throttle inputs.

The results shown in Figure 3.13 indicate that increasing the throttle substantially reduces the completion time and increases the average velocity up to approximately 25% throttle. Beyond this point, further increases in throttle produce only marginal improvements in performance, with the velocity profiles becoming nearly identical.

3.2.4.2 ISO 4138 Circular Test

The test procedure follows the ISO 4138 standard, which specifies open-loop methods for evaluating the steady-state cornering behaviour of passenger cars and light trucks. In this test, the vehicle is driven on a constant-radius circular path while maintaining steady operating conditions, allowing the assessment of lateral dynamics characteristics such as yaw rate response, lateral acceleration, understeer/oversteer behaviour, and overall stability. In the present study, the manoeuvre was performed on a circular track with a radius of 100 m for 70 s.

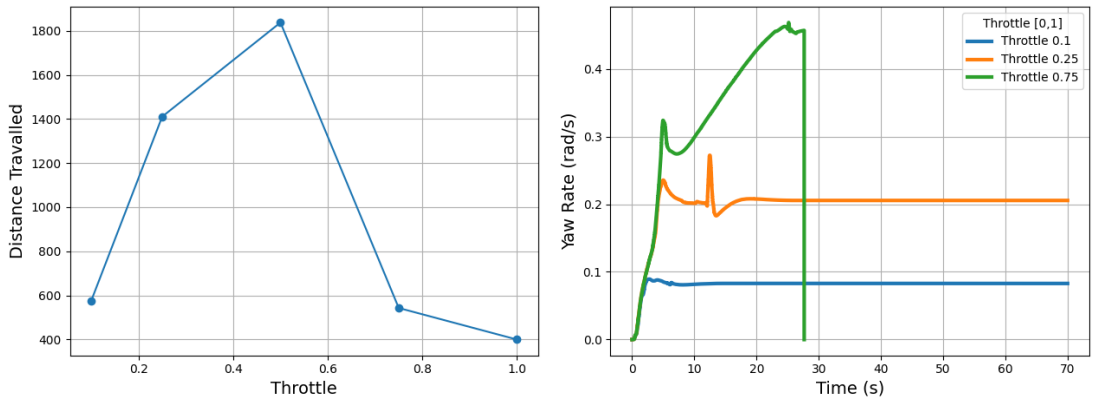


Figure 3.14: ISO 4138 test results: (left) distance traveled for different throttle inputs; (right) yaw rate response over time.

The results described in Figure 3.14 indicate that increasing throttle initially improves performance by allowing greater distance to be traveled. However, excessive throttle leads to instability, reducing the achievable distance. In particular, for throttle values equal to or above 0.5, the vehicle becomes unstable and leaves the track, which explains the degradation in the recorded performance metrics.

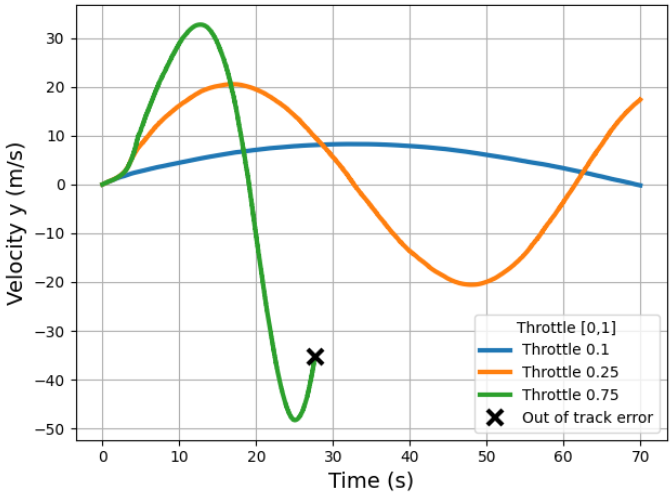


Figure 3.15: Longitudinal velocity as a function of time for different throttle inputs.

The results in Figure 3.15 show the relationship between throttle input and the global Y-direction velocity component. At low throttle (0.1), the response remains smooth and limited in magnitude. For moderate throttle (0.25), the magnitude of v_y increases, reflecting higher vehicle speeds and more pronounced changes in the vehicle trajectory. At high throttle (0.75), the global Y-direction velocity reaches significantly larger values, indicating more aggressive vehicle motion. Under these conditions, the vehicle is unable to maintain the prescribed trajectory and leaves the track, demonstrating a loss of path-tracking capability.

3.2.4.3 Single-Wheel Torque Input Test

To validate drivetrain-to-vehicle coupling, a test was performed by applying torque to a single wheel and observing the resulting yaw response, with the steering fixed at zero to eliminate driver influence.

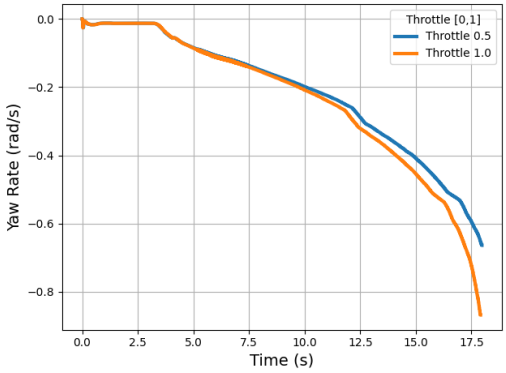


Figure 3.16: Yaw rate response under left-wheel torque input.

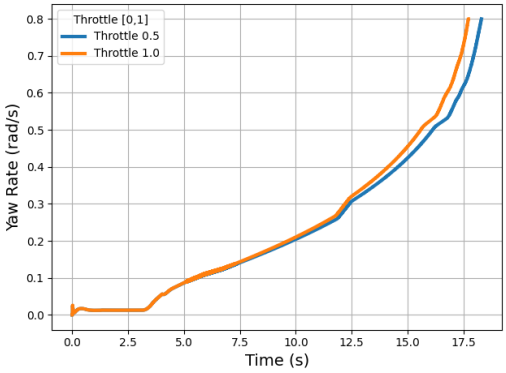


Figure 3.17: Yaw rate response under right-wheel torque input.

The results in Figures 3.16 and 3.17 confirm that asymmetric torque inputs generate a yaw moment, producing a measurable rotational response. Higher torque levels lead to stronger yaw rates, consistent with expected vehicle dynamics behavior.

Overall, the results demonstrate that the ROS-Simulink integration is functioning correctly. The simulation captures both longitudinal and lateral dynamics, including nonlinear effects and stability limits, providing a reliable foundation for further controller development and system validation.

3.2.4.4 Simulation Timing and Controller Execution

The simulation was executed using a fixed-step discrete solver with a sampling period of $T = 1$ ms (1000 Hz). The vehicle dynamics model, torque vectoring controller, and traction control controller were executed synchronously at this rate. Consequently, all control algorithms were implemented in discrete time, with the PI controllers updating their proportional and integral terms at each sampling instant.

3.3 Algorithm

Figure 3.18 illustrates the proposed hierarchical control architecture for yaw-rate tracking and rear-wheel torque vectoring. The overall control structure is adapted from the work of Fujimoto et al. [16] and consists of two major layers: an upper-level yaw moment controller and a lower-level torque allocation controller.

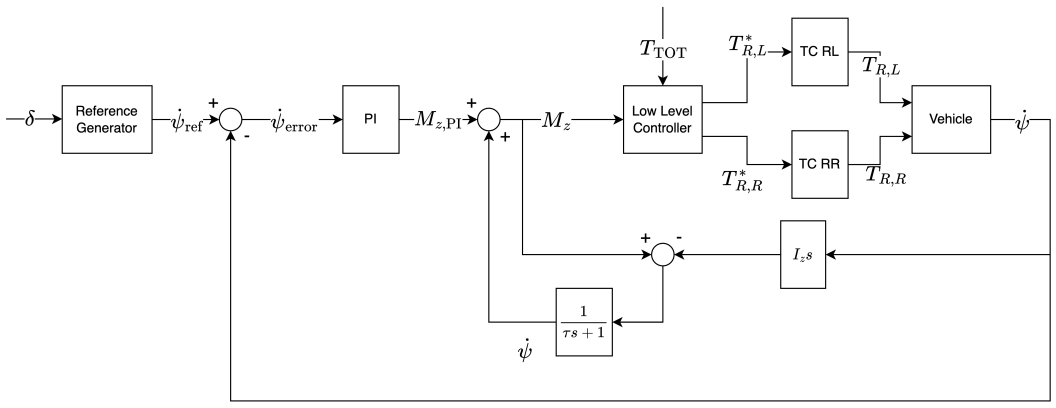


Figure 3.18: Hierarchical yaw-rate tracking and torque distribution control architecture for the rear-wheel torque vectoring system. The upper-level controller generates the desired corrective yaw moment based on the yaw-rate tracking error using a PI controller, while the low-level controller allocates the required wheel torques to the left and right rear wheels. The resulting wheel torques are applied to the vehicle dynamics model, and the measured yaw response is fed back through a first-order vehicle yaw dynamics model for closed-loop stability enhancement.

3.3.1 Reference Generator

The steering input is first processed by the reference generator to compute the desired yaw rate $\dot{\psi}_{ref}$, based on the law introduced in Equation 3.4 [50, 11, 13, 8] and is passes throught a maximum lateral acceleration saturation filter, such as presented in equations 3.5 and 3.6 [8] . This reference represents the target vehicle yaw response under the driver steering command and serves as the primary stability objective for the controller.

$$\dot{\psi}_{\text{des}} = \frac{v}{L(1+K_u v^2)}\delta \tag{3.4}$$

$$|\dot{\psi}_{\text{max}}| = \left|\frac{a_{y,\text{max}}}{v_x}\right| \tag{3.5}$$

$$\dot{\psi}_{\text{ref}} = \begin{cases} \dot{\psi}_{\text{des}}, & \text{if } |\dot{\psi}_{\text{des}}| < |\dot{\psi}_{\text{max}}| \\ \dot{\psi}_{\text{max}}, & \text{otherwise} \end{cases} \tag{3.6}$$

where $\dot{\psi}_{\text{ref}}$ is the reference yaw rate, $\dot{\psi}_{\text{des}}$ is the desired yaw rate obtained from the steady-state vehicle model, $\dot{\psi}_{\text{max}}$ is the maximum allowable yaw rate, v is the vehicle longitudinal velocity, v_x is the longitudinal velocity, L is the wheelbase, δ is the steering angle, K_u is the understeer coefficient, and $a_{y,\text{max}}$ is the maximum allowable lateral acceleration.

3.3.2 High-Level Controller

The actual yaw rate $\dot{\psi}$ obtained from the vehicle model is fed back and compared with the reference yaw rate to generate the yaw-rate tracking error. This error is then supplied to a proportional-integral (PI) controller with anti-windup, which computes the corrective yaw moment demand $M_{z,PI}$.

A PI controller was selected for the high-level yaw moment computation because the vehicle yaw dynamics around the operating point can be reasonably approximated by a first-order system. This makes the PI controller suitable for yaw-rate reference tracking, providing a balance between implementation simplicity and control performance. In particular, the integral action eliminates steady-state yaw-rate errors, preventing persistent deviations between the desired and measured vehicle response.

To further account for the vehicle yaw dynamics and improve the transient response, an additional feedback compensation loop based on a disturbance observer, referred to as the yaw-moment observer [16], is incorporated.

The feedforward yaw moment requires the reference yaw acceleration, obtained by differentiating the yaw-rate signal, using the following formula:

$$\ddot{\psi}_k = \frac{\dot{\psi}_k - \dot{\psi}_{k-1}}{\Delta t} \tag{3.7}$$

Numerical differentiation amplifies high-frequency components and discontinuities in the reference signal, which can produce unrealistic feedforward yaw-moment demands. Therefore, a first-order low-pass filter was applied to the computed yaw acceleration. The filter time constant was selected empirically through simulation-based tuning. A time constant of 150 ms was selected to attenuate high-frequency components while preserving the dominant vehicle yaw dynamics relevant to path-following maneuvers. The filtered yaw acceleration was then used in the feedforward term M_z.

$$G(s) = \frac{1}{\tau_s s + 1} \tag{3.8}$$

where τ_s is the vehicle yaw time constant. The output of this model provides the compensation signal I_s, which is added to the PI-generated yaw moment demand.

While more advanced controllers such as LQR or MPC may provide improved performance under significant model uncertainty or varying operating conditions, they require a more accurate vehicle model and increased tuning complexity. The PI-based approach was therefore selected due to its lower implementation complexity, straightforward tuning, and clear physical interpretation.

The final demanded yaw moment is obtained as:

$$M_z = \mathrm{sat}_{[-M_{z,\max}, M_{z,\max}]} \left(M_{z,PI} + I_s\right) \tag{3.9}$$

where $M_{z,\max}$ is determined by the maximum differential torque capability of the rear axle, as illustrated in Figure 3.19 and in algorithm 1. The implementation of the algorithm, including the

implementation details can be seen in Appendix B.

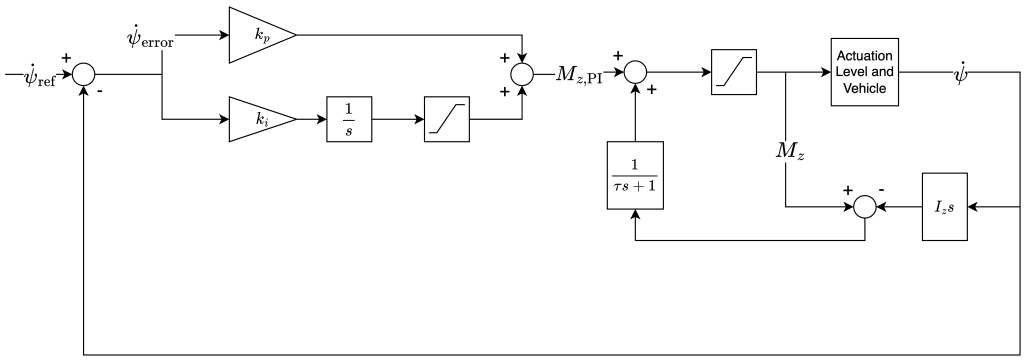


Figure 3.19: Architecture of the High-Level Controller System.

Algorithm 1 Discrete-Time PI Controller with Feedforward Compensation

Require: Measured yaw rate $\dot{\psi}[k]$, reference yaw rate $\dot{\psi}_{\text{ref}}[k]$, sampling time Δt
Ensure: Requested yaw moment $M_z[k]$

$e[k] = \dot{\psi}_{\text{ref}}[k] - \dot{\psi}[k]$

$P[k] = k_p\, e[k]$

$I[k] = I[k-1] + k_i\, \Delta t\, e[k]$

$I[k] = \text{sat}(I[k])$

$M_{z,\text{PI}}[k] = P[k] + I[k]$

$\ddot{\psi}[k] = \dfrac{\dot{\psi}[k] - \dot{\psi}[k-1]}{\Delta t}$

$M_{\text{in}}[k] = I_z\, \ddot{\psi}[k]$

$e_{\text{FF}}[k] = M_z[k-1] - M_{\text{in}}[k]$

$M_{z,\text{FF}}[k] = \alpha M_{z,\text{FF}}[k-1] + (1-\alpha) e_{\text{FF}}[k],\ \alpha = \dfrac{\tau}{\tau + \Delta t}$

$M_z[k] = \text{sat}(M_{z,\text{PI}}[k] + M_{z,\text{FF}}[k])$

return $M_z[k]$

The PI controller gains were determined through an iterative trial-and-error tuning process in simulation. The proportional gain (K_p) was initially increased from a low value until a satisfactory transient response was obtained. Increasing (K_p) improved the responsiveness of the yaw moment controller and reduced tracking error; however, excessively large values resulted in oscillatory behaviour and increased control effort. A value of ($K_p = 7$) was selected as it provided a suitable compromise between response speed and stability.

