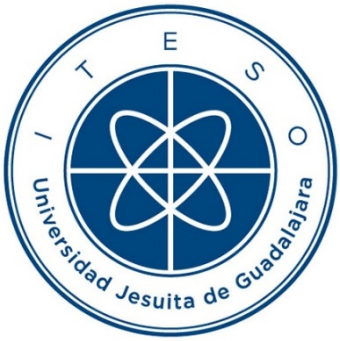
INSTITUTO TECNOLÓGICO
Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018,
publicado en el Diario Oficial de la Federación el 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática

Doctorado en Ciencias de la Ingeniería



MODELADO Y DISEÑO DE CONTROLADORES POR MODOS DESLIZANTES PARA VEHÍCULOS AUTÓNOMOS ELÉCTRICOS

Tesis que para obtener el grado de
Doctor en Ciencias de la Ingeniería
presenta: Luis Arturo Torres Romero

Director de tesis: Dr. Luis Enrique González Jiménez

Tlaquepaque, Jalisco. Marzo de 2026

TITULO: **MODELADO Y DISEÑO DE CONTROLADORES POR MODOS DESLIZANTES PARA VEHÍCULOS AUTÓNOMOS ELÉCTRICOS**

AUTOR: Luis Arturo Torres Romero
Ingeniero Comunicaciones y Electrónica (UdeG, México)
Maestro en Ciencias Computacionales (UAG, México)

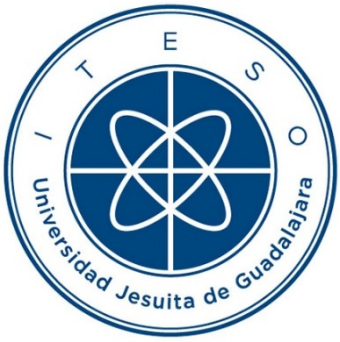
DIRECTOR DE TESIS: Luis Enrique González Jiménez
Departamento de Electrónica, Sistemas e Informática, ITESO
Ingeniero en Sistemas Computacionales (Instituto Tecnológico de Sonora - ITSON)
Maestro en Ciencias de la Eléctrica (Cinvestav, Unidad Gdl.,9 México)
Doctor en Ciencias de la Eléctrica (Cinvestav, Unidad Gdl., México)

NÚMERO DE PÁGINAS: XXXIV, 262

ITESO – The Jesuit University of Guadalajara

Department of Electronics, Systems and Informatics

Doctoral Program in Engineering Sciences



MODELING AND DESIGN OF SLIDING MODE CONTROLLERS FOR AUTONOMOUS ELECTRIC VEHICLES

Thesis to obtain the degree of

Doctor in Engineering Sciences

Presents: Luis Arturo Torres Romero

Thesis Director: Dr. Luis Enrique González Jiménez

Tlaquepaque, Jalisco, Mexico

March 2026

TITLE: **MODELING AND DESIGN OF SLIDING MODE CONTROLLERS FOR AUTONOMOUS ELECTRIC VEHICLES**

AUTHOR: Luis Arturo Torres Romero
Bachelor's degree in Communication Systems and Electronics (UdeG, México)
Master's degree in Computer Science (UAG, México)

THESIS DIRECTOR: Luis Enrique González Jiménez
Department of Electronics, Systems, and Informatics, ITESO
Bachelor's degree in Computer System Engineering (Instituto Tecnológico de Sonora - ITSON)
Master's degree in electrical engineering (Cinvestav, Unidad Gdl., Mexico)
Ph.D degree in electrical engineering (Cinvestav, Unidad Gdl., Mexico)

NUMBER OF PAGES: XXXIV, 262

To Aurorita,
Thanks for joining me in the middle of this journey.
I really needed this support.

To Valeria and Josue,
I have watched you grow and admire you
deeply for your determination since you were little.
Seeing you give your all to grow with such strength inspired me
to do the same alongside you.

Resumen

Esta tesis aborda el diseño y análisis de sistemas de control aplicados a vehículos automotrices, con un enfoque especial en la dinámica lateral y longitudinal. Se desarrollan y analizan modelos físico–matemáticos del vehículo, considerando factores como la resistencia aerodinámica, la interacción de las llantas con el asfalto y otros aspectos relevantes. Asimismo, se evalúan distintos esquemas de motores eléctricos empleados como actuadores para el control de dichas dinámicas. El objetivo principal es desarrollar un modelo del vehículo y diseñar estrategias de control para la dinámica lateral y longitudinal, con miras a su aplicación en vehículos eléctricos y autónomos.

Summary

This thesis addresses the design and analysis of control systems applied to automotive vehicles, with a particular focus on lateral and longitudinal dynamics. Physical–mathematical models of the vehicle are developed and analyzed, taking into account factors such as aerodynamic drag, tire–road interaction, and other relevant aspects. Additionally, different electric motor configurations used as actuators for controlling these dynamics are evaluated. The main objective is to develop a vehicle model and design control strategies for lateral and longitudinal dynamics, aiming at their application in electric and autonomous vehicles.

Acknowledgements

The author wishes to express his heartfelt appreciation and admiration to Dr. Luis Enrique González Jiménez, professor in the Department of Electronics, Systems, and Informatics at ITESO, for his guidance and patience in addressing these challenging topics as thesis director, which at times brought tears of frustration. Sincere gratitude is also extended to Dr. Riemann Ruiz Cruz of ITESO for his invaluable support. Additionally, the author acknowledges the contributions of his course professors, Dr. Ernesto Rayas, Dr. Roberto Osorno, and Dr. José Francisco Cervantes Álvarez, all from ITESO, and Dr. Ivica Kalichanin Balich currently from UDG.

The author gratefully acknowledges the authorities of ITESO for the financial support through an assistantship granted by the ITESO's Agreement with the Industry (Convenio Con la Industria de ITESO).

The author would also like to express his gratitude to Fernando Valenzuela, Raul Oliden, and Hector Martinez from the Sports and Health Department at ITESO. Although not directly involved in the development of this work, they made this long journey more enjoyable and inspiring by introducing him to the world of mountaineering through ITESO's traditional hike to the Nevado de Colima. They witnessed him on the summit, surrounded by books and lost in his calculations. This experience became a cherished part of the PhD program adventure, one that brought him immense joy. Friends, I've reached the summit of this mountain as well… now, let's conquer the next one.

The author also extends his heartfelt gratitude to his colleagues José Javier Méndez Pazarín, his direct supervisor, and Mr. Ricardo Montes, the Research and Development Administrator in the department, both from Continental Automotive. Thank you both for believing in me and for your support.

The author also expresses gratitude to the authorities at Continental Automotive for the financial support provided to pursue his doctoral studies.

Contenido

Resumen . . . . . **VII**

Summary . . . . . **IX**

Agradecimientos . . . . . **XI**

Contenido . . . . . **XII**

Contents . . . . . **XVIII**

Lista de Figuras . . . . . **XXIV**

Lista de Tablas . . . . . **XXXII**

Lista de Algoritmos . . . . . **XXXIII**

Introducción **1**

1. Planteamiento del Problema **5**

2. Marco Teórico **7**

2.1. Modelado en Espacio de Estados 7

2.1.1 Introducción a la Representación en el Espacio de Estados 8

2.1.2 Sistemas Lineales 11

2.1.2.1 Descomposición QR 11

2.1.2.2 Forma Controlador 12

2.2. Estabilidad de Sistemas Dinámicos 13

2.2.1 Estabilidad en el Sentido de Lyapunov 14

2.3. Subsistemas Dinámicos de un Vehículo Automotriz 18
2.3.1 Subsistema de Tren de Potencia 18
2.3.2 Subsistema de Chasis 20
2.3.3 Subsistema Eléctrico y de Control Electrónico 21
2.3.4 Subsistema de Interacción Hombre-Máquina 24
2.3.5 Subsistema de Interacción con el Entorno 25
2.4. Actuadores en Vehículos Eléctricos Automotrices 25
2.4.1 Motor de CD 31
2.4.1.1 Estimación de Parámetros y Simulación de Máquinas Eléctricas de Corriente Directa 33
2.4.1.2 Control de Motores de CD 36
2.4.2 Motor de CD sin Escobillas (BLDC) 37
2.4.2.1 Estimación de Parámetros y Simulación de Máquinas Eléctricas de Corriente Directa sin Escobillas (BLDC) 41
2.4.3 Esquemas de Control para Motor de CD sin Escobillas 43
2.4.3.1 Las Transformaciones de Park y Clarke de los Campos Eléctricos de Máquinas Trifásicas 43
2.4.3.2 Transformación al Modelo DQ de Máquinas Eléctricas de Corriente Directa sin Escobillas 50
2.4.3.3 Control Orientado a Campo para BLDC 52
2.4.3.4 Modulación por Ancho de Pulso de Vector Espacial 58
2.4.4 Motor Síncrono de Imanes Permanentes (PMSM) 61
2.4.5 Motor de Reluctancia Conmutada 62
2.4.6 Motor de Inducción de CA (Asíncrono) 65
2.5. Control por Modos Deslizantes 66
2.5.1 Breve Introdución a la Teoría de Control 66
2.5.2 Principios del Control por Modos Deslizantes 68
2.5.2.1 Control Equivalente 71
2.5.3 Ubicación de Polos 72
2.5.3.1 Forma Regular 72

2.5.3.2 Colocación de Eigenvalores 76
2.5.4 Control por Modos Deslizantes Integrales 77
2.5.5 Control por Modos Deslizantes de Alto Orden 78
2.5.5.1 Controlador Super-Twisting 78
2.5.5.2 Derivador de Levant 80

3. Modelo Dinámico de un Vehículo Eléctrico y Convencional 83
3.1. Dinámicas del Vehículo 83
3.1.1 Dinámicas Longitudinales 84
3.1.1.1 El Modelo Longitudinal del Vehículo 86
3.1.1.2 La Fuerza de la Llanta y la Formula Mágica 88
3.1.1.3 Las Dinámicas de la Rueda 90
3.1.1.4 El Convertidor de Par 91
3.1.1.5 La Transmisión 92
3.1.1.6 El Motor de Combustión 93
3.1.2 Dinámicas Laterales 96
3.1.2.1 Modelo Cinemático de la Bicicleta 97
3.1.2.2 Control del Ángulo de Orientación ψ (Control por Retro de Error) 99
3.1.2.3 El Modelo Dinámico de las Dinámicas Laterales de Vehículo . . 101
3.1.3 Modelo Dinámico en Terminos del Error Respecto del Camino 103
3.1.3.1 Controlador para Dinamicas Laterales con Técnica de Ackermann 105
3.1.3.2 Diseño de Control por Bloques para la Dinámica Lateral 109
3.2. Construcción de un Modelo de Vehículo Eléctrico para Simulación 111
3.2.1 Introducción 112
3.2.2 Conversión de Coordenadas y Errores de Trayectoria 113
3.2.3 El Modelo del Vehículo Eléctrico 119
3.2.3.1 Dinámicas Longitudinales 120
3.2.3.2 La Transmisión 122
3.2.3.3 El Modelo de la Llanta 123
3.2.3.4 Dinámicas Laterales 125

3.2.3.5 Bloques Variantes y el Modelo Completo 128
3.2.4 Simulaciones 130
3.2.5 Conclusiones 137

4. Controladores por Modos Deslizantes para Vehículos Eléctricos . 139

4.1. Controlador de Modos Deslizantes de Seguimiento de Ruta para un Vehículo Eléctrico Considerando la Dinámica del Actuador 139
4.1.1 Introducción 139
4.1.2 Seguimiento de Ruta para un Vehíclulo Automotriz 142
4.1.3 Esquema General de Control 144
4.1.3.1 Controlador de Modos Deslizantes Lateral 145
4.1.3.2 Controlador del Actuador 148
4.2. Control por Bloques de Modos Deslizantes con Super-Twisting para Seguimiento de Ruta de un Vehículo Autónomo Considerando el Actuador 152
4.2.1 Introducción 152
4.2.2 Seguimiento de Ruta para un Vehículo Autodirigido 154
4.2.3 Esquema de Control General 156
4.2.5 Controlador del Actuador 160
4.2.5.1 Dinámicas del Motor BLDC 161
4.2.5.2 Modelo del Sistema de Dirección 162
4.2.5.3 Algoritmo de Control de Posición 162
4.3. Control Lineal por Bloques más Super Twisting para Rechazo de Perturbaciones para Seguimiento de Ruta de un Vehículo Autónomo Considerando el Actuador (Variante) 164
4.3.1 Introducción 164
4.3.2 Controlador Lateral de Modos Deslizantes 164
4.3.3 Actuador y su Controlador 170

5. Análisis de Desempeño de los Controladores Propuestos 171

5.1. Controldaor de Modos Deslizantes de Seguimiento de Ruta para un Vehículo Eléctrico Considerando la Dinamica del Actuador 171

5.1.1 Resultados de las Simulaciones . . . 171
5.1.2 Configuracion del Sistema de Lazo-Cerrado en Simulink . . . 179
5.1.3 Discusión . . . 180
5.1.4 Conclusiones y Trabajo Futuro . . . 180
5.2. Control por Bloques de Modos Deslizantes con Super-Twisting para Seguimiento de Ruta de un Vehículo Autónomo Considerando el Actuador . . . 182
5.2.1 Resultados de la Simulación . . . 182
5.2.2 Discusión . . . 187
5.2.3 Conclusiones y Trabajo Futuro . . . 188
5.3. Control Lineal por Bloques más Super Twisting para Rechazo de Perturbaciones para Seguimiento de Ruta de un Vehículo Autónomo Considerando el Actuador (Variante) . . . 191
5.3.1 Resultados de la Simulación . . . 191
5.3.2 Discusión . . . 195
5.3.3 Conclusiones y Trabajo Futuro . . . 196

General Conclusions . . . 201

Conclusiones Generales . . . 205

Apéndice . . . 209

A. Lista de Reportes Internos de Ivestigación . . . 211
B. Lista de Publicaciones . . . 213
C. Código Fuente . . . 215
C.1 Código de la Sección 02.01 . . . 215
C.2 Código de la Sección 02.02 . . . 219
C.3 Código de la Sección 02.03 . . . 222
C.4 Código de la Sección 02.05 . . . 227
C.5 Código de la Sección 03.01 . . . 232
C.6 Código de la Sección 03.02 . . . 241

Bibliografía . . . 245

Indice de Autores **256**

Indice de Terminos **260**

Contents

Resumen . . . . . **VII**

Summary . . . . . **IX**

Acknowledgements . . . . . **XI**

Contenido . . . . . **XII**

Contents . . . . . **XVIII**

List of Figures . . . . . **XXIV**

List of Tables . . . . . **XXXII**

List of Algorithms . . . . . **XXXIII**

Introduction **1**

1. Problem Statement **5**

2. Theoretical Framework **7**

2.1. State-Space Modeling 7

2.1.1 Introduction to the State-Space Representation 8

2.1.2 Linear Systems 11

2.1.2.1 QR Decomposition 11

2.1.2.2 Controller Forms 12

2.2. Dynamic Systems Stability 13

2.2.1 Stability in the Lyapunov Sense 14

2.3. Dynamic Subsystems of an Automotive Vehicle 18
2.3.1 Powertrain Subsystem 18
2.3.2 Chassis Subsystem 20
2.3.3 Electrical and Electronic Control Subsystems 21
2.3.4 Human–Machine Interface Subsystem 24
2.3.5 Environmental interaction subsystem 25
2.4. Actuators in Automotive Electric Vehicles 25
2.4.1 DC Motor 31
2.4.1.1 Parameter Estimation and Simulation of Direct-Current Electric Machines 33
2.4.1.2 DC Motor Control 36
2.4.2 Brush-less DC Motor (BLDC) 37
2.4.2.1 Parameter Estimation and Simulation of Brush-less Direct-Current Electric Machines (BLDC) 41
2.4.3 Brush-less DC Motor Control Schemes 43
2.4.3.1 The Park and Clarke Transformations of the Electrical Fields of Three-phase Machines 43
2.4.3.2 The DQ Model Transformation for Brush-less Direct-Current Electrical Machines 50
2.4.3.3 Field-Oriented Control for BLDC 52
2.4.3.4 Space-Vector Pulse Width Modulation 58
2.4.4 Permanent Magnet Synchronous Motor (PMSM) 61
2.4.5 Switched Reluctance Motor (SRM) 62
2.4.6 AC Induction Motor (Asynchronous) 65
2.5. Sliding Mode Control 66
2.5.1 Brief Introduction to Control Theory 66
2.5.2 Sliding Mode Control Principles 68
2.5.2.1 Equivalent Control 71
2.5.3 Pole Placement 72
2.5.3.1 Regular Form 72

2.5.3.2 Eigenvalue placement 76
2.5.4 Integral Sliding Mode Control 77
2.5.5 High-Order Sliding Mode Control 78
2.5.5.1 Super-Twisting Controller 78
2.5.5.2 Levant Differentiator 80

3. Dynamic Model of an Electric Vehicle and Conventional Vehicle . 83

3.1. Vehicle Dynamics 83
3.1.1 Longitudinal Dynamics 84
3.1.1.1 The Longitudinal Vehicle Model 86
3.1.1.2 Tire Force and The Magic Formula 88
3.1.1.3 The Wheel Dynamics 90
3.1.1.4 The Torque Converter 91
3.1.1.5 The Transmission 92
3.1.1.6 The Combustion Engine 93
3.1.2 Lateral Dynamics 96
3.1.2.1 The Kinematic Bicycle Model 97
3.1.2.2 Controlling the Heading Angle ψ (Error Feedback Controller) . . 99
3.1.2.3 The Dynamic Model of Lateral Vehicle Dynamics 101
3.1.3 Dynamic Model in Terms of Error Respect to Road 103
3.1.3.1 Ackermann Controller Design for Lateral Dynamics 105
3.1.3.2 Controller Design by Blocks of the Lateral Dynamics 109
3.2. Construction of an Electric Vehicle Model for Simulation 111
3.2.1 Introduction 112
3.2.2 Coordinates Conversion and Trajectory Errors. 113
3.2.3 The Electric Vehicle Model 119
3.2.3.1 Longitudinal Dynamics 120
3.2.3.2 The Transmission 122
3.2.3.3 The Tire Model 123
3.2.3.4 Lateral Dynamics 125

3.2.3.5 Variant Blocks and the Complete Model 128
3.2.4 Simulations 130
3.2.5 Conclusions 137

4. Sliding Mode Controllers for Electric Vehicles 139

4.1. Path Following Sliding Mode Controller for an Electric Vehicle Considering Actuator Dynamics 139
4.1.1 Introduction 139
4.1.2 Path Following for an Automotive Vehicle 142
4.1.3 Overall Control Scheme 144
4.1.3.1 Lateral Sliding Mode Controller 145
4.1.3.2 Actuator Controller 148
4.2. Block Control with Super Twisting Sliding Modes for the Path Following of a Self-Driving Vehicle Considering Actuator Dynamics 152
4.2.1 Introduction 152
4.2.2 Path Following for a Self-Driving Automotive Vehicle 154
4.2.3 Overall Control Scheme 156
4.2.4 Lateral sliding mode controller 157
4.2.5 Actuator Controller 160
4.2.5.1 BLDC Motor Dynamics 161
4.2.5.2 Steering Rack System Model 162
4.2.5.3 Position Control Algorithm 162
4.3. Linear Block Control Plus Super Twisting Sliding Modes for Disturbance Rejection for the Path Following of a Self-Driving Vehicle Considering Actuator Dynamics (Variant) 164
4.3.1 Introduction 164
4.3.2 Lateral sliding mode controller 164
4.3.3 Actuator and Its Controller 170

5. Performance Analysis of the Proposed Controllers 171

5.1. Path Following Sliding Mode Controller for an Electric Vehicle Considering Actuator Dynamics . . . 171
5.1.1 Simulation Results . . . 171
5.1.2 Configuration of the Closed-Loop System in Simulink . . . 179
5.1.3 Discussion . . . 180
5.1.4 Conclusions and Future Work . . . 180
5.2. Block Control with Super Twisting Sliding Modes for the Path Following of a Self-Driving Vehicle Considering Actuator Dynamics . . . 182
5.2.1 Simulation Results . . . 182
5.2.2 Discussion . . . 187
5.2.3 Conclusions and Future Work . . . 188
5.3. Linear Block Control Plus Super Twisting Sliding Modes for Disturbance Rejection for the Path Following of a Self-Driving Vehicle Considering Actuator Dynamics (Variant) . . . 191
5.3.1 Simulation Results . . . 191
5.3.2 Discussion . . . 195
5.3.3 Conclusions and Future Work . . . 196

General Conclusions . . . 201

Conclusiones Generales . . . 205

Appendix . . . 209

A. List of Internal Research Reports . . . 211
B. List of Publications . . . 213
C. Source Code . . . 215
C.1 Code for Section 02.01 . . . 215
C.2 Code for Section 02.02 . . . 219
C.3 Code for Section 02.03 . . . 222
C.4 Code for Section 02.05 . . . 227
C.5 Code for Section 03.01 . . . 232

C.6 Code for Section 03.02 241

Bibliography 245

Index of Authors 256

Subject Index 260

List of Figures

2.1 Block diagram of the linear time-varying system. For time-Invariant the matrix **A,B,C,D** do not depend on time t. 9

2.2 RLC circuit. 10

2.3 Diagram showing a pendulum which has two equilibrium points. a) Unstable equilibrium point. b) Stable equilibrium point. 15

2.4 Powertrain blocks diagram for electric vehicles and combustion vehicles. 20

2.5 Chassis subsystems 21

2.6 Domain vs Zonal architectures. 23

2.7 a) shows the different configurations for electric vehicles with front traction C: clutch; D: differential; FG: fixed gearing; GB: gear box; M: electric motor. b) Tesla model S rear drive. C) Tesla model S all-wheel drive configuration. Taken from [56]. 27

2.8 a) Diagram of assembly of the motor and inverter into the Tesla model S. b) Block diagram of the integrated electric motor assembly. Taken from [67]. 28

2.9 Equivalent circuit of the DC motor. 32

2.10 Simulation of the DC Motor. 34

2.11 Fragment of the motor selection catalog. Taken from Leroy-Somer catalog. 35

2.12 Closed-loop DC motor system using PID. 37

2.13 DC Motor response to the PID control. 37

2.14 BLDC parts. The rotor in made with permanent magnets; the poles in the stator are generated with the electromagnetic fields from the windings in the same stator; the hall sensors detect the position of the rotor in order to the variators be able to feed the current required in the coils. Taken from [13]. 38

2.15 Equivalent circuit of a BLDC motor over a watermark representing the physical construction. 39

2.16 Trapezoidal Back EMF and phase current variation with rotor electrical angle. Taken from [53]. 40

2.17 Stator currents and the trapezoidal function to simulate the back EMF in the simulation for BLDC. 42

2.18 Motor Response on the first 2 seconds. a) Angle position (θ) of the motor shaft. b) Angle speed of the motor shaft. 42

2.19 Electrical phases in the abc, $\alpha\beta$, *dq* reference frames. This represents one complete revolution, the horizontal axis represents the time and the angle in rads too, since the frequency of these signals are $1/2\pi$, and so, we reach 2π rads in 2π seconds. a) The original three electrical phases at $120°$. b) We can transform the three signals in only two equivalent using the Clarke transformation. c) Using the Park-Clarke we can still simplify even more the three phases to only two constants that can be handled as direct-current signals, which simplify the control of electrical machines. 44

2.20 The Clarke and Park-Clarke transformation. a) Shows the resultant vector (dashed-pink arrow) obtained with the vectorial sum of the three phases at $120°$ in a giving instant where the resultant has the magnitude of 24.5 and a direction of $\pi/4$. The three vectors (dotted-red, dashed-blue, solid-green) represent the three-phase voltages in the coils. b) is the equivalent 2 two orthogonal vectors in the α, β reference frame; this is the Clarke transformation. Contrary to a), these two vectors are not physical magnitudes but the mathematical equivalence. c) if we rotate the α, β reference frame (Park-Clarke transformation), the same degrees where the resultant vector is, what we get is the two orthogonal vectors but these no longer sinusoidal rather, what we see is two constants. In green with 0 we only show that the second magnitude is 0 because the blue-dashed is the same magnitude as the resultant. If the voltages were out of phase, we would see both orthogonal vectors different from 0. 45

2.21 Park-Clarke amplitude and power invariant transformations. The magnitude of the resultant vector using power invariant transform (blue) is equal to the vector sum of the 3 electrical phases. The magnitude of the resultant vector after transformation using the amplitude invariant transform is equal to the peak value of the original electrical phases (red). 46

2.22 Effect of the Clarke transformation when three trapezoidal phases are used as input. During the first 15 seconds, a sinusoidal 3 phases signal is used as input, getting in the output 2 phases sinusoidal signals plus 1 zero phase. During the second 15 to 30, it was fed with a trapezoidal signal and, in the output, we got a messy 3 phases signal. 49

2.23 Effect of the Park-Clarke transformation when three trapezoidal phases are used as input. During the first 15 seconds, a sinusoidal 3 phases signal is used as input, getting in the output 2 constant signals. During the second 15 to 30, it was fed with a trapezoidal signal and, in the output, we got a messy 2 phases signal. 49

2.24 Control scheme to control the torque of a BLDC motor. 54

2.25 Control scheme to control a) the angular speed, b) the absolute angle position of a BLDC motor. 54

2.26 Simulink blocks to control the angle of the BLDC. 55

2.27 Simulation of the model in Fig. 2.26. *Angle reference* is the reference angle position commanded to the BLDC.*BLDC shaft angle* is the actual response of the BLDC. . 56

2.28 SVPWM. Space vectors of three-phase bridge inverter showing reference voltage trajectory and segments of adjacent voltage vectors. 59

2.29 V_s vector as a result of the two vectors V_a and V_b separated by $60°$. 60

2.30 Scheme showing the difference between surface permanent magnet and internal permanent magnet. 62

2.31 Scheme showing the construction of the switched reluctance motor of one phase and two poles. 63

2.32 Basic elements of a closed loop control system. 67

2.33 Comparison of a basic system response with a linear controller. a) with no disturbance. b) with disturbance. 70

2.34 Response of the system (2.140) with the SMC control law in (2.142) applied. a) state variables response. b) Control law u, sliding variable σ, and disturbance f. . . 70

2.35 Diagram showing the Levant differentiator performance. 81

3.1 Components of the driveline of a car with SI engine. It is shown the main components in charge to transmit the power of the motor to the wheels. The engine generates the power that is transmitted to the transmission through the torque converter which transmits the torque using a fluid to allow a complete stop without stalling the motor or passing the torque or even multiply it when the motor accelerates. The transmission transfers the torque to the wheels, coupling the velocities using a set of different gears. 85

3.2 Different forces acting over the vehicle. F_{aero} is the air drag force. R_x is the rolling resistance. F_x is the longitudinal force. Mass and gravity are denoted by m and g. . 87

3.3 Experimental graph of the longitudinal force of the tires vs tire slip at a fixed normal load. Here we can appreciate the maximum force F_x according the normal loads such as 2000N,3000N, 4000, 5500N. Taken from [28]. 90

3.4 Torque converter Kotwicki model. This is a simulation of the fluid coupling mode ($\omega_t/\omega_p >= 0.9$). most part of the function grows rapidly to the negative. 92

3.5 Schematic of the transmission model. 93

3.6 Wheel Tractive Force–Speed Curves for Different Gear Ratios (Four-Speed Gearbox). Taken from [4]. 93

3.7 Kinematic model of the vehicle better known as “the bicycle model”. This is a simplified model of the vehicle that only uses two wheels, rear and front. Taken from [72]. 97

3.8 This is the behavior of the kinematic bicycle model. In the upper graph the angle of the steering wheel is shown in radians. We have two maneuvers: the first 5 seconds, we increment 0.8°/s (0.014rads/s). From the second 5 to 10, it decrements 1.6°/s. From second 10 to 30 there is no increment nor decrement. In the lower graph we see the trajectory of the CoG of the vehicle (distance is in meters). 99

3.9 This is the simulation of the controller for ψ. The upper graph shows the trajectory of the vehicle. The lower graph shows the comparison between the reference and the controlled signal. We can appreciate that we have always an error in stable state. 101

3.10 This is the simulation of the controller for ψ for the first 15 seconds. We can appreciate the error in stable state and the transient behavior at the beginning. The red line is the reference, and the blue one is our controlled signal. 101
3.11 Components of the lateral dynamics. 102
3.12 Diagram showing the slip angle component. δ is the steering angle, V the velocity vector of the vehicle with θ_{vf} as its angle. The lateral slip angle α is defined as $\delta - \theta_{vf}$. 102
3.13 Error behavior using the Ackerman controller design for the model of error respect the road. 107
3.14 Trajectory that the vehicle would follow using the Ackerman controller design. . . . 108
3.15 Comparison of the control laws for Ackerman (in red) and by blocks (in blue). . . . 110
3.16 Comparison of the resultant trajectories between Ackerman (dashed green) and by block controller (dashed red). The black line is the reference. 111
3.17 The trajectory of the vehicle (blue) over the reference path (red). Path in meters. a) vehicle trajectory. b) Close-up of the trajectory. 114
3.18 Desired trajectory and the error respecting the vehicle position. 115
3.19 GetRef Simulink® block. 117
3.20 Coordinates conversion block. 118
3.21 Coordinates conversion block inside. 118
3.22 Overall control scheme for an electric vehicle. 119
3.23 Vehicle dynamical model. 120
3.24 Longitudinal controller block diagram. 122
3.25 Longitudinal dynamics Simulink® block. 122
3.26 Simulink® block of the transmission gear. 123
3.27 Simulink® block of the tire. 124
3.28 Block diagram in Simulink for the tire model. 125
3.29 Lateral controller block diagram. 127
3.30 Longitudinal and lateral dynamic model. 127
3.31 Simulink® blocks of the longitudinal dynamics. 129
3.32 Simulink® blocks of the lateral dynamics. 129

3.33 Configuration of the solver in Simulink®. 130

3.34 Current and torque generated by the BLDC for longitudinal dynamics. Horizontal is time is in seconds. Vertical is amperes. 131

3.35 e1 is the error of the position of the vehicle against the center of the lane. 132

3.36 Angle of the motor BLDC used for steering in rads. 132

3.37 Currents in amps. of the BLDC used for steering. Horizontal is time in seconds. Vertical is amperes for the three phases. 133

3.38 Tire dynamics simulation: Above is the tire slip. Below is the force in N produced by the tire. 133

3.39 Longitudinal dynamics simulation: Vx is the actual vehicle speed during the 120 seconds of simulation. 134

3.40 Longitudinal dynamics simulation: Vx_reff is the velocity that the longitudinal controller must follow. Vx_err is Vx_reff minus the actual velocity. PID_controller1 is the control law. Tl is the torque in Nm delivered by the electrical motor to the transmission. 134

4.1 Overall control architecture for path following with different levels of autonomy. . . 140

4.2 Parameters and variables related to the lateral vehicle's dynamics. R corresponds to trajectory instant radius. 143

4.3 Overall control scheme. 145

4.4 The three reference frames V_{abc},$V_{\alpha\beta 0}$, V_{dq0}. 149

4.5 BLDC motor's angular position control scheme. 150

4.6 Overall control architecture for path following with different levels of autonomy. . . 154

4.7 a) Bicycle Model. b) Components of the lateral dynamics. 155

4.8 General control scheme for the local feedback control layer and overview of the software stack layer system for a self-driving vehicle. 157

4.9 Lateral displacement error y_e and orientation error ψ_e of the vehicle with respect to the followed path (lane). The blue-dashed arrow represents the current orientation of the vehicle. The red-dotted line represents the followed path. The orange-dotted line represents the lateral error defined as the shortest distance between the CoG of the vehicle and the desired path. 158

4.10 Control scheme of a BLDC motor. 163

5.1 Results of the path following experiment where the current path (dashed blue line) and the reference trajectory (solid red line) are depicted. 174

5.2 Close-up of the path following experiment depicted in Fig. 5.1 where the tracking error can be appreciated. 175

5.3 Variable errors e_1 (top) and e_2 (middle). Control signal δ (bottom) which is the position of the BLDC shaft that controls the steer angle. 175

