



MOHAMAD FARID BIN ESA

A NON-LINEAR OPTIMAL STRATEGY FOR
SIMULTANEOUS PLANNING AND TRACKING
OF AUTONOMOUS VEHICLES

FACULTY OF ENGINEERING AND APPLIED SCIENCES
Transport Systems

PhD
Academic Year: 2020 - 2025

Supervisor: Dr Efstathios Siampis
Associate Supervisor: Dr Chenhui Lin
Jan 2025



FACULTY OF ENGINEERING AND APPLIED SCIENCES
Transport Systems

PhD

Academic Year 2020 - 2025

MOHAMAD FARID BIN ESA

A NON-LINEAR OPTIMAL STRATEGY FOR
SIMULTANEOUS PLANNING AND TRACKING
OF AUTONOMOUS VEHICLES

Supervisor: Dr Efstathios Siampis
Associate Supervisor: Dr Chenhui Lin
Jan 2025

This thesis is submitted in partial fulfilment of the requirements for
the degree of PhD

(NB. Remove if the degree award is based solely on examination of the thesis)

© Cranfield University 2025. All rights reserved. No part of this
publication may be reproduced without the written permission of the
copyright owner.

Academic integrity declaration

I declare that:

- the thesis submitted has been written by me alone.
- the thesis submitted has not been previously submitted to this university or any other.
- that all content, including primary and/or secondary data, is true to the best of my knowledge.
- that all quotations and references have been duly acknowledged according to the requirements of academic research.

I understand that to knowingly submit work in violation of the above statement will be considered by examiners as academic misconduct.

Abstract

Motion planning and control are critical for autonomous vehicles, enabling trajectory generation and control to meet defined objectives and constraints. However, traditional methods often rely on pre-defined optimised trajectories, which are computationally expensive to generate, and hierarchical schemes, which introduce delays. While many methods exist, the use of a perpendicular line-based reference for trajectory generation remains underexplored. This study introduces a novel controller under a simultaneous planning and tracking framework named the Perpendicular Line-Ahead Planner Controller (PLAPC). Using the Non-linear Model Predictive Control (NMPC) technique, the PLAPC simultaneously generates trajectories and performs control in real-time, without requiring pre-computed trajectories. The controller tracks predefined velocity references while adjusting control inputs to meet various objectives such as minimising distance to the reference line ahead, optimising comfort and improving energy efficiency by incorporating Linear-Exponential (LINEX) loss function terms. This approach integrates simultaneous trajectory planning, velocity tracking, and control into a unified framework.

The PLAPC was evaluated through high-fidelity simulations using MATLAB-Simulink, CasADi symbolic tools, and IPG CarMaker software. Initial validation involved a comparison with a traditional path-tracking controller (PTC) and was further tested across diverse path shapes, distances, and velocity profiles. The results demonstrated that the PLAPC dynamically generates trajectories in real-time and achieves superior velocity tracking, even with a front-wheel steering (FWS) vehicle topology as a baseline. The framework's versatility and computational efficiency highlight its adaptability and practicality for autonomous vehicle systems, significantly contributing to the advancement of autonomous vehicle motion planning and control strategies.

Keywords:

Autonomous Vehicle Control; Motion Planning and Control; Simultaneous Planning and Tracking; Non-linear Model Predictive Control; Optimal Control; Time Optimal; Comfort; Energy Efficient;

Acknowledgements

I would like to express my deepest appreciation to my mentor, Dr. Efstathios Siampis, for his invaluable guidance, relentless support, and encouragement throughout this research journey. His expertise and mentorship have been instrumental in shaping this work. I am thankful to Dr. Chenhui Lin for his pivotal role in completing this research. His insights and support have been deeply appreciated. In addition, thanks to Prof. Efstathios Velenis, whom I have known since my Master's degree in 2013. The initial opportunity he provided laid the foundation for starting this PhD journey.

My heartfelt gratitude goes to my wife, Azmazaira, who, with constant support, patience, and love, has been the foundation of this achievement. Her understanding, encouragement, and sacrifices have enabled me to focus and persevere through this journey. Without her, this milestone would not have been achievable. To my two wonderful children, Faeeq and Alisa, thank you for bringing joy and balance to our life. Your presence has been a source of strength and motivation every step of the way.

To my parents and siblings, whose unwavering support and encouragement, even from afar in Malaysia, I am deeply grateful. Their belief in me and the values they instilled have been the foundation of this milestone.

This PhD is a significant milestone in our life journey, filled with lessons and growth beyond our comfort zone. Through this challenging yet rewarding time, I hope we as a family have all grown into better versions of ourselves and contribute positively to others.

Table of Contents

Academic integrity declaration.....	iii
Abstract	iv
Acknowledgements	vi
List of Figures.....	x
List of Tables	xv
List of Equations.....	xvii
List of Abbreviations	xxi
1 Introductions.....	22
1.1 Overview	22
1.2 Aim and Objectives	29
1.3 Research Contributions	30
1.4 Thesis Structure.....	31
2 Literature Review	33
2.1 Introductions	33
2.2 Motion Planning and Control of an Autonomous Vehicle	34
2.2.1 Planning Methods	36
2.3 Simultaneous Planning and Tracking.....	44
2.3.1 Reference Systems	44
2.3.2 Planning and Tracking Approaches	46
2.4 Objective Functions	48
2.4.1 LINEX Loss Function for Energy-Efficient	50
2.5 Summary	53
3 Research Methodology	54
3.1 Introductions	54
3.2 Framework Development using CasADi	57
3.3 Summary	59
4 Vehicle and Path Model.....	60
4.1 Introduction	60
4.2 The Vehicle Model	62
4.2.1 Wheels Velocity Components	66
4.2.2 Vehicle Load Transfer	68
4.2.3 Tyre Model	70
4.2.4 Friction Circle	72
4.3 Path Model.....	73
4.4 Summary	77
5 Optimal Solutions	78
5.1 Introductions	78
5.2 Optimal Problem Formulations	79
5.2.1 Spatial Dynamics	80
5.2.2 Numerical Discretization and Variables.....	83

5.2.3 Constraints	84
5.2.4 Objective Functions	86
5.2.5 Regularisation and Scaling	88
5.3 Optimal Solution Generations	90
5.3.1 Introductions	90
5.3.2 Sensitivity Study of Comfort Weight Factor	90
5.3.3 Comparison Results Between Objective Functions	94
5.4 Summary	97
6 High-Fidelity Simulation Setup	98
6.1 Introduction	98
6.2 Local Coordinates Conversion	99
6.3 Velocity Profiles Generations	102
6.4 Actual Real-Time Implementation Considerations	106
6.5 Simulations Setup	108
6.6 Summary	108
7 Simultaneous Planning and Tracking Controller Development	109
7.1 Introductions	109
7.2 NMPC Problem Formulations	109
7.2.1 Objective Functions	113
7.2.2 Constraints	119
7.3 Sensitivity Study	121
7.3.1 Varying Integration Node	121
7.3.2 Varying Prediction Horizon Length	124
7.3.3 Varying Sampling Time	126
7.3.4 Varying Reference Line Distances Ahead	128
7.3.5 Varying Velocity Profile During Cornering	131
7.3.6 Varying LINEX Parameters	133
7.4 Robustness Study	140
7.4.1 Varying Total Vehicle Mass	141
7.4.2 Varying Electric Motors Time Delay	145
7.4.3 Combination Robustness Test Case	148
7.5 Summary	153
8 Test Result & Discussions	154
8.1 Introductions	154
8.2 Single Corner Short Path Evaluation	156
8.3 Multi Corners Extended Path Evaluation with Time Optimal Velocity Profile	172
8.4 Multi Corners Extended Path Evaluation with Individual Curvature- based Velocity Profile	178
8.5 Long Path Evaluation with Common Curvature-based Velocity Profile	185
8.6 Long Path Evaluation with Individual Curvature-based Velocity Profile	193
8.7 Overall Discussion	200

8.8 Summary	203
9 Conclusions.....	204
9.1 Summary of Contributions	204
9.2 Recommended for Future Work.....	206
References	208
Appendices.....	224
A The Parameters	224
A.1 Vehicle Parameters	224
A.1.1 Tire Parameters	225
A.2 Reference Path Parameters	226
A.2.1 Short Distance: 90-Degree Path	226
A.2.2 Short Distance: Multi-Corner Path	227
A.2.3 Long Distance: Sepang International Circuit, Malaysia.....	228
A.3 Optimal Solution Parameters.....	229
A.3.1 Minimum Time Objective	229
A.3.2 Optimized Comfort Objective	230
A.4 High Fidelity Simulation PLAPC Parameters	231
A.4.1 Single Corner Test.....	231
A.4.2 Multi-Corner Time Optimal Test.....	234
A.4.3 Multi-Corner Individual Velocity Test.....	234
A.4.4 Full Circuit Simulation, Common Velocity Test	235
A.4.5 Full Circuit Simulation, Individual Velocity Test.....	236
B Path Tracking Model	238
B.1 Problem Formulations	238
C Others.....	240
C.1 Four-Wheel Steering Vehicle Model.....	240
C.2 Multi Corners Extended Path Evaluation with Time Optimal Velocity Profile with 4WS Vehicle Topologies	243
C.3 Direct Method Comparison – Multiple Shooting vs Collocations	247

List of Figures

Figure 1-1 General structure of an Autonomous Vehicle system	23
Figure 1-2 In-car view of reference line ahead (blue cyan colour) [17].....	24
Figure 1-3 A general overview of the developed controller	25
Figure 2-1 Representation of “motion planning” that considers the dynamic of the system and its surroundings [20],[23]	35
Figure 2-2 Comparison of LINEX loss (red), quadratic loss (blue), and hinge loss (green)	50
Figure 3-1 Research framework diagram	54
Figure 3-2 Comparison between CasADi with other symbolic tools [95]	58
Figure 4-1 Vehicle topology	61
Figure 4-2 Vehicle coordinate frame [101]	62
Figure 4-3 Four-wheel double-track FWS vehicle dynamic model	65
Figure 4-4 Side and front view of the vehicle.....	68
Figure 4-5 Friction circle of a wheel.....	72
Figure 4-6 Vehicle top view in curvilinear coordinate for path model.....	73
Figure 4-7 90-degree right-hand corner short path.....	74
Figure 4-8 Multi-corners extended path.....	75
Figure 4-9 Full length of Sepang International Circuit	76
Figure 5-1 Vehicle top view for curvilinear coordinate representation	79
Figure 5-2 Curvilinear coordinate frame representation	80
Figure 5-3 Trajectories comparison with varying comfort weight factor	91
Figure 5-4 Weighted accelerations impact on varying comfort scale factor.....	92
Figure 5-5 Weighted score comparison on varying comfort scale factor	92
Figure 5-6 Vehicle trajectory comparison of all objective functions	94
Figure 5-7 Weighted lateral acceleration comparison of all objective functions	95
Figure 5-8 Weighted longitudinal acceleration comparison of all objective functions	95
Figure 6-1 Simulation setup for PLAPC in the IPG CarMaker	99
Figure 6-2 Local path coordinate frame representation.....	101

Figure 6-3 Example velocity profiles based on the path curvature	104
Figure 6-4 Time delay block for torque command block to the vehicle electric motors in the IPG CarMaker simulation model	107
Figure 7-1 Perpendicular lookahead reference line (cyan) ahead of the current vehicle states and prediction horizons	112
Figure 7-2 Comparison between LINEX and quadratic term in penalization of torque control inputs	118
Figure 7-3 Path constraint based on 2 nd -order polynomial fitting.....	120
Figure 7-4 Sensitivity study on average solved time and RMS error of velocity tracking by varying the integration node	122
Figure 7-5 Vehicle trajectory comparison by varying the integration node	122
Figure 7-6 Sensitivity study on average solved time and RMS error of velocity tracking by varying the prediction horizon length.....	124
Figure 7-7 Sensitivity study on average solved time and RMS error of velocity tracking by varying the sampling time.....	126
Figure 7-8 Sensitivity study on average solved time and RMS error of velocity tracking by varying reference line distance.....	129
Figure 7-9 Vehicle trajectory comparison by varying reference line distance .	129
Figure 7-10 Vehicle trajectory comparison by varying the in-corner velocity..	131
Figure 7-11 Penalty comparison of quadratic rule-based and LINEX functions	134
Figure 7-12 Vehicle trajectory comparison by varying the in-corner velocity..	134
Figure 7-13 Control inputs comparison between LINEX and Rule-based for energy efficient	135
Figure 7-14 Varying scale factor of LINEX parameters	137
Figure 7-15 Total energy consumptions compared to regenerative energy generated with varying scale factor of LINEX term.....	138
Figure 7-16 Net energy usage and energy efficiency ratio with varying scale factor of LINEX term	138
Figure 7-17 Robustness study on average solved time and RMS error of velocity tracking by varying total vehicle mass	141
Figure 7-18 Results comparison by varying total vehicle mass	142
Figure 7-19 Control inputs comparison by varying total vehicle mass.....	143

Figure 7-20 Robustness study on average solved time and RMS error of velocity tracking by varying electric motor time delay	145
Figure 7-21 Vehicle trajectory comparison by varying electric motor time delay	146
Figure 7-22 Torque inputs comparison by varying electric motor time delay..	146
Figure 7-23 Result comparison between all combinations with minimum distance objective	149
Figure 7-24 Control inputs comparison result between all combinations with minimum distance objective.....	150
Figure 7-25 IPOPT solver information for PLAPC simulations for all cases ...	150
Figure 8-1 Result comparison of PTC and PLAPC with optimised comfort solution as a reference.....	158
Figure 8-2 Control inputs comparison of PTC and PLAPC with optimised comfort solution as a reference	159
Figure 8-3 Deviations comparison of PTC and PLAPC with optimised comfort solution as a reference	159
Figure 8-4 Result comparison of PTC and PLAPC using a minimum time solution as a reference.....	160
Figure 8-5 Control inputs comparison of PTC and PLAPC using a minimum time solution as a reference	161
Figure 8-6 Deviations comparison of PTC and PLAPC using a minimum time solution as a reference	161
Figure 8-7 Plots comparison in time domain	165
Figure 8-8 Result comparison of PTC and PLAPC using a minimum time solution as a reference without tire load dependent coefficients.....	167
Figure 8-9 Control inputs comparison of PTC and PLAPC using a minimum time solution as a reference without tire load dependent coefficients.....	168
Figure 8-10 Deviations comparison of PTC and PLAPC using a minimum time solution as a reference without tire load dependent coefficients.....	168
Figure 8-11 PLAPC solved time comparison with optimised comfort as a reference	170
Figure 8-12 PLAPC solved time comparison with minimum time as a reference	170
Figure 8-13 PLAPC solved time without tyre load-dependent coefficient	170
Figure 8-14 Result comparison between minimum time solutions and PLAPC with minimum distance objective.....	173

Figure 8-15 Control inputs comparison result between minimum time solutions and PLAPC with minimum distance objective.....	174
Figure 8-16 IPOPT solver information for PLAPC simulations	174
Figure 8-17 Lateral error between minimum time solutions and PLAPC with minimum distance objective.....	175
Figure 8-18 Velocity tracking error between time optimal solutions and PLAPC with minimum distance objective	175
Figure 8-19 Velocity profiles based on the path curvature	179
Figure 8-20 Vehicle states comparison of between objective functions	180
Figure 8-21 G-G diagram comparison of between objective functions	180
Figure 8-22 Vehicle control inputs comparison of between objective functions	181
Figure 8-23 Velocity tracking error comparison of between objective functions	181
Figure 8-24 IPOPT solver information during the simulations	182
Figure 8-25 Common velocity profiles to test all objective functions	185
Figure 8-26 Vehicle states comparison of various objective PLAPC result for long-distance evaluations	187
Figure 8-27 G-G diagram comparison of various objective PLAPC result for long-distance evaluations	188
Figure 8-28 Vehicle control inputs comparison of various objective PLAPC result for long-distance evaluations	189
Figure 8-29 Velocity tracking error of various objective PLAPC for long-distance evaluations	189
Figure 8-30 IPOPT solver information during the simulations	189
Figure 8-31 Velocity profiles for Sepang International Circuit.....	193
Figure 8-32 Vehicle states comparison of various objective PLAPC results for long distances with distinct velocity profile.....	195
Figure 8-33 G-G diagram comparison of various objective PLAPC results for long distances with distinct velocity profile	196
Figure 8-34 Vehicle control inputs comparison PLAPC result for long distance with the distinct velocity profile	197
Figure 8-35 Velocity tracking error of various objective PLAPC for long distances with distinct velocity profile	197

Figure 8-36 IPOPT solver information during the simulations	197
Figure A-1 90-degree right-hand corner path	226
Figure A-2 90-degree right-hand corner path in IPG CarMaker.....	226
Figure A-3 Multi-corner extended path	227
Figure A-4 Multi-corner extended path in IPG CarMaker	227
Figure C-1 Vehicle topology	240
Figure C-2 Four-wheel double track vehicle dynamic model for 4WS vehicle topologies	240
Figure C-3 4WS result comparison for time objective between optimal solutions and PLAPC-4WS controller results.....	244
Figure C-4 4WS vehicle control inputs comparison for time objective between optimal solutions and PLAPC-4WS controller results	244
Figure C-5 IPOPT solver information during the simulations.....	245
Figure C-6 Lateral error for time-optimal objective between optimal solutions and PLAPC-4WS controller results.....	245
Figure C-7 Speed tracking error for time-optimal objective between optimal solutions and PLAPC-4WS controller results.....	245
Figure C-8 Generated trajectories comparison between the collocation method and multi-shooting method in solving OCP.....	247
Figure C-9 Accelerations and velocities result in comparison between the collocation method and multi-shooting method in solving OCP	248

List of Tables

Table 2-1 Comparison table of effectiveness of motion planning method based on highway application as mentioned in [7]	40
Table 3-1 Overall objective function in this work	56
Table 5-1 Objective functions for Optimal Solutions	86
Table 5-2 Comfort level based on weighted RMS accelerations as mentioned in ISO 2631-1	91
Table 5-3 Comparison results between objective functions for optimal solutions	94
Table 6-1 Type of velocity profile used in this work	105
Table 7-1 Objective functions for Simultaneous Planning and Tracking	113
Table 7-2 Comparison result of varying integration node	121
Table 7-3 Comparison result of varying prediction horizon length	124
Table 7-4 Comparison results of varying sampling time	126
Table 7-5 Comparison result of varying reference line distance ahead	128
Table 7-6 Comparison result of varying velocity profile	132
Table 7-7 Result comparison of LINEX and Rule-based for energy efficient..	135
Table 7-8 Result comparison of varying scale factor of LINEX term for energy efficient	137
Table 7-9 Comparison result of varying total vehicle mass	141
Table 7-10 Comparison result of varying electric motor time delay	145
Table 7-11 Combined case scenario change of parameters	148
Table 7-12 Comparison result of combined case scenario	148
Table 8-1 Simulation testing matrices in Chapter 8	155
Table 8-2 Simulation testing matrices for comparison between PLAPC and PTC	157
Table 8-3 Numerical comparison of PTC and PLAPC with an optimised comfort solution as a reference	162
Table 8-4 Numerical comparison of PTC and PLAPC with a minimum time solution as a reference	162
Table 8-5 Numerical comparison of PTC and PLAPC with a minimum time solution as a reference without tire load dependent coefficients	169

Table 8-6 Result of time objective for the multi-corner simulations	175
Table 8-7 Result of lap time for the multi-corner simulations.....	175
Table 8-8 Result of various objective simulations between velocity profiles...	182
Table 8-9 Result of multi-objectives simulations for long-distance evaluations	190
Table 8-10 Result of various objectives simulations for a long-distance path	198
Table A-1 Vehicle Parameters.....	224
Table A-2 Tyre Parameters	225
Table A-3 Initial and final condition parameters for optimal control problem ..	229
Table A-4 Initial and final condition parameters for PLAPC simulations set in the IPG CarMaker.....	231
Table C-1 Result of time objective in the multi-corner simulations	246
Table C-2 Result of lap time in the multi-corner simulations	246
Table C-3 Lap time and solved time comparison between collocation method and multi-shooting method	248

List of Equations

(4-1)..... 63

(4-2)..... 63

(4-3)..... 63

(4-4)..... 63

(4-5)..... 63

(4-6)..... 63

(4-7)..... 64

(4-8)..... 64

(4-9)..... 66

(4-10)..... 66

(4-11)..... 66

(4-12)..... 67

(4-13)..... 67

(4-14)..... 67

(4-15)..... 67

(4-16)..... 67

(4-17)..... 68

(4-18)..... 69

(4-19)..... 69

(4-20)..... 70

(4-21)..... 70

(4-22)..... 71

(4-23)..... 71

(4-24)..... 72

(4-25)..... 72

(4-26)..... 74

(4-27)..... 74

(5-1)..... 80

(5-2).....	80
(5-3).....	81
(5-4).....	81
(5-5).....	81
(5-6).....	81
(5-7).....	81
(5-8).....	81
(5-9).....	83
(5-10).....	83
(5-11).....	83
(5-12).....	84
(5-13).....	84
(5-14).....	84
(5-15).....	85
(5-16).....	86
(5-17).....	87
(5-18).....	87
(5-19).....	88
(6-1).....	99
(6-2).....	100
(6-3).....	102
(6-4).....	102
(6-5).....	103
(6-6).....	103
(7-1).....	110
(7-2).....	110
(7-3).....	110
(7-4).....	110
(7-5).....	110

(7-6).....	111
(7-7).....	111
(7-8).....	111
(7-9).....	111
(7-10).....	111
(7-11).....	111
(7-12).....	113
(7-13).....	114
(7-14).....	114
(7-15).....	114
(7-16).....	114
(7-17).....	115
(7-18).....	115
(7-19).....	115
(7-20).....	116
(7-21).....	117
(7-22).....	117
(7-23).....	117
(7-24).....	118
(7-25).....	119
(7-26).....	119
(7-27).....	119
(7-28).....	133
(7-29).....	133
(B-1)	238
(B-2)	238
(B-3)	238
(B-4)	238
(B-5)	238

(B-6)	238
(B-7)	239
(C-1)	241
(C-2)	241
(C-3)	241
(C-4)	242

List of Abbreviations

4WS	Four Wheel Steering
ATD	Active Torque Distributions
AV	Autonomous Vehicle
COG	Centre of Gravity
DOF	Degree of Freedom
EOM	Equation of Motion
FFT	Fast Fourier Transform
FWS	Front Wheel Steering
GPS	Global Positioning System
IPOPT	Interior Point Optimizer
IT	Information Technology
LINEX	Linear-Exponential
MF	Magic Formula
MPC	Model Predictive Control
NLP	Non-Linear Programming
NMPC	Non-Linear Model Predictive Control
OCP	Optimal Control Problem
PLAPC	Perpendicular Line Ahead Planner Controller
PTC	Path Tracking Controller
RK4	Runge Kutta 4 th Order
RWS	Rear Wheel Steering
W-RMS	Weighted Root Mean Square

1 Introductions

1.1 Overview

Self-driving, driverless, or autonomous vehicle (AV) technology is expected to bring significant positive changes to global transportation by offering safer, more efficient, and more convenient mobility options. Additionally, this technology has the potential to drive major economic benefits [1]. Over the last decade, many car manufacturers and related companies have taken significant steps in researching and developing autonomous vehicles. For example, Jaguar Land Rover, the largest vehicle manufacturer in the United Kingdom (UK), has prioritised four key areas for technological growth: Autonomous, Connected, Electric, and Shared. To support these goals, they have expanded their global tech hubs to locations such as Munich, Bologna, and Madrid to push forward their autonomous vehicle development [2]. Similarly, Nissan has launched a new UK government-funded project called evolvAD, focusing on creating autonomous mobility solutions for urban neighbourhoods and challenging rural roads. This project is an exciting step toward making autonomous vehicles a reality in everyday driving scenarios [3].

The development of AVs is driven by the need to enhance transportation efficiency, safety, and sustainability. In the UK, the Centre for Connected and Autonomous Vehicles projects that the connected and automated vehicle market could be worth £41.7 billion by 2035, underscoring the economic potential of AV technology [4]. Therefore, achieving good time performance is crucial for AVs to reduce travel durations and improve traffic flow. However, this must be balanced with passenger comfort, as rapid accelerations and abrupt manoeuvres can lead to discomfort and motion sickness, potentially hindering user acceptance and thus affecting the sales of the vehicle [5]. Energy efficiency is another critical factor, especially given the current trend in the adoption of electric vehicles, as AVs have the potential to significantly reduce energy consumption through optimised control and driving patterns. Studies suggest that with a fully autonomous road transport system, the UK's energy consumption by cars could be reduced to 30%–45% of current levels, depending on the trade-off between

energy efficiency and travel time [6]. Addressing these aspects of time efficiency, passenger comfort, and energy efficiency is essential for AVs widespread adoption and success.

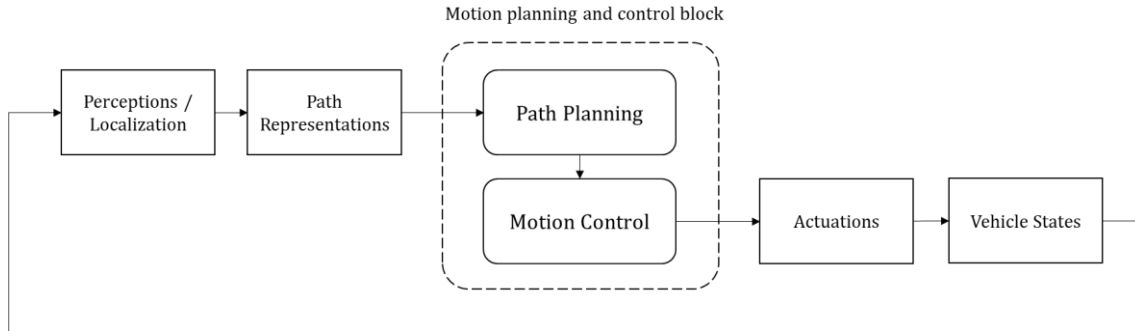


Figure 1-1 General structure of an Autonomous Vehicle system

As the development of AVs progresses, more research is needed, especially in the area of motion planning and control, which is one of the most critical sub-systems in the AV overall scheme, as shown in Figure 1-1 [7]. The motion planning and control layer is responsible for formulating a feasible trajectory given the set of constraints and specific objective function from the vehicle's current states to the target destination ahead provided by the previous layers [8].

Two challenges for motion planning and control remain critical in achieving AVs' full potential, particularly in hierarchical approach and trajectory generation. Trajectory optimisation tasks often demand significant computational resources, making it challenging to integrate planning or re-planning and control within autonomous vehicle systems [9]. For instance, the work [10] highlights that generating a reference trajectory can be extremely time-consuming, especially for long, broad and complex paths, limiting its practicality for large-scale environments. On the other hand, in unstructured environments, the absence of high-precision maps intensifies the problem. Reference path errors can become too large to generate reliable trajectories, leading to tracking overshoots and abrupt vehicle motions [11].

Regarding common hierarchical approaches, they suffer from inherent delays due to their sequential way, where trajectory generation and control execution are treated separately, often producing dynamically infeasible trajectories, leading to potential collisions or instability. Their sequential structure also introduces delays or temporal lag, reducing adaptability in dynamic environments [12], [13]. For example, such frameworks often experience delays during manoeuvres, as trajectory adjustments and vehicle control actions are performed in separate steps, leading to suboptimal performance [14]. Similarly, hierarchical control schemes with multiple control layers, such as those using RRT* and MPC, increase system complexity and response time, resulting in computational delays and reduced efficiency in dynamic scenarios [15]. Thus, when employing a hierarchical structure, the system risks losing critical information during communication between layers, resulting in suboptimal decisions arising from differing objectives at each stage, thereby reducing the overall optimality of the system. [16].

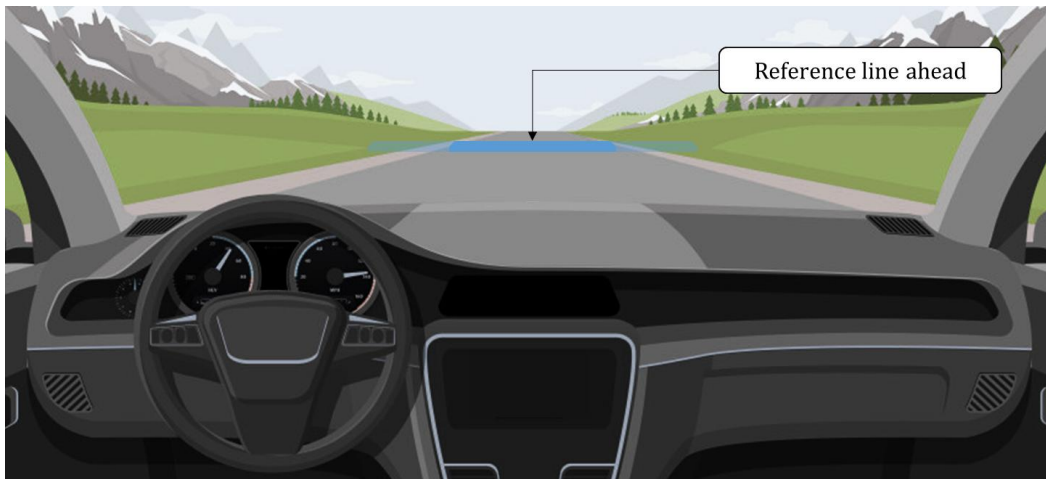


Figure 1-2 In-car view of reference line ahead (blue cyan colour) [17]

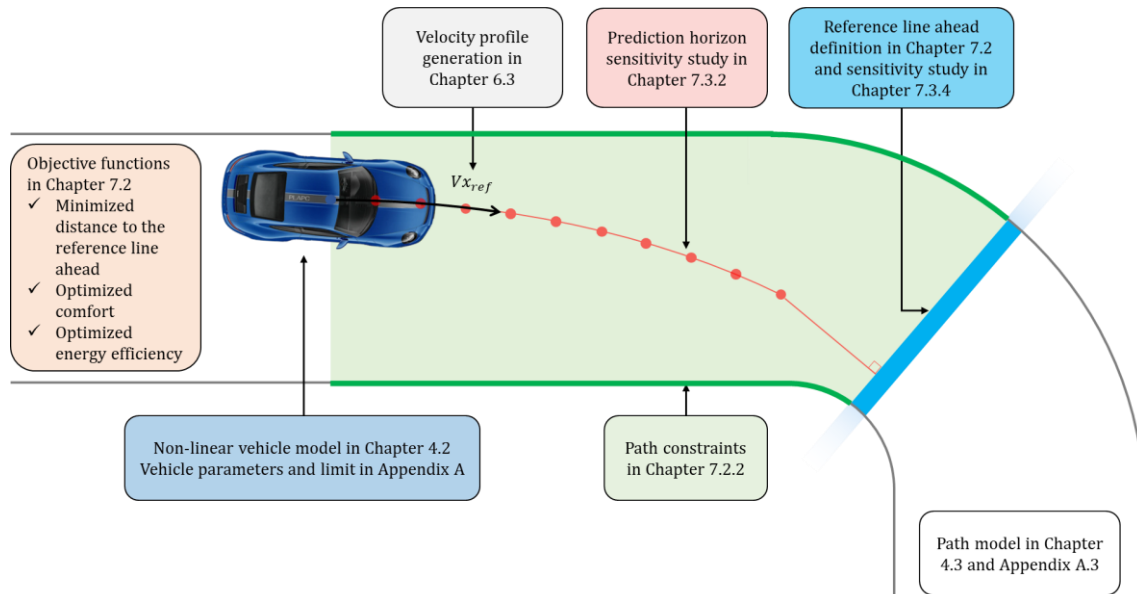


Figure 1-3 A general overview of the developed controller

A key area of innovation in this work lies in the simplification of reference systems through a simultaneous approach. Unlike traditional methods that rely on a hierarchical approach and require precise target point coordinates or pre-computed trajectories, this approach uses a perpendicular lookahead reference line, simplifying the trajectory generation process and eliminating the need for pre-defined references or pre-computed trajectories.

As shown in Figure 1-2, this method is inspired by how human drivers perceive the road, providing a baseline for understanding dynamic driving scenarios and lateral boundary awareness. While this may seem like a limitation for automated driving, the key advantage lies in how the system extends beyond human perception. Unlike human drivers, an automated system can process real-time data from multiple sensors, accurately estimate vehicle positioning, and predict future trajectories to ensure smoother and safer navigation. Additionally, the ability to incorporate multiple safety factors into the controller, such as a continuous path boundary window to keep the vehicle within the path and a limited line width to maintain lateral awareness of the future path, goes beyond human capability, further enhancing reliability in complex environments. By

shifting from rigid target point references to a continuous perpendicular lookahead reference line, this approach provides a flexible and intuitive alternative for motion planning and control.

Figure 1-3 provides a high-level overview of the controller's function, illustrating its key components. The controller integrates various elements, including a path model, a non-linear vehicle dynamic model, velocity profile generation, a reference line ahead, path constraints, and predictive control methods, to ensure effective trajectory planning and execution. The tools used to develop and test the controller are MATLAB-Simulink, CasADi, and IPG CarMaker software.

In order to determine the context of this work, there are three main areas related to the fundamentals of AV navigation which are path planning, path tracking and motion planning and control. Path planning involves determining only a feasible path or trajectory from a starting point to a destination, taking into consideration relevant constraints to ensure safe and collision-free travel [18]. Path tracking refers to maintaining the vehicle on a predefined path or trajectory and adhering to a given velocity profile by using control strategies to minimise deviations from these references [19]. Motion planning and control extends beyond traditional path planning and tracking by continuously generating and adjusting the vehicle's trajectory and control commands in real-time. This process explicitly considers vehicle dynamics, environmental constraints, and meeting the predefined objective functions [7], [20], [21].

Therefore, this thesis contributes to the field of motion planning and control, reframed within the context of a simultaneous planning and tracking framework, leveraging the perpendicular lookahead reference line approach to address existing challenges in trajectory generation and simultaneous implementation. By integrating the planning and tracking processes into a unified framework, the proposed method highlights safety, computational efficiency, and its suitability for real time implementation. The framework aims to enhance the performance of AVs while meeting the demanding requirements for time efficiency, optimised passenger comfort, and energy efficiency across diverse driving scenarios and applicable to common vehicle topologies.

To formalize the definition of the main aspects considered in this work, three primary aspects are studied in this work, described as follows,

- Time efficiency

Aims to minimise travel duration to achieve the shortest possible time to complete a given path. In this work, time efficiency is considered in two key aspects. The first involves achieving minimum-time through optimal solutions (offline approach) that solve the whole path at once, serving as a benchmark for validation. These offline-generated solutions provide an idealized reference trajectory and velocity, representing the shortest possible time to complete a path under given constraints. They act as a performance baseline against which the controller's results are compared. The second aspect focuses on minimising the distance to the reference line ahead in the controller's online implementation. Unlike the offline approach, where a precomputed trajectory is generated and used for reference, the controller dynamically generates trajectories in real-time by referring to a perpendicular reference line ahead. By keeping the vehicle as close as possible to this moving reference, the controller effectively pursues time efficiency, ensuring continuous adaptation toward an optimal path while maintaining real-time feasibility.

- Comfort optimisation

Refers to minimising rapid changes in acceleration and the rate of acceleration (jerk) to enhance ride quality. In this study, comfort is evaluated using ISO 2631-1 [22], which classifies comfort levels based on weighted root mean square (RMS) acceleration values. By penalizing longitudinal and lateral accelerations, the proposed approach ensures smoother vehicle operation, reducing passenger discomfort. Comfort optimisation is applied in both optimal solutions (offline approach), where the trajectory is computed to minimise acceleration terms variations across the entire path and in online implementation, where the controller refers to the reference line ahead and dynamically adjusts control inputs to maintain a smooth driving experience in real-time.

- Energy efficiency

Refers to minimising energy consumption by optimising the applied torque control inputs, which directly represent forces acting on the vehicle. Unlike traditional quadratic penalization methods, this study introduces an asymmetric penalization approach using the Linear-Exponential (LINEX) loss function, which heavily penalizes positive torque while allowing greater utilization of negative torque for regenerative braking. This formulation encourages energy recovery, particularly in electric autonomous vehicles. Unlike time efficiency and comfort optimisation objectives, implementation of energy efficiency is only considered in the online implementation of this work, where the controller dynamically adjusts torque inputs in real-time to optimise energy consumption while maintaining vehicle stability and performance.

1.2 Aim and Objectives

This thesis focuses on integrating simultaneous trajectory generation (planning) and tracking (control) frameworks for AV, emphasising real-time, unified optimisation of planning and control tasks.

Therefore, this research aims to establish an integrated and real-time simultaneous planning and tracking framework for AV, referred to as the Perpendicular Line-Ahead Planner Controller (PLAPC), which enhances autonomous vehicle system capability. This framework uses a numerical optimisation methodology, specifically Nonlinear Model Predictive Control (NMPC). By incorporating constraints and objective functions related to time efficiency, passenger comfort optimisation and energy efficiency, this thesis seeks to demonstrate a novel controller that meets the computational requirements necessary for real-world deployment. To ensure feasibility, the proposed approach will be validated through high-fidelity, real-time simulations.

To achieve this aim, the specific objectives are as follows,

1. To design and develop a controller based on numerical optimisation, utilizing a perpendicular lookahead reference line strategy for simultaneous trajectory planning and control. This includes incorporating constraints and objective functions for time efficiency, passenger comfort optimisation and energy efficiency while ensuring computational feasibility.
2. To evaluate and analyse the effectiveness of the developed controller under various velocity profiles, path shapes and distances, as well as comparison with conventional path-tracking controllers. This includes sensitivity studies to understand the behaviour of the controller.
3. To assess and validate the developed controller in a high-fidelity, real-time simulation environment. This will involve measuring key performance metrics of velocity tracking accuracy and computational load to demonstrate readiness for potential real-world deployment.

1.3 Research Contributions

Following the aim and objective of this research, the research has led to the following applications and contributions,

1. Development of a novel perpendicular line-ahead planner controller

Based on the controller development as in Chapter 7, the controller was developed within a simultaneous planning and tracking framework by defining a perpendicular lookahead reference line as reference instead of a single-point reference coordinate and without the need for pre-computed trajectories.

2. Incorporation of multiple objective functions into the controller

As highlighted in the literature review in Chapter 2 and simulation result in Chapter 8, this research strategies and demonstrates the controller's adaptability to multiple objective functions that address conflicting goals, such as minimising time versus optimising comfort or energy efficiency. This capability allows AVs to dynamically balance different operational demands, ensuring effective and efficient performance in diverse scenarios.

3. Implementation of LINEX term for torque control input penalization for energy efficient

As described in the literature review in Chapter 2 and further studies in Chapter 7, this work introduced the LINEX (Linear-Exponential) term as a method for penalizing torque control inputs, offering an improvement over traditional quadratic rule-based penalty for energy efficiency objective function. This approach promotes better energy efficiency by promoting more energy recovery and also simpler to implement.

1.4 Thesis Structure

This thesis is structured to provide a logical flow from foundational concepts to developing, testing, and validating the proposed controller named PLAPC. Next is Chapter 2, which begins with a review of the existing literature, starting on the broader aspect of motion planning and control for autonomous vehicles and focusing on simultaneous planning and control. This review highlights the challenges and advancements in the field, forming the basis for the research presented in this work.

Chapter 3 outlines the methodological approach, presenting the overall research framework. It details the progression from the development of the vehicle model to the development of PLAPC based on Non-linear Model Predictive Control (NMPC). Chapter 4 describes the numerical models used in this research, including the vehicle dynamics model employed for both the optimal control solution and NMPC implementation. Additionally, it introduces the mathematical models for path representation and constraints, which form the foundation for developing the proposed controller.

Building on this foundation, Chapter 5 addresses optimal solutions generations. It defines the problem formulation, outlines objective functions and constraints, and investigates the sensitivity of selected parameters.

With the groundwork established, Chapter 6 transitions to the high-fidelity real-time simulation environment. This chapter explains how the developed controller was integrated into IPG CarMaker, ensuring real-time feasibility and representation of real-world scenarios.

Chapter 7 presents the PLAPC problem formulation, details objective functions, the constraints as well as sensitivity and robustness study to evaluate how key parameters affect vehicle dynamics and performance. Then, Chapter 8 delves into the validation and evaluation of the developed PLAPC via high-fidelity real-time simulation. This chapter documented the testing result and discussion of this controller development. Starting with validation against traditional path tracking to an actual full length of Sepang International Circuit. It demonstrated the PLAPC

implementation with various objective functions such as time efficiency, optimise for comfort and energy efficiency.

The thesis concludes in Chapter 9, summarizing the key contributions and findings of the research. The chapter also offers recommendations for future work, including potential improvements to the developed algorithms and suggestions for real-world testing and deployment, paving the way for further advancements in autonomous vehicle technologies.

2 Literature Review

2.1 Introductions

The chapter mainly focused on three main areas that contributed to the current research, as described below,

- The first section provides a broader overview of motion planning and control for autonomous vehicles and delves into the scope of the simultaneous planning and tracking frameworks. Various methods will be discussed and compared to identify the most appropriate approach for this research and the gap of knowledge.
- The following section covers various objective functions implemented in the controller, focusing on conflicting objectives like time efficiency, comfort optimisation, and energy efficiency. The study examines how these objectives impact autonomous vehicle trajectories.
- The final section explores the application of the Linear-Exponential (LINEX) loss function to penalize positive torque input, promoting more regenerative braking and more straightforward to implement.

Overall, the literature review identifies gaps in existing research related to simultaneous planning and tracking frameworks for autonomous vehicles and highlights how this research aims to address these gaps to advance the field.

2.2 Motion Planning and Control of an Autonomous Vehicle

The motion planning tier is accountable for generating a feasible trajectory from the vehicle's current state to the desired destination state by satisfying the kinematics and dynamics of the vehicle and its surroundings. The surroundings could be the road constraints and static and dynamic obstacles around an autonomous vehicle. A feasible trajectory means achieving the targeted objective function and satisfying the constraints. In a typical configuration, the motion planner output is then forwarded to the local control tier to execute and regulate the autonomous vehicle to follow this motion planned [8], [7].

Path planning and motion planning are often confused but serve distinct purposes. Path planning focuses on finding a collision-free path from the start to the destination based on its objectives, such as the shortest distance or minimum travel time [18]. It determines a high-level route but does not account for the vehicle's dynamic behaviour or its interaction with the environment. In contrast, motion planning incorporates several fundamental components that are essential for generating feasible trajectories. It considers the dynamics of the vehicle and time-varying conditions. Key components of motion planning include state estimation, which determines the vehicle's current position and condition; time evolution, which predicts how the system will change over time; action planning, which determines appropriate manoeuvres to meet defined objectives; and constraint compliance, ensuring adherence to physical and operational limits or constraints [7], [20]. In addition, motion planning may use an output of path planning as a reference or simultaneously plan the trajectories based on a set of dynamic references, such as moving obstacles [21]. Figure 2-1 illustrates motion planning as a dynamic process considering the abovementioned components.

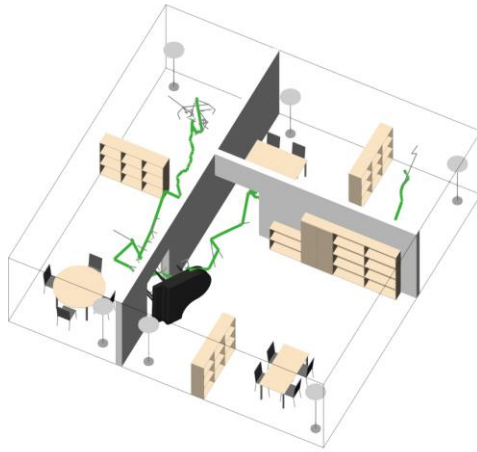


Figure 2-1 Representation of “motion planning” that considers the dynamic of the system and its surroundings [20],[23]

Path planning provides a high-level route, while motion planning incorporates dynamic system behaviour for trajectory generation that meets the constraints and objectives. Therefore, simultaneous planning and tracking of this work builds upon these principles by integrating trajectory generation (planning) and execution of control action (tracking) into a unified framework. Unlike traditional hierarchical methods, this approach dynamically generates trajectories in real-time, ensuring adaptability to environmental changes while adhering to constraints and objectives. The research aligns with this approach by extending motion planning and control principles to develop a controller based on the simultaneous planning and tracking framework.

2.2.1 Planning Methods

Since the scope of this research is within motion planning and control with particular attention to simultaneous planning and tracking framework, it is worth exploring various methods within the broader context of motion planning and control for autonomous vehicles.

How the motion planning problem is formulated influences the selection of motion planning methods. The problem formulation varies significantly based on the type of scene representation used, which may involve different forms of inherited data, such as discrete or continuous, algebraic or analytic, and static or dynamic models [7]. Therefore, one must consider the above-mentioned aspects when selecting motion planning methods for AVs. There is much literature on the motion planning method, especially as mentioned in [7], [8], [21], [24] where there are four main algorithm classifications of motion planning include graph search-based algorithms, sampling-based methods, interpolation curve methods, and numerical optimisation techniques. In addition, other classifications were studied in this work including the bio-inspired planning method and the recent advancement of artificial intelligence (AI).

Graph search-based methods, such as A* and Dijkstra's algorithm, rely on systematic exploration of a discretized search space to find the shortest or optimal route. These methods are widely used in structured environments but suffer from computational inefficiency when scaling to complex, high-dimensional spaces. Moreover, while graph-based methods can handle static obstacles well, they lack adaptability in dynamic environments and struggle to incorporate vehicle kinematics and dynamics, making them less suitable for real-world autonomous driving scenarios [7]. Sampling-based methods, such as Rapidly-exploring Random Trees (RRT) and Probabilistic Roadmaps (PRM), improve computational efficiency by randomly sampling feasible paths within the search space [8]. Although these methods allow greater flexibility in high-dimensional planning problems, they often generate non-smooth trajectories and do not inherently enforce system constraints, which makes them less suitable for real-time AV applications [21].

Interpolation curve methods, such as Bezier curves and B-splines, have been widely used in motion planning to generate smooth trajectories by fitting continuous curves between predefined waypoints. These methods are computationally efficient and allow for precise trajectory shaping [24]. However, they are generally limited in robustness when operating in highly dynamic environments and often require additional optimisation techniques to ensure feasibility under dynamic constraints [7]. Another widely studied approach is numerical optimisation-based motion planning, which formulates the trajectory generation problem as an optimisation problem, incorporating constraints explicitly. Optimisation techniques such as Linear Programming (LP), Quadratic Programming (QP), Nonlinear Programming (NLP), Dynamic Programming (DP), and Model Predictive Control (MPC) have been explored extensively in the literature [7]. QP is often employed for convex optimisation problems but struggles with non-convex constraints, limiting its applicability in real-world vehicle dynamics modelling. LP is often used for problems with linear constraints and objective functions, making it computationally efficient but unsuitable for handling complex vehicle dynamics or non-linear constraints. While NLP is more flexible as it accommodates nonlinear constraints; however, its computational complexity increases significantly, making it less practical for real-time applications [25], [26]. DP, while effective in solving multi-stage decision problems, is notoriously affected by the curse of dimensionality, making it impractical for real-time motion planning [27].

Besides those approaches, bio-inspired planning methods have gained attention due to their adaptability in uncertain and dynamic environments. It leverages optimisation principles derived from natural phenomena to enhance the adaptability of autonomous navigation systems. It exhibits self-organizing patterns, making them robust even under adverse conditions, and readily adapts to dynamic environments, ensuring effective path planning in uncertain and rapidly changing scenarios [28]. These approaches, including swarm intelligence, evolutionary algorithms, and neurodynamics-based methods, have gained significant attention for their ability to handle complex, dynamic, and uncertain environments [29]. It can efficiently explore large search spaces, avoid local

optima, and dynamically adapt to changes in the environment [30]. Swarm intelligence-based algorithms, such as Particle Swarm Optimisation (PSO) and Ant Colony Optimisation (ACO), are inspired by collective behaviours in nature. PSO mimics the movement of bird flocks or fish schools, adjusting the trajectory of candidate solutions based on personal and global best positions [31]. ACO, on the other hand, simulates how ants deposit pheromones to reinforce optimal paths, making it particularly effective for solving shortest-path problems in unknown environments [30]. Evolutionary algorithms, including Genetic Algorithms (GA) and Differential Evolution (DE), rely on principles of natural selection and genetic variation to iteratively improve path solutions [29]. Additionally, neurodynamics-based methods use biologically inspired neural networks to continuously adapt path-planning strategies, making them well-suited for real-time applications but, this approach increases the complexity of computing in large-size environments [29]. Since bio-inspired algorithms are inherently model-free approaches, they do not guarantee an optimal solution and may struggle to consistently achieve globally optimal paths [28]. This limitation justifies the need for model-based techniques, which explicitly consider system dynamics and constraints to optimise the trajectories which is a critical requirement for autonomous vehicles especially operating at high velocity and under highly dynamic conditions.

Given the current advancements in AI, the simultaneous planning and tracking approach presented in this work is highly relevant, particularly in relation to end-to-end learning frameworks. Data-driven AI methods, especially those utilizing images, LiDAR data, and expert driving demonstrations, integrate both tasks, enabling more adaptive decision-making. End-to-End Learning offers a unified framework by directly mapping sensor inputs to control outputs. In Imitation Learning (IL), models learn from expert demonstrations. A study in [32] combines image color-based RGB and LiDAR data to generate control commands, reducing collision rates by 48%. Another study [33] improves trajectory tracking performance using self-supervised learning from large-scale driving videos, achieving accuracy gains ranging from 2% to 100% across different driving tasks. Building on IL, condition imitation learning (CIL) enhances the learning process

with multi-view attention and a 180° field of view, improving lane-following accuracy and reducing infractions [34]. Additionally, a study in [35] introduces Dynamic Conditional Imitation Learning, which combines laser scanner data and camera streams to improve generalization, utilizes Occupancy Grid Mapping for better route planning, and enhances the vehicle's ability to detect and avoid road blockages, improving driving success in varied conditions. Besides, a Convolutional Neural Network (CNN)-based end-to-end driving model predicts steering angles directly from images, demonstrating real-world feasibility in outdoor tests [36]. While another method under end-to-end learning is Reinforcement Learning (RL), which enables vehicles to optimise both planning and tracking through trial-and-error learning. One study [37] applies End-to-End Model-Free RL for Urban Driving, using ResNet-based implicit affordances to refine both path selection and trajectory tracking. Another approach [38] uses Deep Deterministic Policy Gradient (DDPG) combined with a Variational Autoencoder (VAE), allowing a vehicle to learn lane-keeping in just 30 minutes. Additionally, Simulation-Based RL successfully transfers trained policies to full-sized autonomous vehicles in nine real-world driving scenarios [39]. Furthermore, a model-based RL method introduces a forward model of the ego-vehicle, predicting motion outcomes and optimising control policies using Bellman equations, which improves tracking efficiency [40].

Besides, looking at AI-Based Path Planning & Tracking further enhances adaptability. CNN-based path planning methods, such as the End-to-End One-Shot Path-Planning Algorithm, use Fully Convolutional Neural Networks (FCNNs) to compute optimal paths on grid maps, outperforming traditional methods like A and Dijkstra* [41]. A CNN-based navigation model predicts steering commands from images, executing real-world trajectory tracking, and demonstrating end-to-end learning in action [36]. On the other hand, DynGMP, a Graph Neural Network (GNN)-based motion planner, optimises dynamic motion planning by minimising costly replanning, achieving near 100% success rates in unpredictable environments, but it is not an end-to-end method [42].

Although AI-based end-to-end learning methods have gained attention for their potential to simplify the motion planning process by directly mapping sensor inputs to control actions, they are excluded from this study due to concerns regarding transparency and reliability. These methods often function as “black box” models, making it difficult to interpret their decision-making processes [43], which is critical for diagnosing and correcting errors in safety-critical applications such as autonomous driving [7], [44]. Additionally, the reliability of end-to-end approaches is questioned due to their inconsistent performance in diverse and unforeseen driving scenarios, particularly in handling edge cases and dynamic environments. Without extensive training in various situations, these models may fail to consistently produce safe and predictable behaviour [45], [46].

Table 2-1 Comparison table of effectiveness of motion planning method based on highway application as mentioned in [7]

Type of motion planning / Criterias	Real time	Robustness	Stability	Adaptability	Constraints	Complexity	Vehicle Kinematics & Dynamics	Uncertainty
Numerical optimization	Good	Good	Good	Very good	Very good	Good	Very good	Intermediate
Graph-based search, A*	Intermediate	Intermediate	Good	Good	Good	Intermediate	Very poor	Intermediate
Sampling-based search, RRT	Good	Good	Intermediate	Very good	Poor	Poor	Very good	Good
Parametric Curve, Clothoid	Good	Poor	Very poor	Good	Good	Intermediate	Good	Good

In contrast to other motion planning methods, numerical optimisation methods stand out as the most balanced and effective approach across the evaluated criteria in Table 2-1. These methods provide good real-time performance, robustness, stability, and adaptability while also excelling in handling system constraints and vehicle dynamics. Unlike other approaches, numerical optimisation techniques are able to directly incorporate vehicle dynamics into the motion planning process, ensuring that the generated trajectories are feasible and safe to execute according to predefined objectives and constraints.