Subsequently, the integral gain (K_i) was adjusted to eliminate the residual steady-state error. Increasing (K_i) improved the controller's ability to reject persistent disturbances and achieve accurate yaw moment tracking, but excessively high values introduced overshoot and degraded transient performance. Through repeated simulation tests, ($K_i = 25$) was found to provide rapid error correction while maintaining stable operation.

Table 3.2 summarises the controller parameters obtained after tuning.

Table 3.2: Tuned parameters of the PI yaw moment controller.

Parameter	**Symbol**	**Value**
Proportional gain	K_p	7
Integral gain	K_i	25
Anti-windup clamp	W_{aw}	1000
Yaw time constant	τ_s	0.15 s
Control frequency	f_c	1000 Hz

3.3.3 Low-Level Controller

The desired yaw moment generated by the upper-level controller is realized through a difference in the longitudinal forces produced by the left and right rear wheels. Assuming equal moment arms with respect to the vehicle centerline, the resulting yaw moment is given by [4]:

$$M_{z,RL} = \frac{t_r}{2}(F_{x,R} - F_{x,L}) \tag{3.10}$$

where $M_{z,RL}$ is the yaw moment generated by the difference in the longitudinal forces acting on the rear wheels, t_r is the rear track width, and $F_{x,R}$ and $F_{x,L}$ are the longitudinal tire forces at the rear right and rear left wheels, respectively.

The total desired yaw moment M_z is subsequently transmitted to the low-level controller together with the total traction torque demand T_{TOT}. The low-level controller performs torque allocation by determining the desired left and right rear wheel torques, using Equations 3.11 and 3.12, denoted as $T_{R,L}$ and $T_{R,R}$, respectively [10].

$$\Delta T_{\mathrm{LR}} = \frac{M_z \cdot R_{\mathrm{wheel}}}{t_r} \tag{3.11}$$

$$\begin{cases} T_{\mathrm{RL}} = \dfrac{T_{\mathrm{TOT}}}{2} - \Delta T_{\mathrm{LR}}, \\ T_{\mathrm{RR}} = \dfrac{T_{\mathrm{TOT}}}{2} + \Delta T_{\mathrm{LR}}, \end{cases} \tag{3.12}$$

where M_z is the requested yaw moment, R_{wheel} is the wheel radius, t_r is the rear track width, T_{TOT} is the total torque requested to the rear axle, and T_{RL} and T_{RR} are the torque commands applied to the rear left and rear right motors, respectively.

3.3.4 Traction Control

The traction control system is based on a peak-seeking slip control approach. The slip ratio (equation 3.13) is obtained directly from the simulator, making it not necessary to do the slip ratio estimation. Since the primary focus of this work is the evaluation of the proposed torque-vectoring strategy, a computationally lightweight traction controller was adopted instead of a model-based

friction estimation or optimal-slip tracking approach. Rather than estimating the tire-road friction coefficient μ explicitly, the controller monitors the wheel slip-rate and adapts the available drive torque according to the observed slip dynamics.

$$\sigma_x = \frac{R_{\text{wheel}}\omega - v_x}{v_x} \tag{3.13}$$

The slip-rate reflects the dynamic evolution of wheel slip and is influenced by the available tire-road traction. Consequently, changes in road conditions are reflected in the measured slip-rate and can be exploited as a feedback signal for traction control. The commanded torque is adjusted according to the sign of the slip-rate, allowing the controller to adapt to changing operating conditions and tend to maintain operation near the region of maximum available traction. The slip-rate is estimated at each timestep as:

$$\dot{\sigma}_{x,k} = \frac{\sigma_{x,k} - \sigma_{x,k-1}}{\Delta t} \tag{3.14}$$

The controller activates when $\sigma_x > \sigma_{x,th}$, where $\sigma_{x,th}$ is the slip activation threshold. Once active, the commanded torque limit T_{lim} is updated multiplicatively at each timestep according to the sign of the slip-rate:

$$T_{lim,k} = \begin{cases} k_r \cdot T_{lim,k-1} & \text{if } \dot{\sigma}_{x,k} > \varepsilon \\ k_i \cdot T_{lim,k-1} & \text{if } \dot{\sigma}_{x,k} < -\varepsilon \\ T_{lim,k-1} & \text{if } |\dot{\sigma}_{x,k}| \leq \varepsilon \end{cases} \tag{3.15}$$

The rationale is as follows: a positive slip-rate indicates that wheel slip is increasing and therefore triggers a reduction in the allowable drive torque. Conversely, a negative slip-rate indicates that wheel slip is decreasing, allowing the torque limit to be gradually restored. This adaptation mechanism tends to maintain operation near the region of maximum available traction without requiring explicit estimation of the tire-road friction coefficient.

The output torque applied to each rear wheel is then:

$$T_R = \min\left(T_{driver},\ T_{lim,k}\right) \tag{3.16}$$

ensuring the traction control never commands more torque than the driver requested.

The tuning parameters used for the peak-seeking traction control algorithm are summarized in Table 3.3. The slip activation threshold $\sigma_{x,th}$ determines when the controller becomes active, while the torque reduction and recovery factors, k_r and k_i, respectively, govern the rate at which the torque limit is adjusted in response to changes in wheel slip. The slip-rate deadband ε is introduced to prevent excessive control activity caused by measurement noise and small fluctuations in the estimated slip derivative. These parameters were selected empirically to provide stable traction control intervention while maintaining smooth torque delivery.

Table 3.3: Traction control parameters.

Parameter	**Symbol**	**Value**
Slip activation threshold	$\sigma_{x,th}$	0.075
Torque reduction factor	k_r	0.97
Torque recovery factor	k_i	1.0007
Slip-rate deadband	ε	0.0015

Algorithm 2 represents the pseudocode of the traction control system.

Algorithm 2 Slip-rate Traction Control Algorithm.

Require: Slip ratio σ_x, driver torque T_d

Compute slip-rate:

$$\dot{\sigma}_x = \frac{\sigma_x - \sigma_{x,\text{prev}}}{\Delta t}$$

$\sigma_{x,\text{prev}} \leftarrow \sigma_x$
if TCS is inactive **then**
    $T_{\text{lim}} \leftarrow T_d$
    if $\sigma_x > \sigma_{x,\text{th}}$ **then**
        Activate TCS
    end if
    return T_d
end if
if $\sigma_x < 0.8\sigma_{x,\text{th}}$ **then**
    Deactivate TCS
    $T_{\text{lim}} \leftarrow T_d$
    return T_d
end if
if $\dot{\sigma}_x > \varepsilon$ **then**
    $T_{\text{lim}} \leftarrow k_r T_{\text{lim}}$ ▷ Slip increasing: reduce torque
else if $\dot{\sigma}_x < -\varepsilon$ **then**
    $T_{\text{lim}} \leftarrow k_i T_{\text{lim}}$ ▷ Slip decreasing: recover torque
else
    Hold current torque limit
end if
return T_{lim}

Finally, the wheel torques generated by the traction control and torque-vectoring modules are applied to the vehicle dynamics model. The resulting vehicle states are then fed back to the supervisory control architecture for the subsequent control step.

Chapter 4

Results and Discussion

This chapter presents and discusses the results obtained from the vehicle dynamics simulations performed throughout this work. The analysis focuses on evaluating the influence of the implemented torque vectoring strategy on the vehicle behaviour under different driving conditions.

The results are presented through a comparison between the baseline vehicle configuration, without torque vectoring and the vehicle equipped with the proposed torque vectoring system, with the following configurations:

- **Baseline (No TV):** Torque distributed equally to both motors, with no anti-skid implementation.
- **Torque Vectoring (TV):** Torque distributed between both motors to generate the corrective yaw moment required for path tracking. A value of $K_u = 0$ was adopted to provide a neutral steering reference, allowing the torque vectoring controller to compensate for the vehicle's inherent oversteer characteristic resulting from the suspension setup, thereby improving yaw stability. Motor torque was limited according to the TCS strategy described previously.

The test scenarios were chosen in order to isolate some variables, and evaluate and discuss them individually. The main reasoning was to do the separation between the controller and the driver. The Skidpad test was chosen as the end-to-end test to create a validation on a Formula Student track. Being a closed-loop test allows the comparison of lap times with past Formula Student teams implementations.

4.1 Open-Loop Maneuvers with Constant Steering

The first scenario is called constant steering. In this test, all driver inputs are overwritten with constant values in order to remove the driver influence from the results and evaluate the controller in isolation. The test starts with a reference yaw rate of zero and suddenly increases it to higher values in order to assess the controller's response.

The throttle input is overwritten with a constant value corresponding to a specific velocity profile. The steering angle starts at 0° and is then set to a constant value of 10° once the desired velocity is reached. All scenarios use a friction coefficient $\mu = 0.8$.

Figure 4.1 presents the yaw-rate response of the vehicle equipped with the torque vectoring (TV) controller under a constant 10-degree steering input for different vehicle velocities. The results are compared against the reference yaw rate and the vehicle response without torque vectoring (No TV). The controller demonstrates accurate steady-state tracking over the tested velocity range. At higher velocities, the system exhibits small transient oscillations and overshoot due to the higher responsiveness of the vehicle dynamics.

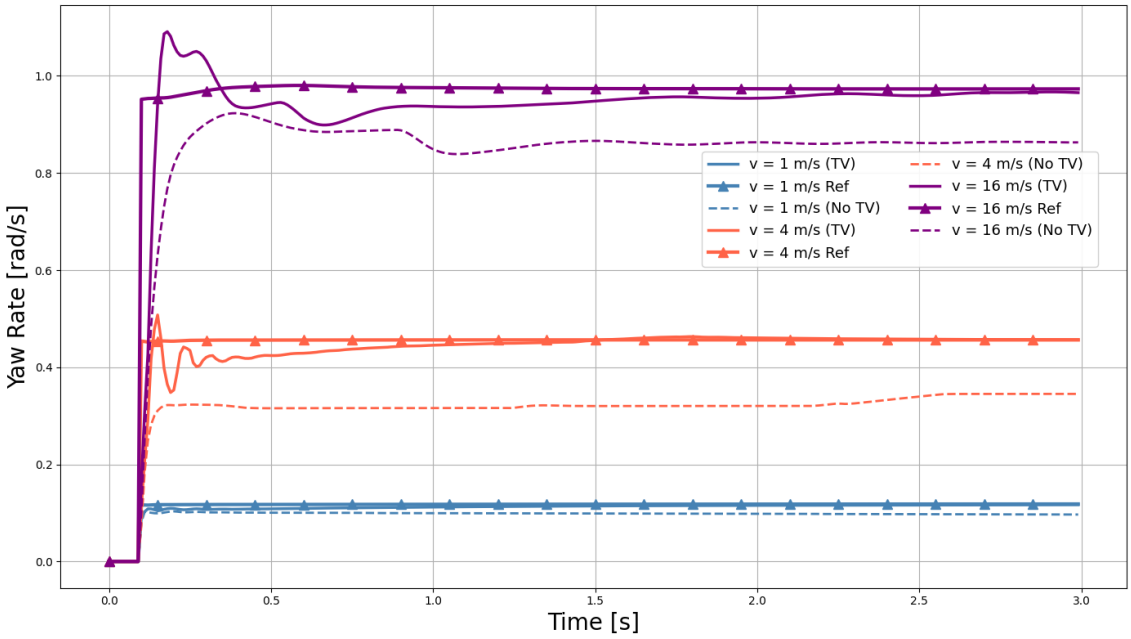


Figure 4.1: Comparison between reference and actual yaw-rate responses for different vehicle velocities under a constant 10-degree steering input.

What we can conclude in Figure 4.1 is that, firstly, in all scenarios, the corrective yaw moment caused by the F_x difference works towards achieving the desired yaw rate.

In all three velocities, the TV implementation converges much more closely to the reference yaw rate than the baseline. One other point worth noting is that the baseline implementation's steady-state error tends to increase with velocity. On the other hand, the TV implementation always converges, but the time until convergence is complete also tends to increase.

Another interesting point is the transient behaviour of the controller. First, we can see that at 16 m/s, the vehicle has an overshoot of approximately 0.11 rad/s, followed by a period of oscillation before settling. This may indicate that the controller's underlying dynamics scale with speed in a way that pushes the closed-loop system towards lower damping ratios at higher speeds.

In conclusion, for all speeds, the TV yaw rate response is always better than the baseline, creating a more reduced damping and overshoot at higher speeds.

Figure 4.2 illustrates the evolution of the vehicle sideslip angle for selected velocities with and without torque vectoring control. At low and medium speeds, both configurations exhibit similar steady-state sideslip behavior, although the TV controller reduces transient oscillations and improves stability.

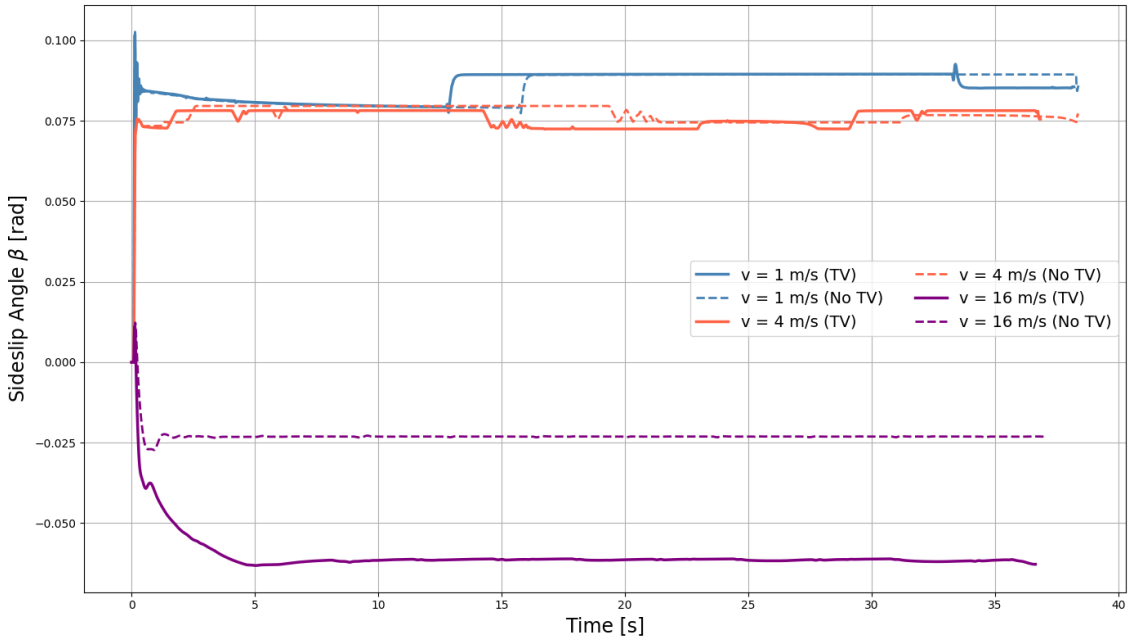


Figure 4.2: Comparison between actual sideslip angles for different vehicle velocities under a constant 10-degree steering input.

As seen, all 3 speeds eventually reach a bounded, steady sideslip value. Analogous to the yaw rate response time, the time to reach this steady sideslip value increases with velocity.

A noticeable deviation is observed around 16 m/s, where TV implementation exhibit an atypical behaviour. Analysis of the simulation data confirms that this operating point corresponds to the vehicle approaching its lateral dynamic limit: the lateral acceleration reaches approximately 15.6 m/s^2, close to the maximum value of 15.7 m/s^2 defined in Table 3.1. At this condition, lateral weight transfer reduces the inner rear wheel normal force from approximately 670 N to 220 N, while the outer rear wheel load increases to 1150 N. The resulting rear slip angles reach approximately 10.8º and 9.5º on the inner and outer wheels respectively.