5.4 States of the lateral dynamics of the vehicle. 176

5.5 BLDC Internal states. 176

5.6 Steering angle δ and its reference δ_{ref}. 177

5.7 Errors of the 4 PIs in the BLDC control loop. 177

5.8 Input voltage for the BLDC motor in the dq reference frame. 178

5.9 Input voltage for the BLDC motor in the abc reference frame. 178

5.10 Lateral control blocks in Simulink model. 179

5.11 a) Complete trajectory tracking of the vehicle in closed-loop. b) Close-up at $t = 30$ s. The solid black line is the reference trajectory and the dashed red line the actual trajectory of the vehicle. 184

5.12 Lateral displacement error y_e, orientation error ψ_e, desired δ_{ref} and actual δ steering angle during the experiment 185

5.13 Velocities and accelerations for the lateral displacement and orientation of the vehicle. 186

5.14 Error variables of the PIs controllers for the BLDC control loop as defined in 4.46. 187

5.15 Voltages of phases a, b, c that feed the BLDC. 188

5.16 Error variables response of the proposed controller against matched disturbances. a) shows the closed loop response of the lateral displacement error y_e and orientation error ψ_e considering disturbances (dashed green lines) and without disturbances (red lines), and b) shows the corresponding steering control input. 189
5.17 Comparison between the closed loop response of the controller proposed in this work (red lines) and the response of the controller in [81]. The improvement in terms of the control law magnitude and its high frequency components is clearly appreciated. 190
5.18 a) complete trajectory of the vehicle. b) close-up where the maximum deviation is shown. The black line is the reference trajectory. Dashed green line represents the actual trajectory of the vehicle using $(\delta_k + \delta_u)$. Solid red line is the trajectory using only δ_k. 194
5.19 Errors in the vehicle dynamics. 195
5.20 Internal states of the vehicle dynamics. 196
5.21 Errors in the PIs controllers for the BLDC control loop. The errors are defined in section 4.46. 197
5.22 Voltages of phases a, b, c that feed the BLDC. 198
5.23 Currents of vectors in the rotational reference frame dq read from the BLDC. . . . 198
5.24 Forces in the steering rack system. 199
5.25 Internal structure behavior of the control algorithm. 199
5.26 Error comparisons with additional disturbances. a) is a chart that represents the lateral error y_e with control $(\delta_k + \delta_u)$, solid red is without additional disturbances, dashed blue with disturbances. b) is the same than a) with ψ_e. c) control δ_k for y_e, solid red line no disturbances, blue dashed with disturbances. d) same as c) for ψ_e . 200

List of Tables

2.1 ADVANTAGES, DISADVANTAGES AND USAGE OF DIFFERENT MOTOR TYPES 29
2.2 ELECTRIC MOTOR DATA 30
2.3 ELECTRIC MOTOR DATA CONTINUATION 30
2.4 DC MOTOR PARAMETER SELECTION FROM CATALOG 34
2.5 HALL EFFECT SIGNALS 42
2.6 BLDC MOTOR PARAMETERS AND PIDs GAINS 56
2.7 DESCRIPTION OF SIMULINK BLOCKS TO CONTROL ANGLE POSITION . 57

3.1 TYPICAL VALUES FOR THE MAGIC FORMULA ACCORDING ROAD CONDITIONS 90
3.2 BLDC ELECTRICAL MOTOR TO POWER AND STEER THE VEHICLE 135
3.3 VEHICLE DATA 135
3.4 ENVIRONMENT DATA 135
3.5 AERODYNAMICS DATA 136
3.6 TIRE DYNAMICS DATA 136

5.1 PARAMETERS OF THE VEHICLE AND THE ACTUATOR 172
5.2 CONTROL GAINS FOR LATERAL AND ACTUATOR CONTROLLERS 173
5.3 PARAMETERS OF THE VEHICLE AND THE ACTUATOR. 182
5.4 CONTROL GAINS FOR LATERAL AND ACTUATOR CONTROLLERS. 183
5.5 RESULTS OF MAXIMUM AND MEAN VALUES FOR THE ERRORS y_e AND ψ_e.184
5.6 PARAMETERS OF THE VEHICLE AND THE ACTUATOR. 192
5.7 CONTROL GAINS FOR LATERAL AND ACTUATOR CONTROLLERS. 193
5.8 RESULTS OF MAXIMUM AND MEAN VALUES FOR THE ERRORS y_e AND ψ_e.193

List of Algorithms

1 RegularForm.mlx . . . 215
2 LateralDynaRF_1.mlx . . . 216
3 LateralDynaRF_2.mlx . . . 217
4 LevantDiff.m . . . 218
5 DCMotor.m . . . 219
6 BLDCMotor.m . . . 220
7 BLDCMotorFs.m . . . 221
8 PID_Controller.c . . . 222
9 dqBLDCModel.m . . . 223
10 InvParkClarke.m . . . 224
11 ParkClarke.m . . . 224
12 SpaceVectorCalculator.m . . . 225
13 EigenValPlacement.mlx . . . 227
14 EigenValPlacement2.mlx . . . 228
15 EigenValPlacement3.mlx . . . 229
16 EigenValPlacement4.mlx . . . 230
17 PureKinematicModel.m . . . 232
18 KinematicPsiCtrl_eFBK_solution.m . . . 233
19 DynaBicycleError.m part 1 . . . 234
20 DynaBicycleError.m part 2 . . . 235
21 StateFBK_Ctrl_lateralRoadError_XY_sim.m part 1 . . . 236
22 StateFBK_Ctrl_lateralRoadError_XY_sim.m part 2 . . . 237
23 StateFBK_Ctrl_lateralRoadError_XY_sim.m part 3 . . . 238
24 GetReference.m . . . 241
25 RotateVector.m . . . 243
26 LongitudinalDynamics.m . . . 244

27 Transmission.m 244
28 WheelsDynamic.m 244
29 TireSlip.m 245
30 MagicFormula.m 245

Introduction

Autonomous and electric vehicles are increasingly being introduced into real traffic environments, where they must operate alongside human-driven vehicles and pedestrians. This transition presents technical and safety challenges, particularly in the modeling and control of vehicle dynamics under uncertain and varying conditions. In addition to academic interest, these challenges are relevant from an industrial perspective, where control solutions must be robust, reproducible, and compatible with established safety practices.

Although significant progress has been made, current autonomous vehicle systems still face limitations when compared to conventional vehicles. Available statistics indicate that the safety performance of autonomous vehicles, measured in terms of miles per accident, remains lower than that of human-driven vehicles, despite a clear trend of improvement [22]. This situation motivates continued research on modeling, simulation, and control methods that allow systematic evaluation of vehicle behavior before real-world deployment.

An important aspect in the development of autonomous electric vehicles is the accurate representation of vehicle dynamics, including longitudinal and lateral motion and the influence of actuators. Electric propulsion systems introduce dynamics such as torque limits, fast responses, and coupling between electrical and mechanical subsystems. If these effects are neglected, control strategies may show acceptable performance in simplified simulations but limited applicability in real vehicles. In this work, vehicle models are developed based on established formulations from the literature and are parameterized, as far as possible, using data representative of components and products available in the automotive market.

Simulation is used as a primary tool to evaluate control strategies under a variety of driving conditions, disturbances, and parameter variations that cannot be exhaustively tested on real vehicles due to safety and cost constraints. The effectiveness of this evaluation depends on the level of detail of the vehicle and actuator models, as well as on the suitability of the selected control approaches.

The purpose of this dissertation is to develop a vehicle model suitable for the study of autonomous electric vehicles and to design control strategies for its longitudinal and lateral dynamics, explicitly considering actuator behavior, particularly that of electric machines. Sliding mode control is explored as a candidate approach due to its robustness properties, and its performance is assessed through simulation using the proposed models.

Through the combination of modeling, control design, and simulation-based evaluation, this dissertation aims to provide a structured and realistic contribution to the study of control systems for autonomous electric vehicles.

This doctoral dissertation is organized as follows:

Chapter 1, *Problem Statement* . This chapter defines the problem addressed in this dissertation, focusing on the modeling, simulation, and control challenges of autonomous electric vehicles. It motivates the use of high-fidelity simulation and robust nonlinear control strategies, particularly sliding mode control, while emphasizing the relevance of actuator dynamics and safety considerations for real-world deployment.

Chapter 2, *Theoretical Framework* . This Chapter gives some mathematical, automotive, and electrical topics that resulted useful for the analysis and design of controllers applied to the vehicle dynamics that make use of electrical machines.

Chapter 3, *Dynamic Model of an Electric Vehicle* . Here, the physics of the automotive vehicles are analyzed and the simulation framework is built using the mathematical software Matlab. The vehicle model resulted from here, will be used to test the performance of the proposed controllers.

Chapter 4, *Sliding Mode Controllers for Electric Vehicles*. Here, based on the theory of latest chapters, different controllers are defined to control the lateral or longitudinal dynamics of a vehicle.

Chapter 5, *Performance Analysis of the Proposed Controllers* . In this chapter we simulate the proposed controllers inserting them into our vehicle model. An analysis and comparison of performance of the controllers created in this dissertation is given.

Finally, this thesis includes three appendices. Appendix A shows the reference list of the internal

research reports that I wrote during the doctoral studies, Appendix B shows the list of conference and journal papers published during my doctoral studies, and Appendix C contains the source code of some key Matlab blocks developed during the PhD program.

1. Problem Statement

The automotive industry is undergoing a profound transformation driven by the rapid adoption of electric propulsion systems and increasing levels of vehicle autonomy. Electric and autonomous vehicles introduce new challenges for control system design due to their strong nonlinear behavior, tight coupling between subsystems, and high sensitivity to external disturbances such as road conditions, tire–road interaction, and parameter uncertainties. In this context, guaranteeing robust performance and safety under a wide range of operating conditions remains a critical and unresolved problem.

A key difficulty in the development of control strategies for autonomous electric vehicles lies in the impossibility of validating all possible driving scenarios exclusively through real-world experimentation. Safety concerns, high economic costs, and the risk of damaging prototypes or endangering human lives make extensive on-road testing impractical during early development stages. Consequently, high-fidelity vehicle simulation becomes an essential tool, allowing the evaluation and verification of control algorithms under diverse and extreme scenarios that would be difficult or unsafe to reproduce in real traffic environments.

However, vehicle simulation itself presents significant challenges. Accurate simulation requires detailed mathematical models capable of capturing longitudinal and lateral dynamics, actuator behavior, tire–road interaction, and the effects of uncertainties and disturbances. Simplified models may lead to misleading conclusions, while overly complex models increase computational burden and reduce applicability for control design. Achieving a balance between model fidelity and practical usability is therefore a fundamental problem in the development of autonomous electric vehicle controllers.

Within this framework, conventional linear control approaches often exhibit limited robustness when confronted with model uncertainties, external disturbances, and nonlinear dynamics inherent to electric and autonomous vehicles. This limitation motivates the exploration of advanced

nonlinear control techniques. Sliding mode control emerges as a particularly attractive approach due to its inherent robustness properties, finite-time convergence, and capability to handle matched uncertainties and disturbances, which are common in automotive systems.

The main problem addressed in this research is the design, analysis, and validation of sliding mode control strategies applicable to electric and autonomous vehicles, ensuring robust trajectory tracking and dynamic stability while explicitly considering actuator dynamics and realistic vehicle behavior. The proposed controllers must be evaluated through simulation environments that reflect real driving conditions, enabling a systematic assessment of performance and robustness before any real-world deployment.

2. Theoretical Framework

This chapter is intended to give enough background about some mathematical material required for the robust control theory and the foundations of electrical machines. All of these, aiming to develop controllers for the lateral and longitudinal dynamics of a vehicle further in this dissertation.

2.1. State-Space Modeling

The analysis and design of control systems require a deep understanding of various physical and mathematical concepts. Although it is impossible to master all aspects of mathematics or physics, a careful selection of relevant topics can significantly aid in this endeavor. This chapter presents key concepts essential for the analysis and design of controllers for electric autonomous vehicles. However, this is not an exhaustive list of topics.

Many engineering programs, such as electrical engineering, offer extensive coursework in physics and mathematics. Nonetheless, the topics discussed here may extend beyond that scope, as control theory often involves a strong mathematical foundation. In fact, it leans more toward applied mathematics, making it highly relevant to mathematicians as well.

The author recommends studying mathematical proofs or real analysis through resources like [30] and [39]. These areas are among the first steps toward more advanced topics beyond the standard mathematics covered in engineering curricula. Although these subjects are more targeted at mathematicians, they provide substantial help in understanding the rigorous language used in the advanced control theory literature. In addition, it is advisable to explore topics on linear and nonlinear systems, as discussed in [35] and [3]. Finally, revisiting foundational mathematics and physics from college courses as needed is highly encouraged.

2.1.1 Introduction to the State-Space Representation

There are several methods for the analysis and design of control systems. The classical method is called design in the frequency domain and consists in transform the differential equations to a transfer function using the Laplace transform, which is an algebraic form. This method has the disadvantage that it only works for single-input single-output (SISO) systems, and they can only be linear and time-invariant.

A second method is called design in the state-space or time-domain, which has the advantage that can represent non-linear, time-variant systems and multiple-input multiple-output (MIMO) systems as well.

Other methods we can mention are the methods that imply a learning method covered by usage of machine learning methodologies such as neural networks (NN), recurrent neural networks (RNN) etc. or fuzzy control methods that allow us to handle uncertainty and imprecision using linguistic rules (like if-then statements) to make decisions in complex systems.

In this section, we will cover briefly the second method mentioned above, which uses the modeling of the physical system in the state-space.

To continue with the state-space representation, let us introduce some definitions.

- **State.** The state of the dynamic system is the knowledge of the values of a minimum set of variables in a certain instant for $t \geq 0$, knowing the initial value at $t = 0$ of these variables.
- **State Variables.** State variable is each variable within the set of the minimum set of variables that describe the *state* of the system. Let us suppose the set of variables $x_1, x_2, x_3...x_n$ is the set that describes the system, which initial conditions are $x_1(0), x_2(0), x_3(0)...x_n(0)$, then, each variable in that set is a state variable. Not all of them necessarily should have a physical meaning or be measurable.
- **State Vector.** The n variables can be considered as the n components of a vector.
- **State-Space.** The n-dimensional space which coordinates axes are each of the state variables.
- **State-Space Equations.** These are the equations that describe the system that are defined in

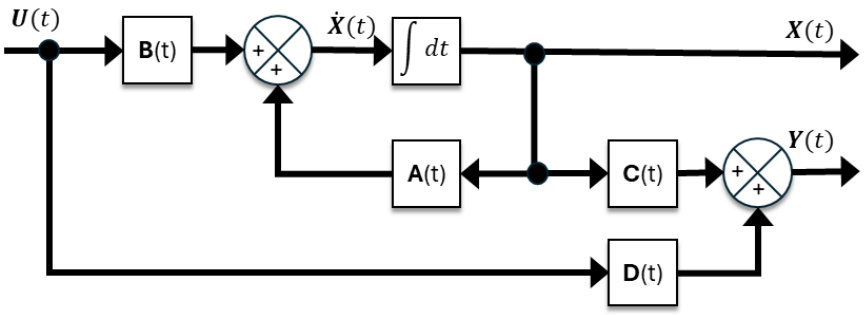


Figure 2.1: Block diagram of the linear time-varying system. For time-Invariant the matrix **A,B,C,D** do not depend on time t.

terms of input variables, output variables and state variables.

A MIMO system of continuous-time and finite-dimensional equations can be described as follows:

$$\dot{\boldsymbol{X}} = \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{U}, t), \tag{2.1}$$

$$\boldsymbol{Y} = \boldsymbol{g}(\boldsymbol{X}, \boldsymbol{U}, t), \tag{2.2}$$

where $\boldsymbol{U} \in \mathbb{R}^r$ is the vector of inputs of the system, $\boldsymbol{Y} \in \mathbb{R}^m$ is the vector of outputs of the system, $\boldsymbol{X} \in \mathbb{R}^n$ is the vector of state variables.

Having linearized the system, it can be described in matrix form as

$$\dot{\boldsymbol{X}} = \boldsymbol{A}(t)\boldsymbol{X}(t) + \boldsymbol{B}(t)\boldsymbol{U}(t), \tag{2.3}$$

$$\boldsymbol{Y} = \boldsymbol{C}(t)\boldsymbol{X}(t) + \boldsymbol{D}(t)\boldsymbol{U}(t), \tag{2.4}$$

which is a linear, time-varying system. Fig. 2.1 shows a block diagram of this system. For linear time-invariant (LTI) system, the time should not appear explicit in the vector functions **f** and **g**, and then the system is expressed as

$$\dot{\boldsymbol{X}} = \boldsymbol{A}\boldsymbol{X}(t) + \boldsymbol{B}\boldsymbol{U}(t), \tag{2.5}$$

$$\boldsymbol{Y} = \boldsymbol{C}\boldsymbol{X}(t) + \boldsymbol{D}\boldsymbol{U}(t). \tag{2.6}$$

As example, let us write the equation of the RLC circuit in Fig. 2.2 using the Kirchhoff's voltage law, so we have

$$v(t) = v_R(t) + v_L(t) + v_C(t), \tag{2.7}$$

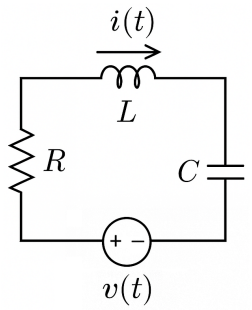


Figure 2.2: RLC circuit.

where $v_R(t), v_L(t), v_C(t)$ are the voltage drop on each element and using the next constitutive equations

$$v_R(t) = Ri(t), \tag{2.8}$$

$$v_L(t) = L\frac{di(t)}{dt}, \tag{2.9}$$

$$v_C(t) = \frac{1}{C}\int i(t)dt, \tag{2.10}$$

$$i(t) = \frac{dq}{dt}, \tag{2.11}$$

then, the differential equation in the mesh is

$$v(t) = Ri(t) + L\frac{di(t)}{dt} + \frac{1}{C}\int i(t)dt; \tag{2.12}$$

choosing i and q as our state variables, and using the relation in (2.11), since it is a second order system, we can convert the system to

$$L\frac{d^2q}{dt^2} + R\frac{dq}{dt} + \frac{1}{C}q = v(t). \tag{2.13}$$

Now, this second order differential equation can be converted to a two first order differential equation system

$$\frac{dx_i}{dt} = a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n + b_i, \tag{2.14}$$

where a_ij and b_i are constants for a LTI system. Then we have

$$\begin{aligned} \frac{dq}{dt} &= i, \\ \frac{di}{dt} &= -\frac{R}{L}i - \frac{1}{LC}q + \frac{1}{L}v(t). \end{aligned} \tag{2.15}$$

If we define $\boldsymbol{X} = [i \;\; q]^T$ and $\boldsymbol{U} = v(t)$, $\boldsymbol{B} = [0 \;\; \frac{1}{L}]^T$ and

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{L} & -\frac{1}{LC} \end{bmatrix}, \tag{2.16}$$

our system in (2.15) takes the form

$$\dot{\boldsymbol{X}} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{B}\boldsymbol{U}, \tag{2.17}$$

which is a linear time-invariant system, and many linear techniques can now be applied to it.

2.1.2 Linear Systems

Having a system described as in the section above, that is, linear systems, linear techniques can be applied. In this section we show some of these techniques which were used in the control design in next chapters.

2.1.2.1 QR Decomposition

QR decomposition is a method in linear algebra where a matrix is broken down into two components: one with orthogonal (perpendicular) vectors and another that is triangular in structure. It is useful for solving systems of linear equations, minimizing errors in overdetermined systems, that is a system of equations that has more equations than unknowns, and finding eigenvalues of matrices. This method is valued for its numerical stability and is widely applied in fields such as data analysis, engineering, and machine learning.

As described in [3], let us suppose we have the system $Ax = b$, where $A \in R^{m,n}$ and the rank of A is n were $n \leq m$ then

$$A = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = \begin{bmatrix} Q_1, & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R, \tag{2.18}$$

where Q is an orthogonal matrix, that is $(Q^T = Q^{-1})$ and $R \in R^{n,n}$ is an upper triangular matrix of full rank n. In Algorithm 1 can be seen how to get this decomposition using a live script in Matlab. After decomposition, the resultant matrix is divided in four sub-matrix to get the regular form.

Now, the QR decomposition can be used to find solutions for equations of the form $Ax = b$. First, determine the QR decomposition of A. Then

$$\begin{aligned} &Q^T Ax = Q^T b, \text{ or} \\ &\begin{bmatrix} R \\ 0 \end{bmatrix} x = Q^T b \text{ since } Q^T = Q^{-1}, \text{ or} \\ &Rx = c, \end{aligned}$$

and we just need to solve the system where R is triangular and

$$c = \begin{bmatrix} I_n, & 0 \end{bmatrix} Q^T b.$$

2.1.2.2 Controller Forms

The *controller form* (regular form, or canonical controllable form) is a specific representation of a linear system in state-space that highlights its controllability properties. In this form, the system matrices A, B, C, and D are structured so that the controllability of the system is immediately evident. Specifically, the A-matrix has a companion form where its last row contains the coefficients of the system's characteristic polynomial in reverse order, and the B-matrix is a column vector with zeros except for the last element, which is 1, in the case of single input u. This structure is particularly useful for designing state feedback controllers. It is possible to convert to the controller form even for multiple inputs, that is $u \in R^m$. Let us see the procedure to convert to the regular form according [76]: let us assume we have a system with the form

$$\dot{x} = Ax + Bu, \tag{2.19}$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, and $D \in \mathbb{R}^{p \times m}$, and let (A, B) be controllable. Assume $Rank(B) = m$, there is an orthogonal matrix such that

$$T_r B = \begin{bmatrix} 0 \\ B_2 \end{bmatrix}, \tag{2.20}$$

as we saw in section 2.1.2.1. where $B_2 \in R^{m,m}$ and it is non singular. This matrix can be obtained through the QR factorization was explained in the section 2.1.2.1 using Matlab code as in Algorithm 1.

Now, the system 2.19 will be transform to a new coordinates

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = T_r x, \tag{2.21}$$

where z_1, z_2 partition the coordinates such that $z_1 \in R^{n-m}$ and $z_2 \in R^m$. The matrices of the system 2.19 become $A \Leftrightarrow T_r A {T_r}^T$ and $B \Leftrightarrow T_r B$ in the z coordinates in such a way that we can write the system now as:

$$\dot{z}_1(t) = A_{11} z_1(t) + A_{12} z_2(t), \tag{2.22}$$

$$\dot{z}_2(t) = A_{21} z_1(t) + A_{22} z_2(t) + B_2 u(t), \tag{2.23}$$

where

$$T_r A {T_r}^T = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}. \tag{2.24}$$

This new representation is the one called regular form or controller form. In this form, the terms that depend on the inputs will appear in (2.23).

Now, from here, in the context of the sliding modes, we can use this structure to find the switching surface through the methods of eigenvalue placement or linear-quadratic minimization. These methods are explained in [76] and an example of this is provided in the Algorithm 2 and 3.

2.2. Dynamic Systems Stability

In system theory, the total response of a system is the sum of the free response of that system and the forced response, that is

$$\text{Total response} = \text{free response} + \text{forced response}. \tag{2.25}$$

The free response is the part of the system output that comes from the initial conditions such as the energy stored from capacitor and inductors, and represents the system behavior when no external input is applied. Mathematically, it is the homogeneous solution of the differential equations.

The forced response is the part of the output response caused by the application of an external

input. Mathematically, it is associated with the particular solution of the differential equations.

An LTI system is stable if the free response decays to 0 as time approaches infinity. An LTI system is unstable if the free response increases unlimited when time approaches infinity. And finally, an LTI system is marginally stable if neither diverges nor converges but remains bounded and oscillatory.

Another definition of stability can be using the concept of bounded-input bounded-output (BIBO) where a system is considered stable if, for all bounded inputs, the system produces a bounded output. Conversely, the system is unstable if for any bounded input, the system produces an unbounded output.

There are several methods to verify stability of a system such as the Routh-Hurwitz method. However, for the purpose of this dissertation, we will use the Lyapunov definition for stability and the Lyapunov method to verify it in the next subsection.

2.2.1 Stability in the Lyapunov Sense

Here, we are going to provide a brief description of the stability and the Lyapunov method; however, a more detailed explanation can be found in [35] and [76] and other literature related to linear or non-linear systems.

Stability in the Lyapunov sense refers to the behavior of a system's solutions near an equilibrium point, characterized by using Lyapunov's methods. It provides a framework to determine whether the solutions stay close to the equilibrium over time. First, let us define what an equilibrium point is.

Equilibrium Point: A point x_e where the system's state does not change, i.e., $f(x_e) = 0$ for a dynamical system

$$\dot{x} = f(x), \tag{2.26}$$

where $f : D \to R^n$ is a locally Lipschitz map from the domain $D \subset R^n$ to R^n. We assume that x and f are dependent on time as well.

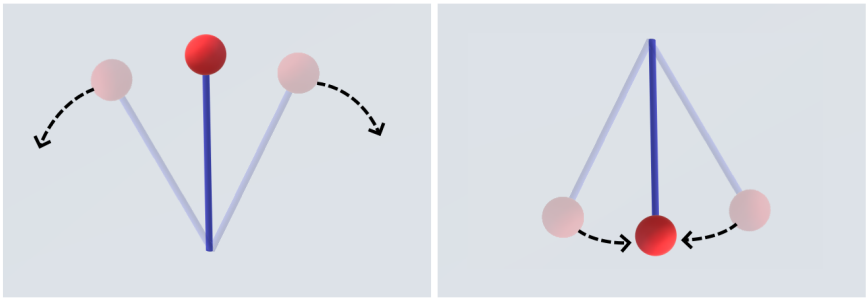


Figure 2.3: Diagram showing a pendulum which has two equilibrium points. a) Unstable equilibrium point. b) Stable equilibrium point.

As example, we can have a pendulum as shown in Fig. 2.3. which has a differential equation as

$$\dot{x}_1 = x_2, \tag{2.27}$$
$$\dot{x}_2 = -a \sin x_1 - b x_2,$$

where x_1 would be the position angle and x_2 is the angle speed, b is the friction factor and a is the factor that relates the gravity ($g = 9.81 m/s^2$) and the length of the pendulum, that is g/L.

Intuitively, we can see two equilibrium points; one when the ball is in the position 0 ($x_1 = 0, x_2 = 0$) Fig. 2.3 right side (b), and the other in position π ($x_1 = \pi, x_2 = 0$), left side (a). Intuitively we can see that the left side is an unstable equilibrium point because any disturbance might cause the ball to fall away and never return (unless an external force acts on it). On the contrary, the right side shows the stable equilibrium point because despite some disturbances, the ball will return to the equilibrium point.

This can be more formally defined:

Stable equilibrium point: *The equilibrium point $x = 0$ is stable if there exists a $\delta = \delta(\epsilon) > 0$ for each $\epsilon > 0$ such that*

$$||x(0)|| < \delta \Rightarrow ||x(t)|| < \epsilon, \forall t \geq 0.$$

We interpret that as if we start close to the equilibrium point, the system solution will remain close to it.

Unstable equilibrium point: If the system does not satisfy the above stability criteria.

Asymptotically stable equilibrium point: if it is stable, and besides δ can be such that

$$||x(0)|| < \delta \Rightarrow \lim_{t\to\infty} x(t) = 0,$$

that means, that besides to keep close to the equilibrium point, the trajectory will always move towards the equilibrium point.

In 1892, Aleksandr Lyapunov, a Russian mathematician and physicist, demonstrated that certain functions can be used to determine the stability of an equilibrium point. Let us call this function V which normally is called a proposed Lyapunov function and it is defined as $V : D \to R$ which is continuously differentiable in $D \subset R^n$ and contains the origin.

Now, the derivative of this function V along the trajectories of the system Eq. 2.26 is

$$\begin{aligned}
\dot{V}(x) &= \sum_{i=1}^{n} \frac{\partial V}{\partial x_i}\dot{x}_i, && (2.28)\\
&= \sum_{i=1}^{n} \frac{\partial V}{\partial x_i} f_i(x), \\
&= \begin{bmatrix} \frac{\partial V}{\partial x_1}, & \frac{\partial V}{\partial x_2}, & \cdots, & \frac{\partial V}{\partial x_n} \end{bmatrix} \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix}, \\
&= \frac{\partial V}{\partial x} f(x). && (2.29)
\end{aligned}$$

Hence, we can see from (2.28) that the derivative of the function V along the system $f(x)$ is dependent on the equations of the same system.

Before to mention the Levant differentiator, let us define what a positive definite function is.

Positive definite function: A function $V : D \to R$ is positive definite in the domain D if

1. $V(x) \geq 0$, $\forall x \in D$ and
2. $V(x) = 0 \iff x = 0$.

We can interpret this as the function V is always positive in the domain D and that $V(x) = 0$

implies that $x = 0$. As example of this kind of functions we can have the function $V : R^2 \to R$ given by $V(x_1, x_2) = {x_1}^2 + {x_2}^2$.

Semi-positive definite function: is a function where only the first condition holds.

Now we can define the Lyapunov stability theorems.

Local Lyapunov stability theorem:

Theorem 1. *Consider the system (2.26) and consider that in the domain D there exists a differentiable positive definite function $V : D \to R$ such that*

$$\dot{V}(x) = \frac{\partial V}{\partial x} f(x) \leq 0, \tag{2.30}$$

then, the origin is stable. Furthermore, if $\dot{V} < 0 \, \forall \, x \neq 0$ then the origin is asymptotically stable.

Global Lyapunov stability theorem:

Theorem 2. *Consider the system (2.26) and consider that in the domain D there exists a differentiable positive definite function which is "radially unbounded" $V : D \to R$ such that*

$$\dot{V}(x) = \frac{\partial V}{\partial x} f(x) \leq 0, \tag{2.31}$$

then, the origin is "globally" stable. Furthermore, if $\dot{V} < 0 \, \forall \, x \neq 0$ then the origin is asymptotically stable.

We understand as *globally unbounded* as the property of a function $V(x)$ such that $V(x) \to \infty$ as $||x|| \to \infty$.

An important note to consider is that the *Lyapunov stability conditions are only sufficient* [35]. This means that failing to find a candidate Lyapunov function $V(x)$ positive definite and whose derivative $\dot{V}(x)$ is negative definite, does not mean that the system is unstable.

Example:

Consider the pendulum equation as (2.27) without friction, that is $b = 0$, namely,

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -a \sin x_1, \end{aligned} \tag{2.32}$$

and now, we will analyze the stability of the equilibrium point at the origin. A natural Lyapunov function candidate is the energy function of a pendulum neglecting the friction

$$V(x) = a(1 - \cos x_1) + \frac{1}{2}x_2^2. \tag{2.33}$$

Clearly, $V(0) = 0$ and $V(x)$ is positive definite over the domain $-2\pi < x_1 < 2\pi$ and $x_2 \in R$.

The derivative of $V(x)$ along the trajectories of the system is given by

$$\dot{V}(x) = a\dot{x}_1 \sin x_1 + x_2\dot{x}_2 = ax_2 \sin x_1 - ax_2 \sin x_1 = 0. \tag{2.34}$$

Thus, conditions of the Lyapunov stability are satisfied because $V(x)$ is positive definite and $\dot{V}(x) \leq 0$, and we conclude that the origin is stable. Since $\dot{V}(x) \equiv 0$, we can also conclude that the origin is not asymptotically stable but it will stay close to the stability point.

2.3. Dynamic Subsystems of an Automotive Vehicle

In this section, we will briefly mention some subsystems to give an overview of the complexity of modern vehicles. In further chapters, especially in chapter 3 we cover in depth the physics and mathematics involved in some systems aiming to simulate and design control algorithms for automotive applications.

2.3.1 Powertrain Subsystem

The powertrain refers to the systems involved in generating and transmitting the power from the source in the vehicle to the wheels that drive the vehicle forward. We can see in Fig. 2.4 we can divide the powertrain into different components. There are some differences in the case for EVs and combustion vehicles.

→ **Energy Source:** In the case of conventional vehicles, it can be gasoline, natural gas, diesel etc. For the case of electrical vehicles, a battery is the power source and, it is a complete different module and at the moment of writing this, it is one of the most critical elements in an EV since the autonomy of the vehicle depends on it; the speed of charging is another critical topic;

the safety on the battery is another key point where, for example, it should avoid the explosion under any circumstances etc. The industry is focusing the efforts in the improvement of the batteries on these and many other aspects.