Additionally, these methods can adapt to variations in the environment in real-time, making them well-suited for dynamic and uncertain driving conditions. Although numerical optimisation methods can be computationally demanding, advancements in hardware technology, optimisation solvers, and real-time strategies, such as warm-starting, help mitigate this challenge [7].

The numerical optimisation problem for motion planning is formulated as a solve-algorithm based on logical and empirical approaches, meaning that the problem is solved using solid data and mathematical models. This optimisation process is typically expressed as either a minimisation or maximisation of an objective function under a set of constraints [7]. In this method, the vehicle model and constraints are mathematically defined to ensure the problem is well-posed. This method is capable of solving motion planning problems objectively and accurately, making them ideal for balancing multiple objectives, such as time efficiency, comfort optimisation and energy efficiency [24].

However, despite their advantages, numerical optimisation methods also have some inherent challenges, as mentioned earlier. Since these methods rely on mathematical formulations, they require accurate modelling and tend to be computationally intensive, particularly in real-time applications. This can introduce complexity in developing the controller, as the optimisation process must balance computational efficiency with accuracy to ensure that the motion planning results remain both feasible and practical for real-world driving scenarios. Therefore, in developing the controller for this work, it is crucial to manage the complexity of the model while ensuring that the optimisation remains effective for real-time autonomous vehicle control [7], [24].

Among the methods previously mentioned, Model Predictive Control (MPC) stands out as one of the most suitable approaches for autonomous vehicle motion planning [47], [48] and is also well aligned with simultaneous planning and tracking in this work. MPC also referred to as receding horizon control, is widely regarded as one of the most effective strategies for managing constrained nonlinear systems with multiple inputs and outputs (MIMO). In MPC, an optimal control sequence is calculated over a finite prediction horizon to minimise a

predefined objective function while adhering to defined constraints. The first control input from this sequence is then applied to the system, and the entire process is repeated at each sampling interval [49], [50].

MPC has been widely applied in autonomous vehicle motion planning, addressing challenges in trajectory tracking, obstacle avoidance, and high-performance driving. One of the earliest implementations was the Model-Predictive Motion Planner (MPMP) for the IARA autonomous car, which successfully navigated urban environments with high precision and obstacle avoidance [48]. Further advancements introduced Nonlinear MPC (NMPC) for dynamic stability constraints, outperforming human drivers in stability and accuracy [51]. Additionally, an MPC-based comfort optimisation strategy used genetic algorithms to tune control parameters, reducing lateral deviation and yaw errors while minimising motion sickness [52]. In autonomous racing, a hierarchical MPC framework enabled time-optimal trajectories, achieving a 20% faster lap time while maintaining real-time feasibility [53]. Recent works explore integrated MPC for simultaneous trajectory planning and tracking, demonstrating effective obstacle avoidance through veDYNA-Simulink simulations [54]. Further advancement with adaptive MPC technique that leverages Particle Swarm Optimisation (PSO) and Back Propagation (BP) neural networks for real-time tuning of prediction horizons, improving control accuracy in high-curvature conditions [55].

Advancements in learning-based MPC have further improved autonomous vehicle performance. Learning MPC (LMPC) refines trajectory optimisation by systematically storing past vehicle trajectories, reducing computational complexity while ensuring real-time feasibility, validated on the Berkeley Autonomous Race Car (BARC) platform [56]. A modified LMPC with error dynamics regression has enhanced handling limit exploration, successfully deployed in the Indy Autonomous Challenge race car [57]. Meanwhile, reinforcement learning-based MPC via Proximal Policy Optimisation (PPO-MPC) dynamically adjusts prediction horizons in real-time, improving trajectory tracking adaptability over conventional fixed-horizon MPC [58].

These studies highlight the versatility of MPC in stability optimisation, adaptive learning, and high-performance driving, with further potential to enhance real-world autonomous vehicle systems.

One of the key advantages of MPC is its ability to perform real-time re-planning, as it uses a prediction horizon to update the trajectory based on new information continuously. This allows the vehicle to adapt to changes in the environment and account for dynamic constraints, making MPC particularly effective for dynamic driving conditions. Additionally, MPC can incorporate complex vehicle dynamics and multiple objectives into the optimisation process. In addition, it has been demonstrated that for nonholonomic systems, integrating planning and kinematic control through MPC can substantially enhance overall system performance [59], [60]. This makes MPC a robust and flexible solution for solving motion planning problems in real-world autonomous driving scenarios.

However, MPC also has some limitations. It is generally better suited for convex problems, and its performance can degrade in non-convex situations or with very complex constraints. Moreover, it can be computationally intensive [61], particularly when the prediction horizon is long or when the vehicle model includes highly non-linear dynamics. To address these challenges, advancements in real-time optimisation strategies, such as warm-starting, help improve the computational efficiency of MPC.

Given the objectives of this research, which require the consideration of non-linear vehicle dynamics, real-time implement-ability, and optimisation of multiple objectives, MPC is the most appropriate choice compared to other numerical optimisation methods, as mentioned in the previous paragraph. Its ability to handle dynamic constraints and adapt to real-time changes makes it a versatile and reliable technique for solving problems simultaneously for autonomous driving.

2.3 Simultaneous Planning and Tracking

A key aspect that highlights the novelty of this work is the simultaneous planning and tracking framework, which leverages a line-ahead reference system in the NMPC problem formulation. This approach is central to the development of a unified methodology for simultaneous planning and tracking. Unlike traditional methods that often separate planning and control, this work integrates these processes, enabling dynamic adaptability and meeting real-time requirements. The following subchapter will be divided into two parts related to the Simultaneous Planning and Tracking framework which are in terms of its reference systems and hierarchical approaches.

2.3.1 Reference Systems

Reference systems play a critical role in defining the desired trajectory for autonomous vehicles, aiding as the foundation for both planning and control tasks. Many approaches use lateral and longitudinal errors as references, measured relative to predefined paths or centrelines. For instance, [62] utilizes a reference system based on the lateral distance error from the lane centre to the vehicle centre of gravity (COG) and target velocity, enabling effective tracking through sequential quadratic programming (SQP) and validated via Hardware-in-the-Loop testing. Similarly, Bézier curves are employed in [63] to generate smooth paths, where lateral deviation and target velocity serve as references for an NMPC formulation in urban autonomous driving scenarios. Also as described in [64], it employs a path-velocity decomposition strategy, using the track centerline as a static geometric reference and precomputed the velocity profile with quasi-steady-state lap time simulations.

In another approach, dual-coordinate approaches are proposed, such as in [65], where Frenet and Cartesian coordinate systems are integrated within the NMPC framework. These approaches penalize deviations in lateral position, velocity, and heading angle relative to a predefined path. Alternatively, reference trajectories based on curvilinear coordinates are used in [53], [65] and [66], with

trajectories optimised offline for lap-time minimisation before being tracked as references in real-time using the NMPC technique.

Dynamic reference systems introduce adaptability by continuously updating targets in real time. For example, as mentioned in [67], a virtual target system defines a point-based reference that dynamically adjusts its position and velocity along a global path generated by the A* algorithm. This work integrates adaptive MPC, where the prediction horizon dynamically adjusts based on the robot-to-target distance, and an event-triggered mechanism to reduce the computational load by re-planning only when necessary. Another work as in [68] utilized a virtual sub-target point as a reference for obstacle avoidance. These dynamically generated points provide a temporary reference for the vehicle, allowing it to escape local minima and continue toward the original target. Similarly, [69] introduces a so-called motion target as a local equilibrium point, using a curvature-based velocity profile to ensure smooth and adaptive motion. Predefined global trajectories are also employed in [13], where the global reference serves as a baseline for simultaneous planning and control, updated locally to account for dynamic obstacles and environmental conditions. While in [70], an adaptive preview strategy is proposed, where the preview distance dynamically adjusts based on road curvature and lateral tracking error. This approach uses a point-based reference that mimics human driving behaviour, enhancing real-time adaptability and improving path tracking performance

This work builds upon the concept of dynamic references by employing a line-ahead reference system, which, to the best of our knowledge, has not been previously explored. This approach enhances the integration of planning and control by dynamically adjusting trajectories to meet constraints, defined objectives and computational efficiency.

2.3.2 Planning and Tracking Approaches

Planning and tracking frameworks for autonomous systems are typically categorized as either hierarchical or simultaneous. Hierarchical approaches separate planning and control into layers, where the high-level planner generates a trajectory that the low-level controller tracks. For instance, [71] introduces a two-layer MPC framework in which the high-level planner generates reference trajectories by minimising deviations from desired velocity and path, while the low-level controller incorporates human inputs to enhance safety and comfort. Similarly, [66] employs an offline trajectory optimisation strategy to generate optimal paths for lap-time minimisation as a reference, which are then tracked using NMPC in real-time to minimise lateral deviation to the reference path. Another similar approach as in [53], that is also minimises the local heading angle.

In contrast, simultaneous approaches integrate planning and control into a unified framework, addressing these limitations by concurrently optimising trajectories and control inputs. For example, [10] applies a simultaneous planning and control framework for unmanned surface vessels, combining an improved artificial potential field (IAPF) for trajectory generation with MPC for tracking. The IAPF ensures collision-free paths while considering angular velocity constraints, and MPC handles dynamic feasibility and system constraints. A similar framework is employed in [72] for automated lane changes on slippery roads, where adaptive MPC handles nonlinear vehicle dynamics and tyre forces to maintain stability and safety with given objectives and safety constraints. Moreover, the most recent [13] introduces a simultaneous planning and tracking framework for autonomous mobile robots, leveraging a kinematic model and dual constraints to ensure safety and smooth motion in dynamic environments. This work integrates trajectory tracking and obstacle avoidance into a single MPC problem where the simultaneous framework was proposed due to the hierarchical method introduces a sequential lag between the planning and the control stage.

Another notable example is the work in [73], which combines planning and tracking within a single NMPC framework by serial cascading models. A single-

track vehicle model is used for precise short-term control, while a simplified point-mass model extends the planning horizon for long-term predictions, reducing computational load. Similarly, in [47], they proposed integrations of a planning and tracking framework combining artificial potential fields and neural network-based nonlinear MPC to directly generate control commands, adapting to road surface variations and dynamic environments. This approach improves stability and safety compared to hierarchical methods. Another study by [16], eliminates the reliance on predefined trajectories by introducing a Hybrid MPC approach that integrates manoeuvre decision-making directly within vehicle motion planning. Demonstrated through overtaking scenarios, this Hybrid MPC approach achieves superior performance by optimising vehicle speeds more effectively and maintaining higher desired speeds than conventional methods.

The simultaneous approach adopted in this work emphasizes the real-time integration of planning and control, allowing for dynamic adaptability to environmental changes while optimising computational costs. By leveraging a unified framework, the proposed method eliminates the sequential lag inherent in hierarchical approaches and improves adaptability in autonomous vehicle navigation.

2.4 Objective Functions

The development of motion planning and control strategies for AVs has increasingly focused on optimising multiple objectives, including time efficiency, passenger comfort optimisation, and energy efficiency. The choice of an objective function depends on the specific application of the AV. For instance, in racing scenarios, minimising lap time is the primary goal, whereas passenger vehicles require a balance between comfort, time efficiency, and energy consumption. To enhance adaptability across various applications, a multi-objective optimisation approach is essential to accommodate these potentially conflicting objectives.

Minimising travel time is one of the earliest and most studied objectives in motion planning. In racing applications, minimum-time trajectory optimisation has been extensively researched, focusing on optimal racing lines to achieve faster lap times. Studies such as [74], [75] have developed optimisation strategies to maximise progress along a track that minimises lap time. These studies use nonlinear optimisation methods to find the optimal racing line while considering vehicle dynamics constraints, such as lateral acceleration limits and tyre friction. However, in non-racing applications, time optimisation must be balanced with other objectives, particularly comfort and energy efficiency, to ensure safe and practical driving for everyday use.

In terms of the objective function for the motion planning strategy, there are various of it, such as maximising efficiency (minimum time or maximum progress or distance), maximising comfort level (rate of change of curvature, reduce jerk), maximising safety (obstacle avoidance) [25], [76]. The selection of the objective function depends on the application of the autonomous function itself.

Passenger comfort is a critical objective in motion planning for AVs, especially in the context of urban and highway driving. Comfort-focused objective functions typically aim to minimise accelerations terms and jerk (rate of change of acceleration) and smooth out velocity and trajectory changes. Research by [77] demonstrates a multi-objective NMPC approach that considers lateral deviation, acceleration minimisation, and jerk reduction to enhance comfort while maintaining trajectory accuracy. This study introduces a hierarchical multi-

domain strategy, prioritising objectives based on risk levels and environmental conditions rather than applying fixed weights to all objectives.

Recent studies have further highlighted the importance of minimising abrupt accelerations and decelerations to improve passenger comfort. For instance, [78] proposes a method incorporating visual occlusion awareness to enhance safety while ensuring passenger comfort through controlled velocity adjustments. This aligns with the findings in [79] that emphasize the need for smooth trajectory planning to improve the overall riding experience, particularly for urban driving scenarios where frequent stops and turns occur.

Another objective function in terms of energy efficiency is becoming increasingly important, particularly for electric AVs, where reducing energy consumption can significantly extend driving range. Ecological velocity planning is a common strategy to achieve energy savings by optimising the vehicle's speed profile. For instance, [80] focuses on reducing energy consumption while maintaining safety and comfort by adjusting velocity based on road conditions and traffic. Similarly, [81] presents a study that uses MPC to optimise velocity profiles for energy efficiency while ensuring that safety constraints are met. These studies demonstrate that integrating energy-efficient strategies with comfort and time optimisation is essential for achieving a balanced and practical motion planning solution for AVs.

The integration of multiple objectives into the developed controller is crucial to developing a versatile and adaptable control strategy that can accommodate different objectives based on driving conditions and user preferences. This research aims to address this challenge by considering a multi-objective objective function that optimises these objectives in a computationally efficient manner, using NMPC numerical optimisation techniques.

2.4.1 LINEX Loss Function for Energy-Efficient

The Linear Exponential (LINEX) loss function is an asymmetric loss function that penalizes overestimation and underestimation differently, making it particularly useful in scenarios where these errors have unequal consequences. In other words, this approach applies a heavier penalty on one side while assigning a lighter penalty to the other [82].

Figure 2-2 compares LINEX loss with two other loss functions: quadratic (squared) loss and hinge loss. LINEX loss is a highly asymmetric, smooth convex function, whereas quadratic loss is a symmetric, smooth convex function, and hinge loss is a non-smooth convex function. The hinge loss function will not be considered in this work due to its non-smooth nature, which could impact the penalization process, potentially affecting vehicle navigation particularly in terms of comfort.

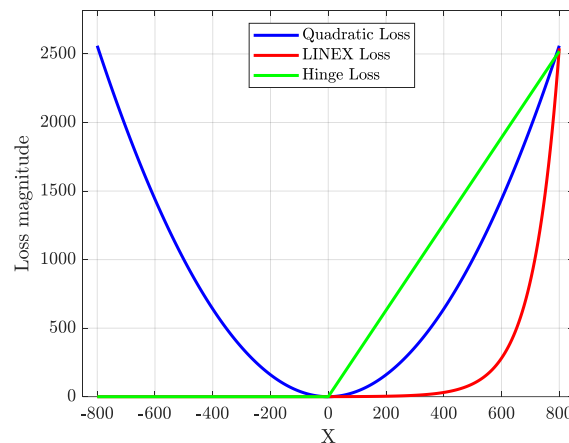


Figure 2-2 Comparison of LINEX loss (red), quadratic loss (blue), and hinge loss (green)

This characteristic is advantageous in control systems and parameter estimation, where the costs associated with deviations can vary in significance. In the context of control systems, the LINEX loss function allows for the design of controllers that can account for asymmetric costs, leading to more efficient and effective control strategies.

The study of the LINEX loss function began with the foundational work of [83], which examined real estate assessments. This work highlighted how overestimating property values could lead to costly appeals, while underestimation resulted in linear revenue loss, emphasizing the need for asymmetric loss functions. In [84], further exploration addressed cases where overestimation and underestimation have unequal consequences, showing that traditional estimators designed for symmetric loss functions might be suboptimal in such scenarios. By deriving optimal estimators using LINEX, this study underscored the importance of tailoring loss functions to better reflect real-world asymmetries in decision-making. More recently, in [85], the LINEX loss function was proposed as a flexible alternative to the traditional symmetric quadratic loss function for process optimisation. This study demonstrated how LINEX could effectively balance quality improvement and cost considerations in manufacturing and process control by adapting penalties to reflect real-world asymmetries.

In autonomous vehicle applications, LINEX loss functions are particularly beneficial as they effectively address scenarios where resource overuse, such as excessive energy consumption, carries significantly higher consequences than underutilization. By customizing penalties, it encourages behaviours that minimise energy expenditure while maintaining performance objectives. This makes it an ideal choice for energy-efficient applications like autonomous vehicle control, where it can prioritize regenerative braking and optimise torque distribution to balance performance and reduce net energy usage.

As previously mentioned, one common approach is the use of quadratic cost functions, which penalize control inputs such as torque input equally in both directions. This method, often implemented in optimisation-based controllers like MPC, provides smooth control inputs and is computationally efficient, making it a popular choice for real-time applications [86]. Another common way is rule-based strategies, such as simple IF-ELSE logic, which are computationally efficient. They enable energy-saving mechanisms like regenerative braking activation or switching to eco-modes during cruising speeds, are commonly used for hybrid electric vehicles [87].

However, to the best of our knowledge, the application of the LINEX loss function specifically for energy efficiency in autonomous vehicles has not yet been explored. This presents a valuable research opportunity, especially in the domains of motion planning and control, where the integration of LINEX loss functions could simplify implementation and promote significant energy efficiency improvements, particularly in electric autonomous AVs.

2.5 Summary

This research tackles critical gaps in the literature to develop a novel controller for simultaneous planning and tracking in AV systems. The proposed controller, Perpendicular Line-Ahead Planner and Controller (PLAPC) offers a distinct alternative to conventional point-based references, improving adaptability to dynamic environments and enhancing safety. This innovative strategy provides an alternative approach to motion planning and control systems, contributing to the advancement of autonomous vehicle technologies. The research ensures real-time feasibility in dynamic scenarios by integrating a multi-objective optimisation framework that addresses time efficiency, comfort optimisation, and/or energy efficiency.

3 Research Methodology

3.1 Introductions

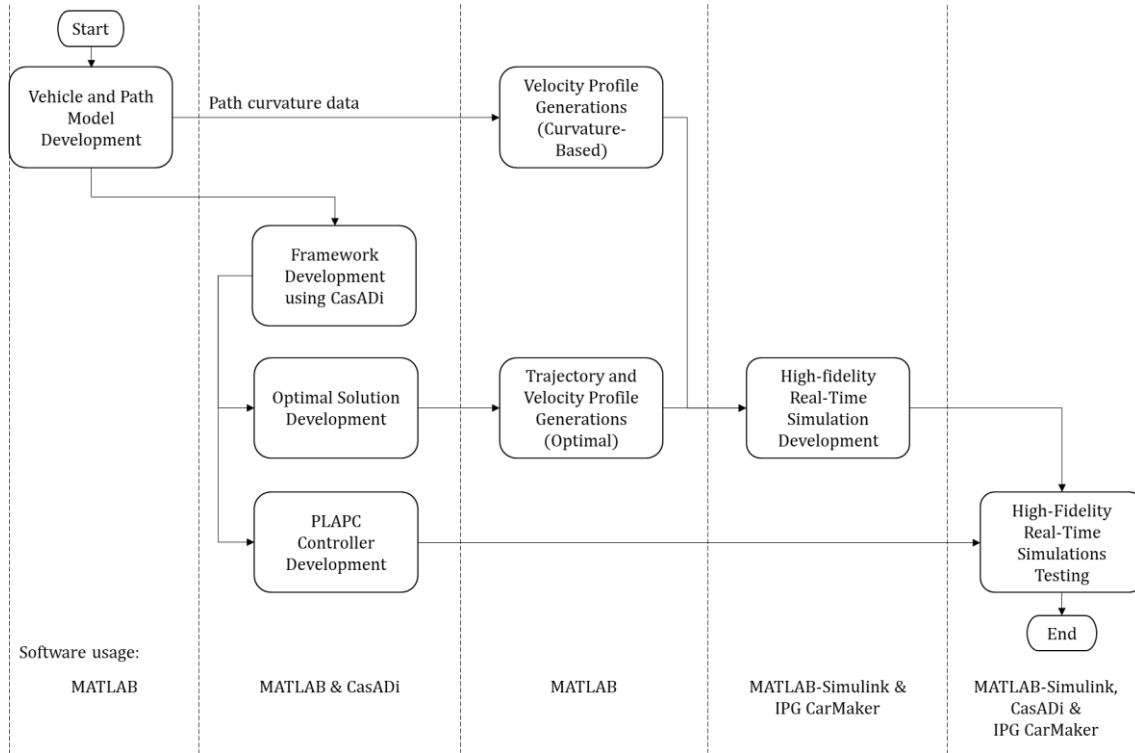


Figure 3-1 Research framework diagram

To achieve the aim and objectives outlined in Chapter 1, a novel simultaneous planning and tracking controller named PLAPC was developed. The development of this controller adopts a methodology based on numerical optimisation and testing through real-time simulation via MATLAB-Simulink with CasADi symbolic tools and IPG CarMaker high-fidelity simulation.

Figure 3-1 illustrates the structured workflow of the research. It begins with the development of vehicle and path models in MATLAB, providing critical inputs for subsequent processes. Using the CasADi symbolic toolbox within MATLAB, the research establishes a computational framework for two main tasks: Optimal Solution Development and PLAPC Controller Development. Overall objective function as detailed in Table 3-1. Two types of velocity profiles are generated which are the curvature-based velocity profile, computed directly from path

curvature data, and the optimal velocity profile, derived from solving the offline optimal solution approach. Alongside the optimal velocity profile, an optimal reference trajectory is also generated. These optimal references serve as benchmarks or baselines for comparison and validation during path-tracking simulations and the PLAPC controller. Depending on the simulation scenario, either the curvature-based or optimal velocity profile is then used as input for the high-fidelity real-time simulations in IPG CarMaker integrated with MATLAB/Simulink.

Two main types of high-fidelity real-time simulation testing were conducted which are path-tracking simulations, which compared the PLAPC performance against a traditional path-tracking approach, and simultaneous planning and tracking simulations, which assessed the integrated planning and control capabilities of the PLAPC. The results from these simulations were analyzed to validate the controller's effectiveness in achieving the research objectives as described in Sub-Chapter 1.2.

The write-up of this work and computational work was conducted on an Asus TUF Gaming A15 laptop (year 2020 model) equipped with an AMD Ryzen 7 4800H processor running at 2.2 GHz, with 8 cores, 16 threads, Radeon Graphics, and 16 GB of RAM. The optimisation tasks were implemented in MATLAB R2017b using CasADi version 3.6.5 with the IPOPT solver for controller development. The developed controller was then embedded into MATLAB-Simulink, which was integrated with IPG CarMaker 7.1.1 for high-fidelity simulation testing within a virtual environment. Both software versions were used because the existing vehicle model in IPG CarMaker was already established, proven to work with these versions, and successfully tested in a real-time hardware environment using a full-scale prototype vehicle [88].

Table 3-1 Overall objective function in this work

Type of objective functions	Methods	Descriptions
Minimised time, J_t^{off}	Optimal solutions (offline)	Minimising total time across the whole track at once.
Optimised comfort, J_c^{off}	Optimal solutions (offline)	A multi-objective function balancing time minimisation and comfort by penalizing acceleration and jerk (rate of acceleration), ISO 2631-1 [22] is referred to define the comfort level.
Minimised distance, J_d^{on}	PLAPC (online)	Minimising distance from the vehicle's current state and each horizon points to the reference line ahead.
Optimised energy, J_e^{on}	PLAPC (online)	A multi-objective function balancing distance to the reference line ahead minimisation and energy efficiency by penalizing positive torque using LINEX terms.
Optimised comfort, J_c^{on}	PLAPC (online)	A multi-objective function balancing distance to the reference line ahead minimisation and comfort by penalizing acceleration and jerk (rate of acceleration), the ISO 2631-1[22] is referred to define the comfort level.

3.2 Framework Development using CasADi

Optimal solutions and PLAPC controller development are based on the CasADi framework written in MATLAB software. It is an open-source symbolic framework designed for numerical optimisation and algorithmic differentiation, widely used for solving nonlinear programming (NLP) and optimal control problems (OCP). CasADi is an open-source tool derived from Algorithmic Differentiation (ADi) based on Computer algebra systems (Cas) syntax [89]. CasADi enables users to symbolically formulate optimisation problems and integrate them with numerical solvers such as Interior Point OPTimizer (IPOPT) to solve them, which has been used in this work. Its symbolic expression tree structure allows efficient evaluation of derivatives through automatic differentiation. CasADi essentially calculates and finds the first-order derivative, gradient or Jacobian of a function to solve an optimisation problem. Its ability to handle constraints, dynamics, and custom functions makes it ideal for complex real-world applications such as motion planning and control for autonomous systems.

Like CasADi, Acados, the successor to ACADO, is also an open-source toolbox that improves real-time NMPC performance through a modular and efficient framework. Unlike ACADO, Acados provides higher flexibility and execution speed by leveraging the Basic Linear Algebra Subroutines for Embedded Optimization (BLASFEO) library and removing dependency on static code generation [90]. Additionally, Acados is able to integrate with CasADi, making it capable of defining complex NMPC problems [90],[91].

Compared to other commonly used toolboxes, FORCESPRO is a commercial solver specifically designed for real-time applications, utilizing interior-point methods for high-speed execution [92]. FORCESPRO also integrated with CasADi library such as CasADi as a default Automatic Differentiation (AD) tool for both of their MATLAB and Python clients [93]. Besides, it has been demonstrated to be 50 times faster than ACADO in MPC applications [94]. However, its closed-source nature and licensing costs limit its accessibility for academic research and customisation, making it less flexible compared to CasADi.

CasADi has also demonstrated superior performance compared to other symbolic toolboxes such as SymPy, YALMIP, or MuPAD, particularly when working with large variable sets, as shown in Figure 3-2 [95], [96]. Its flexibility is further enhanced by seamless integration with multiple programming interfaces, including MATLAB/Octave, Python, and C++ [97]. Although FORCESPRO and Acados are optimised for real-time applications, CasADi provides an unmatched combination of flexibility, solver compatibility, and ease of implementation, making it the ideal choice for this work.

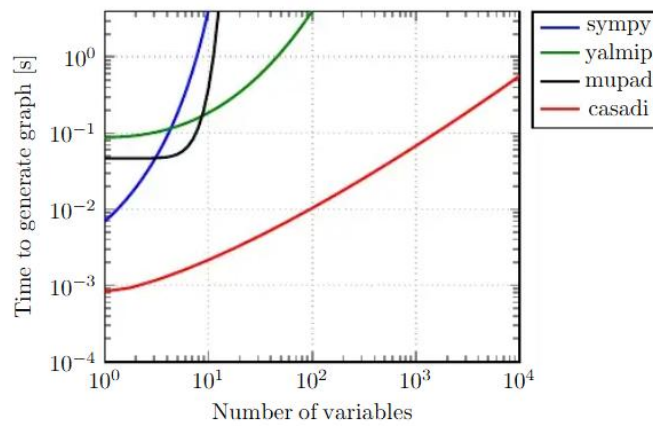


Figure 3-2 Comparison between CasADi with other symbolic tools [95]

While IPOPT is an open-source solver for large-scale NLP problems, it is one of the built solvers in CasADi [98], [97]. It efficiently handles nonlinear objectives and constraints using a primal-dual Interior Point Method and works well with CasADi for automatic differentiation. IPOPT's advantages include scalability, robust handling of complex dynamics, and efficient convergence, making it ideal for development in motion planning and control controllers. It is widely used in applications like MPC and trajectory optimisation, where system dynamics and constraints need to be optimised in real time.

The process begins with the symbolic definition of the problem to construct an optimisation problem using CasADi and solve it with IPOPT. CasADi provides a symbolic framework where the decision variables, the objective function, and constraints are defined using symbolic classes of *SX* for Scalar Expression or *MX*

for Matrix Expression. The decision variables represent the states, X or control inputs, U , the objective function $f(x)$ describes the cost to minimise or maximise, and constraints $g(x)$ ensure equality and inequality conditions are satisfied. Once these elements are defined, they are combined into an NLP structure, which includes the objective function, decision variables, constraints, and fixed parameters. CasADi then interfaces with the IPOPT solver using the *nlpsol* function, where solver options such as tolerances and maximum iterations, are configured. The solver requires an initial guess for the decision variables, bounds on variables, and bounds on constraints to start the optimisation process. In this work, the initial guess was taken from the previously solved result as a warm start. IPOPT then performs numerical optimisation and iteratively solves to find an optimal solution. After convergence, the solution (optimal states and inputs) is retrieved from the solver for real-time implementation or further analysis.

3.3 Summary

This chapter presented the systematic research methodology undertaken to develop and validate the simultaneous planning and tracking controller, PLAPC. The process began with the development of vehicle and track models, which served as the foundation for subsequent stages. The problem was then mathematically formulated as a nonlinear optimisation problem, utilizing CasADi for symbolic formulation and IPOPT for solving both the optimal solutions and the NMPC model of PLAPC. The velocity profile was generated based either on the optimal solution or a generic profile derived from the path curvature. Either of these velocity profiles was implemented as references in High-Fidelity Real-Time Simulations using IPG CarMaker to ensure realistic testing environments. Finally, the methodology concluded by validating the controller with a traditional path tracking controller and further assessing the performance across various predefined paths and velocity profiles under the High-Fidelity Real-Time Simulations Testing.

4 Vehicle and Path Model

4.1 Introduction

This chapter describes the vehicle dynamic and path models used to develop the optimal control solution and PLAPC for real-time implementation in high-fidelity simulation using IPG CarMaker software.

This research uses a double-track vehicle dynamic model, including a non-linear Pacejka's tyre model with a load-dependent coefficient tyre model. Lateral and longitudinal load transfer were considered, too. This is the minimum vehicle model required in this work due to the consideration of being as close as possible to an actual vehicle, which must take into account an important aspect of load transfer, especially that involves time efficiency related as a cost function.

Double track vehicle dynamic model with seven degree-of-freedom (DOF) and two load transfer elements are considered in this work, which are longitudinal translation, lateral translation, yaw motion at the vehicle mass centre, all wheels dynamic as well as longitudinal and lateral load transfer. The vehicle model was selected as this has the best trade-off between representing the actual vehicle model with transient dynamics [99] and the complexity of solving the optimisation calculation. This is essential to ensure that the vehicle model is as close as possible to an actual vehicle so that the controller that will be implemented captures the required details. Referring to [100], the 7 DOF vehicle model is the simplified version in terms of solving time and closer representation to the actual vehicle compared to a more complex model of 10 or even 14 DOF. Less DOF could solve faster but less accurate; however, higher DOF created complexity and burden to solve it in time which will be crucial for the real-time implementation.

In this work, the vehicle platform will be based on a small 874.5 kg two seater electric vehicle (EV) equipped with active torque distribution (ATD) of an in-wheel motor on each rear wheel and equal torque distribution of the front wheels sides that is driven by motors through a differential as shown in Figure 4-1.

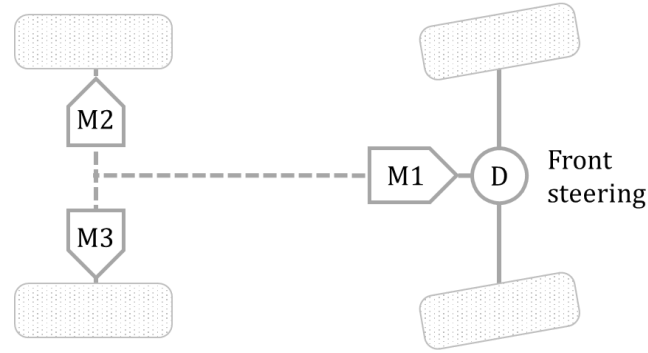


Figure 4-1 Vehicle topology

The path or track used in this study was predefined and developed using a curvilinear approach, where road curvature information is incorporated to generate X and Y coordinates, path angles, and distances. The study considered multiple path configurations to comprehensively evaluate the proposed controller with the assumption of,

- A continuous path without any junctions or intersections.
- The path without any static or dynamic obstacles.

The path models in this work include:

- A single 90-degree right-hand turn to analyze specific aspects of validating the developed controller against a traditional path-tracking controller.
- Multi-corner extended paths to study the combined effects of longitudinal and lateral dynamics in driving scenarios.
- A full-length path of the Sepang International Circuit to assess the PLAPC's effectiveness on a realistic, long-distance, high-speed, and demanding track.

This combination of models and path configurations thoroughly evaluates the proposed controller, from simplified cases to realistic and challenging environments under various scenarios.

4.2 The Vehicle Model

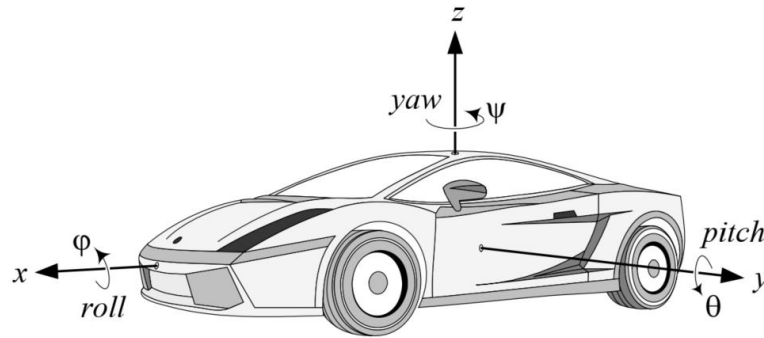


Figure 4-2 Vehicle coordinate frame [101]

As mentioned, a 7 DOF vehicle dynamic model is used in this research, with the equation of motion presented in a coordinate frame centred to the COG of the vehicle, as shown in Figure 4-2. An assumption is needed in consideration to reduce the complexity of the vehicle model in order to ensure implement-ability in real-time. Therefore, the following are neglected in this work,

- The steering Ackerman Principle, where both wheels of the front vehicle are assumed to steer equally,
- The rolling resistance, aligning moment, toe and camber angle of each tyre,
- The suspension dynamics,
- The pitch and roll motion of the vehicle,
- The characteristic of the transmission and brake systems is assumed to be a direct transition from the source to the wheel without losses,
- The tire-road interaction is modelled assuming dry road conditions with a friction coefficient of $\mu = 1$, consistent with the controlled testing environment described in the methodology.

The equation of motion for the vehicle model can be derived based on Newton's 2nd Law of motion, which states that net force acting on a vehicle is a product of acceleration and mass. Therefore, looking at the vehicle coordinate frame point of view, the forces are defined as,

$$\sum F_x = ma_x \quad (4-1)$$

$$\sum F_y = ma_y \quad (4-2)$$

Where m is the vehicle's total mass and longitudinal acceleration a_x and lateral acceleration a_y can be classified as velocity components for longitudinal, V_x and lateral, V_y respectively by applying the translational scope of the Newton-Euler principle,

$$a_x = \dot{V}_x - \dot{\psi}V_y \quad (4-3)$$

$$a_y = \dot{V}_y + \dot{\psi}V_x \quad (4-4)$$

Where $\dot{\psi}$ is the yaw rate of a moving vehicle. Looking at the rotational scope of the Newton-Euler principle, the 7 DOF equation of motion for the nonlinear double-track vehicle dynamic model in this work can be expressed as follows,

$$\begin{aligned} F_x = m(\dot{V}_x - \dot{\psi}V_y) = & F_{flx} \cos(\delta_f) - F_{fly} \sin(\delta_f) + \\ & F_{frx} \cos(\delta_f) - F_{fry} \sin(\delta_f) + \\ & F_{rlx} + F_{rrx} - F_{drag} \end{aligned} \quad (4-5)$$

$$\begin{aligned} F_y = m(\dot{V}_y + \dot{\psi}V_x) = & F_{flx} \sin(\delta_f) + F_{fly} \cos(\delta_f) + \\ & F_{frx} \sin(\delta_f) + F_{fry} \cos(\delta_f) + \\ & F_{rly} + F_{rry} \end{aligned} \quad (4-6)$$

While for the yaw moment of the vehicle,

$$\begin{aligned}
 I_z \ddot{\psi} = & l_f (F_{flx} \sin(\delta_f) + F_{frx} \sin(\delta_f) + F_{fly} \cos(\delta_f) + F_{fry} \cos(\delta_f)) - \quad (4-7) \\
 & l_r (F_{rly} + F_{rry}) - \\
 & \frac{w}{2} (F_{flx} \cos(\delta_f) - F_{fly} \sin(\delta_f) + F_{rlx}) + \\
 & \frac{w}{2} (F_{frx} \cos(\delta_f) - F_{fry} \sin(\delta_f) + F_{rrx})
 \end{aligned}$$

In the equation (4-5) to (4-7), $\dot{\psi}$ is the yaw rate and I_z is vehicle's moment of inertia at vertical axis of COG. The distance between the front axle to COG is noted as l_f while the opposite side, to the rear axle is noted as l_r in which total become the vehicle wheelbase. For the other dimension, w is vehicle track width with assumption of equal distributions and δ_f as front wheel steering. While longitudinal and lateral tyre forces are denoted as F_{ija} with $i = (f, r)$ for front and rear position, $j = (l, r)$ for left and right and $a = (x, y)$ as axis with respect to the COG as shown in Figure 4-3.

Another DOFs are from the rotational of each wheel, called as wheel dynamics. The vehicle topology (as in seen in Figure 4-1) came with equally distributed front wheel torque between left and right while rear wheel equipped with ATD. Therefore, the following equation is derived from torque balance along the wheel axis of rotations,

$$\begin{aligned}
 I_w \dot{\omega}_{fl} = & \left(\frac{T_f}{2} \right) - F_{flx} R_w \quad (4-8) \\
 I_w \dot{\omega}_{fr} = & \left(\frac{T_f}{2} \right) - F_{frx} R_w \\
 I_w \dot{\omega}_{rl} = & T_{rl} - F_{rlx} R_w \\
 I_w \dot{\omega}_{rr} = & T_{rr} - F_{rrx} R_w
 \end{aligned}$$

Where I_w is the wheel inertia, ω_{ij} is the wheel rotational rate of each wheels, T_{ij} is the torque distribution to each wheels and R_w is wheel radius.

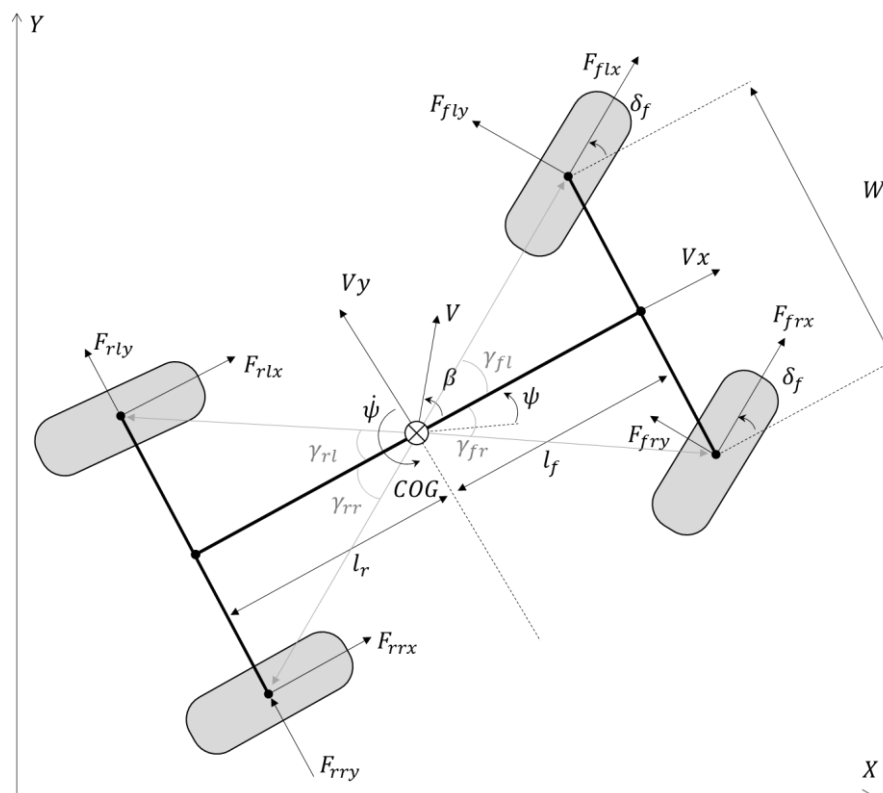


Figure 4-3 Four-wheel double-track FWS vehicle dynamic model

4.2.1 Wheels Velocity Components

Referring to equations (4-5) to (4-8), the tyre forces F_{ija} are derived as functions of tyre slip by using Pacejka's Magic Formula with a load-dependent coefficient [102]. Load-dependent coefficient represented as variable friction between tyres that is essential to capture like in real-world scenario where the normal force acting on tyre varies due to load transfer during acceleration, deceleration, braking, and cornering.

Therefore, to derive the longitudinal and lateral forces, the derivation will initiate with individual wheel velocity components at the centre of the wheel. V_{ijx} and V_{ijy} can be defined as,

$$\begin{aligned}
 V_{flx} &= V \cos(\delta_f - \beta) - \dot{\psi} l_{fl} \sin(\gamma_{fl} - \delta_f) \\
 V_{fly} &= -V \sin(\delta_f - \beta) + \dot{\psi} l_{fl} \cos(\gamma_{fl} - \delta_f) \\
 V_{frx} &= V \cos(\delta_f - \beta) + \dot{\psi} l_{fr} \cos(\gamma_{fr} + \delta_f) \\
 V_{fry} &= -V \cos(\delta_f - \beta) + \dot{\psi} l_{fr} \cos(\gamma_{fr} + \delta_f) \\
 \\
 V_{rlx} &= V \cos(-\beta) - \dot{\psi} l_{rl} \sin(\gamma_{rl}) \\
 V_{rly} &= -V \sin(-\beta) - \dot{\psi} l_{rl} \cos(\gamma_{rl}) \\
 V_{rrx} &= V \cos(-\beta) + \dot{\psi} l_{rr} \cos(\gamma_{rr}) \\
 V_{rry} &= -V \cos(-\beta) - \dot{\psi} l_{rr} \cos(\gamma_{rr})
 \end{aligned} \tag{4-9}$$

Where the vehicle slip is defined as,

$$\beta = \text{atan}\left(\frac{V_y}{V_x}\right) \tag{4-10}$$

And velocity vector as,

$$V = \sqrt{V_x^2 + V_y^2} \tag{4-11}$$

The dimension mentioned in (4-9) is the diagonal distance of the centre of each wheel to COG,

$$l_{ij} = \sqrt{l_i^2 + (W/2)^2} \quad (4-12)$$

With the corresponding angle at each wheel relative to the vehicle's COG of,

$$\gamma_{ij} = \text{atan}\left(\frac{W}{l_i}\right) \quad (4-13)$$

The total slip of each tyre is calculated based on [102] as follows,

$$s_{ij} = \sqrt{s_{ijx}^2 + s_{ijy}^2} \quad (4-14)$$

Where practical slip quantities [102] of longitudinal tyre slip are defined as,

$$s_{ijx} = \frac{V_{ijx} - (\omega_{ij} R_w)}{|V_{ijx}|} \quad (4-15)$$

And lateral tyre slips as,

$$s_{ijy} = (1 + s_{ijx}) \frac{V_{ijy}}{|V_{ijx}|} \quad (4-16)$$

4.2.2 Vehicle Load Transfer

One of the important aspects of the vehicle dynamic model is the load transfer that will significantly impact the vehicle's motion. The load transfer is divided into static load and dynamic load distribution. Under dynamic load is so-called load transfer that is affected due to longitudinal and lateral acceleration.

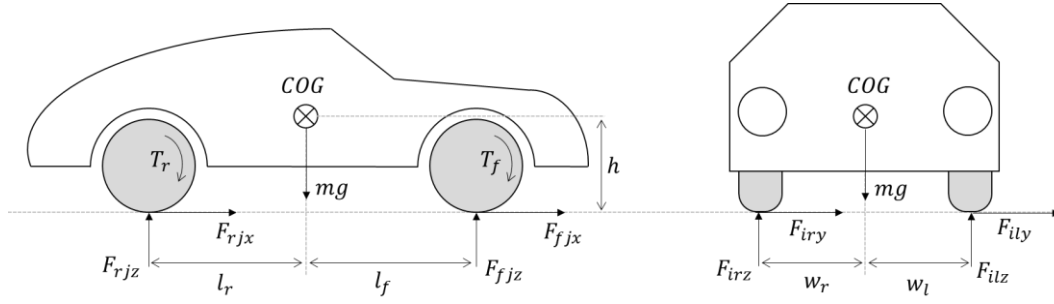


Figure 4-4 Side and front view of the vehicle

As shown in Figure 4-4, unmoving vehicle or static load distribution on each side of the wheels can be derived as,

$$\begin{aligned}
 F_{flz}^o &= \frac{mgl_r w_r}{(l_f + l_r)(w_l + w_r)}, & F_{frz}^o &= \frac{mgl_r w_l}{(l_f + l_r)(w_l + w_r)} \\
 F_{rlz}^o &= \frac{mgl_f w_r}{(l_f + l_r)(w_l + w_r)}, & F_{rrz}^o &= \frac{mgl_f w_l}{(l_f + l_r)(w_l + w_r)}
 \end{aligned} \tag{4-17}$$

In a moving vehicle, the actual load on a tyre differs as it moves depending on longitudinal and lateral acceleration, called the load transfer effect. In the context of minimum time application, capturing the load transfer effect into the vehicle model is crucial.

As mentioned in 4.1, this model is assumed to have a fixed roll axis and a rigid body. Therefore, with the impact of longitudinal, a_x and lateral a_y acceleration, the dynamic load, F_{ly}^Δ and F_{jx}^Δ of the tires can be expressed as,

$$\begin{aligned} F_{lx}^\Delta &= \frac{mhw_r}{(l_f + l_r)(w_l + w_r)} a_x, & F_{rx}^\Delta &= \frac{mhw_l}{(l_f + l_r)(w_l + w_r)} a_x \\ F_{fy}^\Delta &= \frac{mhl_r}{(l_f + l_r)(w_l + w_r)} a_y, & F_{fy}^\Delta &= \frac{mhl_f}{(l_f + l_r)(w_l + w_r)} a_y \end{aligned} \quad (4-18)$$

Where h is height of the COG to the ground as shown in Figure 4-4.

Therefore, with consideration of these load transfers effect during the vehicle movement, the total vertical or normal force acting on each tyre can be expressed as in (4-19), which has a good correlation to experimental data [103],

$$\begin{aligned} F_{flz} &= F_{flz}^o - F_{lx}^\Delta - F_{fy}^\Delta, & F_{frz} &= F_{flz}^o - F_{rx}^\Delta + F_{fy}^\Delta \\ F_{rlz} &= F_{flz}^o + F_{lx}^\Delta - F_{ry}^\Delta, & F_{rrz} &= F_{flz}^o + F_{rx}^\Delta + F_{ry}^\Delta \end{aligned} \quad (4-19)$$

4.2.3 Tyre Model

The only contact point of a vehicle to the ground is through a tyre. Therefore, capturing the tyre dynamic as accurately as possible in this vehicle model is essential. There are three types of tyre models based on an empirical model by detail analysis and data fitting of tyre force characteristics, a semi-empirical model that integrates the analysis of experimental data and theoretical model and another one, the theoretical model itself, by studying the tyre mechanics [104]. The choice of tyre models depends on the specific vehicle dynamics problem being addressed, whether it requires a more detailed theoretical model or an empirical model geared towards practical engineering applications. In the context of this work, an empirical tyre model is much more practical to apply, especially when considering real-time implementation. Many tyre models exist, but the two most popular are based on Pacejka [61] and Dugoff[105]. In this work, a simplified Pacejka tyre model with load-dependent is chosen due to its better accuracy than the Dugoff model, especially in the large slip ratio applicable in this research context [106], [107]. From the total slip in (4-14), the total friction coefficient is derived using Pacejka's Magic Formula with a load-dependent coefficient tyre model as described below,

$$\mu_{ij} = -D_{ij} \sin \left(C \tan^{-1}(B S_{ij}) \right) \quad (4-20)$$

where D_{ij} , C , B are the peak value, shape factor and stiffness factor respectively. Then, assuming symmetric tyre characteristics concerning the longitudinal and lateral position, the total friction force stays within the friction circle [102].

Since this is based on Magic Formula with a load-dependent tyre model, the coefficient of D_{ij} depending on F_{ijz} and each wheel's coefficient of friction μ_{ij} which consider the load sensitivity while coefficients C and B remain constant.

$$D_{ij} = \rho_0 + \rho_2 \frac{F_{ijz} - F_{z0}}{F_{z0}} \quad (4-21)$$

Where ρ_0 is the baseline peak coefficient of friction at the nominal tyre normal force F_{z0} and ρ_2 is the sensitivity of the peak friction coefficient (D_{ij}) to changes in the normalized load. It controls the linear scaling effect of the vertical load on the force peak, contributing to how the tyre's grip or maximum force capacity changes as the load on the tyre varies especially with load transfer.

Therefore, the tyre friction force components are defined as in (4-22), where the negative sign indicates the fact that friction is opposed to the relative motion of the tyre on the path,

$$\mu_{ijx} = -\frac{S_{ijx}}{S_{ij}} \mu_{ij}, \quad \mu_{ijy} = -\frac{S_{ijy}}{S_{ij}} \mu_{ij} \quad (4-22)$$

From the above and calculated normal forces with load transfer as in (4-19), the total longitudinal and lateral forces can be concluded as,

$$F_{ijx} = \mu_{ijx} F_{ijz}, \quad F_{ijy} = \mu_{ijy} F_{ijz} \quad (4-23)$$

4.2.4 Friction Circle

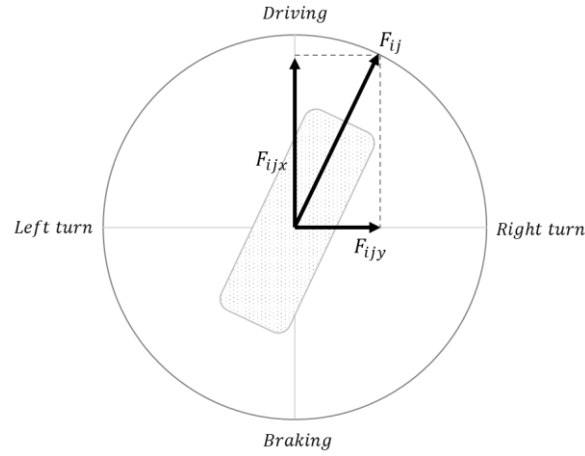


Figure 4-5 Friction circle of a wheel

The friction circle is crucial in this work as it represents the limit of the tyre's grip and helps understand how it would generate forces during different driving conditions, primarily involving longitudinal and lateral forces. When a tyre operates near its limit, the sum of these forces, F_{ij} cannot exceed the maximum frictional capability,

$$F_{ij} = \sqrt{F_{ijx}^2 + F_{iyy}^2} \quad (4-24)$$

Which then the tipping point of the maximum resultant forces, as shown in Figure 4-5, is always within the friction circle with the assumption that a homogeneous tyre with symmetric characteristics in both directions as shown below,

$$\left(\frac{F_{ijx}}{F_{ijx}^{max}} \right)^2 + \left(\frac{F_{iyy}}{F_{iyy}^{max}} \right)^2 = 1 \quad (4-25)$$

4.3 Path Model

For the road or path model, the path is considered as a strip defined by the centerline coordinates and the lateral deviation of the vehicle from this centerline, as described in [108]. The curvilinear coordinate system is used to represent the path, where the path centerline is described in terms of the arc length, S and the path curvature, K , as shown in Figure 4-6.