Notably, this load imbalance is more pronounced in the TV case (440 N / 1150 N) than in the baseline (670 N / 950 N). Since lateral load transfer depends mainly on lateral acceleration, this larger imbalance is consistent with the higher a_y sustained by the TV controller at this operating point. As a result, the inner rear tire operates closer to, or beyond, its peak slip angle under TV than under the baseline, which is consistent with the larger steady-state sideslip magnitude observed for the TV case at this speed.

Figure 4.3 compares the vehicle trajectories obtained with the Torque Vectoring (TV) and baseline at vehicle speeds of 1 m/s, 4 m/s, and 16 m/s. For all tested speeds, the TV-equipped vehicle follows a trajectory with a smaller turning radius, while the non-TV configuration creates

a wider path throughout the maneuver. The difference between the two trajectories becomes more noticeable as vehicle speed increases, indicating that the TV system effectively generates additional yaw moment and enhances cornering performance. These results demonstrate the ability of torque vectoring to improve vehicle agility and directional response over a wide range of operating conditions.

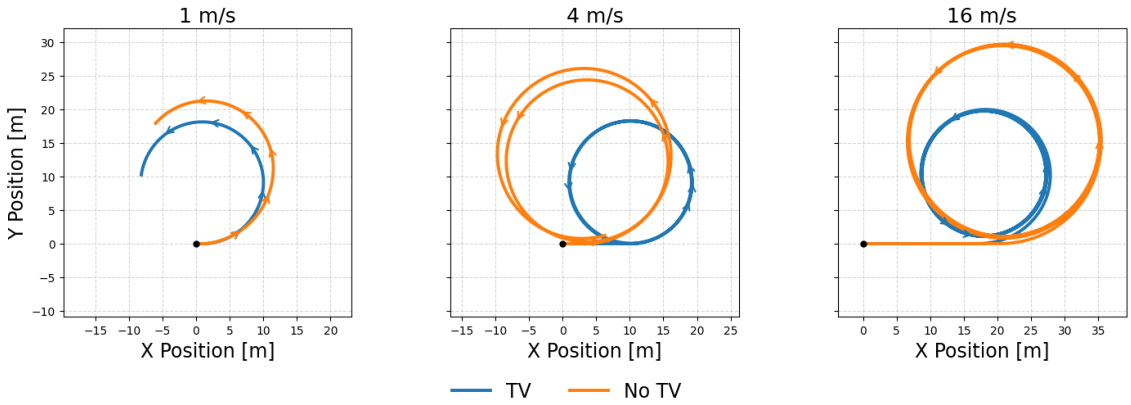
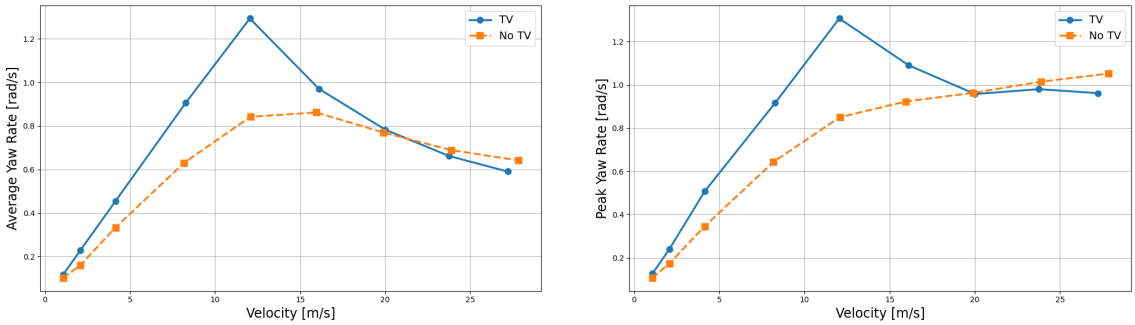


Figure 4.3: Comparison of vehicle trajectories for Torque Vectoring (TV) and non-Torque Vectoring (non-TV) architectures at a constant steering angle of 10° and longitudinal velocities of 1, 4, and 16 m/s.

Figure 4.4a represents the variation of the average yaw rate as a function of vehicle velocity for the TV and baseline configurations, while Figure while Figure 4.4b shows the corresponding peak yaw rate signal. In both figures, the yaw rate increases with velocity until 12 m/s, after which it starts to decrease for both configurations.



(a) Average yaw rate versus vehicle velocity.

(b) Peak yaw rate versus vehicle velocity.

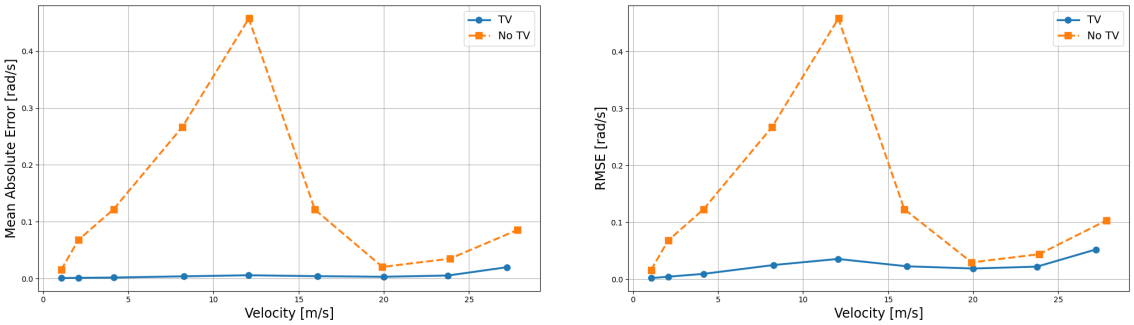
Figure 4.4: Comparison of average and peak yaw rates as a function of vehicle velocity for the TV and No TV configurations under a constant 10-degree steering input.

Until 20 m/s, the torque vectoring controller produces consistently higher average and peak yaw rates than the baseline, indicating improved responsiveness. At velocities higher that 20 m/s, this trend inverts: the baseline's average and peak yaw rates are greater than those of the TV configuration. At these speeds, the baseline's steady-state yaw rate begins to exceed the reference signal, and its response becomes increasingly oscillatory, producing larger overshoots that increase

both the average and peak yaw rate metrics. In contrast, the TV-controlled vehicle continues to reach the reference yaw rate, so its lower absolute values at high speed reflect closer reference tracking error. This corroborates with the time-domain behavior observed in Figure 4.1, where the TV response settles much closer to the reference than the baseline.

Overall, these results indicate that torque vectoring improves responsiveness and reduces error across the tested speed range.

Figure 4.5a shows the mean yaw-rate tracking error for different vehicle velocities. Analogously, Figure 4.5b presents the RMSE for the same velocities.



(a) Mean absolute yaw-rate tracking error versus vehicle velocity.

(b) Root mean squared error (RMSE) of the yaw-rate response versus vehicle velocity.

Figure 4.5: Comparison of yaw-rate tracking performance metrics as a function of vehicle velocity for the TV and No TV configurations under a constant 10-degree steering input.

The most important conclusion is that the TV configuration reaches the reference yaw rate in all of the tested velocities, including at 1 m/s, indicating that even at low speeds, torque vectoring is advantageous for tracking the desired yaw rate.

Another interesting insight is the error peak at 12 m/s, present in both the mean absolute error and the RMSE for the baseline configuration. This happens because this is the velocity profile where the reference yaw rate is highest among all tested conditions, so the absolute magnitude of any tracking error is naturally larger here as well.

A deeper observation is that, for the baseline, the MAE and RMSE curves are nearly identical at every velocity, indicating that the tracking error is consistent in sign and magnitude over time. For the TV configuration, by contrast, the RMSE is noticeably larger than the MAE at several velocities, suggesting that the small residual error in the TV case is more transient/oscillatory in nature rather than a steady offset, consistent with the overshoot behaviour observed in Figure 4.1.

The controller performance results for all tested velocities are summarized in Tables 4.1 and 4.2.

Table 4.1: Tracking error comparison vs. velocity.

Velocity	**No TV Mean Absolute Error**	**TV Mean Absolute Error**	**No TV RMSE**	**TV RMSE**
1.0	0.0153	0.0008	0.0162	0.0019
2.0	0.0678	0.0012	0.0681	0.0043
4.0	0.1218	0.0017	0.1224	0.0093
8.0	0.2668	0.0039	0.2672	0.0248
12.0	0.4576	0.0057	0.4582	0.0355
16.0	0.1213	0.0041	0.1229	0.0226
20.0	0.0177	0.0031	0.0294	0.0187
24.0	-0.0332	0.0053	0.0438	0.0220
28.0	-0.0793	0.0199	0.1031	0.0521

Table 4.2: Yaw comparison vs. velocity.

Velocity	**No TV Peak Yaw Rate**	**TV Peak Yaw Rate**	**No TV Average Yaw Rate**	**TV Average Yaw Rate**
1.0	0.1061	0.1259	0.1017	0.1177
2.0	0.1736	0.2405	0.1601	0.2283
4.0	0.3451	0.5077	0.3331	0.4555
8.0	0.6432	0.9164	0.6305	0.9063
12.0	0.8506	1.3073	0.8421	1.2935
16.0	0.9231	1.0911	0.8622	0.9704
20.0	0.9635	0.9578	0.7696	0.7829
24.0	1.0147	0.9803	0.6894	0.6621
28.0	1.0523	0.9618	0.6420	0.5907

Figure 4.6 represents the longitudinal forces on the rear tires during the manoeuvre. For all velocities, transient oscillations are visible immediately after the steering input, until approximately $t = 0.5$ s, indicating rapid corrective torque actions from the TV controller. After the initial transient, the longitudinal forces diverge progressively.

When talking about the low velocities (1 m/s and 4 m/s), the difference of the longitidinfal forces make the outer wheel with a positive traction force, while the other produces braking force. At high speed (v = 16 m/s), the rear wheel longitudinal forces exhibit a markedly different behaviour. Although a short transient response is still visible immediately after the manoeuvre initiation, the force trajectories rapidly stabilise. Both wheels maintain predominantly positive longitudinal forces, and the torque-vectoring action becomes smoother and more continuous. This indicates that at higher speeds the controller relies on subtler torque redistribution to generate the required yaw moment while preserving vehicle stability and avoiding excessive wheel force oscillations.

The initial force peaks correspond to a short-lived initialization transient associated with the establishment of tire slip and the controller settling from the initial conditions. As the wheel slip did not exceed the traction control activation threshold, no TCS intervention was required.

Overall, the results demonstrate that the torque-vectoring controller adapts its longitudinal force distribution strategy according to vehicle speed. Lower velocities require more aggressive and asymmetric force interventions to achieve the desired yaw response, whereas higher velocities permit smoother and more stable torque allocation between the rear wheels.

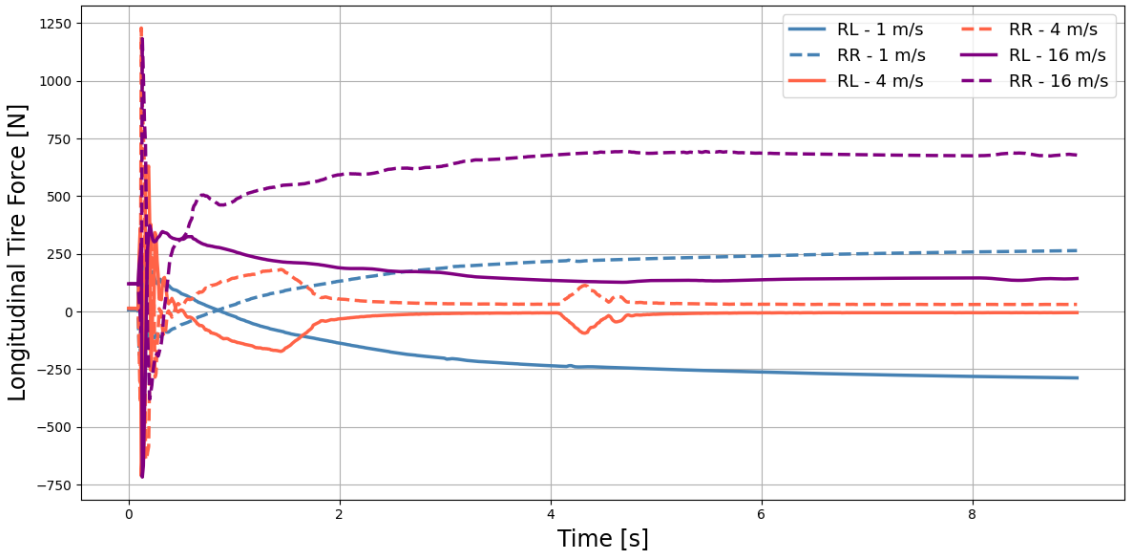


Figure 4.6: Rear wheel longitudinal tire forces for the torque-vectoring configuration at $v = 1\,\mathrm{m/s}$, $v = 4\,\mathrm{m/s}$ and $v = 16\,\mathrm{m/s}$.

In conclusion, in the constant steering test, the torque vectoring controller outperformed the baseline at every speed tested. When the vehicle speed changes, the controller adjusts its torque redistribution strategy accordingly. At low speeds, torque interventions are more pronounced and asymmetric in the longitudinal direction, while at higher speeds, torque is allocated more smoothly and continuously.

In a high-level analysis, these results indicate that the implemented controller not only improves yaw rate tracking accuracy but also enhances the vehicle's responsiveness to driver inputs. One trade-off, however, is reduced damping at the upper end of the speed range.

4.2 Skidpad

The skidpad test was used to evaluate the steady-state cornering performance of the vehicle and to assess the influence of the torque vectoring (TV) controller under different tire-road friction conditions, while also making the closed-loop simulations with the driver. Three scenarios were considered, corresponding to friction coefficients of $\mu = 0.9$, $\mu = 0.6$, and $\mu = 0.3$, representing high-grip, medium-grip and low-grip surfaces, respectively.

The skidpad event is a standard Formula Student dynamic test designed to evaluate the lateral handling characteristics of a vehicle under sustained cornering conditions (more information about this and the other Formula Student events can be found on Appendix C). The track consists of two

interconnected circular sections that require the vehicle to maintain a constant-radius turn while achieving the highest possible cornering speed. This configuration provides a suitable environment for assessing the effectiveness of torque vectoring strategies, as it emphasizes lateral tire force generation, yaw response, and vehicle balance. The geometry of the Formula Student skidpad track used in the simulations is illustrated in Figure 4.7. The final lap time T_{team} is calculated as the average of the second lap of each of the sides [12].

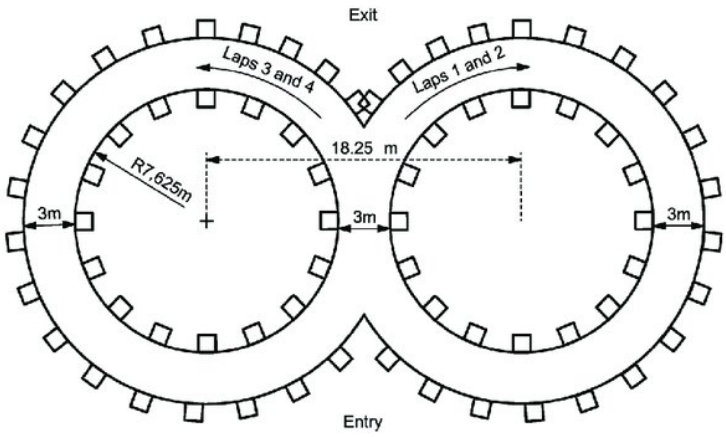


Figure 4.7: Formula Student Skidpad Track, from [76].

Then, the final score of the team is calculated based on the time of the team and the time of the team with the lowest lap time on all event (with penalties, such as cone down or out (DOO)) [12].

$$\text{SCORE} = 47.5 \left(\frac{1.35T_{\text{min}} - T_{\text{team}}}{0.35T_{\text{min}}} \right)^2 + 2.5 \tag{4.1}$$

where:

- SCORE is the score awarded to the team.
- T_{min} is the minimum (fastest) completion time achieved among all teams.
- T_{team} is the completion time achieved by the evaluated team.