→ **SI Engine:** The spark ignite engine is the conventional motor which is now challenged by the propulsion of electric machines. The SI engine is a complex mechanical system of high precision. Some components we can mention are the **cylinder block**, **the pistons** that moves up and down compressing the air-fuel mixture and transmit the force to the **crankshaft**. It requires a **cooling system** that runs throughout the engine block that has water jackets. It requires a **lubrication system** to reduce the friction of the moving parts. Other key parts are the **air intake system** and the **exhaust system**. The SI engine is considered one of the greatest engineering demonstrations of humankind.

→ **Electric Machine:** For the EVs, this is the component that generates the power for traction (ignoring the hybrid vehicles). Typical motors used for this purpose are induction motors (IM), used by Tesla, permanent magnet synchronous motors (PMSM) used by Nissan, BMW, brush-less motors (BLDC), switch reluctance motor (SRM), and recently in experimental designs axial flux motors are being applied.

→ **Transmission:** The SI engine doesn't have a continuous torque along the motor speed. There is a specific speed where the torque is maximum; above or below that point the torque decreases. The transmission keeps the relation between the wheel speed and motor RPMs in a point that allows the motor to keep working properly.

→ **Transmission Shaft and differential:** The transmission shaft transmit the rotational power from the engine to other components like the differential or the wheels, while the differential is in charge to compensate the speed of the inner or outer wheels when cornering.

→ **Gear Box:** Used in EVs, this is similar to the transmission, but it is simpler since the electrical motors can deliver practically all the torque throughout all the speed range.

→ **Tires:** It is the interface between the car and the ground. At the end, the force generated by the rubber on the ground is the force shoving the vehicle.

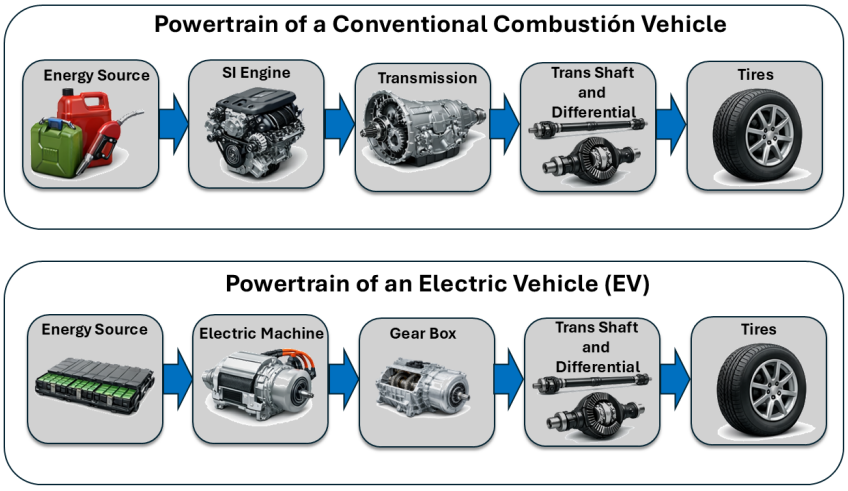


Figure 2.4: Powertrain blocks diagram for electric vehicles and combustion vehicles.

2.3.2 Chassis Subsystem

The chassis is the backbone of the vehicle. It is the structure that supports and connects all the major components. The next list shows the main components that the chassis subsystem supports:

→ **Suspension:** Springs, dampers, control arms, stabilizer bars that provide comfort and stability.

→ **Steering:** It is what allows the directional control of the vehicle. In a conventional vehicle it is composed of the **steering wheel**, **column**, **rack and pinion**, **steering box** and linkages. However, modern vehicles implement electrical power steering (EPS) which uses electrical motors to provide direction and feedback to the driver.

→ **Brakes:** The brakes are the components that decelerates the vehicle when needed. Some components in the brakes can be the **disk/drum brakes**, **calipers**, brake lines, **master cylinder**, anti-lock braking (**ABS**) modules.

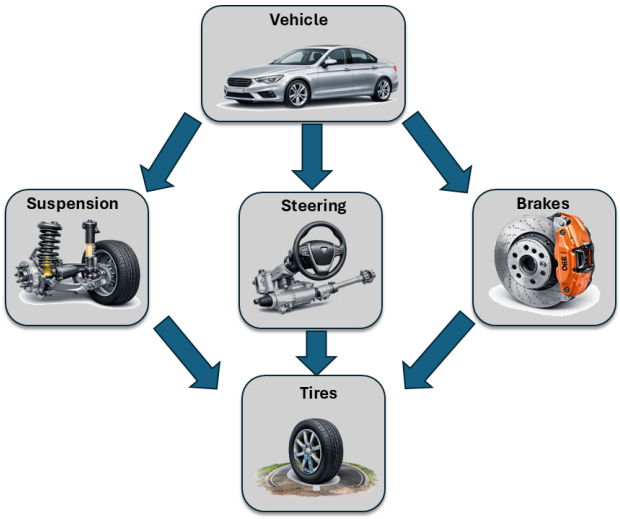


Figure 2.5: Chassis subsystems

2.3.3 Electrical and Electronic Control Subsystems

Automotive vehicles have become more complex each time. Many functionalities are added year to year; from safety and security to comfort and connectivity. Traditional vehicles can have between 10 and 150 electronic control units (ECU) which require a complex network for power and communications, with the challenges it brings, for example, long wiring and many connectors, which make the construction of vehicles more difficult and more prone to quality defects. Automakers have evolved the architectures to cope with these complexities.

Currently, there are two approaches for the automotive electric and electronic architecture at vehicle level. These are the **zonal and domain architecture**. The domain architecture is the more traditional approach which consists in grouping the electrical or electronic functionalities by logical domains or functions. Below is a list of traditional ECUs with the corresponding example of domain:

Powertrain and Engine:

→ **Engine Control Module (ECM)** fuel injection, ignition, emissions.
→ **Powertrain Control Module (PCM)** integrates ECM + TCM.
→ **Transmission Control Module (TCM)** gear shifting optimization.
→ **Turbocharger Control Module** manages boost pressure.

→ **Fuel Pump Control Module** regulates fuel delivery.

→ **Cooling Fan Control Module** thermal management.

Safety and Chassis:

→ **Brake Control Module (EBCM/ABS ECU)** ABS, traction, stability.

→ **Airbag Control Module (ACM)** crash detection and deployment.

→ **Suspension Control Module (SCM)** adaptive damping, ride height.

→ **Steering Control Module (EPS ECU)** electronic power steering.

→ **Adaptive Cruise Control ECU** radar-based speed and distance control.

→ **Collision Avoidance ECU** emergency braking, lane departure.

Body and Comfort:

→ **Body Control Module (BCM)** lights, locks, windows.

→ **Central Control Module (CCM)** coordinates body electronics.

→ **Central Timing Module (CTM)** timing for lights, wipers.

→ **Seat Control Module** heating, positioning, memory.

→ **Door Control Module** locks, windows, mirrors.

→ **HVAC Control Module** heating, cooling, ventilation.

Infotainment and Connectivity:

→ **Infotainment ECU** audio, navigation, user interface.

→ **Telematics Control Unit (TCU)** vehicle-to-cloud communication.

→ **Instrument Cluster ECU** dashboard displays.

→ **Head-Up Display ECU** projection systems.

→ **Antenna Control Module** radio, GPS, connectivity.

Electrification and EV-Specific:

→ **Battery Management System (BMS)** monitors cell health.

→ **Inverter Control Module** manages DC/AC conversion.

→ **Onboard Charger ECU** charging operations.

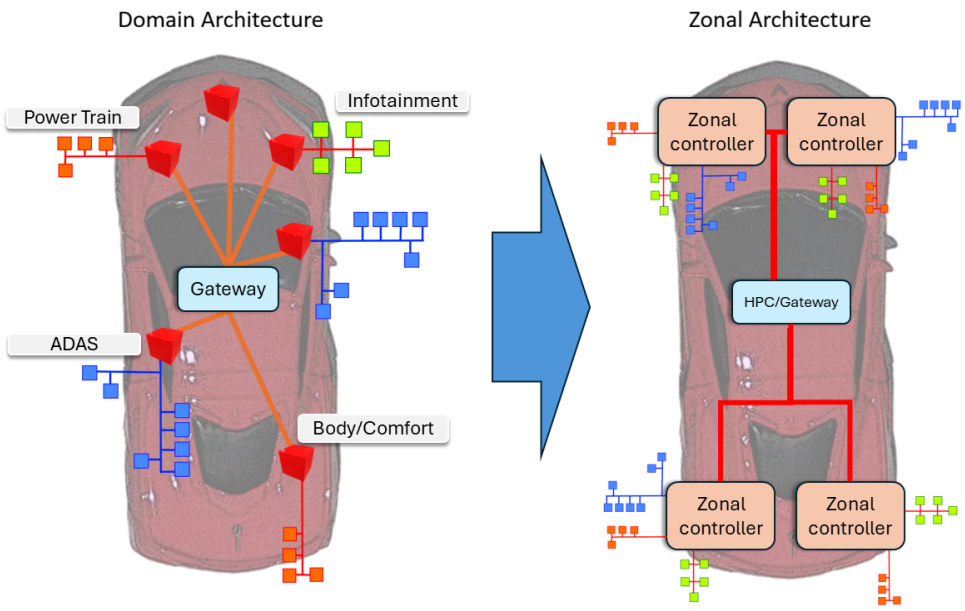


Figure 2.6: Domain vs Zonal architectures.

→ **DC/DC Converter ECU** voltage regulation.

→ **Thermal Management ECU** battery cooling/heating.

This architecture is an advancement compared to the old flat wiring; however, this architecture is being taken to its limits and more scalability and flexibility is required, especially since the arriving of the concept of the software define vehicle (SDV). The **software defined vehicle** is a modern automobile in which most functions and features are controlled, updated, and enhanced through software, rather than being permanently tied to fixed hardware. It is formed by centralized computing platforms where a powerful processor replaces many controllers; functions are separated into layers, enabling updates without hardware changes; new features, bug fixes, and performance improvements are delivered remotely through updates Over-The-Air (OTA); integration with cloud services, IoT, and digital ecosystems are taken in account from the design.

The new approach of the industry is called **zonal architecture**, which is intended to mitigate the problems of the classical domain architecture and increase the scalability. In this approach, the functions are organized by zones, which typically correspond to physical areas within the vehicle such as the front dashboard, passenger cabin, front-end, rear-end etc. Fig. 2.6 shows the difference between these two approaches.

2.3.4 Human–Machine Interface Subsystem

The Human–Machine Interface (HMI) subsystem encompasses all the technologies and components that enable communication between the driver, passengers, and the vehicle's electronic systems. Its purpose is to provide intuitive, safe, and efficient interaction, ensuring that complex vehicle functions remain accessible and comprehensible to the user.

- → **Instrument Cluster:** Digital or hybrid displays that present critical information such as speed, fuel level, battery status, and warning indicators. Modern clusters often integrate reconfigurable layouts and augmented reality overlays.
- → **Infotainment System:** Centralized multimedia and connectivity hub, including navigation, entertainment, smartphone integration, and voice assistants. It serves as the primary interface for non-critical vehicle functions.
- → **Head-Up Display (HUD):** Projects essential driving information onto the windshield or a dedicated combiner, reducing driver distraction by keeping eyes on the road.
- → **Touch and Haptic Interfaces:** Touchscreens, capacitive buttons, and haptic feedback systems that provide tactile confirmation of user inputs, improving usability in dynamic driving conditions.
- → **Voice and Gesture Recognition:** Natural language processing and sensor-based gesture controls that allow hands-free operation of vehicle functions, enhancing safety and accessibility.
- → **Driver Monitoring Systems (DMS):** Cameras and sensors that track driver attention, fatigue, and engagement, forming part of advanced safety and ADAS integration.

In modern vehicles, the HMI subsystem is tightly coupled with the *Environment Sensing* and *ADAS* subsystems, creating a holistic interaction loop. The driver receives processed information from external sensors and vehicle controllers through the HMI, while inputs from the driver are transmitted back into the control architecture. This bidirectional flow is fundamental for the transition toward **Software-Defined Vehicles**, where user experience and adaptability are primarily driven by software updates and interface evolution.

2.3.5 Environmental interaction subsystem

Traditionally, we could talk about the aerodynamics and tires as the only factors that are in contact with the exterior world, and any of those are already complex. However, in modern cars this is too much more complex. Now we have like lane departure systems, advance driving assistance, adaptive cruise controls, attention detection systems... etc. that require a more complex interaction with the surroundings. Let us list some of the subsystems that we could mention:

→ **Vehicle Aerodynamics:** Involves the interaction of the car body with airflow, influencing drag, lift, stability, and fuel efficiency. Modern systems integrate active aerodynamic components such as adjustable spoilers, grille shutters, and underbody diffusers to optimize performance dynamically.

→ **Tires:** Serve as the only physical contact with the road surface, transmitting forces for traction, braking, and cornering. Advanced tire technologies include pressure monitoring systems (TPMS), run-flat designs, and compounds engineered for specific weather or performance conditions, all of which directly affect safety and efficiency.

→ **Environment Sensing:** Encompasses the suite of sensors and cameras that allow the vehicle to perceive its surroundings. This includes radar, LIDAR, ultrasonic sensors, and vision systems used for lane departure warnings, adaptive cruise control, collision avoidance, pedestrian detection, and driver attention monitoring. These subsystems are foundational for ADAS and autonomous driving capabilities.

2.4. Actuators in Automotive Electric Vehicles

Due to the global pressure to respond to the climate change in the short coming, most countries in the world have agreed to reduce its greenhouse gas (GHG) emissions, where 23 percent of the current GHG emissions in the world is caused by transportation [85]. As part of this agreement, it was stated that at least 20 percent of all road transportation globally is to be electrically driven by 2030 [85]. These changes force the companies to take strong decisions to accomplish the expectations of the political and social agendas aiming to reduce and/or control the climate change. Such is the case

of the spin-off Vitesco Technologies that was the division of powertrain of Continental AG. The restructuring to separate the Powertrain unit from continental's automotive division was a response to increase the company's flexibility to make acquisitions in the tech space and increase its exposure to vehicle electrification and autonomous driving [26] making the innovation in electric machines a very relevant and valuable topic for research in the next years.

There are several types of electric vehicles (EV) whose terminology depends on the way the energy is provided; these types are listed in [56] as follows: a) Battery electric vehicle (BEV). EVs with only batteries to provide power to the drive train. b) Hybrid electric vehicle (HEV). It requires both internal combustion engine (ICE) and electrical power to power the vehicle. c) Plug-in hybrid electric vehicle (PHEV). The difference against HEV is that it uses electric propulsion as main driving force. d) Fuel cell electric vehicle (FCEV). These use fuel cells that uses chemical reactions to produce electricity. Most of them use hydrogen as fuel. The most environmentally beneficial type of electric vehicle is the Battery Electric Vehicle (BEV). Notable brands leading in this category include Nissan, with its Leaf model, Tesla, and the Chinese brand BYD. Therefore, it is worthwhile to conduct research focused on these brands to investigate the characteristics and requirements of the motors used in their BEVs. Although the models analyzed here are academic motor models, they provide valuable insights into the behavior and performance of these motors. The electric motor is one of the core components of any EV and in this chapter, we analyze and simulate some of them trying to accomplish the requirements needed for some models of EVs, and in this section we gather those requirements. In Table 2.1, taken in part from [56], is shown the advantages and disadvantages of electric motors used in some vehicle brands. At the moment of writing this, the trend is to use permanent magnet synchronous motor (PMSM) [27] like the Chinese model Xpeng P7 [96] and Tesla model 3 uses a new technology that they call permanent magnet synchronous reluctance motors (PMSRM) [5]. Let us talk about the Tesla model S. It is a rear drive vehicle, and this drive is composed of one Induction motor, an inverter, a fixed gear box, and a differential like Fig. 2.7 b) . Fig. 2.7 c) shows the Tesla model S all-wheel drive (AWD) configuration. Possibly the patent used for the assembly of the motor in the car, is the one of number US 2015/0083505 A1[67]; In Fig. 2.8 b) is the block diagram of this patent and in Fig. 2.8 a) is the assembly diagram of the motor, inverter, transmission and differential. In Table 2.2 and 2.3 are the requirements for

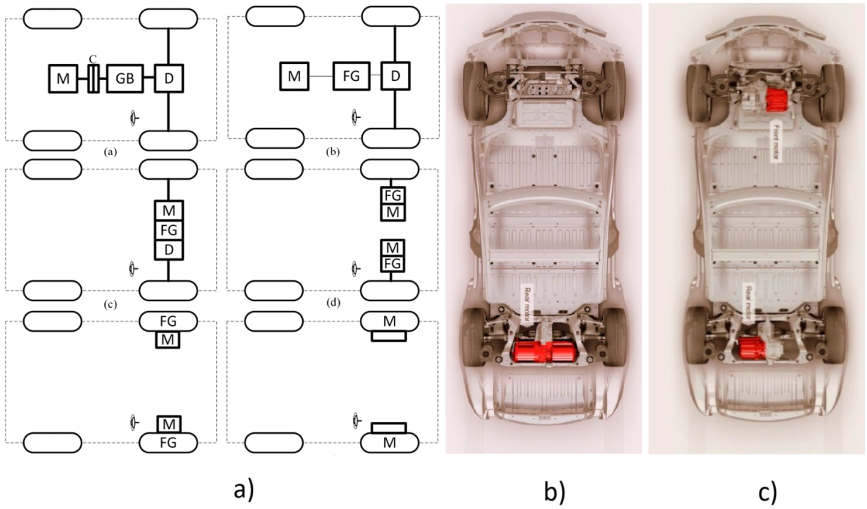


Figure 2.7: a) shows the different configurations for electric vehicles with front traction C: clutch; D: differential; FG: fixed gearing; GB: gear box; M: electric motor. b) Tesla model S rear drive. C) Tesla model S all-wheel drive configuration. Taken from [56].

the electric motors for different vehicle models. These are taken from [27], where more parameters can be found for more vehicle models. Besides the configuration depicted by the tesla vehicles on the motor and gears, there are several more configurations that are common today such as the ones depicted in Fig. 2.7 a).

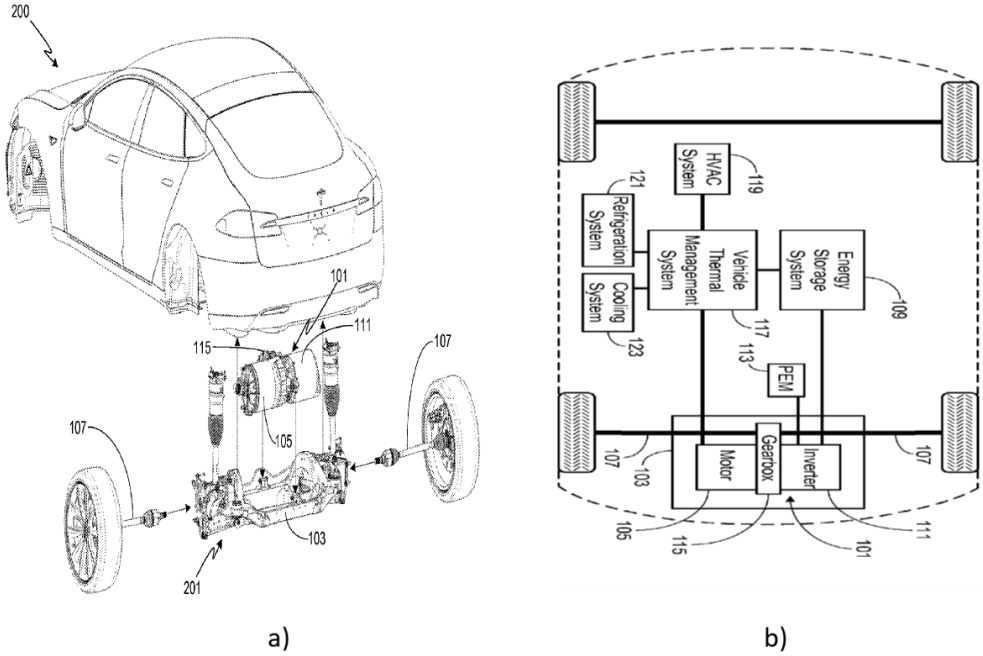


Figure 2.8: a) Diagram of assembly of the motor and inverter into the Tesla model S. b) Block diagram of the integrated electric motor assembly. Taken from [67].

TABLE 2.1. ADVANTAGES, DISADVANTAGES AND USAGE OF DIFFERENT MOTOR TYPES

Motor Type	**Advantage**	**Disadvantage**	**Vehicles Used In**
Brushed DC Motor	• Maximum torque at low speed	• Bulky structure • Low efficiency • Heat generation at brushes	Fiat Panda Electra (Series DC motor), Conceptor G-Van (Separately excited DC motor)
Permanent Magnet Brushless DC Motor (BLDC)	• No rotor copper loss • More efficient than induction motors • Lighter • Smaller • Better heat dissipation • More reliability • More torque density • More specific power	• Short constant power range • Decreased torque with increase in speed • High cost because of PM	Toyota Prius (2005)
Permanent Magnet Synchronous Motor (PMSM)	• Operable in different speed ranges without using gear systems • Efficient • Compact • Suitable for in-wheel application • High torque even at very low speeds	• Huge iron loss at high speeds during in-wheel operation	Toyota Prius, Nissan Leaf, Soul EV, Xpeng P7
Induction Motor (IM)	• The most mature commutatorless motor drive system • Can be operated like a separately excited DC motor by employing field orientation control	• None listed	Tesla Model S, Tesla Model X, Toyota RAV4, GM EV1

TABLE 2.2. ELECTRIC MOTOR DATA

Model	**EM Motor Type**	**EM Max. Torque (Nm)**	**EM Max. Power (kW)**	**EM Speed at Max Torque (rpm)**	**EM Speed at Max Power (rpm)**	**EM Max. Speed (rpm)**	**Trans Ratio**
Model S 60	IM Cu.	430	283	5000	5k-8k	16k	9.73
Model S 85D	2 IM Cu.	658	F,R 193	5.8k		16k	9.73
Model S P85D	3 IM Cu.	967	F: 193, R:375	5.1k (170π rad/s)	5k-8.6k	16k	9.73
Leaf'15	IPMSM	254	80	3k	3k-10k	10.5k	8.1938
e-Golf	PMSM	270	85	3k	3k	12k	9.747

TABLE 2.3. ELECTRIC MOTOR DATA CONTINUATION

Model	**Battery Voltage (V)**	**Top Speed (km/h)**	**Acc. Time 0-100 km/h (s)**	**Acc. Time 0-60 km/h (s)**	**Range NEDC (km)**
Model S 60	400	190	6.2		335
Model S 85D	400	249	4.4	4.2	435
Model S P85D		249	3.3	3.1	407
Leaf '15	360	140	11.5		135
e-Golf	323	140	10.4		134

2.4.1 DC Motor

DC motors use to have permanent magnets in the stator; rotors have brushes to provide energy to the rotor. The advantages of these motors are the ability to provide maximum torque in low speed. The disadvantage is its bulky structure, low efficiency, heat generated due to brushes and associated drop in efficiency [56] besides the increased maintenance required due to the brushes against other constructions of electrical motors (EM). The following equations (2.42-2.45) describe the behavior of a DC motor. A more detailed explanation of it, is given in [8]. Looking in Fig. 2.9 we can use the Kirchhoff's voltage law to obtain the equation that describe the electrical characteristics:

$$V_a - V_{La} - V_{Ra} - V_{em} = 0, \tag{2.35}$$

with

$$V_{La} = L_a \frac{di_a(t)}{dt}, \tag{2.36}$$

$$V_{Ra} = R_a i_a(t), \tag{2.37}$$

$$V_{em} = e_m(t). \tag{2.38}$$

Where V_a is the input voltage to the armature, and $e_m\,(t)$ is the so called back electromotive force (back EMF) and it is produced as a voltage induced, since the armature is a coil rotating in a magnetic field. This force is opposed to the rotating movement, and it is proportional to the rate or rotation of the armature and the flux linked by the coil $K_b\omega\,(t)$.

Now, the mechanical characteristics are described as follows taking into account the torque balance (energy balance):

$$T(t) - T_a - T_b - T_L = 0, \tag{2.39}$$

where

$$T_a = J_m \frac{d\omega(t)}{dt}, \tag{2.40}$$

$$T_b = b_m \omega(t), \tag{2.41}$$

and we have that T_a is the torque due to the rotational acceleration of the rotor, T_b is the damping torque due to the friction of the rotor and the angular speed, T_L is the torque due to the load, and

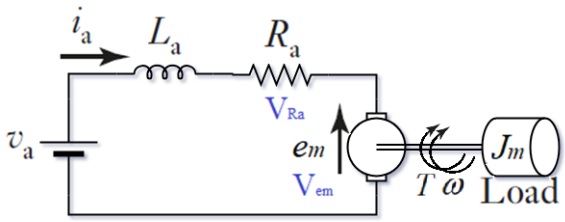


Figure 2.9: Equivalent circuit of the DC motor.

$T(t)$ is the electromagnetic torque. Combining (2.35) with (2.36), (2.37) and (2.38), and (2.39) with (2.40) and (2.41), and adding the relation between the torque $T(t)$ and the current i_a, and the relation between the back EMF $e_m(t)$ and the angular speed $\omega(t)$, we have

$$V_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + e_m(t), \tag{2.42}$$

$$T(t) = J_m \frac{d\omega(t)}{dt} + b_m \omega(t) + T_L, \tag{2.43}$$

$$T(t) = K_T i_a(t), \tag{2.44}$$

$$e_m(t) = K_b \omega(t). \tag{2.45}$$

Fig. 2.9 shows the diagram of these equations where R_a is the equivalent resistance, L_a is the inductance of the coil, i_a is the current in the coil, e_m is the back electromotive force (Back EMF) denoted as V_b in [8], J_m is the moment of inertia or rotational inertia denoted as I in [8], b_m is a constant that relates the damping torque and it is proportional to ω, where ω is the motor shaft angular velocity. K_T, K_b are constants that relate the torque T and the back EMF, respectively.

The same set of equations but in the state-space would look like as follows. Having

$$X_1 = i_a, \tag{2.46}$$

$$X_2 = \omega, \tag{2.47}$$

as our state variables and

$$U_1 = V_a, \tag{2.48}$$

$$U_2 = T_L, \tag{2.49}$$

as the inputs. Replacing (2.44), (2.45) in (2.42), (2.43) we have

$$V_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + K_b \omega(t) \tag{2.50}$$

$$K_T i_a(t) = J_m \frac{d\omega(t)}{dt} + b_m \omega(t) + T_L. \tag{2.51}$$

Solving for i_a (X_1) and ω (X_2) we can write the equations as a first order differential equation system:

$$\begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \end{bmatrix} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K_b}{L_a} \\ \frac{K_T}{J_m} & -\frac{b_m}{J_m} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} & 0 \\ 0 & -\frac{1}{J_m} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}. \tag{2.52}$$

2.4.1.1 Parameter Estimation and Simulation of Direct-Current Electric Machines

Let us take from a catalog of DC motors a motor with similar features defined for the motors in Table 2.2 and 2.2. The catalog that provides the parameters needed for our models is [40], in this case we choose based on a similar torque than a tesla model S (430Nm).

Taking from the catalog in Fig. 2.11 let us approximate K_b, b_m, and K_T for our model from the highlighted row where Table 2.4 summarizes this selected row and then we calculate

$$K_T = \frac{T}{i_a} = \frac{430 \text{ Nm}}{198 \text{ A}} = 2.1717 \text{ Nm/A},$$

$$V_{em} = e_m = V_a - (R_a i_a) = 420 \text{ V} - 0.193\ \Omega \times 198 \text{ A} = 381 \text{ V},$$

$$K_b = \frac{e_m}{\omega} = \frac{381 \text{ V}}{52\pi \text{ rad/s}} = 2.332232 \text{ Vs},$$

$$b_m = \frac{T}{\omega} = \frac{430 \text{ Nm}}{52\pi \text{ rad/s}} = 2.632177 \text{ Nms}.$$

The Matlab code in Algorithm 5 implements the equations in the state-space form as (2.52).

Fig. 2.10 shows the result of the simulation. It shows ω, that is, the angular velocity of the shaft, and i_a, the current in the armature. We can observe the over-current when the motor is accelerating until reach a steady speed.

TABLE 2.4. DC MOTOR PARAMETER SELECTION FROM CATALOG

Symbol	Value	unit
i_a	198	A
V_a	420	V
L_a	3.83	mH
R_a	0.193	Ω
T	430	Nm
J_m	0.6	kg/m^2
ω	52π	rad/s

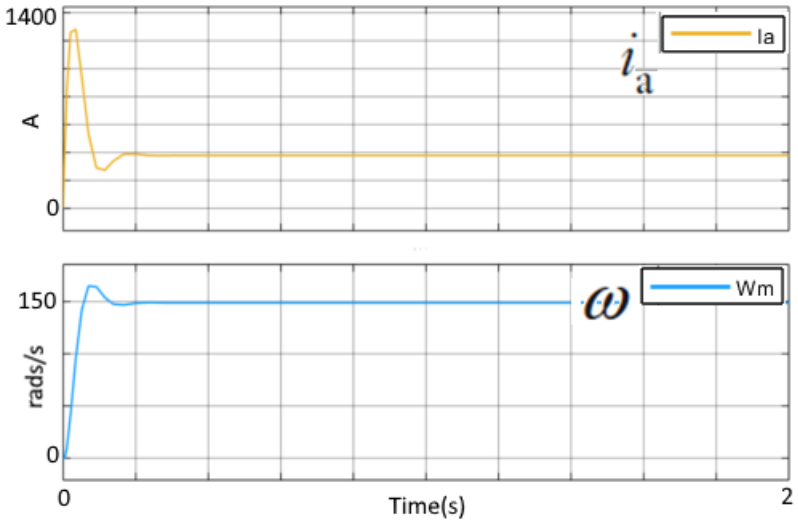


Figure 2.10: Simulation of the DC Motor.

LSK 1604 L - LSK 1604C L
D.C. motors
Electrical characteristics

E4 - Selection tables (IC 06)

The electrical characteristics are given for:
- 3-phase supply with full bridge
- degree of protection IP 23S
- cooling method IC 06 (FV)
- continuous S1 duty
- ambient temperature ≤ 40°C

Total weight: 375 kg
Moment of inertia: 0.6 kg.m²
Field power: 1.4 kW

379 - 508 N.m

$n_{max\ mech}$: 4000 min⁻¹

Key to abbreviations: see page 86.

P	Speed of rotation n for armature voltage U							$n_{max\ elec*}$		M	I	η	L	$R_{115°}$	U_{max}	Code	Deliv-ery
	260 V	400 V	420 V	440 V	460 V	500 V	600 V	N.C.	C.			Not inc.					
kW	min⁻¹	min⁻¹	min⁻¹	min⁻¹	min⁻¹	min⁻¹	min⁻¹	min⁻¹	min⁻¹	N.m	A	excitation	mH	Ω	V		
41.6	970							1530	2450	410	198	0.85	3.83	0.193	600		
66.2		1480						2180	3490	427	198	0.88	3.83	0.193	600		
70.3			1560					2290	3660	430	198	0.89	3.83	0.193	600		
74.4				1620				2390	3820	439	198	0.89	3.83	0.193	600	07	✶
76.5					1690			2490	3980	432	193.5	0.90	3.83	0.193	600		
80.2						1840		2650	4000	416	185.5	0.91	3.83	0.193	600		
94.5							2210	3020	4000	408	182	0.91	3.83	0.193	600		

Figure 2.11: Fragment of the motor selection catalog. Taken from Leroy-Somer catalog.

2.4.1.2 DC Motor Control

In this subsection, we construct a simulation using PID of the DC motor seen in chapter 2.4.1.1 using the same parameters of the simulation given using (2.52) and the Matlab Algorithm 5 in appendix. Equation (2.53) shows the PID controller where $u(t)$ is the control law, and $e(t)$ is the error of the system we wish to vanish. $K_{\mathrm{p}}, K_{\mathrm{i}}, K_{\mathrm{d}}$ are the constants proportional, integral and derivative respectively,

$$u(t) = K_{\mathrm{p}} e(t) + K_{\mathrm{i}} \int_0^t e(\tau)\,\mathrm{d}\tau + K_{\mathrm{d}} \frac{\mathrm{d}e(t)}{\mathrm{d}t}. \tag{2.53}$$

In Algorithm 8, there is an implementation of the PID in C language intended to be applied in an embedded system. The function PID_Control() must be executed each SAMPLING_TIME seconds.