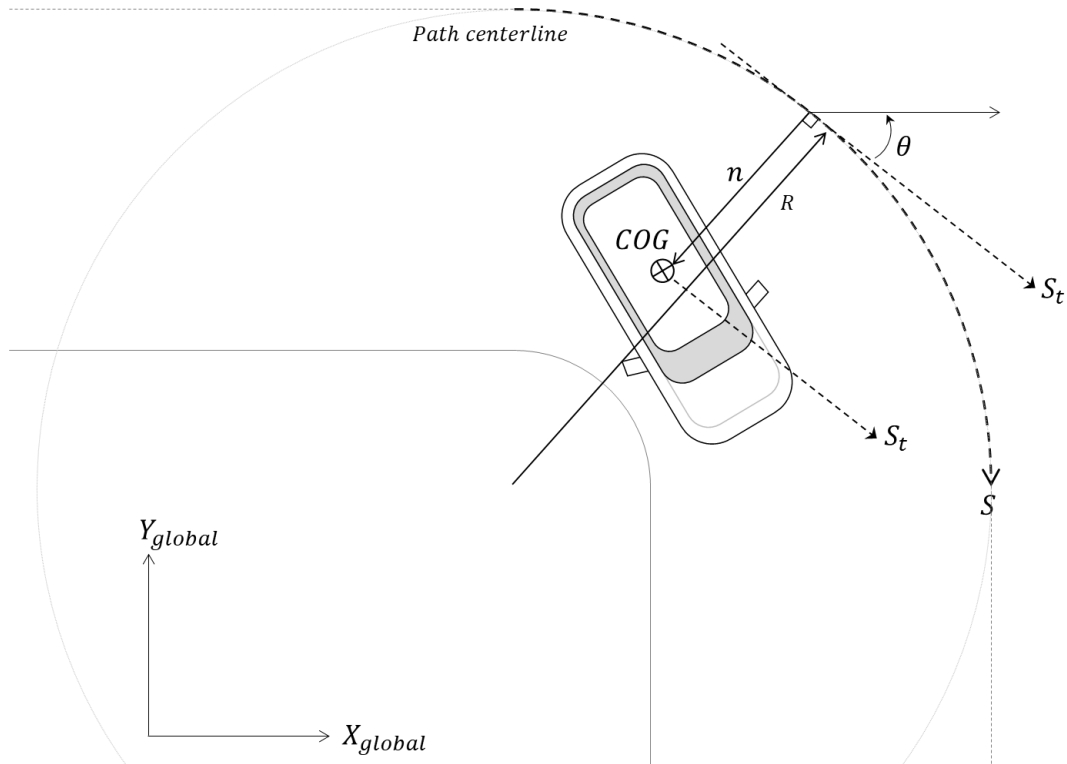


Figure 4-6 Vehicle top view in curvilinear coordinate for path model

In the curvilinear coordinate system, the vehicle's position along the path is determined by the arc length, S , which measures the distance travelled along the centerline, and the curvature function $K(S)$, which defines how sharply the path curves at each point. The heading angle of the path at any given point is represented by $\theta(S)$, and the vehicle's position on the Cartesian coordinates is denoted by (X_{global}, Y_{global}) .

The path is generated by defining the curvature profile $K(S)$ along different segments of the track where straight path segments are defined with a curvature of zero, $K = 0$ and curved path segments are defined with a constant curvature defined as the inverse of the radius,

$$K = \frac{1}{R} \quad (4-26)$$

Where R is the radius of the turn. The curvature profile is constructed by concatenating straight and curved segments to represent the full path. Once the curvature profile is defined, the following differential equations are used to compute the path's heading angle and coordinates of the path,

$$\frac{d\theta}{dS} = K(S) , \quad \frac{dx}{dS} = \cos(\theta) , \quad \frac{dy}{dS} = \sin(\theta) \quad (4-27)$$

In this work, there are two path shapes with 6 meters trackwidth with 20 meters radius were considered, which are a 90-degree right-hand turn as in Figure 4-7 and a multi-corners extended path that covers 90-degree turn, U-turn, S-shape and representing double lane change as shown in Figure 4-8. Details parameters as in the Appendix A.4.1 and Appendix A.2.2.

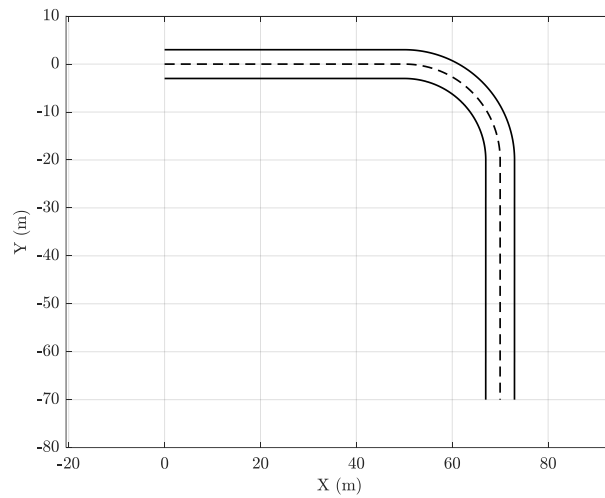


Figure 4-7 90-degree right-hand corner short path

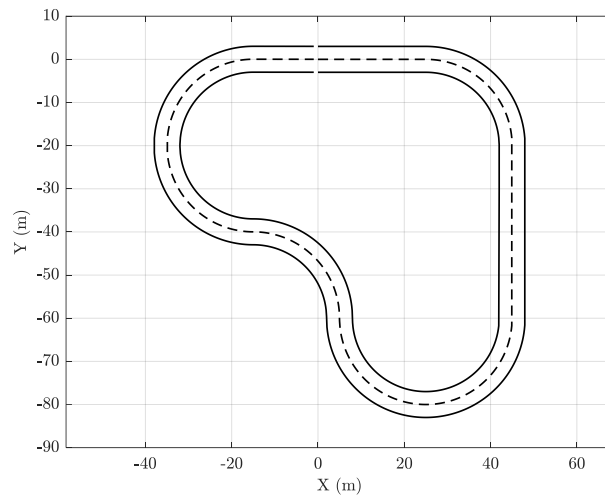


Figure 4-8 Multi-corners extended path

Another crucial path shape used to validate the developed controller is a long-distance path based on the Sepang International Circuit in Malaysia, as illustrated in Figure 4-9. This circuit spans a total length of 5,543 meters with a track width ranging from 16 to 22 meters. For this simulation, a consistent track width of 16 meters was considered, representing the narrowest and most challenging scenario for testing. The Sepang Circuit is renowned for its demanding layout, featuring a mix of long straights, low-speed technical corners, and high-speed sweeping bends, making it an ideal testbed to evaluate the trajectory and performance of the developed motion planning and control controller.

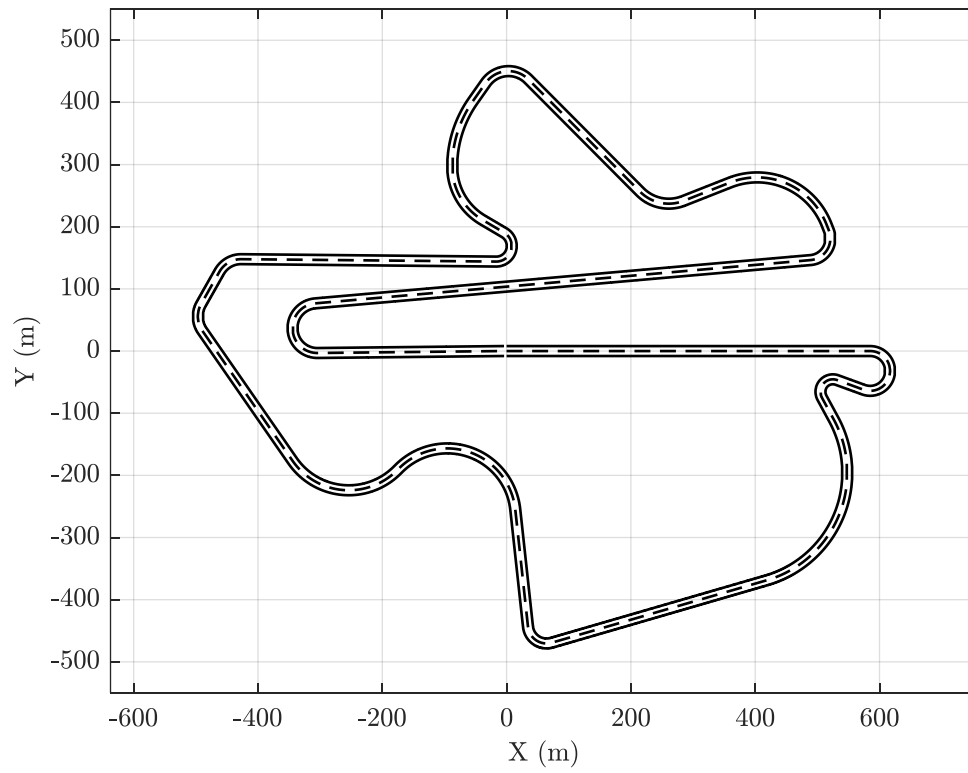


Figure 4-9 Full length of Sepang International Circuit

4.4 Summary

This chapter presented the vehicle and path models used in developing and testing the PLAPC. A 7 DOF double-track vehicle model was selected to balance accuracy and computational efficiency for real-time implementation. Key vehicle dynamics such as load transfer, individual wheel forces, and tyre slip were incorporated, with Pacejka's load-dependent tyre model providing a realistic representation of tyre behaviour under varying loads. However, for real-time online implementation in IPG CarMaker, the vehicle model DOF was reduced to 3 DOF by eliminating the wheel dynamics (Equation (4-8)) to ensure real-time implement-ability.

The path models included a range of path geometries for controller validation, such as a short path 90-degree turn, extended path of multi-corner scenarios, and long distance of the Sepang International Circuit provided a challenging test environment with diverse straights, sharp corners, and sweeping bends. This combination of detailed vehicle dynamics and complex track layouts enabled comprehensive validation of the PLAPC's performance.

Overall, this chapter established a strong foundation for testing the developed controller in high-fidelity simulations, capturing essential vehicle dynamics while maintaining real-time feasibility,

5 Optimal Solutions

5.1 Introductions

The Optimal Control Problem (OCP) formulation in this work aims to generate optimal solutions for both trajectory and velocity profiles. It solves the optimisation problem offline across the predefined path at once to generate an ideal trajectory and velocity profile that satisfies the defined objective functions and constraints. The multiple shooting method is employed to discretize the continuous trajectory optimisation problem, transforming it into a series of NLP problems. The system dynamics are approximated within each shooting interval using the explicit Runge-Kutta 4th Order (RK4) integration scheme, which discretizes the continuous dynamics over smaller steps to ensure numerical accuracy, stability and robustness [77]. In this work, the OCP was developed using MATLAB and CasADi.

The resulting solution represents an ideal trajectory and velocity profile for the given path, serving as a reference for real-time simulation studies in this work. Additionally, this offline-generated trajectory acts as a benchmark for evaluating the performance of the developed PLAPC, providing a comparison to measure how closely the real-time implementation aligns with the offline optimal solution. By leveraging this ideal trajectory, the study establishes a validation process for the proposed controller in both offline and online real-time scenarios, highlighting its robustness and adaptability under varying conditions. This approach provides a foundation for validating the performance of the PLAPC in various scenarios and for further development.

5.2 Optimal Problem Formulations

In this work, the OCP is formulated using a curvilinear coordinate system as shown in Figure 5-2, where the distance travelled along the path (arc length) is used as the independent variable instead of time. This formulation differs from traditional Cartesian-based approaches by solving the optimisation problem directly in curvilinear space. Using curvilinear coordinates provides several advantages, including reducing the number of problem variables, simplifying path constraints (For example, road boundaries), and the ability to evaluate vehicle states and control inputs at any point along the given path [75]. Figure 5-1 shows the detail parameters of curvilinear coordinates and Figure 5-2 shows the coordinate frame of it where S is distance travelled along the path.

A key benefit of using curvilinear coordinates is that the path curvature remains constant and independent of the control inputs, as distance is no longer a function of time. Additionally, the progress distance becomes redundant and can be omitted from the state vector, simplifying the model and improving computational efficiency in space-curvilinear formulations [53].

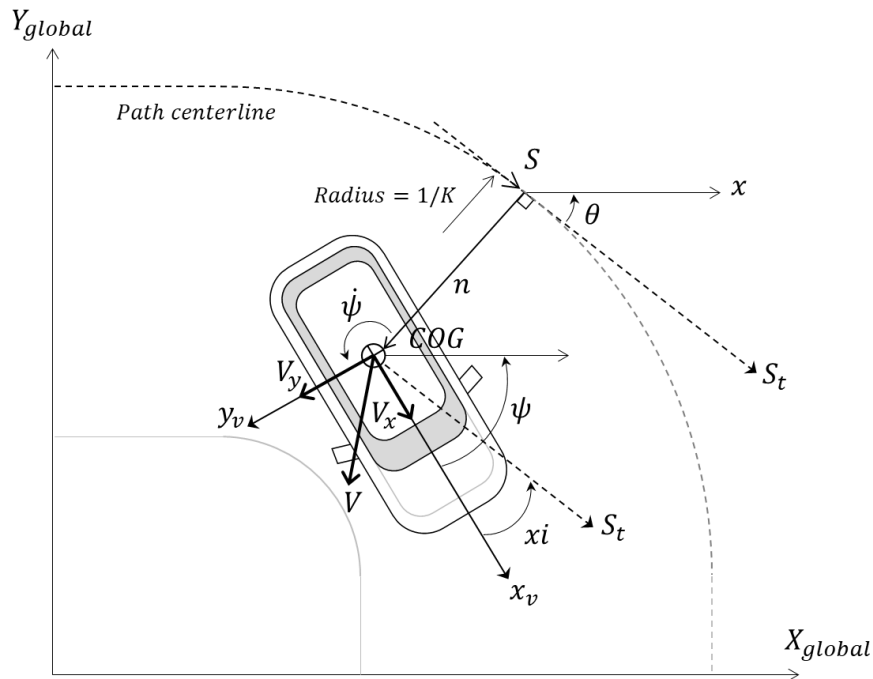


Figure 5-1 Vehicle top view for curvilinear coordinate representation

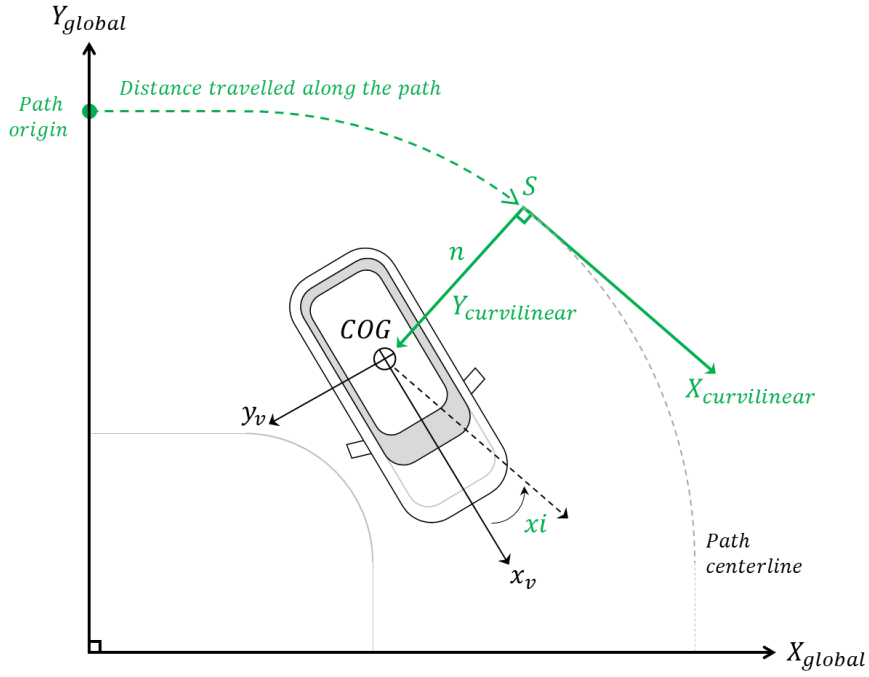


Figure 5-2 Curvilinear coordinate frame representation

5.2.1 Spatial Dynamics

Prior to the change of independent variable, the derivation in time-dependent started with resolving velocity in S_t direction as shown in Figure 5-1, where

$$\frac{ds}{dt} - n \frac{d\theta}{dt} = V_x \cos(xi) - V_y \sin(xi) \quad (5-1)$$

Where V_x is the longitudinal velocity and V_y is the lateral velocity, K is the path curvature at a given point (inverse of radius, R), which is equivalent to the inverse of the corner radius, n is the lateral deviation from vehicle COG to the centre line of the path and xi is the angle deviation from the tangent of the path centre line as shown in Figure 5-2 and θ is the path heading angle where,

$$\frac{d\theta}{dt} = \frac{d\theta}{ds} \cdot \frac{ds}{dt} = K \frac{ds}{dt} \quad (5-2)$$

Which then gives,

$$\frac{ds}{dt} = \frac{V_x \cos(xi) - V_y \sin(xi)}{1 - nK} \quad (5-3)$$

Then the rate of change of lateral error, n can be expressed as,

$$\frac{dn}{dt} = V_x \cos(xi) - V_y \sin(xi) \quad (5-4)$$

And the rate of change for angle deviation, xi is given by,

$$\frac{dxi}{dt} = \frac{d\psi}{dt} - K \frac{ds}{dt} \quad (5-5)$$

It is known that the “distance travelled, S_f ” and “time, t ” are related independent variables which could be described as,

$$dt = \frac{dt}{ds} ds = S_f(s) ds \quad (5-6)$$

In which S_f can be derived in reference to Figure 5-1 and Equation (5-3),

$$S_f = \left(\frac{ds}{dt} \right)^{-1} = \frac{1 - nK}{V_x \cos(xi) - V_y \sin(xi)} \quad (5-7)$$

Which therefore, multiplying both sides with S_f for a change of independent variables to express the equation of motion as a function of travelled distance instead of time,

$$\frac{dx}{ds} = f(x, u) \frac{dt}{ds} = f(x, u) S_f \quad (5-8)$$

Where $f(x, u)$ is the function contains the states and control inputs as stated in Equation (4-5) until (4-8).

5.2.2 Numerical Discretization and Variables

The spatial dynamics described in the previous section form the basis of the OCP. To solve this OCP, the system dynamics are discretised over the spatial domain s , instead of in the time domain t . The discretisation was implemented using the 4th-order Runge-Kutta (RK4) technique, ensuring precise numerical integration of the system dynamics over the prediction horizon. RK4 provides a balance between computational efficiency, solution accuracy and ease of implementation, making it suitable for real-time applications within the high-fidelity simulation framework.

Therefore, the states, $st^{off}(s)$ and controls $in^{off}(s)$ for the vehicle model in this optimal control problem formulation are defined as follows,

$$st^{off}(s) = \{V_x(s), V_y(s), \dot{\psi}(s), S_n(s), xi(s), \omega_{ij}(s), \psi(s)\}^T \quad (5-9)$$

$$in^{off}(s) = \{T_f(s), T_{rl}(s), T_{rr}(s), \delta_f(s), \delta_r(s)\}^T \quad (5-10)$$

Moreover, additional variables named auxiliary variables other than states and controls for acceleration and tyre load sensitivity,

$$aux^{off}(s) = \{a_x(s), a_y(s), D_{ij}(s)\}^T \quad (5-11)$$

5.2.3 Constraints

The constraints in this optimal control problem are formulated to ensure the vehicle's dynamics remain feasible and within safe limits. The first constraint is to ensure the tyre forces stay within the available friction limits as described in Sub-Chapter 4.2.4, which can be expressed as,

$$\left(\frac{F_{ijx}}{F_{ijx}^{max}} \right)^2 + \left(\frac{F_{ijy}}{F_{ijy}^{max}} \right)^2 \leq 1 \quad (5-12)$$

This constraint limits longitudinal and lateral tyre forces to prevent tyre slip beyond the tyre friction capabilities. Another important aspect related to the tyre model, assuming equal friction coefficients in longitudinal and lateral directions, is the load-dependent friction coefficient relationship defined by the Magic Formula, which can be defined as,

$$D_{ij} - \rho_0 + \rho_2 \frac{F_{ijz} - F_{z0}}{F_{z0}} = 0 \quad (5-13)$$

where D_{ij} is the tyre friction demand of each tyre, ρ_0 is the base friction coefficient, ρ_2 represents the load sensitivity factor that adjusts the peak friction coefficient based on the change in vertical load, F_{ijz} and F_{z0} is the nominal vertical load.

In addition, more constraints are needed for lateral and longitudinal accelerations since both have been defined as auxiliary variables in this work, as in Equation (5-11). This is to incorporate the effects of load transfer and ensure consistent dynamic behaviour, which is defined as follows,

$$a_x - \dot{V}_x = 0, \quad a_y - \dot{V}_y = 0 \quad (5-14)$$

Here, a_x and a_y are longitudinal and lateral accelerations, while $\dot{V}_x$ and $\dot{V}_y$ are the corresponding rates of change of longitudinal and lateral velocities.

In addition, in order to ensure the vehicle stays within the road limits, the vehicle's lateral position n is defined using a curvilinear approach is constrained within the left W_p^l and right W_p^r boundaries of the path which can be addressed as,

$$W_p^l \leq n \leq W_p^r \quad (5-15)$$

Additional constraints are implemented on the initial and final conditions, as well as the physical limits of states and control inputs (such as minimum and maximum steering angles, and torques). These constraints parameters are described in further detail in Appendix A.1.

5.2.4 Objective Functions

This chapter considers two objective functions: one is extreme on both ends, one minimises time, and another optimises comfort as summarized in Table 5-1.

Table 5-1 Objective functions for Optimal Solutions

Type of Objective Functions	Descriptions
Minimised time, J_t^{off}	Minimising total time across the whole track at once.
Optimised comfort, J_c^{off}	A multi-objective function balancing time minimisation and comfort by penalizing acceleration and jerk (rate of acceleration), the ISO 2631-1 [21] is referred to define the comfort level.

The first one is in terms of time as one of the objective functions, which could be expressed in distance as an independent variable,

$$J_t^{off} = \min_{x,u} \sum_{k=0}^{N-1} S_f \quad (5-16)$$

Where N is the total number of discrete steps over the entire path length and S_f as mentioned in Equation (5-7). In this approach, distance is used as the independent variable, and the objective function is expressed in terms of the distance travelled S_f meaning to minimise time given a set of path lengths from the start ($s = 0$) to the end of the manoeuvre ($s = s_{final}$).

Another objective function described in this work is related to comfort optimisation. This objective inherently contradicts the goal of minimising travel

time, as it prioritizes smoothness and ride comfort. This is a crucial consideration for AVs, particularly for public usage, where user comfort and ride quality are paramount. The comfort aspect can be expressed as,

$$J_c^{off} = Q_t J_t + \min_{x,u} \sum_{k=0}^{N-1} (Q_{cax}(a_x^2) + Q_{cay}(a_y^2) + Q_{cdax}(\dot{a}_x^2) + Q_{cday}(\dot{a}_y^2)) \quad (5-17)$$

Where Q_{cax} and Q_{cay} are the weight for longitudinal and lateral acceleration and Q_{cdax} and Q_{cday} are the weight for longitudinal and lateral jerk which is the differentiation of the acceleration components. Q_t represent a small weight for time-related terms to balance optimising time and achieving comfort. This formulation penalizes rapid changes in acceleration and jerk, ensuring smoother trajectories and transitions on both longitudinal and lateral during vehicle motion that promote better comfort.

Therefore, the total objective for this OCP can be expressed as,

$$J^{off} = J_i^{off} + J_{reg}^{off}, \quad i = t, c \quad (5-18)$$

subject to,

$$x_0 = x_{initial}, u_0 = u_{initial}$$

$$x_{k+1} = f_d(x_k, U_k)$$

$$x_k^{min} \leq x_k \leq x_k^{max}$$

$$u_k^{min} \leq u_k \leq u_k^{max}$$

$$y_k^{min} \leq y_k \leq y_k^{max}, k = 0, \dots, N - 1$$

5.2.5 Regularisation and Scaling

As mentioned in [75], [109], [110] solving an optimal control problem may result in singularities or oscillations in the solution, leading to undesirable noise or non-smooth behaviour in the control inputs. Such a situation can negatively impact the performance and feasibility of the generated trajectory. Regularisation terms are often introduced into the objective function to mitigate this issue. These terms smooth out the control inputs and ensure that the solutions remain well-behaved in terms of control effort and dynamic feasibility.

Regularisation can take several forms. The simplest approach involves adding a penalty for the square of the control inputs, effectively encouraging reduced control effort. More advanced methods target the solutions' non-smoothness by penalising the control inputs' first derivative, thereby smoothing their progression.

In this work, regularization terms are applied to the control inputs U to ensure smooth trajectories and feasible control signals. The specific regularisation technique involves penalizing the derivatives of these variables, as expressed in Equation (5-19). The term includes two penalty terms: one penalizes the magnitude of the control inputs to reduce aggressive actions by squaring the inputs, and the other penalizes the difference between consecutive control inputs to avoid sudden changes. The choice of regularization weight parameters is crucial; it must be reasonable enough to avoid significantly altering the original solution to the optimal control problem while still effectively addressing noise and oscillations. The parameters used in this study are detailed in Appendix A.3 and A.4.

The inclusion of these regularization terms enhances all objective functions used in this work, contributing to smoother trajectories that align with the desired performance objectives. The regularization terms can be expressed as,

$$J_{reg}^{off} = \min_u \sum_{k=0}^{N-1} \left(U_k^T Q_u U_k + (U_k - U_{k-1})^T Q_{du} (U_k - U_{k-1}) \right) \quad (5-19)$$

where U_k represents the control input at step k , U_{k-1} is previous control input, Q_u is the weight matrix for control effort and Q_{du} is the weight matrix for control input smoothness. The first term penalizes large control inputs, while the second term ensures minimal changes between consecutive inputs, avoiding sudden change and smoothness of input. These weight matrices are listed in Appendix A.3 and A.4.

In terms of scaling of the OCP, the performance of optimisation solvers is significantly influenced by the scaling of decision variables, a well-documented limitation of solution algorithms [75]. As highlighted in [111], no universal best practice for scaling exists, but several approaches have been proposed. A common method involves using non-dimensional variables normalized based on the problem's magnitude to improve solver performance. In this study, all decision variables are scaled to lie within the range of $[-1,1]$, as this range has been observed to enhance computational efficiency and reduce solve times.

When the maximum value of a variable is known, scaling is achieved by dividing the variable by its maximum value. This straightforward approach ensures uniform scaling across decision variables, facilitating numerical stability and convergence during optimisation. Additionally, path constraints are scaled to ensure their values are of a similar order of magnitude as the decision variables, further improving the solver's performance. This scaling strategy standardises the problem and allows the solver to process the optimisation more effectively by reducing condition number issues in the underlying matrices.

5.3 Optimal Solution Generations

5.3.1 Introductions

This subchapter presents the results of optimal solutions obtained through offline optimisation, focusing on two objectives: minimising travel time and optimising comfort. These objectives were achieved by solving the OCP formulated in Section 5.2.

5.3.2 Sensitivity Study of Comfort Weight Factor

This sub-chapter will describe the impact of varying all comfort weight factors of the optimal solution over a 90-degree right-hand corner. The initial starting position and initial longitudinal velocity are all the same, which are set at $(X, Y) = (0, 3)$, starting distance of $S = 0$ and initial velocity of 20 m/s .

As Equation (5-17) mentioned, the comfort objective function includes lateral and longitudinal acceleration and rate of the acceleration (jerk). A sweep of comfort weight factors was performed in this study in order to identify the best trade-off between comfort and time while ensuring the weighted RMS accelerations stayed within “Not Uncomfortable” zone. This zone was classified based on ISO 2631-1 [22] which provides guidelines for evaluating human exposure to vibrations, specifically those that affect comfort. The weighted RMS accelerations are calculated to assess vibration effects on human comfort. Based on the solved accelerations data, it is processed using frequency weightings (such as W_d for X and Y axis of the seat surface) to account for how sensitive humans are to vibrations at different frequencies. The accelerations are transformed into the frequency domain using FFT, where the weightings are applied. Afterwards, the data is converted back to the time domain, and the RMS is calculated to give a single value representing the overall vibration intensity. This weighted RMS value is compared to comfort thresholds provided in the ISO standard, as shown in Table 5-2, which classify the comfort level based on the intensity of the vibrations.

Table 5-2 Comfort level based on weighted RMS accelerations as mentioned in ISO 2631-1

Weighted RMS accelerations range	Classifications
$a_{rms} < 0.315$	Not uncomfortable
$0.315 \leq a_{rms} \leq 0.63$	A little uncomfortable
$0.500 \leq a_{rms} \leq 1.00$	Fairly uncomfortable
$0.800 \leq a_{rms} \leq 1.60$	Uncomfortable
$1.25 \leq a_{rms} \leq 2.50$	Very uncomfortable
$a_{rms} > 2.00$	Extremely uncomfortable

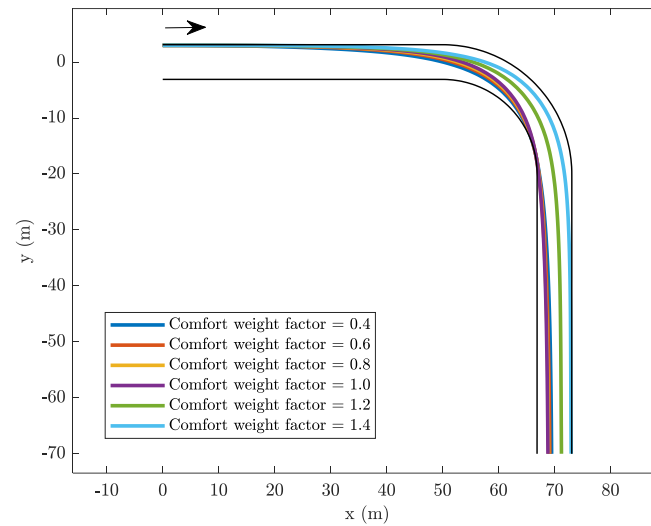


Figure 5-3 Trajectories comparison with varying comfort weight factor

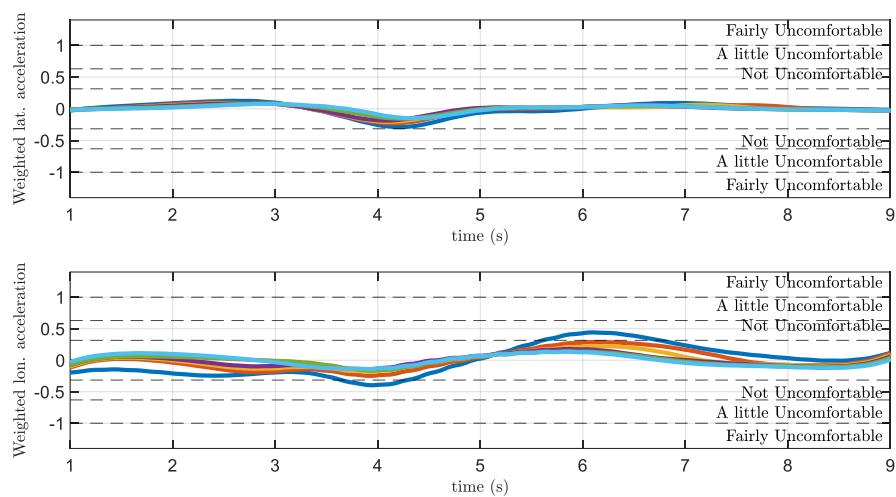


Figure 5-4 Weighted accelerations impact on varying comfort scale factor

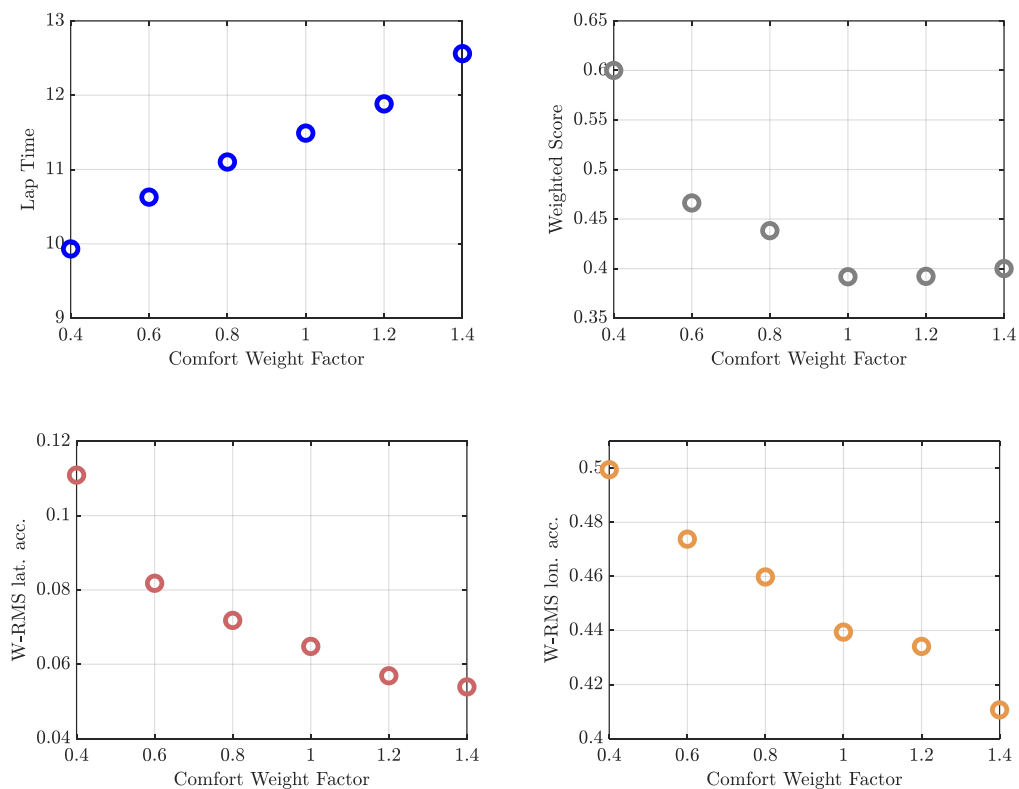


Figure 5-5 Weighted score comparison on varying comfort scale factor

As seen in Figure 5-5, a weight factor of 1.4 offers the lowest W-RMS acceleration values for both longitudinal and lateral, indicating the highest comfort level. It comes with a trade-off in the form of the slowest lap time. On the other hand, a weight factor of 1.2 provides nearly the same level of comfort, with only slightly higher W-RMS acceleration values, but with the advantage of a slightly better lap time of 11.8331 seconds. Comparatively, although a weight factor of 0.4 results in the shortest lap time, it also leads to the highest W-RMS values for both accelerations, implying a significantly less comfortable ride which it's out of the "not uncomfortable" zone as seen in Figure 5-4. Based on these considerations, the weight factor of 1.0 strikes the best balance between comfort and performance, as seen in the weighted score considering 40% on the lap time and 60% on the combined comfort.

5.3.3 Comparison Results Between Objective Functions

As mentioned in the problem formulation in Sections 5.2.4, two objective functions were considered which are minimised time and optimised comfort problems. Upon the sensitivity analysis conducted in Sub-Chapter 5.3.2, here is the comparison between each objective function with the same initial starting position and initial velocity in terms of its trajectory, lap time and comfort level.

Table 5-3 Comparison results between objective functions for optimal solutions

Objective function	Lap time (s)	Peak ax_{wrms}	Peak ay_{wrms}	Total Energy usage (kWh)	Solved time (s)
Minimised time	5.9819	2.0577	2.5745	0.06730	44.1441
Optimised Comfort	11.2311	0.1866	0.0645	0.04735	51.2680

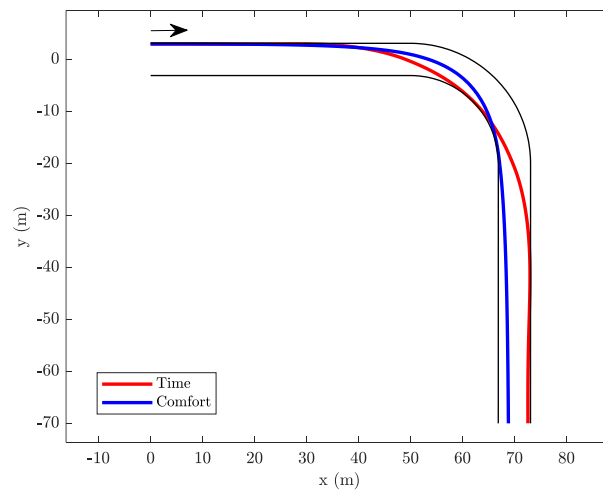


Figure 5-6 Vehicle trajectory comparison of all objective functions

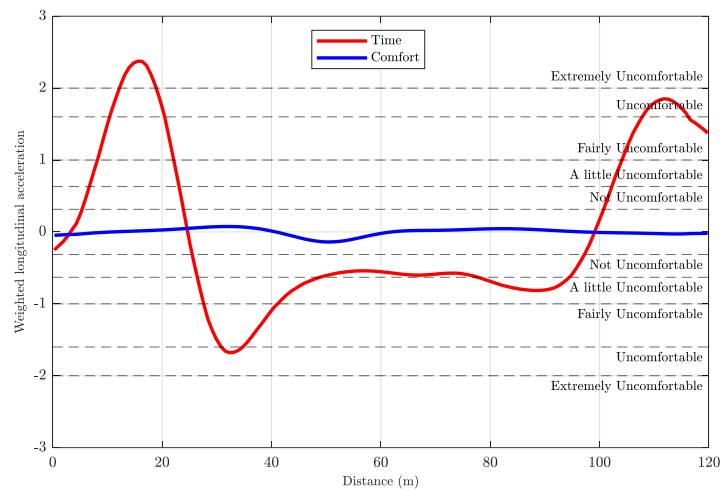


Figure 5-7 Weighted lateral acceleration comparison of all objective functions

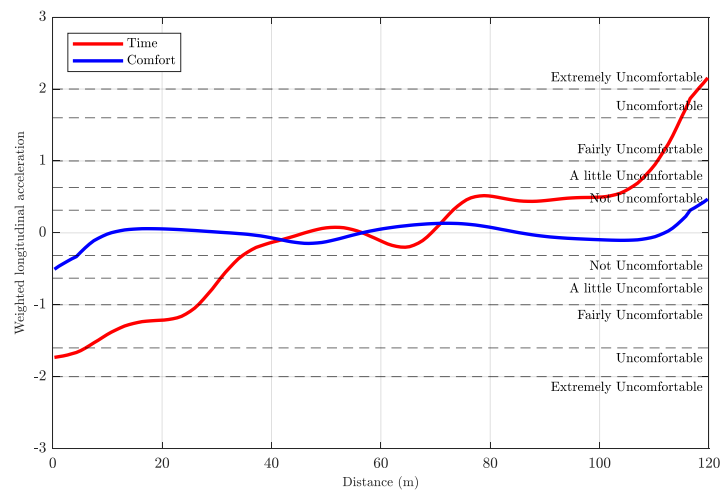


Figure 5-8 Weighted longitudinal acceleration comparison of all objective functions

The results presented in Figure 5-6 to Figure 5-8 and Table 5-3 highlight the differences in vehicle trajectory and performance metrics across two contradicting objectives functions. The time optimal objective achieved the fastest lap time (5.9819 seconds) but with the highest energy consumed (0.0673 kWh) and the most aggressive weighted RMS acceleration values (2.0577 m/s²)

longitudinal and 2.5745 m/s^2 lateral). This approach maximises performance at the expense of both comfort and energy efficiency.

In contrast, the Optimised Comfort objective significantly reduces peak weighted RMS accelerations (0.1866 m/s^2 longitudinal and 0.0645 m/s^2 lateral) to provide a smoother driving experience. However, this comes with the trade-off of the longest lap time (11.2311 seconds) and lower energy usage (0.04735 kWh). This strategy conserves energy while maintaining moderate performance and comfort. The findings demonstrate how varying the optimisation focus can directly impact vehicle dynamics, with each objective serving a distinct purpose based on the priorities of time or comfort.

Another important point to note is the significantly high time required to solve each objective function. This observation aligns with the discussion in Section 1.1, which highlights that trajectory optimisation tasks often demand substantial computational resources, making it challenging to adapt to changing courses during AV navigation.

5.4 Summary

This chapter comprehensively presented the formulation, implementation, and evaluation of optimal solutions for two contradicting objective functions. The focus was to evaluate differences in between time efficiency and optimised for comfort, which is critical for further evaluation of PLAPC.

The chapter began with the formulation of the OCP in curvilinear coordinates, leveraging spatial dynamics to simplify constraints and improve computational efficiency. Two key objective functions of minimising time and optimising comfort were elaborated, with detailed explanations of the mathematical formulations. Constraints such as tyre friction limits, load transfer, and path boundaries were introduced to ensure that the solutions remain physically feasible and practical.

Finally, the comparison of results between objective functions demonstrated the effectiveness of the optimal control formulation in generating suitable velocity and trajectory references. This approach provided clear insights into the trade-offs between minimum time and optimised comfort objectives, establishing essential baseline references for validating the subsequent real-time implementation of the PLAPC controller. However, obtaining these optimal solutions proved computationally intensive, highlighting practical limitations for direct real-time application such as the requirement for re-generating or re-planning and emphasizing the necessity for simplified methods for real-time implementation.

6 High-Fidelity Simulation Setup

6.1 Introduction

Before proceeding to the online implementation, explaining the high-fidelity simulation setup used in this work is crucial. The real-time simulations were conducted using IPG CarMaker, a recognized and proven vehicle testing and experimentation simulation platform. This platform facilitates the validation of developed controllers closer to realistic driving conditions.

IPG CarMaker provides a real-time simulation environment for developing and testing various vehicle systems, including autonomous vehicles, ADAS, and vehicle dynamics [112]. Its comprehensive vehicle models and real-time capabilities make it ideal for virtual testing, enabling efficient and accurate studies without relying on physical prototypes. This makes it particularly beneficial for evaluating the PLAPC, ensuring that its control performance is tested in a realistic setting, considering the complexities of vehicle behaviour and environmental interactions. This high-fidelity simulation framework allows for a detailed analysis of vehicle responses and serves as a testbed for potential hardware implementation in future development.

The IPG CarMaker software was integrated with MATLAB/Simulink, allowing the implementation of the PLAPC in the NMPC framework. The CasADi code for solving the NMPC optimisation problem was embedded within a MATLAB/Simulink PLAPC controller block, as depicted in Figure 6-1. This integration enabled real-time execution for the testing of the developed controller. Furthermore, the IPG CarMaker model used in this research has been validated for testing of full-scale vehicles in real-time experiments, as referred in [88],[113], making it a reliable platform for high-fidelity simulation and testing evaluation. The vehicle and simulation parameters used in this work are detailed in Appendix A. This simulation setup provides a robust framework for analysing the performance of the proposed controller in real-time and under various driving scenarios.

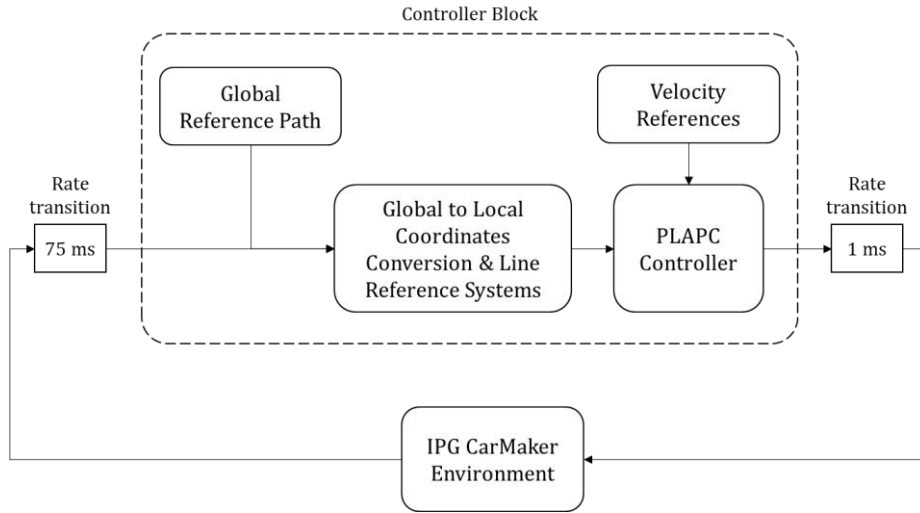


Figure 6-1 Simulation setup for PLAPC in the IPG CarMaker

6.2 Local Coordinates Conversion

In this work, the global coordinates defined in IPG CarMaker were converted into local coordinates relative to the vehicle's position along the given path as shown in Figure 6-2. This approach ensures that the vehicle's position is referenced to the nearest point on the path, simplifying the implementation and improving numerical stability by avoiding large numbers using global coordinates. Adopting this method, all references are tied to the path-aligned local coordinate system, treating the path's nearest point as the origin (0,0). This local reference frame is crucial for maintaining consistent, path-centric references. The local coordinates conversion started with identifying the nearest point of the vehicle on the path (X_p, Y_p) where the Euclidean distance between the vehicle's current global position (X_g, Y_g) and each point on the path is calculated as follows,

$$D = (X_p - X_g)^2 + (Y_p - Y_g)^2 \quad (6-1)$$

This calculated distance enables the determination of the closest point on the path, which serves as the reference for subsequent calculations in the local coordinate system. The path used in this work is defined with a step distance of

0.01 meters, which is fine enough to ensure smooth conversion to the local coordinate system. This step is a foundational component of the developed controller, facilitating precise calculations and references that are made relative to the vehicle's instantaneous position.

The index position corresponding to the closest distance is determined by finding the minimum value of D . This process identifies the nearest reference coordinate (X_n^{ref}, Y_n^{ref}) , which serves as the basis for local coordinate conversion. To further ensure continuity and avoid angular discontinuities, the vehicle's global yaw angle is wrapped to the range of $[-\pi, \pi]$ Radians. Therefore, the transformation from global coordinates to the local frame can be expressed as,

$$X_{local} = \cos(-\varphi_n^{ref})(X_g - X_n^{ref}) - \sin(-\varphi_n^{ref})(Y_g - Y_n^{ref}) \quad (6-2)$$

$$Y_{local} = \sin(-\varphi_n^{ref})(X_g - X_n^{ref}) + \cos(-\varphi_n^{ref})(Y_g - Y_n^{ref})$$

This transformation ensures that the vehicle's position and orientation are consistently referenced relative to the local path, facilitating accurate motion planning and control. By transforming the vehicle's global position and orientation into the local frame aligned with the reference path, the developed controller could utilize the lateral position, Y_{local} to be an additional safety factor for the vehicle to stay within path boundaries. This approach simplifies the control objectives and enhances controller accuracy by consistently referencing the nearest point on the path.

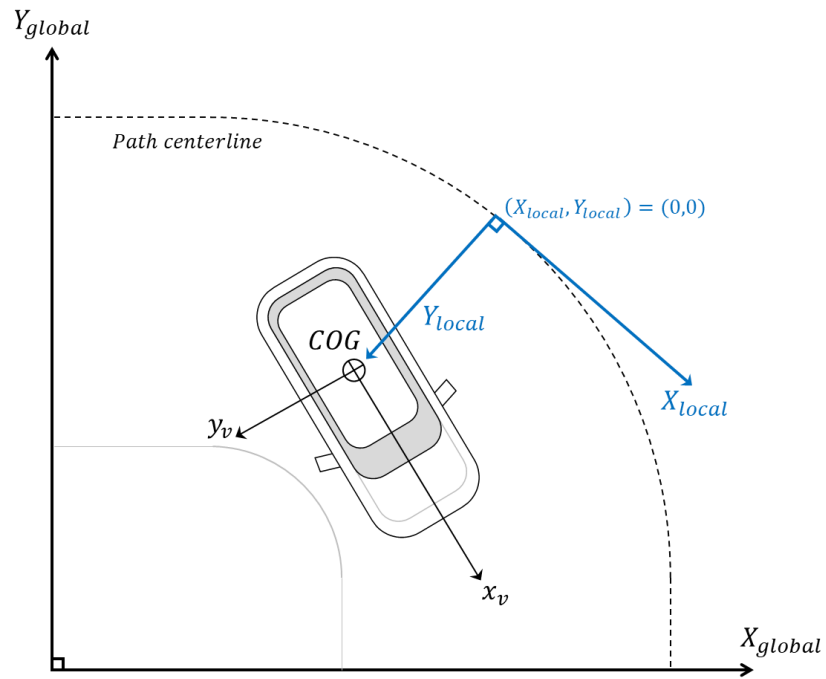


Figure 6-2 Local path coordinate frame representation

6.3 Velocity Profiles Generations

This section presents the velocity profiles as reference inputs for both PTC and PLAPC simulations. Velocity profiles are critical as they define the desired velocity along the path, serving as benchmarks to evaluate the controller's performance. Table 6-1 shows all type of velocity profiles generated that are used in this work.

The velocity profiles were generated using two approaches. The first approach derives from the optimal solution obtained in Chapter 5, based on the specified objective functions, such as minimum time or optimised comfort. However, generating these profiles is computationally intensive.

The second approach is quicker (less than one second) and automatically determines a safe velocity profile by considering the path's curvature. In this method, velocity is calculated based on the curvature or radius of the path, ensuring safe operation. This curvature-based velocity profile can be derived based on the vehicle following a curved path, it experiences lateral accelerations due to the centripetal forces required to stay on the curve. Based on the Newton's 2nd law, this force is given by,

$$F_c = ma_c = m \frac{V^2}{R} \quad (6-3)$$

Where F_c is the centripetal force, m is vehicle mass, V is vehicle's velocity and R is path radius curvature. Therefore, for a vehicle to maintain grip and not slide off the path, the available friction force between the tyres and the path must provide this required centripetal force. The tyre friction forces can be expressed as,

$$F_{friction} = \mu mg \quad (6-4)$$

Where μ is the tire coefficient of friction to the road and g is the gravitational acceleration.

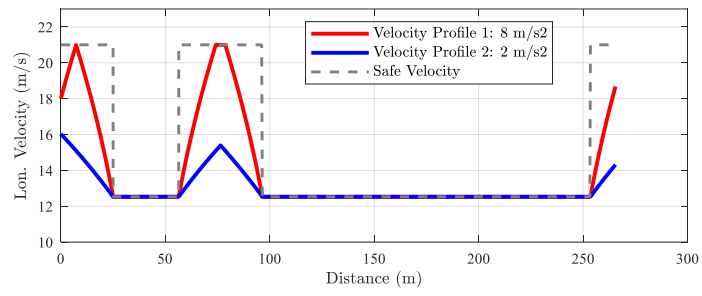
Since the friction force is the maximum force available to sustain the turn before the vehicle starts skidding, therefore, the relation between Equation (6-3) and (6-4) can be expressed as,

$$\mu mg = m \frac{V^2}{R} \quad (6-5)$$

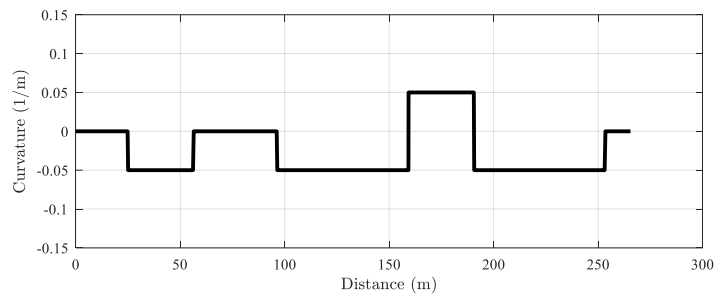
By solving both side of Equation (6-5), the curvature-based velocity profile can be expressed as,

$$V_{curv} = \sqrt{\mu g R} = \sqrt{\frac{\mu g}{|K|}} \quad (6-6)$$

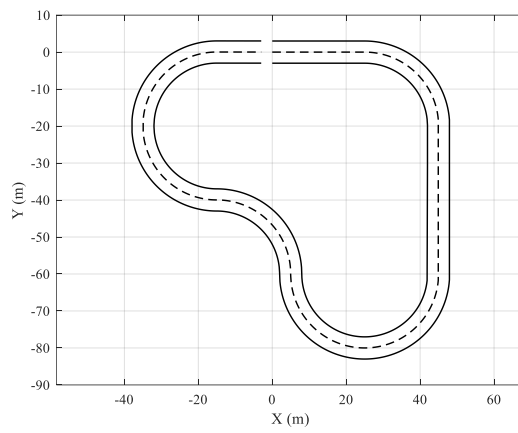
Where K is the path curvature which is the inverse of path radius, R . The assumption needed to apply this curvature-based velocity profile is that it is a flat road with no slipping condition. As an example in Figure 6-3 (a), there are two velocity profiles generated with acceleration and deceleration slopes of 8 m/s^2 and 2 m/s^2 that represent an aggressive manoeuvre, and the later is more on comfort velocity for a short multi-corners path. This velocity profile generation will be used to evaluate the developed PLAPC as in Sub-Chapter 8.4. In addition, longer path of Sepang International Circuit will use the same curvature-based velocity profile method too. Table 6-1 described all the velocity profiles applied in simulation testing of this work.



(a) Velocity profile



(b) Path curvature



(c) Multi-corners path

Figure 6-3 Example velocity profiles based on the path curvature

Table 6-1 Type of velocity profile used in this work

Profile name	Descriptions
A	Optimal solutions generated velocity profiles with optimised comfort objective.
B	Optimal solutions generated velocity profiles with minimum time objective.
C	Optimal solution (minimum time objective) with 90% factor generated velocity profile. This is to test the controller further with a reduced factor that helps the solver solve in real-time, as an optimal solution especially for minimum time could be impractical and harder to solve.
D	Curvature-based velocity profile with acceleration and deceleration slope of 2 m/s^2 that is aimed for comfort profile.
E	Curvature-based velocity profile with acceleration and deceleration slope of 5 m/s^2 that is aimed as a middle ground to assess all objective functions.
F	Curvature-based velocity profile with acceleration and deceleration slope of 8 m/s^2 that is aimed for time efficiency.

These velocity profiles were essential for testing the controller's performance under a variety of scenarios. By providing a range of velocity reference inputs, this study systematically evaluated the controller's ability to adapt to different objectives, from achieving time efficiency to maintaining ride comfort and ensuring computational feasibility. The results obtained from simulations demonstrated the impact of each profile on trajectory tracking, comfort level, energy consumption, and overall controller performance.