For each of the scenarios, two subtests were made, one is called the worst case scenario. This test consists in overriding the driver throttle and using its steering. Firstly in an acceleration part, a higher fixed throttle is given to achieve a desired velocity, then the throttle is fixed to a specific value. This test tries to find the maximum allowed velocity for each combination of friction coefficient and torque distribution configuration that does not end up in an out of the track error.

The second test incorporates all driver inputs to simulate a full lap, comparing the lap time performance of the baseline and torque vectoring configurations over the course of the manoeuvre.

The analysis focuses on three key performance indicators: yaw-rate tracking, lateral acceleration, longitudinal velocity and final lap time. For each friction level, the vehicle behavior with torque vectoring enabled is compared against the baseline case without torque vectoring.

4.2.1 High-Friction Surface

The first scenario uses a friction coefficient $\mu = 0.9$, representing the dry asphalt surface [77], the one that may be most comparable to most of the competitions' friction surfaces.

4.2.1.1 Worst Case Scenario

As shown in Figure 4.8, both vehicles are able to follow the reference yaw-rate profile with reasonable accuracy due to the increased tire grip. Nevertheless, the torque-vectoring controller exhibits a noticeably smaller tracking error, particularly during the transitions between left and right turns and during the steady-state cornering sections. This improvement is confirmed by the average yaw-rate error reported in Table 4.3, which decreases from 0.193 rad/s to 0.036 rad/s.

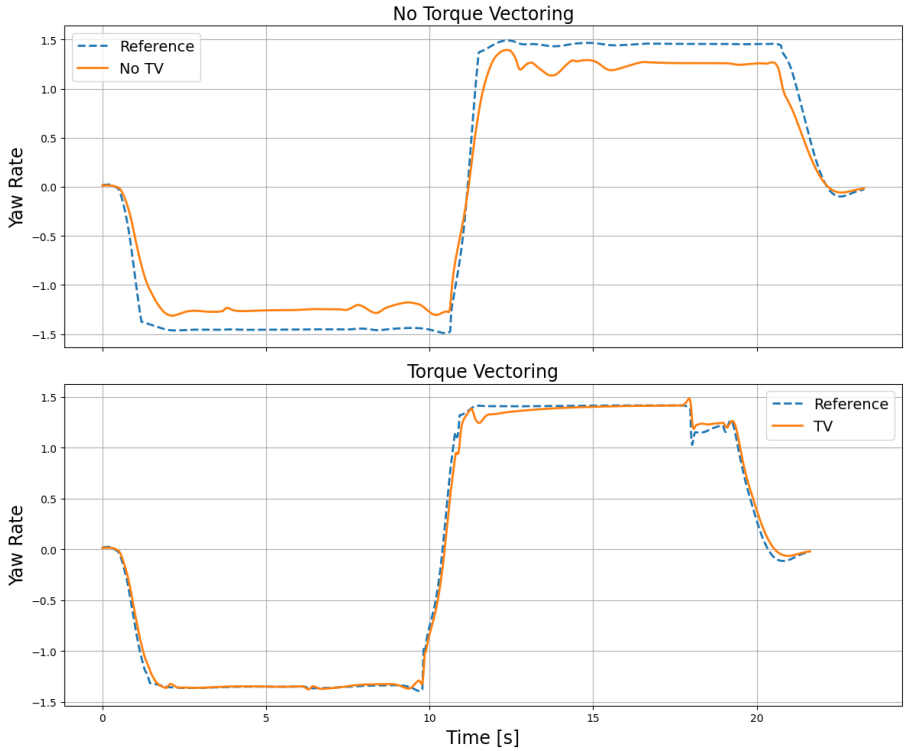


Figure 4.8: Yaw-rate tracking comparison for $\mu = 0.9$.

In figure 4.9, the velocity profile over time is shown for both configurations, the worst case scenario shows the possibility of achieving higher velocities, without being out of the track.

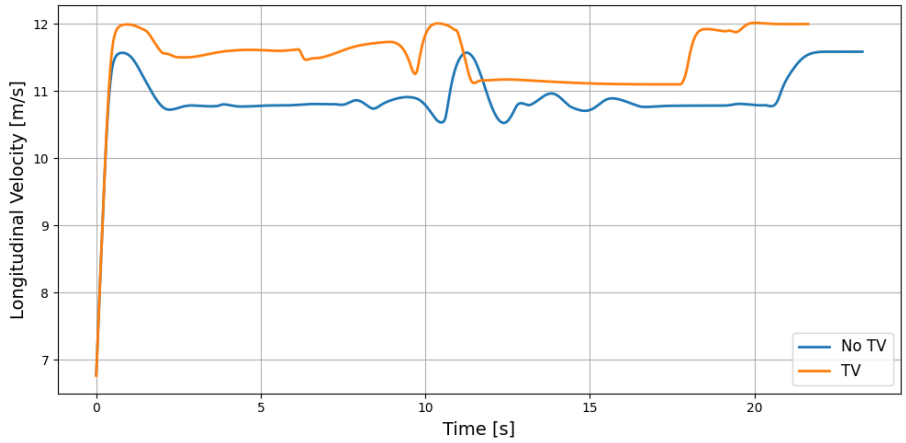


Figure 4.9: Longitudinal velocity over time for TV and baseline implementations and $\mu = 0.9$.

The effect of torque vectoring on cornering performance can be further observed through the lateral acceleration response shown in Figure 4.10.

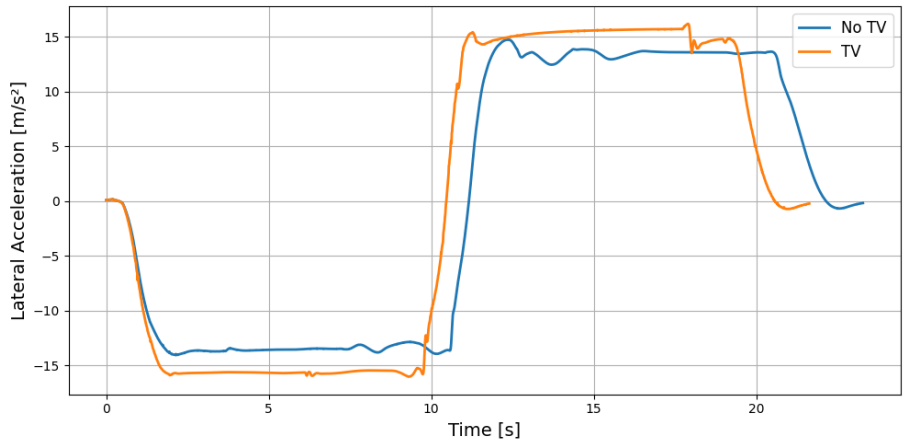


Figure 4.10: Lateral acceleration comparison for $\mu = 0.9$.

Both configurations achieve high lateral acceleration levels, indicating that the available tire grip is being effectively utilized. However, the torque-vectoring vehicle consistently reaches higher lateral acceleration values throughout the maneuver. This behavior suggests that the additional yaw moment generated by the controller allows a more efficient exploitation of the available tire forces. Consequently, the average lateral acceleration increases from 1.291 g to 1.467 g.

The quantitative performance metrics for this scenario are summarized in Table 4.3.

Table 4.3: Effect of torque vectoring on skidpad performance for $\mu = 0.9$.

Metric	**No TV**	**TV**
Average lateral acceleration [g]	1.291	1.467
Peak velocity [m/s]	11.583	12.012

Overall, the results indicate that torque vectoring remains beneficial even on high-friction surfaces. Although the baseline vehicle already demonstrates good handling performance due to the increased tire grip, the torque-vectoring controller further improves yaw-rate tracking accuracy, increases achievable lateral acceleration, and allows the vehicle to maintain slightly higher speeds throughout the maneuver. These improvements demonstrate that the controller enhances both vehicle responsiveness and cornering performance even when operating under favorable road conditions.

4.2.1.2 Driver

When a human driver is placed in the loop, the system behaviour changes considerably compared to the autonomous case. As shown in Figure 4.11, the torque vectoring controller remains robust in tracking the yaw-rate reference despite the increased oscillations introduced by the human driver. The TV response closely follows the reference signal throughout most of the manoeuvre, with noticeable deviations occurring primarily during the transition between the two circular sections of the skidpad.

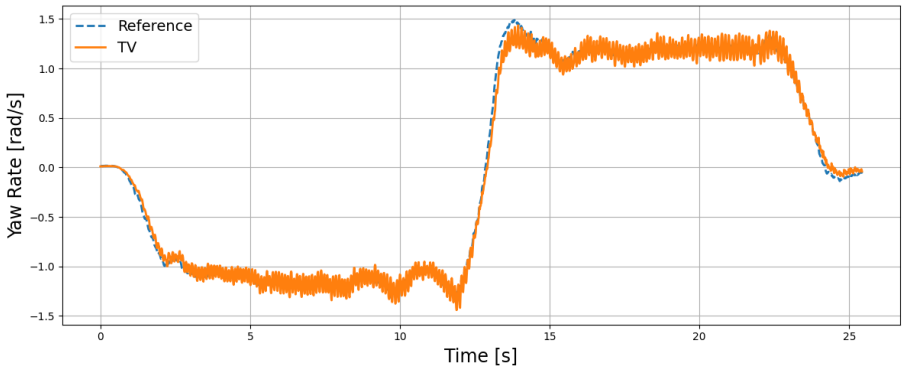


Figure 4.11: Yaw Rate Tracking error over time for TV implementation adn $\mu = 0.9$.

The resulting vehicle trajectory, presented in Figure 4.12, confirms that the driver was able to complete the full figure-of-eight skidpad pattern. The two circular arcs are well-defined and consistent, indicating that the overall path-following behaviour remained stable throughout the run.

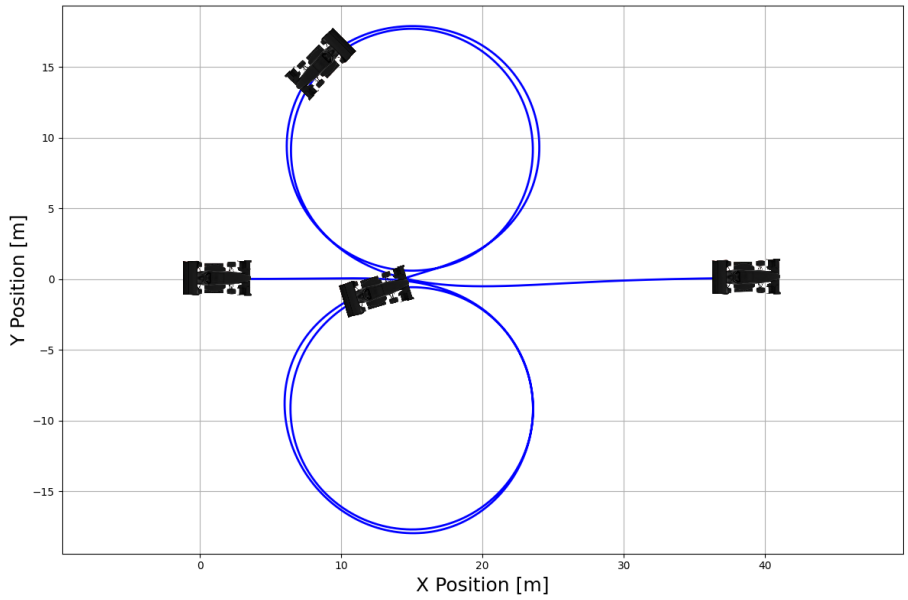


Figure 4.12: TV implementation vehicle trajectory for $\mu = 0.9$.

4.2.2 Medium-Friction Surface

The second set of simulations was performed with a friction coefficient of $\mu = 0.6$. Under these conditions, the available tire forces are limited, making the vehicle more susceptible to understeer and tracking errors during cornering manoeuvres. This friction coefficient could represent surfaces such as wet asphalt [78] or gravel [77].

4.2.2.1 Worst Case Scenario

Figure 4.13 presents the yaw-rate tracking performance for both control strategies. Without torque vectoring, a significant discrepancy between the reference and actual yaw rate can be observed throughout the manoeuvre. The vehicle is unable to generate the required yaw moment, resulting in a reduced yaw response. When torque vectoring is enabled, the measured yaw rate closely follows the reference profile, substantially reducing the tracking error and improving the vehicle's rotational response.

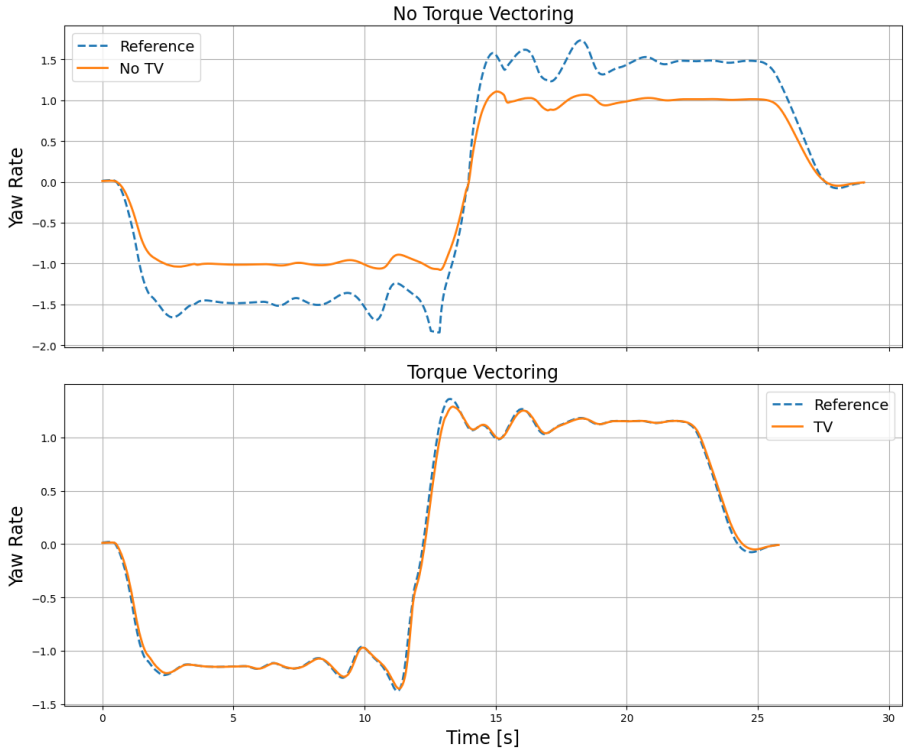


Figure 4.13: Yaw-rate tracking comparison for $\mu = 0.6$.

The longitudinal velocity profiles for both configurations are shown in Figure 4.14. The torque vectoring vehicle maintains a consistently higher speed throughout the manoeuvre, approaching 10 m/s, whereas the baseline vehicle stabilises at approximately 8.6 m/s. This difference highlights the controller's ability to exploit the available friction more effectively, allowing higher cornering speeds without compromising stability.