In Fig. 2.12 we have, numbered from left to right, a Simulink block constant representing the desired angular speed of our motor, which is set to 30 rads/s (1). Next in (2), we have a subtraction symbol that calculates the error between the reference and the feedback with the actual angular speed of the motor shaft. In (3) we have the PID block and bellow it (4) a constant set to 430Nm for the motor load. Next, in (5) we have the DC model with the equations 2.52 written in the Matlab Algorithm 5. In (6) we have an integrator block which is in charge of solving the differential equations in the time-domain.

Currently in the market and industry there exist many tools to tune the PIDs. Here, The Simulink®tool for PID tuning was used. The process of tuning gave us the PID parameters $K_{\mathrm{p}} = 0.837812207200967$, $K_{\mathrm{i}} = 54.359588349525$, $K_{\mathrm{d}} = 0$. For this simulation, the solver was configured as fixed step with step size of 0.001 seconds. The solver is ODE3. In Fig. 2.13 we can see the result of the simulation. In *"Reference"* graph is shown the reference set to 30rad/s; In *"Actual angular speed"* graph we can see how at the beginning the motor starts with oscillations but after a certain time the actual velocity converges to 30rad/s. In the same manner, the *"Error"* graph shows the error size that becomes zero after a time; In *"Control action"* graph we can see the voltage applied by the controller to take the motor to the desired speed.

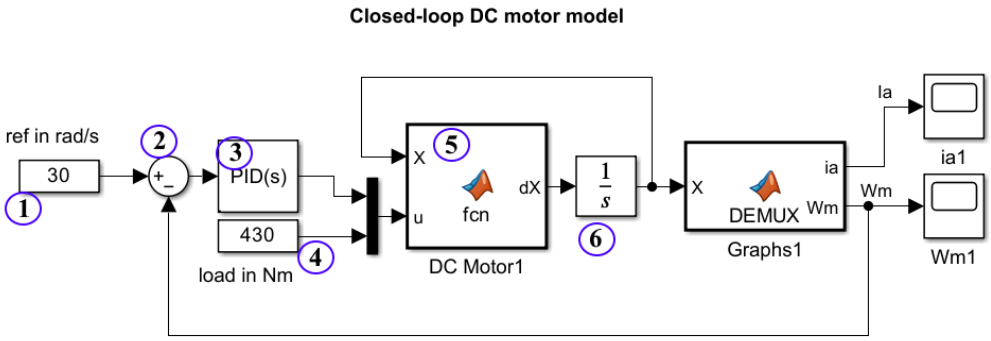


Figure 2.12: Closed-loop DC motor system using PID.

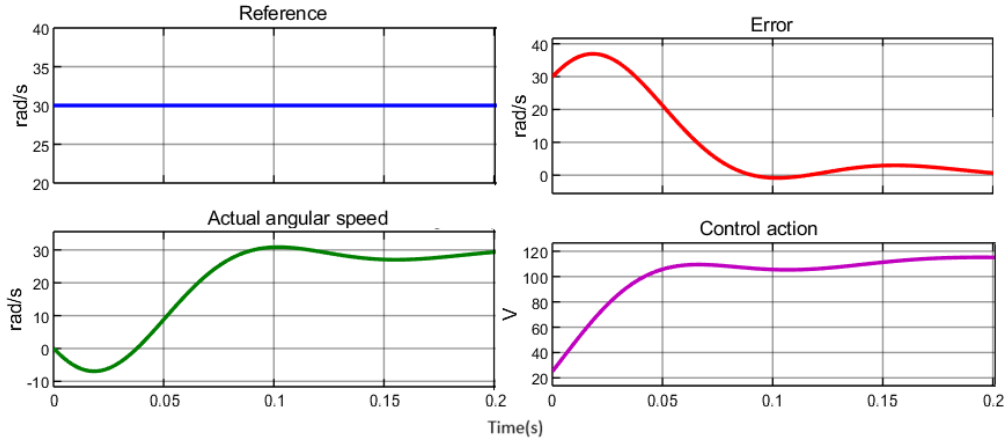


Figure 2.13: DC Motor response to the PID control.

2.4.2 Brush-less DC Motor (BLDC)

The brushless DC motor (BLDC) is one in which its rotor is a permanent magnet commonly made of NdFeB [56], and the stator is supplied with alternating current (AC) obtained from a DC source and converted to AC through an inverter. This motor is lighter, smaller, and better dissipates heat. Some of the drawbacks of this technology are the demagnetization and, overall, the cost of the rare earths. Rare-earths permanent magnets are currently produced in an amount of approximately 1.4×10^5 tons per year, with 80% of the world's supply of rare earths from China. This fact causes the rare earth crisis of 2011 with huge increases in the price of rare earth [17]. The uncertainty of the supply of rare earths has been a major discourage for utilization in mobility; however, for many automakers this is the option and they are still working for the usage of this technology and creating new one with a minor utilization of these materials such as Toyota that is working in a new

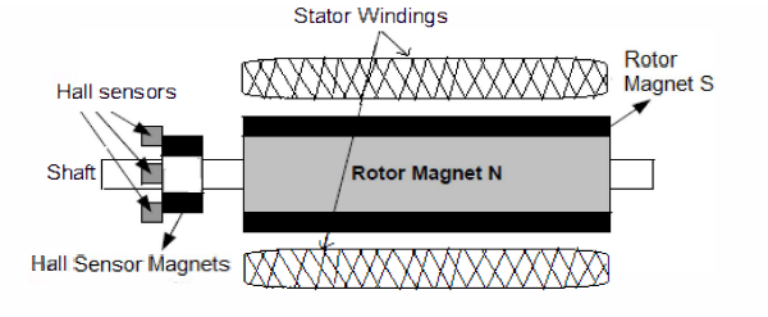


Figure 2.14: BLDC parts. The rotor in made with permanent magnets; the poles in the stator are generated with the electromagnetic fields from the windings in the same stator; the hall sensors detect the position of the rotor in order to the variators be able to feed the current required in the coils. Taken from [13].

motor technology that reduces the need for neodymium by 20% [16] and even Tesla has made a major change in their model 3 which uses permanent magnets (PM) instead of induction motors as its predecessors, and with this change there's a possibility that the global markets experiment additional strain in the supplies for rare earths [17].

BLDC can be found in single phase, two phase and three phase configurations, the three-phase configuration being the most common. As we said before, the BLDC is constructed with a permanent magnet rotor and a wire wound stator poles. Instead of the brush commutator, it has hall effect sensors to detect the position of the rotor and with inverters feed the signal required to activate or deactivate the coils in the stator to produce the rotor movement [53]. See Fig. 2.14.

In Fig. 2.15 we can see the equivalent circuit of the BLDC. The motor is fed by three phase voltage sources. The shape of the voltage can be sinusoidal, square or any other wave shape [16]. The modeling of the BLDC motor is based in the next assumptions: a) The motor is not saturated and should be operated with the rated current. b) The resistances of the three stator phase windings are equal. c) Self-inductance and mutual inductance are constant. d) Semiconductor switches are ideal. Now, the model of the armature winding is as follows [69]

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} L_{aa} & L_{ba} & L_{ca} \\ L_{ba} & L_{bb} & L_{cb} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix}, \tag{2.54}$$

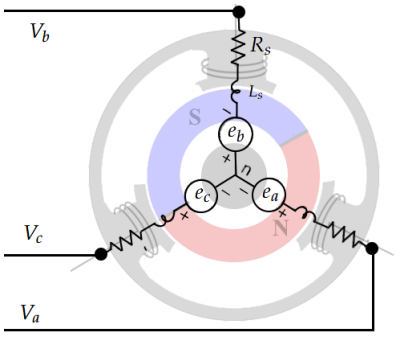


Figure 2.15: Equivalent circuit of a BLDC motor over a watermark representing the physical construction.

where

$$R_a = R_b = R_c = R : \text{Phase resistance},$$
$$L_{aa}, L_{bb}, L_{cc} : \text{Phase self-inductance},$$
$$L_{ab}, L_{ca}, L_{cb} : \text{Phase mutual inductance},$$
$$V_a, V_b, V_c : \text{Phase voltages},$$
$$i_a, i_b, i_c : \text{Phase currents},$$
$$e_a, e_b, e_c : \text{Back EMFs}.$$

Since $L_{aa} = L_{bb} = L_{cc} = L$ and $L_{ab} = L_{ca} = L_{cb} = M$ and according to the star connection and assuming the loads are balanced we have that $i_a + i_b + i_c = 0$ and therefore $Mi_b + Mi_c = -Mi_a$, and so we can simplify as follows:

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} L-M & 0 & 0 \\ 0 & L-M & 0 \\ 0 & 0 & L-M \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix}. \tag{2.55}$$

Rearranging in the state space we have:

$$\frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} \frac{1}{L-M} & 0 & 0 \\ 0 & \frac{1}{L-M} & 0 \\ 0 & 0 & \frac{1}{L-M} \end{bmatrix} \left(\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} - \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} - \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \right). \tag{2.56}$$

The back EMFs e_a, e_b, e_c as we see in the DC motor model, are the voltages that oppose the

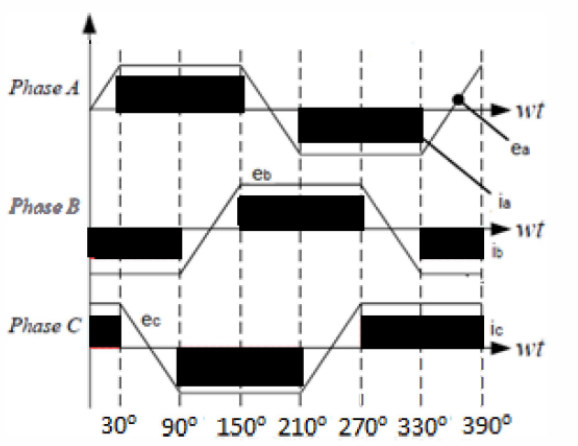


Figure 2.16: Trapezoidal Back EMF and phase current variation with rotor electrical angle. Taken from [53].

main voltage. The polarity of the back EMF is opposite direction of the source voltage. Each phase has 120° phase difference.

Table 2.5 shows signals of the effect hall sensors which are used by the variator to produce the input phases. Fig. 2.16 shows the back EMFs related to the degree of position of the rotor with the current signal. For the back EMFs we have:

$$e_a = \lambda\omega f_a \quad \text{where} \quad f_a = f(\theta), \tag{2.57}$$

$$e_b = \lambda\omega f_b \quad \text{where} \quad f_b = f\left(\theta - \frac{2\pi}{3}\right), \tag{2.58}$$

$$e_c = \lambda\omega f_c \quad \text{where} \quad f_c = f\left(\theta + \frac{2\pi}{3}\right), \tag{2.59}$$

where λ is the Back EMF constant in v/rad/s, θ is the electrical rotor angle, ω is the angular velocity of the rotor in rad/s, and f_a, f_b, f_c are functions that have the same shape as e_a, e_b, e_c with a maximum magnitude of ± 1. The three functions are the same but displaced 120º as in f. The equation for the electromagnetic torque is:

$$T_e = (e_a i_a + e_b i_b + e_c i_c)/\omega, \text{ or} \tag{2.60}$$

$$T_e = (\lambda f_a i_a + \lambda f_b i_b + \lambda f_c i_c), \tag{2.61}$$

and the equation of motion is:

$$\dot{\omega} = (T_e - T_L - b\omega)/J, \tag{2.62}$$

where T_L is the torque load, J is the inertia of the rotor, b is the friction coefficient, and the electrical rotor speed and position is defined as:

$$\dot{\theta} = \frac{P}{2}\omega, \tag{2.63}$$

P is the number of poles; in our case for simulation is 4. Putting all together, the system in the state-space is as in [38] combining (2.56-2.63):

$$\dot{x} = Ax + Bu, \tag{2.64}$$

where

$$x = [i_a\ i_b\ i_c\ \omega\ \theta]', \tag{2.65}$$

$$u = [V_a\ V_b\ V_c\ \ T_L]', \tag{2.66}$$

$$A = \begin{bmatrix} \frac{-R}{L-M} & 0 & 0 & \frac{-\lambda f_a}{L-M} & 0 \\ 0 & \frac{-R}{L-M} & 0 & \frac{-\lambda f_b}{L-M} & 0 \\ 0 & 0 & \frac{-R}{L-M} & \frac{-\lambda f_c}{L-M} & 0 \\ \frac{\lambda f_a}{J} & \frac{\lambda f_b}{J} & \frac{\lambda f_c}{J} & \frac{-b}{J} & 0 \\ 0 & 0 & 0 & \frac{P}{2} & 0 \end{bmatrix}, \tag{2.67}$$

$$B = \begin{bmatrix} \frac{1}{L-M} & 0 & 0 & 0 \\ 0 & \frac{1}{L-M} & 0 & 0 \\ 0 & 0 & \frac{1}{L-M} & 0 \\ 0 & 0 & 0 & \frac{-1}{J} \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{2.68}$$

2.4.2.1 Parameter Estimation and Simulation of Brush-less Direct-Current Electric Machines (BLDC)

Unfortunately, we did not find a catalog for big BLDC motors where we could get the characteristics for a real motor, and hence here we present the simulation for a fictitious BLDC motor. The parameters for the simulation are $R = 0.7, L = 0.00383, M = 0.003, J = 0.6, P = 4, b = 0.0632177, \lambda = 0.9$. Figs. 2.18 and 2.17 show the angular velocity and the phases of the BLDC simulation, respectively. You can find the Simulink® 2019 model in [48].

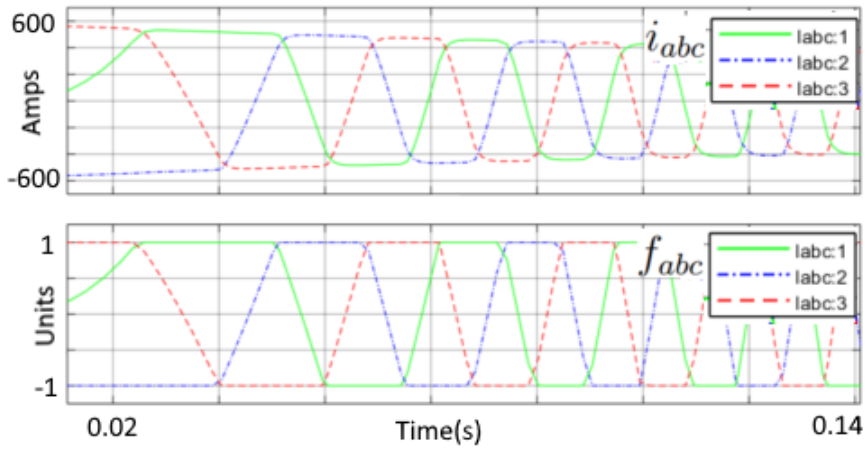


Figure 2.17: Stator currents and the trapezoidal function to simulate the back EMF in the simulation for BLDC.

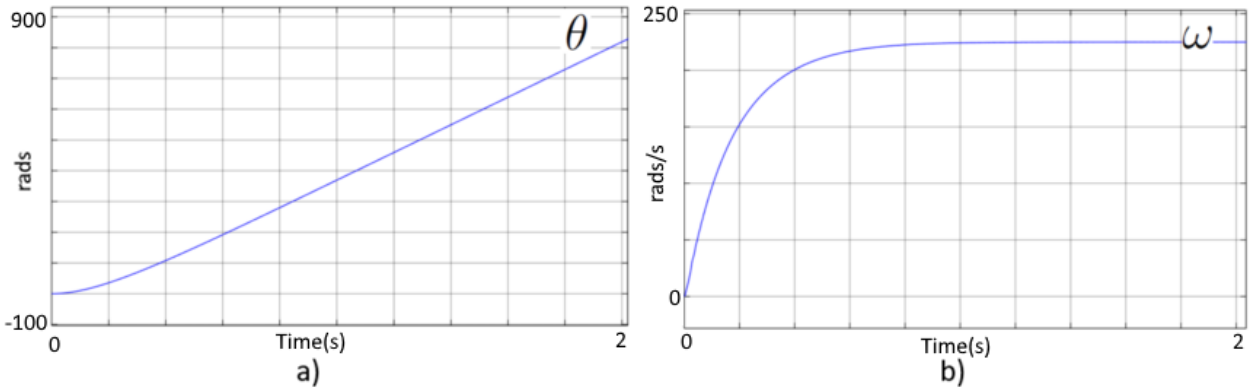


Figure 2.18: Motor Response on the first 2 seconds. a) Angle position (θ) of the motor shaft. b) Angle speed of the motor shaft.

TABLE 2.5. HALL EFFECT SIGNALS

Switching interval degree	**H1**	**H2**	**H3**
0-60	1	0	1
60-120	0	0	1
120-180	0	1	1
180-240	0	1	0
240-300	1	1	0
300-360	1	0	0

2.4.3 Brush-less DC Motor Control Schemes

2.4.3.1 The Park and Clarke Transformations of the Electrical Fields of Three-phase Machines

The Park and Clarke transformations are mathematical tools commonly used in control of electrical motors like AC motors such as permanent magnet synchronous motors (PMSM), brushless direct current (BLDC) motors and induction motors etc. These transformations simplify the control of three-phase AC systems by converting these signals to two sinusoidal signals or even to only two constant signals, simplifying this way all calculations and analysis.

The Clarke transformation converts three-phase quantities (a, b, c) into two-phase quantities (α, β). Working in the (α, β) reference frame simplifies the analysis by reducing the number of signals from three to two.

The Park transformation extends the Clarke transformation by converting the two sinusoidal signals obtained in the stationary reference frame (α, β) into two approximately DC signals (d, q) in a rotating reference frame. When both transformations are combined, the resulting operation is known as the Park–Clarke transformation, which directly maps the three-phase signals (a, b, c) into the (d, q) reference frame.

The motor equations defined before (2.56) and (2.64) with the equivalent diagram in Fig. 2.15 can be reduced in complexity if those are transformed from the three axis at 120° for $a-b-c$ phases to two imaginary and stationary and orthogonal axis $\alpha - \beta$ and after that to a rotational reference frame $d-q$. Fig. 2.19 shows how this transformation happen: a) having 3 balanced phases, applying the Clarke transform we get the b) $\alpha\beta$ reference frame and rotating this frame at the same speed of the electrical phases we get c) the dq reference frame. Fig. 2.20 in b) shows that the sum of the magnetomotive force (mmf) produced by the three electrical phases can be represented by a static 2-axis reference frame [10] called the $\alpha\beta$ reference frame. Fig. 2.20 in c) shows how, if we rotate the $\alpha\beta$ reference frame in b) at the same speed of the electrical angle, then we will have the two stable DC signals. To use these transformations, we need to assume that the three electrical phases

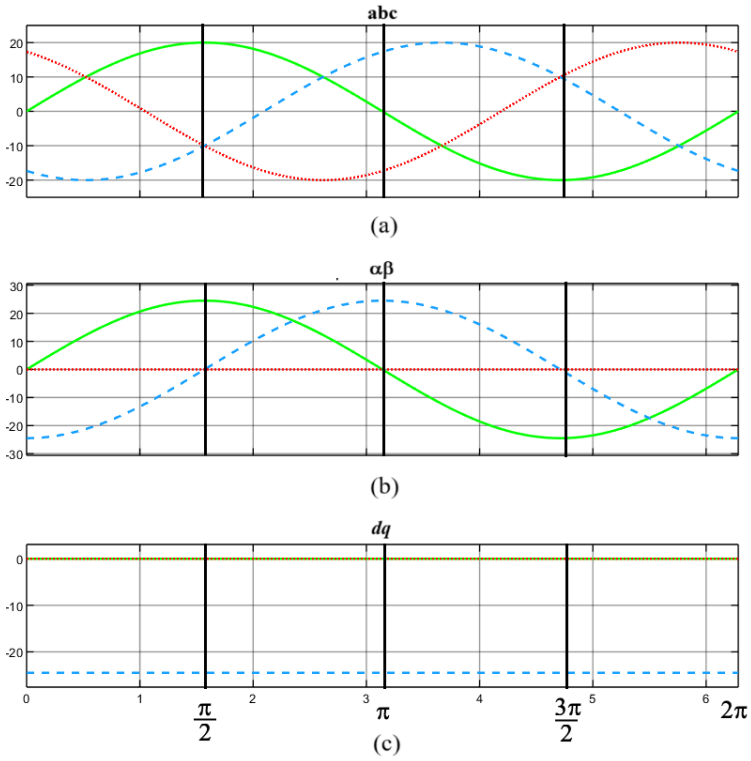


Figure 2.19: Electrical phases in the abc, $\alpha\beta$, *dq* reference frames. This represents one complete revolution, the horizontal axis represents the time and the angle in rads too, since the frequency of these signals are $1/2\pi$, and so, we reach 2π rads in 2π seconds. a) The original three electrical phases at $120°$. b) We can transform the three signals in only two equivalent using the Clarke transformation. c) Using the Park-Clarke we can still simplify even more the three phases to only two constants that can be handled as direct-current signals, which simplify the control of electrical machines.

are balanced, sinusoidal and displaced 120º from each other, that is:

$$\left\{\begin{array}{rl} v_a(t) = & V_a \cos \omega t \\ v_b(t) = & V_b \cos \omega t - \frac{2\pi}{3} \\ v_c(t) = & V_c \cos \omega t + \frac{2\pi}{3} \end{array}\right\}, \tag{2.69}$$

and of course, the loads inside the motor must be balanced, that is $R_a = R_b = R_c$ see (2.54). Fig. 2.19 in a) shows the three phases as those should be in order the Park and Clarke transformations to work.

In this section, different matrices used to convert the reference frame are presented, where we have the so-called power-invariant version and amplitude invariant version [57]. The difference between power invariant and amplitude invariant is that in the power invariant version, the resultant

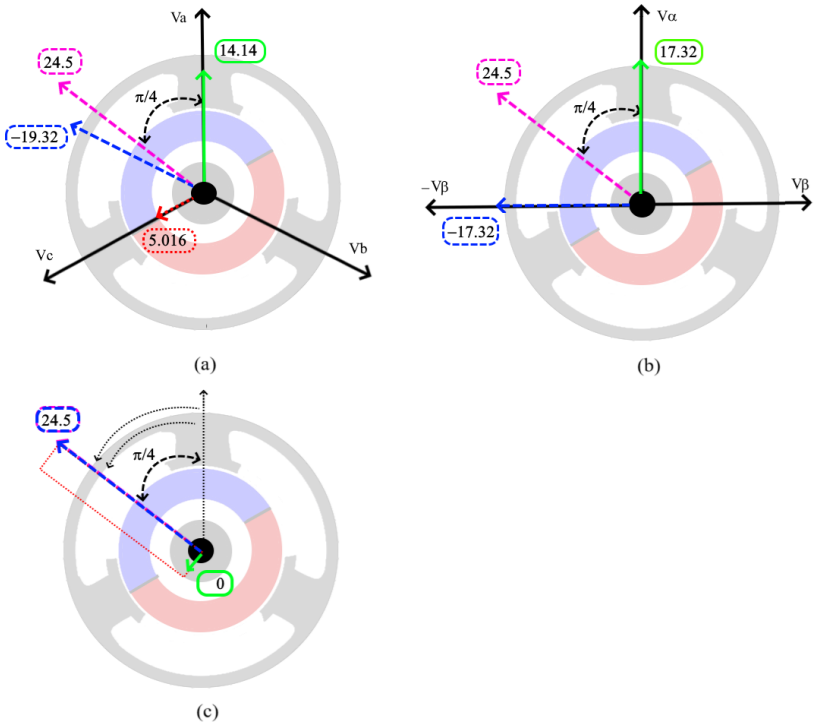


Figure 2.20: The Clarke and Park-Clarke transformation. a) Shows the resultant vector (dashed-pink arrow) obtained with the vectorial sum of the three phases at $120°$ in a giving instant where the resultant has the magnitude of 24.5 and a direction of $\pi/4$. The three vectors (dotted-red, dashed-blue, solid-green) represent the three-phase voltages in the coils. b) is the equivalent 2 two orthogonal vectors in the α, β reference frame; this is the Clarke transformation. Contrary to a), these two vectors are not physical magnitudes but the mathematical equivalence. c) if we rotate the α, β reference frame (Park-Clarke transformation), the same degrees where the resultant vector is, what we get is the two orthogonal vectors but these no longer sinusoidal rather, what we see is two constants. In green with 0 we only show that the second magnitude is 0 because the blue-dashed is the same magnitude as the resultant. If the voltages were out of phase, we would see both orthogonal vectors different from 0.

vector remains invariant after the transformation; however, the amplitude of the signals is affected and has a different magnitude. In contrast, in amplitude invariant, the amplitude of the signals remains the same as the original ones, however, the resultant vector after the vector sum (that would be the power) results in a different magnitude. Fig. 2.21 shows the difference in the result of the vector sum after the transformation using the power invariant or the vector invariant versions of the Park-Clarke transform. The sinusoidal signals are the original three-phase signals whose amplitudes are 20. The two constant signals are the vector sum of the signals after the Park-Clarke transformation. The amplitude invariant has the value of 20, while the power invariant kept the value of 24.5 which is also the result of the vector sum of the original signals. Following, the amplitude

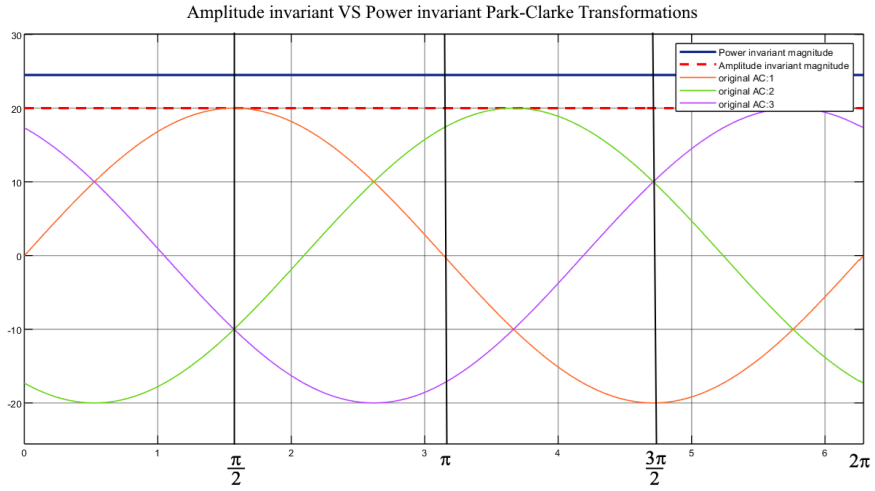


Figure 2.21: Park-Clarke amplitude and power invariant transformations. The magnitude of the resultant vector using power invariant transform (blue) is equal to the vector sum of the 3 electrical phases. The magnitude of the resultant vector after transformation using the amplitude invariant transform is equal to the peak value of the original electrical phases (red).

and power invariant versions of the transformations are presented.

To convert from abc reference frame to $\alpha\beta$ the Clarke transformation matrix in (2.70) is used:

Clarke "Power-invariant" Transform. From abc to stationary $\alpha\beta 0$ reference frame:

$$ {}^{abc}T_{\alpha\beta 0} = \sqrt{\frac{2}{3}} \begin{pmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}. \tag{2.70} $$

Inverse Clarke "Power-invariant" Transform. From $\alpha\beta 0$ to abc reference frame:

$$ {}^{\alpha\beta 0}iT_{abc} = \sqrt{\frac{2}{3}} \begin{pmatrix} 1 & 0 & \frac{1}{\sqrt{2}} \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & \frac{1}{\sqrt{2}} \\ -\frac{1}{2} & \frac{-\sqrt{3}}{2} & \frac{1}{\sqrt{2}} \end{pmatrix}. \tag{2.71} $$

Now, the amplitude invariant matrix are defined as: **Clarke "Amplitude-invariant" Transform. From abc to stationary $\alpha\beta 0$ reference frame:**

$$ {}^{abc}T_{\alpha\beta 0} = \frac{2}{3} \begin{pmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{pmatrix}. \tag{2.72} $$

Inverse Clarke "Amplitude-invariant" Transform. From $\alpha\beta0$ to abc reference frame:

$$^{\alpha\beta0}iT_{abc} = \begin{pmatrix} 1 & 0 & 1 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \\ -\frac{1}{2} & \frac{-\sqrt{3}}{2} & 1 \end{pmatrix}. \tag{2.73}$$

To convert from the $\alpha\beta0$ reference frame to the $dq0$ a pure rotation transformation matrix in (2.74) is used [57]:

$$^{\alpha\beta0}T_{dq0} = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{2.74}$$

and the inverse is:

$$^{dq0}iT_{\alpha\beta0} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{2.75}$$

The $\alpha\beta$ transformations (2.70), (2.72) rotating with the transform (2.75) is the Park or Park-Clarke transformation: **Park "Power-invariant" Transform. From abc to dq0 synchronous rotating reference frame:**

$$T_{\text{PC}} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta) & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ -\sin(\theta) & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}. \tag{2.76}$$

Inverse Park "Power-invariant" Transform. From dq0 to abc stationary reference frame:

$$iT_{\text{PC}} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & \frac{1}{\sqrt{2}} \\ \cos\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta - \frac{2\pi}{3}\right) & \frac{1}{\sqrt{2}} \\ \cos\left(\theta + \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) & \frac{1}{\sqrt{2}} \end{bmatrix}. \tag{2.77}$$

Park "Amplitude-invariant" Transform. From abc to dq0 synchronous rotating reference frame:

$$T_{\text{PC}} = \frac{2}{3} \begin{bmatrix} \cos(\theta) & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ -\sin(\theta) & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}. \tag{2.78}$$

Inverse Park "Amplitude-invariant" Transform. From dq0 to abc stationary reference frame:

$$iT_{\text{PC}} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 1 \\ \cos\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta - \frac{2\pi}{3}\right) & 1 \\ \cos\left(\theta + \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) & 1 \end{bmatrix}. \tag{2.79}$$

The graphic difference between the amplitude invariant and the power invariant is that, for the amplitude invariant the magnitude of the vector after transformation is equal to the peak value of v_a (or v_b or v_c) and for the power invariant transform the magnitude of the vector after transformation is equal to $\sqrt{v_a^2 + v_b^2 + v_c^2}$ Fig. 2.21 shows the resultant magnitude of the vector dq using the amplitude invariant versus power invariant transform. In several applications, it is convenient to use the amplitude invariant transformation; however, for electrical motor control the election is power invariant so that the power between the original system and the transformed remains the same.

Now, what happens if the sinusoidal and balanced load requirements are not met? if we see Fig. 2.22, we can observe the behavior of the transformation when introducing the 3 sinusoidal and balanced signals. We get 2 sinusoidal signals plus a "signal" in 0 that we can omit. However, in the next half of the diagram we introduce 3 signals with trapezoidal shape and what we obtained was 3 complete different signals that are not sinusoidal and neither trapezoidal. In Fig. 2.23 we did the same but using the Park-Clarke transform and what we got was that during the sinusoidal part we got in the output 2 constant signals, but in the trapezoidal part we got 2 irregular signals. This is just to show that the Park and Clarke transforms only work for sinusoidal and balanced signals.

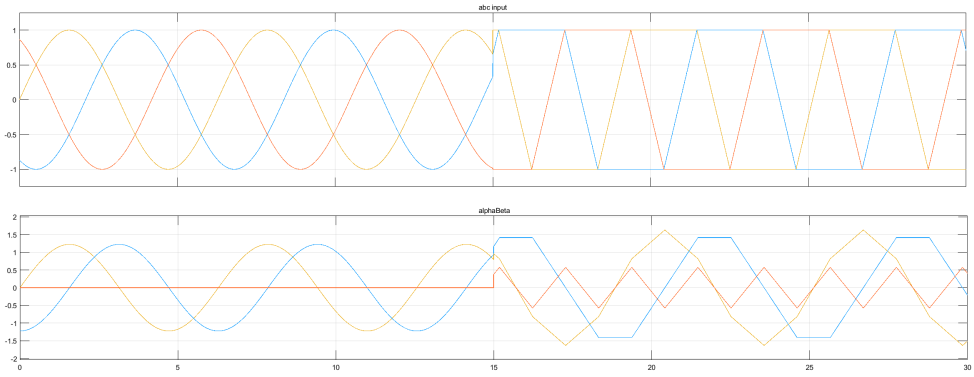


Figure 2.22: Effect of the Clarke transformation when three trapezoidal phases are used as input. During the first 15 seconds, a sinusoidal 3 phases signal is used as input, getting in the output 2 phases sinusoidal signals plus 1 zero phase. During the second 15 to 30, it was fed with a trapezoidal signal and, in the output, we got a messy 3 phases signal.