6.4 Actual Real-Time Implementation Considerations

In this work, the scope is up to high-fidelity simulation by using IPG CarMaker software but with consideration to preparing the model for the actual vehicle testing in real-time. The following are considered in the development of the controller and simulation setup,

- i. Local coordinates conversion where this method offers significant benefits, including simplified control calculations, enhanced numerical stability, and reduced computational load. This approach also aligns well with sensor data, as most sensors measure relative to the vehicle, making data integration smoother and more accurate. A local frame improves real-time performance, making control more efficient, especially on complex paths.
- ii. Yaw angle ranges wrapping within $[-\pi, \pi]$ radian or $[-360, 360]$ degrees is beneficial in real-time control because it ensures that the yaw angle remains continuous and consistent. Without wrapping, the yaw angle figure kept increasing or could abruptly jump when crossing this boundary, leading to instability in calculations and control actions, especially when determining the difference between two angles. By keeping yaw angles within a standardized range, wrapping helps maintain smooth and predictable control, making it easier for the controller to track heading deviations accurately and respond to changes in orientation effectively.
- iii. A time delay of 0.01 seconds is considered in this IPG CarMaker model for real-time implementation where there will be delayed in terms of data transmission, sensor processing, actuator dynamic, signal conversion and others that could potentially affect the responsiveness and stability of the controller and ultimately for safety of the vehicle. One consideration is torque command input with a time delay of 0.01 seconds, that is necessary due to data transmission delay.

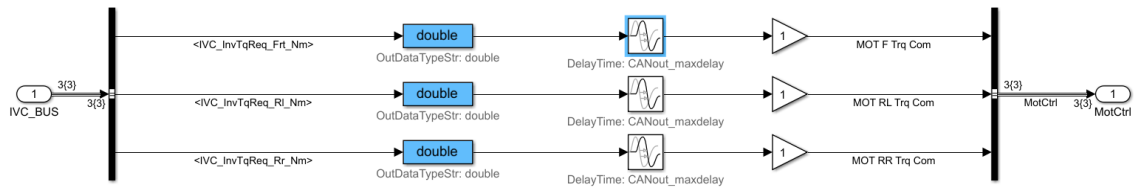


Figure 6-4 Time delay block for torque command block to the vehicle electric motors in the IPG CarMaker simulation model

- iv. Flag status and solved time are captured for each iteration that records and monitor the solved status and solving time that are essential for enhancing the reliability and efficiency of real-time control systems. Flag status provides feedback on system conditions, enabling quick error detection and necessary adjustments, strengthening system robustness. Monitoring solving time ensures that control actions are computed within real-time limits, maintaining a stable control loop frequency crucial for predictable performance. Consistent solving times indicate stable operation, while variations highlight areas for optimisation, ensuring the control system is both responsive and capable of handling real-time demands effectively.

6.5 Simulations Setup

The simulation setup in IPG CarMaker began with generating the path and velocity profiles. The path generation followed the method described in Sub-Chapter 4.3, while the velocity profile was derived either from optimal solutions based on Chapter 5 and curvature-based velocity profile as discussed in Sub-Chapter 6.3. For consistency in comparison, the initial conditions were set in the IPG CarMaker with a consistent initial velocity and the same lateral offset position. The controller was tested on three distinct path shapes: a simple 90-degree right-hand corner, a mixed corner extended path, and a longer path based on the actual circuit.

6.6 Summary

This chapter presented a high-fidelity simulation setup using IPG CarMaker integrated with MATLAB/Simulink to validate the proposed controller, PLAPC. Local coordinate conversion simplified control calculations to represent actual situations and improved numerical stability, while yaw angle wrapping ensured smooth and continuous orientation tracking. A 0.01-second time delay accounted for real-time implementation challenges such as communication, sensor and actuator delays. Monitoring flag status and solving times also ensured robust and reliable operation under real-time constraints. Overall, the setup provides an effective framework for testing the controller under near-realistic conditions, enabling a transition to real-world applications.

7 Simultaneous Planning and Tracking Controller Development

7.1 Introductions

This chapter highlights the development of a novel Perpendicular Line Ahead Planner Controller (PLAPC) via a simultaneous planning and tracking framework. This controller falls within the broader scope of motion planning and control as it not only involves controlling the vehicle to align with a novel reference system that generates a trajectory on the fly but also generates that trajectory while considering various constraints, objective functions and tracking a set of velocity profiles. As shown in Figure 1-2 and Figure 7-1, this approach provides a wide visual representation of the driving path while maintaining awareness of lateral boundaries. It serves as an alternative method for motion planning and control by generating a trajectory and performing control simultaneously while adapting to real-time requirements.

The controller is developed using a NMPC approach that employs a novel reference method of perpendicular lookahead reference line that is developed and simulated within the IPG CarMaker environment, using a 3 DOF double-track vehicle model that incorporates both longitudinal and lateral load transfers, along with a load-dependent Magic Formula tyre model.

7.2 NMPC Problem Formulations

The NMPC problem formulation for the development of PLAPC in this work adopts a novel method for defining the reference system. Instead of relying on discrete coordinate points, this approach utilizes a perpendicular lookahead reference line, as illustrated in Figure 1-2 and Figure 7-1. This line-based reference system provides a continuous representation of the defined path, enabling the PLAPC to adjust the vehicle's trajectory more dynamically. In addition, this controller is developed with reference to a velocity profile either based on optimal solutions or a curvature-based velocity profile.

For the real-time development of the NMPC formulation of the PLAPC controller in IPG CarMaker, the vehicle model was reduced from 7 DOF to a 3 DOF representation by eliminating wheel dynamics (Equation (4-8)). This reduction ensures computational efficiency for real-time implementation while maintaining essential vehicle dynamics.

The vehicle's position is described in local coordinates (X, Y), as explained in Chapter 6.2. The equations of motion governing the vehicle's longitudinal and lateral dynamics, as well as its yaw motion, are defined as follows,

$$\dot{X} = \frac{dX}{dt} = V_x \cos(\psi) - V_y \sin(\psi) \quad (7-1)$$

$$\dot{Y} = \frac{dY}{dt} = V_x \sin(\psi) + V_y \cos(\psi) \quad (7-2)$$

$$\dot{\psi} = \frac{d\psi}{dt} = r \quad (7-3)$$

Re-instated the equation of motion as in Equation (4-5) to (4-7), longitudinal and lateral dynamics equations has been described as,

$$\begin{aligned} \dot{V}_x = \frac{1}{m} (F_{flx} \cos(\delta_f) - F_{fly} \sin(\delta_f) + \\ F_{frx} \cos(\delta_f) - F_{fry} \sin(\delta_f) + F_{rlx} + F_{rrx}) + \psi V_y \end{aligned} \quad (7-4)$$

$$\begin{aligned} \dot{V}_y = \frac{1}{m} (F_{flx} \sin(\delta_f) + F_{fly} \cos(\delta_f) + \\ F_{frx} \sin(\delta_f) + F_{fry} \cos(\delta_f) + F_{rly} + F_{rry}) - \psi V_x \end{aligned} \quad (7-5)$$

While for the yaw moment of the vehicle,

$$\begin{aligned} \ddot{\psi} = & \frac{1}{I_z} (l_f (F_{flx} \sin(\delta_f) + F_{frx} \sin(\delta_f) + F_{fly} \cos(\delta_f) + F_{fry} \cos(\delta_f)) - \\ & l_r (F_{rly} + F_{rry}) - \\ & \frac{w}{2} (F_{flx} \cos(\delta_f) - F_{fly} \sin(\delta_f) + F_{rlx}) + \\ & \frac{w}{2} (F_{frx} \cos(\delta_f) - F_{fry} \sin(\delta_f) + F_{rrx})) \end{aligned} \quad (7-6)$$

To reduce computational load for real-time applications, the vehicle model is simplified from a 7 DOF to a 3 DOF representation by neglecting wheel rotational dynamics. Under this assumption, the tire adhesion limit is assumed to remain within bounds in the longitudinal direction, allowing the longitudinal tire force to be directly related to the applied driving or braking torque. Therefore, the longitudinal and lateral forces can be expressed as follows,

$$F_{flx} = F_{frx} = \frac{T_f}{2R_w} \quad (7-7)$$

$$F_{rlx} = \frac{T_{rl}}{R_w}, \quad F_{rrx} = \frac{T_{rr}}{R_w} \quad (7-8)$$

Therefore, the states vector for this NMPC formulation can be described as,

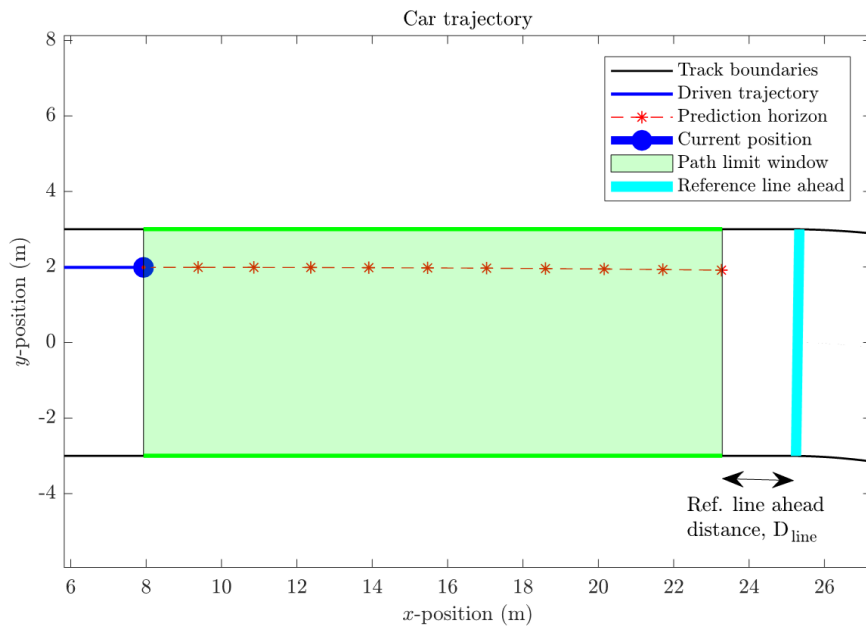
$$st^{on} = [X, Y, \psi, V_x, V_y, r]^T \quad (7-9)$$

And the control inputs for FWS vehicle topology are,

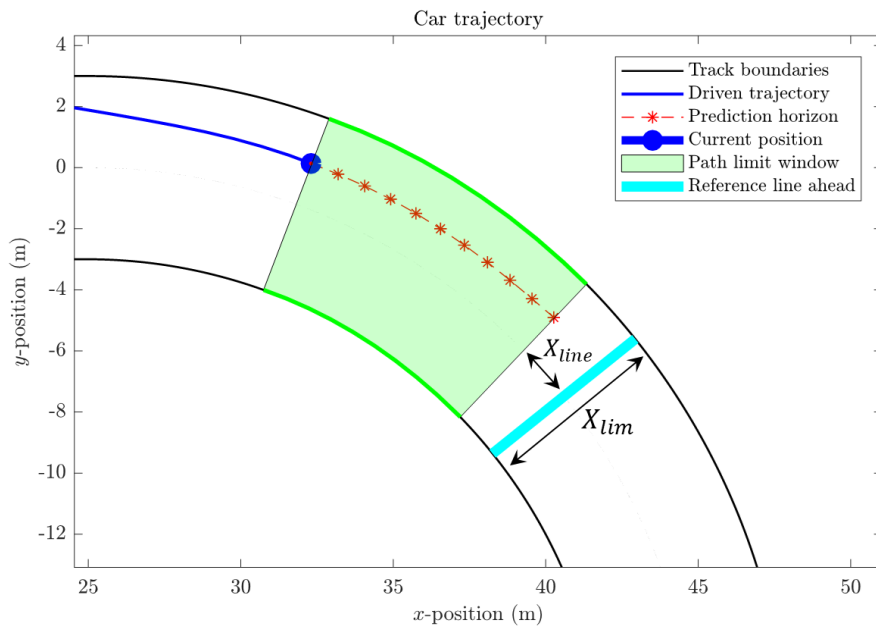
$$in^{on} = [T_f, T_{rl}, T_{rr}, \delta_f]^T \quad (7-10)$$

In addition, there are auxiliary variables of,

$$aux^{on} = [a_x, a_y, D_{fl}, D_{fr}, D_{rl}, D_{rr}]^T \quad (7-11)$$



(a) Visualisation of the implemented PLAPC in straight path



(a) Visualisation of the implemented PLAPC in the corner

Figure 7-1 Perpendicular lookahead reference line (cyan) ahead of the current vehicle states and prediction horizons

7.2.1 Objective Functions

This NMPC problem formulation for PLAPC can be formulated for each objective function by discretizing the nonlinear continuous-time system using the Runge-Kutta 4th order method as summarized in Table 7-1.

Table 7-1 Objective functions for Simultaneous Planning and Tracking

Type of Objective Functions	Descriptions
Minimised distance, J_d^{on}	Minimising distance from the vehicle's current state and each horizon points to the reference line ahead.
Optimised energy, J_e^{on}	A multi-objective function balancing distance to the reference line ahead minimisation and energy efficient by penalizing positive torque using LINEX terms.
Optimised comfort, J_c^{on}	A multi-objective function balancing distance to the reference line ahead minimisation and comfort by penalising acceleration and jerk (rate of acceleration), the ISO 2631-1 [22] is referred to define the comfort level.

For minimising the distance to the reference line ahead (time efficiency related),

$$J_d^{on} = \min_{x,u} \sum_{k=0}^{N-1} (X_{line}^2 Q_{line} + X_{lim} Q_{lim} + \Delta V_{xk}^2 Q_{vel}) \quad (7-12)$$

For optimising energy,

$$J_e^{on} = \min_{x,u} \sum_{k=0}^{N-1} (X_{line}^2 Q_{line} + X_{lim} Q_{lim} + \Delta V_{xk}^2 Q_{vel} + U_{linex}) \quad (7-13)$$

For optimising comfort,

$$J_c^{on} = \min_{x,u} \sum_{k=0}^{N-1} (X_{line}^2 Q_{line} + X_{lim} Q_{lim} + \Delta V_{xk}^2 Q_{vel} + Y_{comfort}) \quad (7-14)$$

And regularisation term as described in Sub-Chapter 5.2.5,

$$J_{reg}^{on} = \min_u \sum_{k=0}^{N-1} (U_k^T Q_u U_k + (U_k - U_{k-1})^T Q_{du} (U_k - U_{k-1})) \quad (7-15)$$

Therefore, the total objective function can be expressed as,

$$J^{on} = J_p^{on} + J_{reg}^{on}, \quad p = d, e, c \quad (7-16)$$

subject to,

$$x_0 = x_{initial}, u_0 = u_{initial}$$

$$x_{k+1} = f_d(x_k, U_k)$$

$$x_k^{min} \leq x_k \leq x_k^{max}$$

$$u_k^{min} \leq u_k \leq u_k^{max}$$

$$y_k^{min} \leq y_k \leq y_k^{max}, k = 0, \dots, N-1$$

The objective functions as seen in Equation (7-12) to (7-14) seek to minimise the distance from each prediction horizon point to the perpendicular lookahead reference line with Equation (7-13) considered energy efficiency and (7-14) seek to optimise for comfort. Equation (7-16) gave the total objective function that includes regularisation term as mentioned in Sub-Chapter 5.2.5. The objective subject to all initial conditions defined for states $x_{initial}$ and input $u_{initial}$ and also specified limit of each states (x_k^{min}, x_k^{max}) , inputs (u_k^{min}, u_k^{max}) and auxiliary variables (y_k^{min}, y_k^{max}) . f_d represent as the discrete-time system of the controller model, which is derived from the RK4 method on a continuous time system.

The rest of the terms include,

- i. Distance to reference line ahead penalty, X_{line} : Defined by the distance to a reference line ahead of the prediction horizon points,

$$X_{line} = \frac{|a_l X_{local,k} + b_l Y_{local,k} + c_l|}{\sqrt{a_l^2 + b_l^2}} \quad (7-17)$$

Where, $X_{local,k}^x$ and $Y_{local,k}^y$ are the vehicle X and Y local coordinates along the prediction horizon and coefficient a_l , b_l and c_l define the reference line ahead at a distance D_{line} , which were derived from the path's orientation based on the local yaw angle, ϕ_l^{local} . The polynomial functions of the line reference can be expressed in the form of as,

$$a_l x + b_l y + c_l = 0 \quad (7-18)$$

With the coefficient a_l , b_l and c_l as,

$$a_l = \cos(\phi_l^{local}) \quad (7-19)$$

$$b_l = \sin(\phi_l^{local})$$

$$c_l = -(a_l X_l^{local} + b_l Y_l^{local})$$

This moving line serves as a “target” or reference for the PLAPC, where the velocity of this line is set to be the same as the vehicle's current state. By incorporating X_{line} into the objective function, weighted by a factor Q_{line} , the NMPC penalizes deviations from the vehicle's current state, and each prediction horizon points to the reference line ahead. The closest distance is a perpendicular point, which keeps the vehicle closer to the reference line and effectively generates its trajectory on the fly.

- ii. Reference line width limit penalty, X_{lim} : Another important aspect is to ensure the vehicle stay within the path limit while keep referencing to the line ahead that would be another safety aspect for vehicle motion and control. This can be expressed as,

$$X_{lim} = \begin{cases} P_L^2 Q_{road}, & \text{if out of left edge} \\ P_R^2 Q_{road}, & \text{if out of right edge} \end{cases} \quad (7-20)$$

$$P_L = Y_{local} + P_{hw}$$

$$P_R = P_{hw} - Y_{local}$$

Where, Q_{road} is the weight factor for penalization for the reference always within the path limit, P_L and P_R is to calculate the distances to the path edge and P_{hw} is path half width. This became a soft constraint to the controller in order to ensure the vehicle stays within the path boundary and at the same time, has continuous reference in case it is out of the boundary.

- iii. Longitudinal velocity deviation penalty, ΔV_{xk} : For the vehicle to move according to the desired velocity, a reference longitudinal velocity is generated based on either an optimal solution result, V_{opt} or the curvature-based, V_{curv} velocity profile of the pre-defined path, which can be expressed as V_{xk}^{ref} and velocity tracking weight factor of Q_{vel} . This can be formulated as,

$$\Delta V_{xk} = \left(V_{xk} - V_{xk}^{ref^j} \right), \quad j = V_{opt}, V_{curv} \quad (7-21)$$

- iv. Torque control input penalty, U_{linex} : Another input penalty is on torque, where instead of conventional quadratic penalization, which equally penalizes the positive and negative sides of the torque. The term Linear-Exponential (LINEX) term is introduced in this penalty term, which heavily penalises the positive torque, allowing more negative torque for regenerative braking essential for energy efficiency. This asymmetric approach can be expressed in the general form of,

$$L(x) = c e^{ax} - bx + a_0 \quad (7-22)$$

In which this defined the torque input penalty as,

$$U_{linex} = \sum_{k=1}^N \left(c e^{a U(T,k)} - b U(T,k) + a_0 \right) \quad (7-23)$$

Where T represent torque of front, T_f , rear left, T_{rl} and rear right, T_{rr} and a, a_0, b and c are LINEX parameters tuned to encourage efficient energy usage. As shown in Figure 7-2 and equation (7-22) describes the general form of LINEX term where, by using this penalty function method, it encourages an energy-efficient by penalizing positive torque more than negative torque. This asymmetric approach reduces the energy consumed for propulsion while allowing for more negative torque when deceleration or

braking is necessary, maximising energy recovery through regenerative braking. In contrast, the quadratic penalty, being symmetric, does not inherently differentiate between positive and negative torque, leading to less optimal energy usage.

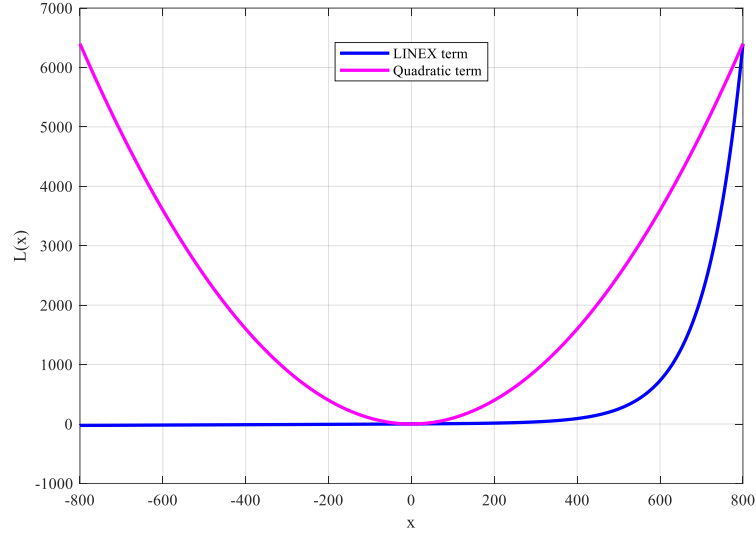


Figure 7-2 Comparison between LINEX and quadratic term in penalization of torque control inputs

- v. Comfort penalty, $Y_{comfort}$: The comfort aspect in this NMPC formulation was described based on the penalization of longitudinal, a_x and lateral, a_y accelerations and derivative of it, jerk. This term could be defined as,

$$Y_{comfort} = Q_{cax}(a_x^2) + Q_{cay}(a_y^2) + Q_{cdax}(\dot{a}_x^2) + Q_{cday}(\dot{a}_y^2) \quad (7-24)$$

Where the Q_{cax} , Q_{cay} , Q_{cdax} and Q_{cday} are the weight factor for each accelerations and derivatives term.

7.2.2 Constraints

Regarding constraints for the NMPC problem formulation, the same conditions mentioned in the optimal solution problem as in Equation (5-12) until (5-17) are applied. However, for real-time implementation in the IPG CarMaker environment, where time is the independent variable, additional constraints are introduced to ensure the vehicle remains within the path boundaries for safety.

To generate a path boundary for the vehicle, a second-order polynomial function is fitted using three equally spaced points along the prediction horizon: the initial point, (x_i, y_i) , the middle point, (x_m, y_m) , and the endpoint of the horizon (x_f, y_f) . All these points were fed in the form of local coordinates which made (x_i, y_i) is $(0,0)$. The general form of the polynomial fitting is given by,

$$P(x) = p_1x^n + p_2x^{n-1} + \dots + p_nx + p_{n+1} \quad (7-25)$$

Where p_n is the polynomial coefficient determined by fitting the given data points for n order of polynomials.

For this work, a second-order polynomial ($n = 2$) is used. Given the longitudinal positions $X_{local}^p = \{x_i, x_m, x_f\}$ and the corresponding lateral positions $Y_{local}^p = \{y_i, y_m, y_f\}$, the polynomial function takes the form as,

$$Y^p(x) = p_1x^2 + p_2x + p_3 \quad (7-26)$$

The longitudinal positions X_{local}^p are then evenly spaced along the length of the prediction horizon. By applying the polynomial function in Equation (7-26), the corresponding lateral positions, Y_k^p are obtained at each time step k . The path boundary constraints can then be defined as,

$$W_p^l \leq Y_k - Y_k^p \leq W_p^r \quad (7-27)$$

Where W_p^l and W_p^r represent the left and right path boundaries, respectively.

By evaluating the polynomial at each time step, k and calculating the deviation from the vehicle's state lateral position Y_k to the polynomial points Y_k^p , these path constraints ensure that the vehicle remains within the defined boundaries. This method enforces the safe aspect of the controller, as illustrated example in Figure 7-3 below.

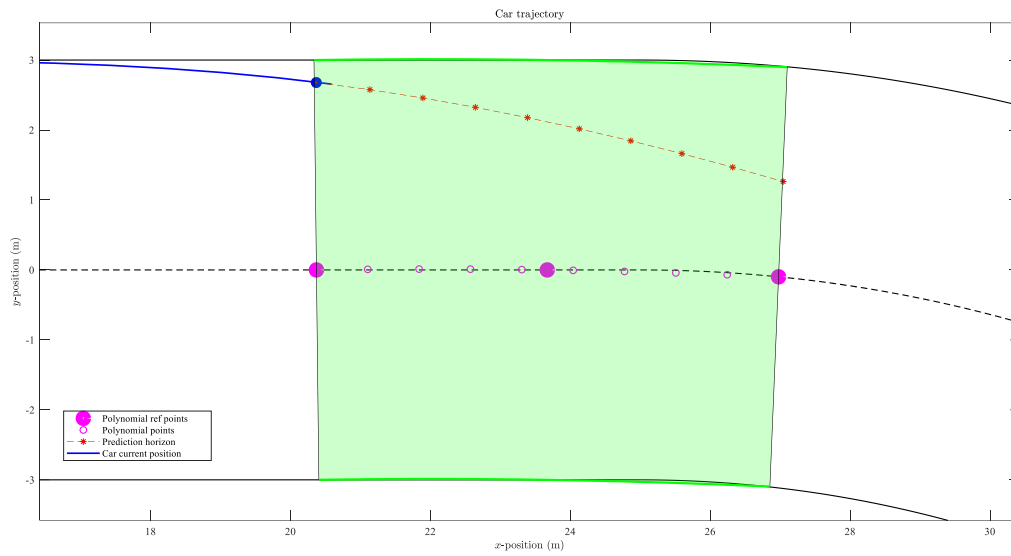


Figure 7-3 Path constraint based on 2nd-order polynomial fitting

7.3 Sensitivity Study

It is essential to perform a sensitivity study to understand the behaviour of the PLAPC, determine the optimal parameters for its development, and ensure real-time adaptability. This sensitivity study were conducted online in IPG CarMaker software. This subchapter studied various aspects, including effect on varying integration nodes, prediction horizon length, sampling time, line-ahead reference distance, velocity profile during cornering and LINEX parameters. In this study, infeasibility problems, overrun problems, solved beyond the sampling time and reaching maximum number of iterations from the solver is defined as “solver errors”.

7.3.1 Varying Integration Node

A sensitivity study was conducted by varying the integration nodes of the RK4 to understand the controller's sensitivity to changes in the variation of the integration interval. A multi-corner path was chosen to evaluate this, where it is set with a fixed sampling time of 0.05 seconds, a prediction horizon length of 10 steps and 2 meters fixed distance of reference line ahead. The RK4 nodes were incrementally varied from 1 to 5 to evaluate the accuracy and impact on solution time, a critical factor for real-time implementation.

Table 7-2 Comparison result of varying integration node

Node	1	2	3	4	5
Average solved time (s)	0.0288	0.0305	0.0310	0.0316	0.0323
Peak solved time (s)	0.0494	0.0499	0.0496	0.0494	0.0487
No. of solver errors	0	0	0	0	0
RMS error of velocity tracking (m/s)	0.3003	0.2988	0.2975	0.2969	0.2966

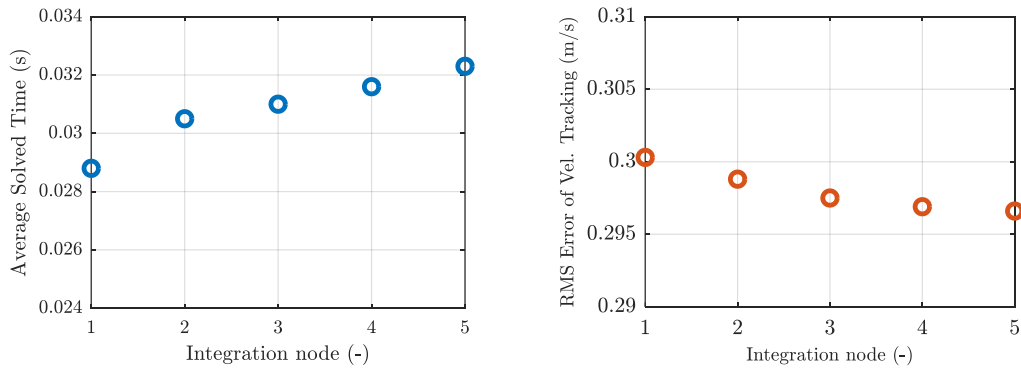


Figure 7-4 Sensitivity study on average solved time and RMS error of velocity tracking by varying the integration node

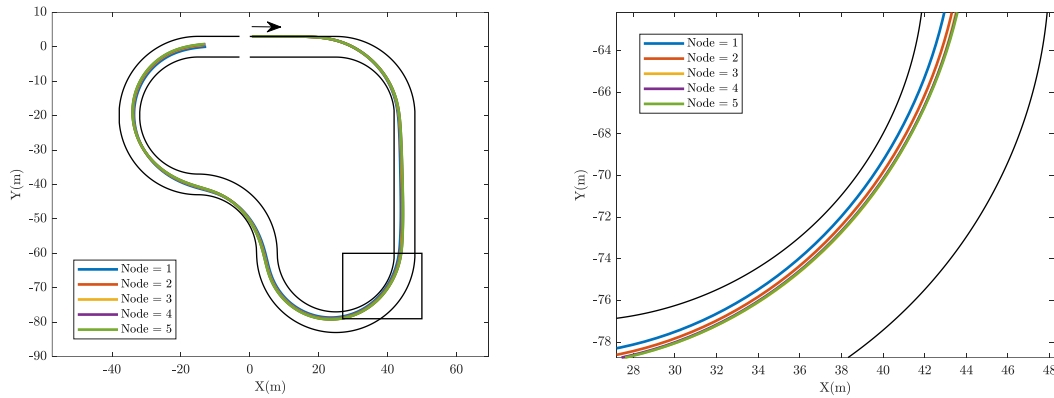


Figure 7-5 Vehicle trajectory comparison by varying the integration node

Table 7-2 shows that increasing the number of integration nodes from 1 to 5 improves accuracy in terms of velocity tracking but comes at the cost of higher computational load. Interestingly, the peak solved time does not increase with more nodes; in fact, Node 5 has the lowest peak solved time. This could suggest that a finer integration interval helps to manage occasional computational spikes more effectively. Regarding accuracy, the RMS error between the reference longitudinal velocity and the vehicle's longitudinal velocity decreases as the number of nodes increases, indicating that higher node counts result in better accuracy, albeit with a higher computational cost. Additionally, the trajectories for

different nodes become closer as the node count increases, especially between Nodes 3 and 5, showing a more refined solution with finer integration. For real-time control, it is crucial to balance accuracy and computational effort. Therefore, Node 3 is best for its efficiency, accuracy and computational load balance.

7.3.2 Varying Prediction Horizon Length

Based on the RK4 integration step of 3, this study examines variations of prediction horizon durations to assess their impact on the controller's performance. The controller's sampling time is fixed at 0.05 seconds and 2 meters fixed distance of reference line ahead, while the prediction horizon is varied between 0.3 seconds and 0.7 seconds, corresponding to a range of 6 to 14 nodes within the prediction horizon.

Table 7-3 Comparison result of varying prediction horizon length

Horizon length (s)	0.3	0.4	0.5	0.6	0.7
No. of horizon nodes	6	8	10	12	14
Average solved time (s)	0.0281	0.0292	0.0305	0.0384	0.0491
Peak solved time (s)	0.411	0.410	0.465	0.495	0.624
No. of solver errors	0	0	0	4	7
RMS error of velocity tracking (m/s)	0.2607	0.2782	0.2977	0.3201	0.3647

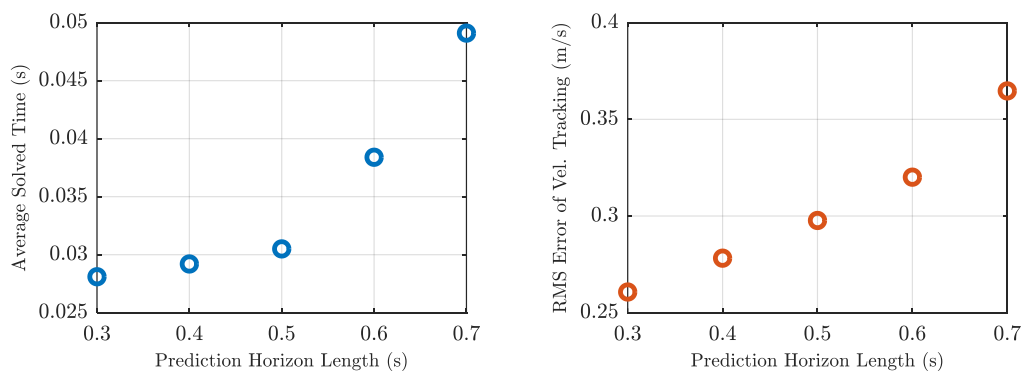


Figure 7-6 Sensitivity study on average solved time and RMS error of velocity tracking by varying the prediction horizon length

The results presented in Table 7-3 and Figure 7-6 demonstrate the impact of varying the prediction horizon length on control performance, as evaluated through the average solved time and the RMS error of velocity tracking. As the prediction horizon length increases, there is a noticeable rise in the average time required to solve the control problem. This increase in computational demand is expected, as a longer horizon requires the controller to process a broader future window, which adds complexity. Specifically, the average solved time rises from 0.0281 seconds at a 0.3-second horizon to 0.0491 seconds at a 0.7-second horizon, with a significant jump observed between the 0.5 and 0.7-second horizons. This trend suggests that extending the prediction horizon may approach the boundaries of real-time computational feasibility.

Additionally, the RMS error of velocity tracking also tends to increase with longer prediction horizons, indicating a trade-off between computational efficiency and control accuracy. This degradation in tracking performance suggests that, although longer horizons enable the controller to consider a more extensive future trajectory, it also introduces challenges in accurately matching the velocity reference.

The observed trends highlight the need to balance solved time and tracking accuracy when selecting a prediction horizon. Based on the data, shorter horizons (such as 0.3 to 0.4 seconds) appear optimal, offering better velocity tracking accuracy and faster computation times. However, with a limited view of the future, the controller may struggle to anticipate upcoming changes in the reference trajectory. While longer horizons may theoretically improve the controller's foresight, they lead to increased computational demands and higher tracking errors, which may not be practical for real-time implementations. Therefore, a moderate prediction horizon length of 0.5 seconds or 10 points of prediction horizons appears to be the most effective choice for balancing real-time computational requirements with control performance in this work.

7.3.3 Varying Sampling Time

A further sensitivity study was conducted to test the sensitivity of the developed controller to changes in sampling time, and the sampling time varied from 0.025 seconds to 0.1 seconds. The prediction horizon was fixed at 10 nodes as set in the previous section, 2 meters fixed distance of reference line ahead, and the internal integration steps remained consistent with the settings determined in the prior sensitivity study.

Table 7-4 Comparison results of varying sampling time

Sampling time (s)	0.025	0.05	0.075	0.1
Average solved time (s)	0.0400	0.0327	0.0345	0.0717
Δ Time (s) / margin-left (%)	-0.0150 (-)	0.0173 (34.6%)	0.0405 (54%)	0.0283 (28.3%)
Peak solved time (s)	0.0452	0.0469	0.0610	0.0923
No. of solver errors	12	0	0	5
RMS error of velocity tracking (m/s)	0.2110	0.2797	0.3818	0.6341

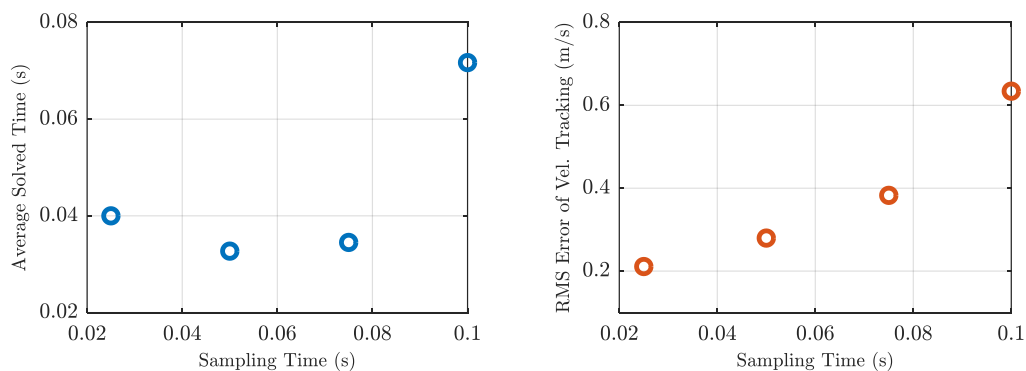


Figure 7-7 Sensitivity study on average solved time and RMS error of velocity tracking by varying the sampling time

The results, summarized in Table 7-4, indicate that increasing the sampling time led to higher RMS errors in velocity tracking, reflecting a decline in tracking accuracy as sampling time increased. The average solved times varied across sampling times, as seen in Figure 7-7, with the lowest at 0.05 seconds (0.0327 seconds average solved time) and the highest at 0.1 seconds (0.0717 seconds average solved time). Shorter sampling intervals such as 0.025 seconds allowed for more frequent control updates, enhancing accuracy, but due to the model's complexity, it required more computational effort beyond the sampling time, as reflected in the average solve time of 0.0400 seconds. Notably, when the sampling time reached 0.1 seconds, there was a significant rise in RMS error for velocity tracking (0.6341 m/s), highlighting a noticeable degradation in tracking performance due to less frequent control updates. Interestingly, the results indicate that a sampling time of 0.075 seconds effectively balances control performance and computational efficiency. With an average solve time of 0.0345 seconds, this configuration provides slightly more than 50% buffer within each 0.075-second interval, which is favourable for real-time implementation as it gives some margins to handle additional computational demands or unforeseen delays.

7.3.4 Varying Reference Line Distances Ahead

In this section, the sensitivity of the developed controller to variations in the reference line distance ahead, D_{line} is examined (Refer to Figure 7-1 for visualisation position of D_{line}), focusing on its impact on vehicle trajectory and solving time. The distance of the reference line ahead of the vehicle was varied from 1 to 5 meters to observe how these adjustments influence control performance and generated trajectory. For this test, the velocity reference was set the same across all test sets and the controller parameters were fixed at a sampling time of 0.075 seconds and a prediction horizon of 10 nodes, as set in the previous section. The internal integration steps were also held constant, consistent with the settings defined in the prior sensitivity study. Varying only the reference line distance allows for assessing its isolated effects on controller efficiency, trajectory generation, and real-time computational demands, offering insights into an optimal balance between predictive accuracy and computational feasibility.

Table 7-5 Comparison result of varying reference line distance ahead

Line distance ahead of the last node of prediction horizon (m)	1	2	3	4
Average solved time (s)	0.0333	0.0385	0.0391	0.0424
Peak solved time (s)	0.0625	0.0697	0.0953	0.0977
No. of solver errors	0	0	4	8
RMS error of vel. tracking (m/s)	0.3706	0.3818	0.4070	0.4183

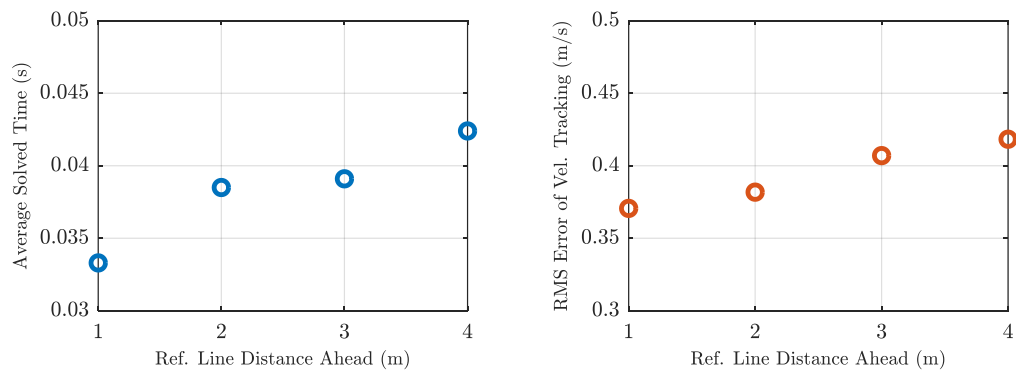
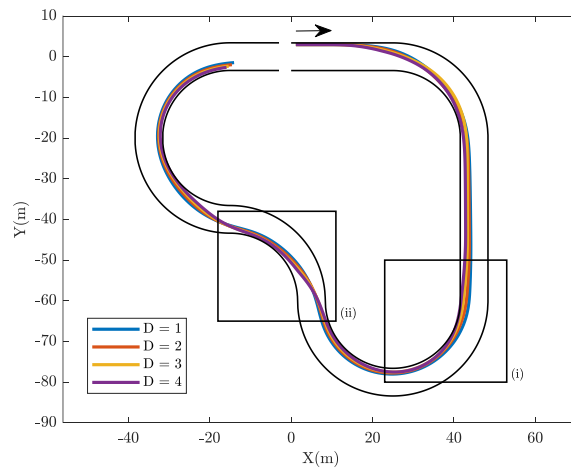
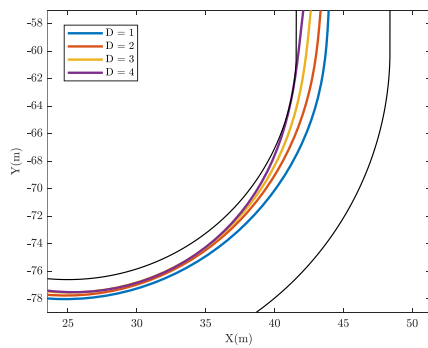


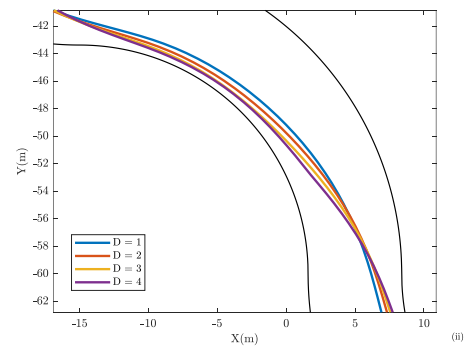
Figure 7-8 Sensitivity study on average solved time and RMS error of velocity tracking by varying reference line distance



(a) Trajectories



(b) Zoom-in at box (i)



(b) Zoom-in at box (ii)

Figure 7-9 Vehicle trajectory comparison by varying reference line distance

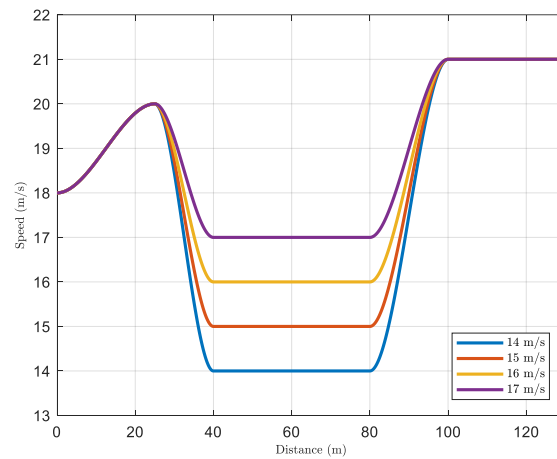
The sensitivity study on varying the reference line distance ahead, D_{line} reveals significant differences in controller performance, as shown in Table 7-5 and Figure 7-9. The results indicate that as the line distance ahead increases, the average solved time and the RMS error of velocity tracking tend to increase. This outcome suggests that larger reference distances make the control task more computationally demanding and slightly less accurate in tracking the target velocity reference.

Regarding trajectory accuracy, the visualization in Figure 7-9 shows that the reference distance ahead (D) influences the vehicle's trajectory. At reference distance of 4 meters ($D = 4$), the trajectory became closer to the inner path boundary but remains within the limits. This highlights that while a larger D allows for better anticipation of the upcoming corner, the controller behaviour also leads to an earlier turn-in, bringing the trajectory closer to the inner path and potentially nearing to the path limit.

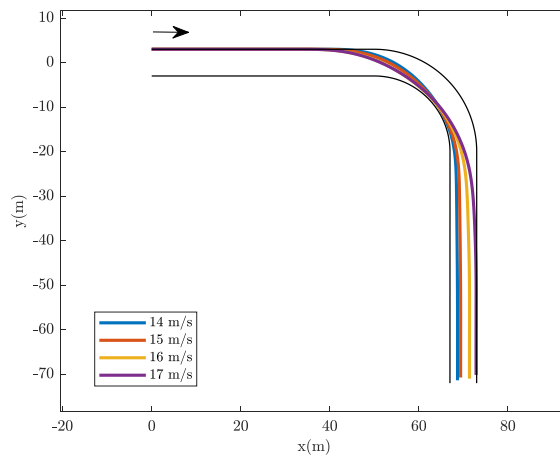
For this case, a 2-meter reference line distance effectively balances computational efficiency and control accuracy. Although a 1-meter distance yields the lowest velocity RMS error and solved time, a 2-meter distance only slightly increases these metrics while offering further view ahead. As the reference line distance ahead increases, the vehicle can still follow the perpendicular reference line, but it tends to go further inside the path, particularly in the curved sections of the track. Therefore, it is essential to note the effect of reference line ahead distance in optimising PLAPC control performance without significantly impacting computational load, while maintaining sufficient accuracy and staying within the path limit.

7.3.5 Varying Velocity Profile During Cornering

To further analyse the effect of applying a reference line ahead on vehicle trajectory, it is essential to study the impact of varying reference velocity, as this significantly influences the vehicle's path. For this analysis, the reference line was set at a fixed distance of 2 meters ahead of the vehicle. The controller parameters were set with a sampling time of 0.075 seconds, a prediction horizon of 10 nodes, and an integration step of 3, consistent with the settings defined in the previous sensitivity study.



(a)



(b)

Figure 7-10 Vehicle trajectory comparison by varying the in-corner velocity

Table 7-6 Comparison result of varying velocity profile

In-corner velocity (m/s)	14	15	16	17
Average solved time (s)	0.0316	0.0321	0.0356	0.0364
Peak solved time (s)	0.0611	0.0532	0.0690	0.0959
No. of solver errors	0	0	0	6

The analysis of the varying velocity profiles and their effect on vehicle trajectory and computation time reveals important trade-offs between cornering velocity, generated trajectory, and computational efficiency. The trajectory plot in Figure 7-10 (b) shows that the vehicle's trajectory took a wider exit path after the corner as the in-corner velocity increased. This is visually evident as the higher velocity (at 17 m/s) leads to a trajectory that approaches the outer boundaries of the track.

In terms of computational demand, as shown in Table 7-6, the average solved time increases as the in-corner velocity increase. This trend reflects that higher velocity requires more control adjustments, thereby demanding more from the solver. The increase in solved time at higher velocity could impact real-time implementation, especially in scenarios with frequent velocity changes. Additionally, the peak solved time also increases with higher in-corner velocity. The number of instances where the solver exceeded the set sampling time further highlights the challenge of higher velocity. At 15 m/s, there were no solver errors of exceeding the sampling time, whereas at 17 m/s, there were 6 instances of solver errors where the solver took longer than the specified sampling time. It suggests that while higher in-corner velocity improve lap times, it will compromise the control system's ability to meet real-time requirements and might deviate from the path.

The overall trade-off indicates that while higher in-corner velocity improves lap times by allowing a more aggressive trajectory, it comes at the cost of increased computational time. Therefore, this effect in generating velocity profile for real-time implementation is worth noting.

7.3.6 Varying LINEX Parameters

The LINEX method for torque input penalization is one of the key contributions of this work, as mentioned in Sub-Chapter 1.3. It asymmetrically penalizes the torque control input by applying a heavier penalty to positive torque while imposing a smaller penalty on negative torque. This method is simple to implement, and its asymmetry directly promotes energy efficiency by encouraging greater energy recovery through regenerative braking. Before investigating the effect of varying LINEX parameters, it is crucial to compare the LINEX method with a more conventional approach, namely the Rule-Based method. The Rule-Based method employs a quadratic function with if-else logic to penalize positive torque to minimise energy consumption. Both penalties can be expressed as,

LINEX term,

$$U_{linex} = \sum_{k=1}^N (c e^{a U(T,k)} - b U(T,k) + a_0) \quad (7-28)$$

Rule-based term,

$$U_{rules} = \sum_{k=1}^N (Q_p \text{ if } (U(T,k) > 0) U(T,k)^2) \quad (7-29)$$

$$- \sum_{k=1}^N (Q_n \text{ if } (U(T,k) < 0) U(T,k)^2)$$

Where $U(T,k)$ is torque control input at time k for each torque component T of the front, Tf , rear left, Trl and rear right wheel torque, Trr . Then Q_p and Q_n are weighting factors for positive and negative torque. Moreover, the *if* term is conditions that ensure only positive torque is penalized. For a fair comparison,

both methods were implemented with equal penalty magnitudes, as shown in Figure 7-11 and evaluated under the same velocity profile.

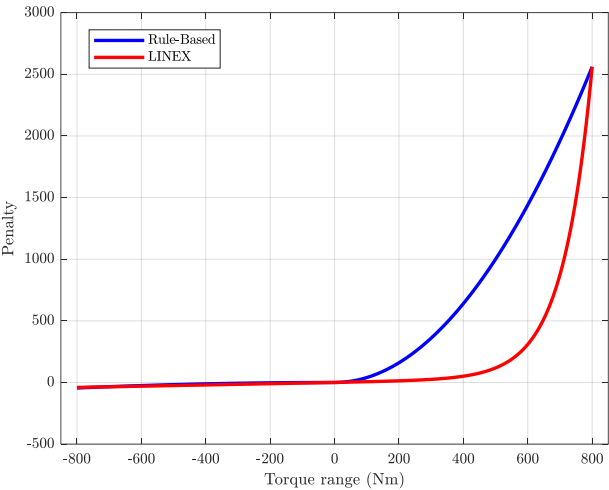


Figure 7-11 Penalty comparison of quadratic rule-based and LINEX functions

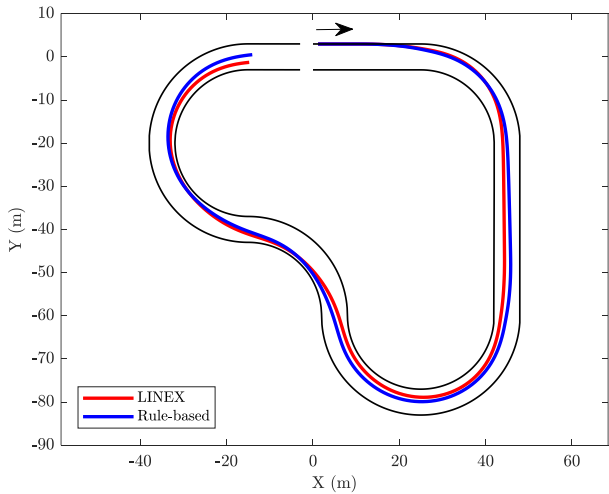
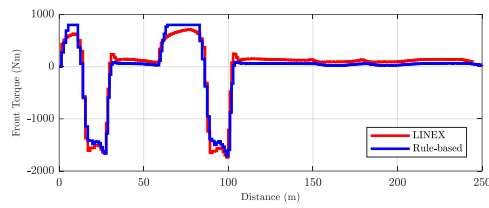
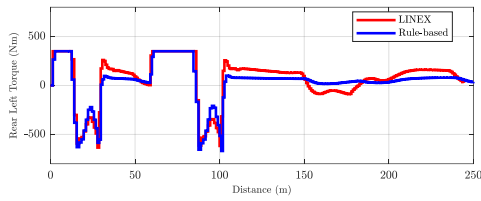


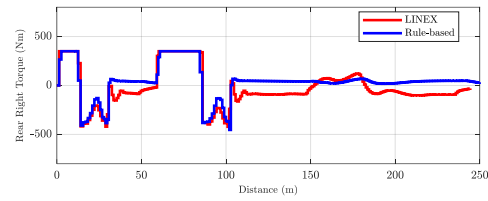
Figure 7-12 Vehicle trajectory comparison by varying the in-corner velocity



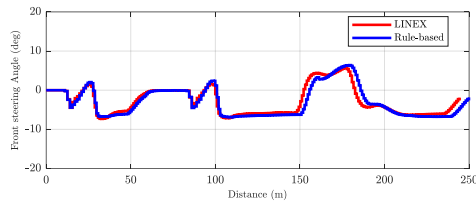
(a) Front wheel torque



(b) Rear left wheel torque



(c) Rear right wheel torque



(d) Front Steering angle

Figure 7-13 Control inputs comparison between LINEX and Rule-based for energy efficient

Table 7-7 Result comparison of LINEX and Rule-based for energy efficient

Key Metrics	LINEX	Rule-based	Differences
Total energy consumption (kWh)	0.0717	0.0690	+ 3.9%
Regenerative energy generated (kWh)	1.1444	0.9703	+ 17.9 %
Net energy usage (kWh)	-1.0727	-0.9013	+ 19.0 %
Lap time (s)	17.925	18.45	- 2.8 % (- 0.525 s)
Average solved time (s)	0.0333	0.0347	- 4.0%
Peak solved time (s)	0.0623	0.0636	-
No. of solver errors	0	0	-

The performance of the LINEX-based and Rule-Based objective functions was evaluated across several key metrics, as shown in Table 7-7, including computational efficiency, energy terms, regenerative braking energy recovery, and lap time. The results indicate that the LINEX method outperforms the Rule-Based approach in multiple aspects. LINEX achieved a faster average solve time of 0.0333 seconds compared to 0.0347 seconds for the Rule-Based method, demonstrating slightly better computational efficiency, which is advantageous for real-time implementation.