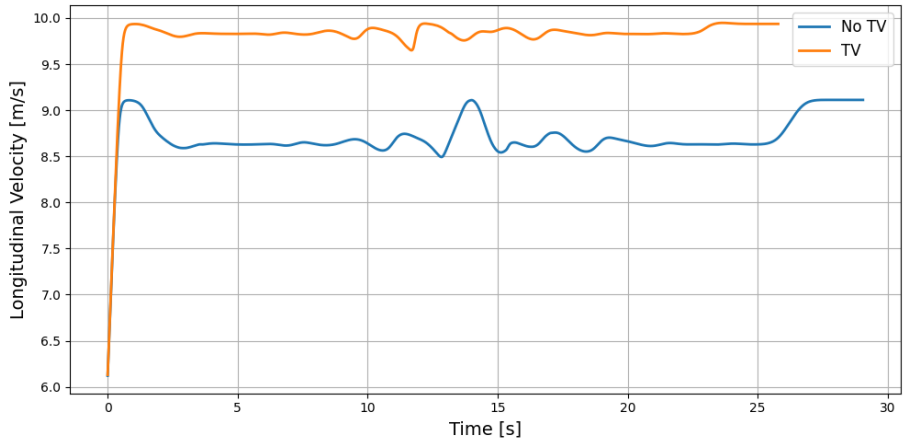


Figure 4.14: Longitudinal velocity over time for TV and baseline implementations and $\mu = 0.6$.

The improvement in yaw-rate tracking is further reflected in the lateral acceleration response shown in Figure 4.15. The torque vectoring controller enables the vehicle to achieve significantly

higher lateral acceleration levels, indicating a greater utilisation of the available tire-road friction. In contrast, the baseline vehicle remains limited by its inherent understeering behaviour, producing considerably lower lateral acceleration values.

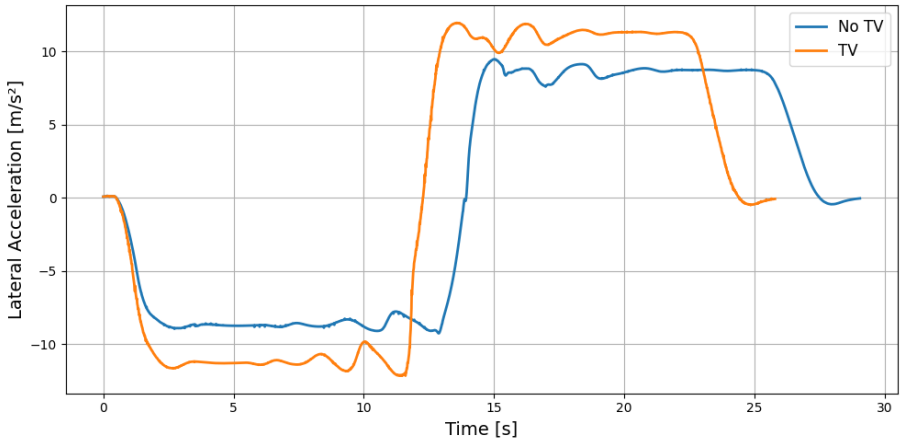


Figure 4.15: Lateral acceleration comparison for $\mu = 0.6$.

The quantitative performance metrics for this scenario are summarised in Table 4.4. The influence of the torque vectoring controller is considerably more pronounced than in the high-grip condition. The average lateral acceleration increased from 0.840 g to 1.075 g, and the peak vehicle velocity increased from 9.113 m/s to 9.948 m/s, confirming the controller's ability to enhance cornering performance under limited adhesion conditions.

Table 4.4: Effect of torque vectoring on skidpad performance for $\mu = 0.6$.

Metric	**No TV**	**TV**
Average lateral acceleration [g]	0.840	1.075
Peak velocity [m/s]	9.113	9.948

Overall, the results obtained for $\mu = 0.6$ demonstrate that torque vectoring provides a substantial improvement in vehicle handling performance under medium-friction conditions. The controller enhances yaw-rate tracking accuracy, increases achievable lateral acceleration, and enables higher cornering speeds while maintaining stability.

4.2.2.2 Driver

When a human driver is placed in the loop under medium-friction conditions, the behaviour observed is consistent with that recorded for $\mu = 0.9$. The yaw-rate tracking performance under driver control is presented in Figure 4.16. The vehicle closely follows the reference yaw-rate profile throughout most of the manoeuvre, with the most notable deviations occurring during the transition between the two circular sections. This behaviour is consistent with the results observed

in the high-friction case, suggesting that the controller maintains its tracking accuracy regardless of the friction condition.

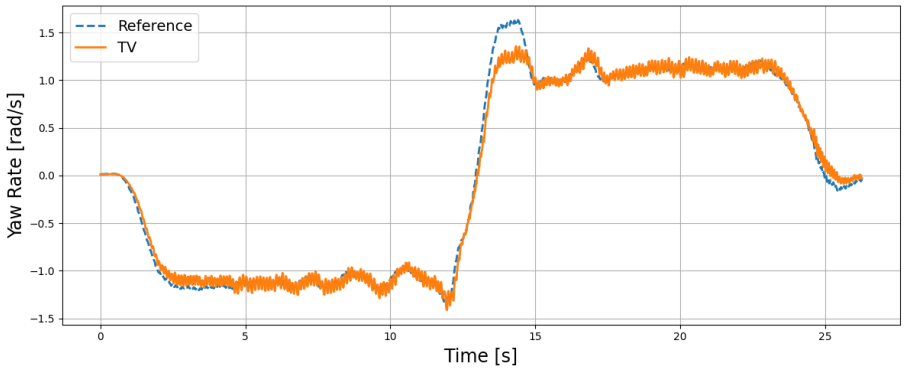


Figure 4.16: Yaw-rate tracking error over time for TV implementation and $\mu = 0.6$.

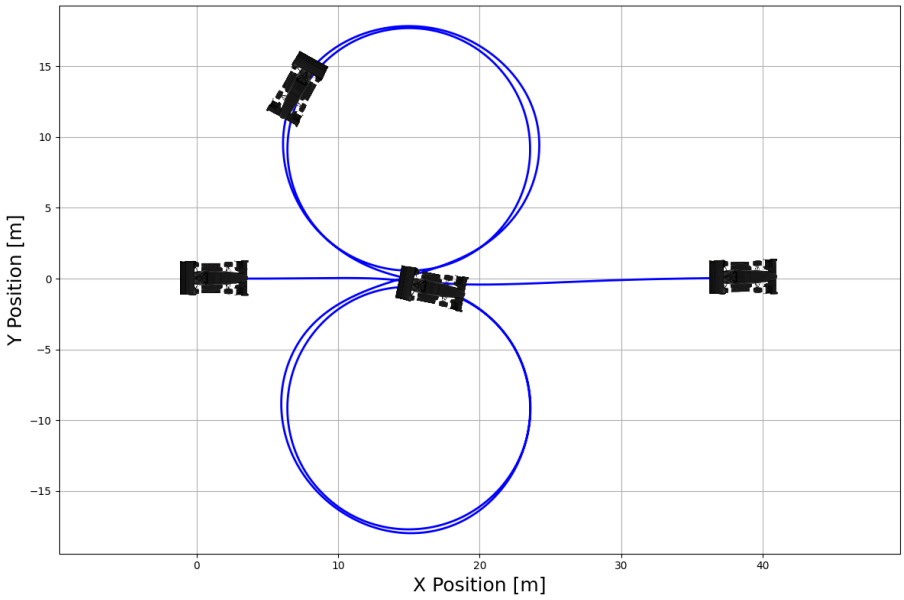


Figure 4.17: TV implementation vehicle trajectory for $\mu = 0.6$.

The resulting vehicle trajectory, presented in Figure 4.17, confirms that the driver was able to complete the full figure-of-eight skidpad pattern under medium-friction conditions. The two circular arcs are well-defined and repeatable, indicating that the torque vectoring controller provided sufficient yaw stabilisation to enable consistent path-following behaviour.

4.2.3 Low-Friction Surface

The last scenario considers a low-friction road surface with a friction coefficient of $\mu = 0.3$, representative of compacted snow road condition [79].

4.2.3.1 Worst Case Scenario

Figure 4.18 compares the yaw-rate tracking performance of the vehicle with and without torque vectoring under severely limited friction conditions.

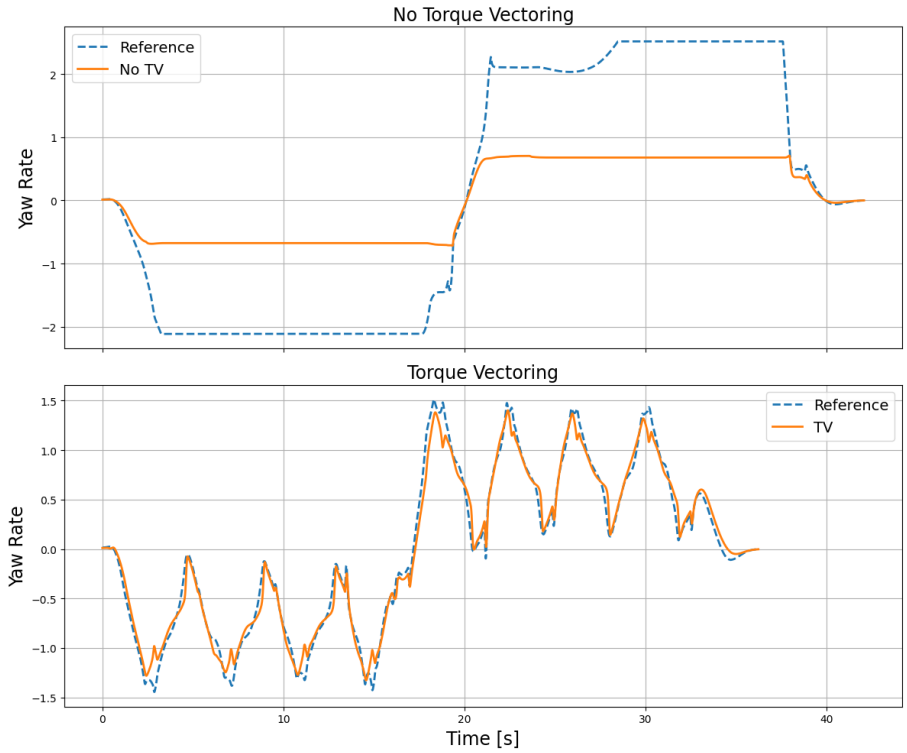


Figure 4.18: Yaw-rate tracking comparison for $\mu = 0.3$.

Without torque vectoring, the vehicle exhibits large steady-state tracking errors and is unable to achieve the commanded yaw-rate levels throughout the manoeuvre. The response saturates at significantly lower yaw-rate values than those demanded by the reference signal, indicating that the available lateral tire forces are insufficient to generate the required yaw moment under such reduced adhesion. When torque vectoring is enabled, the vehicle is able to closely follow the reference yaw-rate profile, with only minor deviations observed around peak responses. Although some oscillations are visible, the overall behaviour remains stable, demonstrating the controller's ability to operate effectively even under severely limited friction conditions.

The longitudinal velocity profiles for both configurations are shown in Figure 4.19. The torque vectoring vehicle displays a markedly different velocity pattern compared to the baseline, exhibiting large periodic oscillations between approximately 5.5 m/s and 8.3 m/s throughout the manoeuvre. These oscillations are attributed to the repeated triggering of the traction control system, which intervenes frequently under the severely limited friction conditions to prevent excessive wheel slip, as shown in Figure 4.20, where the outer rear wheel consistently exhibits the highest slip ratio. However, this behaviour also highlights a limitation of the traction control strategy, as it

is unable to completely stabilize the vehicle's longitudinal velocity under such low-friction conditions, resulting in repeated acceleration and deceleration cycles. In contrast, the baseline vehicle maintains a near-constant speed of approximately 6.2 m/s throughout the manoeuvre.

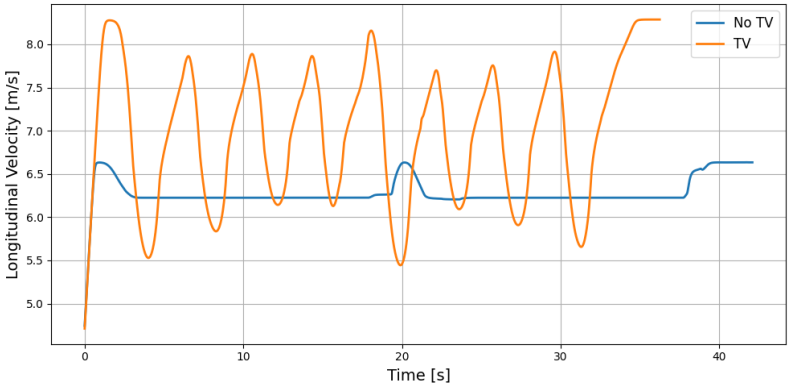
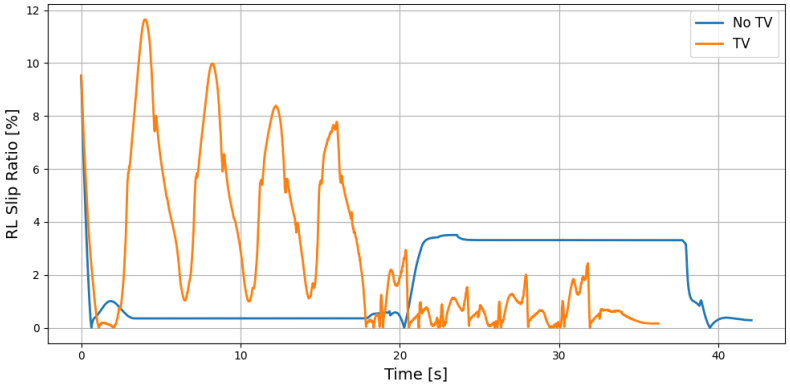
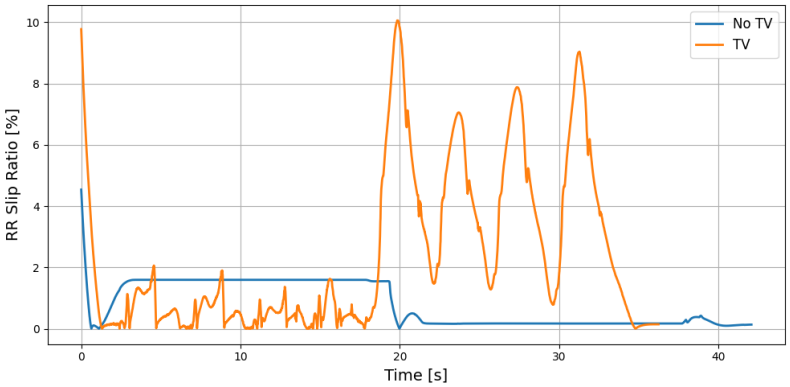


Figure 4.19: Longitudinal velocity over time for TV and baseline implementations and $\mu = 0.3$.



(a) Rear-left wheel slip ratio.



(b) Rear-right wheel slip ratio.

Figure 4.20: Slip ratios of the rear wheels during the skidpad test.

The lateral acceleration response is presented in Figure 4.21. The torque vectoring controller

achieves significantly higher lateral acceleration levels than the baseline, indicating improved utilisation of the available tire forces. The baseline vehicle saturates at considerably lower values, reflecting its inability to generate the required yaw moment. The oscillatory character of the torque vectoring lateral acceleration is consistent with the velocity oscillations observed above, and is a consequence of the more aggressive torque redistribution required at this friction level.

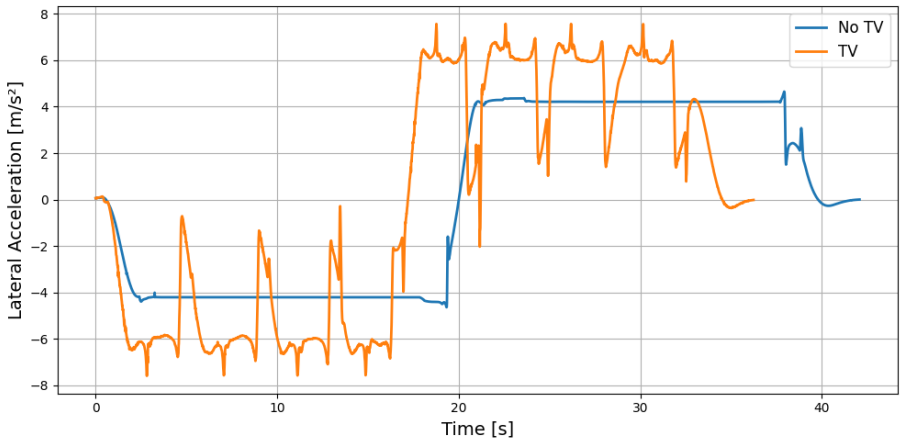


Figure 4.21: Lateral acceleration comparison for $\mu = 0.3$.