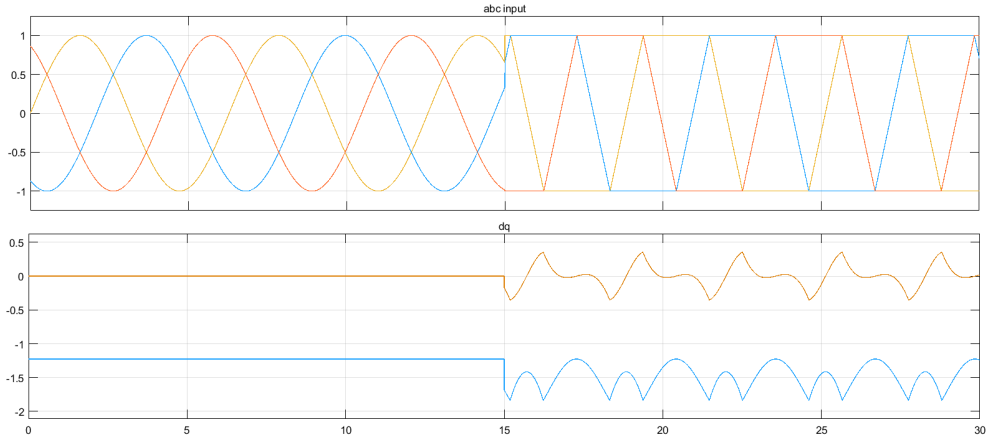


Figure 2.23: Effect of the Park-Clarke transformation when three trapezoidal phases are used as input. During the first 15 seconds, a sinusoidal 3 phases signal is used as input, getting in the output 2 constant signals. During the second 15 to 30, it was fed with a trapezoidal signal and, in the output, we got a messy 2 phases signal.

2.4.3.2 The DQ Model Transformation for Brush-less Direct-Current Electrical Machines

In this section we will transform our BLDC equation in (2.56) based in the abc phases to the dq form to simplify the model and simplify the control methods to use on the motor. The procedure we follow is similar to the one followed in [70]

We are going to multiply (2.56) by the Park-Clarke transform (2.76) that is the power invariant transform. We choose power invariant transform to use the same that was used by [70] and with this, our resultant system will maintain the same power as the original system.

For our purpose, we need the derivative of the Park-Clarke transform $\frac{dT_{\text{PC}}}{dt}$ in terms of the same Park-Clarke transform as follows

$$\frac{dT_{\text{PC}}}{dt} = \sqrt{\frac{2}{3}} \begin{pmatrix} -\sin(\theta) & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) \\ -\cos(\theta) & -\cos\left(\theta - \frac{2\pi}{3}\right) & -\cos\left(\theta + \frac{2\pi}{3}\right) \\ 0 & 0 & 0 \end{pmatrix} \frac{d\theta}{dt}. \tag{2.80}$$

Now, we want to define (2.80) in terms of (2.76), that is the same Park-Clarke transform, and we obtain the A_3 matrix doing $A_3 = \frac{dT_{\text{PC}}}{d\theta} T_{PC}^{-1}$ where $T_{PC}^{-1} = iT_{\text{PC}} = T_{\text{PC}}^{T}$

$$A_3 = \frac{dT_{\text{PC}}}{d\theta} T_{PC}^{-1} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{2.81}$$

then

$$\text{dT}_{\text{PC}} = A_3 \, T_{\text{PC}} \, \frac{\partial \theta}{\partial t}. \tag{2.82}$$

Next step is to multiply our BLDC equation by the Park-Clarke transform:

$$T_{PC} \frac{d}{dt} \begin{pmatrix} i_a \\ i_b \\ i_c \end{pmatrix} = \begin{pmatrix} \frac{1}{L-M} & 0 & 0 \\ 0 & \frac{1}{L-M} & 0 \\ 0 & 0 & \frac{1}{L-M} \end{pmatrix} T_{PC} \left(\begin{pmatrix} V_a \\ V_b \\ V_c \end{pmatrix} - \begin{pmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{pmatrix} \begin{pmatrix} i_a \\ i_b \\ i_c \end{pmatrix} - \begin{pmatrix} e_a \\ e_b \\ e_c \end{pmatrix} \right). \tag{2.83}$$

Now, we want to find the element $T_{PC}\frac{d}{dt}\begin{pmatrix} i_a \\ i_b \\ i_c \end{pmatrix}$; we'll do this by using the rule for product derivative and using our A_3 matrix:

$$\frac{d}{dt}\left(T_{PC}I_{abc}\right) = \dot{T}_{PC}I_{abc} + T_{PC}\dot{I}_{abc}, \tag{2.84}$$

$$T_{PC}\frac{d}{dt}\begin{pmatrix} i_a \\ i_b \\ i_c \end{pmatrix} = \frac{d}{dt}\left(T_{PC}I_{abc}\right) - \dot{T}_{PC}I_{abc} = \dot{I}_{dq0} - A_3 I_{dq0}\frac{\partial\theta}{\partial t}, \tag{2.85}$$

because $\frac{d}{dt}\left(T_{PC}I_{abc}\right) = \dot{I}_{dq0}$ and $\dot{T}_{PC}I_{abc} = A_3 I_{dq0}\frac{\partial\theta}{\partial t}$ since $\dot{T}_{PC}I_{abc} = A_3 T_{PC}I_{abc}\frac{\partial\theta}{\partial t}$ and $T_{PC}I_{abc} = I_{dq0}$, using the terms we found in (2.85) over (2.83) and using matrix notation:

$$\dot{I}_{dq0} = A_3 I_{dq0}\frac{\partial\theta}{\partial t} + L_M V_{dq0} - L_M R_M I_{dq0} - L_M e_{dq0}, \tag{2.86}$$

where

$$R_M = \begin{pmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{pmatrix}, \tag{2.87}$$

$$L_M = \begin{pmatrix} \frac{1}{L-M} & 0 & 0 \\ 0 & \frac{1}{L-M} & 0 \\ 0 & 0 & \frac{1}{L-M} \end{pmatrix}, \tag{2.88}$$

$$e_{dq0} = T_{PC}\begin{pmatrix} e_a \\ e_b \\ e_c \end{pmatrix}, \tag{2.89}$$

$$V_{dq0} = T_{PC}\begin{pmatrix} V_a \\ V_b \\ V_c \end{pmatrix}, \tag{2.90}$$

and rearrange it we have the simplified dq BLDC model:

$$\dot{I}_{dq0} = \left(A_3\frac{d\theta}{dt} - L_M R_M\right) I_{dq0} + L_M V_{dq0} - L_M e_{dq0}. \tag{2.91}$$

If we consider that the electrical angle θ is a function of a defined frequency, let us say $\theta = 2\pi ft$ where f is the frequency then we should multiply the matrix A_3 by $\frac{d\theta}{dt}$ or $2\pi f$ according (2.82).

(2.86) and (2.91) are our model in the dq reference frame. From here we can just make just some simplifications or adaptions.

A simplification we can make is to eliminate the third term of the vector that results from the multiplication by the Park-Clake transform since if the loads are balanced, the third element is 0. that is, if we consider 3 sinusoidal and balanced phases at 120º each, f_a, f_b, f_c, and we multiply it by the Park-Clarke transform we have

$$T_{\text{PC}} \begin{pmatrix} f_a \\ f_b \\ f_c \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{2}\, f_b \, \sin(\theta)}{2} - \frac{\sqrt{2}\, f_c \, \sin(\theta)}{2} + \frac{\sqrt{2}\,\sqrt{3}\, f_a \, \cos(\theta)}{3} - \frac{\sqrt{2}\,\sqrt{3}\, f_b \, \cos(\theta)}{6} - \frac{\sqrt{2}\,\sqrt{3}\, f_c \, \cos(\theta)}{6} \\ \frac{\sqrt{2}\, f_b \, \cos(\theta)}{2} - \frac{\sqrt{2}\, f_c \, \cos(\theta)}{2} - \frac{\sqrt{2}\,\sqrt{3}\, f_a \, \sin(\theta)}{3} + \frac{\sqrt{2}\,\sqrt{3}\, f_b \, \sin(\theta)}{6} + \frac{\sqrt{2}\,\sqrt{3}\, f_c \, \sin(\theta)}{6} \\ \frac{\sqrt{3}\,(f_a + f_b + f_c)}{3} \end{pmatrix}. \tag{2.92}$$

We can observe in the third element, that is in function of the arithmetic sum of the three phases $(f_a + f_b + f_c = 0)$ and then the third element is always 0 and we can eliminate it. For that reason we call it dq0 before eliminating this term and dq once we remove this element. With what we stated before, we can create a Park-Clarke transform that delivers the only 2 components dq:

$$T_{PC} = \frac{\sqrt{2}}{2} \begin{pmatrix} \sqrt{3}\,\cos(\theta) & \sin(\theta) & -\sin(\theta) \\ -\sqrt{3}\,\sin(\theta) & \cos(\theta) & -\cos(\theta) \end{pmatrix}. \tag{2.93}$$

2.4.3.3 Field-Oriented Control for BLDC

Field-oriented Control (FOC) is a control method applied in controlling AC motors in application where high precision in torque and speed is required. Permanent magnet synchronous motors (PMSM), BLDC motors, induction motors etc. are commonly controlled using FOC. FOC has the characteristic that decouple the torque and flux, giving the possibility of controlling both of them independently.

Unlike DC motors, AC motors have a more complex interaction between the applied voltage and

the resulting magnetic fields. Traditional control methods like scalar control, which is the control of voltage and frequency in in variable frequency drives (VFD), primarily manage the motor's speed by maintaining a fixed voltage-to-frequency ratio. However, this approach lacks the ability to precisely control torque, which limits its effectiveness in dynamic applications.

Field-oriented Control overcomes these limitations by transforming the stator currents into a rotating reference frame, allowing the torque and flux to be regulated independently. This results in improved efficiency and dynamic performance, making FOC well-suited for applications such as electric vehicles, robotics, and industrial automation.

FOC works by transforming the motor's three-phase stator currents into two orthogonal components, referred to as the direct axis (d-axis) and quadrature axis (q-axis). These components are aligned with the rotor's magnetic field. Direct axis (d-axis) corresponds to the current i_d, controls the rotor's magnetic flux. Quadrature axis (q-axis) corresponds to the current i_q, controls the motor's torque. In FOC, the d-axis current i_d is usually maintained at a constant level, or zero in the case of PMSMs or BLDC where the rotor flux is provided by permanent magnets. The q-axis current i_q is adjusted to control the torque output, establishing a linear relationship between torque and the i_q current.

In order to be able to apply FOC, we have to convert the three-phase system in the abc reference frame to the rotational dq reference frame as we saw in chapter 2.4.3.1. by convention, the q-axis is oriented towards the a-axis as reference for the angle position between the "dq" frame and the abc. Once it is converted, it can be applied PID control to both currents i_d, i_q separately. Once the desired i_d and i_q currents are computed, an inverse Park and Clarke transformation is applied to return to the original three-phase reference frame. These signals are then used to generate pulse width modulation (PWM) signals, which control the inverter and modulate the voltage applied to the motor's stator.

The scheme for implementing FOC is depicted in Fig. 2.24. As we described before, we convert the three sinusoidal current measurements abc to two DC currents dq using the **PCT** block and using the angle position measurement δ. These currents are the feedback of the closed-loop system. As references we have $I_{q,ref}$ which is used to establish the desired torque; meanwhile we set to zero the

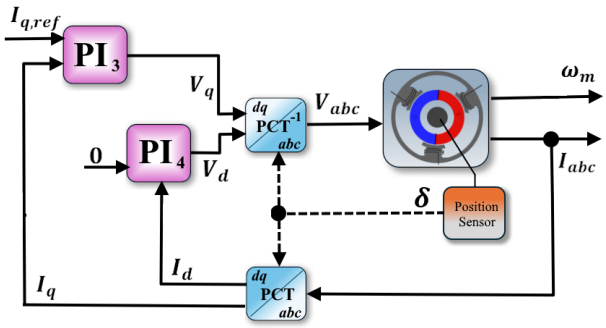


Figure 2.24: Control scheme to control the torque of a BLDC motor.

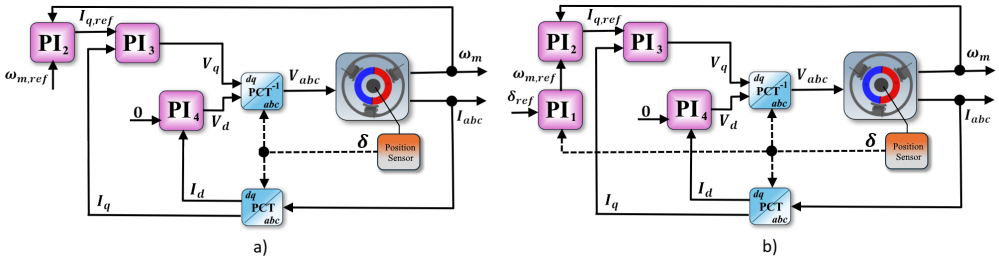


Figure 2.25: Control scheme to control a) the angular speed, b) the absolute angle position of a BLDC motor.

reference for the I_d current which is the equivalent to the rotor flux. With this arrangement we only need to control one signal $I_{q,ref}$ that will control the torque of our motor. The two outputs of the PI controllers PI_3 and PI_4 are V_q, V_d respectively of the voltages required, however, these are in the qd reference system which are not useful in these form, and hence, we convert these signals to the abc voltages using the inverse of the Park-Clarke transformation $\mathbf{PCT}^{-1}$; it is worth to notice that the variator required to create the sinusoidal voltages is not shown in the diagram; it would be located after the inverse of the PTC transform and before the BLDC motor. Now, if we add one PI controller more (PI_2) to the last scheme, we can control the angular speed of the motor; taking as feedback the angular speed ω_m and the output of the additional PI would be the reference $I_{q,ref}$. Fig. 2.25 a) shows the new scheme. And now, the new reference for the system is $\omega_{m,ref}$ We can add another PI (PI_1) to control the absolute position of the motor. This new PI receives the feedback of the position of the motor shaft and takes as reference the desired angle position δ_{ref} as in Fig. 2.25 b).

Fig. 2.26 shows the Simulink model of the BLDC with a control scheme to control the absolute angle position, and Table 2.7 gives a brief description of the numbered elements. The way to tune the PIs was using the automatic tool of Simulink and by hand to fine tune. First, PI_3 and PI_4 where

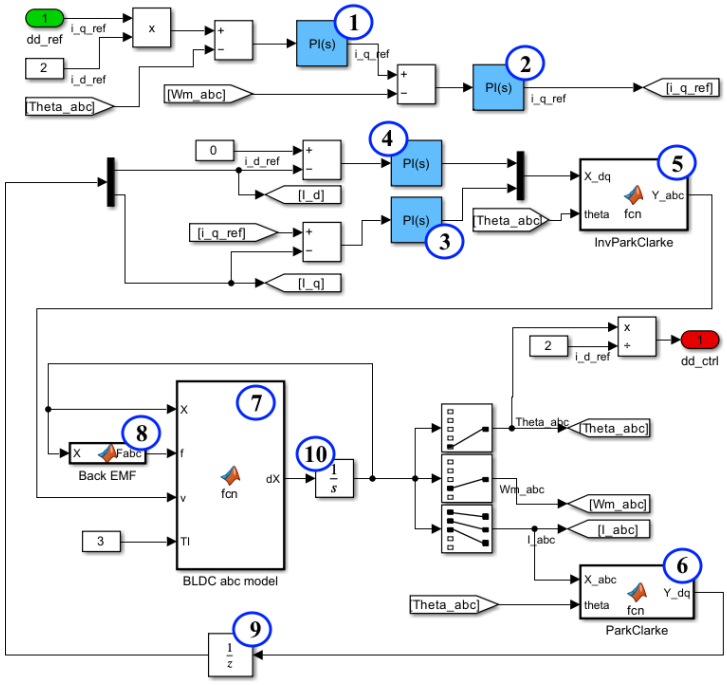


Figure 2.26: Simulink blocks to control the angle of the BLDC.

tuned; when these two were stable, PI_2 was added and tuned; finally, PI_1 was added and tuned. Fig. 2.27 shows the response of the BLDC with this control scheme. Table 2.6 shows the parameters of the BLDC and the gains of the PIs controllers.

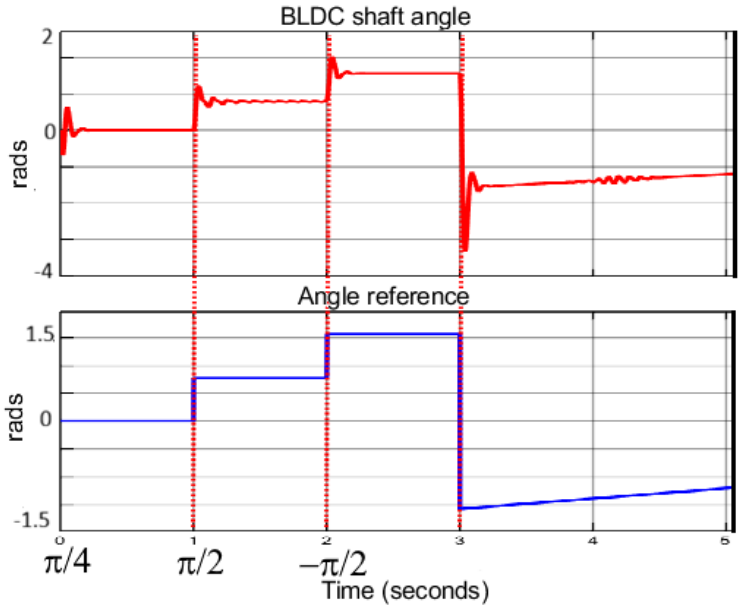


Figure 2.27: Simulation of the model in Fig. 2.26. *Angle reference* is the reference angle position commanded to the BLDC.*BLDC shaft angle* is the actual response of the BLDC.

TABLE 2.6. BLDC MOTOR PARAMETERS AND PIDS GAINS

Symbol	Value	unit	Gain	Value
λ	0.0433		PI_1, K_p	21.2472922845302
R	0.35	Ω	PI_1, K_i	1497.59558537531
L	0.001	H	PI_2, K_p	-0.39820378606854
M	0	H	PI_2, K_i	-44.4991237925307
J	0.00018	kgm^2	PI_3, K_p	70
P	4	-	PI_3, K_i	0.3
b	0.0034	Nms	PI_4, K_p	70
			PI_4, K_i	0.3

TABLE 2.7. DESCRIPTION OF SIMULINK BLOCKS TO CONTROL ANGLE POSITION

Num	**Description**
1	PI_1 Which controls the angle position.
2	PI_2 Which controls the angular speed of the BLDC shaft.
3	PI_3 Controls the quadrature component, or the torque or the BLDC.
4	PI_4 Controls the direct component, or the rotor flux. Notice this is set to 0.
5	Inverse of Park-Clarke transform. See Algorithm 10
6	Park-Clarke transform. See Algorithm 11
7	BLDC model in the abc reference frame. See Algorithm 6
8	Shapes of the back electromagnetic force. See Algorithm 7
9	Standard Simulink block *"Unit delay"*. This is required to provide the feedback.
10	Standard Simulink block *"Integrator"*. This integrator solves the differential equations of the BLDC.

2.4.3.4 Space-Vector Pulse Width Modulation

When a three-phase sinusoidal and balanced voltage, given by the equations

$$v_a = V_m \cos(\omega t), \tag{2.94}$$

$$v_b = V_m \cos(\omega t - \frac{2\pi}{3}), \tag{2.95}$$

$$v_c = V_m \cos(\omega t + \frac{2\pi}{3}), \tag{2.96}$$

that are the same shown in (2.69) but using a simpler notation, are applied to a three phase induction motor it can be shown that the space vector V with magnitude V_m rotates in a circular orbit at angular velocity ω where the direction of rotation depends on the phase sequence of the voltages.

Looking at the Fig. 2.28 we can see the different vectors that are formed depending in the state of the phases. For example, the vector V_1 is (1 0 0) that means that the state of the phases are (f_a = On, f_b = Off, f_c = Off), the vector V_2 (1 1 0) means that (f_a = On, f_b = On, f_c = Off) and so on. The vectors V_0 and V_7 mean that the phases are (f_a = Off, f_b = Off, f_c = Off) and (f_a = On, f_b = On, f_c = On) and hence, there is no electrical power in these two vectors and for that reason they are called the zero vectors.

The idea of the space-vector PWM is to commute between two adjacent vectors at a speed such that the sum of the two vectors results in a vector of the size and angle required to provide the power and angular speed desired of the motor. In the following part we will calculate the proportional times for the duty cycle of pwm, of the vectors required to have the resultant vector of size $|V_s|$ and angle θ. Following we will analyze the process, taking in account that it is for the under-modulation state.

Having $T_1 \propto V_a$, $T_2 \propto V_b$ where T_1, T_2 are the time active of each vector, θ is the angle of the vector reference V_s in the Fig. 2.28; looking at the Fig. 2.29 we can get the relative sizes of the vectors V_a and V_b in this way:

Having V_s as our desired vector, and θ its angle. Taking the assumption that the angle is in the first section (see Fig. 2.28), we want the proportionality of V_a and V_b which represents the phases in the state V_1 (1 0 0) and V_2 (1 1 0) that corresponds to the angles 0 and $\pi/3$ ($60°$) correspondingly

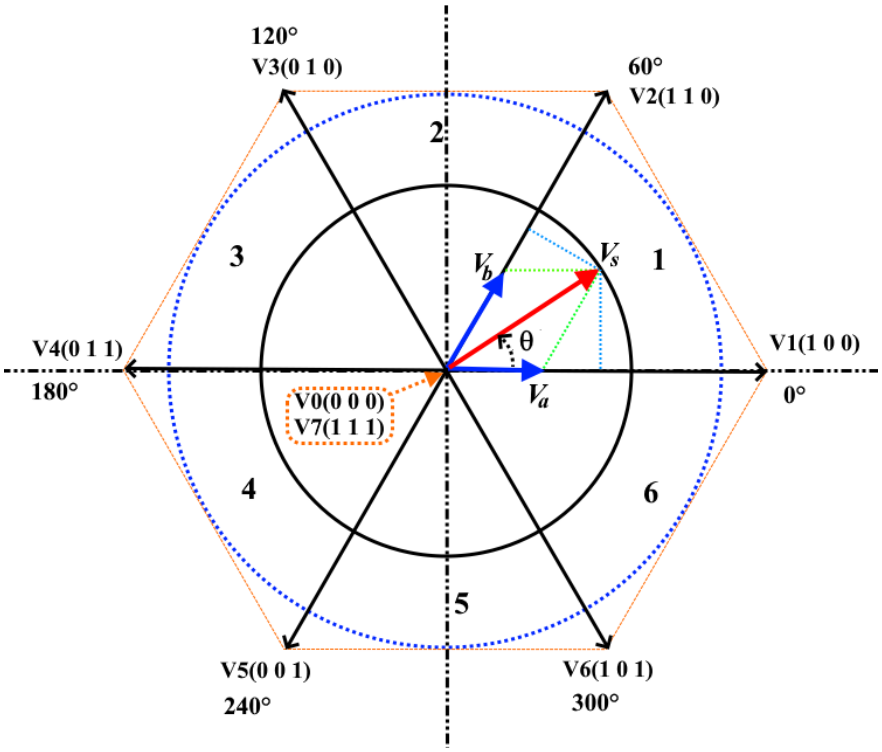


Figure 2.28: SVPWM. Space vectors of three-phase bridge inverter showing reference voltage trajectory and segments of adjacent voltage vectors.

(see Fig. 2.29).

$$V_s \angle\theta = V_a(\cos 0 + i \sin 0) + V_b(\cos \frac{\pi}{3} + i \sin \frac{\pi}{3}), \tag{2.97}$$

we separate in the real and complex components as follows:

$$V_s(\cos\theta + i \sin\theta) = V_a(\cos 0 + i \sin 0) + V_b(\cos \frac{\pi}{3} + i \sin \frac{\pi}{3}), \tag{2.98}$$

$$V_s \cos\theta = V_a \cos 0 + V_b \cos \frac{\pi}{3}, \tag{2.99}$$

$$V_s i \sin\theta = V_a i \sin 0 + V_b i \sin \frac{\pi}{3}, \tag{2.100}$$

with the equivalences for cosine and sine of $\frac{\pi}{3}$:

$$\cos \frac{\pi}{3} = \frac{1}{2}, \tag{2.101}$$

$$\sin \frac{\pi}{3} = \frac{\sqrt{3}}{2}, \tag{2.102}$$

we simplify to

$$V_s \cos\theta = V_a + \frac{V_b}{2}, \tag{2.103}$$

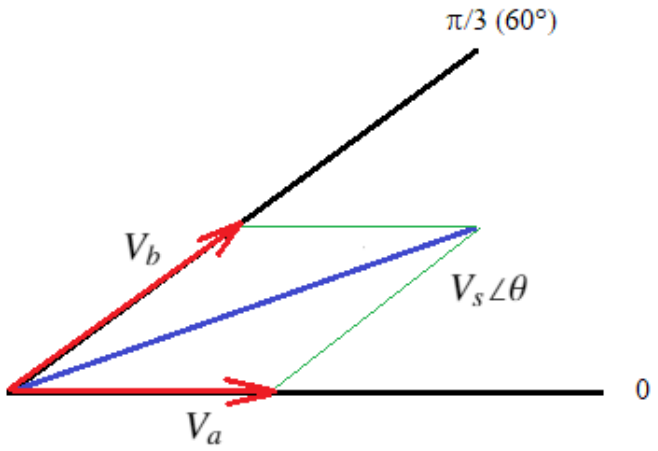


Figure 2.29: V_s vector as a result of the two vectors V_a and V_b separated by $60°$.

$$V_s i \sin\theta = iV_b \frac{\sqrt{3}}{2}. \tag{2.104}$$

Solving for V_a and V_b:

$$V_b = \frac{2V_s}{\sqrt{3}} \sin\theta, \tag{2.105}$$

having already solved V_b we proceed to solve for V_a combining (2.105) with (2.103)

$$V_a = V_s \cos\theta - \frac{V_b}{2}, \tag{2.106}$$

$$V_a = V_s \cos\theta - \frac{V_s}{\sqrt{3}} \sin\theta, \tag{2.107}$$

rearranging and simplifying with trigonometric identities (sum of two angles)

$$V_a = \frac{2V_s}{\sqrt{3}} \left(\frac{\sqrt{3}}{2} \cos\theta - \frac{1}{2} \sin\theta \right) = \frac{2V_s}{\sqrt{3}} \left(\cos\frac{\pi}{6} \cos\theta - \sin\frac{\pi}{6} \sin\theta \right), \tag{2.108}$$

$$V_a = \frac{2V_s}{\sqrt{3}} \cos\left(\theta + \frac{\pi}{6}\right), \tag{2.109}$$

using the trigonometric identities:

$$\sin x = \cos\left(\frac{\pi}{2} - x\right), \tag{2.110}$$

$$\cos x = \sin\left(\frac{\pi}{2} - x\right), \tag{2.111}$$

we simplify to use only sine function, we have:

$$V_a = \frac{2V_s}{\sqrt{3}} \sin\left(\frac{\pi}{2} - \theta + \frac{\pi}{6}\right), \tag{2.112}$$

$$V_a = \frac{2V_s}{\sqrt{3}} \sin\left(\frac{\pi}{3} - \theta\right). \tag{2.113}$$

Looking at the Fig. 2.28 we see that the vectors V_a and V_b are aligned to the voltages V_1 and V_2. Considering T_c as the period of our pwm, let us define our t_a, t_b as the times that the voltages V_1 and V_2 are ON during one cycle of the pwm; the t_0, therefore, would be the time in Off of the cycle.

$$V_s = V_a + V_b = V_1 \frac{t_a}{T_c} + V_2 \frac{t_b}{T_c} + (V_0 \text{ or } V_7)\frac{t_0}{T_c}, \tag{2.114}$$

therefore

$$t_a = \frac{V_a}{V_1} T_c, \tag{2.115}$$

$$t_b = \frac{V_b}{V_2} T_c, \tag{2.116}$$

$$t_0 = T_c - (t_a + t_b). \tag{2.117}$$

With these times we can construct a pwm pattern for each phase. M code in Algorithm 12 shows an example of code to calculate the vectors V_a, V_b. The Simulink models for simulations can be found in [80].

2.4.4 Permanent Magnet Synchronous Motor (PMSM)

This is a short overview of the permanent magnet synchronous motors. It is an AC motor that uses permanent magnets embedded in the rotor to create a constant magnetic field. The stator generates a rotating magnetic field, and the rotor locks in synchrony with it, ensuring high efficiency, precise speed control, and compact design. PMSMs are widely used in electric vehicles, robotics, and industrial automation due to their excellent torque density and reliability.

There are two types of PMSM motors by its construction. Sinusoidal surface magnet machine (SPM) where the magnets are mounted on the surface of the iron rotor using epoxy adhesive. On

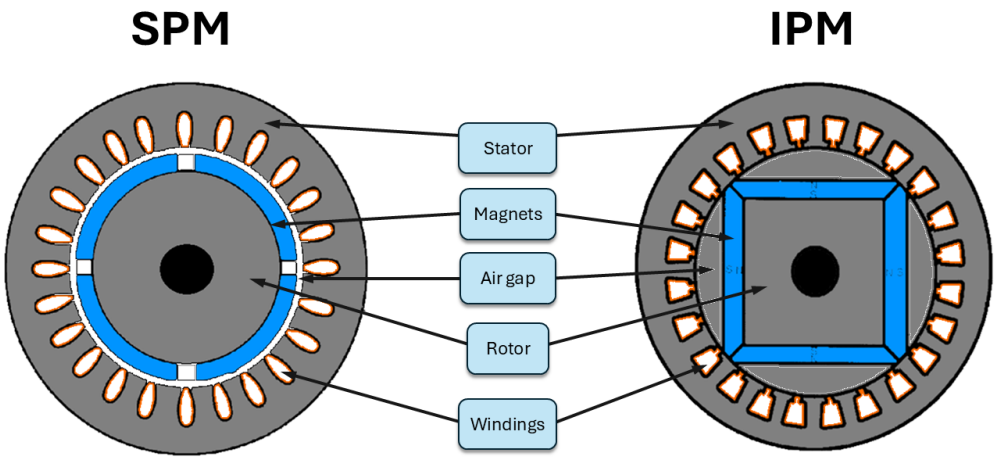


Figure 2.30: Scheme showing the difference between surface permanent magnet and internal permanent magnet.

the contrary, sinusoidal interior magnet machine (IPM), the magnets are mounted inside the rotor. The advantages of IPM are that it is more robust where the operation speed can be much higher and the armature reaction effect becomes dominant

Below is the model for the IPM in the dq refference frame:

$$\frac{di_{qs}}{dt} = \frac{\omega_b}{X_{qs}}\left[v_{qs} - R_s i_{qs} - \frac{\omega_e}{\omega_b} X_{ds} i_{ds} - \frac{\omega_e}{\omega_b} V_f\right], \tag{2.118}$$

$$\frac{di_{ds}}{dt} = \frac{\omega_b}{X_{ds}}\left[v_{ds} - R_s ids - \frac{\omega_e}{\omega_b} X_{qs} i_{qs}\right], \tag{2.119}$$

$$T_e = \frac{3P}{4\omega_b}\left[\left(F'_{ds} + V_f\right) i_{qs} - F_{qs} i_d s\right], \tag{2.120}$$

where $V_f = \omega_b \hat{\psi}_f$, $X_{qs} = \omega_b L_{qs}$, $X_{ds} = \omega_b L_{ds}$, $F'_{ds} = \omega_b \psi'_{ds}$, $F_{qs} = \omega_b \psi_{qs}$, ω_b is the base frequency, L_{qs} and L_{ds} are the q and d axis magnetizing inductance, ψ_{qs} is the q axis stator flux linkage, i_{qs} and $i_d s$ are the q and d axis currents, v_{qs} and v_{ds} are the q and d axis voltages. More information about these motors can be found in [10].