Regarding lap time, the LINEX method completed the track in 17.925 seconds, significantly faster than the 18.45 seconds achieved by the Rule-Based approach. This faster lap time highlights the LINEX method's ability to optimise torque application more effectively, which can be seen in torque profile comparison in Figure 7-13, enabling improved velocity performance while maintaining energy efficiency.

Energy analysis revealed that while the Rule-Based method recorded a slightly lower total energy consumption of 0.06898 kWh (compared to 0.07169 kWh for LINEX), it fell short in energy recovery. The LINEX approach recovered 1.1444 kWh of regenerative braking energy, representing a 17.9% improvement over the 0.9703 kWh recovered by the Rule-Based method. This superior energy recovery allowed LINEX to achieve a net energy usage of -1.0727 kWh, demonstrating a 19% improvement over the -0.9013 kWh net energy usage achieved by the Rule-Based approach.

In conclusion, the LINEX method provides superior energy efficiency by maximising regenerative braking recovery while maintaining faster lap times and better computational efficiency. Although the Rule-Based method is slightly better in minimising raw energy consumption, it sacrifices performance and regenerative energy recovery. Therefore, for applications where energy efficiency, performance, and real-time feasibility are critical, the LINEX method is the preferred choice.

To further understand the effect of LINEX parameters, a sensitivity study was conducted to see the effect of varying scale factor, c of LINEX parameter as Equation (7-28).

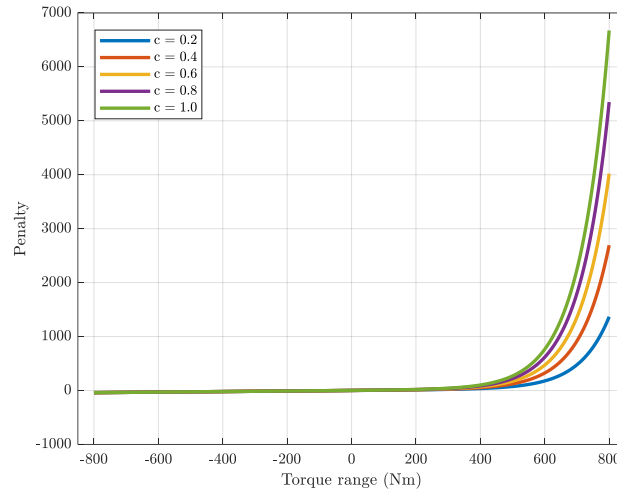
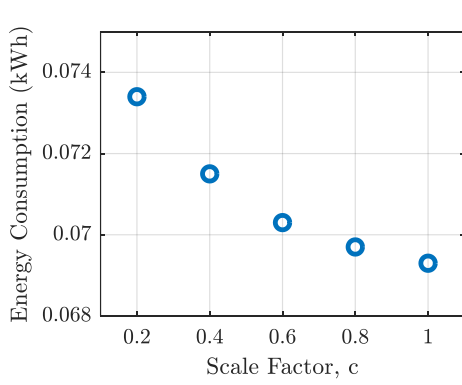


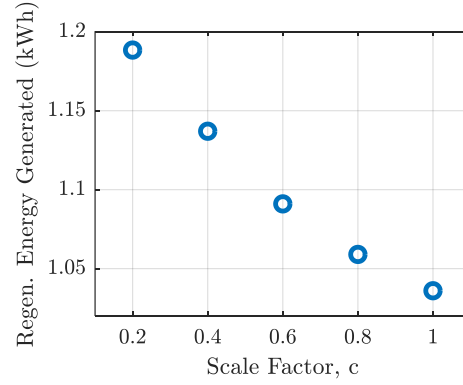
Figure 7-14 Varying scale factor of LINEX parameters

Table 7-8 Result comparison of varying scale factor of LINEX term for energy efficient

LINEX scale factor, c	0.2	0.4	0.6	0.8	1
Total energy consumption (kWh)	0.0734	0.0715	0.0703	0.0697	0.693
Regenerative energy generated (kWh)	1.1884	1.1370	1.091	1.059	1.036
Net energy usage (kWh)	-1.1150	-1.0655	-1.0212	-0.9899	-0.9671
Lap time (s)	17.925	17.925	18.00	18.00	18.075
Average solved time (s)	0.0330	0.0348	0.0344	0.0335	0.0356
Peak solved time (s)	0.0613	0.0647	0.0642	0.0632	0.0670
No. of solver errors	0	0	0	0	0

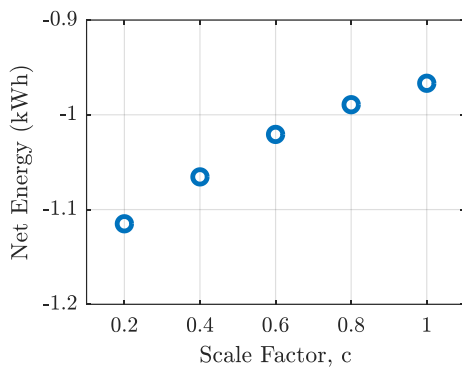


(a)

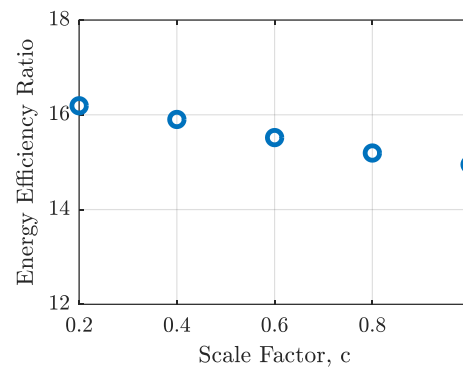


(b)

Figure 7-15 Total energy consumptions compared to regenerative energy generated with varying scale factor of LINEX term



(a)



(b)

Figure 7-16 Net energy usage and energy efficiency ratio with varying scale factor of LINEX term

The results in Table 7-8, Figure 7-15 and Figure 7-16 show that increasing the scaling factor c in the LINEX term led to a clear trade-off between energy consumption, regenerative energy recovery, and performance. As c increases, total energy consumption decreases, indicating that higher penalties on positive torque promote more conservative energy usage. However, this comes at the cost of reduced regenerative energy recovery and increased lap times. The

Energy Efficiency Ratio also declines with increasing C , highlighting the diminishing effectiveness of energy recovery as the torque control input becomes more conservative. Solver performance remains consistent, with no solver errors observed across all c values.

In conclusion, smaller c values favour higher energy recovery and faster lap times, while larger c values prioritize energy savings at the expense of energy recovery and lap time performance. Selecting the optimal scaling factor, c depends on balancing these trade-offs based on application priorities, which is worth noting throughout this work progress.

7.4 Robustness Study

After analyzing the behavior of the PLAPC through the sensitivity study in the previous chapter, it is essential to evaluate its robustness against real-world variations that autonomous vehicles commonly encounter. This robustness study focuses on two key external factors which are variations in vehicle mass and electric motor time delay.

The study was conducted in real-time using IPG CarMaker, ensuring an accurate assessment of the PLAPC's performance under these varying external conditions. The modifications were applied only to the vehicle plant model in IPG CarMaker, while the internal vehicle model parameters within the PLAPC remained unchanged.

To evaluate the robustness of the PLAPC, a multi-corner path with an 8 m/s^2 curvature-based velocity profile with minimum distance to the reference line ahead as an objective functions. The controller parameters were set with a sampling time of 0.075 seconds, a prediction horizon of 10 nodes, an integration step of 3, and 2 meters fixed distance of reference line ahead consistent with the settings defined in the previous sensitivity study.

7.4.1 Varying Total Vehicle Mass

A robustness study was conducted by varying the total vehicle mass in the vehicle model within IPG CarMaker. The additional mass accounts for two passengers, considering that this is a small, lightweight car, along with extra luggage. Based on the highest average body mass of 80 kg in the North American region [114], the two passengers contribute 160 kg. Additionally, an extra 100 kg was considered for luggage. To further analyze the impact of increased mass on vehicle performance, an additional 100 kg was added, bringing the total mass variation to up to 360 kg. The total vehicle mass was incrementally varied to assess its impact on both the trajectory tracking performance and solution time.

Table 7-9 Comparison result of varying total vehicle mass

Total vehicle mass (kg)	874.5	954.5	1034.5	1134.5	1234.5
Mass added (kg)	0 kg	80 kg	160 kg	260 kg	360 kg
Average solved time (s)	0.0349	0.0379	0.0386	0.0392	0.0401
Peak solved time (s)	0.066	0.0726	0.0664	0.0664	0.0841
No. of solvers error	0	0	0	0	4
RMS error of velocity tracking (m/s)	0.9921	1.0847	1.1755	1.2686	1.3577

Note: Blue cells indicate the baseline value

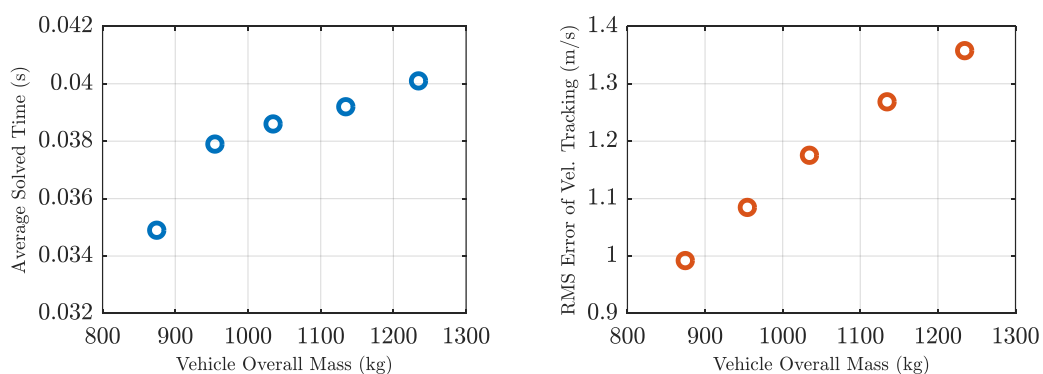
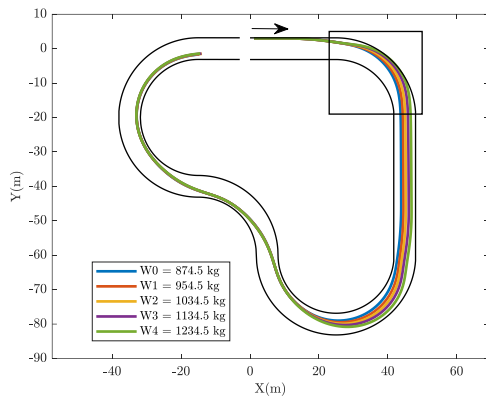
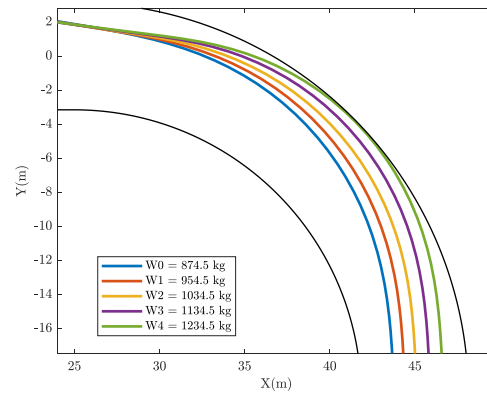


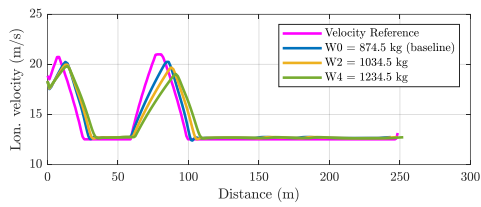
Figure 7-17 Robustness study on average solved time and RMS error of velocity tracking by varying total vehicle mass



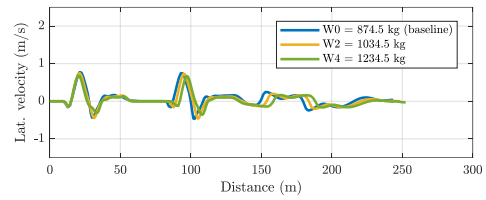
(a) Trajectory



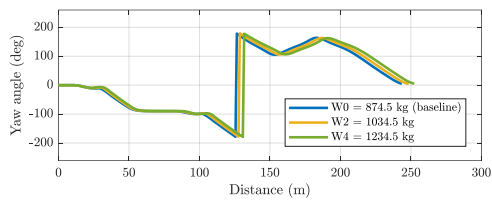
(b) Enlarged trajectory



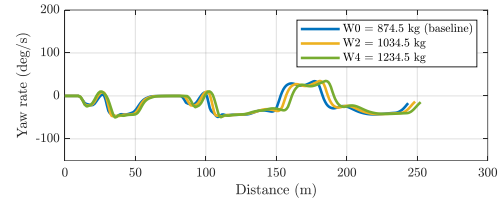
(c) Longitudinal Velocity



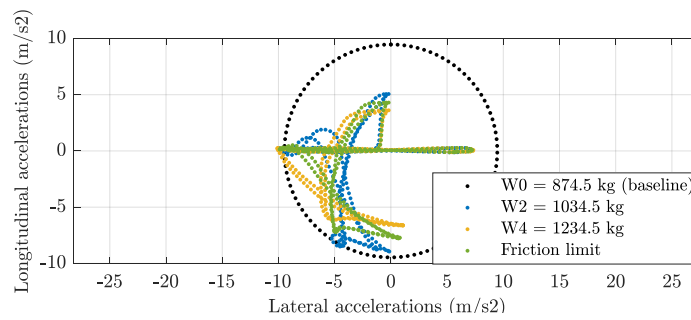
(d) Lateral Velocity



(e) Yaw Angle



(f) Yaw Rate



(g) G-G Diagram

Figure 7-18 Results comparison by varying total vehicle mass

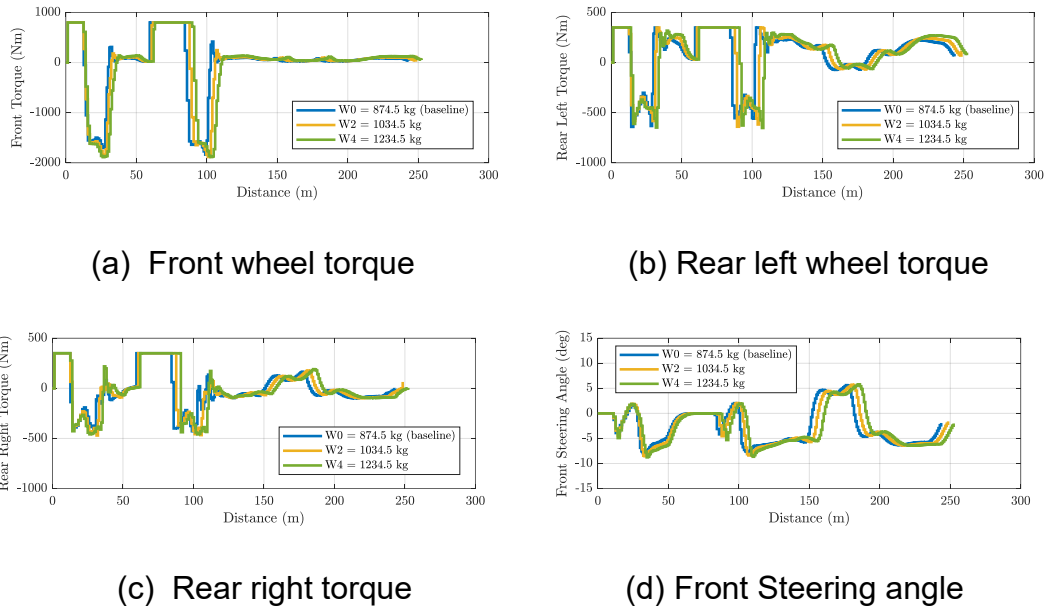


Figure 7-19 Control inputs comparison by varying total vehicle mass

The robustness study assessed the impact of varying total vehicle mass on the generated trajectory, computational performance and velocity tracking accuracy of the PLAPC. The results, summarized in Table 7-9, demonstrate a clear trend where increasing vehicle mass leads to a gradual rise in average solved time, peaking at 0.0401 seconds for the heaviest configuration with solver errors (4 occurrences) are recorded. This suggests that as vehicle mass increases, the it becomes more computationally demanding, pushing the solver closer to its limits.

From a control performance perspective, the RMS error of velocity tracking follows a worsening trend, rising from 0.9921 m/s at the baseline mass (874.5 kg) to 1.3577 m/s at the highest tested mass. This indicates that as the vehicle mass increases, the controller struggles to maintain the desired velocity profile with the same level of accuracy. This degradation in performance is likely due to the additional inertia requiring greater corrective forces, which may exceed the capabilities of the control and actuation system.

The trajectory comparison in Figure 7-18 shows noticeable variations in vehicle trajectories as the total vehicle mass increases with the heavier vehicles

configurations tend to deviate outward in the corners, indicating a reduction in cornering performance. This is due to the increased in mass that increased the inertia and the redistribution of normal loads on the tires, leading to reduced lateral responsiveness.

Overall, these findings indicate that while the PLAPC controller maintains reasonable robustness across moderate mass variations, performance degradation becomes significant at the highest tested masses. The results highlight the importance of limiting the maximum total vehicle mass to ensure the vehicle's safety and maintain stable performance.

7.4.2 Varying Electric Motors Time Delay

Another essential aspect to be tested is the variation in electric motor delay. This study simulates the effects of communication delays, load conditions, power supply, physical condition, or environmental factors that could impact the actuator's ability to deliver torque in a timely manner. A variation of 0 to 40 ms (Baseline time delay is 10 ms) was tested to analyze its impact. The modifications were implemented in IPG CarMaker under the motor control torque command parameters by adding "Transport Delay" blocks as seen in Figure 6-4.

Table 7-10 Comparison result of varying electric motor time delay

Time delay (ms)	0	10	20	40
Average solved time (s)	0.0347	0.0325	0.0367	0.0380
Peak solved time (s)	0.0696	0.0643	0.0793	0.0777
No. of solvers error	0	0	0	2
RMS error of velocity tracking (m/s)	0.95629	0.9898	1.0309	1.1060
Lap time (s)	17.775	17.775	17.925	18.150

Note: Blue cells indicate the baseline value

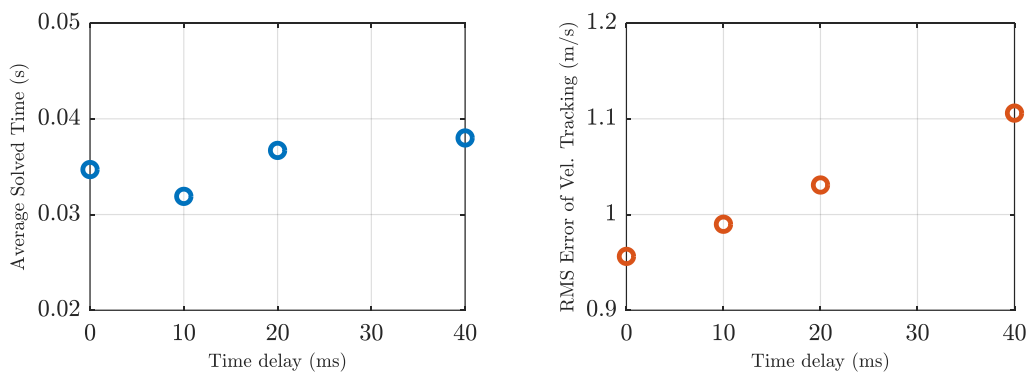


Figure 7-20 Robustness study on average solved time and RMS error of velocity tracking by varying electric motor time delay

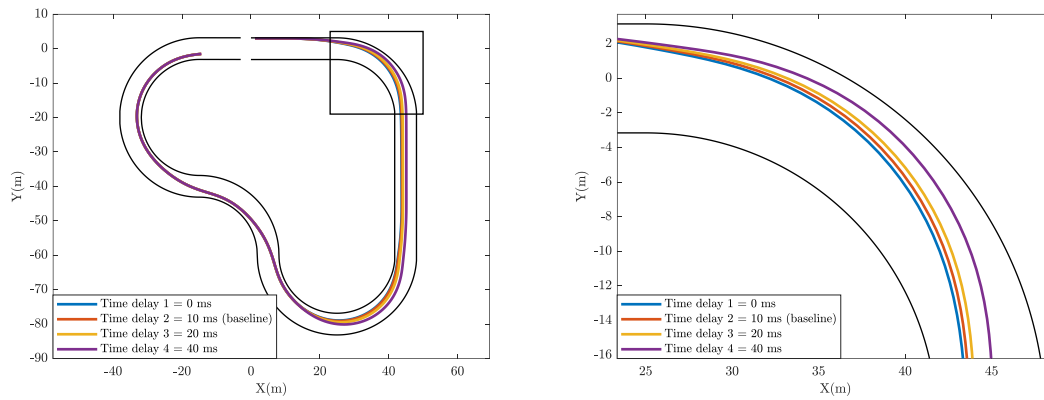
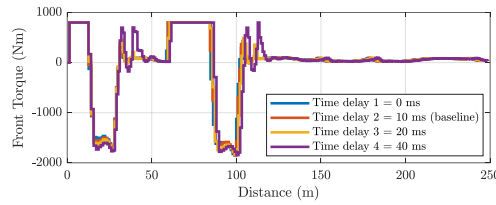
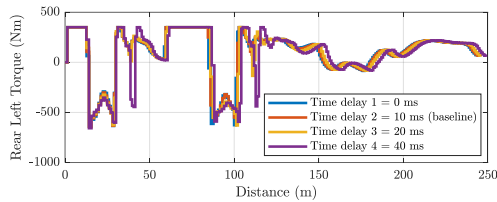


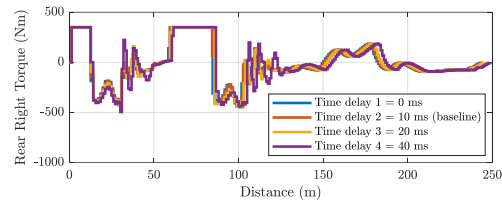
Figure 7-21 Vehicle trajectory comparison by varying electric motor time delay



(a) Front wheel torque



(b) Rear left wheel torque



(c) Rear right torque

Figure 7-22 Torque inputs comparison by varying electric motor time delay

The robustness study on electric motor time delay was conducted to assess its impact on trajectory tracking, computational performance, and torque inputs. Delays ranging from 0 to 40 ms were tested in IPG CarMaker to evaluate the effect of actuator response lag under extreme conditions. The results show that delays up to 20 ms had minimal impact on vehicle trajectory, while the highest

tested delay of 40 ms resulted in solver errors, wider trajectory deviations, and noticeable performance degradation.

As shown in Table 7-10, both average and peak solver times remained relatively consistent across all delay scenarios, with no solver errors recorded up to 20 ms. However, at a 40 ms delay, two solver errors occurred, indicating potential stability concerns under extreme delay conditions. Similarly, the RMS velocity tracking error remained below 1 m/s for all cases except the 20 ms and 40 ms delays, where a slight increase was observed. Lap time differences were minimal, suggesting that speed tracking was largely unaffected except at the highest delay setting.

Furthermore, trajectory analysis in Figure 7-21 confirms that the vehicle followed a nearly identical path for delays up to 20 ms of delay, indicating that actuator delay within this range does not significantly impact trajectory tracking. However, small deviations were observed in the 40 ms case, where the vehicle followed a slightly wider path which likely due to delayed torque execution. Likewise, torque inputs in Figure 7-22 showed minor variations across most delay cases. However, at 40 ms, small oscillations and lag in torque delivery were observed, indicating the controller's unable to deliver torque commands in a timely manner.

Overall, the study confirms that electric motor time delays up to 20 ms do not significantly affect trajectory tracking, computational load, or torque distribution. However, at the extreme end (40 ms), number of solver errors and trajectory deviations indicate performance degradation, highlighting a potential limit in the controller's robustness. Despite this, the PLAPC controller remained stable across practical delay variations, ensuring reliable vehicle performance under typical communication or actuation delays. Nevertheless, it is essential to define a maximum allowable delay threshold to ensure safe and robust operation.

7.4.3 Combination Robustness Test Case

To further assess the robustness of the PLAPC, the following section evaluates the combined effects of variations in total vehicle mass and delays in the electric motor command. Two distinct cases are investigated:

- Combination 1: Using maximum parameters values before having solver errors based on previous individual robustness study
- Combination 2: Maximum parameters values tested

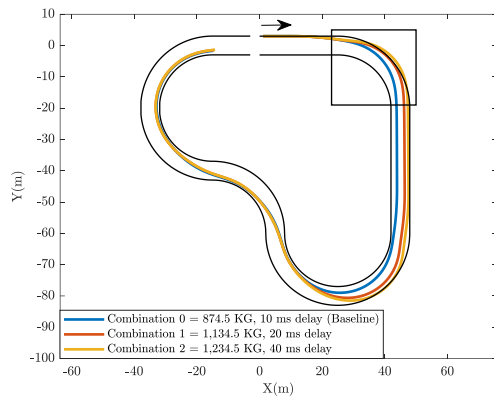
Table 7-11 Combined case scenario change of parameters

Combination no.	0	1	2
Case scenario	Baseline	Max. value before solver errors	Max. value tested
Total vehicle mass (kg)	874.5	1134.5	1234.5
Additional mass (kg)	0	+ 260	+ 360
Electric motor time delay (ms)	10	20	40
Additional time delay (ms)	0	+ 10	+ 30

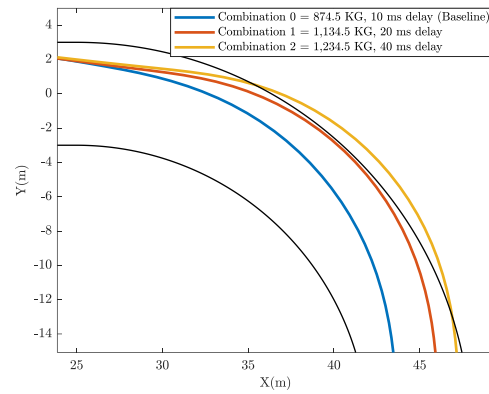
Table 7-12 Comparison result of combined case scenario

Case scenario	Baseline	Max. value before solver errors	Max. value tested
Total vehicle mass (kg)	874.5	1134.5	1234.5
Electric motor time delay (ms)	10	20	40
Average solved time (s)	0.0345	0.0376	0.0416
Peak solved time (s)	0.0643	0.706	0.0883
No. of solvers error	0	0	5
RMS error of velocity tracking (m/s)	0.9921	1.3080	1.4477
Lap time (s)	17.775	18.375	18.825

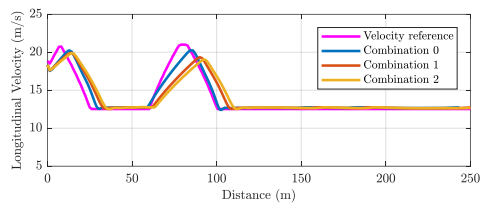
Note: Blue column indicate the baseline value



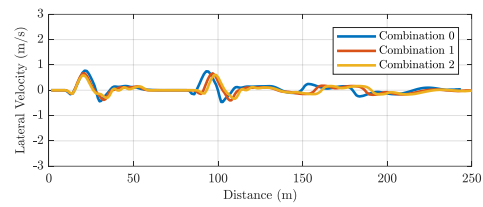
(a) Trajectory



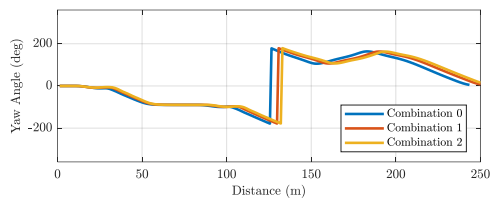
(b) Enlarged trajectory



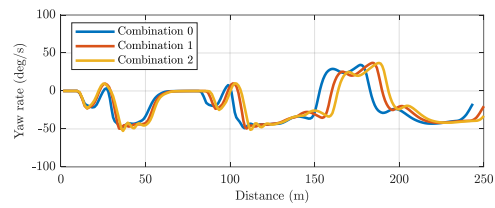
(c) Longitudinal Velocity



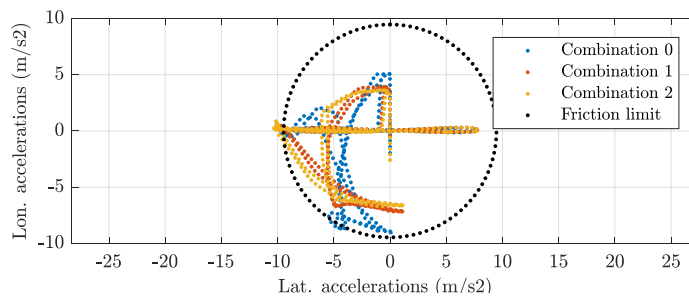
(d) Lateral Velocity



(e) Yaw Angle

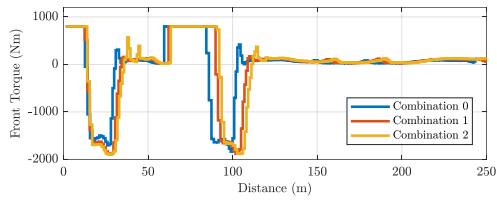


(f) Yaw Rate

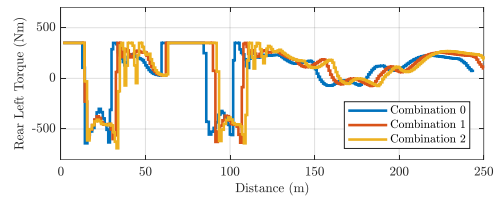


(g) G-G Diagram

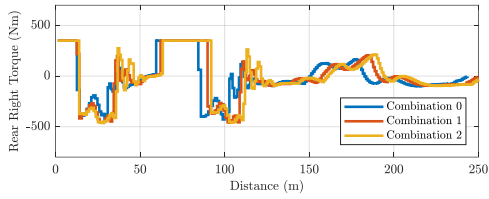
Figure 7-23 Result comparison between all combinations with minimum distance objective



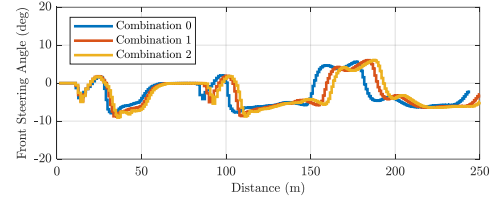
(a) Front wheel torque



(b) Rear left wheel torque

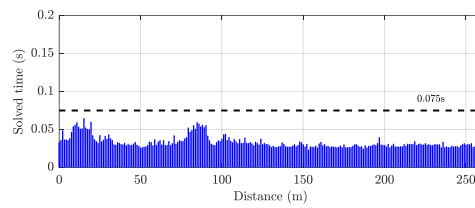


(c) Rear right wheel torque

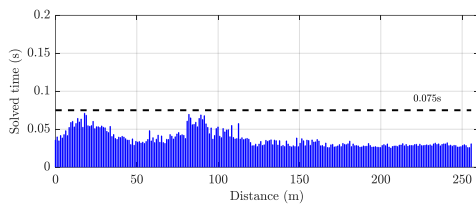


(d) Front Steering angle

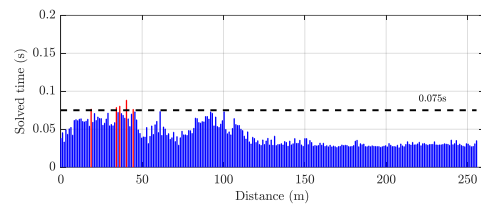
Figure 7-24 Control inputs comparison result between all combinations with minimum distance objective



(a) Combination 0 as the baseline



(b) Combination 1



(c) Combination 2

Figure 7-25 IPOPT solver information for PLAPC simulations for all cases

The combined robustness test was conducted to evaluate how increased in total vehicle mass and electric motor command delay affect the performance of the PLAPC controller. The test involved two scenarios: Combination 1, with a moderate increase in total vehicle mass (1134.5 kg total) and motor delay (20 ms), and Combination 2, featuring a substantial increase in total vehicle mass (1234.5 kg total) along with a longer motor delay (40 ms). The performance in these scenarios was compared against the baseline condition (874.5 kg total mass, 10 ms delay) to clearly quantify performance degradation under more challenging combined conditions.

As shown in Table 7-12, the average solved time increased from 0.0345 seconds to 0.0416 seconds, indicating a higher computational demand in the Combination 2 scenarios. However, no solver errors were recorded except for Combination 2 with 5 solver errors instances. Velocity tracking performance degraded, with the RMS error increasing from 0.9921 m/s to 1.4477 m/s, suggesting that the controller had to work harder to maintain velocity consistency under the combined worst case situations. Additionally, lap time increased from 17.775 s to 18.825 seconds, indicating a slower overall performance due to this combined scenarios.

Trajectory analysis in Figure 7-23 (a) shows that the Combination 2 resulted in a deviated from the path, particularly in the first corner sections, confirming that the impact of high vehicle mass and lag in torque delivery due to significant delays. In addition, the G-G diagram as in Figure 7-23 (g) highlights a shift in acceleration limits, where the Combination 2 scenarios exhibits lower longitudinal acceleration and more lateral acceleration levels. This indicates that increased vehicle mass significantly affected both acceleration and braking performance. Regarding control inputs, Combination 2 exhibited oscillations due to delays in torque delivery, reflecting the influence of the increased electric motor command delay.

Overall, the results indicate that while the PLAPC controller remains stable under Combination 1 scenarios, while notable performance degradation occurs in Combination 2 regarding velocity tracking accuracy, solver computation time, and lap time. Moreover, the deviation or overrun from the intended path observed in Combination 2 significantly depends on the initial velocity when approaching the

corner. To mitigate these effects, it is recommended to incorporate a vehicle mass variation estimator. This could involve, for instance, estimating suspension travel while the vehicle is stationary during initialization, thereby enabling the controller to detect and adapt to changes in total vehicle mass. Such proactive estimation would improve the controller's robustness, enhance vehicle performance, and maintain safety. This finding highlights an essential consideration for future enhancements in PLAPC design and underlines the importance of acknowledging its current limitations.

7.5 Summary

This chapter is critical in highlighting the development of PLAPC. It covers the formulation of the NMPC problem, the definition of constraints, sensitivity and robustness study conducted to identify optimal parameter settings and understand the developed controller's behaviour and limitations. From the sensitivity study, node 3 for integration steps, a prediction horizon of 0.5 seconds (10 nodes), and a sampling time of 0.075 seconds are recommended for this work, consistently balancing accuracy, efficiency, and real-time feasibility across scenarios. While, a robustness study was conducted to assess the controller's performance under varying total vehicle mass and electric motor time delays. The results able to identify the limit parameters of the controller and must be taken into consideration to ensure the safety of the autonomous vehicle. These insights provide a strong foundation for further testing and refinement of the controller's development.

8 Test Result & Discussions

8.1 Introductions

This chapter builds upon the problem formulation, sensitivity and robustness study outlined in the previous chapter by presenting the simulation testing results conducted in IPG CarMaker. The PLAPC underwent testing across three main path scenarios: a single-corner short path, a multiple-corners extended path, and a long-distance path on an actual circuit. Additionally, it utilized two distinct velocity reference profiles: one derived from optimal solutions and the other based on a generated curvature-based velocity profile. These tests evaluate the controller's effectiveness under varying path and velocity conditions, demonstrating its adaptability and performance.

The initial validation compares the Path Tracking Controller (PTC) with from [115] (further described in Appendix B) to assess its tracking accuracy against the optimal solution trajectory. Additionally, the PLAPC trajectory generation is evaluated against the same optimal solution trajectory.

By benchmarking these results, the simulations not only confirm the feasibility of the generated optimal trajectory but also establish a baseline for further PLAPC development and improvements in simultaneous planning and tracking strategies. This controller will undergo further testing under various conditions, as outlined in the simulation testing matrix in Table 8-1. In this study, infeasibility problems, overrun problems, solved beyond the sampling time and reaching maximum number of iterations from the solver is defined as “solver errors”.

Table 8-1 Simulation testing matrices in Chapter 8

Type of velocity profile (Refer to Table 6-1)	Type of Controller	Type of Path	Objective Function Tested
A	PTC PLAPC	Single corner short path	Optimise comfort
B	PTC PLAPC	Single corner short path	Minimise distance to reference line ahead
C	PLAPC	Multi corner extended path	Minimise distance to reference line ahead
D, F	PLAPC	Multi corner extended path	Minimise distance to reference line ahead, Optimise comfort
E	PLAPC	Long distance path	Minimise distance, Optimise comfort, Optimise energy
D, F	PLAPC	Long distance path	Minimise distance to reference line ahead, Optimise comfort

8.2 Single Corner Short Path Evaluation

The results from the optimal solution discussed in Chapter 5 provide a baseline for developing PLAPC in this research. This subchapter demonstrates the integration of the optimal solution into the path-tracking application, validating the optimal solution while offering a more comprehensive understanding of developing the PLAPC.

The comparison between the path-tracking controller (PTC) and the proposed PLAPC is essential to evaluate how well each approach handles driving objectives, namely minimised time and optimised comfort. While the PTC represents a traditional approach to trajectory point tracking based on pre-computed trajectory, the PLAPC integrates simultaneous planning and tracking that dynamically generates trajectories in real-time based on a reference line ahead without pre-computed trajectories. This comparison highlights the strengths and limitations of each approach, offering insight into the benefits of the developed controller.

The PTC has been developed based on the NMPC method and simulated in the IPG CarMaker environment, with a double track 3 DOF vehicle dynamic model. The formulation considers the nonlinearities of the vehicle dynamics and the constraints of the control inputs, except the PTCs do not consider the load-dependent Magic Formula tyre model; instead, fixed tyre parameters were used. Further details of this PTC can be found in [113] and as described in Appendix B.

In addition, the bicycle model used in the original formulation of PTC in [113] for reference generations of steering angle inputs has been removed. Hence, x_{ref} and u_{ref} are obtained entirely from the offline optimal solution results. The PTC has a sampling time of 0.02 seconds and a prediction horizon of one second for this purpose. Regarding the weighting matrices, the weighting matrices used as described in Appendix A.4.1.

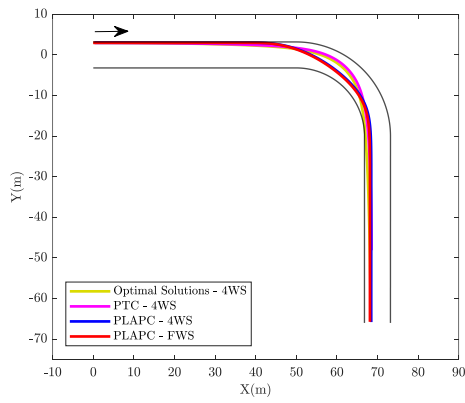
In this test, a short path single corner of a 90-degree was used to evaluate the effectiveness of both PTC and PLAPC. Additionally, it is important to note that the PTC uses a 4WS vehicle configuration. Therefore, it is essential to test the

PLAPC with a 4WS configuration for a fair comparison and include the FWS configuration as a baseline for further evaluation of the controller's development. In addition, both PTC and PLAPC incorporated with the same torque distribution method using Active Torque Distribution (ATD) or torque vectoring for rear wheels and equal distribution on the front wheels since both are using the same vehicle topology as shown in Figure 4-1.

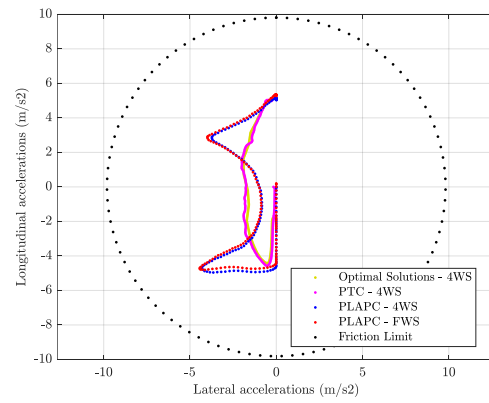
For the validations, only two extreme ends were considered as references from optimal solution results, minimised time and optimised comfort. Therefore, the objective function for PLAPC is to minimise the distance to the reference line ahead and optimise for comfort. Table 8-2 presented the test matrices for this evaluation,

Table 8-2 Simulation testing matrices for comparison between PLAPC and PTC

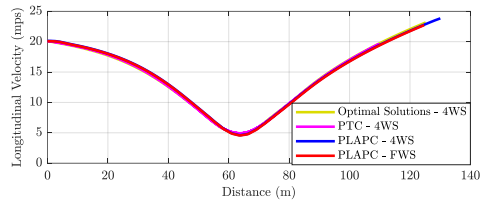
Test no.	Type of Controller	Vehicle topology	Obj. function	Tire load-dependent coefficient	Type of velocity profile	Trajectory references
1	PTC	4WS	Track the references	Fixed	Optimal velocity	Optimal trajectory
	PLAPC	4WS	Optimised comfort	Varied	Optimal velocity	No trajectory reference
	PLAPC	FWS	Optimised comfort	Varied	Optimal velocity	No trajectory reference
2	PTC	4WS	Track the references	Fixed	Optimal velocity	Optimal trajectory
	PLAPC	4WS	Minimised distance	Varied	Optimal velocity	No trajectory reference
	PLAPC	FWS	Minimised distance	Varied	Optimal velocity	No trajectory reference
3	PTC	4WS	Track the references	Fixed	Optimal velocity	Optimal trajectory
	PLAPC	4WS	Minimised distance	Fixed	Optimal velocity	No trajectory reference



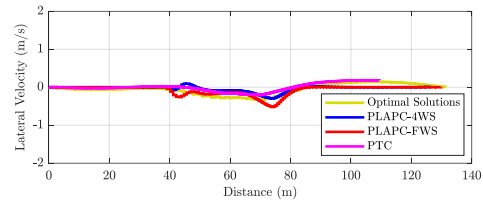
(a) Trajectories



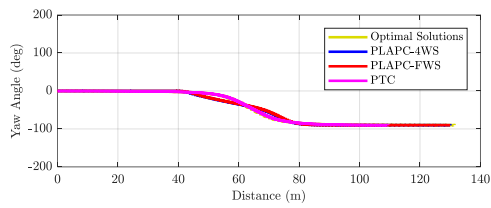
(b) G-G Diagram



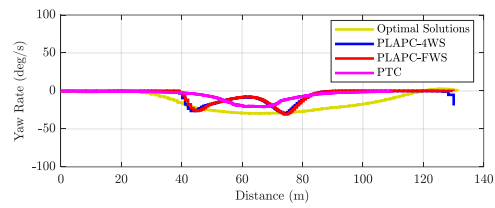
(c) Longitudinal Velocity



(d) Lateral Velocity

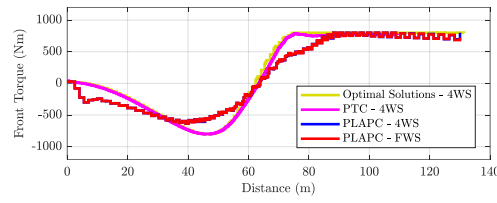


(e) Yaw Angle

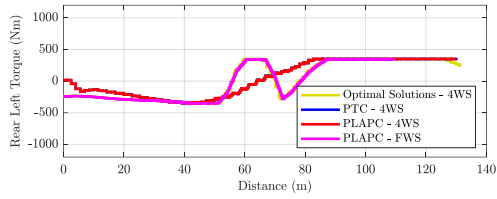


(f) Yaw Rate

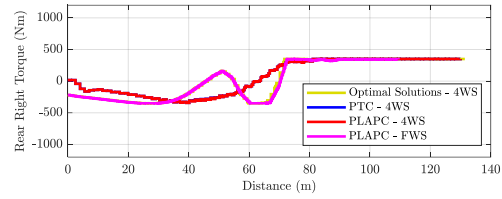
Figure 8-1 Result comparison of PTC and PLAPC with optimised comfort solution as a reference



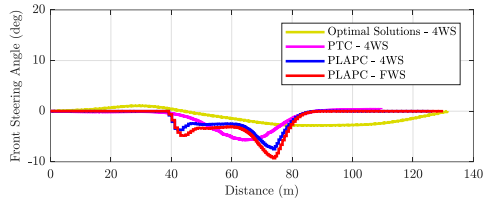
(a) Front-wheel torque



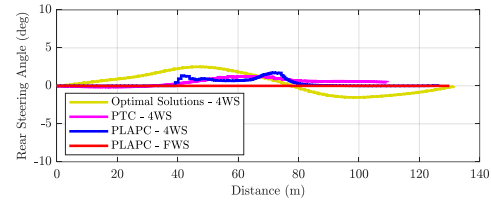
(b) Rear left wheel torque



(c) Rear right wheel torque

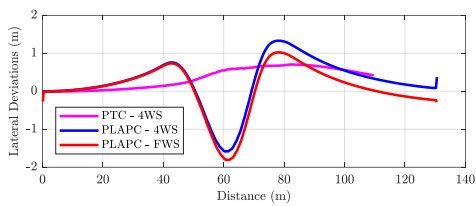


(d) Front Steering angle

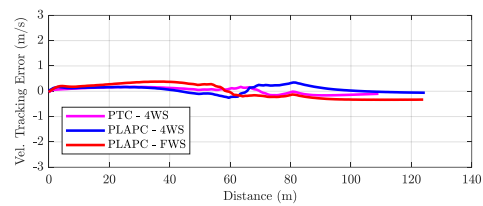


(e) Rear Steering angle

Figure 8-2 Control inputs comparison of PTC and PLAPC with optimised comfort solution as a reference

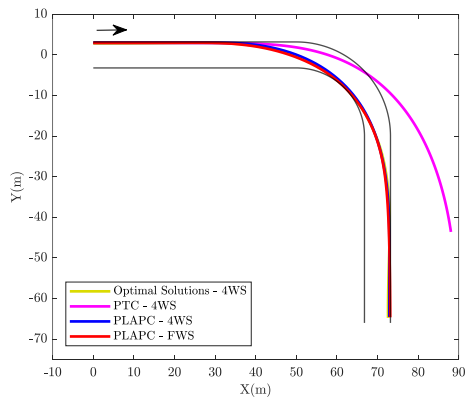


(a) Lateral deviations

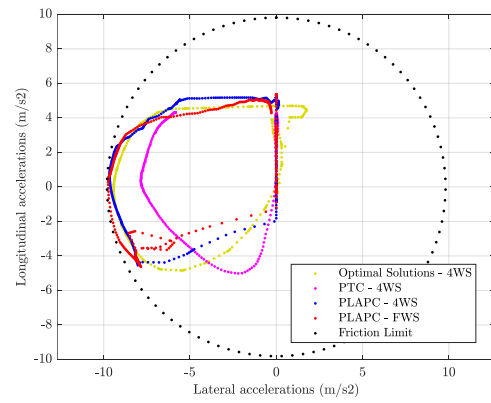


(b) Velocity deviations

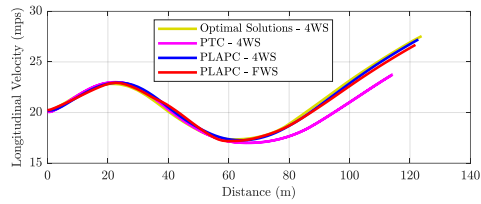
Figure 8-3 Deviations comparison of PTC and PLAPC with optimised comfort solution as a reference



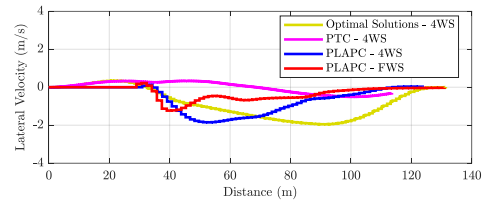
(a) Trajectories



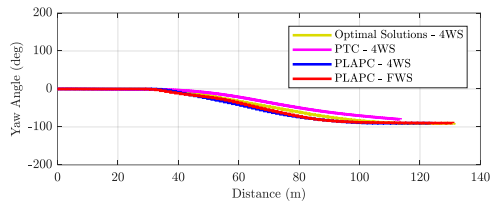
(b) G-G Diagram



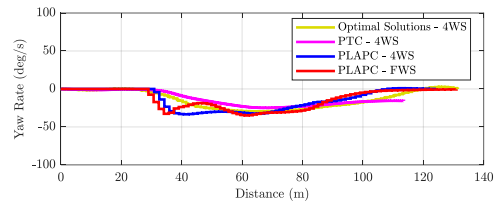
(c) Longitudinal Velocity



(d) Lateral Velocity

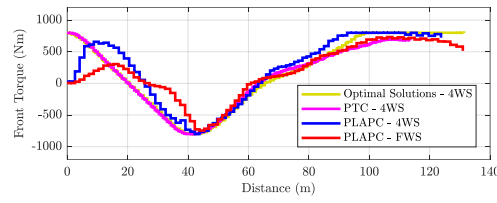


(e) Yaw Angle

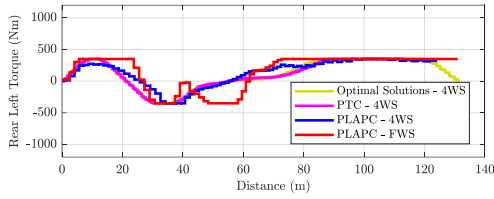


(f) Yaw Rate

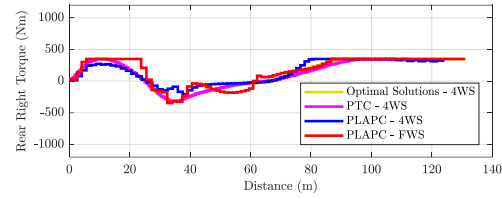
Figure 8-4 Result comparison of PTC and PLAPC using a minimum time solution as a reference



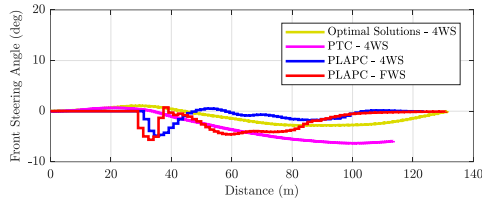
(a) Front-wheel torque



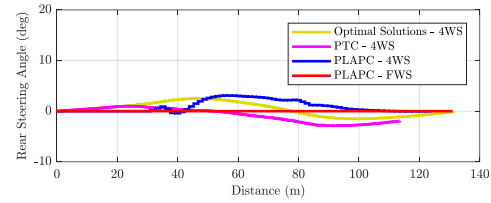
(b) Rear left wheel torque



(c) Rear right wheel torque

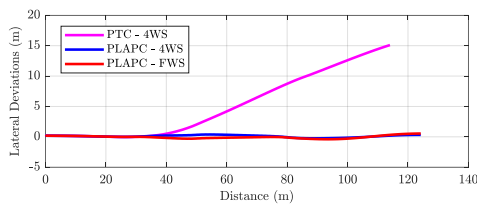


(d) Front Steering angle

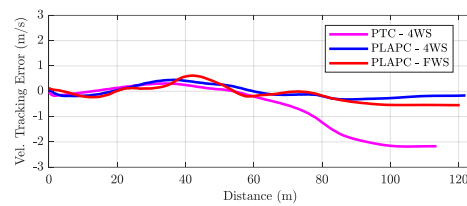


(e) Rear Steering angle

Figure 8-5 Control inputs comparison of PTC and PLAPC using a minimum time solution as a reference



(a) Lateral deviation



(b) Velocity deviations

Figure 8-6 Deviations comparison of PTC and PLAPC using a minimum time solution as a reference

Table 8-3 Numerical comparison of PTC and PLAPC with an optimised comfort solution as a reference

Key Metrics	PTC-4WS	PLAPC-4WS	PLAPC-FWS
Maximum trajectory tracking error to the optimal solution (m)	0.7035	1.572	1.7954
RMS velocity tracking error to the optimal solution (m/s)	0.1367	0.2190	0.21145

Table 8-4 Numerical comparison of PTC and PLAPC with a minimum time solution as a reference

Key Metrics	PTC-4WS	PLAPC-4WS	PLAPC-FWS
Maximum trajectory tracking error to the optimal solution (m)	Deviated out of the path	0.385	0.545
RMS velocity tracking error to the optimal solution (m/s)	1.1357	0.24409	0.33419

The results reveal distinct differences in terms of trajectories, accelerations and velocity tracking between the PTC (magenta), PLAPC-4WS (blue), and PLAPC-FWS (red) across different objectives, as seen in Figure 8-1 and Table 8-3 for optimised comfort and Figure 8-4 and Table 8-4 time-optimal as reference.

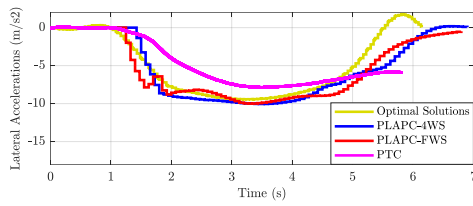
Figure 8-1 presents the results for the optimised comfort objective for both the PTC and PLAPC performed effectively, with minimal deviations from the optimal solution. In subplot Figure 8-1 (a), all trajectories of optimal solution result (yellow), PTC result (magenta), PLAPC-4WS (blue), and PLAPC-FWS (red) results were aligned closely, indicating strong path adherence. This accuracy is further confirmed in subplot Figure 8-1 (b), where both controllers maintain low lateral error within 2 meters, as seen in Table 8-3. The acceleration plot in Figure 8-1 (c) reveals that while the PLAPC has slight fluctuations, the PTC achieves

smoother tracking since it tracks exact coordinates. Finally, the velocity profiles in Figure 8-1 (d) are nearly identical across all methods, reflecting the lower velocity demands of the optimised comfort objective. Overall, both controllers adapt well to smooth, low-velocity requirements, with PTC excelling in acceleration stability and the PLAPC able to maintain close to the optimal solution trajectory adherence.