The quantitative performance metrics for this scenario are summarised in Table 4.5. The torque vectoring controller achieves a peak velocity of 8.287 m/s compared to 6.635 m/s for the baseline, and increases the average lateral acceleration from 0.422 g to 0.569 g. These results confirm that, even under severely degraded friction conditions, the controller meaningfully enhances the vehicle's cornering performance.

Table 4.5: Effect of torque vectoring on skidpad performance for $\mu = 0.3$.

Metric	**No TV**	**TV**
Average lateral acceleration [g]	0.422	0.569
Peak velocity [m/s]	6.635	8.287

Overall, the results for $\mu = 0.3$ demonstrate that torque vectoring remains effective even under severely limited adhesion conditions, substantially improving yaw-rate tracking accuracy, lateral acceleration, and peak cornering speed. However, the more aggressive torque redistribution required at this friction level introduces notable velocity oscillations, highlighting a trade-off between tracking performance and longitudinal stability at very low friction coefficients.

4.2.3.2 Driver

Under low-friction conditions, the driver-in-the-loop test resulted in a simulation failure, as the vehicle exceeded the track boundaries before completing the full figure-of-eight manoeuvre. This outcome is reflected across all of the following figures and should be interpreted in that context.

The yaw-rate tracking performance, presented in Figure 4.22, reflects the loss of vehicle control. A large and growing discrepancy between the reference and actual yaw rate is observed, with the vehicle unable to generate the lateral forces required to follow the intended path. The simulation was terminated upon detection of the out-of-track condition, which explains the abbreviated time horizon visible in the figure.

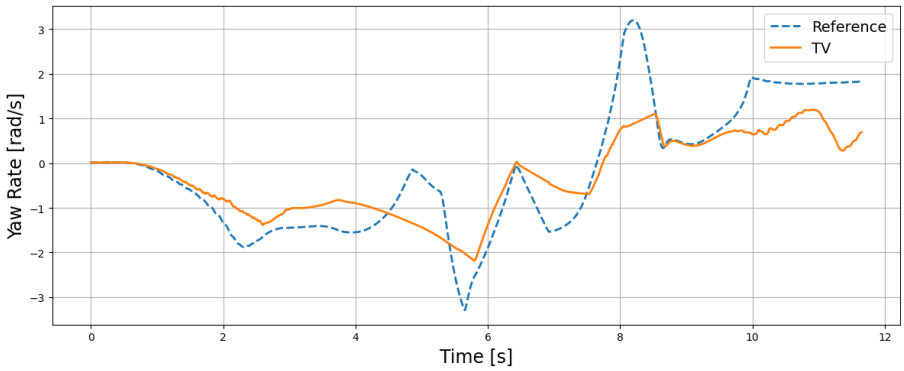


Figure 4.22: Yaw-rate tracking error over time for TV implementation and $\mu = 0.3$.

The resulting trajectory, presented in Figure 4.23, confirms that the vehicle departed the track during the first circular arc, never reaching the transition to the second loop. This result demonstrates that, under a friction coefficient of $\mu = 0.3$, the combination of driver inputs and torque vectoring assistance is insufficient to keep the vehicle within the track limits, highlighting the practical boundaries of the controller under severely degraded adhesion conditions.

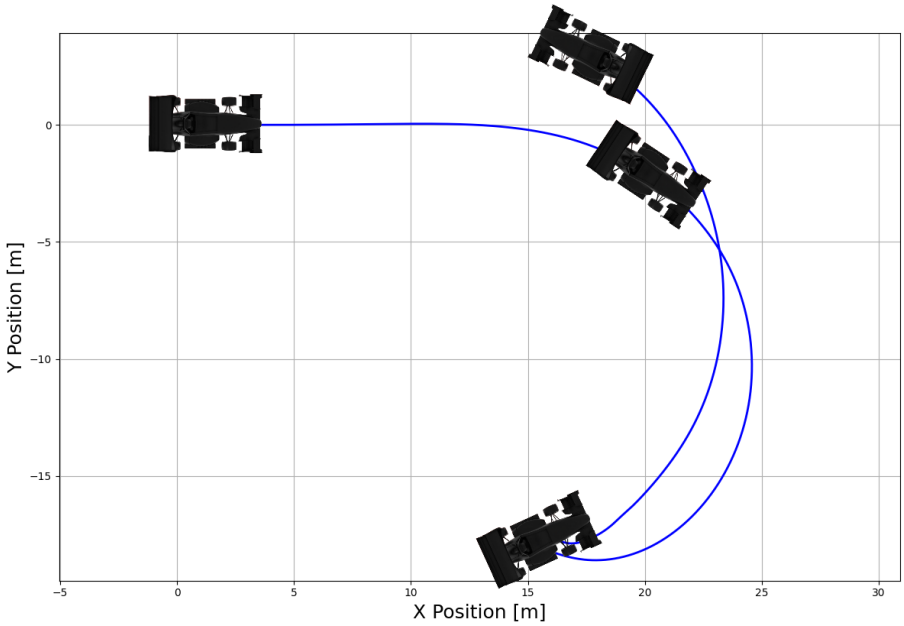


Figure 4.23: TV implementation vehicle trajectory for $\mu = 0.3$.

4.2.4 Summary of Results

Table 4.6 summarizes the lap times obtained during the skidpad simulations for the worst case scenarios. For the low-friction scenario ($\mu = 0.6$), torque vectoring significantly improved performance, reducing the lap time from 6.71 s to 5.93 s, corresponding to an improvement of approximately 11.6%. These results indicate that the benefits of torque vectoring become more pronounced under low-grip conditions, where the controller can effectively compensate for the reduced tire-road adhesion and improve vehicle cornering performance.

Table 4.6: Effect of torque vectoring on skidpad lap time.

Friction Coefficient	**No TV [s]**	**TV [s]**	**Improvement**
$\mu = 0.3$	9.67	8.40	13.1%
$\mu = 0.6$	6.71	5.93	11.6%
$\mu = 0.9$	5.26	5.01	4.75%

At Formula Student Germany 2025, FS FEUP 02 recorded a skidpad lap time of 5.95 s, yielding a score of 2.50 points. Assuming a friction coefficient of $\mu = 0.9$, the simulation results presented in this work suggest that the torque vectoring controller in FS FEUP 03 could reduce the lap time to 5.01 s, which, when applied to the official scoring formula, would correspond to a score of 22.99 points at FSG 2025 [80].

In summary, the benefits of torque vectoring remain evident at high friction levels, although they are less pronounced than in the low-friction scenario. The controller improves yaw-rate tracking accuracy, enhances the smoothness of the lateral acceleration response (for $\mu = 0.6$ and $\mu = 0.9$), and allows a more efficient exploitation of the available grip. The greatest performance gains are therefore observed when tire adhesion becomes the limiting factor, highlighting the effectiveness of torque vectoring in demanding low-grip conditions.

Chapter 5

Conclusion

This dissertation addressed the design, implementation, and simulation-based evaluation of a torque vectoring and traction control architecture for a two-motor Formula Student electric vehicle. Motivated by the development of the next-generation FS FEUP 03 prototype, the work sought to exploit the independent wheel actuation capability of a dual rear-motor drivetrain to improve yaw stability, handling performance, and longitudinal traction under representative Formula Student operating conditions.

The specific objectives defined at the outset of this work were addressed and validated through simulation. In the context of yaw stability and cornering performance, the proposed hierarchical torque vectoring controller demonstrated accurate yaw-rate reference tracking across a wide range of operating conditions. Under steady-state constant-steering tests, the controller consistently reduced mean and root mean squared yaw-rate tracking errors relative to the baseline open-loop configuration, while also achieving higher peak yaw rates at low and intermediate velocities, indicating improved vehicle agility.

The most compelling performance gains were observed under the skidpad evaluations, where three tire-road friction conditions were considered. On a medium-friction surface ($\mu = 0.6$), torque vectoring increased average lateral acceleration by 28.0%, and raised peak cornering speed by 9.2%, resulting in an 11.6% reduction in skidpad lap time. These results confirm that the controller is particularly effective when tire adhesion is the limiting factor, precisely the scenario in which active yaw moment generation provides the greatest benefit. Under severely degraded adhesion ($\mu = 0.3$), the controller continued to improve yaw-rate tracking, lateral acceleration, and peak cornering speed, albeit at the cost of increased longitudinal velocity oscillations due to frequent traction control intervention; the driver-in-the-loop test at this friction level also revealed that the combination of driver inputs and controller assistance was insufficient to keep the vehicle within track limits, highlighting the practical boundaries of the system under extreme conditions. On a high-friction surface ($\mu = 0.9$), the lap time improvement was more modest (4.75%, from 5.26 s to 5.01 s), but yaw-rate tracking accuracy was still markedly improved, confirming that the benefits of the controller extend to smoother and more controllable vehicle behaviour under high-grip scenarios as well.

From a software architecture perspective, the controller was implemented as a ROS2 node and validated within an IPG CarMaker simulation environment using a vehicle model representative of the FS FEUP 03 prototype. The use of ROS2 middleware ensures direct compatibility with the autonomous system architecture of the vehicle, minimising the integration effort required for future deployment on the real hardware platform. The computational structure of the hierarchical controller was designed to remain within the processing constraints of a Formula Student control unit, addressing the feasibility requirement for real-time implementation.

5.1 Limitations

Despite the positive results, this work is subject to a number of limitations that should be acknowledged. The evaluation was conducted exclusively in simulation, and the controller has not yet been validated on the physical FS FEUP 03 vehicle. Although the simulation environment models nonlinear tire dynamics and motor characteristics, discrepancies between the simulated and real vehicle behaviour are inevitable and may affect the generalisability of the presented results.

Additionally, the traction control layer employs a simplified slip controller without explicit estimation of the road friction coefficient. While this approach was shown to be effective under the low-friction conditions, it exhibited oscillatory behaviour, indicating that the controller requires further tuning or a more advanced slip control strategy for such conditions. Consequently, its robustness under a broader range of surface types and combined longitudinal-lateral loading scenarios remains to be evaluated.

5.2 Future Work

Several directions for future work arise naturally from the findings and limitations of this dissertation. The most immediate priority is the deployment and experimental validation of the developed controller on the FS FEUP 03 prototype. Real-vehicle testing will be essential to assess controller performance under real sensor noise, actuator delays, and the variability of competition track surfaces, and will provide the empirical evidence needed to complement the simulation results presented here.

The traction control architecture could be extended to incorporate online road surface estimation, enabling more adaptive slip-ratio thresholds and improving controller robustness under unknown or varying friction conditions. Integration of combined slip control, accounting for the coupling between lateral and longitudinal tire forces, would further enhance stability during simultaneous acceleration and cornering.

Finally, as Formula Student increasingly incorporates autonomous driving challenges, the integration of the torque vectoring strategy with a higher-level path-tracking or trajectory optimisation module represents a promising direction. By closing the loop between path following and vehicle

dynamics control, the combined system could allow the autonomous vehicle to exploit the full performance potential of the dual-motor architecture in both dynamic and steady-state competition scenarios.

In summary, this dissertation demonstrated that a hierarchical torque vectoring and traction control architecture can substantially improve the yaw stability, cornering performance, and longitudinal traction of a two-motor electric Formula Student vehicle within a simulation environment. The results provide a strong foundation for the forthcoming experimental validation on the FS FEUP 03 platform and contribute to the broader effort of advancing software-defined vehicle dynamics control in the context of competitive electric motorsport.

References

[1] International Energy Agency. *Global EV Outlook 2026*. Accessed: 2026-06-02. Paris, France: International Energy Agency, 2026. URL: https://www.iea.org/reports/global-ev-outlook-2026.

[2] Y. Hori. “Future vehicle driven by electricity and Control-research on four-wheel-motored "UOT electric march II"”. In: *IEEE Transactions on Industrial Electronics* 51.5 (2004), pp. 954–962. DOI: 10.1109/TIE.2004.834944.

[3] Anton Zanten and Robert Bosch. “Evolution of Electronic Control Systems for Improving the Vehicle Dynamic Behavior”. In: *Proc. Int. Symp. Adv. Vehicle Control (AVEC* (Jan. 2002).

[4] Rajesh Rajamani. *Vehicle Dynamics and Control*. 2nd. Vol. 14. Mechanical Engineering Series. Springer US, 2012. ISBN: 978-1-4614-1432-2. DOI: 10.1007/978-1-4614-1433-9.

[5] Alena Erke. “Effects of electronic stability control (ESC) on accidents: A review of empirical evidence”. In: *Accident Analysis Prevention* 40.1 (2008), pp. 167–173. ISSN: 0001-4575. DOI: https://doi.org/10.1016/j.aap.2007.05.002. URL: https://www.sciencedirect.com/science/article/pii/S0001457507000851.

[6] Valentina De Pascale et al. “Torque Vectoring Control for fully electric Formula SAE cars”. In: Sept. 2019.

[7] Henrique de Carvalho Pinheiro and Massimiliana Carello. “System-Level Performance of Torque Vectoring Controllers for Electric Vehicles with In-Wheel Motors”. In: *SAE International Journal of Vehicle Dynamics, Stability, and NVH* 10.4 (Mar. 2026). ISSN: 2380-2162. DOI: https://doi.org/10.4271/10-10-04-0029. URL: https://doi.org/10.4271/10-10-04-0029.

[8] Andrea Fricano. “Design of Torque Vectoring System for Formula SAE Electric Vehicle”. Corso di laurea magistrale in Ingegneria Meccanica. Master’s thesis. Torino, Italy: Politecnico di Torino, — 2019. URL: https://webthesis.biblio.polito.it/12157/.

[9] Mia Grahovic and Madeleine Rosicki. “Development and Evaluation of a Torque-Vectoring Algorithm on RWD Racing Cars using a Dual Clutch”. TFRT-6081, ISSN 0280-5316. Master’s thesis. Lund, Sweden: Lund University, Department of Automatic Control, — 2019. URL: https://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=8987522&fileOId=8987524.

[10] 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”. In: *Meccanica* 56.5 (May 2021), pp. 991–1010. DOI: 10.1007/s11012-021-01317-3.

[11] João Pedro Marques Antunes. "Torque Vectoring for a Formula Student Prototype". Master's thesis. Lisbon, Portugal: Instituto Superior Técnico, Universidade de Lisboa, June 2017.

[12] Formula Student Germany. *Formula Student Germany Rules 2026*. Accessed: April 2026. Formula Student Germany e.V. 2026. URL: https://www.formulastudent.de/fileadmin/user_upload/all/2026/rules/FS-Rules_2026_v1.1.pdf.

[13] Eloi Morera Torres. *Real-Data-Based Modelling and Torque Vectoring Algorithm for a 4-Wheel-Drive Formula Student Vehicle*. Bachelor's thesis. Treball Final de Grau. July 2020. URL: https://upcommons.upc.edu/entities/publication/9252c327-6062-4d04-87da-0c278823bd09.

[14] Huifan Deng et al. "Deep Reinforcement Learning-Based Torque Vectoring Control Considering Economy and Safety". In: *Machines* 11.4 (2023). ISSN: 2075-1702. DOI: 10.3390/machines11040459. URL: https://www.mdpi.com/2075-1702/11/4/459.

[15] Jianyuan Liu et al. "A Reconfigurable Dual-Motor Compound-Planetary Electric Drive Axle for an Expanded Torque-Vectoring Envelope". In: *Actuators* 15.5 (2026). ISSN: 2076-0825. URL: https://www.mdpi.com/2076-0825/15/5/268.