2.4.5 Switched Reluctance Motor (SRM)

The Switched Reluctance Motor (SRM) is one of the earliest electrical machines that became practical in the 1970's thanks to advances in power electronics. Today, it is gaining attention due to its unique advantages compared to conventional motors. It does not require permanent magnets or

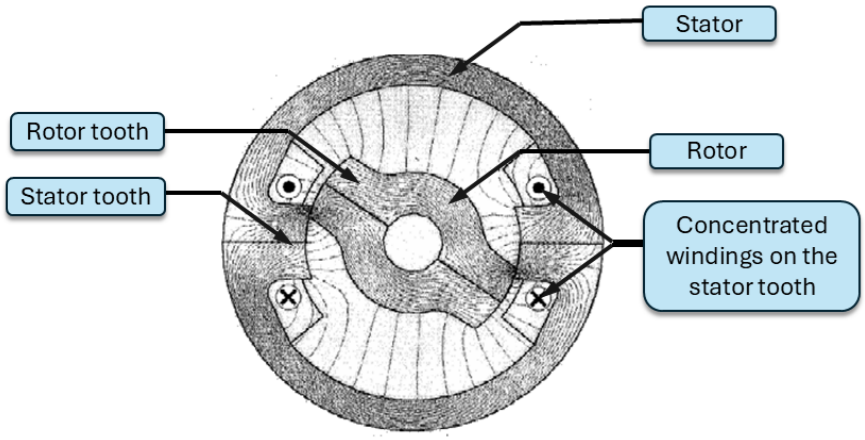


Figure 2.31: Scheme showing the construction of the switched reluctance motor of one phase and two poles.

rotor windings; therefore, the production cost is lower and independent from rare-earth materials. The losses are concentrated in the stator. It is phase independent because it is electrically and magnetically isolated, improving fault tolerance and maintenance. It is also high speed capable. Among the disadvantages are the high torque ripple, vibration, and acoustic noise. It requires complex control since it needs precise synchronization of phase excitation, and advanced controllers must handle the nonlinearities. It is required accurate identification of the inductance and flux linkage, which is difficult, requiring experimental data, finite element analysis, or machine learning methods. Fig. 2.31 shows the elements of construction of the SRM. As can be seen, this construction is simpler than motors.

The SRM model is as follows:

$$v = e + \frac{R}{i}, \tag{2.121}$$

$$e = \frac{\partial \psi}{\partial t}, \text{ or} \tag{2.122}$$

$$e = \omega_m \frac{\partial \psi}{\partial \theta}, \text{ if speed is constant,} \tag{2.123}$$

where ω_m is the angular velocity in rad/s. ψ is the flux linkage. e is the back EMF. θ is the rotor position in rad.

The average torque is

$$T_{e(avg)} = \frac{SW}{2\pi}, \tag{2.124}$$

where S is the number of strokes per revolution, W is the energy converted from electrical to mechanical, and N_r is the number of rotor poles and m the number of phases in the equation below:

$$S = mN_r. \tag{2.125}$$

Now, the instantaneous torque is computed from a derivative of coenergy W_c or W_f, and it is defined as

$$W_f = \int id\psi, \tag{2.126}$$

$$W_c = \int \psi di, \tag{2.127}$$

and the instantaneous electromagnetic torque is

$$T_e = \frac{\partial W_c(i,\theta)}{\partial \theta} = -\frac{\partial W_f(\psi,\theta)}{\partial \theta}. \tag{2.128}$$

The flux linkage ψ is evaluated as

$$\psi = \frac{\partial W_c(i,\theta)}{\partial i}. \tag{2.129}$$

We can get another expression for the back EMF as

$$e = \omega_m \frac{\partial^2 W_c}{\partial \theta \partial i}. \tag{2.130}$$

A suggestion how to compute the store energy is given in [52] using a predefined spline from θ perspective for example:

$$W_f = c_0 + c_1\theta + c_2\theta^2 + c_3\theta^3. \tag{2.131}$$

Then, according to (2.128), the instantaneous torque is

$$T_e = -(c_1 + 2c_2\theta + 3c_3\theta^2). \tag{2.132}$$

This model can be found in [52] and [11].

2.4.6 AC Induction Motor (Asynchronous)

An alternating current (AC) induction motor is one of the most widely used electric motors in industry and households. It works by converting AC electrical energy into mechanical energy through electromagnetic induction. Its advantages include simplicity, robustness, and low cost, while its disadvantages involve lower efficiency at light loads and difficulty in speed control.

To produce rotation, the stator produces a rotating magnetic field when AC flows through its windings. This induces a current in the rotor, which creates its own magnetic field. Some advantages of this type of motors are the low cost compared to other motors, low maintenance because it does not use brushes, high starting torque etc. Among the disadvantages we can mention we have, the control is difficult, low efficiency, and it is not suitable for precision applications.

In early versions of Tesla vehicles, these motors were used. Now, they are moving (and almost all EVs) to permanent magnet motors because of their efficiencies.

Assuming three sinusoidal phases feed the stator windings

$$i_a = I_m \cos \omega_e t, \tag{2.133}$$

$$i_b = I_m \cos \omega_e t - \frac{2\pi}{3}, \tag{2.134}$$

$$i_c = I_m \cos \omega_e t + \frac{2\pi}{3}, \tag{2.135}$$

where I_m is the RMS magnetizing current, ω_e is the stator or line frequency in rad/s.

The magnetomotive force (mmf) at angle θ is

$$F(\theta, t) = \frac{3}{2} N I_m \cos \omega_e t - \theta. \tag{2.136}$$

The rotational synchronous speed in RPMs is

$$N_e = 120 \frac{f_e}{P}, \tag{2.137}$$

where $f_e = \omega_e / 2\pi$, P are the stator frequency in Hz and the number of poles, respectively.

The torque production is given by the difference between the rotor speed N_r and the synchronous speed of the magnetic fields in the stator N_e. This is called the slip speed and it is $N_e - N_r$ and can

be calculated the per unit slip S as

$$S = \frac{N_e - N_r}{N_e} = \frac{\omega_e - \omega_r}{\omega_e} = \frac{\omega_{sl}}{\omega_e}, \tag{2.138}$$

where ω_e is the stator supply frequency, ω_r is the rotor electrical speed, ω_{sl} is the slip frequency. The rotor mechanical speed is $\omega_m = (2/P)\omega_r$.

$$T_e = \pi \left(\frac{P}{2}\right) lr B_p F_p \sin\delta, \tag{2.139}$$

where l is the axial length of the motor, r is the motor radius, B_p is the peak value of the air gap flux density, F_p is the peak value of the rotor mmf and $\delta = \pi/2 + \theta_r$ is the torque angle. θ_r is the rotor angle.

2.5. Sliding Mode Control

2.5.1 Brief Introduction to Control Theory

Control theory is a field that studies the dynamics of systems and looks to use a feedback from the system to regulate the action of an actuator that keeps the system in an stable state according a given reference. We can find two types of control systems, open-loop and closed-loop. Open-loop control systems are those that do not have feedback signals to compensate the error in the output. Some examples of open-loop control system can be a washing machine's timer. On the other hand we have the closed-loop systems which, based on the output signal (the feedback) can compensate the existing error respect the desired input; as example we have the cruise control system in a vehicle. Fig. 2.32 shows a diagram with the basic elements of a closed-loop system. Following there is a brief explanation of the elements in the diagram:

Control elements. These are the elements that, based in the error signal, calculate the action required to reduce the error and keep it stable in zero. Sometimes, it is not possible to reach zero error, but keep the error bounded in an small and acceptable range; this remaining error is called error in steady state.

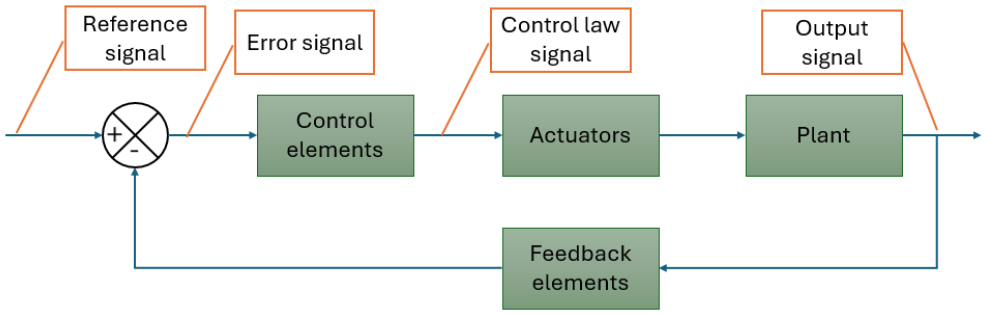


Figure 2.32: Basic elements of a closed loop control system.

Actuators. These are the elements that actually actuate over the plant that we want to control. These receive the signal from the controller and perform the physical action. As example, a controller that controls the temperature of a hair dryer using a electrical resistance, the controller can increase or reduce the voltage applied to the resistance, that is our actuator, and the resistance increase or reduce the temperature making warmer the air. Another example can be an electric motor in a steer by wire system; the controller commands the angle of the steering required to keep the orientation and converts this angle to the required voltage signals to move the motor and keep it in position.

Plant. It refers to the physical system or process that we want to control. It is the system whose behavior needs to be regulated or manipulated to achieve a desired output. The plant is typically modeled mathematically to predict its response to different inputs and disturbances, enabling the design of a control strategy.

Feedback elements. These are the sensors or elements that provide the feedback to the controller. Based on this feedback, the error against the reference signal is calculated. For example, in the hair dryer a temperature sensor can be used to provide the feedback to the controller about the actual temperature of the system. In the case of the steer by wire, the feedback element can be an encoder or hall effect position sensor in the motor shaft that provides the feedback about the actual position of the steering.

Reference signal. This signal is the input to the controller and commands the desired situation of the system. It can be the desired temperature, position, speed etc.

Error signal. It is the deviation of the desired state of the system and the actual state.

Control law signal. Is the signal that commands the controller to compensate the error state.

Output signal. This is the actual state of the system that we are interested in controlling.

The system stability is a highly important concept in control theory. It refers to the ability of a system to return to its equilibrium point after being disturbed. We say that a system is stable if it is capable to maintain the desired behavior over the time; on the contrary, unstable is a system that diverges from the desired behavior or produces uncontrollable oscillatory responses. Asymptotic stability is when a system, after a disturbance, the system error approaches zero, or the desired state, over time. As example, we can see a pendulum which is perturbed, it will recover its stable state after some time. Marginal stability refers to a system which after being perturbed, does not recover completely the stable state, oscillating within a boundary but not diverging. Unstable, is a system which we can not control because it never approaches the desired state or even, the response diverges over time.

These few concepts can be useful to understand the control applied to electric machines in this chapter.

2.5.2 Sliding Mode Control Principles

Sliding Mode Control (SMC) is a robust nonlinear control strategy that forces a system's state onto a predefined sliding surface and maintains it there using a high-frequency discontinuous control law. By rapidly switching between different discrete control states, the system becomes insensitive to external disturbances and parameter variations, a property known as invariance. While SMC simplifies complex dynamics and ensures stability, its primary practical challenge is chattering, which consists of high-frequency oscillations caused by the high-speed switching that can potentially damage mechanical actuators. However, new techniques have emerged that reduce or even eliminate this chattering, such as high order SMC techniques that we will cover succinctly later in this dissertation.

First, let us describe the main principle of SMC which is called conventional SMC. Let us

assume that we have the following system:

$$\dot{x}_1 = x_2, \tag{2.140}$$
$$\dot{x}_2 = u + f(x_1, x_2, t),$$

with the initial conditions

$$x_1(0) = x_{i1},$$
$$x_2(0) = x_{i2},$$

where u is the control law and $f(x_1, x_2, t)$ is the unknown disturbance. The problem to solve is to drive the state variables x_1, x_2 to zero asymptotically or in finite time.

Taking as an example the system (2.140), that could represent x_1 distance, and x_2 the speed, with the initial conditions:

$$x_1(0) = 3,$$
$$x_2(0) = 0.4,$$

and the linear control

$$u = -2x_1 - 3x_2, \tag{2.141}$$

we get the response depicted in Fig. 2.33. In the left a) we see how the two states converge asymptotically to zero; however, if we inject a disturbance in the system, for example $0.8 \sin 4t$ we get the response in b); concluding that a linear control is not effective in driving the states to zero completely in the presence of disturbances. These disturbances usually come from unmodeled dynamics as these mathematical models are usually a linearization or simplification of the real system, or dynamics that are usually neglected; for example, the air in the exterior in a car, the approximation of the tire dynamics, the super-elevation in a road, the type of terrain etc. Now, if we change u by the SMC law

$$u = -\rho \text{sign}(\sigma), \tag{2.142}$$
$$\sigma = x_2 + cx_1, \tag{2.143}$$

with $c = 1.3$, $\rho = 4$. Then the response of the system that we get is the one depicted in Fig. 2.34

In Fig. 2.34 a) we see the response, converging to zero, of the state variables x_1, x_2 when the

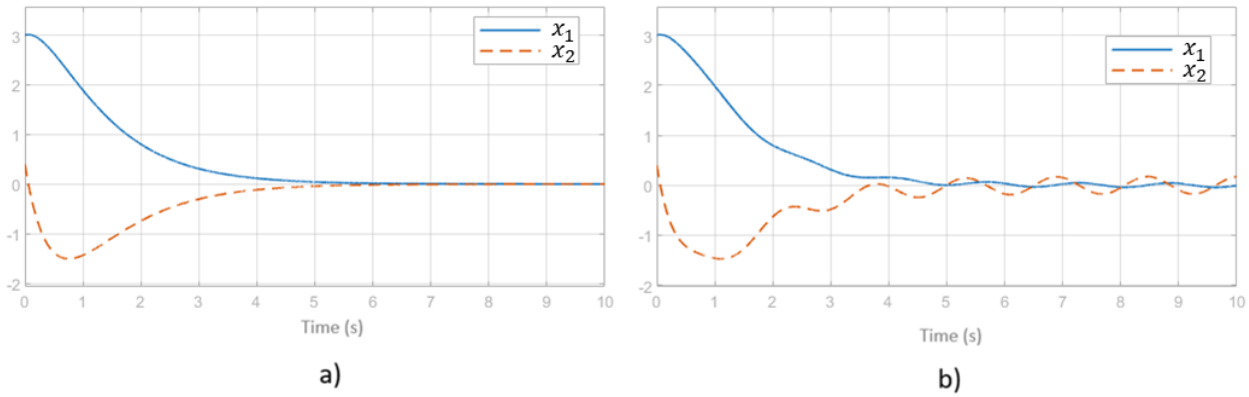


Figure 2.33: Comparison of a basic system response with a linear controller. a) with no disturbance. b) with disturbance.

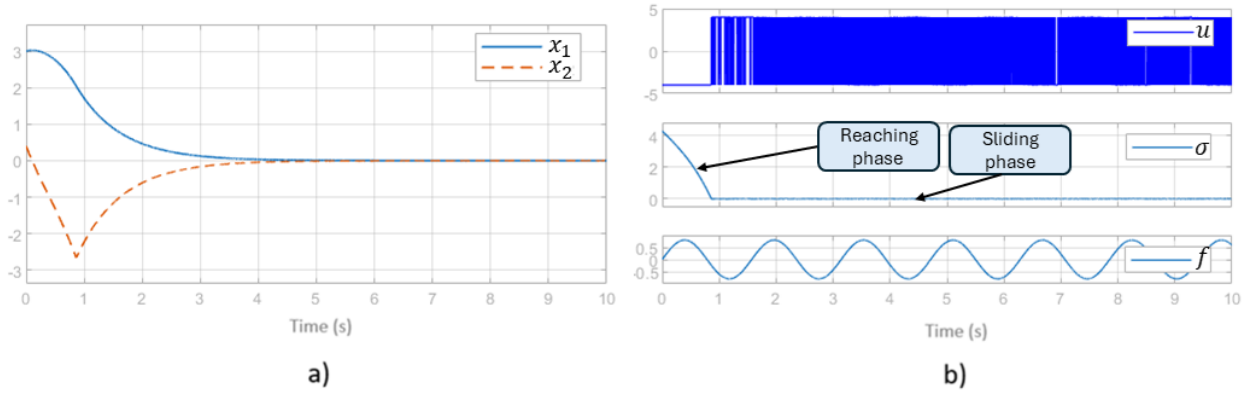


Figure 2.34: Response of the system (2.140) with the SMC control law in (2.142) applied. a) state variables response. b) Control law u, sliding variable σ, and disturbance f.

SMC (2.142) is applied. In b) we see the control law u which is switching at high frequency. The sliding variable σ can be seen how converges to zero. The period of time, in this case $0 < t < 0.9$, before the sliding variable σ reaches zero, and the control starts switching at high frequency is called *the reaching phase*. In the same graph, we can see the disturbance added to the system f as a sinusoidal signal.

In Fig. 2.34 b) in the σ graph, we can see two phases, the reaching phase and the sliding phase. The reaching phase occurs when the sliding variable approaches to the sliding surface. We can observe that, during this phase, there is no high frequency in the control u. Once the sliding variable reaches the sliding surface, the control u starts to switch rapidly. This is the sliding phase. In order

for the sliding phase to occur, it is required to meet the reachability condition:

$$\sigma\dot{\sigma} \leq -\frac{\alpha}{\sqrt{2}}|\sigma|, \tag{2.144}$$

where α is a parameter where the larger α, the shorter the reaching time. In conclusion, we can observe that the SMC technique is effective in rejecting uncertainties if they are bounded and these boundaries are known. One of the drawbacks of this technique is the chattering effect due to the high frequency switching, which in most applications is undesirable.

2.5.2.1 Equivalent Control

In sliding mode control, the equivalent control is the hypothetical continuous (average) control input that would keep the system trajectory exactly on the sliding surface once it has reached it. It is obtained by assuming ideal sliding (i.e., the sliding variable and its derivative are zero) and solving for the control that satisfies this condition using the system's nominal model. The equivalent control represents the smooth, model-based part of the sliding mode law and captures the desired system dynamics on the sliding surface, while the discontinuous switching term is added to drive the system to the surface and ensure robustness against uncertainties and disturbances.

When the system has entered in sliding mode, we have that the sliding variable and its derivative are $\sigma = \dot{\sigma} = 0$ and hence

$$\dot{\sigma} = \frac{\partial\sigma}{\partial x}\frac{\partial x}{dt} = 0. \tag{2.145}$$

Assuming we have the system

$$\dot{x} = f(x) + g(x)u + d, \tag{2.146}$$

where d is the disturbance, f and g are continuous functions, and u is the control signal, then we have with (2.145) and (2.146) that

$$\dot{x} = \frac{\partial\sigma}{\partial x}f(x) + \frac{\partial\sigma}{\partial x}g(x)u_{eq} + \frac{\partial\sigma}{\partial x}d = 0, \tag{2.147}$$

where u_{eq} is the equivalent control. Solving for u_{eq} and assuming $\frac{\partial\sigma}{\partial x}g(x)$ is invertible, we have

$$u_{eq} = -\left(\frac{\partial\sigma}{\partial x}g(x)\right)^{-1}\frac{\partial\sigma}{\partial x}f(x) - \left(\frac{\partial\sigma}{\partial x}g(x)\right)^{-1}\frac{\partial\sigma}{\partial x}d. \tag{2.148}$$

Here, we can observe that the disturbance is present in the definition of the equivalent control. and hence, this definition is not implementable since it is supposed to be unknown d although we know its limits. This equivalent control concept is useful for analyzing the stability of the reduced-order system once the sliding mode is reached. A numerical example is provided in [76].

2.5.3 Pole Placement

The pole placement technique for the conventional sliding mode control (SMC) is used to design the system dynamics on the sliding surface, that is, to define how the system behaves once it reaches that surface. Essentially, the sliding surface (e.g., s=Cx) is chosen so that the reduced-order dynamics of the system have poles located at specific positions in the complex plane, ensuring stability, desired response speed, and damping. Thus, pole placement does not directly design the discontinuous control law, but rather shapes the internal dynamics of the system during the sliding regime, guaranteeing a stable and predictable behavior once sliding motion is established. An alternative approach for shaping these dynamics is the Linear Quadratic Regulator (LQR), which determines the sliding surface by minimizing a quadratic cost function that balances state deviation and control effort, leading to an optimal but less directly interpretable pole distribution. In contrast, pole placement offers explicit control over the closed-loop eigenvalues, making the dynamic specifications more transparent and intuitive. For this reason, in the following we focus on and explain the pole placement approach.

2.5.3.1 Regular Form

In conventional sliding mode control, converting the system to regular (normal) form is necessary because eigenvalue (pole) placement is performed on the equivalent dynamics that govern the motion on the sliding surface. The regular form explicitly separates the control-affected state from the internal dynamics, yielding a reduced-order system whose dynamics on the sliding manifold are linear and controllable. This makes the system matrix on the sliding surface well defined, allowing the desired eigenvalues to be assigned meaningfully; without this transformation, the sliding dynamics are not explicit, and eigenvalue placement cannot be correctly applied.

Let us consider the nth-order linear time-invariant system with m inputs

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{2.149}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $1 \leq m \leq n$. We want to design the switching function

$$\sigma = Sx(t), \tag{2.150}$$

with $S \in \mathbb{R}^{m\times n}$ and $\sigma : \mathbb{R} \to \mathbb{R}^m$.

The procedure is as follows, applying the procedure described in section 2.1.2.2. Assume B is full rank = m. Then, there exists an orthogonal matrix $T_r \in \mathbb{R}^{n\times n}$ such that

$$T_r B = \begin{bmatrix} 0 \\ B_2 \end{bmatrix}, \tag{2.151}$$

with $B_2 \in \mathbb{R}^{m\times m}$ and it is nonsingular. There are several ways to find a transformation T_r. One way is the so called QR factorization as mention in section 2.1.2.1.

Let us define the new coordinate system based on the transform T_r:

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = T_r x, \tag{2.152}$$

where $z_1 \in \mathbb{R}^{n-m}$, that is, the elements where there is no control signal u acting, and $z_2 \in \mathbb{R}^m$, that is, the portion where the control is acting. Then transforming the system with the matrices A, B as follows

$$A \leftrightarrow T_r A T_r^T = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix}, \tag{2.153}$$

$$B \leftrightarrow T_r B, \tag{2.154}$$

and the system obtains the form

$$\dot{z}_1(t) = \alpha_{11} z_1(t) + \alpha_{12} z_2(t), \tag{2.155}$$

$$\dot{z}_2(t) = \alpha_{21} z_1(t) + \alpha_{22} z_2(t) + B_2 u(t), \tag{2.156}$$

where $\alpha_{11}, \alpha_{12}, \alpha_{21}, \alpha_{22}$ are submatrices of a partitioned matrix.

In this new coordinated system, the matrix that defines the switching function becomes

$$S T_r^T = [\, S_1 \quad S_2 \,] \tag{2.157}$$

with $S_1 \in \mathbb{R}^{m\times(n-m)}$ and $S_2 \in \mathbb{R}^{m\times m}$. Now, during sliding motion we have that

$$S_1 z_1(t) + S_2 z_2(t) = 0, \tag{2.158}$$

then, solving (2.158) for z_2 and combining with (2.155) we have

$$\begin{aligned} z_2(t) &= -M z_1(t), \\ \dot{z}_1(t) &= (\alpha_{11} - \alpha_{12} M) z_1(t), \end{aligned} \tag{2.159}$$

where $M = S_2^{-1} S_1$. We can see how z_1 acts as a control for z_2. The switching function now can be found with the next structure

$$S = S_2 \, [\, M \quad I_{m\times m} \,] \, T_r. \tag{2.160}$$

As example, let us convert the system for the lateral dynamics of an automotive vehicle in (3.62) but using only the B_1 matrix, and for the matrix A we will use only A_{ij} for the coefficients. This is a system with only one control signal δ; however, the method can be used for MIMO systems. Then we have

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & A_{22} & A_{23} & A_{24} \\ 0 & 0 & 0 & 1 \\ 0 & A_{42} & A_{43} & A_{44} \end{pmatrix} \tag{2.161}$$

$$B_1 = \begin{pmatrix} 0 \\ B_{1,2} \\ 0 \\ B_{1,4} \end{pmatrix}. \tag{2.162}$$

(2.163)

Applying the QR factorization to the B_1 matrix, we obtain the transform matrix whose rank is 4

$$T_r = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & \frac{{B_{1,4}}^2}{|B_{1,4}|\sqrt{{B_{1,2}}^2+{B_{1,4}}^2}} & 0 & -\frac{B_{1,2}\, B_{1,4}}{|B_{1,4}|\sqrt{{B_{1,2}}^2+{B_{1,4}}^2}} \\ 1 & 0 & 0 & 0 \\ 0 & \frac{B_{1,2}}{\sqrt{{B_{1,2}}^2+{B_{1,4}}^2}} & 0 & \frac{B_{1,4}}{\sqrt{{B_{1,2}}^2+{B_{1,4}}^2}} \end{pmatrix}, \tag{2.164}$$

and we can observe that

$$\hat{B}_1 = T_r B_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \sqrt{{B_{1,2}}^2 + {B_{1,4}}^2} \end{pmatrix}, \tag{2.165}$$

and $\hat{A} = T_r A T_r^T =$

$$\begin{pmatrix} 0 & -\frac{B_{1,2}\,|B_{1,4}|}{B_{1,4}\,\sigma_5} & 0 & \frac{B_{1,4}}{\sigma_5} \\ \frac{A_{23}\,{B_{1,4}}^2}{|B_{1,4}|\,\sigma_5} - \frac{A_{43}\,B_{1,2}\,B_{1,4}}{|B_{1,4}|\,\sigma_5} & \frac{|B_{1,4}|\,\sigma_2}{\sigma_5} - \frac{B_{1,2}\,|B_{1,4}|\,\sigma_1}{B_{1,4}\,\sigma_5} & 0 & \frac{B_{1,2}\,\sigma_2}{\sigma_5} + \frac{B_{1,4}\,\sigma_1}{\sigma_5} \\ 0 & \frac{|B_{1,4}|}{\sigma_5} & 0 & \frac{B_{1,2}}{\sigma_5} \\ \frac{A_{23}\,B_{1,2}}{\sigma_5} + \frac{A_{43}\,B_{1,4}}{\sigma_5} & \frac{|B_{1,4}|\,\sigma_4}{\sigma_5} - \frac{B_{1,2}\,|B_{1,4}|\,\sigma_3}{B_{1,4}\,\sigma_5} & 0 & \frac{B_{1,2}\,\sigma_4}{\sigma_5} + \frac{B_{1,4}\,\sigma_3}{\sigma_5} \end{pmatrix}, \tag{2.166}$$

where

$$\begin{aligned} \sigma_1 &= \frac{A_{24}\,{B_{1,4}}^2}{|B_{1,4}|\ \sigma_5} - \frac{A_{44}\,B_{1,2}\,B_{1,4}}{|B_{1,4}|\ \sigma_5}, \\ \sigma_2 &= \frac{A_{22}\,{B_{1,4}}^2}{|B_{1,4}|\ \sigma_5} - \frac{A_{42}\,B_{1,2}\,B_{1,4}}{|B_{1,4}|\ \sigma_5}, \\ \sigma_3 &= \frac{A_{24}\,B_{1,2}}{\sigma_5} + \frac{A_{44}\,B_{1,4}}{\sigma_5}, \\ \sigma_4 &= \frac{A_{22}\,B_{1,2}}{\sigma_5} + \frac{A_{42}\,B_{1,4}}{\sigma_5}, \\ \sigma_5 &= \sqrt{{B_{1,2}}^2 + {B_{1,4}}^2}, \end{aligned} \tag{2.167}$$

Then we have our system in regular form, where δ is the steering angle,

$$\dot{z} = \hat{A}z + \hat{B}_1\delta, \text{ or} \tag{2.168}$$

$$\dot{z}_1(t) = \hat{A}_{11}z_1(t) + \hat{A}_{12}z_2(t), \tag{2.169}$$

$$\dot{z}_2(t) = \hat{A}_{21}z_1(t) + \hat{A}_{22}z_2(t) + \hat{B}_{1,2}\delta.$$

where the partitions of the matrix $\hat{A} \in \mathbb{R}^{n\times n}$ correspond, in this case, to $\hat{A}_{11} = A(1:3, 1:3)$, $\hat{A}_{12} = A(1:3, 4)$, $\hat{A}_{21} = A(4:4, 1:3)$, and $\hat{A}_{22} = A(4, 4)$.

2.5.3.2 Eigenvalue placement

Considering the system in the regular form as in (2.155), we have that, during the sliding motion, we have the reduced order system as in (2.159) and then, we want to find the switching function matrix S, which guarantees the stability in the sliding motion, designing the M matrix in the regular form system in (2.160). For this purpose, we can apply the eigenvalue (pole) placement method. Another method is Linear-quadratic minimization (LQR) which can be applied. However, this will not be covered in this dissertation.

The eigenvalue placement method consists in finding the characteristic polynomial of the matrix

$$(\alpha_{11} - \alpha_{12}M), \tag{2.170}$$

in the reduced dynamical system (2.159), and designing the matrix M by comparison with the desired characteristic equation, having the matrix $(\alpha_{11} - \alpha_{12}M)$ stable eigenvalues in this way.

Consider the canonical characteristic equation

$$\lambda^{n-1} + s_{n-1}\lambda^{n-2} + ... + s_2\lambda + s_1 = 0, \tag{2.171}$$

where $s_1, ...s_n$ are scalars chosen to make the polynomial Hurwitz, then we have to find the elements of M such that, the coefficients of

$$|\lambda I - (\alpha_{11} - \alpha_{12}M)|, \tag{2.172}$$

are equal to the coefficients of the canonical equations.

After finding the coefficients of the matrix M we need to covert this coefficients in the regular form coordinates to the original coordinates by using the transform matrix T_r in the form (2.160) that it is mention here again

$$S = S_2 \left[\, M \quad I_{m\times m} \, \right] T_r. \tag{2.173}$$

The complete example in this section was run using symbolic math script in Matlab. Since the example was done symbolically, the resultant equations are large. However, in practice, it is solved numerically. See the algorithm 13 in the appendix.

2.5.4 Integral Sliding Mode Control

Integral Sliding Mode (ISM) is a variant of sliding mode control designed to eliminate the reaching phase and ensure robustness from the initial time. Unlike conventional sliding mode control, where the system must first reach the sliding surface before robustness is guaranteed, ISM defines a sliding manifold that includes an integral term of the system states or tracking error, constructed so that the system starts on the sliding surface at $t = 0$. As a result, matched disturbances and model uncertainties are compensated immediately, improving transient performance and reducing sensitivity to initial conditions. This makes integral sliding mode especially attractive for systems requiring strong robustness throughout the entire response, not only after the sliding condition is achieved.

Despite its advantages, ISM also has several drawbacks. The design of the integral sliding manifold is more complex than in conventional sliding mode control and typically requires accurate knowledge of the nominal system model; model mismatch can degrade performance. The presence of integral terms may lead to windup or large control transients if not carefully handled, especially under actuator saturation. In addition, ISM does not inherently eliminate chattering, so additional techniques (such as boundary layers or higher-order sliding modes) are often needed. Finally, implementation can be more demanding in practice due to increased computational effort and sensitivity to measurement noise in the integrated signals.

Without loss of generality, let us explain the linear form of the method although it works even in non linear systems.

Let us consider the LTI system

$$\dot{x}(t) = Ax(t) + B(u_0(t) + u_1(t)) + \phi(t, x), \quad \phi(t, x) = B\xi(t, x), \tag{2.174}$$

where ϕ is the matching uncertainties of the system, u_0 is the nominal control designed to compensate the ideal system without the disturbance. The method for designing u_0 can be pole placement, Ackermann formula, by blocks etc.