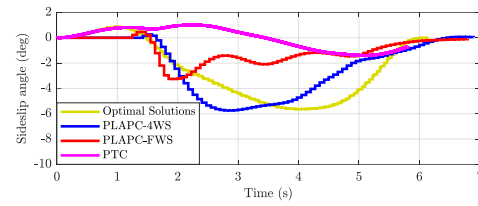
In Figure 8-4, the performance of the PTC and PLAPC with a minimum time solution as a reference shows significant differences. In Figure 8-4 (a), both PLAPC-4WS (blue) and PLAPC-FWS (red) closely aligned with the optimal solution trajectory (yellow), while the PTC (magenta) deviated significantly. This divergence is emphasized in Figure 8-6 (a), where the PTC's lateral error rises sharply, indicating difficulty in maintaining an aggressive trajectory, whereas the PLAPC manages a stable, low lateral deviations for both 4WS and even with the FWS steering systems. The acceleration plot Figure 8-4 (b) reveals that the PLAPC correlates well with optimal solutions, while the PTC struggles to meet the acceleration demands. Finally, in Figure 8-4 (c), the velocity profile confirms that the PLAPC closely tracks the optimal velocity profile, while the PTC has significant velocity deviation, as seen in Figure 8-6 (b) and Table 8-4. However, the velocity tracking for the PLAPC-FWS has the highest deviation due to a longer reference line distance at 7.8 meters ahead in order to meet the optimal trajectory reference as compared to PLAPC-4WS at 5.2 meters, which reduced the velocity tracking accuracy as mentioned in the Sub-Chapter 7.3.4. Overall, the PLAPC proves more effective for the minimum distance to reference line ahead objective, maintaining velocity and a closer trajectory to the optimal solution, while the PTC falls short in this aggressive manoeuvre.

It is worth further discussing the significant differences in lateral error between the PTC and PLAPC. These two controllers differ fundamentally in their reference methods: the PTC follows a conventional coordinate-based approach with a pre-computed trajectory from optimal solution result [113], while the PLAPC relies purely on a perpendicular line-ahead as a reference that generates the trajectory on the fly and do the control. The PTC exhibits significant deviation high varied velocity, which diminishes its tracking accuracy. However, it shows excellent tracking capability in the lower velocity, as seen in, Figure 8-1. This discrepancy in lateral deviation underscores the limitations of the PTC in varied high-velocity challenging scenarios.

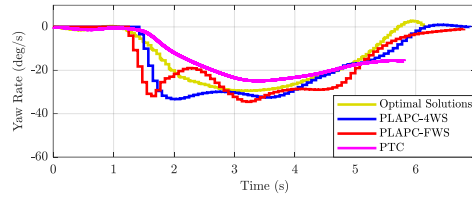
In contrast to PLAPC, which solved the planning and tracking simultaneously, PTC solved hierarchically by the need to feed pre-computed trajectory and velocity references to its traditional path-tracking formulation. The deviations observed in the PTC results with time-optimal reference are caused by its decoupling limitation, as it focuses mainly on lateral dynamics [113] neglecting longitudinal dynamics. This limitation makes the PTC struggle to manage sharp turns and varied high-velocity scenarios. In contrast, the PLAPC approach dynamically adjusts both longitudinal and lateral dynamics, ensuring that it is feasible, closer to an optimal trajectory and stays within the path.



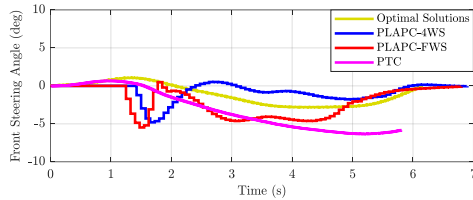
(a) Lateral accelerations



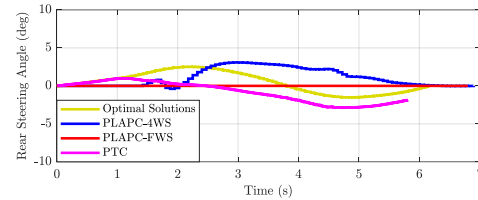
(b) Sideslip angle



(c) Yaw rate



(d) Front steering angle



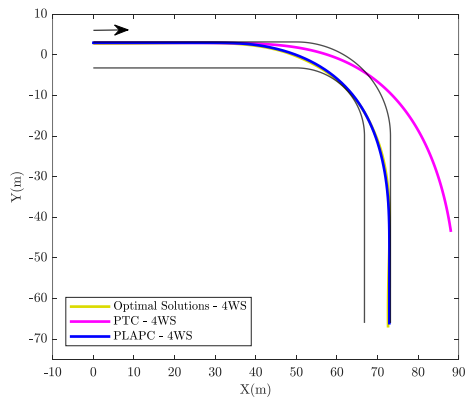
(e) Rear steering angle

Figure 8-7 Plots comparison in time domain

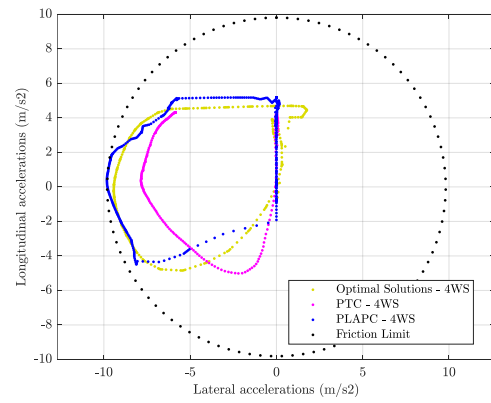
This is further analyzed on the time-domain plots as in Figure 8-7, where PTC exhibited a slower yaw rate response, primarily due to less aggressive steering adjustments which delayed the generation of the required lateral forces, preventing the vehicle from achieving the necessary cornering performance in time. Additionally, PTC's higher sideslip angle indicates increased lateral slippage, meaning the vehicle experienced more sideways which made it deviated out. The PTC's steering inputs of both front and rear were smoother and less reactive, limiting its ability to make sharp turn. In contrast, PLAPC demonstrated a more effective steering input strategy, which improved yaw rate response and allowed the vehicle to generate the necessary lateral forces more efficiently, keeping it closer compare to the optimal solutions trajectory.

Another notable point are the acceleration and trajectory differences observed in Figure 8-1 (b) and Figure 8-3 (b), where the PLAPC shows deviations from the optimal solution under optimised comfort objectives. This discrepancy highlights a limitation inherent to the working principles of the PLAPC method. These differences arise because PLAPC relies on a line-ahead reference and real-time adjustments to maintain the closest possible alignment with the reference line. While the optimal solution computes a global trajectory, PLAPC's real-time corrections, combined with its finite prediction horizon and to meet with the constraints, lead to localized variations in acceleration. This behaviour is solely due to the working principle of PLAPC itself. This contrasts with the PTC, which is good for lower velocity situations, which follow the path more consistently, resulting in a smoother, more stable acceleration profile.

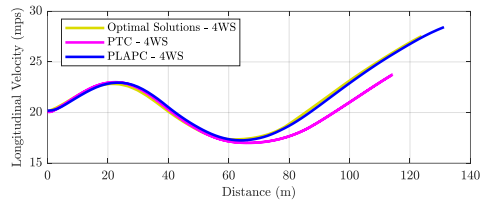
To ensure a fair comparison between PTC and PLAPC, another simulation was conducted by ensuring both controllers were set up identically. Both controllers used the same vehicle topology, equipped with 4WS, rear-wheel ATD and without a tyre load-dependent coefficient. In terms of references, the minimum time solution as a reference for PTC and PLAPC does not have any trajectory reference but only with its objective function to minimise distance to the line ahead for PLAPC.



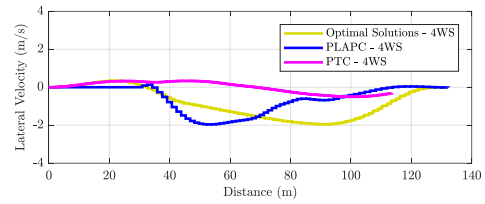
(a) Trajectories



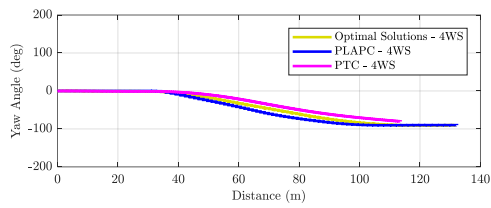
(b) G-G Diagram



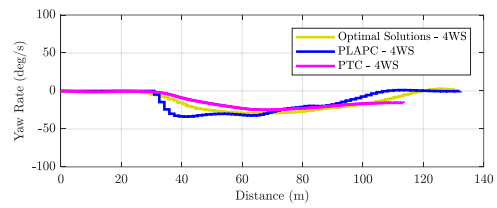
(c) Longitudinal Velocity



(d) Lateral Velocity

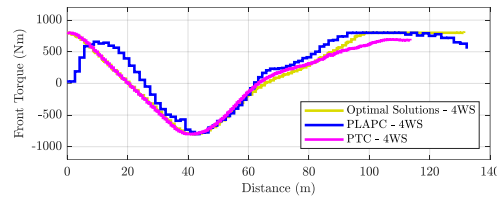


(e) Yaw Angle

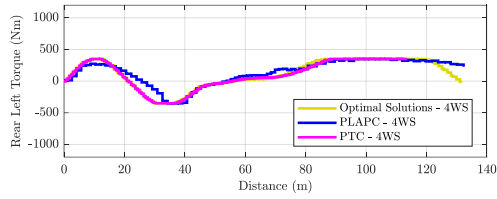


(f) Yaw Rate

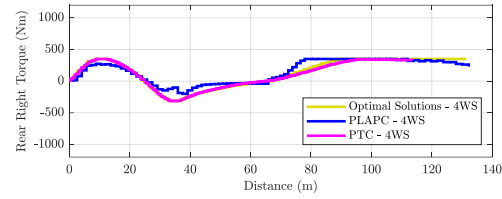
Figure 8-8 Result comparison of PTC and PLAPC using a minimum time solution as a reference without tire load dependent coefficients



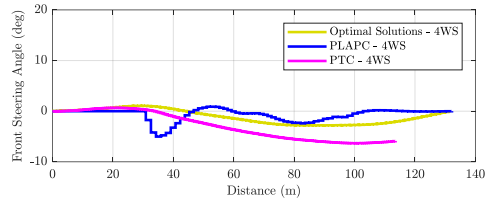
(a) Front-wheel torque



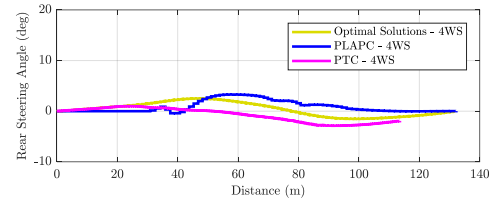
(b) Rear left wheel torque



(c) Rear right wheel torque

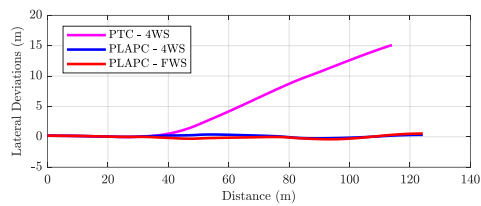


(d) Front steering angle

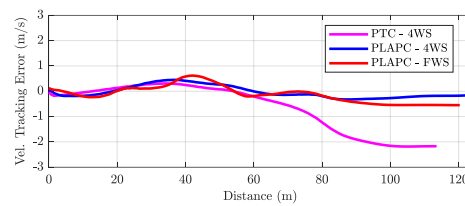


(e) Rear steering angle

Figure 8-9 Control inputs comparison of PTC and PLAPC using a minimum time solution as a reference without tire load dependent coefficients



(a) Lateral deviation



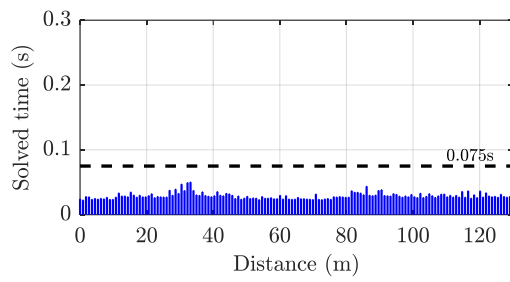
(b) Velocity deviations

Figure 8-10 Deviations comparison of PTC and PLAPC using a minimum time solution as a reference without tire load dependent coefficients

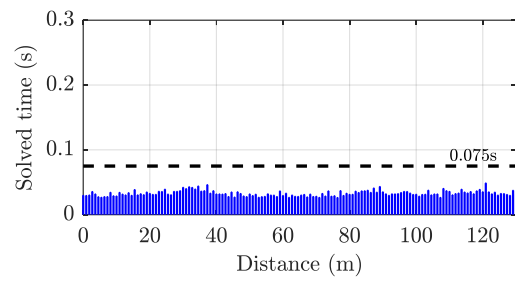
Table 8-5 Numerical comparison of PTC and PLAPC with a minimum time solution as a reference without tire load dependent coefficients

Key Metrics	PTC-4WS	PLAPC-4WS
Maximum trajectory tracking error to the optimal solution (m)	Deviated out of the path	0.328
RMS velocity tracking error to the optimal solution (m/s)	1.128	0.2359

In this study, the peak tire coefficient was assumed to be fixed and equal across all wheels, with $D_{ij} = 1.166$. Since the PTC-based vehicle model, as in [113], could not be modified and has already been tested and proven, adjustments were made to the PLAPC vehicle model instead. Specifically, the tire load-dependent coefficient was removed from PLAPC to match the assumptions in PTC, ensuring a fair comparison. Based on the results are presented in Figure 8-8 and Figure 8-9, PLAPC without tire load-dependent coefficient effectively aligned well with the optimal solution trajectory, with a maximum lateral error to optimal solution of 0.328 meters as shown in Table 8-5. The results were nearly identical to those obtained with the tire load-dependent coefficient in Figure 8-4. Additionally, since the model had fewer variables, this was reflected in a reduced solved time, as seen in Figure 8-13. Therefore, this comparison ensures fairness, and the tire load-dependent coefficient was retained in the PLAPC model to accurately represent an actual vehicle model.

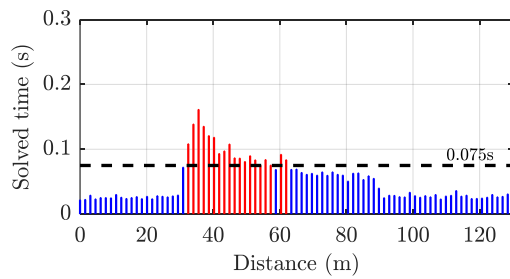


(a) PLAPC-FWS

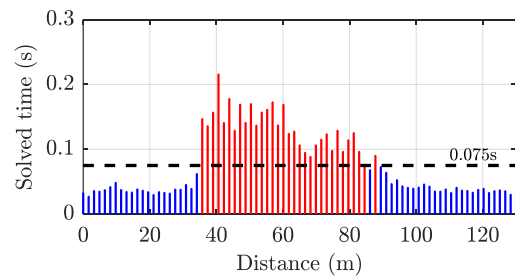


(b) PLAPC-4WS

Figure 8-11 PLAPC solved time comparison with optimised comfort as a reference

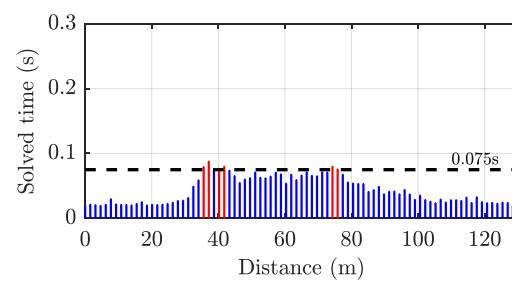


(a) PLAPC-FWS



(b) PLAPC-4WS

Figure 8-12 PLAPC solved time comparison with minimum time as a reference



(a) PLAPC-4WS

Figure 8-13 PLAPC solved time without tyre load-dependent coefficient

Another important thing to highlight is the solved time for the PLAPC, as shown in Figure 8-11 and Figure 8-12. These figures compare the solved times for different objectives of minimum distance to reference line ahead and optimised comfort for both FWS and 4WS. The optimised comfort objective was solved below the threshold. This objective is smoother with lower velocity and less demanding, allowing the controller to maintain lower solved times, highlighting the reduced computational requirements for comfort-focused control. Meanwhile, for the minimum distance objective, the solved time exceeded the 0.075 seconds threshold, especially between 50 and 100 meters distance mark, which is in the corner, indicating a high computational load required for high velocity above 17 m/s at a tight corner. This is aligned with the sensitivity study conducted in Sub-Chapter 7.3.5. Additional simulation for a fair comparison by excluding load-dependent tire coefficient where the solved time shows significant reduction as shown in Figure 8-13 but there are few solver errors that exceeded specified sampling time. Therefore, further studies are required to investigate the developed controller's effectiveness and strategies for its implementation.

Overall, the results of PLAPC are promising compared to the traditional PTC, even when the PTC is tested with a 4WS configuration. A critical observation in this scenario is that the addition of rear-wheel steering does not result in significant differences in the generated trajectory but does incur a higher computational load. This finding establishes a baseline for focusing solely on the FWS vehicle configuration in future studies in the following chapter to further develop the PLAPC.

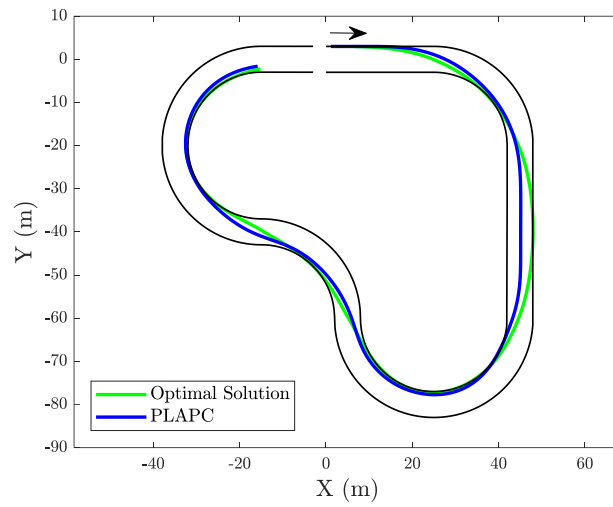
8.3 Multi Corners Extended Path Evaluation with Time Optimal Velocity Profile

The previous chapter compared the PLAPC with offline optimal solutions and a traditional path tracking controller (PTC) [113] under comfort and time optimality scenarios. These evaluations were conducted using both 4WS configurations and the standard FWS setup (only for PLAPC). The results demonstrated that the PLAPC closely aligns with the optimal solution's trajectory, whereas the PTC struggled to stay within path limits, even with the enhanced capability of 4WS configurations.

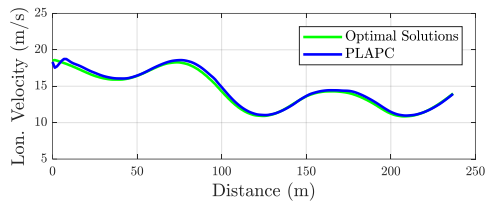
For this reason, the PLAPC has been further developed using FWS configurations, thereby reducing modelling complexity and computational time while aligning with the steering systems commonly found in most vehicles today. This approach ensures that the effectiveness of the PLAPC is validated for conventional vehicles without advanced steering systems, such as 4WS.

To further evaluate the PLAPC, an extended path with multiple corners was introduced, as shown in Figure 4-8. This setup provides a comprehensive validation of how closely the generated trajectory aligns with the optimal solution trajectory.

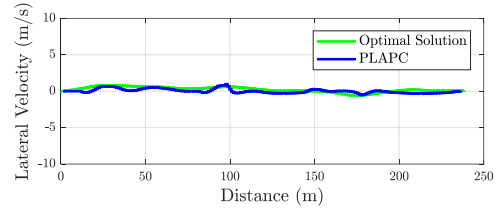
The velocity reference for this test is derived from the optimal solution based on a specific objective function, allowing a direct comparison of the controller's tracking performance under more complex path conditions. However, as noted in the previous chapter, the PLAPC's computational time increased in corners due to the complexity at higher velocities. To address this for real-time implementation, the velocity reference was reduced to a 90% factor of the optimal value, as ideal solutions can often be impractical in such scenarios.



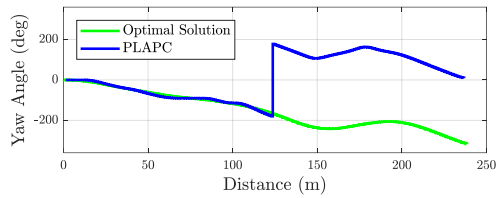
(a) Trajectories



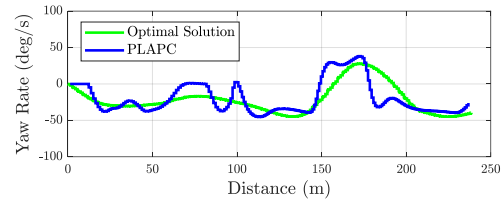
(b) Longitudinal Velocity



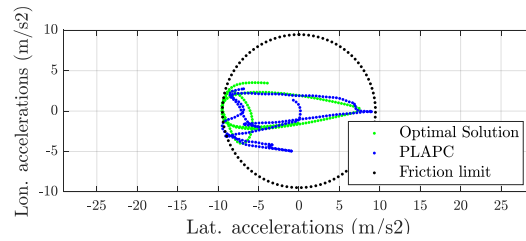
(c) Lateral Velocity



(d) Yaw Angle

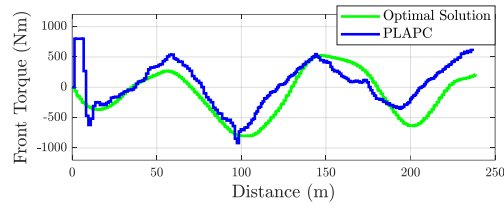


(e) Yaw Rate

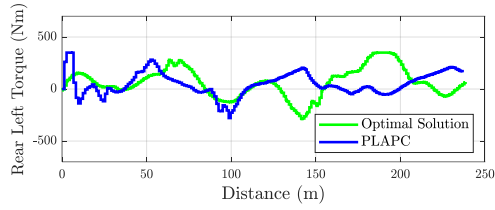


(f) G-G Diagram

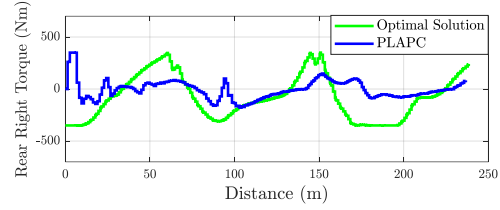
Figure 8-14 Result comparison between minimum time solutions and PLAPC with minimum distance objective



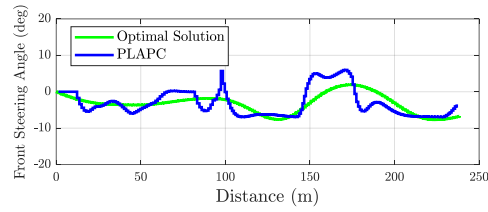
(a) Front-wheel torque



(b) Rear left wheel torque



(c) Rear right wheel torque



(d) Front Steering angle

Figure 8-15 Control inputs comparison result between minimum time solutions and PLAPC with minimum distance objective

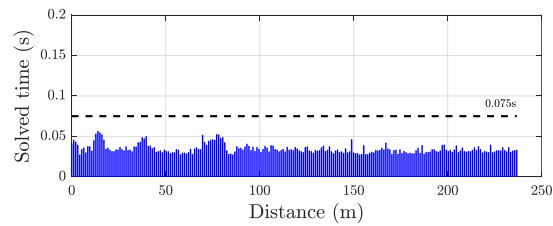


Figure 8-16 IPOPT solver information for PLAPC simulations

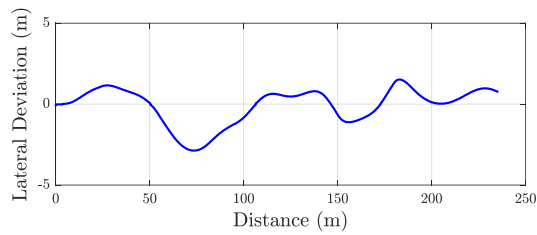


Figure 8-17 Lateral error between minimum time solutions and PLAPC with minimum distance objective

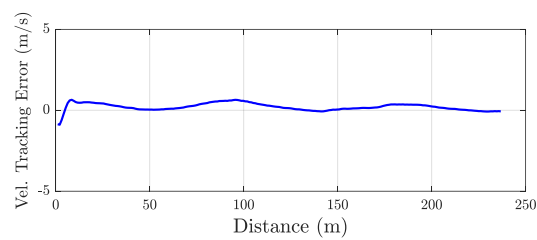


Figure 8-18 Velocity tracking error between time optimal solutions and PLAPC with minimum distance objective

Table 8-6 Result of time objective for the multi-corner simulations

Vehicle topology	FWS
Average solved time (s)	0.0346
Peak solved time (s)	0.0562
No. of solver errors	0
RMS error of velocity tracking (m/s)	0.2918
RMS error of generated trajectory to optimal solution (m)	1.1248

Table 8-7 Result of lap time for the multi-corner simulations

Vehicle topology	FWS
PLAPC lap time (s) – with 90% velocity factor	16.4895
Optimal solution lap time (s)	14.0135

The results presented in Figure 8-14 demonstrated the effectiveness of the PLAPC with FWS vehicle topology. In terms of trajectory, Figure 8-17 (a) shows that the controller is able to match some parts of the reference trajectory derived from the optimal solutions; however, peak deviations of 2.87 meters were observed after the first corner, at the straight path section. While this indicates that the controller can generate and track a feasible path, this level of deviation remains significant in high-precision applications or in tight space, highlighting areas for further refinement and one of the limitation of this method. Additionally, the G-G diagram in Figure 8-14 (f) shows that the friction limit is reached on the lateral side, demonstrating efficient utilization of the tyre forces.

The velocity tracking performance, illustrated in Figure 8-18, highlights that the controller maintains close adherence to the reference velocity profile with deviations below 1 m/s. This level of accuracy ensures minimal compromise in velocity tracking, even during high-velocity scenarios.

Furthermore, the control input demands, including torque and steering angle, shown in Figure 8-15, exhibited a consistent trend, though slight jumps are observed due to real-time implementation with limited horizon length compared to idealized optimal solutions.

Figure 8-16 highlights the real-time feasibility of the controller where the average time and peak time remained below the specified sampling time of 0.075 seconds with zero solver errors of each step. However, increased computational load during high-velocity cornering warrants attention, especially for real-time implementation in challenging scenarios.

In terms of lap time comparison, as presented in Table 8-7, when compared to the optimal solution lap time, the FWS configuration with a 90% velocity factor results in a lap time approximately 2.4 seconds longer than the optimal solution. This deviation is acceptable, given that the PLAPC operates in real-time, while the optimal solution represents an idealized scenario.

Overall, the PLAPC achieves a balance between trajectory accuracy, velocity tracking, and control effort, all while maintaining real-time feasibility. Minor

deviations in lateral position and velocity, particularly in sharp corners, are attributed to the dynamic nature of the perpendicular line-based reference method. Future improvements could focus on alternative velocity profile generation methods and expanding testing across diverse scenarios to evaluate the effectiveness.

8.4 Multi Corners Extended Path Evaluation with Individual Curvature-based Velocity Profile

The previous subchapter derived the velocity reference from an optimal solution velocity profile scaled to 90% of its original values to balance performance and feasibility. While this approach is effective for evaluating the PLAPC close to ideal conditions, it has limitations in real-world applications. Optimal solutions are computationally intensive, often impractical for real-time implementation, and challenging to generate, particularly for longer paths or maps [10] and also if require for re-planning. A curvature-based velocity profile is introduced as an automated method for generating velocity profiles to address these challenges.

This transition is particularly relevant for passenger vehicles, prioritising safety and comfort while occasionally requiring dynamic performance and quick generation of velocity profiles, unlike time-optimal velocity profiles, better suited for racing applications or fixed paths.

This ensures that testing aligns more closely with real-world scenarios, where both comfort and agility are essential. Additionally, it demonstrates the controller's ability to adapt to varying velocity references, bridging the gap between idealized optimal solutions and practical real-time applications. This approach enhances feasibility across diverse vehicle platforms by reducing computational demands and ensuring compatibility with common vehicle configurations.

The evaluation of the PLAPC continues with the creation of two distinct velocity profiles based on the path curvature, as mentioned in Sub-Chapter 6.3. These profiles were tailored to address two contrasting objectives: minimising the distance to the reference line ahead and optimising comfort.

Another point to note is that since the curvature-based velocity profile is not optimised for time, the objective is referred to as minimising the distance to the reference line ahead rather than time.

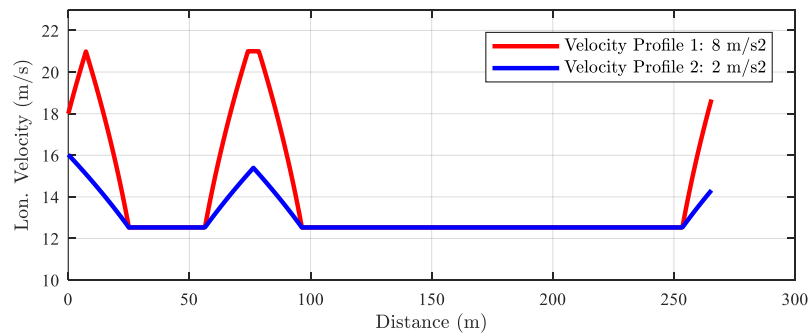
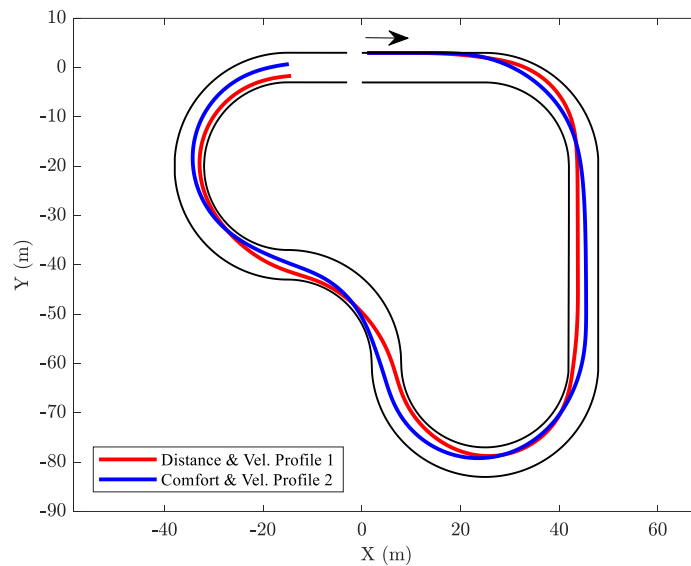
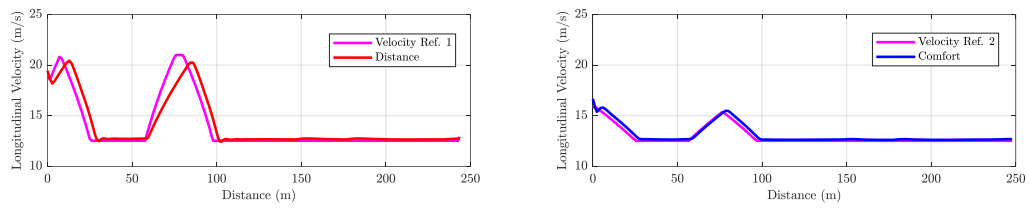


Figure 8-19 Velocity profiles based on the path curvature

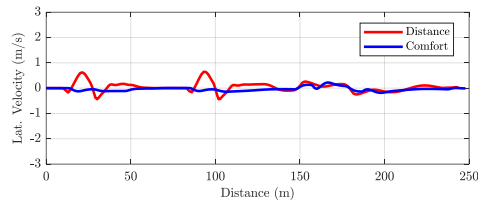
As depicted in Figure 8-19, two velocity profiles are presented. The first, more aggressive profile mean for the distance objective features an acceleration and deceleration slope of 8 m/s^2 . In contrast, the second profile is more moderate, designed for comfort, with a maximum acceleration and deceleration slope of 2 m/s^2 , emphasizing comfort. The initial velocity tested in this evaluation is 18 m/s , with maximum velocity at zero curvature or straight path at 21 m/s and started at the same initial position.



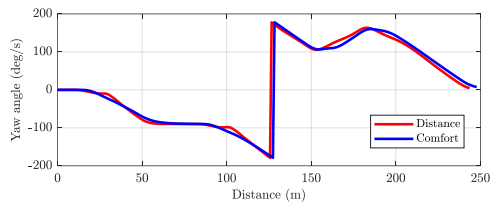
(a) Trajectories



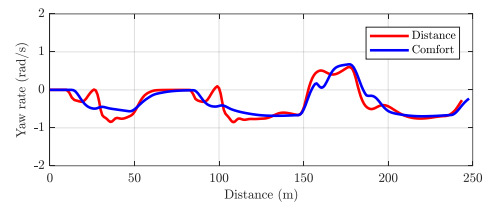
(b) Longitudinal velocity



(c) Lateral velocity



(d) Yaw Angle



(e) Yaw Rate

Figure 8-20 Vehicle states comparison of between objective functions

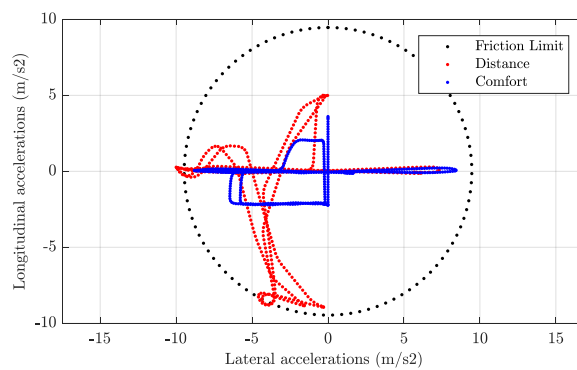
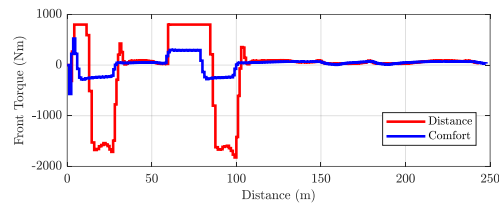
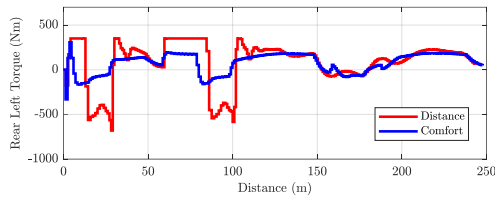


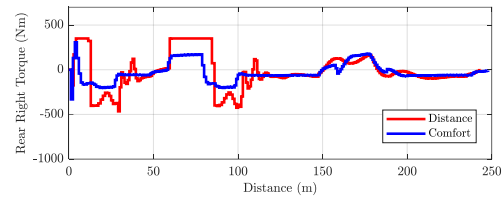
Figure 8-21 G-G diagram comparison of between objective functions



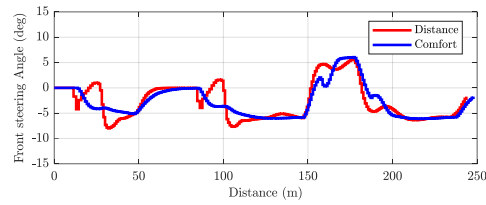
(a) Front wheel torque



(b) Rear left wheel torque



(c) Rear right wheel torque



(d) Front Steering angle

Figure 8-22 Vehicle control inputs comparison of between objective functions

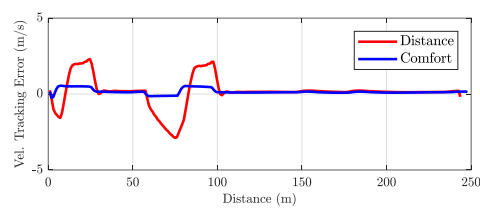


Figure 8-23 Velocity tracking error comparison of between objective functions

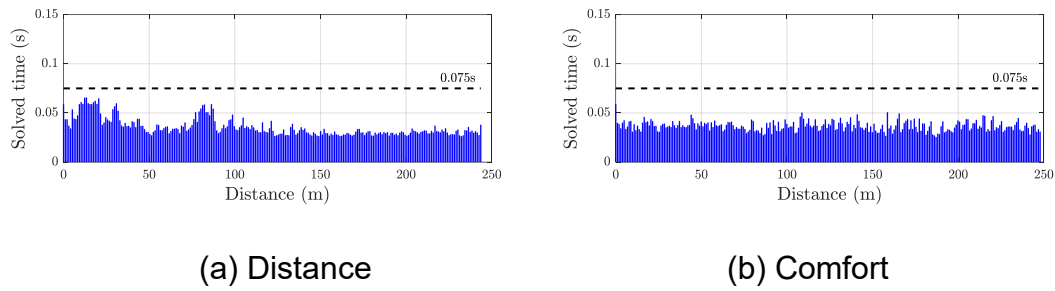


Figure 8-24 IPOPT solver information during the simulations

Table 8-8 Result of various objective simulations between velocity profiles

Velocity profile	1	2	
Objective Functions	Distance	Comfort	Differences
Energy Consumptions (kWh)	0.0734	0.0358	51.2%
RMS error of velocity reference (m/s)	0.9658	0.2302	76.1%
RMS weighted lon. acceleration (m/s^2)	2.2522	0.7320	67.5%
RMS weighted lat. acceleration (m/s^2)	1.6986	1.1486	32.4%
Lap time (s)	17.7750	18.9750	1.2s
Average solved time (s)	0.0356	0.0362	-
Peak solved time (s)	0.0655	0.0453	-
No. of solver errors	0	0	-

Note: Green cells indicate the lowest values in the respective rows.

The simulation results demonstrate the effectiveness of the developed PLAPC in achieving stated objectives with two distinctive velocity profiles. The trajectories generated for each objective, as seen in Figure 8-20 (a), illustrate the controller's ability to adapt to different objectives and velocity references.

The longitudinal velocity error in Figure 8-23 indicates effective velocity tracking across both objectives, with minimal deviations, where all are shown below 1.0 m/s RMS velocity error to the reference velocity, as in Table 8-8. It can be seen that a mellow velocity profile, such as profile 2, tracked better than aggressive velocity profile 1. The distance objective has the highest velocity tracking error

due to the limitation of torque available especially on the front wheel torque. However, it is still a good indication that the controller is able to track the velocity well without too much deviations.

The G-G diagrams, as shown in Figure 8-21, highlight the differences in friction utilization. As expected, the distance objective utilizes a larger portion of the friction circle, indicative of aggressive manoeuvres. In contrast, the comfort objective, it can be seen that its minimised the lateral acceleration to promote the vehicle's comfort ride.

As seen in Figure 8-22, the vehicle control inputs provide additional insights into the controller's performance. The distance objective exhibits aggressive torque input and more abrupt steering angles, while the comfort objectives show smoother torque profiles and reduced steering variations, enhancing ride quality and reducing energy consumption.

Importantly, all solver times were well below the sampling time threshold of 0.075 seconds with a maximum peak time solved of 0.0655 seconds for a minimised distance objective with no solver errors recorded, as shown in Figure 8-24 and Table 8-8.

Regarding comfort, velocity profile 2 with the comfort objective achieves the lowest lateral and longitudinal weighted RMS accelerations, confirming its effectiveness in improving ride comfort, as seen in Table 8-8. As expected, the most uncomfortable case is velocity profile 1 with the distance objective, due to its aggressive acceleration of the generated velocity profile.

In terms of energy saving, energy consumption is obviously less with unaggressive velocity profile 2, as stated in Table 8-8. Furthermore, as expected, the comfort objective achieves the best comfort level with the lowest RMS values for both weighted longitudinal and lateral accelerations with Velocity Profile 2, which is very relaxed in acceleration and deceleration. In comparison, the distance objective, as expected, achieves the fastest lap time due to the velocity profile, which has higher accelerations.

This testing was conducted separately across two distinct velocity profiles to evaluate its performance under specific velocity profile and individual objectives. This approach allowed for a focused evaluation of the controller's performance when prioritizing a single objective under specific driving conditions. In addition, it highlights the trade-offs between performance, energy efficiency, and comfort, offering valuable insights into the PLAPC's adaptability and effectiveness across different use cases.

8.5 Long Path Evaluation with Common Curvature-based Velocity Profile

To further assess the effectiveness of the PLAPC, testing on a longer and more complex path is essential. For this purpose, the Sepang International Circuit, with a total length of 5,534 meters and an average track width of 16 meters, was selected. A detailed plot of the circuit can be found in the Appendix A.2.3.

A common curvature-based velocity profile was generated to evaluate the PLAPC across all objective functions. As shown in Figure 8-25, the velocity profile is designed with an acceleration and deceleration rate of 5 m/s^2 , a maximum velocity of 24 m/s , and a minimum velocity determined by the path curvature, as described in Sub-Chapter 6.3. This acceleration and deceleration rate represents a balanced approach, ensuring suitability for all objective functions while maintaining realistic performance characteristics.

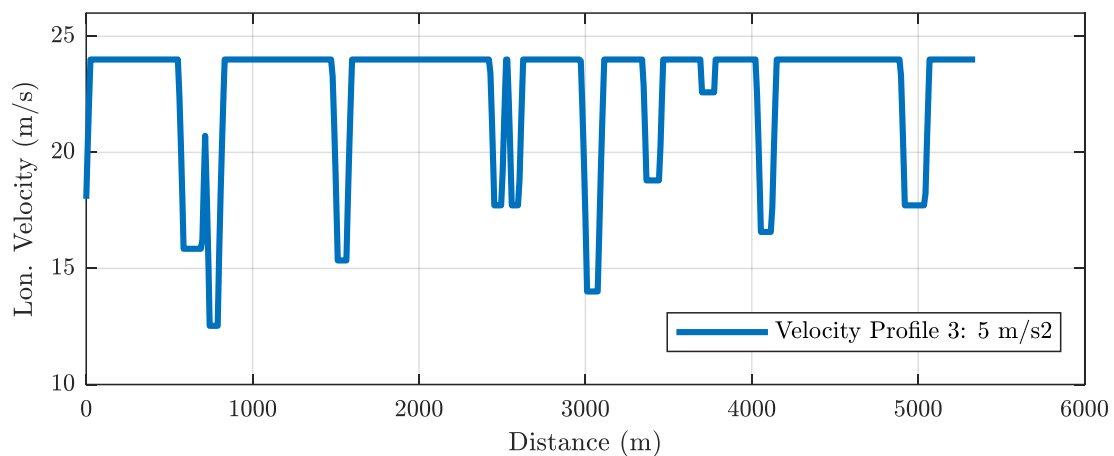
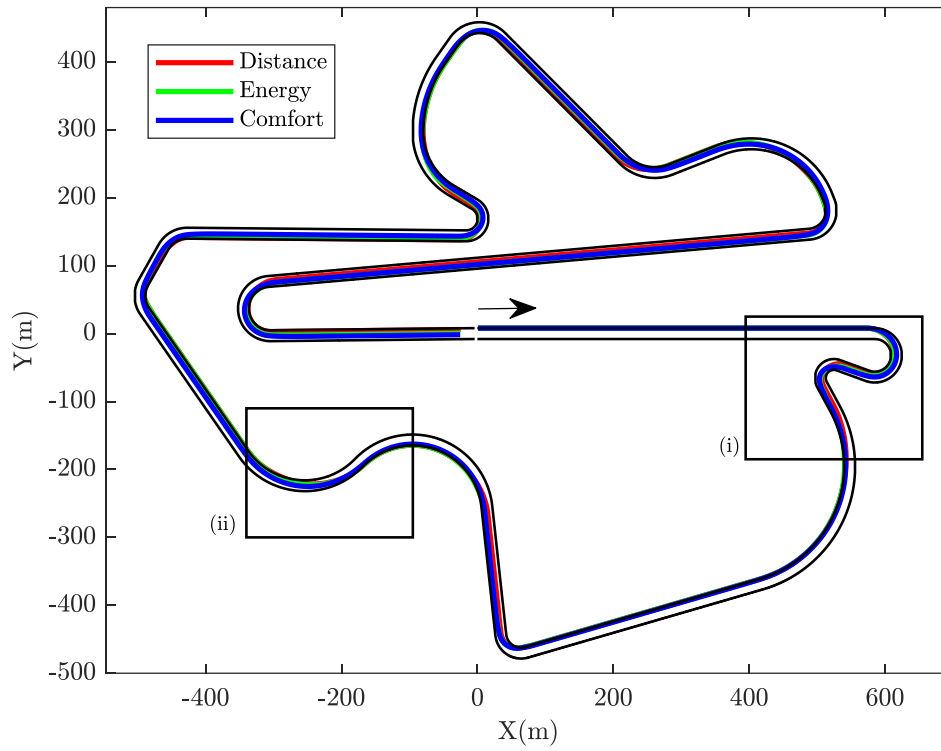
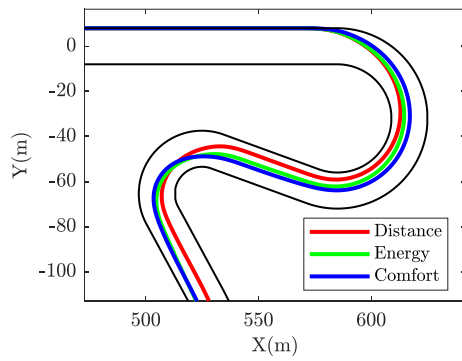


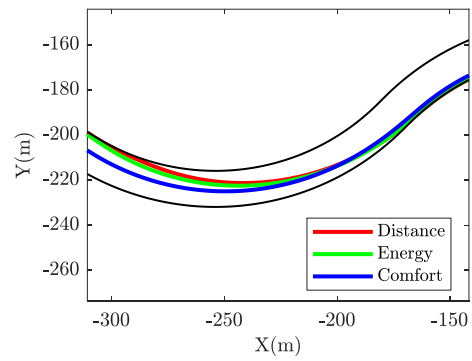
Figure 8-25 Common velocity profiles to test all objective functions



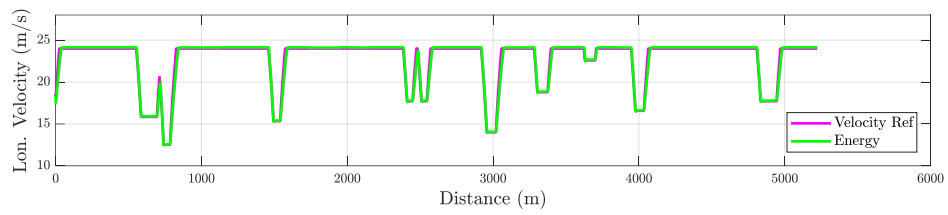
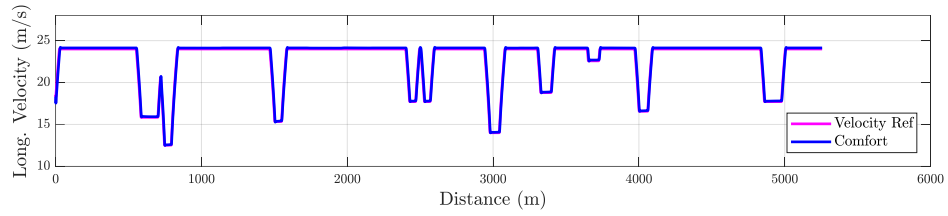
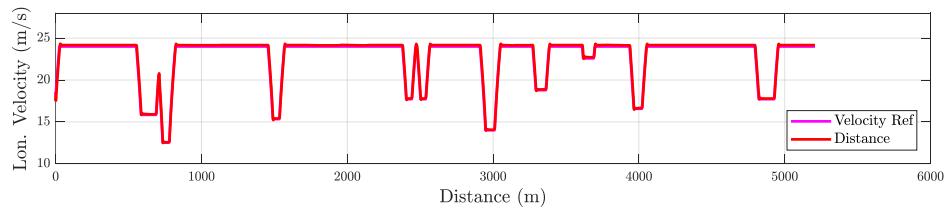
(a) Trajectories



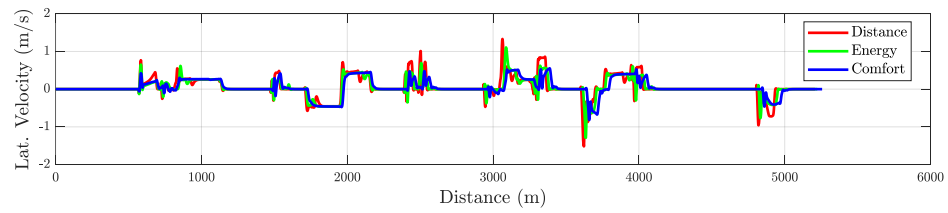
(b) Zoom-in at box (i)



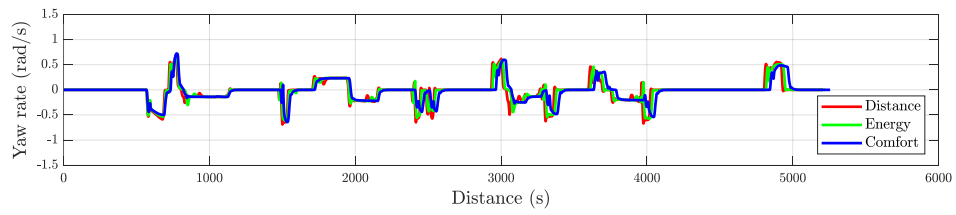
(c) Zoom-in at box (ii)



(d) longitudinal velocity



(e) lateral velocity



(f) Yaw rate

Figure 8-26 Vehicle states comparison of various objective PLAPC result for long-distance evaluations

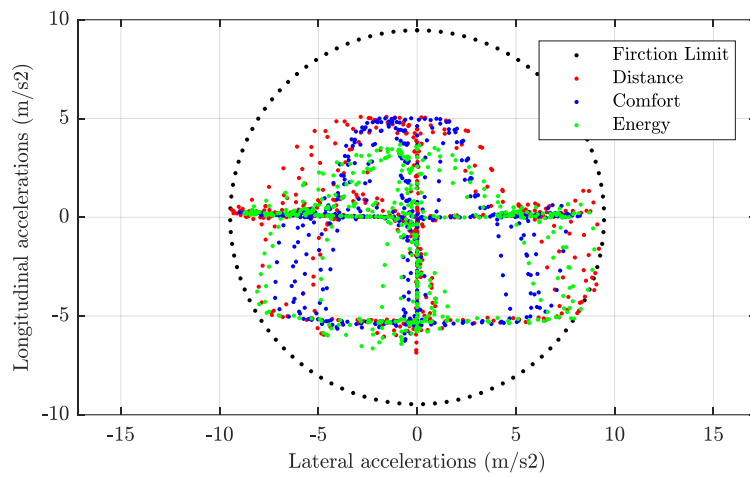
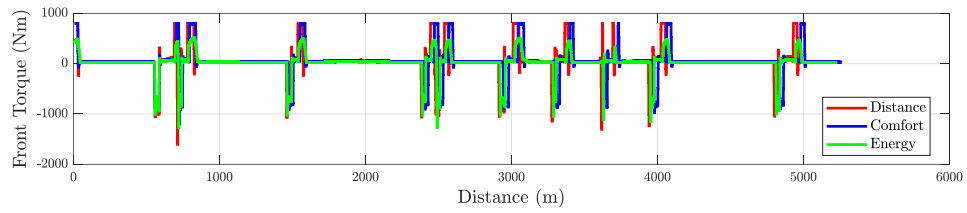
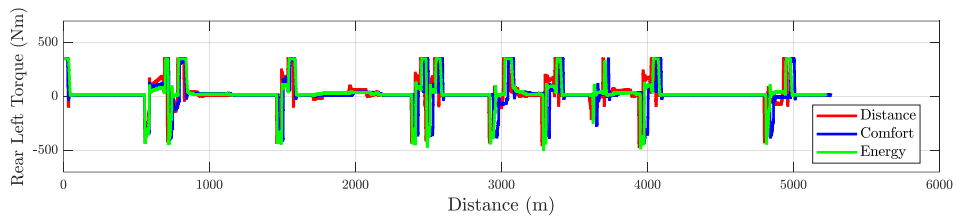


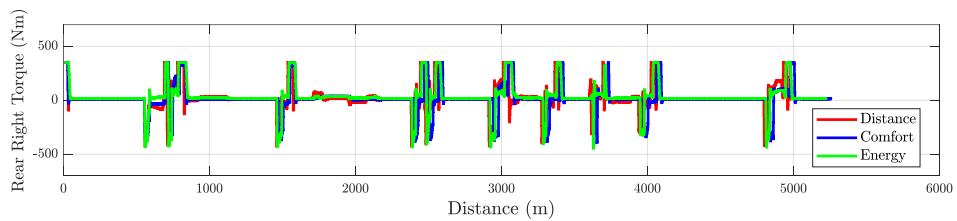
Figure 8-27 G-G diagram comparison of various objective PLAPC result for long-distance evaluations



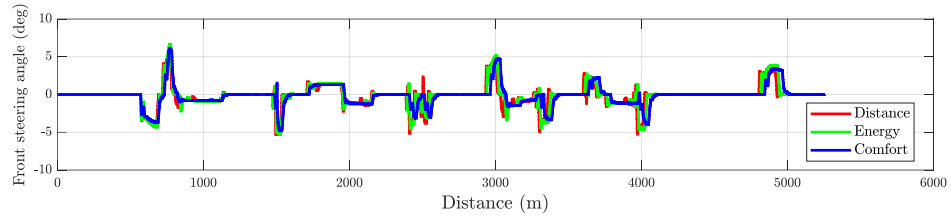
(a) Front wheel torque



(b) Rear left wheel torque



(c) Rear right wheel torque



(d) Front Steering angle

Figure 8-28 Vehicle control inputs comparison of various objective PLAPC result for long-distance evaluations

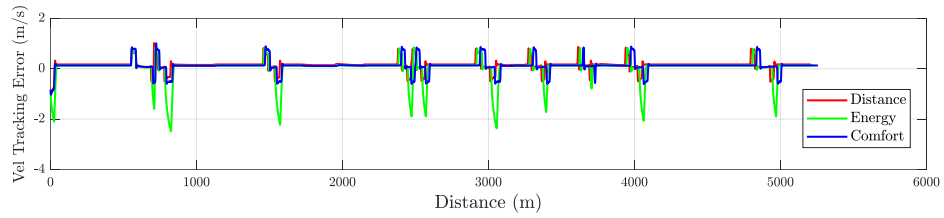
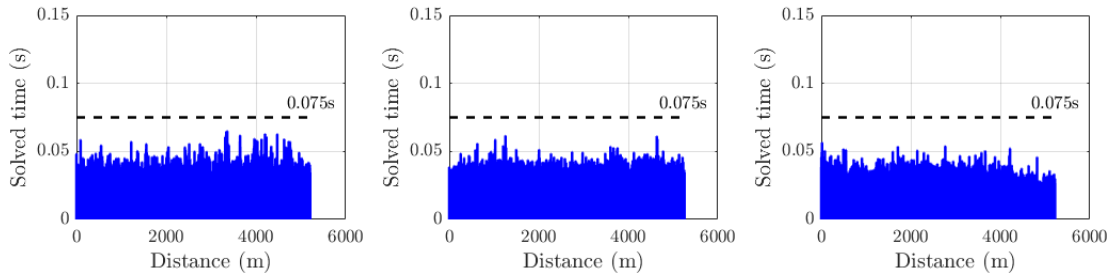


Figure 8-29 Velocity tracking error of various objective PLAPC for long-distance evaluations



(a) Distance

(b) Comfort

(c) Energy

Figure 8-30 IPOPT solver information during the simulations

Table 8-9 Result of multi-objectives simulations for long-distance evaluations

Velocity profile	Common profile at 5 m/s ²		
Objective functions	Distance	Comfort	Energy
Energy consumption (kWh)	0.7549	0.7256	0.7263
Regen. energy generated (kWh)	0.3451	0.3272	0.3296
Net energy consumption (kWh)	0.4098	0.3984	0.3967
RMS error of velocity ref. (m/s)	0.2383	0.2465	0.4516
RMS weighted lon. acc. (m/s ²)	0.9046	0.8335	0.7589
RMS weighted lat. acc. (m/s ²)	0.8814	0.5153	0.7776
Lap time (s)	231.90	235.35	234.30
Average solved time (s)	0.0350	0.0342	0.0324
Peak solved time (s)	0.0650	0.0615	0.0564
No. of solver errors	0	0	0

Note: Green cells indicate the lowest values in the respective rows, and red cells indicate the highest values in the respective rows.