[16] H. Fujimoto et al. "Motion control and road condition estimation of electric vehicles with two in-wheel motors". In: *Proceedings of the 2004 IEEE International Conference on Control Applications, 2004.* Vol. 2. 2004, 1266–1271 Vol.2. DOI: 10.1109/CCA.2004.1387547.

[17] IPG Automotive GmbH. *IPG Automotive GmbH – Index Page*. https://www.ipg-automotive.com/en/. Accessed on 21 January 2026. 2026. URL: https://www.ipg-automotive.com/en/.

[18] Mohamed Belrzaeg et al. "Vehicle dynamics and tire models: An overview". In: *World Journal of Advanced Research and Reviews* 12 (Oct. 2021), pp. 331–348. DOI: 10.30574/wjarr.2021.12.1.0524.

[19] International Organization for Standardization. *Road vehicles - Vehicle dynamics and road-holding ability - Vocabulary*. Standard ISO 8855:2011. Geneva, Switzerland: International Organization for Standardization, Dec. 2011.

[20] Moad Kissai et al. "Adaptive Robust Vehicle Motion Control for Future Over-Actuated Vehicles". In: *Machines* 7 (Apr. 2019), p. 26. DOI: 10.3390/machines7020026.

[21] Hans B. Pacejka. "Chapter 4 - Semi-Empirical Tire Models". In: *Tire and Vehicle Dynamics (Third Edition)*. Ed. by Hans B. Pacejka. Third Edition. Oxford: Butterworth-Heinemann, 2012, pp. 149–209. ISBN: 978-0-08-097016-5. DOI: https://doi.org/10.1016/B978-0-08-097016-5.00004-8. URL: https://www.sciencedirect.com/science/article/pii/B9780080970165000048.

[22] Derek Seward. *Race Car Design*. London: Palgrave, 2014. ISBN: 9781137030146.

[23] "Aerodynamics of race cars". In: *Annu. Rev. Fluid Mech* 38 (Jan. 2006), pp. 27–63. DOI: 10.1146/annurev.fluid.38.050304.092016.

[24] Kanwar Singh and Srikanth Sivaramakrishnan. "Extended Pacejka Tire Model for Enhanced Vehicle Stability Control". In: (May 2023). DOI: 10.48550/arXiv.2305.18422.

[25] C. Canudas de Wit and P. Tsiotras. "Dynamic tire friction models for vehicle traction control". In: *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304)*. Vol. 4. 1999, 3746–3751 vol.4. DOI: 10.1109/CDC.1999.827937.

[26] Carlos Canudas de Wit et al. “A new model for control of systems with friction”. In: *IEEE Trans. Autom. Control.* 40 (1995), pp. 419–425. URL: https://api.semanticscholar.org/CorpusID:124774219.

[27] R. Brach and R. Brach. *Tire Models for Vehicle Dynamic Simulation and Accident Reconstruction*. Tech. rep. 2009-01-0102. SAE Technical Paper, 2009. DOI: 10.4271/2009-01-0102. URL: https://doi.org/10.4271/2009-01-0102.

[28] Shu-en Zhao, Yuling Li, and Xian Qu. “Vehicle Chassis Integrated Control Based on Multimodel and Multilevel Hierarchical Control”. In: *Mathematical Problems in Engineering* 2014 (Apr. 2014), pp. 1–13. DOI: 10.1155/2014/248676.

[29] I. B. A. op het Veld. “Enhancing the MF-Swift tyre model for inflation pressure changes”. MA thesis. Eindhoven University of Technology, Jan. 2008.

[30] Masahiko Mizuno et al. “Development of a tyre force model incorporating the influence of the tyre surface temperature”. In: *Vehicle System Dynamics - VEH SYST DYN* 43 (Jan. 2005), pp. 395–402. DOI: 10.1080/00423110500140187.

[31] Dao-Hu Wang and Wei-Hsin Liao. “Magnetorheological fluid dampers: A review of parametric modelling”. In: *Smart Materials and Structures* 20 (Jan. 2011), p. 023001. DOI: 10.1088/0964-1726/20/2/023001.

[32] James Balkwill. “Chapter 5 - Cornering”. In: *Performance Vehicle Dynamics*. Ed. by James Balkwill. Butterworth-Heinemann, 2018, pp. 95–179. ISBN: 978-0-12-812693-6. DOI: https://doi.org/10.1016/B978-0-12-812693-6.00005-5. URL: https://www.sciencedirect.com/science/article/pii/B9780128126936000055.

[33] William F. Milliken and Douglas L. Milliken. *Race Car Vehicle Dynamics*. Warrendale, PA: Society of Automotive Engineers, 1995. ISBN: 978-1-56091-526-3.

[34] Janick Frasch et al. “An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles”. In: July 2013, pp. 4136–4141. DOI: 10.23919/ECC.2013.6669836.

[35] Claude Rouelle. *Getting More from Your Yaw Diagrams*. http://downloads.optimumg.com/Technical_Papers/RCE3.pdf. Accessed: 2025-10-03. n.d.

[36] Christopher Carlson and J Gerdes. “Optimal Rollover Prevention With Steer by Wire and Differential Braking”. In: vol. 72. Nov. 2003. DOI: 10.1115/IMECE2003-41825.

[37] Ikkei Kobayashi et al. “Research on Yaw Moment Control System for Race Cars Using Drive and Brake Torques”. In: *Vehicles* 5.2 (2023), pp. 515–534. ISSN: 2624-8921. DOI: 10.3390/vehicles5020029. URL: https://www.mdpi.com/2624-8921/5/2/29.

[38] M. Abe and W. Manning. “Vehicle Handling Dynamics”. In: *Vehicle Handling Dynamics* (Jan. 2009). DOI: 10.1016/B978-1-85617-749-8.X0001-3.

[39] Rama Vadapalli, Nagamanikandan Govindan, and Madhava Krishna. “Design And Analysis Of Three-Output Open Differential with 3-DOF”. In: (Dec. 2021). DOI: 10.48550/arXiv.2112.11254.

[40] Mirko Karakašić et al. “Influence of Open Differential Design on the Mass Reduction Function”. In: *Applied Sciences* 13.24 (2023). ISSN: 2076-3417. DOI: 10.3390/app132413300. URL: https://www.mdpi.com/2076-3417/13/24/13300.

[41] Henrique de Carvalho Pinheiro, Massimiliana Carello, and Elisabetta Punta. "Torque Vectoring Control Strategies Comparison for Hybrid Vehicles with Two Rear Electric Motors". In: *Applied Sciences* 13.14 (2023). ISSN: 2076-3417. DOI: 10.3390/app13148109. URL: https://www.mdpi.com/2076-3417/13/14/8109.

[42] Subhajit Konar and Vijay Gautam. "Design And Analysis Of An Open Differential". In: *INTERNATIONAL JOURNAL OF ADVANCED PRODUCTION AND INDUSTRIAL ENGINEERING* 4 (Apr. 2019).

[43] Marco Gadola and Daniel Chindamo. "The Mechanical Limited-Slip Differential Revisited: High-Performance and Racing Car Applications". In: *International Journal of Applied Engineering Research* 13.2 (2018), pp. 1478–1495.

[44] Drexler Automotive GmbH. *M-Diff Formula Student V2 Adjustable*. 2025. URL: https://www.drexler-automotive.com/en/products/m-diff-formula-student-v2 (visited on 06/02/2026).

[45] Valentin Ivanov, Dzmitry Savitski, and Barys Shyrokau. "A Survey of Traction Control and Antilock Braking Systems of Full Electric Vehicles With Individually Controlled Electric Motors". In: *IEEE Transactions on Vehicular Technology* 64.9 (2015), pp. 3878–3896. DOI: 10.1109/TVT.2014.2361860.

[46] Pushkin Kachroo and Masayoshi Tomizuka. "Vehicle Traction Control And its Applications". In: *Institute of Transportation Studies, UC Berkeley, Institute of Transportation Studies, Research Reports, Working Papers, Proceedings* (Jan. 1994).

[47] Kenta Maeda, Hiroshi Fujimoto, and Yoichi Hori. "Four-wheel driving-force distribution method for instantaneous or split slippery roads for electric vehicle with in-wheel motors". In: *2012 12th IEEE International Workshop on Advanced Motion Control (AMC)*. 2012, pp. 1–6. DOI: 10.1109/AMC.2012.6197053.

[48] Fredrik Gustafsson. "Slip-based tire-road friction estimation". In: *Automatica* 33.6 (1997), pp. 1087–1099. ISSN: 0005-1098. DOI: https://doi.org/10.1016/S0005-1098(97)00003-4. URL: https://www.sciencedirect.com/science/article/pii/S0005109897000034.

[49] Toshiaki Tsuyama et al. "Traction control system using estimated road surface friction". Patent US5353225A. Oct. 1994. URL: https://patents.google.com/patent/US5353225A/en.

[50] Michele Asperti, Michele Vignati, and Edoardo Sabbioni. "On Torque Vectoring Control: Review and Comparison of State-of-the-Art Approaches". In: *Machines* 12.3 (2024). ISSN: 2075-1702. DOI: 10.3390/machines12030160. URL: https://www.mdpi.com/2075-1702/12/3/160.

[51] Reza Jafari et al. "Reinforcement Learning for Torque Vectoring in Electric Vehicles: A Review of Stability and Energy Optimization Methods". In: *IEEE Open Journal of Vehicular Technology* 7 (2026), pp. 354–380. DOI: 10.1109/OJVT.2025.3638680.

[52] Leonardo De Novellis et al. "Direct yaw moment control actuated through electric drivetrains and friction brakes: Theoretical design and experimental assessment". In: *Mechatronics* 26 (2015), pp. 1–15. ISSN: 0957-4158. DOI: https://doi.org/10.1016/j.mechatronics.2014.12.003. URL: https://www.sciencedirect.com/science/article/pii/S0957415814002104.

[53] Qian Lu et al. “H loop shaping for the torque-vectoring control of electric vehicles: Theoretical design and experimental assessment”. In: *Mechatronics* 35 (2016), pp. 32–43. ISSN: 0957-4158. DOI: https://doi.org/10.1016/j.mechatronics.2015.12.005. URL: https://www.sciencedirect.com/science/article/pii/S0957415815002214.

[54] Mayuresh Mutha et al. “Influence of vehicle parameters on handling characteristics and its control using torque vectoring”. In: *IOP Conference Series: Materials Science and Engineering* 624 (Oct. 2019), p. 012015. DOI: 10.1088/1757-899X/624/1/012015.

[55] Thomas D. Gillespie. *Fundamentals of Vehicle Dynamics*. Revised. Warrendale, PA: SAE International, 2021. ISBN: 978-1-4686-0176-3. DOI: 10.4271/9781468601770.

[56] Karl Johan Åström and Tore Hägglund. *Advanced PID Control*. ISA, 2006.

[57] Zhengyuan Wang et al. “A gain scheduled robust linear quadratic regulator for vehicle direct yaw moment Control”. In: *Mechatronics* 51 (2018), pp. 31–45. ISSN: 0957-4158. DOI: https://doi.org/10.1016/j.mechatronics.2018.01.013. URL: https://www.sciencedirect.com/science/article/pii/S0957415818300138.

[58] Kyoungseok Han et al. “Model Predictive Control Framework for Improving Vehicle Cornering Performance Using Handling Characteristics”. In: *IEEE Transactions on Intelligent Transportation Systems* 22.5 (2021), pp. 3014–3024. DOI: 10.1109/TITS.2020.2978948.

[59] Erik Mikuláš, Martin Gulan, and Gergely Takács. “Model Predictive Torque Vectoring Control for a Formula Student Electric Racing Car”. In: *2018 European Control Conference (ECC)*. 2018, pp. 581–588. DOI: 10.23919/ECC.2018.8550124.

[60] Hongqian Wei et al. “Deep reinforcement learning based active safety control for distributed drive electric vehicles”. In: *IET Intelligent Transport Systems* 16 (Mar. 2022). DOI: 10.1049/itr2.12176.

[61] Reza Jafari et al. “A TD3-Based Reinforcement Learning Algorithm With Curriculum Learning for Adaptive Yaw Control in All-Wheel-Drive Electric Vehicles”. In: *IEEE Access* 13 (2025), pp. 127150–127169. DOI: 10.1109/ACCESS.2025.3587938.

[62] Reza Jafari et al. “Optimal Torque Allocation for All-Wheel-Drive Electric Vehicles Using a Reinforcement Learning Algorithm”. In: *2024 13th Mediterranean Conference on Embedded Computing (MECO)*. 2024, pp. 1–5. DOI: 10.1109/MECO62516.2024.10577860.

[63] Huifan Deng et al. “Fault-Tolerant Predictive Control With Deep-Reinforcement-Learning-Based Torque Distribution for Four In-Wheel Motor Drive Electric Vehicles”. In: *IEEE/ASME Transactions on Mechatronics* 28.2 (2023), pp. 668–680. DOI: 10.1109/TMECH.2022.3233705.

[64] Henrique de Carvalho Pinheiro and Massimiliana Carello. “Reinforcement learning based control for torque allocation in electric vehicles: a preliminary analysis”. In: *2024 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA)*. 2024, pp. 1–6. DOI: 10.1109/ACDSA59508.2024.10467491.

[65] Hongqian Wei et al. “Deep reinforcement learning based direct torque control strategy for distributed drive electric vehicles considering active safety and energy saving performance”. In: *Energy* 238 (2022), p. 121725. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2021.121725. URL: https://www.sciencedirect.com/science/article/pii/S0360544221019733.

[66] Steven Macenski et al. “Robot Operating System 2: Design, architecture, and uses in the wild”. In: *Science Robotics* 7.66 (2022), eabm6074. DOI: 10.1126/scirobotics.abm6074. URL: https://www.science.org/doi/abs/10.1126/scirobotics.abm6074.

[67] The MathWorks Inc. *MATLAB version: 9.14.0 (R2023a)*. Natick, Massachusetts, United States, 2023. URL: https://www.mathworks.com.

[68] Robert C. Martin. *Agile Software Development, Principles, Patterns, and Practices*. Prentice Hall, 2002.

[69] Edward M Kasprzak and David Gentz. “The Formula SAE Tire Test Consortium–Tire Testing and Data Handling”. In: SAE Technical Paper. 2006. DOI: 10.4271/2006-01-3606.

[70] IPG Automotive GmbH. *Vehicle Dynamics*. Technical brochure. Accessed: 12 June 2026. 2019. URL: https://images.next-move.fr/images/IPG_Automotive_Vehicle_Dynamics_ab1b7232df.pdf.

[71] DTI (Dynamic Systems). *DTI Formula Student Motor (F-MOT) Datasheet, V1.5*. Accessed: May 2026. 2026. URL: https://www.dti.dk/.

[72] *IPGDriver User Manual*. Version 13.1.1. IPG Automotive GmbH. 2024.

[73] U Kiencke and L Nielsen. “Automotive Control Systems: For Engine, Driveline, and Vehicle”. In: *Measurement Science and Technology* 11 (Nov. 2000), p. 1828. DOI: 10.1088/0957-0233/11/12/708.

[74] Martin Olofsson and Jens Pettersson. “Parameterization and Validation of Road and Driver Behavior Models for CarMaker Simulations and Transmission HIL-Rig”. In: 2015. URL: https://api.semanticscholar.org/CorpusID:106800122.

[75] ISO. *Passenger cars — Steady-state circular driving behaviour — Open-loop test methods*. Standard ISO 4138:2021. International Organization for Standardization, Sept. 2021. URL: https://www.iso.org/standard/81710.html.

[76] Krzysztof Ziemba et al. “Validation of Key Parameters Between Real and Computer Model of Suspension System Used in FSAE Vehicle”. In: *Advances in Science and Technology Research Journal* 14 (Sept. 2020), pp. 65–74. DOI: 10.12913/22998624/122568.