Then we need to define the σ function

$$\sigma(x) = G(x(t) - x(0)) - G\int_0^t (Ax(\tau) + Bu_0(\tau))d\tau, \tag{2.175}$$

that basically means $\sigma(x) =$ (Actual response of the system) - (nominal response assuming u_0). Where $G \in \mathbb{R}^{m\times n}$ is the projection matrix with

$$\begin{aligned} det(GB) &\neq 0, \\ \dot{\sigma}(x) &= GB(u_1 + \xi), \end{aligned} \tag{2.176}$$

and the control u_1 is design as

$$u_1 = -\rho(t,x)\frac{(GB)^T\sigma}{||(GB)^T\sigma||}. \tag{2.177}$$

The demonstration of the method in explained in [76].

2.5.5 High-Order Sliding Mode Control

Conventional sliding mode control (SMC) enforces the system trajectories to reach and remain on a predefined sliding surface by using a discontinuous control action, which guarantees robustness against matched uncertainties and external disturbances. However, this discontinuity typically leads to the well-known chattering phenomenon, caused by high-frequency switching in the control signal. High-order sliding mode (HOSM) control extends the conventional SMC framework by enforcing the sliding condition not only on the sliding variable itself but also on its higher-order time derivatives. By doing so, HOSM algorithms, such as the super-twisting algorithm, achieve finite-time convergence while providing a continuous or smoother control input, significantly reducing chattering without sacrificing robustness. As a result, high-order sliding mode control is particularly well suited for systems with actuator dynamics and practical implementation constraints.

2.5.5.1 Super-Twisting Controller

The super-twisting algorithm is a second-order sliding mode control technique designed for systems with relative degree one with respect to the sliding variable (σ). Unlike conventional sliding

mode control, it generates a continuous control signal while ensuring finite-time convergence of both the sliding variable and its first time derivative to zero. The algorithm achieves robustness against bounded matched uncertainties without requiring measurements of the derivative of the sliding variable ($\dot{\sigma}$). Due to its reduced chattering and strong robustness properties, the super-twisting algorithm is particularly suitable for practical implementations involving actuator dynamics and mechanical systems.

Let us assume we have the dynamic system of relative degree 1

$$\dot{x} = a(t,x) + b(t,x)u, \quad \sigma = \sigma(t,x), \tag{2.178}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}$ and σ is the only signal we can measure. The smooth function a, b, σ are unknown. Then we assume that

$$\dot{\sigma} = h(t,x) + g(t,x)u, \tag{2.179}$$

then for some positive constants C, K_M, K_m, U_M, q we have

$$|\dot{h}| + U_M|\dot{g}| \leq C, \quad 0 \leq K_m \leq g(t,x) \leq K_M,$$
$$|h/g| < qU_M, \quad 0 < q < 1,$$

and defining the control

$$u = -\lambda|\sigma|^{1/2}\text{sign}(\sigma) + u_1, \tag{2.180}$$
$$\dot{u}_1 = \begin{cases} -u, & |u| > U_M \\ -\alpha\text{sign}(\sigma), & |u| \leq U_M \end{cases}$$

then we have the next theorem, taken from [76]:

Theorem 3. *with $k_m\alpha > C$ and λ sufficiently large, the controller guarantees the appearance of a 2-sliding mode $\sigma = \dot{\sigma} = 0$ in system (2.179), which attracts the trajectories in fine time. The control u enters in finite time the segment $[-U_M, U_M]$ and stays there. It never leaves the segment , if the initial value is inside at the beginning.*

2.5.5.2 Levant Differentiator

The Levant differentiator is a method from control theory used to estimate signal derivatives in real time with high accuracy and robustness. Unlike classical numerical differentiation, which amplifies measurement noise, this method relies on sliding mode techniques that make it more resistant to disturbances.

Originally introduced by Arie Levant, the method is based on finite-time stability and variable structure systems. Its main advantage is the ability to estimate derivatives without significantly amplifying noise, which makes it useful in applications such as control system design, state estimation, and fault detection.

In nonlinear control systems, particularly those operating in uncertain or noisy environments, the Levant differentiator is widely used in fields such as robotics, autonomous vehicles, and aerospace engineering.

Arie Levant defines an exact differentiator D as one whose output coincides with the derivative of the input signal $f(t)$. A differentiator is called robust if its output tends uniformly to $Df(t)$ when the input tends uniformly to $f(t)$ [42]. A differentiator is therefore called correct if it is both exact and robust. In practice, the differentiator must be tuned so that it is not overly sensitive to noise while still accurately estimating the derivative of the desired signal.

Below are the equations to implement the differentiator that can be found in [41],

$$\dot{z}_0 = -\lambda_r L^{1/r} |z_0 - \sigma|^{(r-1)/r} \text{sign}(z_0 - \sigma) + z_1, \tag{2.181}$$

$$\dot{z}_k = -\lambda_(r-k) L^{1/(r-k)} |z_k - \dot{z}_{(k-1)}|^{(r-k-1)/(r-k)} \text{sign}(z_k - \dot{z}_{k-1}) + z_{k+1}, \tag{2.182}$$

$$\dot{z}_{k-1} = -\lambda_1 L \text{sign}(z_{r-1} - \dot{z}_{r-2}), \tag{2.183}$$

where $k = 1, ..., r-2$, r is the relative degree, and λ_x are the gains to be tuned. The algorithm 4 shows the implementation of this differentiator of degree two.

Fig. 2.35 shows the result of this differentiator. The input signal is the sine function and the output results to be cosine.

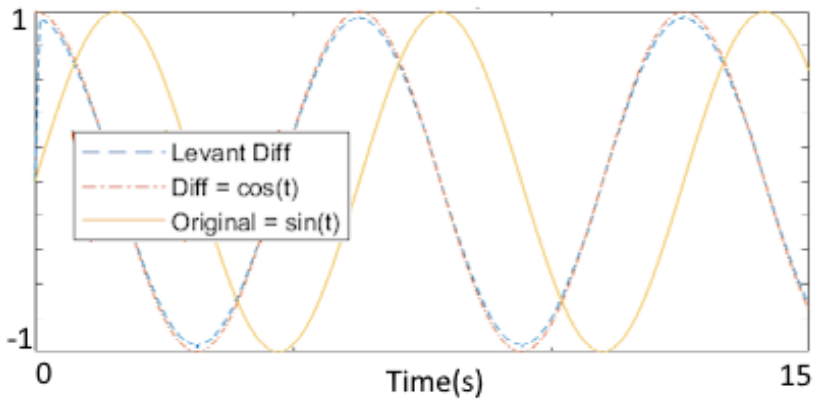


Figure 2.35: Diagram showing the Levant differentiator performance.

3. Dynamic Model of an Electric Vehicle and Conventional Vehicle

3.1. Vehicle Dynamics

Vehicle dynamics is a field that studies the physics of automotive vehicles, focusing on the forces that cause and influence their movement. These forces arise from numerous factors, the most prominent being the vehicle's propulsion system. It is used the term propulsion system because modern vehicles utilize different systems, such as spark-ignition (SI) engines, diesel engines, electric motors, hydrogen fuel cells, and more. Traditional internal combustion engines (SI, diesel) are gradually being phased out in favor of cleaner alternatives, like electric motors and hydrogen-based propulsion, due to environmental concerns. These propulsion systems generate the force required to propel the vehicle, which is transmitted to the driven tires. The interaction between the tires and the road surface creates the actual traction force that moves the vehicle forward.

During driving, air resistance (aerodynamic drag) works to reduce the vehicle's speed. On inclines, gravity acts either to slow down or speed up the vehicle, depending on whether it is going uphill or downhill. The friction in the tires opposes their rotation, limiting acceleration. During cornering, the inertia of the vehicle's mass creates lateral forces that push the center of gravity (CoG) towards the outside of the turn, potentially leading to rollover if the forces are too great. Steering inputs generate lateral forces in the tires, allowing the vehicle to change its direction of travel. These are just a few of the many forces acting on a vehicle in motion which the vehicle dynamics field is in charge to study and analyze.

Analyzing these forces and dynamics, we can find more ways of vehicle construction, create more efficient motors, design controllers to make safer and comfortable trips. It can help us to improve the autonomous mobility, etc. In this dissertation, we will treat the tire dynamics which studies the forces that involve the wheels, longitudinal dynamics which deals with the forces and

motion along the vehicle's forward and backward axis. This includes acceleration, deceleration, braking, and how power from the engine or electric motors is transmitted to the wheels, and lateral dynamics which are involved in the cornering and steering of the vehicle.

Studying this field, we'll have the knowledge to be applied in automotive applications such as adaptive cruise control (ACC), forward collision warning (FCW), evasive maneuver assist (EMA), lane departure warning (LDM), emergency steering assist (ESA), among many other systems that are known as advanced driving assistance systems (ADAS).

3.1.1 Longitudinal Dynamics

In this chapter, we analyze longitudinal vehicle dynamics, which are central to many automotive safety systems, including adaptive cruise control and automatic emergency braking. This analysis is key to understanding how vehicles accelerate, decelerate, and maintain speed under different driving conditions.

We explore various aspects of longitudinal dynamics, focusing on how factors such as road incline, vehicle mass, and external forces influence vehicle performance. By examining these dynamics, we gained insight into the behavior of vehicles during acceleration, braking, and steady-state motion, providing a foundation for the design of more efficient and safer control systems.

The insights from this analysis are particularly relevant for the development of ADAS systems that rely on accurate longitudinal dynamics models. Below is a list of ADAS systems that benefit from a deeper understanding of longitudinal vehicle dynamics.

- Adaptive Cruise Control (ACC): Automatically adjusts the vehicle's speed to maintain a safe distance from the car in front, managing longitudinal dynamics during cruising.
- Automatic Emergency Braking (AEB): Detects an imminent collision and applies the brakes to avoid or mitigate the impact, improving longitudinal deceleration control.
- Forward Collision Warning (FCW): Provides alerts to the driver of a potential frontal collision, helping the driver take corrective action in longitudinal dynamics.
- Traffic Jam Assist (TJA): Controls acceleration and braking in stop-and-go traffic, managing

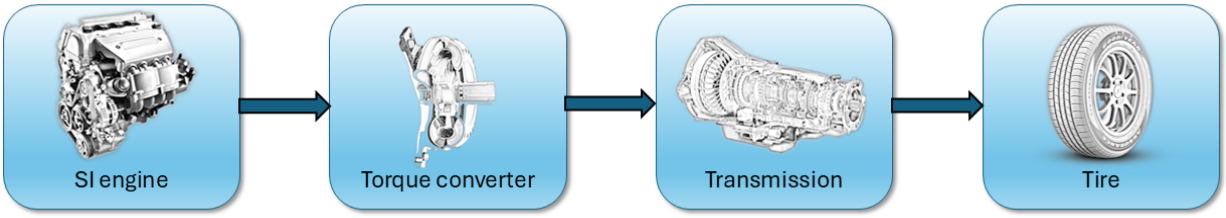


Figure 3.1: Components of the driveline of a car with SI engine. It is shown the main components in charge to transmit the power of the motor to the wheels. The engine generates the power that is transmitted to the transmission through the torque converter which transmits the torque using a fluid to allow a complete stop without stalling the motor or passing the torque or even multiply it when the motor accelerates. The transmission transfers the torque to the wheels, coupling the velocities using a set of different gears.

longitudinal dynamics in slow-moving conditions.

- Collision Avoidance System (CAS): Uses longitudinal control to automatically brake and prevent collisions with obstacles.
- Hill Start Assist: Prevents the vehicle from rolling back when starting on an incline, maintaining control of the longitudinal motion.
- Evasive Maneuver Assist (EMA): Helps manage longitudinal forces by adjusting the speed and deceleration during emergency evasive maneuvers.

The longitudinal dynamics are basically related to the subsystem of the vehicle that is called *"the vehicle driveline"* and it is responsible for transmitting power from the engine to the wheels, enabling movement. It includes components such as the engine, transmission, torque converter or clutch, driveshaft, differential, and axles. The components of the driveline are shown in Fig.3.1 according to [72]. This is the path that the torque follows from its generation in the combustion engine to the wheels, where the force generated from the rubber of the tires against the road propels the vehicle forward. The components analyzed here are:

a) **The combustion engine.** The mathematical model analyzed here is called the mean value model. It predicts the mean value of the major external engine variables [72], such as crankshaft speed and manifold pressure.

b) **The torque converter.** This component couples the engine with the transmission through a fluid.

c) **The transmission.** This element converts the high speed of the crankshaft into a strong torque to the wheels by using different sets of gears.

d) **The wheel dynamics.** This refers to the modeling of the forces involved in the tires due to the interaction of the rubber with the road, the rim, etc.

e) **The longitudinal dynamics.** The forces generated by the tires cause the movement of the vehicle. However, other forces are involved in this dynamic, such as air drag, road inclination, rolling resistance, etc.

3.1.1.1 The Longitudinal Vehicle Model

The primary vehicle dynamic equation is based in the Newton's second law:

$$m\ddot{x} = F_{xf} + F_{xr} - F_{aero} - R_{xf} - R_{xr} - mg\sin(\theta). \tag{3.1}$$

If we look at the diagram in Fig. 3.2 we can see how the Newton's second law is met. The force that pushes the vehicle forward $m\ddot{x}$ is equal to the sum of all forces acting on the vehicle, that is, the force produced by the front tires F_{xf}, for the case of front wheel drive, against the ground, plus the force produced by the rear tires F_{xr} in the case of rear wheel drive, minus the forced produced by the air drag against the vehicle movement F_{aero}, minus the rolling resistance forces R_{xf}, R_{xr}, that are the forces, such as friction, that prevent the wheels from turning, and the weight component $mg\sin(\theta)$ in the opposition or in favor to the movement. Now, let us see each element,

Front longitudinal force:

$$F_{xf} = C_{\sigma f}\sigma_{xf}, \tag{3.2}$$

where

$$\sigma_{xf} = \begin{cases} \frac{r_{eff}\omega_{wf} - \dot{x}}{\dot{x}} & \text{during braking,} \\ \frac{r_{eff}\omega_{wf} - \dot{x}}{r_{eff}\omega_{wf}} & \text{during acceleration.} \end{cases} \tag{3.3}$$

Rear longitudinal force:

$$F_{xr} = C_{\sigma r}\sigma_{xr}, \tag{3.4}$$

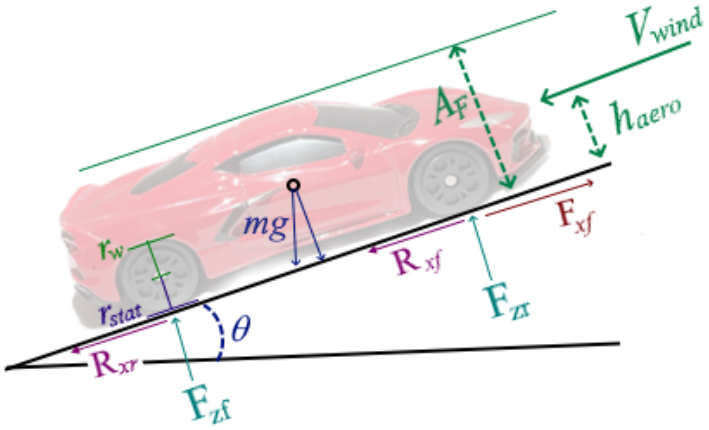


Figure 3.2: Different forces acting over the vehicle. F_{aero} is the air drag force. R_x is the rolling resistance. F_x is the longitudinal force. Mass and gravity are denoted by m and g.

where

$$\sigma_{xr} = \begin{cases} \frac{r_{eff}\omega_{wr} - \dot{x}}{\dot{x}} & \text{during braking,} \\ \frac{r_{eff}\omega_{wr} - \dot{x}}{r_{eff}\omega_{wr}} & \text{during acceleration.} \end{cases} \tag{3.5}$$

Here C_σ is a constant called the tire stiffness and σ_x is the slip ratio that is the ratio between the actual vehicle velocity and the equivalent velocity given by the angular velocity of the wheels per the effective tire radius. This relation is good only for a small slip ratio, less than 0.1. In following chapters, it will be analyzed a more precise equation that is called "the magic formula" which approximates more to the real-life curves according to the different parameters of the tires.

Rolling resistance, which is the opposition to the rolling of the wheels due to the interaction of the rubber on ground, the normal load, friction etc. that is,

$$R_{xf} + R_{xr} = f(F_{zf} + F_{zr}), \tag{3.6}$$

where f is not a function, it is a proportional constant and F_{zf} is the front normal tire force, that is

$$F_{zf} = \frac{-F_{aero}h_{aero} - m\ddot{x}h - mg\sin(\theta) + mgl_r\cos(\theta)}{l_f + l_r}, \tag{3.7}$$

and the rear normal tire is

$$F_{zr} = \frac{F_{aero}h_{aero} + m\ddot{x}h + mg\sin(\theta) + mgl_f\cos(\theta)}{l_f + l_r}, \tag{3.8}$$

where F_{aero} is the aerodynamic drag force, h_{aero} is the height at which equivalent aerodynamic drag

force acts, θ is the angle of inclination of the road on which the vehicle is traveling.

The aerodynamic drag force:

$$F_{aero} = \frac{1}{2}\rho C_d A_F \left(\dot{x} + V_{wind}\right)^2, \tag{3.9}$$

where ρ is mass density of air, C_d is the aerodynamic drag coefficient, A_F is the frontal area of the vehicle, and V_{wind} is wind velocity.

3.1.1.2 Tire Force and The Magic Formula

Advanced driving systems more oriented to the tires can be named among others:

- Traction Control System (TCS): Monitors and adjusts wheel slip during acceleration, optimizing tire-road interaction for improved traction in longitudinal dynamics.
- Anti-lock Braking System (ABS): Prevents the wheels from locking up during braking, improving tire traction and control during deceleration.
- Electronic Stability Control (ESC): Monitors tire slip and lateral stability, automatically adjusting braking and engine power to prevent skidding and maintain tire grip.
- Tire Pressure Monitoring System (TPMS): Alerts the driver if tire pressure is outside the optimal range, ensuring the tires maintain their designed performance for proper traction and handling.
- Torque Vectoring System: Adjusts the power delivered to individual wheels to improve cornering performance, optimizing tire dynamics for better lateral control.
- Active Suspension Systems: Continuously adjusts suspension settings to maintain optimal tire contact with the road surface, improving both lateral and longitudinal stability.

For all these devices a comprehension of the tires physics is crucial. Important concepts related to this that we can find are the tire slip defined in (3.3) and (3.5), the normal force, the rolling resistance, the longitudinal and lateral forces... and precisely these forces this chapter is about. As explained in section 3.1.1.1 the linear formula given in the base of the wheel slip is not good for slip ratios greater than 0.1, then the magic formula (MF) can be used. The formula is explained in [28][62][63]. In Fig.3.3 we can appreciate that the force is cuasi linear for slip values below 0.05

and has an asymptotic behavior for greater values. This graph was obtained with experimental data. The MF approximates this behavior depending on the parameters used. These parameters will be discussed at continuation. The following is the "magic formula"

$$y(x) = D \sin\left\{C \tan^{-1}\left[Bx - E\left(Bx - \tan^{-1}(Bx)\right)\right]\right\}, \tag{3.10}$$

where the parameters are B the stiffness factor, C the shape factor, D the peak value, E the curvature factor; "y" can be F_x or F_y that is the longitudinal and lateral forces respectively, and x is the slip component. Among other properties we have y_m that is the relative maximum of the function and y_a is the value of the horizontal asymptote. Among several properties of this function we have the relative maximum $y_m = y(x_m) = D$ where x_m meets (3.11) as follows:

$$B(1 - E)x_m + E \tan^{-1}(Bx_m). \tag{3.11}$$

And the horizontal asymptote is:

$$y_a = \lim_{x \to +\infty} y(x) = D \sin\left(\frac{C\pi}{2}\right), \tag{3.12}$$

Now, the procedure to get the parameters from the graph in Fig.3.3 as example, would be as follows:

$$D = y_m, \tag{3.13}$$

$$C = 2 - \frac{2}{\pi} \tan^{-1}\left(\frac{y_a}{D}\right), \tag{3.14}$$

$$B = \frac{y'(0)}{CD}, \tag{3.15}$$

$$E = \frac{Bx_m - \tan\left(\frac{\pi}{2C}\right)}{Bx_m - \tan^{-1}(Bx_m)}. \tag{3.16}$$

Recall that $y_m, x_m, y_a, y'(0)$ are values taken from the graph obtained from experimental data, maximum value, asymptotic value, x_m in the asymptotic value, and the slope value in the point where the slip is 0, respectively. Another way is to use typical values tables such as in Table 3.1, taken from [49], however, these values are for a different form of the MF where the normal load is considered as

$$F_x = f(\kappa, F_z) = F_z \cdot D \cdot \sin\left(C \cdot \arctan\left(B\kappa - E\left[B\kappa - \arctan\left(B\kappa\right)\right]\right)\right), \tag{3.17}$$

where κ is the longitudinal tire slip, F_x the longitudinal force, and F_z is the normal force over the tire.

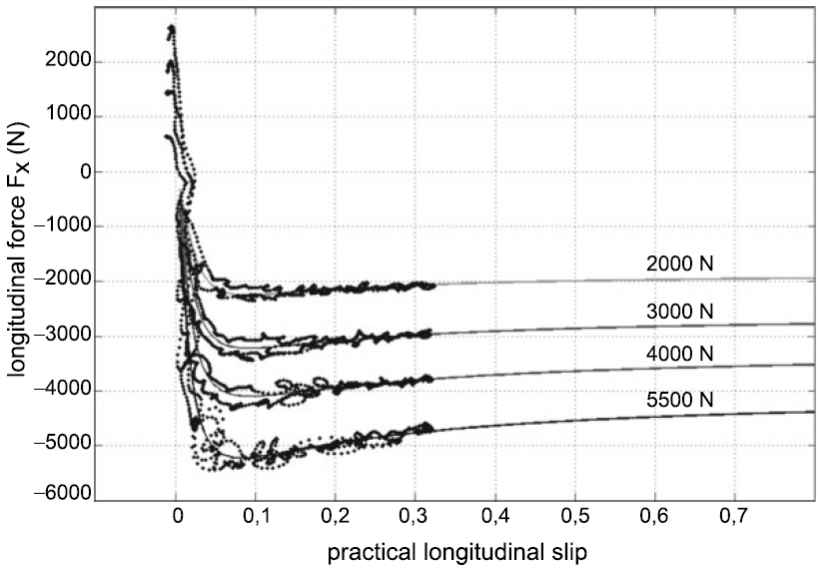


Figure 3.3: Experimental graph of the longitudinal force of the tires vs tire slip at a fixed normal load. Here we can appreciate the maximum force F_x according the normal loads such as 2000N,3000N, 4000, 5500N. Taken from [28].

TABLE 3.1. TYPICAL VALUES FOR THE MAGIC FORMULA ACCORDING ROAD CONDITIONS

Surface	**B**	**C**	**D**	**E**
Dry tarmac	10	1.9	1	0.97
Wet tarmac	12	2.3	0.82	1
Snow	5	2	0.3	1
Ice	4	2	0.1	1

3.1.1.3 The Wheel Dynamics

The dynamic of the wheels is described by [72] as follows for the drive wheel:

$$I_w \dot{\omega}_f = T_{wheel} - r_{eff} F_{xf}, \tag{3.18}$$

where ω_f is the angular velocity, I_w is the momentum of the wheel, T_{wheel} is the torque applied to the wheels coming from the engine, r_{eff} is the effective radius of the tire, and F_{xf} is the longitudinal tire force explained in section 3.1.1.2. The non-driven wheels would be as follows assuming the rear

wheels are the non-driven:

$$I_w\dot{\omega}_r = -r_{eff}F_{xr}. \tag{3.19}$$

The effective radius is calculated as follows:

$$r_{eff} = \frac{\sin\left(\cos^{-1}\left(\frac{r_{stat}}{r_w}\right)\right)}{\cos^{-1}\left(\frac{r_{stat}}{r_w}\right)}r_w, \tag{3.20}$$

where r_{stat} is the height from the ground to the axis of the wheel with load, it is explained in [72] how to calculate it based on the normal force F_z, r_w is the radius of the wheel without load.

3.1.1.4 The Torque Converter

The torque converter connects the engine with the transmission with a fluid. When the engine is turning slow, only a small amount of torque is transmitted to the transmission, and so, keeping the car still requires a light pressure in the brake pedal [72]. At higher speeds, the transmission catches up to the engine until reaching almost the same speed. Ideally the transmission should move at the same speed as the engine, since this difference is a waste of power. To counter this, some torque converters have a lockup clutch. When the two speeds, the pump and the turbine, are getting closer, this clutch locks them together, eliminating the slippage between both [72]. The main components of the torque converted are the pump or impeller shaft, the turbine, and the transmission fluid. The pump of the torque converter is attached to the flywheel of the engine. The turbine is attached to the transmission. The torque converter model presented here is the static model of Kotwicki (1982). The following equations are the model for the torque converter in converter mode (i.e. $\omega_t/\omega_p < 0.9$)

$$T_p = 3.4325 \cdot 10^{-3} \cdot \omega_p^2 + 2.2210 \cdot 10^{-3} \cdot \omega_p \cdot \omega_t - 4.6041 \cdot 10^{-3} \cdot \omega_t^2, \tag{3.21}$$

$$T_t = 5.7656 \cdot 10^{-3} \cdot \omega_p^2 + 0.3107 \cdot 10^{-3} \cdot \omega_p \cdot \omega_t - 5.4323 \cdot 10^{-3} \cdot \omega_t^2, \tag{3.22}$$

T_p is the torque in the pump, T_t is the torque in the turbine, ω_p is the angular speed of the pump, and ω_t is the angular speed of the turbine. For fluid coupling mode (i.e. $\omega_t/\omega_p >= 0.9$) the torques are:

$$T_t = T_p = -6.7644 \cdot 10^{-3} \cdot \omega_p^2 + 32.0024 \cdot 10^{-3} \cdot \omega_p \cdot \omega_t - 25.2441 \cdot 10^{-3} \cdot \omega_t^2 \tag{3.23}$$

The Fig.3.4 shows a simulation of the equation in fluid coupling mode of the torque converter.

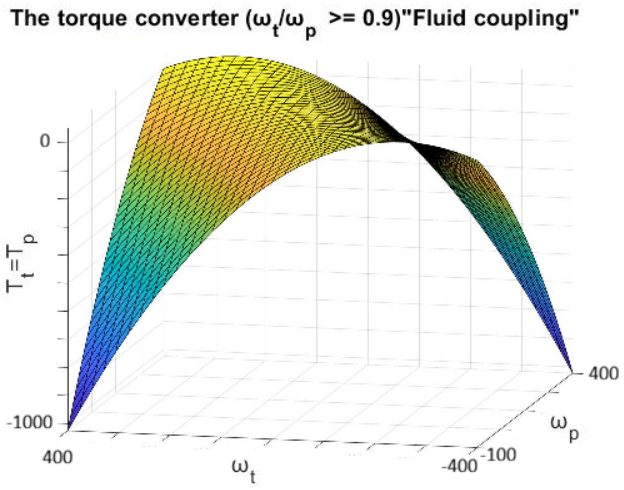


Figure 3.4: Torque converter Kotwicki model. This is a simulation of the fluid coupling mode ($\omega_t/\omega_p >= 0.9$). most part of the function grows rapidly to the negative.

Here is possible to see the small range where the function is useful in order to include this model in a wider model that simulates the car.

3.1.1.5 The Transmission

Basically, the transmission is only a gear set to increase or decrease the input torque or angular speed. For the transmission dynamics, here is presented a simplified model. More complex models make use of state machines to simulate the gear changes, in the case of several rates; however, our focus will be on electrical motors for EVs, which gearbox only require one gear ratio or even no gearbox at all. The equations for the transmission used here are:

$$T_{wheel} = \frac{1}{R} T_t, \tag{3.24}$$

$$\omega_t = \frac{1}{R} \omega_w, \tag{3.25}$$

where R is the gear ratio and depends on the operation gear.

Combustion engine vehicles require more complex transmissions with several ratios and the mathematical models, often model the dynamics of the behavior between changes. Image 3.6 shows the relation of the tire force at different vehicle speed according their corresponding gear ratio in a combustion engine vehicle; there, can be seen the necessity to change to different ratios to sustain

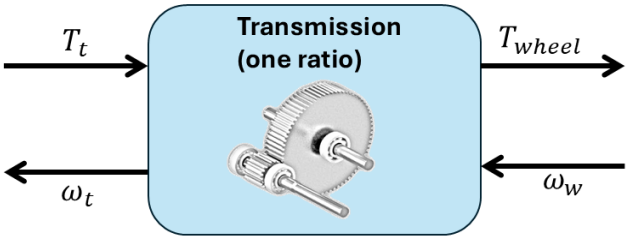


Figure 3.5: Schematic of the transmission model.

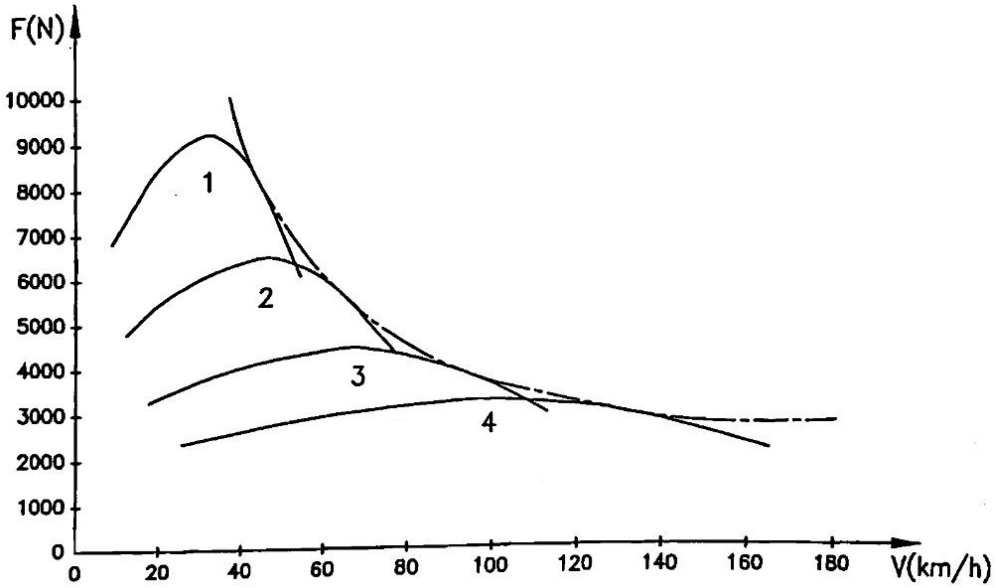


Figure 3.6: Wheel Tractive Force–Speed Curves for Different Gear Ratios (Four-Speed Gearbox). Taken from [4].

the maximum force at different vehicle speeds. EV's don't need to do these changes because electric motors can deliver the full torque in the almost the complete range, except when entering in field weakening mode.

3.1.1.6 The Combustion Engine

The engine dynamics is as follows as described in [72]:

$$I_e\dot{\omega}_e = T_i - T_f - T_a - T_p, \tag{3.26}$$

$$T_e = T_i - T_f - T_a, \tag{3.27}$$

where T_i is the engine combustion torque, T_f is the torque frictional losses, T_a is the accessory torque, and T_p is the pump torque that represents the load from the torque converter.

$$I_e\dot{\omega}_e = T_e - T_p, \tag{3.28}$$

Now, the following models are called the mean value model for spark-ignition (SI) engines, which model the mean values of a combustion engine's main components. Changing the symbols as follows:

$$T_{\text{ind}} = T_i, \quad T_{\text{load}} = T_p, \tag{3.29}$$

The indicated combustion torque is:

$$T_{\text{ind}} = \frac{H_u\eta_i\dot{m}_f}{\omega_e}, \tag{3.30}$$

where H_u is the fuel energy constant, η_i is the thermal efficiency multiplier for the cooling and exhaust system losses, and $\dot{m}_f$ is the fuel mass flow rate, determined by a fuel injection control system.

$$\dot{m}_f = \frac{\dot{m}_{ao}}{\lambda L_{\text{th}}}, \tag{3.31}$$

where $\dot{m}_{ao}$ is the air mass flow rate out of the intake manifold and into the cylinder, λ is the air/fuel equivalence ratio, and L_{th} is the stoichiometric air/fuel mass ratio for gasoline.

For the friction losses we have:

$$F_{\text{loss}} = a_0\omega_e^2 + a_1\omega_e + a_2, \tag{3.32}$$

The hydrodynamic friction is expressed as a polynomial expression (3.32) in terms of the engine speed ω_e (rad/s).