The developed controller is further validated with a longer distance path, and tested with a common velocity profile across all objective functions. The trajectories generated for each objective, as seen in Figure 8-26(a), illustrate the controller's ability to adapt to different objectives. All three objectives generate different trajectories depending on their function. As expected, the Comfort and Energy objective results in a wider trajectory than the Distance objectives, as it aims to reduce the impact of longitudinal and lateral accelerations, enhancing ride comfort and minimising energy consumption.

The longitudinal velocity profiles in Figure 8-29(b) indicate effective velocity tracking across all objectives, with minimal deviations, where all are shown below 0.5 m/s RMS velocity error to the reference velocity, as in Table 8-9. The highest velocity tracking error is from the energy objective function, which due to heavy penalization on the positive torque, thus impacting the velocity tracking,

especially during accelerations and high speed. On the other hand, the less aggressive comfort objective shows better tracking of the reference velocity.

The G-G diagrams, as shown in Figure 8-27, highlight the differences in friction utilization among the objectives, where it can be seen that the distance objective has the most tyre friction utilization. In contrast, the energy objective function minimised the positive torque, which reduced the longitudinal acceleration, and the Comfort objective minimised both impacts on the longitudinal and lateral accelerations for better comfort level.

As seen in Figure 8-28, the vehicle control inputs provide additional insights into the controller's performance. The distance objective exhibits aggressive torque inputs and more abrupt steering angles, while Comfort objectives show smoother torque profiles and reduced steering variations, enhancing ride quality. On the other hand, for the energy objective, heavy penalization on the positive torque reduces energy consumption and allows for more negative torque to promote more energy recovery through regenerative braking.

Importantly, all objectives maintain solver times well below the sampling time threshold of 0.075 seconds with a maximum peak time solved of 0.0650 seconds for minimise distance objective with no solver errors recorded, as shown in Figure 8-30 and Table 8-9. In terms of energy saving, the energy objective reduced energy consumption by 3.3% over the distance objective. Furthermore, the comfort objective achieves the lowest RMS values for weighted lateral accelerations as it prioritizes comfort. However, the energy objective has the lowest RMS value for weighted longitudinal acceleration due to its heavy penalization on the positive torque. The distance objective achieved the fastest lap time, and the comfort objective was the slowest that met their objectives.

In terms of comfort level, the comfort objective achieves the lowest lateral weighted RMS acceleration, as seen in Table 8-9, reducing lateral forces and improving ride quality. Surprisingly, the energy objective results in the lowest longitudinal weighted RMS acceleration, which can be attributed to the heavy penalization on positive torque. This constraint limits aggressive acceleration, naturally smoothing longitudinal forces. While the comfort objective effectively

minimises lateral impact, the energy objective unintentionally reduces longitudinal impact due to torque constraints. However, this outcome is ultimately influenced by the velocity profile itself, which is not explicitly optimised for either comfort or energy efficiency.

As for energy usage, the energy objective that utilized LINEX loss function technique achieved the lowest net energy consumption, as seen in Table 8-9. Surprisingly, the comfort objective is nearly as efficient, showing that prioritizing comfort does not significantly impact energy usage. The distance objective consumes the most energy but also recovers the highest regenerative energy, partially offsetting its consumption. Overall, the energy objective is the most efficient, while the comfort objective remains competitive in energy savings.

In summary, the results showcase the versatility of the developed PLAPC in those objectives. The distance objective prioritizes lap time by minimising the distance line ahead, the energy objective minimises net energy consumption by promoting more energy recovery, and the comfort objective enhances ride comfort by penalising longitudinal and lateral acceleration terms. The controller achieves these goals while maintaining real-time feasibility, highlighting its potential for practical autonomous driving applications.

8.6 Long Path Evaluation with Individual Curvature-based Velocity Profile

In this subchapter, individual velocity profiles assess the controller's ability to track the reference velocity while adhering to specified objectives. These profiles are designed with a maximum velocity of 24 m/s, and the lowest velocity depends on the path curvature, as mentioned in Sub-Chapter 6.3. Two distinct velocity profiles were generated to represent different driving objectives, featuring acceleration and deceleration of 8 m/s^2 (Velocity Profile 1) and 2 m/s^2 (Velocity Profile 2), respectively. The profiles are illustrated in Figure 8-31 below.

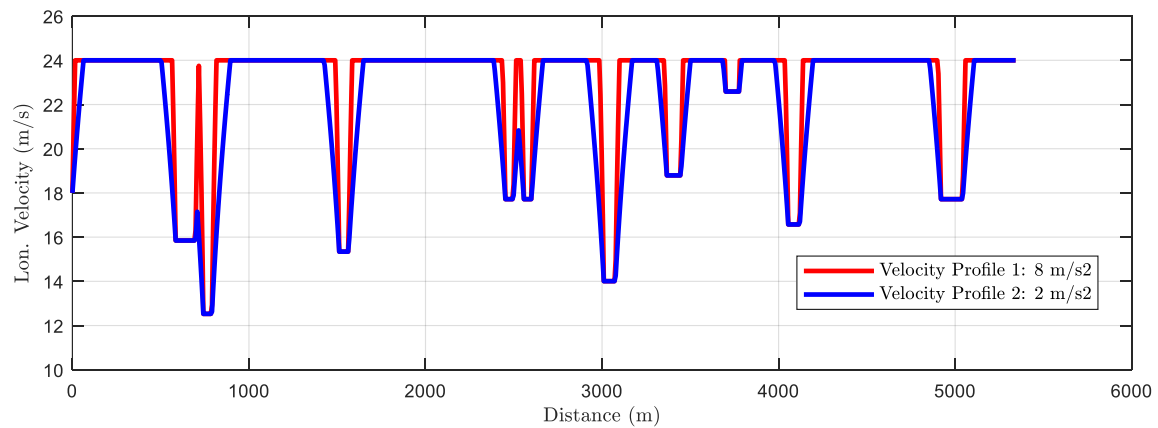
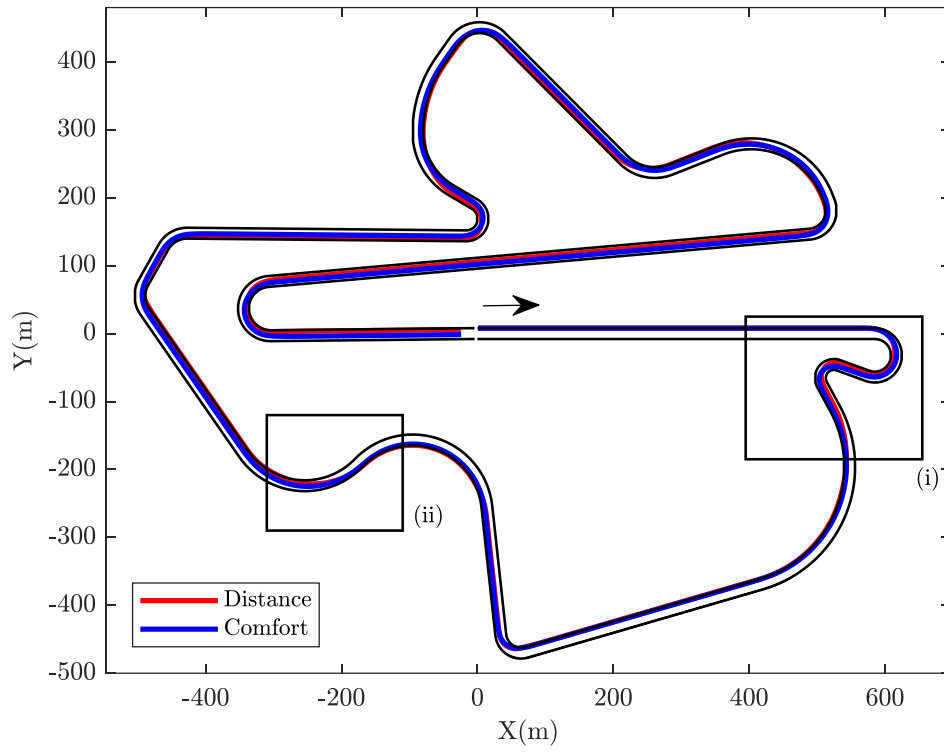
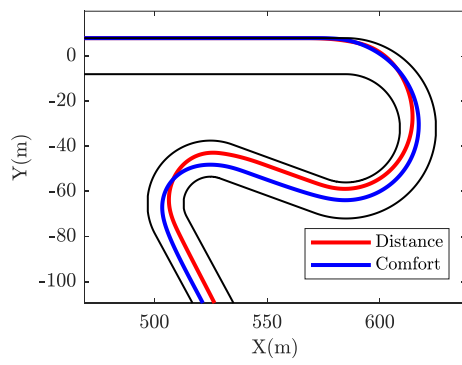


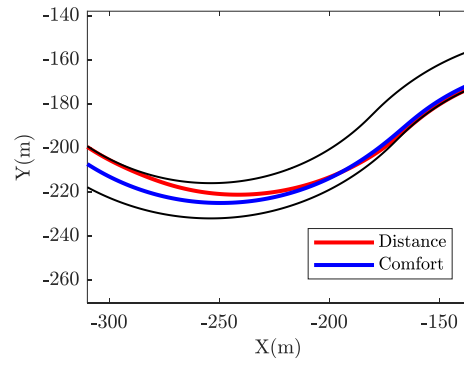
Figure 8-31 Velocity profiles for Sepang International Circuit



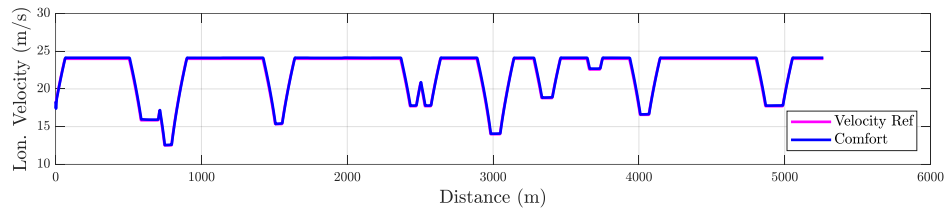
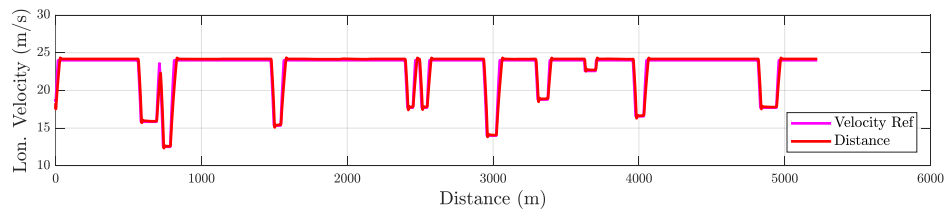
(a) Trajectories



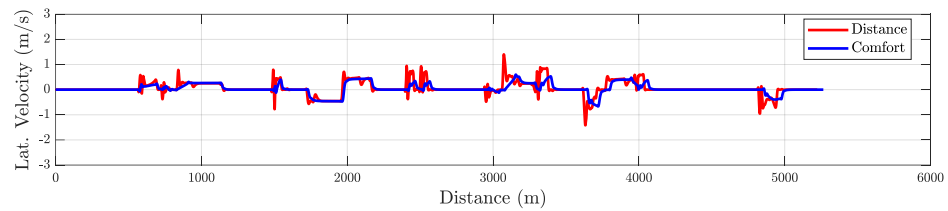
(b) Zoom-in at box (i)



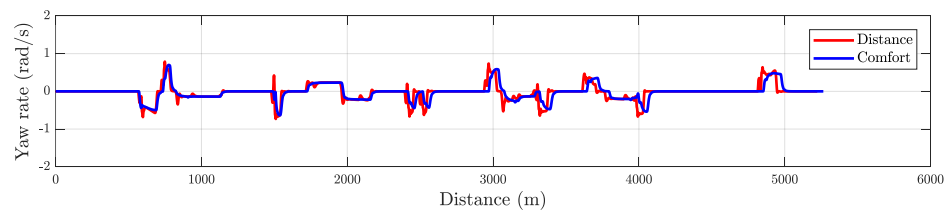
(c) Zoom-in at box (ii)



(d) longitudinal velocity



(e) lateral velocity



(f) Yaw rate

Figure 8-32 Vehicle states comparison of various objective PLAPC results for long distances with distinct velocity profile

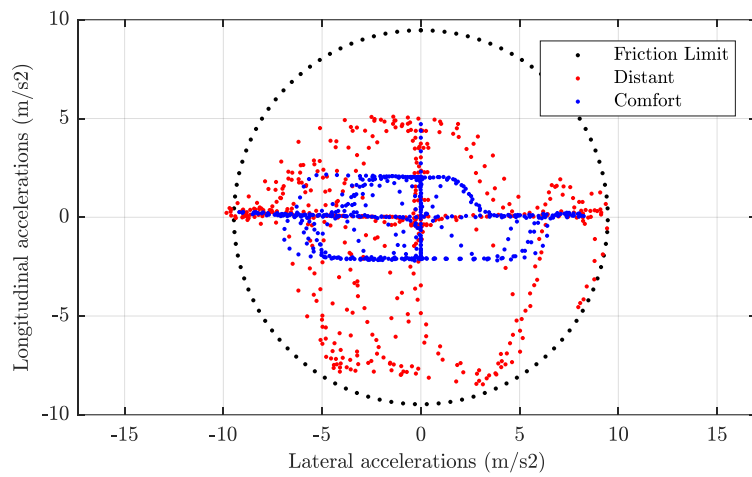
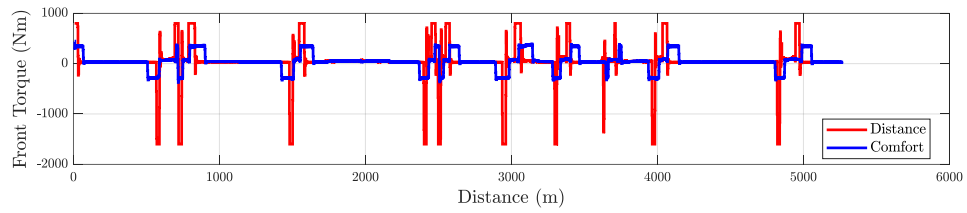
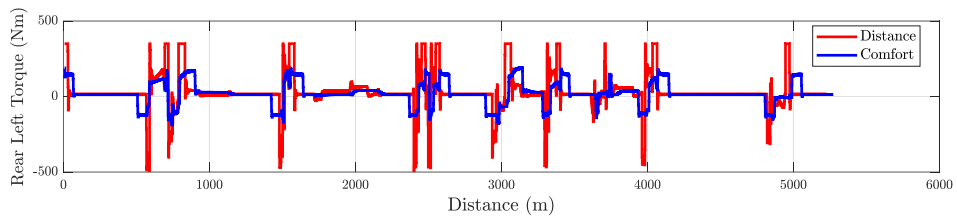


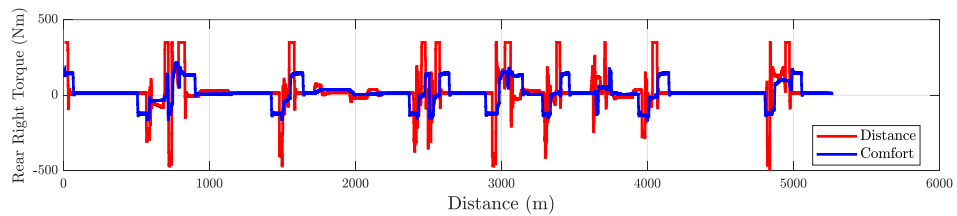
Figure 8-33 G-G diagram comparison of various objective PLAPC results for long distances with distinct velocity profile



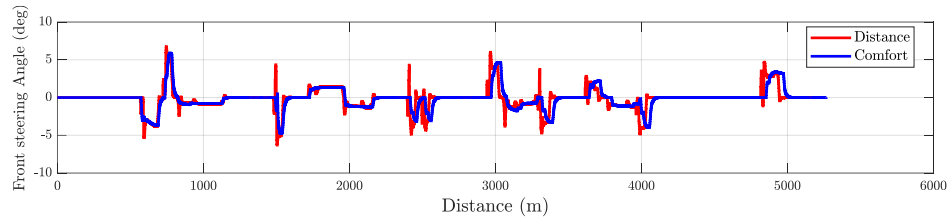
(a) Front wheel torque



(b) Rear left wheel torque



(c) Rear right wheel torque



(d) Front Steering angle

Figure 8-34 Vehicle control inputs comparison PLAPC result for long distance with the distinct velocity profile

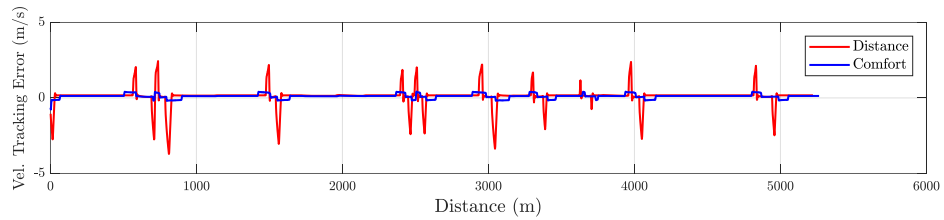
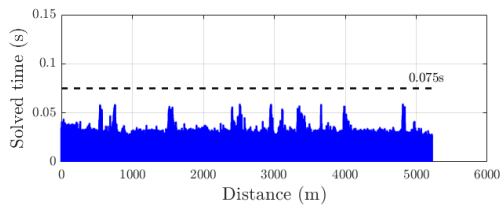
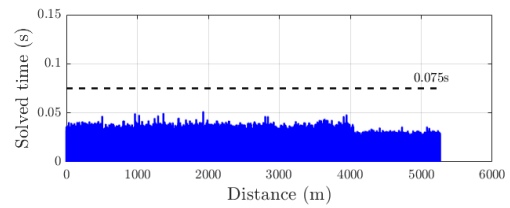


Figure 8-35 Velocity tracking error of various objective PLAPC for long distances with distinct velocity profile



(a) Distance



(b) Comfort

Figure 8-36 IPOPT solver information during the simulations

Table 8-10 Result of various objectives simulations for a long-distance path

Velocity profile	1	2	
Objective Functions	Distance	Comfort	Differences
Energy Consumptions (kWh)	0.7812	0.6659	17.3 %
RMS error of velocity reference (m/s)	0.5766	0.1715	236.1 %
RMS weighted lon. acceleration (m/s^2)	1.1220	0.3776	197.1 %
RMS weighted lat. acceleration (m/s^2)	0.8318	0.4319	92.6 %
Lap time (s)	231.6750	241.7250	- 10.05 s
Average solved time (s)	0.0321	0.0315	-
Peak solved time (s)	0.0590	0.0512	-
No. of solver errors	0	0	-

Note: Green cells indicate the lowest values in the respective rows.

Validation of the developed controller continues in long-distance paths with individual velocity profiles. The trajectories generated for each objective, as seen in Figure 8-32(a), illustrate the controller's ability to adapt to different objectives. Both objectives show different trajectories depending on their objective functions. As expected, the comfort objective resulted in a wider trajectory than the distance objectives, as it aims to reduce the impact of longitudinal and lateral accelerations, thereby enhancing ride comfort. This is also reflected in the magnitude of the comfort level, as shown in Table 8-10, where it falls within the 'A little uncomfortable' classification, as mentioned in ISO 2631-1. On the other hand, the distance objective results is classified as 'Uncomfortable,' as it was tested with higher acceleration levels.

The longitudinal velocity profiles in Figure 8-32(d) indicate effective velocity tracking across all objectives, with minimal deviations, where all are shown below 0.6 m/s RMS velocity error to the reference velocity, as in Table 8-10. The less aggressive, the better tracking to the reference velocity. In addition, the resulting lateral velocity, as in Figure 8-32 (e), further supports that the comfort objective minimises lateral impact, aligning with its goal of reducing ride discomfort.

The G-G diagrams in Figure 8-33 highlight the differences in friction utilization among the objectives. This may not be significant as the velocity profile is not optimised to any objective function. However, it can be seen that the distance objective uses the most tyre friction, while the comfort objective minimises both impacts on the longitudinal and lateral accelerations.

As seen in Figure 8-34, the vehicle control inputs provide additional insights into the controller's performance. The distance objective exhibits aggressive torque input and more abrupt steering angles, while the comfort objectives show smoother torque profiles and reduced steering variations, enhancing ride quality and reducing discomfort.

As seen in previous tests with the multi-corner extended path, velocity profile selection significantly impacts comfort levels, as shown in Table 8-10. The lower acceleration velocity profile (comfort objective) reduces both longitudinal and lateral accelerations, improving ride comfort. However, this also affects energy consumption, leading to a reduction in energy usage but at the cost of a notably slower lap time.

Importantly, all objectives maintain solver times well below the sampling time threshold of 0.075 seconds with a maximum peak time solved of 0.0590 seconds for minimising distance objective with no solver errors recorded, as shown in Figure 8-36 and Table 8-10. Regarding energy saving, the comfort objective gives the lowest energy consumption compared to the distance objective, which significantly differs in accelerations or velocity profile. Furthermore, the comfort objective achieves the lowest RMS values for weighted lateral accelerations as it prioritizes comfort. The distance objective achieved the fastest lap time, and the comfort objective was the slowest, with a significant gap of 10.05 seconds.

In summary, the results showcase the versatility of PLAPC in both contradicting objectives in different velocity profiles. The distance objective prioritizes lap time by minimising the distance line ahead, and the comfort objective enhances ride comfort by penalising longitudinal and lateral accelerations. The controller achieves these goals while maintaining real-time feasibility, highlighting its potential for practical autonomous driving applications.

8.7 Overall Discussion

The proposed PLAPC offers notable advantages, particularly in its ability to generate its own trajectory on the fly. Unlike traditional methods that require specific point coordinates, the PLAPC operates based on a line ahead of the vehicle as a reference. This approach offering a wide visual representation of the driving path while maintaining awareness of lateral boundaries. Additionally, the controller is not dependent on exact coordinates, making it more robust to road conditions, map inaccuracies or borderless or unmarked road.

The evaluation began with a short single-corner path, as described in Sub-Chapter 8.2, where the PLAPC was compared against an existing traditional path-tracking controller using a velocity reference derived from an optimal solution. Both FWS and 4WS vehicle topologies were tested to demonstrate the applicability of the PLAPC. Notably, even with the conventional FWS configuration, without the additional benefit of rear-wheel steering, the PLAPC could closely replicate the optimal solution trajectory and effectively track the velocity profile.

The evaluation continued using the FWS vehicle topology as a baseline, applied to an extended distance of multi-corner path, as detailed in Sub-Chapters 8.3 and 8.4. The key difference between these two evaluations lies in the velocity profiles used. In Sub-Chapter 8.3, the velocity profile was based on 90% of the time-optimal solution, whereas in Sub-Chapter 8.4, the profile was generated based on the path curvature. Different velocity profiles demonstrate two distinct methods applicable to a wide range of autonomous vehicle scenarios: one focused on performance optimisation and the other on practicality while meeting the real-time requirement.

Lastly, to ensure the real-world implementation of the PLAPC, the controller was further tested on the full-length, 5,543-meter Sepang International Circuit as in Sub-Chapters 8.5 and 8.6. The evaluation began with a common velocity profile that covered all objective functions and was followed by tests with individual velocity profiles aimed at optimising specific objectives, either optimising comfort or minimising the distance to the reference line. These tests explored the

controller's ability to handle extreme scenarios on both ends of the performance spectrum.

Despite the advantages outlined earlier, the developed controller does exhibit certain limitations that need to be addressed to enhance its overall capability and applicability in diverse driving scenarios. One of the notable limitations of the controller in general is that the trajectory generated is perpendicular to the line, making it impossible to do curve trajectory in the straight-line path unless changing the reference line length width or angle. However, the obstacle is not considered in this research. Addressing this limitation represents a critical area for improvement and has been proposed for enhancements in Chapter 9.2.

The proposed controller has been designed under the assumption stated in Sub-Chapter 4.1, which considers a continuous path and without static or dynamic obstacles. It is designed for structured environments with well-defined road boundaries, ensuring operation within clear constraints. However, in complex environments such as T-junctions, intersections, or shared spaces with multiple road users traveling in different directions, additional considerations and further study are required.

An effective perception system is essential such as integrating data from LiDAR, cameras, or V2X communication to detect and calculate the distance to dynamically adapt based on the changing environment. In scenarios like intersections or T-junctions with multiple vehicles at a time, a systematic coordinated approach can be implemented in this work, as studied in [116], where an MPC-based coordination strategy prioritizes collision-free passage and minimal travel time across the intersection. A centralized coordinator via V2X manages vehicle movements within specified zones and orders vehicles to move to the highest allowed velocity, ensuring smooth acceleration and passenger comfort by enforcing acceleration constraints. This study also includes time-based priorities to prevent conflicts, minimise stops and optimise traffic flow. A more advanced approach is proposed in [117], utilizing reinforcement learning, modelled as a Partially Observable Markov Decision Process. The framework leverages deep Q-learning to learn optimal policies by predicting future

trajectories and risk levels of surrounding agents, including vehicles, pedestrians, and cyclists. A graph-based representation is used, where nodes represent traffic participants, and edges encode interaction strengths. The model is tested more challenging environment with non-signalized T-junctions and lane merge intersections, handling complex multi-agent interactions while optimising for safety, comfort (low jerkiness), and energy efficiency. However, further study is required to assess its applicability to PLAPC, as this aspect is not covered in this work.

In shared spaces, where road users might not follow strict lane boundaries, dynamic object detection and avoidance become critical. The controller should not only ensure lane discipline where applicable but also actively minimise the vehicle's exposure to dangerous areas, such as the middle of an intersection, by minimising the time spent in high-risk zones. For object and obstacle avoidance, an adaptive reference line width is proposed to enable safe manoeuvres around obstacles while maintaining feasible trajectories.

Future work outlined in Sub-Chapter 9.2 addresses these challenges by further exploring the controller's adaptability in complex scenarios, ensuring safe and efficient navigation for autonomous vehicles.

8.8 Summary

This chapter concludes the development and evaluation of the simultaneous planning and tracking controller named PLAPC. The study compared path-tracking performance comprehensively, showcasing the PLAPC's capability to generate precise trajectory adherence closer to optimal solution trajectory. Then, the PLAPC was tested in scenarios involving optimal solutions requiring aggressive vehicle manoeuvres, highlighting its ability to handle challenging driving. Additional tests included simulations on extended distances of multi-corner paths to evaluate the controller's adaptability in directional changes. Finally, the PLAPC was validated on a long-distance, realistic path, demonstrating its applicability over further extended distances.

Moreover, the controller was assessed in multi-objective scenarios, where it was successfully able to manoeuvre with good velocity tracking and stay within road limits for safety. These tests emphasized its versatility and ability to adapt to various driving situations, from smooth cruising to challenging manoeuvres in constrained environments.

In summary, this chapter illustrates the comprehensive development and testing of the developed PLAPC, showcasing its capability to operate effectively across various driving scenarios. The findings serve as a solid foundation for the development of a controller that benefits autonomous vehicles.

9 Conclusions

This thesis presents the development of a novel controller under a simultaneous planning and tracking framework, named Perpendicular Line Ahead Planner Controller (PLAPC), which introduces an innovative approach of utilizing line ahead reference system and an automatic curvature-based velocity profile. This controller provides a complete solution that eliminates reliance on pre-defined optimised trajectories and addresses the limitations of hierarchical methods. This framework offers flexibility, enabling its implementation across various vehicle configurations. It is capable of satisfying diverse objectives without requiring adjustments to the underlying strategy. This ability to seamlessly adapt to varying scenarios makes the proposed framework highly robust, safe and practical for real-world deployment. The following subchapter highlights the contribution made in this work.

9.1 Summary of Contributions

1) Development of novel controller, PLAPC

The PLAPC replaces traditional point-based references with a line-ahead reference system, removing the need for pre-computed trajectory references via a simultaneous planning and tracking framework.

2) Multi-Objective Optimisation

The framework integrates multi-objective functions to address diverse and conflicting goals, such as related to time efficiency, optimised comfort and energy efficiency. This allows the controller to perform effectively under varying operational requirements.

3) Introduction of the LINEX Penalty Term for Energy Efficient

A novel LINEX (Linear-Exponential) penalty term for torque control inputs was introduced to promote energy recovery and improve energy efficiency.

Compared to traditional quadratic penalties, this approach simplifies the controller design while enhancing regenerative braking capabilities.

4) Applicability to Baseline Configurations

The PLAPC has been successfully implemented and validated with a baseline Front-Wheel Steering (FWS) configuration, demonstrating its versatility and capability even in conventional setups without advanced steering systems such as 4WS.

5) Real-Time Validation

High-fidelity simulations in IPG CarMaker validated the real-time implementability of the PLAPC, ensuring the controller operates within specified computational limits and is suitable for diverse scenarios and objectives.

This thesis advances autonomous vehicle motion planning and control by introducing novel methodologies and validated strategies that address key gaps in the field. The proposed PLAPC framework eliminates reliance on pre-defined trajectories, incorporates multi-objective optimisation, and ensures real-time adaptability, paving the way for practical, efficient, and safe autonomous driving systems. These contributions provide a foundation for further research and development in motion planning and control for autonomous vehicles.

9.2 Recommended for Future Work

The work presented in this thesis demonstrates a strong foundation of the developed controller; however, there are several avenues for further improvement and exploration. Key recommendations for future work include,

- 1) **Real-Time Implementation on a Scaled or Full-Scale Prototype Vehicle.**
While the controller has been validated through simulation using industry-proven high-fidelity software, IPG CarMaker, the next step involves implementing it on a real-time system. Testing on a scaled prototype or a full-scale vehicle would allow for further validation of its practical applicability and performance in real-world conditions. This would help identify potential issues related to hardware constraints, communication delays, and sensor integration that may not have been evident in the simulation.
- 2) **Testing in Diverse Driving Environments.** Expanding the testing scenarios to include varying weather conditions (such as rain, snow), road surfaces (such as gravel, wet asphalt), gradient (such as uphill, downhill) and uneven road boundary would provide a more comprehensive evaluation of the controller's performance and the impact on the generated trajectory.
- 3) **Develop further strategies for obstacle avoidance.** This could involve augmenting the perpendicular line reference approach with additional considerations, such as predicted obstacle trajectories with adaptive line reference width. By tackling these limitations, future iterations of the controller can achieve greater versatility, enabling it to perform effectively in a broader range of scenarios, including those with static and dynamic obstacles or varied path width geometries. Such improvements would further enhance the controller's applicability to real-world autonomous driving challenges.
- 4) **Further develop strategies for applying the controller in unstructured environments,** such as roads with poor or missing boundaries, as well as complex road layouts like intersections and T-junctions. This would be particularly beneficial, as the controller is designed for autonomous vehicle applications.

- 5) Removing dependency on velocity references. Another potential improvement for the developed controller involves eliminating its dependency on predefined velocity references. Currently, the controller relies on velocity references to guide longitudinal motion, constraining its adaptability in dynamic environments where velocity profiles may need to be adjusted in real-time. Future work could focus on enabling the controller to autonomously determine optimal velocity profiles based on real-time environmental conditions, road geometry, and traffic scenarios.

The developed controller can be further refined and validated by addressing these recommendations, paving the way for broader applicability in real-world autonomous driving systems. These future developments would enhance its robustness, scalability, and adaptability, ultimately contributing to safer and more efficient autonomous vehicle solutions.

References

- [1] "Connected and automated mobility 2025: realising the benefits of self-driving vehicles - GOV.UK." Accessed: Jan. 11, 2025. [Online]. Available: https://www.gov.uk/government/publications/connected-and-automated-mobility-2025-realising-the-benefits-of-self-driving-vehicles?utm_source=chatgpt.com
- [2] "JAGUAR LAND ROVER CREATES FURTHER THREE GLOBAL TECH HUBS TO DEVELOP AUTONOMOUS CARS | JLR Media Newsroom." Accessed: Nov. 18, 2024. [Online]. Available: <https://media.jaguarlandrover.com/news/2023/02/jaguar-land-rover-creates-further-three-global-tech-hubs-develop-autonomous-cars>
- [3] "evolvAD: Nissan-backed research project to bring autonomous mobility to UK residential and rural roads." Accessed: Nov. 18, 2024. [Online]. Available: <https://europe.nissannews.com/en-GB/releases/evolvad-nissan-backed-research-project-to-bring-autonomous-mobility-to-uk-residential-and-rural-roads>
- [4] "UK on the cusp of a transport revolution, as self-driving vehicles set to be worth nearly £42 billion by 2035 - GOV.UK." Accessed: Nov. 18, 2024. [Online]. Available: <https://www.gov.uk/government/news/uk-on-the-cusp-of-a-transport-revolution-as-self-driving-vehicles-set-to-be-worth-nearly-42-billion-by-2035>
- [5] Y. Zhang *et al.*, "Mitigation of Motion Sickness and Optimization of Motion Comfort in Autonomous Vehicles: Systematic Survey," *IEEE Transactions on Intelligent Transportation Systems*, 2024, doi: 10.1109/TITS.2024.3462495.
- [6] L. Tate *et al.*, "Energy Efficiency of Autonomous Car Powertrain," *SAE Technical Papers*, vol. 2018-April, Apr. 2018, doi: 10.4271/2018-01-1092.
- [7] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A Review of Motion Planning for Highway Autonomous Driving," *IEEE Transactions on*

- Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1826–1848, May 2020, doi: 10.1109/TITS.2019.2913998.
- [8] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, “A survey of motion planning and control techniques for self-driving urban vehicles,” *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016, doi: 10.1109/TIV.2016.2578706.
 - [9] M. D. Tezerjani, D. Carrillo, D. Qu, S. Dhakal, A. Mirzaeinia, and Q. Yang, “Real-time Motion Planning for autonomous vehicles in dynamic environments,” Jun. 2024, [Online]. Available: <http://arxiv.org/abs/2406.02916>
 - [10] X. Wang, J. Liu, H. Peng, X. Qie, X. Zhao, and C. Lu, “A Simultaneous Planning and Control Method Integrating APF and MPC to Solve Autonomous Navigation for USVs in Unknown Environments,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 105, no. 2, pp. 1–16, Jun. 2022, doi: 10.1007/S10846-022-01663-8/METRICS.
 - [11] S. Zhang, H. Liang, P. Zhao, Z. Li, and T. Su, “Reference Path Generation for Autonomous Driving in Off-road Environment,” *ACM International Conference Proceeding Series*, pp. 19–25, Mar. 2022, doi: 10.1145/3529261.3529265.
 - [12] S. Li, Z. Li, Z. Yu, B. Zhang, and N. Zhang, “Dynamic Trajectory Planning and Tracking for Autonomous Vehicle with Obstacle Avoidance Based on Model Predictive Control,” *IEEE Access*, vol. 7, pp. 132074–132086, 2019, doi: 10.1109/ACCESS.2019.2940758.
 - [13] Z. Wang, S. Wang, S. Zheng, S. Q. Xie, Y. Xie, and H. Wu, “Simultaneous Planning and Tracking Framework for Obstacle Avoidance of Autonomous Mobile Robots in Dynamic Scenarios,” *IEEE Transactions on Industrial Electronics*, 2025, doi: 10.1109/TIE.2024.3522489.
 - [14] Z. Gao, W. Wen, Y. Xing, and A. Tsourdos, “An Integrated Framework for Autonomous Driving Planning and Tracking based on NNMPCC Considering

- Road Surface Variations,” *IEEE Transactions on Intelligent Vehicles*, 2024, doi: 10.1109/TIV.2024.3418951.
- [15] J. Shi, Y. Su, C. Piao, Y. Tang, Y. Liang, and Z. Wang, “Integrated automatic parking path planning and trajectory tracking optimization method,” *Transactions of the Institute of Measurement and Control*, Oct. 2024, doi: 10.1177/01423312241282832/ASSET/IMAGES/LARGE/10.1177_01423312241282832-FIG16.JPEG.
- [16] C. Tu, Z. Li, B. Leng, and L. Xiong, “A Seamless Motion Planning Integrating Maneuver Decision Based on Hybrid Model Predictive Control,” *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 3228–3234, 2023, doi: 10.1109/ITSC57777.2023.10422155.
- [17] “Vehicle salon inside car driver view with rudder Vector Image.” Accessed: Jan. 25, 2025. [Online]. Available: <https://www.vectorstock.com/royalty-free-vector/vehicle-salon-inside-car-driver-view-with-rudder-vector-20685098>
- [18] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, “Path planning and trajectory planning algorithms: A general overview,” *Mechanisms and Machine Science*, vol. 29, pp. 3–27, Jan. 2015, doi: 10.1007/978-3-319-14705-5_1/FIGURES/9.
- [19] P. Stano *et al.*, “Model predictive path tracking control for automated road vehicles: A review,” *Annu Rev Control*, vol. 55, pp. 194–236, Jan. 2023, doi: 10.1016/J.ARCONTROL.2022.11.001.
- [20] C. Zhou, B. Huang, and P. Fränti, “A review of motion planning algorithms for intelligent robotics,” no. DI, pp. 1–38, Feb. 2021, Accessed: May 10, 2021. [Online]. Available: <http://arxiv.org/abs/2102.02376>
- [21] M. D. Tezerjani, D. Carrillo, D. Qu, S. Dhakal, A. Mirzaeinia, and Q. Yang, “Real-time Motion Planning for autonomous vehicles in dynamic environments”.

- [22] “ISO 2631-1:1997 - Mechanical vibration and shock — Evaluation of human exposure to whole-body vibration — Part 1: General requirements.” Accessed: Jan. 20, 2025. [Online]. Available: <https://www.iso.org/standard/7612.html>
- [23] M. Rickert, A. Sieverling, and O. Brock, “Balancing exploration and exploitation in sampling-based motion planning,” *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1305–1317, Dec. 2014, doi: 10.1109/TRO.2014.2340191.
- [24] L. Kong, Z. Wang, R. Xi, L. Shen, Z. Ji, and K. Li, “A review of intelligent vehicle motion planning algorithms,” <https://doi.org/10.1117/12.3005968>, vol. 12799, pp. 23–27, Oct. 2023, doi: 10.1117/12.3005968.
- [25] W. Xu, J. Wei, J. M. Dolan, H. Zhao, and H. Zha, “A real-time motion planner with trajectory optimization for autonomous vehicles,” *Proc IEEE Int Conf Robot Autom*, pp. 2061–2067, 2012, doi: 10.1109/ICRA.2012.6225063.
- [26] M. G. Plessen, P. F. Lima, J. Martensson, A. Bemporad, and B. Wahlberg, “Trajectory planning under vehicle dimension constraints using sequential linear programming,” *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 2018-March, pp. 1–6, Mar. 2018, doi: 10.1109/ITSC.2017.8317665.
- [27] C. G. Atkeson and C. Liu, “Trajectory-Based Dynamic Programming,” *Cognitive Systems Monographs*, vol. 18, pp. 1–15, 2013, doi: 10.1007/978-3-642-36368-9_1.
- [28] S. Poudel, M. Y. Arafat, and S. Moh, “Bio-Inspired Optimization-Based Path Planning Algorithms in Unmanned Aerial Vehicles: A Survey,” *Sensors* 2023, Vol. 23, Page 3051, vol. 23, no. 6, p. 3051, Mar. 2023, doi: 10.3390/S23063051.
- [29] J. Li, S. X. Yang, and Z. Xu, “A survey on robot path planning using bio-inspired algorithms,” *IEEE International Conference on Robotics and*

- Biomimetics*, *ROBIO* 2019, pp. 2111–2116, Dec. 2019, doi: 10.1109/ROBIO49542.2019.8961498.
- [30] F. Aljalaud, H. Kurdi, and K. Youcef-Toumi, “Bio-Inspired Multi-UAV Path Planning Heuristics: A Review,” *Mathematics* 2023, Vol. 11, Page 2356, vol. 11, no. 10, p. 2356, May 2023, doi: 10.3390/MATH11102356.
- [31] fetoh hason, “A review: On bio-inspired optimization methods for path planning of mobile robot,” *Iraqi Journal of Intelligent Computing and Informatics (IJICI)*, Accessed: Mar. 06, 2025. [Online]. Available: https://www.academia.edu/105786762/A_review_On_bio_inspired_optimization_methods_for_path_planning_of_mobile_robot
- [32] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger, “TransFuser: Imitation With Transformer-Based Sensor Fusion for Autonomous Driving,” *IEEE Trans Pattern Anal Mach Intell*, vol. 45, no. 11, pp. 12878–12895, Nov. 2023, doi: 10.1109/TPAMI.2022.3200245.
- [33] P. Wu, L. Chen, H. Li, X. Jia, J. Yan, and Y. Qiao, “Policy Pre-training for Autonomous Driving via Self-supervised Geometric Modeling,” *11th International Conference on Learning Representations, ICLR 2023*, Jan. 2023, Accessed: Mar. 19, 2025. [Online]. Available: <https://arxiv.org/abs/2301.01006v2>
- [34] Y. Xiao, F. Codevilla, D. Porres, and A. M. Lopez, “Scaling Vision-based End-to-End Driving with Multi-View Attention Learning,” Feb. 2023, Accessed: Mar. 19, 2025. [Online]. Available: <https://arxiv.org/abs/2302.03198v3>
- [35] H. M. Eraqi, M. N. Moustafa, and J. Honer, “Dynamic Conditional Imitation Learning for Autonomous Driving,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 22988–23001, Nov. 2022, doi: 10.1109/TITS.2022.3214079.
- [36] T. D. Do, M. T. Duong, Q. V. Dang, and M. H. Le, “Real-Time Self-Driving Car Navigation Using Deep Neural Network,” *Proceedings 2018 4th*

- International Conference on Green Technology and Sustainable Development, GTSD 2018*, pp. 7–12, Dec. 2018, doi: 10.1109/GTSD.2018.8595590.
- [37] M. Toromanoff, E. Wirbel, and F. Moutarde, “End-to-End Model-Free Reinforcement Learning for Urban Driving using Implicit Affordances,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 7151–7160, Nov. 2019, doi: 10.1109/CVPR42600.2020.00718.
- [38] A. Kendall *et al.*, “Learning to Drive in a Day,” *Proc IEEE Int Conf Robot Autom*, vol. 2019-May, pp. 8248–8254, Jul. 2018, doi: 10.1109/ICRA.2019.8793742.
- [39] B. Osinski *et al.*, “Simulation-based reinforcement learning for real-world autonomous driving,” *Proc IEEE Int Conf Robot Autom*, pp. 6411–6418, Nov. 2019, doi: 10.1109/ICRA40945.2020.9196730.
- [40] D. Chen, V. Koltun, and P. Krähenbühl, “Learning to drive from a world on rails,” *Proceedings of the IEEE International Conference on Computer Vision*, pp. 15570–15579, May 2021, doi: 10.1109/ICCV48922.2021.01530.
- [41] T. Bian, Y. Xing, and A. Zolotas, “End-to-End One-Shot Path-Planning Algorithm for an Autonomous Vehicle Based on a Convolutional Neural Network Considering Traversability Cost,” *Sensors 2022, Vol. 22, Page 9682*, vol. 22, no. 24, p. 9682, Dec. 2022, doi: 10.3390/S22249682.
- [42] W. Zhang *et al.*, “DynGMP: Graph Neural Network-Based Motion Planning in Unpredictable Dynamic Environments,” *IEEE International Conference on Intelligent Robots and Systems*, pp. 858–865, 2023, doi: 10.1109/IROS55552.2023.10342326.
- [43] “Tesla’s robotaxi push hinges on ‘black box’ AI gamble | Reuters.” Accessed: Jan. 13, 2025. [Online]. Available:

<https://www.reuters.com/technology/tesla-gambles-black-box-ai-tech-robotaxis-2024-10-10/>

- [44] W. Zeng *et al.*, “End-to-end Interpretable Neural Motion Planner,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 8652–8661, Jan. 2021, doi: 10.1109/CVPR.2019.00886.
- [45] A. Kendall *et al.*, “Learning to Drive in a Day,” *Proc IEEE Int Conf Robot Autom*, vol. 2019-May, pp. 8248–8254, Jul. 2018, doi: 10.1109/ICRA.2019.8793742.
- [46] J. Kwon, A. Khalil, D. Kim, and H. Nam, “Incremental End-to-End Learning for Lateral Control in Autonomous Driving,” *IEEE Access*, vol. 10, pp. 33771–33786, 2022, doi: 10.1109/ACCESS.2022.3160655.
- [47] X. Yang and H. Li, “Model Predictive Motion Planning for Autonomous Vehicle in Mid-high Overtaking Scene,” in *IEEE Vehicular Technology Conference*, Institute of Electrical and Electronics Engineers Inc., May 2020. doi: 10.1109/VTC2020-Spring48590.2020.9128897.
- [48] V. Cardoso *et al.*, “A Model-Predictive Motion Planner for the IARA autonomous car,” in *Proceedings - IEEE International Conference on Robotics and Automation*, Institute of Electrical and Electronics Engineers Inc., Jul. 2017, pp. 225–230. doi: 10.1109/ICRA.2017.7989028.
- [49] L. Grüne and J. Pannek, “Nonlinear Model Predictive Control,” *Communications and Control Engineering*, no. 9780857295002, pp. 43–66, 2011, doi: 10.1007/978-0-85729-501-9_3/FIGURES/1.
- [50] M. Walid, M. Elrafi, M. Said, and S. John’, “Optimization Based Solutions for Control and State Estimation in Non-holonomic Mobile Robots: Stability, Distributed Control, and Relative Localization,” Mar. 2018, doi: 10.13140/rg.2.2.22309.60644.
- [51] X. Chen, G. Wu, M. Ren, X. Chen, G. Wu, and M. Ren, “Nonlinear Model Predictive Control of Autonomous Vehicles Considering Dynamic Stability

- Constraints,” *SAE Int J Adv Curr Pract Mobil*, vol. 2, no. 5, pp. 2974–2986, Apr. 2020, doi: 10.4271/2020-01-1400.
- [52] S. Luciani, A. Bonfitto, N. Amati, and A. Tonoli, “Model predictive control for comfort optimization in assisted and driverless vehicles,” *Advances in Mechanical Engineering*, vol. 12, no. 11, 2020, doi: 10.1177/1687814020974532.
- [53] J. L. Vazquez, M. Bruhlmeier, A. Liniger, A. Rupenyan, and J. Lygeros, “Optimization-Based Hierarchical Motion Planning for Autonomous Racing,” *IEEE/RJS International Conference on Intelligent RObots and Systems*, pp. 2397–2403, Oct. 2020, doi: 10.1109/IROS45743.2020.9341731.
- [54] H. Guo, C. Shen, H. Zhang, H. Chen, and R. Jia, “Simultaneous Trajectory Planning and Tracking Using an MPC Method for Cyber-Physical Systems: A Case Study of Obstacle Avoidance for an Intelligent Vehicle,” *IEEE Trans Industr Inform*, vol. 14, no. 9, pp. 4273–4283, Sep. 2018, doi: 10.1109/TII.2018.2815531.
- [55] Y. Gao, X. Wang, J. Huang, and L. Yuan, “Adaptive Model Predictive Control for Intelligent Vehicle Trajectory Tracking Considering Road Curvature,” *International Journal of Automotive Technology*, vol. 25, no. 5, pp. 1051–1064, Oct. 2024, doi: 10.1007/S12239-024-00086-8/FIGURES/9.
- [56] U. Rosolia and F. Borrelli, “Learning How to Autonomously Race a Car: A Predictive Control Approach,” *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2713–2719, Nov. 2020, doi: 10.1109/TCST.2019.2948135.
- [57] H. Xue, E. L. Zhu, J. M. Dolan, and F. Borrelli, “Learning Model Predictive Control with Error Dynamics Regression for Autonomous Racing,” *Proc IEEE Int Conf Robot Autom*, pp. 13250–13256, Sep. 2023, doi: 10.1109/icra57147.2024.10611628.

- [58] Z. Chen, J. Lai, P. Li, O. I. Awad, and Y. Zhu, "Prediction Horizon-Varying Model Predictive Control (MPC) for Autonomous Vehicle Control," *Electronics* 2024, Vol. 13, Page 1442, vol. 13, no. 8, p. 1442, Apr. 2024, doi: 10.3390/ELECTRONICS13081442.
- [59] M. Farrokhsiar, "MPC- based mobile robots motion planning and control in uncertain dynamic environment," 2015, doi: 10.14288/1.0074418.
- [60] K. Yang, S. K. Gan, and S. Sukkarieh, "An efficient path planning and control algorithm for RUAV's in unknown and cluttered environments," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 57, no. 1–4, pp. 101–122, Jan. 2010, doi: 10.1007/S10846-009-9359-1/METRICS.
- [61] T. Novi, A. Liniger, R. Capitani, and C. Annicchiarico, "Real-time control for at-limit handling driving on a predefined path," *Vehicle System Dynamics*, vol. 58, no. 7, pp. 1007–1036, Jul. 2020, doi: 10.1080/00423114.2019.1605081.
- [62] D. Dang, F. Gao, and Q. Hu, "Motion Planning for Autonomous Vehicles Considering Longitudinal and Lateral Dynamics Coupling," *Applied Sciences* 2020, Vol. 10, Page 3180, vol. 10, no. 9, p. 3180, May 2020, doi: 10.3390/APP10093180.
- [63] X. Qian, I. Navarro, A. De La Fortelle, and F. Moutarde, "Motion planning for urban autonomous driving using bezier curves and MPC," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 826–833, Dec. 2016, doi: 10.1109/ITSC.2016.7795651.
- [64] S. Nekkah *et al.*, "The Autonomous Racing Software Stack of the KIT19d," *SAE International Journal of Connected and Automated Vehicles*, vol. 5, no. 1, Oct. 2020, doi: 10.4271/12-05-01-0007.
- [65] X. Xing, B. Zhao, C. Han, D. Ren, and H. Xia, "Vehicle Motion Planning With Joint Cartesian-Frenét MPC," *IEEE Robot Autom Lett*, vol. 7, no. 4, pp. 10738–10745, Oct. 2022, doi: 10.1109/LRA.2022.3194330.