[77] Andreas Andersson. “Road-Tire Friction Estimation for AFS Vehicle Control”. In: 2006. URL: https://api.semanticscholar.org/CorpusID:114674056.

[78] Jiye Huang et al. “A Fusion Estimation Method for Tire-Road Friction Coefficient Based on Weather and Road Images”. In: *Lubricants* 13.10 (2025). ISSN: 2075-4442. DOI: 10.3390/lubricants13100459. URL: https://www.mdpi.com/2075-4442/13/10/459.

[79] Kaoru Ichihara and Makoto Mizoguchi. “Skid Resistance of Snow- or Ice-Covered Roads”. In: *Snow Removal and Ice Control Research*. Vol. 115. Highway Research Board Special Report. Washington, D.C.: Highway Research Board, 1970, pp. 104–114.

[80] Formula Student Germany. *Results FSG 2025*. 2025. URL: https://www.formulastudent.de/fsg/results/2025 (visited on 06/13/2026).

Appendix A

Motor Information

2. TECHNICAL SPECIFICATIONS

Parameters provided by Fischer Elektromotoren.

2.1 Motor peak and nominal condition

Table 1. motor peak and nominal specifications

	F-MOT	Unit
Peak Torque[1]	31,6	Nm
Peak Current[2]	85	Arms
Maximum Speed	20.000	rpm
Peak Power[3]	35	kW
Nominal Torque	11,1	Nm
Nominal Current	22,6	Arms
Nominal Speed	13250	rpm
Nominal Power	15,4	Kw

2.2 Motor basic parameters

Table 2. basic specifications

	F-MOT	Unit
Torque Constant	0,492	Nm/Arms
BEMF Constant	0,031	Vrms/rpm
Maximum DC bus voltage	600	Vdc
Pole pairs	4	
Temperature sensor[4]	PT1000	
Position sensor type	TE connectivity 2-1393047-5	
Peak efficiency	>96	%
Phase resistance at 20 °C[5]	0,126	Ω
Phase inductance[6]	0,393	mH

[1] Corrected to DTI measured value
[2] Corrected to DTI measured value
[3] Measured at 10.500 rpm
[4] Isolation voltage 2kV
[5] Measured resistance per phase
[6] Measurend inductance per phase

Figure A.1: DTI F-MOT motor datasheet [71].

Appendix B

C++ Implementations

B.1 Steady-State Yaw Rate Reference Generator

Listing B.1: Implementation of the steady-state yaw rate reference generator.

```
#include "reference_generators/yaw_rate_reference_generators/
    steady_state_yaw_rate_reference_generator.hpp"

SteadyStateYawRateReferenceGenerator::SteadyStateYawRateReferenceGenerator
    (
                                                const CarParameters&
                                                    car_params,
                                                bool override_understeer,
                                                double
                                                    understeer_coefficient)
    : car_parameters_(car_params) {

    if (override_understeer) {
        understeer_coefficient_ = understeer_coefficient;
    } else {
        double Cf = car_parameters_.cornering_stiffness_front;
        double Cr = car_parameters_.cornering_stiffness_rear;
        double lf = car_parameters_.front_axle_distance;
        double lr = car_parameters_.rear_axle_distance;
        double m = car_parameters_.mass;
        understeer_coefficient_ = m / (lr + lf) * (lr / Cf - lf / Cr);
    }
}

double SteadyStateYawRateReferenceGenerator::generate_reference_yaw_rate(
                                            VehicleState state,
                                            DriverCommand command) {

```

```
25      double L = car_parameters_.front_axle_distance +
26                 car_parameters_.rear_axle_distance;
27      double V = state.velocity_x;
28      double delta = command.steering;
29
30      if (V < 0.001) {
31          return 0.0;
32      }
33
34      double den = L + understeer_coefficient_ * V * V;
35      double yaw_rate_ref = V / den * delta;
36
37      double a_y_max = car_parameters_.max_lateral_acceleration;
38      double yaw_rate_limit = a_y_max / V;
39
40      if (yaw_rate_ref > yaw_rate_limit) {
41          yaw_rate_ref = yaw_rate_limit;
42      } else if (yaw_rate_ref < -yaw_rate_limit) {
43          yaw_rate_ref = -yaw_rate_limit;
44      }
45
46      return yaw_rate_ref;
47  }
```

B.2 PID High Level Controller

Listing B.2: Implementation of the PID high level controller.

```
1   #include "high_level_controllers/pid_controller.hpp"
2
3   #include <algorithm>
4   #include "rclcpp/rclcpp.hpp"
5
6   PIDController::PIDController(
7           const CarParameters& car_params,
8           double kp,
9           double ki,
10          double kd,
11          double dt,
12          double integral_limit,
13          double tau)
14      : HighLevelController(car_params),
15        kp_(kp),
16        ki_(ki),
```

```
17        kd_(kd),
18        dt_(dt),
19        tau_(tau),
20        integral_error_(0.0),
21        previous_error_(0.0),
22        previous_yaw_rate_(0.0),
23        filtered_ff_(0.0),
24        previous_yaw_moment_(0.0),
25        integral_limit_(integral_limit)
26  {
27  }
28
29  void PIDController::reset()
30  {
31      integral_error_ = 0.0;
32      previous_error_ = 0.0;
33      previous_yaw_rate_ = 0.0;
34      filtered_ff_ = 0.0;
35      previous_yaw_moment_ = 0.0;
36  }
37
38  YawMomentTorqueRequest PIDController::compute_control(
39      const VehicleState& state,
40      const Reference& reference,
41      const DriverCommand& driver_command)
42  {
43      YawMomentTorqueRequest request{};
44
45
46      const double error =
47          reference.yaw_rate - state.yaw_rate;
48
49      integral_error_ += error * dt_;
50
51      integral_error_ = std::clamp(
52          integral_error_,
53          -integral_limit_,
54           integral_limit_);
55
56      const double mz_pi =
57          kp_ * error +
58          ki_ * integral_error_;
59
60
61      const double yaw_acc =
```

```
62          (state.yaw_rate - previous_yaw_rate_) / dt_;
63
64      previous_yaw_rate_ = state.yaw_rate;
65
66
67      const double mz_in =
68          car_params_.moment_of_inertia_z * yaw_acc;
69
70
71      const double ff_error =
72          previous_yaw_moment_ - mz_in;
73
74
75      const double alpha =
76          tau_ / (tau_ + dt_);
77
78      filtered_ff_ =
79          alpha * filtered_ff_
80          + (1.0 - alpha) * ff_error;
81
82      const double mz_ff =
83          filtered_ff_;
84
85
86      double yaw_moment =
87          mz_pi + mz_ff;
88
89      yaw_moment = std::clamp(
90          yaw_moment,
91          -car_params_.max_yaw_moment,
92           car_params_.max_yaw_moment);
93
94      previous_yaw_moment_ = yaw_moment;
95
96      request.yaw_moment = yaw_moment;
97
98      request.total_torque = std::clamp(
99          driver_command.throttle *
100             car_params_.max_drive_torque,
101         0.0,
102         car_params_.max_drive_torque);
103
104     return request;
105 }
```

B.3 Low Level Controller

Listing B.3: Implementation of the low-level controller.

```
#include "low_level_controllers/simple_approach.hpp"

TorqueOutput SimpleApproach::compute_control(const YawMomentTorqueRequest&
     yaw_moment_torque_request) {
    TorqueOutput output;

    double delta_torque = yaw_moment_torque_request.yaw_moment /
        car_params_.rear_axle_distance;

    output.fl_torque = 0;
    output.fr_torque = 0;
    output.rl_torque = yaw_moment_torque_request.total_torque / 2 -
        delta_torque;
    output.rr_torque = yaw_moment_torque_request.total_torque / 2 +
        delta_torque;

    return output;
}
```

B.4 Traction Control

Listing B.4: Implementation of the traction control.

```
#include "traction_control/traction_control.hpp"

TractionControl::TractionControl(const CarParameters& car_params,
                                 double dt,
                                 double slip_threshold,
                                 double k_reduce,
                                 double k_increase,
                                 double slope_deadband)
    : dt_(dt),
      slip_threshold_(slip_threshold),
      k_reduce_(k_reduce),
      k_increase_(k_increase),
      slope_deadband_(slope_deadband),
      previous_slip_(0.0),
      torque_limit_(0.0),
      active_(false)
{}

```

```cpp
double TractionControl::compute_control(double slip_ratio,
                                        double driver_torque,
                                        VehicleState state)
{

    double slip_slope = (slip_ratio - previous_slip_) / dt_;
    previous_slip_ = slip_ratio;

    if (!active_) {
        torque_limit_ = driver_torque;

        if (slip_ratio > slip_threshold_) {
            active_ = true;
        }
        return driver_torque;
    }

    if (slip_ratio < slip_threshold_ * 0.8) {
        active_ = false;
        torque_limit_ = driver_torque;
        return driver_torque;
    }

    if (slip_slope > slope_deadband_) {
        torque_limit_ *= k_reduce_;
    } else if (slip_slope < -slope_deadband_) {
        torque_limit_ *= k_increase_;
    }

    torque_limit_ = std::clamp(torque_limit_, 0.0, driver_torque);

    return torque_limit_;
}

void TractionControl::reset()
{
    previous_slip_ = 0.0;
    torque_limit_  = 0.0;
    active_        = false;
}
```

Appendix C

Formula Student Disciplines

The Formula Student competition evaluates university teams through a comprehensive set of static and dynamic disciplines, designed to reflect the full lifecycle of an engineering project: from concept development and business planning to real-world vehicle performance. In recent years, the competition has expanded to explicitly include autonomous disciplines, reflecting current trends in the automotive industry.

A maximum of 1000 points can be achieved, divided into Static Events (325 points) and Dynamic Events (675 points). Dynamic events are further split into manual and autonomous disciplines. The official point allocation follows the Formula Student Germany 2026 ruleset [12].

C.1 Overview of Competition Disciplines

Figure C.1 illustrates the overall distribution of points across all competition disciplines, highlighting the relative importance of engineering design, endurance performance, and autonomous capabilities.

C.2 Static Events

Static events assess the overall engineering quality, economic feasibility, and commercial potential of the vehicle. These disciplines emphasize structured engineering processes, documentation quality, and communication skills rather than on-track performance.

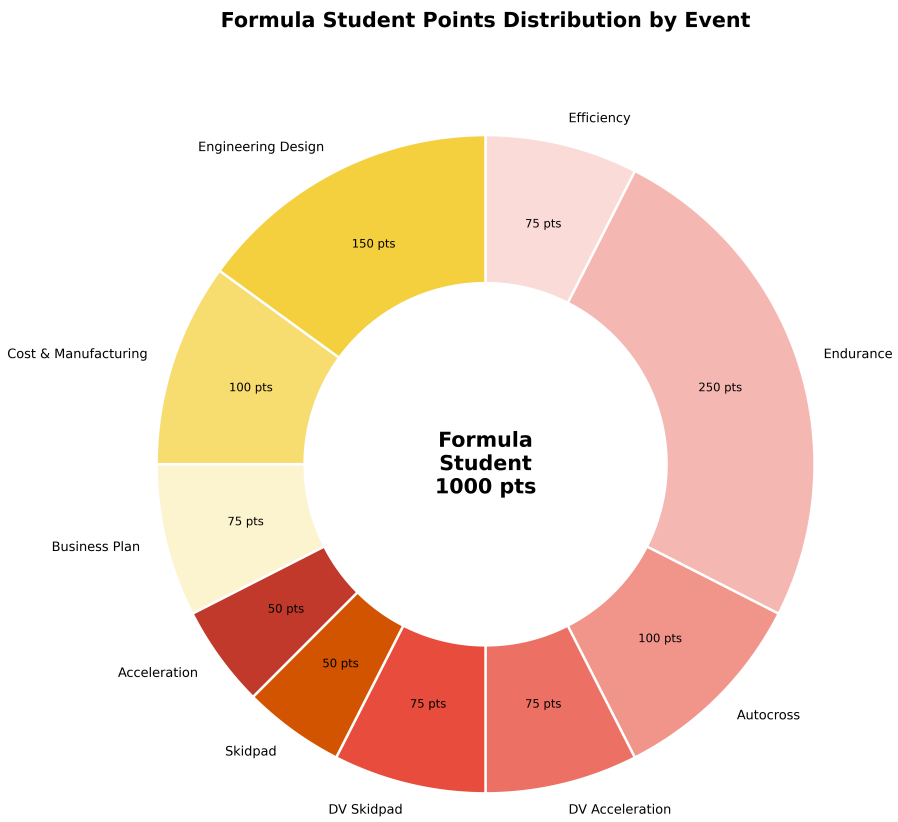


Figure C.1: Formula Student points distribution by event.

C.2.1 Static Events Points Distribution

Table C.1: Static events points distribution.

Static Event	Points
Engineering Design	150
Cost and Manufacturing	100
Business Plan Presentation	75
Total Static Points	**325**

C.2.2 Engineering Design

The Engineering Design event evaluates the technical merit of the vehicle and the team's ability to apply engineering principles. Judges assess the complete design process, including requirement definition, concept selection, detailed design, simulation, validation, and testing.

For autonomous vehicles, additional emphasis is placed on sensor selection, perception algorithms, control strategies, redundancy concepts, and functional safety considerations.

C.2.3 Cost and Manufacturing

The Cost and Manufacturing event assesses the economic viability of the vehicle design. Teams must demonstrate that their car could be produced in a limited series at reasonable cost while maintaining performance and safety standards.

C.2.4 Business Plan Presentation

The Business Plan Presentation evaluates the commercial potential of the project. Teams present a business concept in which a Formula Student assembly serves as the basis for a marketable product.

Strong communication skills and a coherent, realistic business strategy are critical to success in this discipline.

C.3 Dynamic Events

Dynamic events evaluate the real-world performance of the vehicle under competitive conditions. These disciplines focus on vehicle dynamics, powertrain performance, reliability, energy efficiency, and autonomous driving capability.

C.3.1 Dynamic Events Points Distribution

Table C.2: Dynamic events points distribution.

Dynamic Event	**Points**
Acceleration	50
Skidpad	50
Driverless Acceleration	75
Driverless Skidpad	75
Autocross	100
Endurance	250
Efficiency	75
Total Dynamic Points	**675**

C.3.2 Acceleration

The Acceleration event measures straight-line performance over a short distance. It primarily evaluates powertrain output, traction, launch control, and drivetrain efficiency.

C.3.3 Skidpad

The Skidpad event assesses lateral dynamics and cornering performance. Vehicles are driven through a constant-radius figure-eight or circular track, emphasizing suspension tuning, tire selection, and chassis balance.

C.3.4 Driverless Acceleration

Driverless Acceleration evaluates straight-line performance under fully autonomous control. The event tests the integration of perception, state estimation, trajectory planning, and longitudinal control.

C.3.5 Driverless Skidpad

The Driverless Skidpad event focuses on autonomous lateral control and vehicle stability at the limits of adhesion. Precision, repeatability, and safe operation without human intervention are critical evaluation criteria.

C.3.6 Autocross

Autocross is a timed handling event on a tight and technical circuit. It evaluates agility, braking performance, driver skill, and overall vehicle balance. For driver-operated vehicles, consistency and controllability are key factors.

C.3.7 Endurance

The Endurance event is the most heavily weighted discipline and represents a realistic driving scenario over a long distance. It evaluates reliability, thermal management, mechanical robustness, and sustained performance.

Completion of the Endurance event is a prerequisite for scoring points in the Efficiency discipline.

C.3.8 Efficiency

The Efficiency event rewards vehicles that achieve strong performance with minimal energy consumption. It directly links vehicle dynamics, powertrain efficiency, and strategic energy management, making it a critical indicator of overall system optimization.