- [66] J. T. Li, C. K. Chen, and H. Ren, "Time-Optimal Trajectory Planning and Tracking for Autonomous Vehicles," *Sensors* 2024, Vol. 24, Page 3281, vol. 24, no. 11, p. 3281, May 2024, doi: 10.3390/S24113281.
- [67] C. Wang, X. Chen, C. Li, R. Song, Y. Li, and M. Q. H. Meng, "Chase and Track: Toward Safe and Smooth Trajectory Planning for Robotic Navigation in Dynamic Environments," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 1, pp. 604–613, Jan. 2023, doi: 10.1109/TIE.2022.3148753.
- [68] Y. Zhang, K. Liu, F. Gao, and F. Zhao, "Research on Path Planning and Path Tracking Control of Autonomous Vehicles Based on Improved APF and SMC," *Sensors*, vol. 23, no. 18, Sep. 2023, doi: 10.3390/S23187918.
- [69] J. J. Park, C. Johnson, and B. Kuipers, "Robot navigation with model predictive equilibrium point control," *IEEE International Conference on Intelligent Robots and Systems*, pp. 4945–4952, 2012, doi: 10.1109/IROS.2012.6386195.
- [70] C. Dai, C. Zong, and G. Chen, "Path tracking control based on model predictive control with adaptive preview characteristics and speed-assisted constraint," *IEEE Access*, vol. 8, pp. 184697–184709, 2020, doi: 10.1109/ACCESS.2020.3029635.
- [71] Z. Cheng, X. Zeng, H. Fang, G. Wang, and L. Dou, "Hierarchical MPC-based Motion Planning for Automated Vehicles in Parallel Autonomy," <https://doi.org/10.1142/S2301385024500286>, vol. 12, no. 5, pp. 927–938, May 2023, doi: 10.1142/S2301385024500286.
- [72] L. Zhang *et al.*, "A Novel Simultaneous Planning and Control Scheme of Automated Lane Change on Slippery Roads," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 20696–20706, Nov. 2022, doi: 10.1109/TITS.2022.3186429.
- [73] V. A. Laurence and J. Christian Gerdes, "Long-Horizon Vehicle Motion Planning and Control through Serially Cascaded Model Complexity," *IEEE*

Transactions on Control Systems Technology, vol. 30, no. 1, pp. 166–179, Jan. 2022, doi: 10.1109/TCST.2021.3056315.

- [74] A. Liniger, A. Domahidi, and M. Morari, “Optimization-based autonomous racing of 1:43 scale RC cars,” *Optim Control Appl Methods*, vol. 36, no. 5, pp. 628–647, Sep. 2015, doi: 10.1002/oca.2123.
- [75] G. Perantoni and D. J. N. Limebeer, “Optimal control for a Formula One car with variable parameters,” *Vehicle System Dynamics*, vol. 52, no. 5, pp. 653–678, May 2014, doi: 10.1080/00423114.2014.889315.
- [76] X. Li, Z. Sun, Q. Zhu, and D. Liu, “A unified approach to local trajectory planning and control for autonomous driving along a reference path,” *2014 IEEE International Conference on Mechatronics and Automation, IEEE ICMA 2014*, pp. 1716–1721, 2014, doi: 10.1109/ICMA.2014.6885959.
- [77] Y. Han, X. Ma, B. Wang, H. Zhang, Q. Zhang, and G. Chen, “A Strategy for Integrated Multi-Demands High-Performance Motion Planning Based on Nonlinear MPC,” *Applied Sciences 2023, Vol. 13, Page 12443*, vol. 13, no. 22, p. 12443, Nov. 2023, doi: 10.3390/APP132212443.
- [78] D. Wang, W. Fu, J. Zhou, and Q. Song, “Occlusion-Aware Motion Planning for Autonomous Driving,” *IEEE Access*, vol. 11, pp. 42809–42823, 2023, doi: 10.1109/ACCESS.2023.3268072.
- [79] C. Wang, X. Zhao, R. Fu, and Z. Li, “Research on the Comfort of Vehicle Passengers Considering the Vehicle Motion State and Passenger Physiological Characteristics: Improving the Passenger Comfort of Autonomous Vehicles,” *International Journal of Environmental Research and Public Health 2020, Vol. 17, Page 6821*, vol. 17, no. 18, p. 6821, Sep. 2020, doi: 10.3390/IJERPH17186821.
- [80] S. Bae, Y. Choi, Y. Kim, J. Guanetti, F. Borrelli, and S. Moura, “Real-time Ecological Velocity Planning for Plug-in Hybrid Vehicles with Partial Communication to Traffic Lights,” *IEEE Conference on Decision and*

- Control*, vol. 2019-December, pp. 1279–1285, Dec. 2019, doi: 10.1109/CDC40024.2019.9030166.
- [81] A. Meshginqalam and J. Bauman, “Two-level mpc speed profile optimization of autonomous electric vehicles considering detailed internal and external losses,” *IEEE Access*, vol. 8, pp. 206559–206570, 2020, doi: 10.1109/ACCESS.2020.3038050.
 - [82] Y. Ma, Q. Zhang, D. Li, and Y. Tian, “LINEX Support Vector Machine for Large-Scale Classification,” *IEEE Access*, vol. 7, pp. 70319–70331, 2019, doi: 10.1109/ACCESS.2019.2919185.
 - [83] H. R. Varian, “A Bayesian approach to real estate assessment,” 1975, *Amsterdam*.
 - [84] A. Zellner, “Bayesian estimation and prediction using asymmetric loss functions,” *J Am Stat Assoc*, vol. 81, no. 394, pp. 446–451, Jun. 1986, doi: 10.2307/2289234.
 - [85] Y. C. Chang and W. L. Hung, “LINEX loss functions with applications to determining the optimum process parameters,” *Qual Quant*, vol. 41, no. 2, pp. 291–301, Apr. 2007, doi: 10.1007/S11135-005-5425-3/METRICS.
 - [86] S. Zhang, Y. Luo, K. Li, and V. Li, “Real-Time Energy-Efficient Control for Fully Electric Vehicles Based on an Explicit Model Predictive Control Method,” *IEEE Trans Veh Technol*, vol. 67, no. 6, pp. 4693–4701, Jun. 2018, doi: 10.1109/TVT.2018.2806400.
 - [87] H. Banvait, S. Anwar, and Y. Chen, “A rule-based energy management strategy for plugin hybrid electric vehicle (PHEV),” *Proceedings of the American Control Conference*, pp. 3938–3943, 2009, doi: 10.1109/ACC.2009.5160242.
 - [88] C. Lin, E. Siampis, and E. Velenis, “Real-Time Path-tracking MPC for an Over-Actuated Autonomous Electric Vehicle,” *Proceedings of the American Control Conference*, vol. 2022-June, pp. 2006–2011, 2022, doi: 10.23919/ACC53348.2022.9867858.

- [89] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi: a software framework for nonlinear optimization and optimal control,” *Math Program Comput*, vol. 11, no. 1, pp. 1–36, Mar. 2019, doi: 10.1007/S12532-018-0139-4/TABLES/3.
- [90] R. Verschueren, Gianluca *et al.*, “acados: a modular open-source framework for fast embedded optimal control,” *Math Program Comput*, vol. 14, no. 1, pp. 147–183, Oct. 2019, doi: 10.1007/s12532-021-00208-8.
- [91] “acados — acados documentation.” Accessed: Mar. 08, 2025. [Online]. Available: <https://docs.acados.org/>
- [92] “Overview - embotech.com.” Accessed: Mar. 08, 2025. [Online]. Available: <https://www.embotech.com/softwareproducts/forcespro/overview/>
- [93] “4. Backward Compatibility — FORCESPRO 6.3.0 documentation.” Accessed: Mar. 20, 2025. [Online]. Available: https://forces.embotech.com/documentation/back_compatibility/index.html
- [94] F. Aubeck, V. Kumar, N. Murgovski, and S. Pischinger, “Performance Comparison of Real-Time Solver Implementations for Powertrain Nonlinear Energy Management Optimization with MPC,” *European Control Conference 2020, ECC 2020*, pp. 483–490, May 2020, doi: 10.23919/ECC51009.2020.9143843.
- [95] “Introduction To Optimization and Optimal Control Using Casadi | PDF | Optimal Control | Nonlinear Programming.” Accessed: Jan. 17, 2025. [Online]. Available: <https://www.scribd.com/document/361299741/Introduction-to-Optimization-and-Optimal-Control-using-Casadi>
- [96] A. Rodríguez Alcázar, “Path planning for autonomous buses based on optimal control,” 2018, Accessed: Jan. 17, 2025. [Online]. Available: <https://upcommons.upc.edu/handle/2117/167488>
- [97] “CasADi - Docs.” Accessed: Jan. 17, 2025. [Online]. Available: <https://web.casadi.org/docs/>

- [98] "Ipopt: Documentation." Accessed: Jan. 17, 2025. [Online]. Available: <https://coin-or.github.io/Ipopt/>
- [99] J. Subosits and J. C. Gerdes, "Impacts of Model Fidelity on Trajectory Optimization for Autonomous Vehicles in Extreme Maneuvers," *IEEE Transactions on Intelligent Vehicles*, 2021, doi: 10.1109/TIV.2021.3051325.
- [100] R. Lot and N. Dal Bianco, "The significance of high-order dynamics in lap time simulations," *The Dynamics of Vehicles on Roads and Tracks - Proceedings of the 24th Symposium of the International Association for Vehicle System Dynamics, IAVSD 2015*, pp. 553–562, 2016, doi: 10.1201/B21185-60/UNIFICATION-ACCELERATION-ENVELOPE-DRIVEABILITY-CONCEPTS-SALISBURY-LIMEBEER-TREMLET-MASSARO.
- [101] Reza N. Jazar, "Vehicle Dynamics: Theory and Application."
- [102] E. Bakker, L. Nyborg, and H. B. Pacejka, "Tyre Modelling for Use in Vehicle Dynamics Studies," *SAE Technical Papers*, Feb. 1987, doi: 10.4271/870421.
- [103] U. Kiencke and L. Nielsen, "Automotive control systems: For engine, driveline, and vehicle: Second edition," *Automotive Control Systems: For Engine, Driveline, and Vehicle: Second Edition*, pp. 1–512, 2005, doi: 10.1007/B137654/COVER.
- [104] R. Wang and J. Wang, "Tire–road friction coefficient and tire cornering stiffness estimation based on longitudinal tire force difference generation," *Control Eng Pract*, vol. 21, no. 1, pp. 65–75, Jan. 2013, doi: 10.1016/J.CONENGPRAC.2012.09.009.
- [105] N. Ding and S. Taheri, "A Modified Dugoff Tire Model for Combined-slip Forces," *Tire Sci Technol*, vol. 38, no. 3, pp. 228–244, Sep. 2010, doi: 10.2346/1.3481696.

- [106] A. Bhoraskar and P. Sakthivel, "A review and a comparison of Dugoff and modified Dugoff formula with Magic formula," *2017 International Conference on Nascent Technologies in Engineering, ICNTE 2017 - Proceedings*, Jun. 2017, doi: 10.1109/ICNTE.2017.7947898.
- [107] M. Bian *et al.*, "A Dynamic Model for Tire/Road Friction Estimation under Combined Longitudinal/Lateral Slip Situation," *SAE Technical Papers*, vol. 1, Apr. 2014, doi: 10.4271/2014-01-0123.
- [108] R. Lot and F. Biral, "A Curvilinear Abscissa Approach for the Lap Time Optimization of Racing Vehicles," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 7559–7565, Jan. 2014, doi: 10.3182/20140824-6-ZA-1003.00868.
- [109] F. Christ, A. Wischnewski, A. Heilmeier, and B. Lohmann, "Time-optimal trajectory planning for a race car considering variable tyre-road friction coefficients," *Vehicle System Dynamics*, vol. 59, no. 4, pp. 588–612, 2021, doi: 10.1080/00423114.2019.1704804.
- [110] A. Jiménez Elbal, A. Zarzuelo Conde, and E. Siampis, "Simultaneous Optimisation of Vehicle Design and Control for Improving Vehicle Performance and Energy Efficiency Using an Open Source Minimum Lap Time Simulation Framework," *World Electric Vehicle Journal 2024, Vol. 15, Page 366*, vol. 15, no. 8, p. 366, Aug. 2024, doi: 10.3390/WEVJ15080366.
- [111] J. T. Betts, "Practical Methods for Optimal Control Using Nonlinear Programming, Third Edition," *Practical Methods for Optimal Control Using Nonlinear Programming, Third Edition*, Jan. 2020, doi: 10.1137/1.9781611976199.
- [112] "CarMaker | IPG Automotive." Accessed: Jan. 25, 2025. [Online]. Available: <https://www.ipg-automotive.com/en/products-solutions/software/carmaker/>
- [113] C. Lin, E. Siampis, and E. Velenis, "Path-tracking control at the limits of handling of a prototype over-actuated autonomous vehicle," *Vehicle System Dynamics*, Jun. 2024, doi: 10.1080/00423114.2024.2361726.

- [114] S. C. Walpole, D. Prieto-Merino, P. Edwards, J. Cleland, G. Stevens, and I. Roberts, "The weight of nations: An estimation of adult human biomass," *BMC Public Health*, vol. 12, no. 1, pp. 1–6, Jun. 2012, doi: 10.1186/1471-2458-12-439/TABLES/4.
- [115] C. Lin, "Path tracking control of a multi-actuated autonomous vehicle at the limits of handling.," 2021, *Cranfield University*. Accessed: Oct. 08, 2024. [Online]. Available: <https://dspace.lib.cranfield.ac.uk/handle/1826/21136>
- [116] A. Mihály, Z. Farkas, B. Zsuzsanna, and P. Gáspár, "Performance Analysis of Model Predictive Intersection Control for Autonomous Vehicles," *IFAC-PapersOnLine*, vol. 54, no. 2, pp. 240–245, Jan. 2021, doi: 10.1016/J.IFACOL.2021.06.029.
- [117] E. Zhang, R. Zhang, and N. Masoud, "Predictive trajectory planning for autonomous vehicles at intersections using reinforcement learning," *Transp Res Part C Emerg Technol*, vol. 149, p. 104063, Apr. 2023, doi: 10.1016/J.TRC.2023.104063.

Appendices

A The Parameters

A.1 Vehicle Parameters

This work specifies a list of vehicle parameters for optimal solution vehicle model, PLAPC and IPG CarMaker Simulation setup, as shown in the table below.

Table A-1 Vehicle Parameters

Parameters	Unit	Value
Vehicle mass, m	kg	874.5
Vehicle inertia, I_z	kgm ²	1597.7
Wheelbase length, l	m	1.995
Front to COG, l_f	m	0.815
Rear to COG, l_r	m	1.180
Left track width, w_l	m	0.765
Right track width, w_r	m	0.765
Total track width, w	m	1.530
Height to CoG, h	m	0.297
Wheel radius, R_w	m	0.315
Front steering angle limit, δ_f	deg	± 40
Torque front limit, T_f	Nm	800
Torque rear left limit, T_{rl}	Nm	350
Torque rear right limit, T_{rr}	Nm	350

A.1.1 Tire Parameters

A list of tyre parameters is specified in this work for the controller and IPG CarMaker Simulation setup, as shown in the table below.

Table A-2 Tyre Parameters

Parameters	Unit	Value
Magic Formula stiffness factor, B	-	9.500
Magic Formula shape factor, C	-	1.626
Magic Formula peak factor, D	-	1.166

A.2 Reference Path Parameters

A.2.1 Short Distance: 90-Degree Path

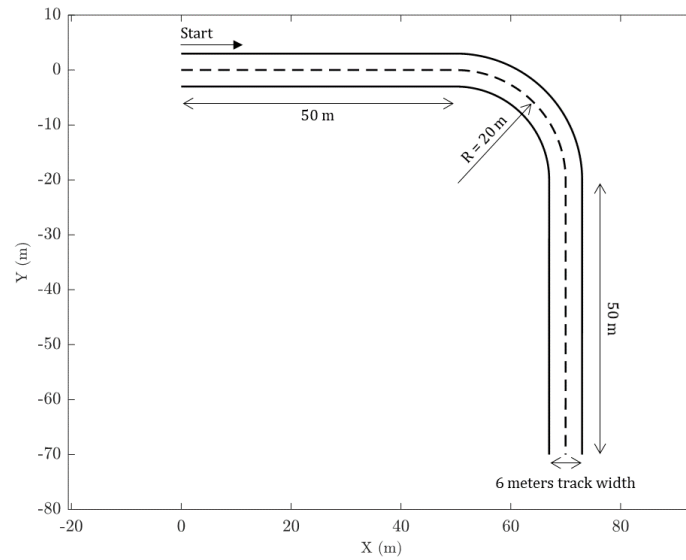


Figure A-1 90-degree right-hand corner path

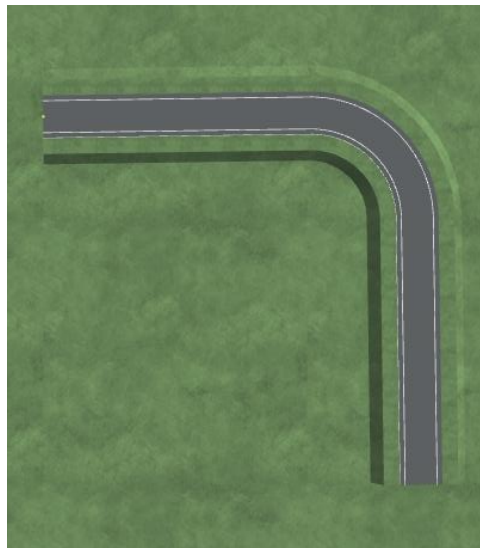


Figure A-2 90-degree right-hand corner path in IPG CarMaker

A.2.2 Short Distance: Multi-Corner Path

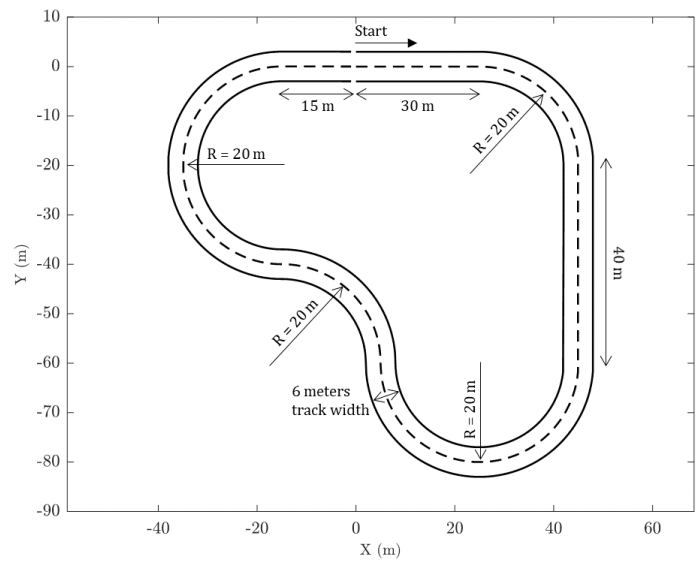


Figure A-3 Multi-corner extended path

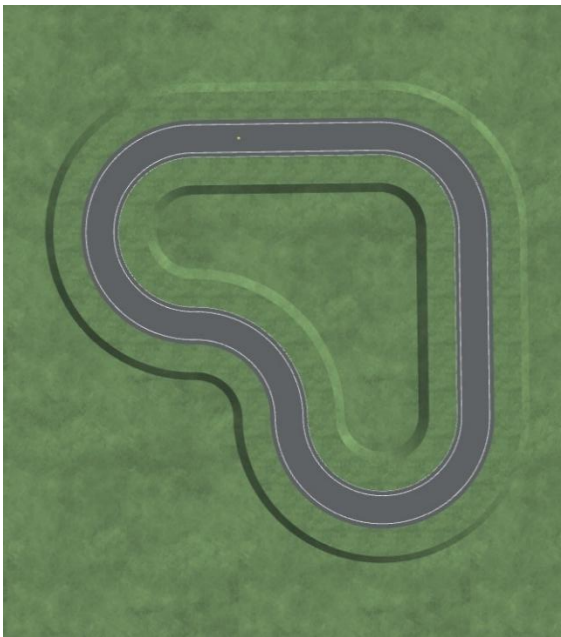


Figure A-4 Multi-corner extended path in IPG CarMaker

A.2.3 Long Distance: Sepang International Circuit, Malaysia

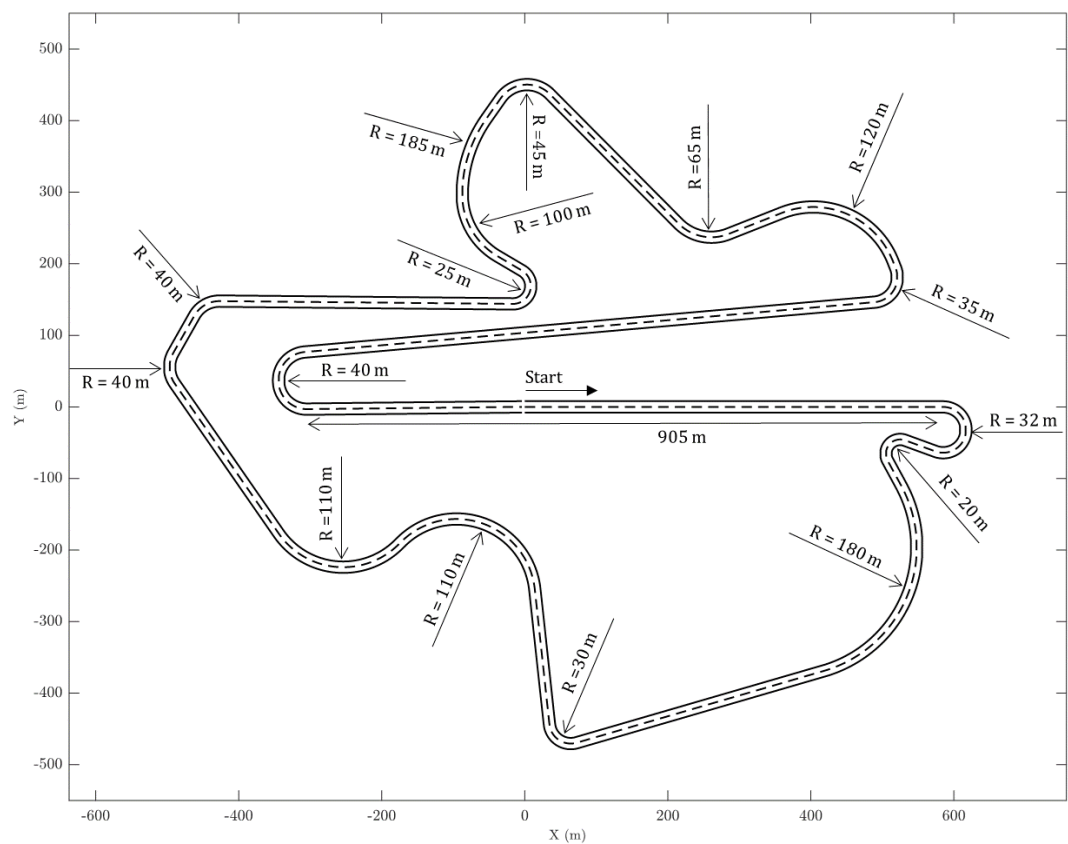


Figure A-3 Long-distance path of Sepang International Circuit

A.3 Optimal Solution Parameters

Table A-3 Initial and final condition parameters for optimal control problem

Initial conditions parameters	Final condition parameters
$V_x^o = 20;$	$V_x^f = open;$
$V_y^o = 0.001;$	$V_y^f = 0.001;$
$\dot{\psi}^o = 0;$	$\dot{\psi}^f = 0;$
$S_n^o = open;$	$S_n^f = open;$
$xi = 0;$	$xi^f = open;$
$\omega_{ij}^o = open;$	$\omega_{ij}^f = open;$
$\psi^o = 0;$	$\psi^f = track\ end\ yaw\ angle;$
$T_f^o = open;$	$T_f^f = open;$
$T_{rl}^o = open;$	$T_{rl}^f = open;$
$T_{rr}^o = open;$	$T_{rr}^f = open;$
$\delta_f^o = 0;$	$\delta_f^f = 0;$
$\delta_r^o = 0;$	$\delta_r^f = 0;$

A.3.1 Minimum Time Objective

Regularisation factor for result in Chapter 5.3.3,

$$Q_u = diag ([2, 2,5e - 10, 5 - 11,5e - 11]);$$

$$Q_{du} = diag ([5,5, 1e - 7, 5e - 9, 5e - 9]);$$

A.3.2 Optimized Comfort Objective

Regularisation factor for Equation (5-19),

$$Q_u = \text{diag} ([2, 2,5e - 10, 5 - 11,5e - 11]);$$

$$Q_{du} = \text{diag} ([5,5, 1e - 7, 5e - 9, 5e - 9]);$$

Weighting components for Equation (5-17),

$$Q_t = 20; Q_{cax} = 0.05; Q_{cay} = 1; Q_{cdax} = 15; Q_{cday} = 10;$$

A.4 High Fidelity Simulation PLAPC Parameters

Table A-4 Initial and final condition parameters for PLAPC simulations set in the IPG CarMaker

Initial conditions parameters	Final condition parameters
$X^o = 2.99$; (for short and multi-corner extended path) $X^o = 7.99$; (for long path) $Y^o = 0$; $\psi^o = 0$; $V_x^o = 18$; $V_y^o = 0$; $\dot{\psi}^o = 0$; $T_f^o = open$; $T_{rl}^o = open$; $T_{rr}^o = open$; $\delta_f^o = open$; $\delta_r^o = open$; (for 4WS only)	$X^f = open$; $Y^f = open$; $\psi^f = open$; $V_x^f = open$; $V_y^f = open$; $\dot{\psi}^f = open$; $T_f^f = open$; $T_{rl}^f = open$; $T_{rr}^f = open$; $\delta_f^f = open$; $\delta_r^f = open$; (for 4WS only)

A.4.1 Single Corner Test

PTC parameters in Chapter 0,

$$Q_{PTC} = diag ([10, 10, 1313.1, 1, 1, 32.8])$$

$$R_{PTC} = diag ([2872.5, 2872.5, 1.36e - 4, 7.14e - 4, 7.14e - 4])$$

PLAPC-4WS parameters for minimum distance in Chapter 0

$$Q_{line} = 75; Q_{lim} = 40; Q_{vel} = 1800;$$

$$D_{line} = 5.2 \text{ meters}$$

Regularisation factor,

$$Q_u = \text{diag} ([0.00023, 0.00023, 0.00009, 13680, 26680]);$$

$$Q_{du} = \text{diag} ([1e - 5, 1e - 5, 1e - 5, 1e - 5, 1e - 5]);$$

PLAPC-FWS parameters for minimum distance in Chapter 0

$$Q_{line} = 75; Q_{lim} = 40; Q_{vel} = 1800;$$

$$D_{line} = 7.8 \text{ meters}$$

Regularisation factor,

$$Q_u = \text{diag} ([0.00023, 0.00023, 0.0012, 3600]);$$

$$Q_{du} = \text{diag} ([1e - 5, 1e - 5, 1e - 5, 1e - 5]);$$

PLAPC-4WS parameters for comfort optimized in Chapter 0

$$Q_{line} = 75; Q_{lim} = 40; Q_{vel} = 1800;$$

$$D_{line} = 2.6 \text{ meters}$$

Comfort weighting components,

$$Q_{cax} = 0.01; Q_{cay} = 0.4; Q_{cdax} = 0.04; Q_{cday} = 1;$$

Regularisation factor,

$$Q_u = \text{diag} ([7.6e - 5, 7.6e - 5, 3.8e - 5, 1680, 3360]);$$

$$Q_{du} = \text{diag} ([1e - 5, 1e - 5, 1e - 5, 1e - 5, 1e - 5]);$$

PLAPC-FWS parameters for comfort optimized in Chapter 0

$$Q_{line} = 75; Q_{lim} = 40; Q_{vel} = 1800;$$

$$D_{line} = 2.6 \text{ meters}$$

Comfort weighting components,

$$Q_{cax} = 0.01; Q_{cay} = 0.4; Q_{cdax} = 0.04; Q_{cday} = 1;$$

Regularisation factor,

$$Q_u = \text{diag} ([7.6e - 5, 7.6e - 5, 3.8e - 5, 1260]);$$

$$Q_{du} = \text{diag} ([1e - 5, 1e - 5, 1e - 5, 1e - 5]);$$

PLAPC-4WS parameters for minimum distance in Chapter 0 without tire load-dependent coefficient

$$Q_{line} = 75; Q_{lim} = 40; Q_{vel} = 1800;$$

$$D_{line} = 5.2 \text{ meters}$$

Regularisation factor,

$$Q_u = \text{diag} ([0.00023, 0.00023, 0.00009, 11520, 23920]);$$

$$Q_{du} = \text{diag} ([1e - 5, 1e - 5, 1e - 5, 1e - 5, 1e - 5]);$$

A.4.2 Multi-Corner Time Optimal Test

PLAPC parameters for time optimal in Chapter 8.3

$$Q_{line} = 180; Q_{lim} = 250; Q_{vel} = 2500;$$

$$D_{line} = 2 \text{ meters}$$

Regularisation factor,

$$Q_u = \text{diag} ([7.6e - 5, 7.6e - 5, 3.8e - 5, 1260]);$$

$$Q_{du} = \text{diag} ([1e - 5, 1e - 5, 1e - 5, 1e - 5]);$$

A.4.3 Multi-Corner Individual Velocity Test

PLAPC parameters for Velocity Profile 1, Distance Objective in Chapter 8.4

$$Q_{line} = 180; Q_{lim} = 350; Q_{vel} = 2500;$$

$$D_{line} = 2 \text{ meters}$$

Regularisation factor,

$$Q_u = \text{diag} ([2.645e - 4, 2.645e - 4, 1.08e - 4, 14000]);$$

$$Q_{du} = \text{diag} ([1e - 5, 1e - 5, 1e - 4, 1e4]);$$

PLAPC parameters for Velocity Profile 2, Comfort Objective in Chapter 8.4

$$Q_{line} = 180; Q_{lim} = 250; Q_{vel} = 2500;$$

$$D_{line} = 2 \text{ meters}$$

Comfort weighting components,

$$Q_{cax} = 2; Q_{cay} = 2; Q_{cdax} = 20; Q_{cday} = 20;$$

Regularisation factor,

$$Q_u = \text{diag} ([7.6e - 5, 7.6e - 5, 3.8e - 5, 1260]);$$

$$Q_{du} = \text{diag} ([1e - 5, 1e - 5, 1e - 5, 1e - 5]);$$

A.4.4 Full Circuit Simulation, Common Velocity Test

PLAPC parameters for Distance Objective in Chapter 8.5

$$Q_{line} = 180; Q_{lim} = 350; Q_{vel} = 5500;$$

$$D_{line} = 2 \text{ meters}$$

Regularisation factor,

$$Q_u = \text{diag} ([2.76e - 4, 2.76e - 4, 9e - 5, 20000]);$$

$$Q_{du} = \text{diag} ([1e - 4, 1e - 4, 1e - 4, 1e4]);$$

PLAPC parameters for Energy Objective in Chapter 8.5

$$Q_{line} = 162; Q_{lim} = 350; Q_{vel} = 5500;$$

$$D_{line} = 2 \text{ meters}$$

LINEX parameters,

$$a = 0.01; b = -0.3; c = 20; a0 = 0;$$

Regularisation factor,

$$Q_u = \text{diag} ([2.76e - 4, 2.76e - 4, 9e - 5, 20000]);$$

$$Q_{du} = \text{diag} ([1e - 4, 1e - 4, 1e - 4, 1e4]);$$

PLAPC parameters for Comfort Objective in Chapter 8.5

$$Q_{line} = 162; Q_{lim} = 350; Q_{vel} = 5500;$$

$$D_{line} = 2 \text{ meters}$$

Comfort weighting components,

$$Q_{cax} = 4; Q_{cay} = 2; Q_{cdax} = 40; Q_{cdax} = 10;$$

Regularisation factor,

$$Q_u = \text{diag} ([2.76e - 4, 2.76e - 4, 9e - 5, 20000]);$$

$$Q_{du} = \text{diag} ([1e - 4, 1e - 4, 1e - 4, 1e4]);$$

A.4.5 Full Circuit Simulation, Individual Velocity Test

PLAPC parameters for Velocity Profile 1, Distance Objective in Chapter 8.6

$$R = \text{diag} ([0.00001, 0.00001, 0.00001, 0.01]);$$

$$Q_{line} = 180; Q_{lim} = 350; Q_{vel} = 5500;$$

$$D_{line} = 2 \text{ meters}$$

Regularisation factor,

$$Q_u = \text{diag} ([2.76e - 4, 2.76e - 4, 9e - 5, 20000]);$$

$$Q_{du} = \text{diag} ([1e - 4, 1e - 4, 1e - 4, 1e4]);$$

PLAPC parameters for Velocity Profile 2, Comfort Objective in Chapter 8.6

$$Q_{line} = 162; Q_{lim} = 350; Q_{vel} = 5500;$$

$$D_{line} = 2 \text{ meters}$$

Comfort weighting components,

$$Q_{cax} = 4; Q_{cay} = 2; Q_{cdax} = 40; Q_{cday} = 10;$$

Regularisation factor,

$$Q_u = \text{diag} ([2.76e - 4, 2.76e - 4, 9e - 5, 20000]);$$

$$Q_{du} = \text{diag} ([1e - 4, 1e - 4, 1e - 4, 1e4]);$$

B Path Tracking Model

B.1 Problem Formulations

For the NMPC problem formulations of path tracking, this work is referred to [88], [113] where the vehicle's position is described in Cartesian coordinates, and the equations of motion for both its position and yaw angle are defined as,

$$\frac{dX}{dt} = V_x \cos(\psi) - V_y \sin(\psi) \quad (\text{B-1})$$

$$\frac{dY}{dt} = V_x \sin(\psi) + V_y \cos(\psi) \quad (\text{B-2})$$

$$\frac{d\psi}{dt} = r \quad (\text{B-3})$$

Referring to the equation of motion as in (4-5) - (4-7), the states vector for the path tracking NMPC formulation is given by

$$st_{ptc} = [X, Y, \psi, V_x, V_y, r]^T \quad (\text{B-4})$$

and the control inputs is,

$$in_{ptc} = [T_{rr}, T_{rl}, T_f, \delta_f, \delta_r]^T \quad (\text{B-5})$$

In addition, there is auxiliary variables of,

$$aux_{ptc} = [a_x, a_y, D_{fl}, D_{fr}, D_{rl}, D_{rr}]^T \quad (\text{B-6})$$

This NMPC controller used an explicit Runge-Kutta 4th-order technique to discretize this non-linear continuous-time system. From the path tracking perspective, the controller is expected to track or follow the given reference path and velocity from the optimal solution result. Therefore, this NMPC problem can be formulated as,

$$\min_{x,u} \sum_{k=0}^{N-1} ((x_{k+1} - x_{k+1}^{ref})^T Q_{PTC} (x_{k+1} - x_{k+1}^{ref}) + (u_k - u_k^{ref})^T R_{PTC} (u_k - u_k^{ref})) \quad (\text{B-7})$$

subject to,

$$x_0 = x_{initial}$$

$$x_{k+1} = f_d(x_k, U_k)$$

$$x_k^{min} \leq x_k \leq x_k^{max}$$

$$u_k^{min} \leq u_k \leq u_k^{max}$$

$$y_k^{min} \leq y_k \leq y_k^{max}, k = 0, 1, \dots, N - 1$$

Where N is the length of the prediction horizon, Q_{PTC} is the weighting matrices for states vector, R_{PTC} is weighting matrices for control input and f_d represent as the discrete-time system of this path-tracking model, which is derived from the RK4 method on a continuous time system.

C Others

C.1 Four-Wheel Steering Vehicle Model

This appendix section shows a four-wheel steering (4WS) vehicle model that is used for optimal solution generation, PTC and PLAPC simulations.

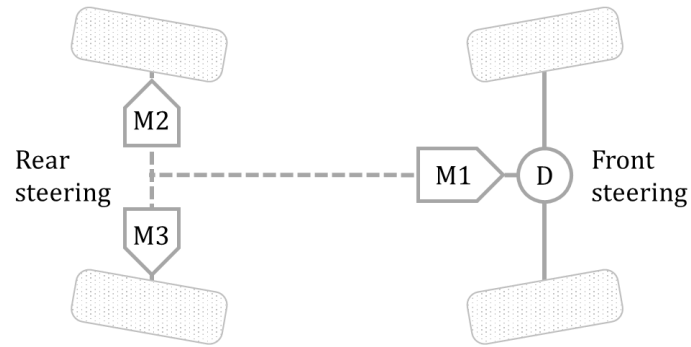


Figure C-1 Vehicle topology

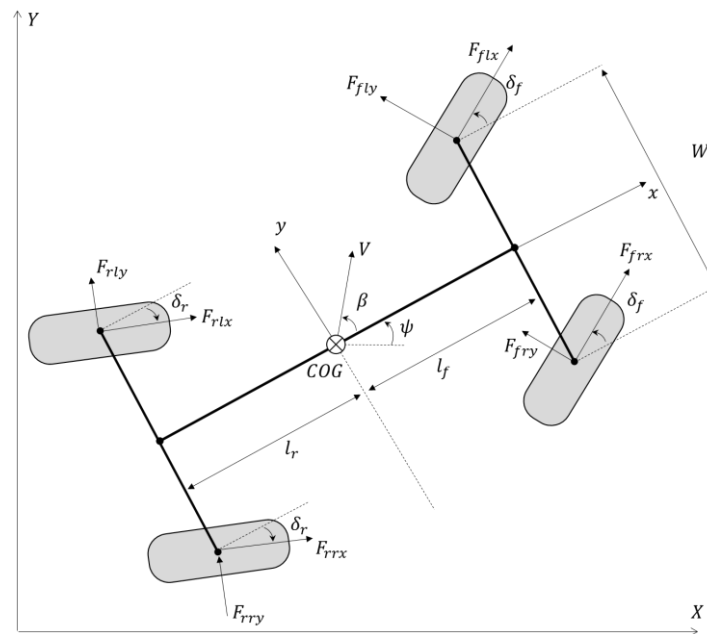


Figure C-2 Four-wheel double track vehicle dynamic model for 4WS vehicle topologies

Longitudinal, lateral and rotational dynamics for the 4WS vehicle can be expressed as,

$$\begin{aligned}
 F_x = m(\dot{V}_x - \dot{\psi}V_y) = & F_{flx} \cos(\delta_f) - F_{fly} \sin(\delta_f) + \\
 & F_{frx} \cos(\delta_f) - F_{fry} \sin(\delta_f) + \\
 & F_{rlx} \cos(\delta_r) - F_{rly} \sin(\delta_r) + \\
 & F_{rrx} \cos(\delta_r) - F_{rry} \sin(\delta_r) - F_{drag}
 \end{aligned} \tag{C-1}$$

$$\begin{aligned}
 F_y = m(\dot{V}_y + \dot{\psi}V_x) = & F_{flx} \sin(\delta_f) - F_{fly} \cos(\delta_f) + \\
 & F_{frx} \sin(\delta_f) - F_{fry} \cos(\delta_f) + \\
 & F_{rlx} \sin(\delta_r) - F_{rly} \cos(\delta_r) + \\
 & F_{rrx} \sin(\delta_r) - F_{rry} \cos(\delta_r)
 \end{aligned} \tag{C-2}$$

$$\begin{aligned}
 I_z \ddot{\psi} = & l_f (F_{flx} \sin(\delta_f) + F_{frx} \sin(\delta_f) + F_{fly} \cos(\delta_f) + F_{fry} \cos(\delta_f)) - \\
 & l_r (F_{rlx} \sin(\delta_r) + F_{rrx} \sin(\delta_r) + F_{rly} \cos(\delta_r) + F_{rry} \cos(\delta_r)) - \\
 & W_l (F_{flx} \cos(\delta_f) - F_{fly} \sin(\delta_f) + F_{rlx} \cos(\delta_r) - F_{rly} \sin(\delta_r)) + \\
 & W_r (F_{frx} \cos(\delta_f) - F_{fry} \sin(\delta_f) + F_{rrx} \cos(\delta_r) - F_{rry} \sin(\delta_r))
 \end{aligned} \tag{C-3}$$

In the equation (4-5), (4-6) and (4-7), $\dot{\psi}$ is the yaw rate and I_z is vehicle's moment of inertia at vertical axis of COG. The distance between front axle to COG is noted as l_f while the opposite side, to the rear axle is noted as l_r in which total become the vehicle wheelbase. For the other dimension, l_{wl} and l_{wr} are both represent vehicle track width or left and right distance to the CG. Since this is a multi-actuated system, rear wheel steering is denoted as δ_r with δ_f as front. While longitudinal and lateral tire forces denoted as F_{ija} with $i = (f, r)$ for front and rear position, $j = l, r$ for left and right and $a = x, y$ as axis with respect to the CG as shown in Figure C-2.

And the velocity components for the 4WS vehicle model are defined as,

$$V_{flx} = V \cos(\delta_f - \beta) - \dot{\psi} l_{fl} \sin(\gamma_{fl} - \delta_f) \quad (\text{C-4})$$

$$V_{fly} = -V \sin(\delta_f - \beta) + \dot{\psi} l_{fl} \cos(\gamma_{fl} - \delta_f)$$

$$V_{frx} = V \cos(\delta_f - \beta) + \dot{\psi} l_{fr} \cos(\gamma_{fr} + \delta_f)$$

$$V_{fry} = -V \sin(\delta_f - \beta) + \dot{\psi} l_{fr} \sin(\gamma_{fr} + \delta_f)$$

$$V_{rlx} = V \cos(\delta_r - \beta) - \dot{\psi} l_{rl} \sin(\gamma_{rl} - \delta_r)$$

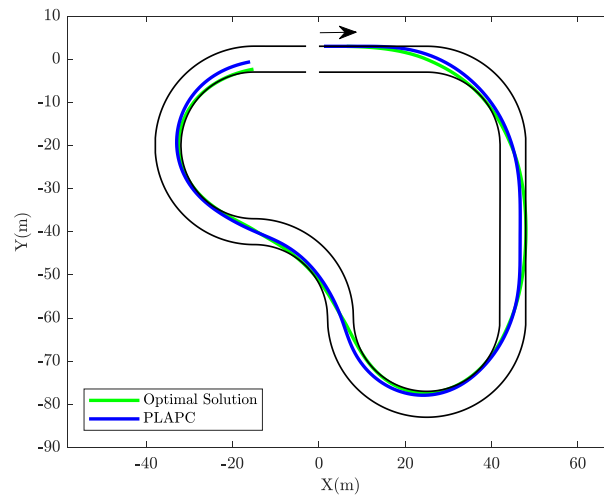
$$V_{rly} = -V \sin(\delta_r - \beta) - \dot{\psi} l_{rl} \cos(\gamma_{rl} - \delta_r)$$

$$V_{rrx} = V \cos(\delta_r - \beta) + \dot{\psi} l_{rr} \cos(\gamma_{rr} + \delta_r)$$

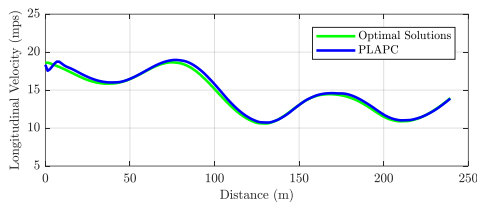
$$V_{rry} = -V \sin(\delta_r - \beta) - \dot{\psi} l_{rr} \sin(\gamma_{rr} + \delta_r)$$

C.2 Multi Corners Extended Path Evaluation with Time Optimal Velocity Profile with 4WS Vehicle Topologies

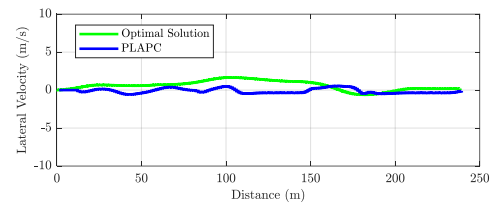
This subchapter contains information about the PLAPC evaluation using 4WS vehicle topologies with distance objective againsts time optimal objectives of optimal solutions. Based on the result presented here, an additional rear-wheel steering (RWS) actuator impacts the generated trajectory, which is closer to an optimal solution in terms of lateral deviations than FWS-only configurations.



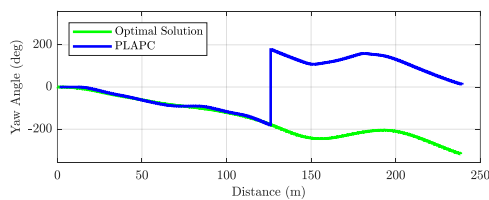
(a) Trajectories



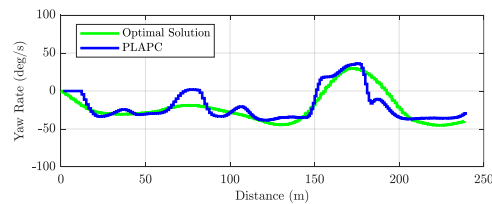
(b) Longitudinal Velocity



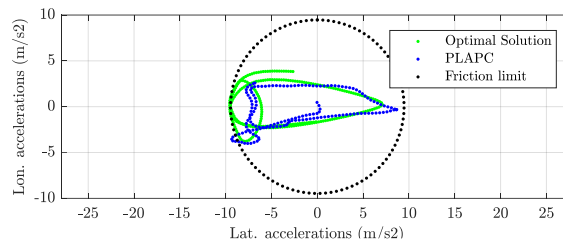
(c) Lateral Velocity



(d) Yaw Angle

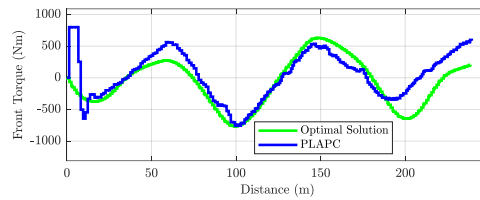


(e) Yaw Rate

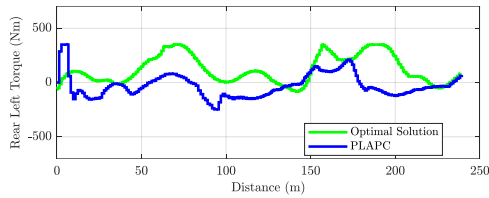


(f) G-G Diagram

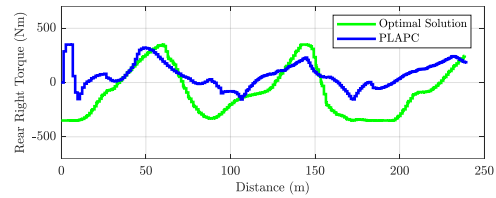
Figure C-3 4WS result comparison for time objective between optimal solutions and PLAPC-4WS controller results



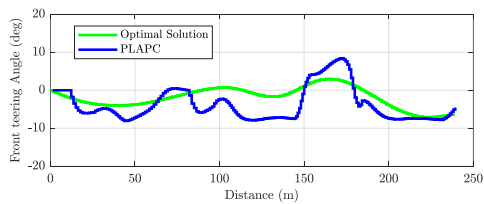
(a) Front torque



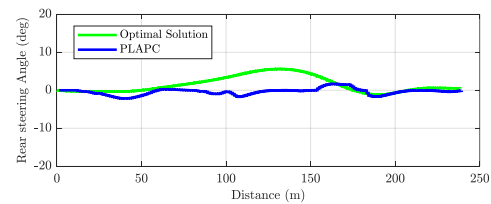
(b) Rear left torque



(c) Rear right torque



(d) Front Steering angle



(e) Rear Steering angle

Figure C-4 4WS vehicle control inputs comparison for time objective between optimal solutions and PLAPC-4WS controller results

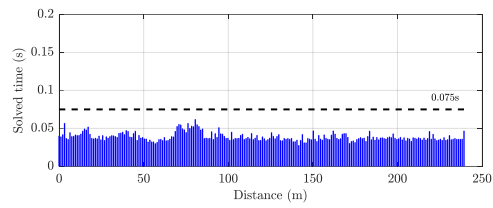


Figure C-5 IPOPT solver information during the simulations

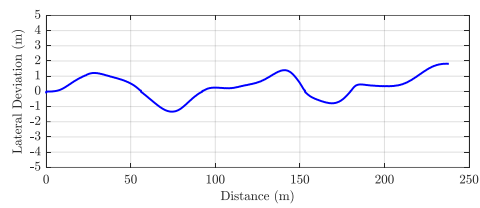


Figure C-6 Lateral error for time-optimal objective between optimal solutions and PLAPC-4WS controller results

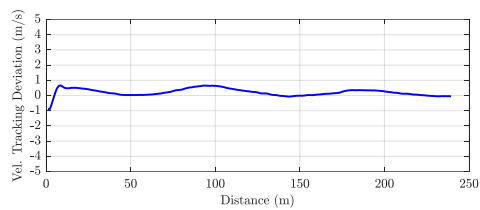


Figure C-7 Speed tracking error for time-optimal objective between optimal solutions and PLAPC-4WS controller results

Table C-1 Result of time objective in the multi-corner simulations

Vehicle topology	4WS
Average solved time (s)	0.0389
Peak solved time (s)	0.0617
No. of solver errors	0
RMS error of velocity tracking (m/s)	0.3015
RMS error of lateral deviation (m) (Generated trajectory to optimal solution)	0.8320

Table C-2 Result of lap time in the multi-corner simulations

Vehicle topology	4WS
PLAPC lap time (s) – with 90% velocity factor	16.3985
Optimal solution lap time (s)	13.9035
Time difference (s)	2.4950

Nonetheless, although 4WS configurations have smaller deviations for both velocity tracking and lateral deviation to an optimal solution, the FWS configuration is retained as a baseline for the PLAPC development. This is because FWS is a common vehicle topologies and additional actuators, such as rear-wheel steering, are expected to give more flexibility and manoeuvrability but higher computational cost.

C.3 Direct Method Comparison – Multiple Shooting vs Collocations

In this appendix, the direct method via the collocation technique is presented. In this additional information, the RK4 and collocation methods are compared by testing them on a 90-degree right-hand corner with open initial conditions for both starting position and with a time-optimal objective function. As illustrated in Figure C-8 and Figure C-9, there is no significant difference in the generated trajectories or velocity profiles between the two methods. Furthermore, Table C-3 shows that the lap times are nearly identical, with a minimal difference of just 0.0212 seconds. However, regarding computation time, the collocation method is significantly faster than the multiple-shooting method.

Nonetheless, although the collocation method solves problems faster than the RK4 method, the RK4 method is retained for optimal solution generation to ensure consistency throughout the online implementation of PTC validation and PLAPC development.

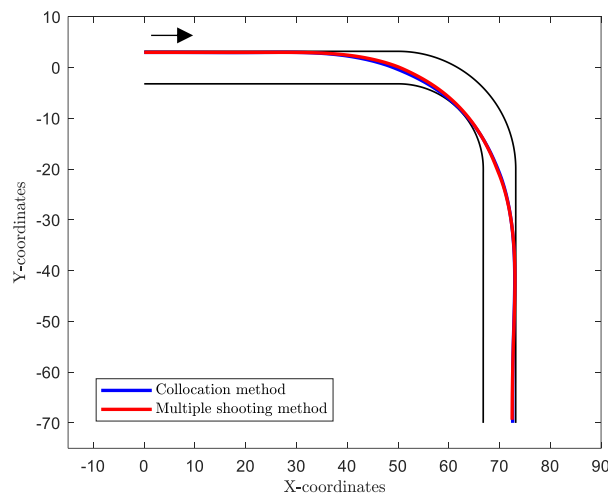


Figure C-8 Generated trajectories comparison between the collocation method and multi-shooting method in solving OCP

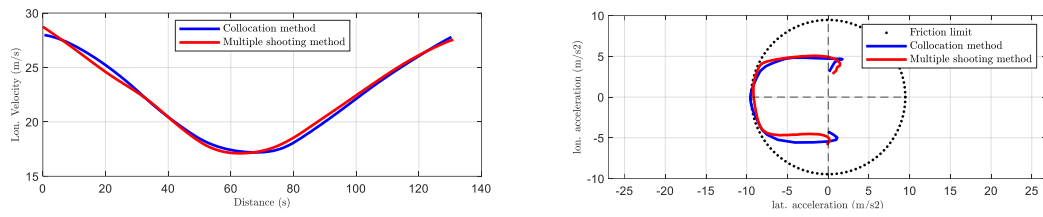


Figure C-9 Accelerations and velocities result in comparison between the collocation method and multi-shooting method in solving OCP

Table C-3 Lap time and solved time comparison between collocation method and multi-shooting method

	Lap time (s)	Solved time (s)
Collocation method	5.9946	14.2480
Multiple-shooting method	5.9607	44.1441