Besides the friction losses, there are pumping losses as follows:

$$P_{\text{loss}} = b_0\omega_e P_{\text{man}} + b_1 P_{\text{man}}, \tag{3.33}$$

Equation (3.33) represents the pumping losses, which are proportional to the pumping mean effective pressure $P_{\text{exh}} - P_{\text{man}}$, i.e., the difference between the exhaust pressure and the pressure in the manifold. However, since the exhaust pressure is nearly constant and equal to atmospheric pressure, the expression can be modeled as in (3.32).

Now, the total friction and pumping losses described in (3.26) would be the combination of (3.32) and (3.33):

$$T_f = a_0\omega_e^2 + a_1\omega_e + a_2 + b_0\omega_e P_{\text{man}} + b_1 P_{\text{man}}, \tag{3.34}$$

where a_0, a_1, a_2, b_0, b_1 are engine-dependent parameters.

The intake manifold pressure is as follows:

$$\dot{P}_{\text{man}} = \frac{RT_{\text{man}}}{V_{\text{man}}}\left(\dot{m}_{ai} - \dot{m}_{ao}\right), \tag{3.35}$$

where R is the ideal gas constant, T_{man} is the intake manifold temperature, and V_{man} is the manifold volume.

The outflow rate $\dot{m}_{ao}$ from the intake manifold is as follows:

$$\dot{m}_{ao} = \frac{\eta_{\text{vol}}\omega_e}{4\pi V_d}\frac{P_{\text{man}}}{RT_{\text{man}}}, \tag{3.36}$$

where η_{vol} is the volumetric efficiency, and V_d is the displacement volume of the engine cylinders. $P_{\text{man}}/RT_{\text{man}}$ represents the density of air in the intake manifold.

The volumetric efficiency can be modeled empirically as follows:

$$\eta_{\text{vol}} = (24.5\omega_e - 3.1\times 10^4)m_a^2 + (-0.167\omega_e + 222)m_a + (8.1\times 10^{-4}\omega_e + 0.352), \tag{3.37}$$

where m_a represents the air mass flow rate.

The inflow rate $\dot{m}_{ai}$ into the intake manifold is as follows:

$$\dot{m}_{ai} = \text{MAX}\cdot \text{TC}(\alpha)\cdot \text{PRI}, \tag{3.38}$$

where MAX is a constant dependent on the size of the throttle body, and represents the maximum possible intake airflow rate. $\text{TC}(\alpha)$ is the projected area that the flow sees through the throttle angle α, and α' is the minimum throttle angle.

$$\text{TC}(\alpha) = 1 - \cos(\alpha + \alpha'), \tag{3.39}$$

PRI is the pressure ratio influence function:

$$\text{PRI} = \begin{cases} \sqrt{1 - \left(\frac{p_r - p_c}{1 - p_c}\right)^2}, & p_r > p_c \quad \text{(choked)} \\ 1, & p_r \le p_c \quad \text{(sonic)} \end{cases}, \tag{3.40}$$

where $p_r = P_{\text{man}}/P_{\text{amb}}$ and p_c is the critical pressure ratio.

The complete equation for the manifold pressure is:

$$\dot{P}_{\text{man}} = \frac{RT_{\text{man}}}{V_{\text{man}}}\text{MAX} \cdot \text{TC}(\alpha) \cdot \text{PRI}\left(\frac{P_{\text{man}}}{P_{\text{amb}}}\right) - \frac{\omega_e}{4\pi}\frac{V_d}{V_{\text{man}}}\eta_{\text{vol}}P_{\text{man}}. \tag{3.41}$$

3.1.2 Lateral Dynamics

In this chapter, we explore the lateral vehicle dynamics models that are used in several safety applications such as lane-keeping systems, reviewed state of the art controller algorithms for lateral dynamic, design several types of controllers to assess their performance in automotive applications, and gain insight about the control design process and theory. Next, we list ADAS systems that benefit from the study of the lateral dynamics:

- Lane Departure Warning (LDW): Alerts the driver when the vehicle unintentionally drifts out of its lane, addressing lateral stability.
- Lane Keeping Assist (LKA): Automatically steers the vehicle back into its lane if it starts to drift, controlling lateral dynamics to maintain lane position.
- Blind Spot Detection (BSD): Monitors vehicles in adjacent lanes and provides alerts during lane changes, enhancing lateral awareness.
- Rear Cross-Traffic Alert (RCTA): Monitors lateral traffic when reversing out of parking spaces, warning the driver of approaching vehicles.
- Emergency Steering Assist (ESA): Assists the driver by applying corrective steering during emergency maneuvers to avoid obstacles, impacting lateral dynamics.
- Active Steering Assist: Actively adjusts the steering to maintain vehicle stability during high-speed maneuvers, ensuring control over lateral dynamics.
- Traffic Jam Pilot: Combines lane-keeping functions with longitudinal control, allowing the vehicle to maintain lateral stability in traffic.

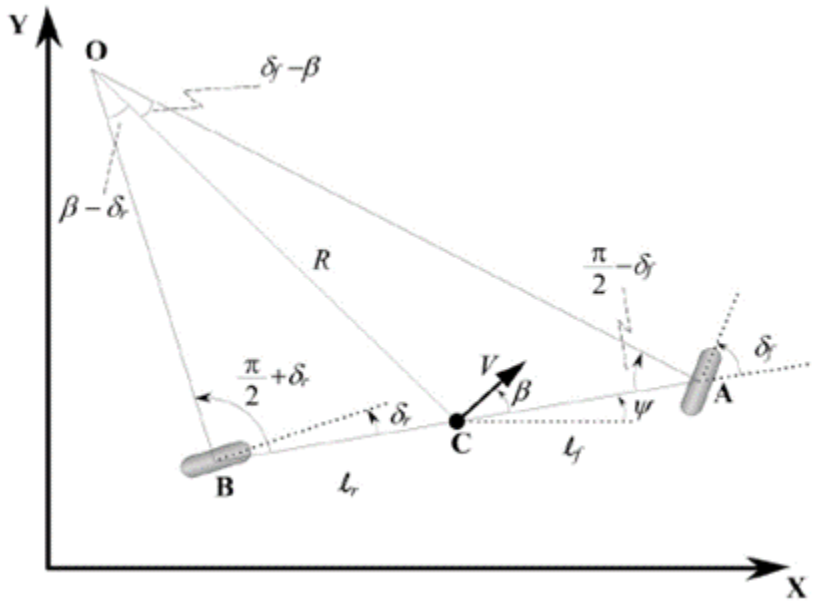


Figure 3.7: Kinematic model of the vehicle better known as "the bicycle model". This is a simplified model of the vehicle that only uses two wheels, rear and front. Taken from [72].

3.1.2.1 The Kinematic Bicycle Model

This is a brief explanation of the kinematic model [72], commonly used to analyze and construct algorithms for Advanced Driver Assistance Systems (ADAS) and to introduce the nomenclature and concepts in automotive applications.

The bicycle model Fig.3.7 is widely used as a simplification of the vehicle model, although other models such as the Ackermann model exist. The model consists of two wheels connected by a solid bar with a length L, and it assumes that both wheels can be steered. The vehicle is assumed to have planar motion, meaning the pitch and roll angles are neglected.

The components of the model are:

- A, B: The axes of wheel rotation and the points where the wheels can be steered.
- C: The center of gravity (CoG) of the vehicle.
- O: The center of the circle describing the trajectory of C (CoG) when the vehicle is in motion.
- R: The distance from O to C (the imaginary radius $\overline{OC}$).
- V: The velocity of C (CoF).

- β: The slip angle, which is the angle between V and the longitudinal axis of the vehicle (line $\overline{CA}$).
- ψ: The heading angle (yaw angle).
- $\gamma = \psi + \beta$: The angle between V and the X-axis of the inertial reference frame, called the course angle.
- δ_f, δ_r: Front and rear steering angles (points A and B, respectively).
- l_f, l_r: The distances from C to A ($\overline{CA}$) and from C to B ($\overline{CB}$), respectively.
- $L = l_f + l_r$: The total wheelbase $\overline{BA}$.
- X, Y: The axes of the inertial reference frame.

The kinematic model equations are summarized below. The detailed derivation of these equations is explained in [72]:

$$\dot{X} = V\cos(\psi + \beta), \tag{3.42}$$

$$\dot{Y} = V\sin(\psi + \beta), \tag{3.43}$$

$$\dot{\psi} = \frac{V\cos(\beta)}{l_f + l_r}\left(\tan(\delta_f) - \tan(\delta_r)\right), \tag{3.44}$$

$$\beta = \tan^{-1}\left(\frac{l_f\tan(\delta_r) + l_r\tan(\delta_f)}{l_f + l_r}\right). \tag{3.45}$$

Fig.3.8 shows the simulation of the above equations. We can see the simulation of the vehicle kinematics considering a constant velocity of 10m/s in this case and moving the steering wheel ($_f$) with a constant rate of 0.014 rads/s, incrementing during the first 5 seconds and decrementing 0.028 rads/s until the second 10 where we stop moving the steering wheel in the last position. We can appreciate in the trajectory of the CoG that the vehicle first moved to the left and later to the right and describing a circle as we expected if we keep the steering wheel in a position different that 0°. The Matlab code in Algorithm 17 shows the needed code to run this simulation.

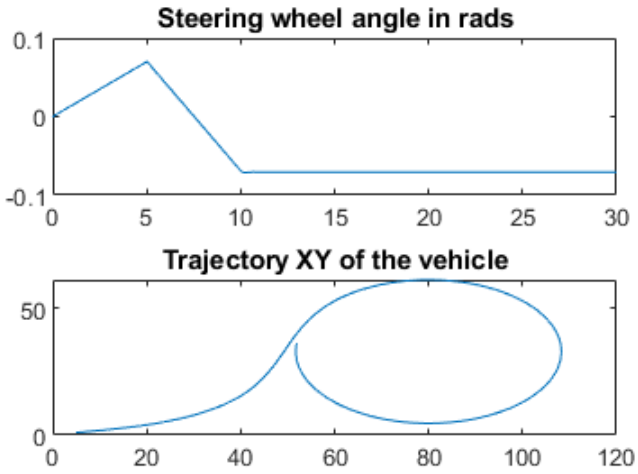


Figure 3.8: This is the behavior of the kinematic bicycle model. In the upper graph the angle of the steering wheel is shown in radians. We have two maneuvers: the first 5 seconds, we increment 0.8°/s (0.014rads/s). From the second 5 to 10, it decrements 1.6°/s. From second 10 to 30 there is no increment nor decrement. In the lower graph we see the trajectory of the CoG of the vehicle (distance is in meters).

3.1.2.2 Controlling the Heading Angle ψ (Error Feedback Controller)

As an exercise, a controller algorithm was developed using error feedback technique based on the bicycle model. The error, defined as the difference between the desired heading angle ψ_{ref} and the actual heading angle ψ, is used to control the front steering angle δ_f. We define ψ_{ref} in terms of the time t as

$$\psi_{ref} = \frac{\pi}{4} \sin\left(\frac{t}{20}\right), \tag{3.46}$$

this reference is arbitrary and the solely objective of this is to see if our controller is capable to make the plant to follow a sinusoidal movement. Since we only have one variable for control, in this case δ_f (angle of the front steering wheel) we can only control one output variable in this case we use ψ.

Defining our error, we have:

$$e = \psi_{ref} - \psi, \tag{3.47}$$

$$e = \frac{\pi}{4} \sin\left(\frac{t}{20}\right) - \psi. \tag{3.48}$$

Now, forcing the error dynamic as proportional to the same error

$$\dot{e} = ke, \tag{3.49}$$

where k is a constant gain and $\dot{e}$ is defined as

$$\dot{e} = \dot{\psi}_{ref} - \dot{\psi}, \tag{3.50}$$

$$\dot{e} = \frac{\pi}{80} \cos\left(\frac{t}{20}\right) - \dot{\psi}. \tag{3.51}$$

Combining (3.49) and (3.51) we have defined the dynamic of the error we want:

$$ke = \frac{\pi}{80} \cos\left(\frac{t}{20}\right) - \dot{\psi}. \tag{3.52}$$

Now, since $\dot{\psi}$ is defined in our bicycle model in (3.44). Solving for δ_f using a symbolic mathematical software such as Matlab, that in control theory is our controller U, we have:

$$\delta_f = U = -\arctan\left(\frac{3200\left(80Ke - \pi\cos\left(\frac{t}{20}\right)\right)\sqrt{\frac{(80V-80Ke+\pi\cos\left(\frac{t}{20}\right))(80V+80Ke-\pi\cos\left(\frac{t}{20}\right))}{2560000}}}{\pi^2\cos^2\left(\frac{t}{20}\right) - 6400V^2 + 6400K^2e^2 - 160K\pi e\cos\left(\frac{t}{20}\right)}\right). \tag{3.53}$$

This control law (3.53) can be found with the Algorithm 18 in Matlab.

The system in the state space we want to simulate, assuming $\delta_r = 0$, $\delta_f = U$ and combining with the equations (3.42-3.45) takes the form

$$\begin{aligned} \dot{X}_1 &= V\cos(\psi + \beta), \\ \dot{X}_2 &= V\sin(\psi + \beta), \\ \dot{X}_3 &= \frac{V\cos(\beta)}{l_f + l_r}\tan(U). \end{aligned} \tag{3.54}$$

It is important to note that in this system we considered the velocity V as a constant. δ_r, the rear steering wheel is considered 0; this means that we can only control the front steering wheel as in conventional bicycles.

A simulation is shown in Fig.3.9 and Fig. 3.10 with the following parameters: $l_f = 1, l_r =$

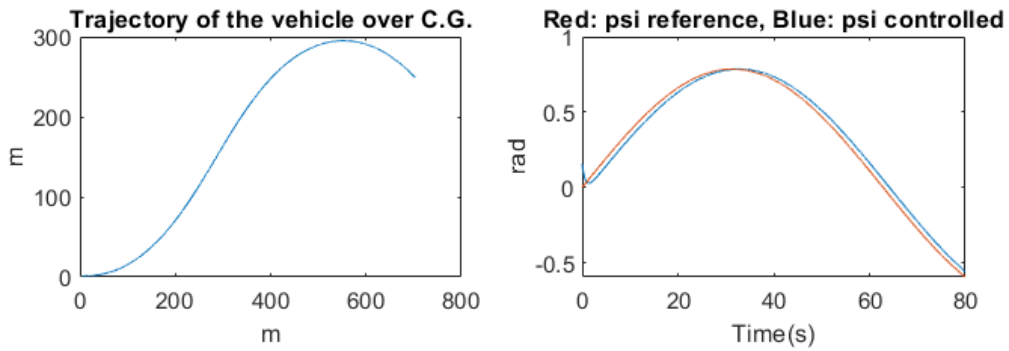


Figure 3.9: This is the simulation of the controller for ψ. The upper graph shows the trajectory of the vehicle. The lower graph shows the comparison between the reference and the controlled signal. We can appreciate that we have always an error in stable state.

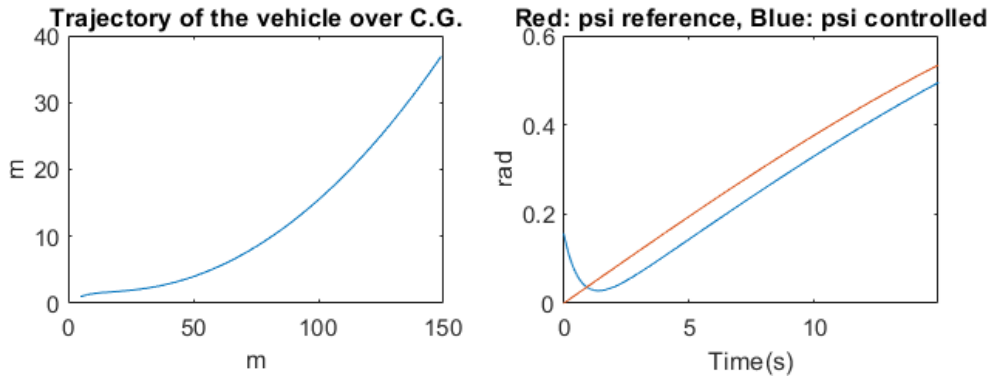


Figure 3.10: This is the simulation of the controller for ψ for the first 15 seconds. We can appreciate the error in stable state and the transient behavior at the beginning. The red line is the reference, and the blue one is our controlled signal.

$1, V = 10, k = 1.5$, initial states $X = [51/20]$. In Fig.3.9 we can see the simulation of the vehicle dynamic considering a constant velocity of 10m/s in this case. We can appreciate the sinusoidal movement as we defined as a reference in (3.46). We can see the constant error in stable state. In Fig.3.10 there is a zoom in of the same graph where we can see the transient state in the blue line.

3.1.2.3 The Dynamic Model of Lateral Vehicle Dynamics

This dynamic model, unlike the kinematic model, does not assume that the velocity of each wheel follows the direction of the wheels. It presents a dynamic model that accounts for the forces involved in the system. The model has two degrees of freedom: the yaw angle ψ and the lateral position y, measured with respect to the center of rotation of the vehicle O, as described in Section

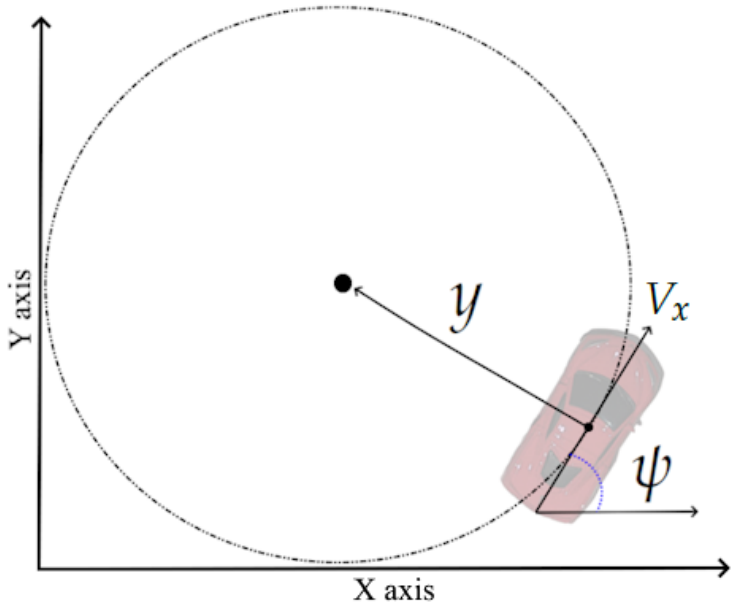


Figure 3.11: Components of the lateral dynamics.

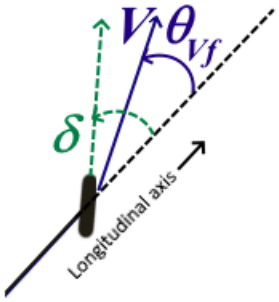


Figure 3.12: Diagram showing the slip angle component. δ is the steering angle, V the velocity vector of the vehicle with θ_{vf} as its angle. The lateral slip angle α is defined as $\delta - \theta_{vf}$.

3.1.2.1. The longitudinal velocity is denoted as V_x, and the yaw angle ψ is measured relative to the global X-axis, as shown in Fig. 3.11.

Using Newton's second law, the motion along the Y-axis is defined as:

$$ma_y = F_{yf} + F_{yr}, \tag{3.55}$$

where F_{yf} and F_{yr} are the front and rear lateral tire forces, respectively. For small slip angles α (the angle between the velocity vector and the tire angle ($\alpha = \delta - \theta_{vf}$, Fig. 3.12), the lateral forces are considered to be proportional to the slip angle.

Here is presented the state space model according [72] where are more details about the deduction of it.

$$\frac{d}{dt}\begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{\alpha f}+2C_{\alpha r}}{mV_x} & 0 & -V_x - \frac{2C_{\alpha f}l_f - 2C_{\alpha r}l_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2C_{\alpha f}l_f - 2C_{\alpha r}l_r}{I_z V_x} & 0 & -\frac{2C_{\alpha f}l_f^2 - 2C_{\alpha r}l_r^2}{I_z V_x} \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2C_{\alpha f}l_f}{I_z} \end{bmatrix} \delta_f, \tag{3.56}$$

where the new constants are defined as:

- $C_{\alpha r}, C_{\alpha f}$: Cornering stiffness of the rear and front wheels, which measure the tire's ability to resist deformation while cornering.
- I_z: Moment of inertia of the vehicle about the Z-axis.
- V_x: Longitudinal velocity.
- l_f, l_r: Distances from the center of gravity (CoG) to the front and rear axles.

Using a differential equation solver such as Matlab-Simulink, from (3.56) we can get $V_y = \dot{y}$, $\dot{\psi}$, and ψ. Now, having the lateral velocity V_y, the orientation ψ, and the longitudinal velocity V_x which for simplification we consider it as constant, we rotate the lateral and longitudinal velocities ψ rads and integrate them to get the coordinates as follows:

$$\begin{bmatrix} X - X_0 \\ Y - Y_0 \end{bmatrix} = \int_0^t \left(\begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} V_x \\ V_y \end{bmatrix} \right) dt, \tag{3.57}$$

where X_0 and Y_0 are the initial coordinates at $t = 0$. These global coordinates will be used for simulation, and represent our GPS current position.

3.1.3 Dynamic Model in Terms of Error Respect to Road

A method used in automatic lane-keeping systems, presented in [72], is explored here. This model represents the system in terms of error with respect to a reference (e.g., the road centerline), and a controller is designed accordingly.

First, the errors are defined as:

- e_1: The lateral displacement of the CoG from the road centerline.

- e_2: The orientation error of the vehicle with respect to the road.

The lateral error is given by:

$$\ddot{e}_1 = (\ddot{y} + V_x\dot{\psi}) - \frac{V_x^2}{R} = \ddot{y} + V_x(\dot{\psi} - \dot{\psi}_{\text{des}}), \tag{3.58}$$

where y is the lateral acceleration, ψ is the yaw rate, V_x is the longitudinal velocity, and R is the radius of the turning circle.

The orientation error is:

$$e_2 = \psi - \psi_{\text{des}}. \tag{3.59}$$

After integrating (3.58), the lateral error becomes:

$$\dot{e}_1 = \dot{y} + V_x(\psi - \psi_{\text{des}}). \tag{3.60}$$

It is important to notice that V_x is being considered as a constant. If it is required to define it as variable, the result of integrating (3.58) and using the derivative of (3.59) results:

$$\dot{e}_1 = \dot{y} + \int V_x(t)\dot{e}_2\, dt. \tag{3.61}$$

This new equation results in a model that is non linear and time varying making it difficult for control system design. For that reason, we take the assumption that the velocity is constant.

The dynamic model in terms of the error is as follows:

$$\mathbf{X} = \begin{bmatrix} e_1 \\ \dot{e_1} \\ e_2 \\ \dot{e_2} \end{bmatrix}, \tag{3.62}$$

$$\dot{\mathbf{X}} = \mathbf{AX} + \mathbf{B_1}\delta + \mathbf{B_2}\psi_{\text{des}} + \mathbf{B_3}\sin(\phi),$$

where $\dot{\psi}_{des} = \frac{V_x}{R}$ and as explained before R is the radius from the point O to the CoG of the circle that describes the vehicle when it is turning. e_1 is the lateral position error respect the road and is related to (3.58). e_2 is the yaw angle error with respect the road (3.59). δ is the front wheel steering

angle. ϕ is the bank angle. The matrices $\mathbf{A}, \mathbf{B_1}, \mathbf{B_2}, \mathbf{B_3}$ are defined as follows:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{\alpha f}+2C_{\alpha r}}{mV_x} & \frac{2C_{\alpha f}+2C_{\alpha r}}{m} & \frac{-2C_{\alpha f}l_f+2C_{\alpha r}l_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2C_{\alpha f}l_f-2C_{\alpha r}l_r}{I_zV_x} & \frac{2C_{\alpha f}l_f-2C_{\alpha r}l_r}{I_z} & \frac{-(2C_{\alpha f}l_f^2+2C_{\alpha r}l_r^2)}{I_zV_x} \end{bmatrix},$$

$$\mathbf{B_1} = \begin{bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2C_{\alpha f}l_f}{I_z} \end{bmatrix}, \quad \mathbf{B_2} = \begin{bmatrix} 0 \\ -\frac{2C_{\alpha f}l_f-2C_{\alpha r}l_r}{mV_x} - V_x \\ 0 \\ -\frac{2C_{\alpha f}l_f^2+2C_{\alpha r}l_r^2}{I_zV_x} \end{bmatrix}, \quad \mathbf{B_3} = \begin{bmatrix} 0 \\ g \\ 0 \\ 0 \end{bmatrix},$$

We run the simulation using the next parameters: $C_{\alpha r} = 80,000, C_{\alpha f} = 80,000, I_z = 2873, V_x = 8m/s, l_f = 1.1m, l_r = 1.58m, m = 1573kg, g = 9.81m/s^2$, and δ with 0° and 4.3° , and $R = 1000$. The implementation in Matlab code is as follows in Algorithm 19.

3.1.3.1 Ackermann Controller Design for Lateral Dynamics

The first controller we design is the Ackerman controller described in [58]. Recalling the Ackerman's formula for the system $\dot{\mathbf{X}} = \mathbf{AX} + \mathbf{B}u$ where u would be our δ:

$$u = -\mathbf{KX}, \tag{3.63}$$

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix} \left[\mathbf{B} \,|\, \mathbf{AB} \,|\, \cdots \,|\, \mathbf{A}^{n-1}\mathbf{B}\right]^{-1} \mathbf{H}(\mathbf{A}). \tag{3.64}$$

And we call the controllability matrix $\mathbf{M_c}$ to the matrix formed as $[\mathbf{B} \,|\, \mathbf{AB} \,|\, \cdots \,|\, \mathbf{A}^{n-1}\mathbf{B}]$. Having the poles $\{p_{d1}, p_{d2}, p_{d3}, \ldots, p_{dn}\}$ and $\mathbf{I}$ is the identity matrix:

$$\mathbf{H}(\mathbf{A}) = (\mathbf{A} - p_{d1}\mathbf{I})(\mathbf{A} - p_{d2}\mathbf{I}) \cdots (\mathbf{A} - p_{dn}\mathbf{I}). \tag{3.65}$$

In our case, we make $\mathbf{B} = \mathbf{B_1}$ as in (3.62), and the poles for the simulation are $p_d = \{-5 - 3i, -5 + 3i, -7, -10\}$. Using (3.65) and (3.66),

$$\mathbf{M_c} = \left[\mathbf{B_1} \,\mathbf{AB_1}\, \mathbf{A}^2\mathbf{B_1}\, \mathbf{A}^3\mathbf{B_1}\right], \tag{3.66}$$

$$\mathbf{H}(\mathbf{A}) = (\mathbf{A} - p_{d1}\mathbf{I})(\mathbf{A} - p_{d2}\mathbf{I})(\mathbf{A} - p_{d3}\mathbf{I})(\mathbf{A} - p_{d4}\mathbf{I}). \tag{3.67}$$

Since the rank of the matrix $\mathbf{M_c}$ (3.66) is the same as the number of poles, this matrix is invertible; in other words, our system is controllable. Hence, the control gain $\mathbf{K}$ is obtained in the form:

$$\mathbf{K} = [0\,0\,0\,1]\,\mathbf{M_c}^{-1}\mathbf{H}(\mathbf{A}). \tag{3.68}$$

We run the simulation using the following parameters: $C_{\alpha r} = 80,000, \quad C_{\alpha f} = 80,000, \quad I_z = 2873, \quad V_x = 8\,\text{m/s}, \quad l_f = 1.1\,\text{m}, \quad l_r = 1.58\,\text{m}, \quad m = 1573\,\text{kg}, \quad R = 1000, \quad g = 9.81\,\text{m/s}^2$.

The maneuver is as follows: during the first second, the street is straight, and then the street follows the curve with radius $R = 1000$, so that:

$$\dot{\psi}_{\text{des}} = \begin{cases} 0, & t < 1 \\ \frac{V_x}{R}, & t \geq 1 \end{cases} \tag{3.69}$$

To convert the error state space history of the system to a trajectory based on global coordinates of the inertial space we use (3.70-3.71):

$$dX = V\cos(\psi_{\text{des}}) - \left[\dot{e}_1 \sin(e_2 + \psi_{\text{des}}) + e_1 \cos(e_2 + \psi_{\text{des}})(\dot{e}_2 + \dot{\psi}_{\text{des}})\right], \tag{3.70}$$

$$dY = V\sin(\psi_{\text{des}}) + \left[\dot{e}_1 \cos(e_2 + \psi_{\text{des}}) - e_1 \sin(e_2 + \psi_{\text{des}})(\dot{e}_2 + \dot{\psi}_{\text{des}})\right]. \tag{3.71}$$

These equations are solved within the same solver in the simulation together with the model. These are implemented in the Algorithm 21, part 3, to draw the trajectory of the vehicle using the error states.

We can see the results of the simulation in Fig.3.13. The error e_1 converges with critically damped behavior, while the error e_2 shows an under-damped behavior. Both error variables present a steady-state error.

Fig. 3.14 shows how the trajectory of the vehicle would be, where the dashed red line is the behavior of the system, and the black line is the reference.

In Algorithm 21 part 1-3 is shown the implementation and a explanation is given.

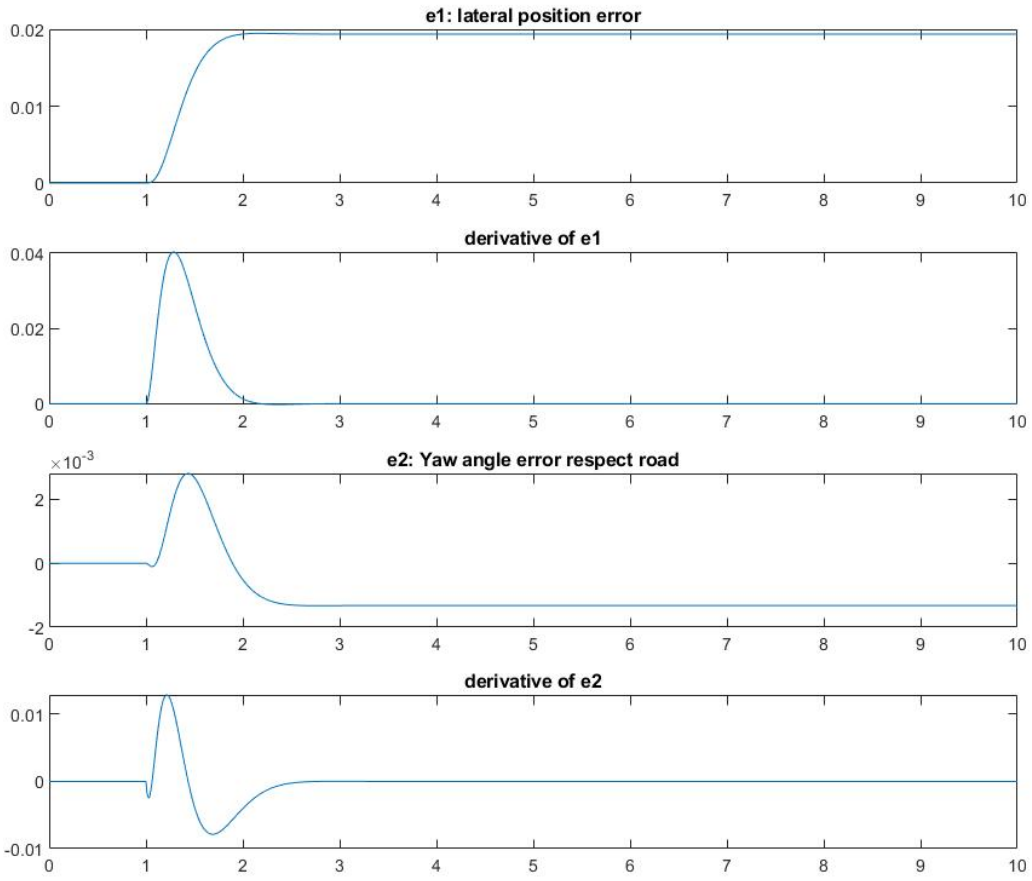


Figure 3.13: Error behavior using the Ackerman controller design for the model of error respect the road.

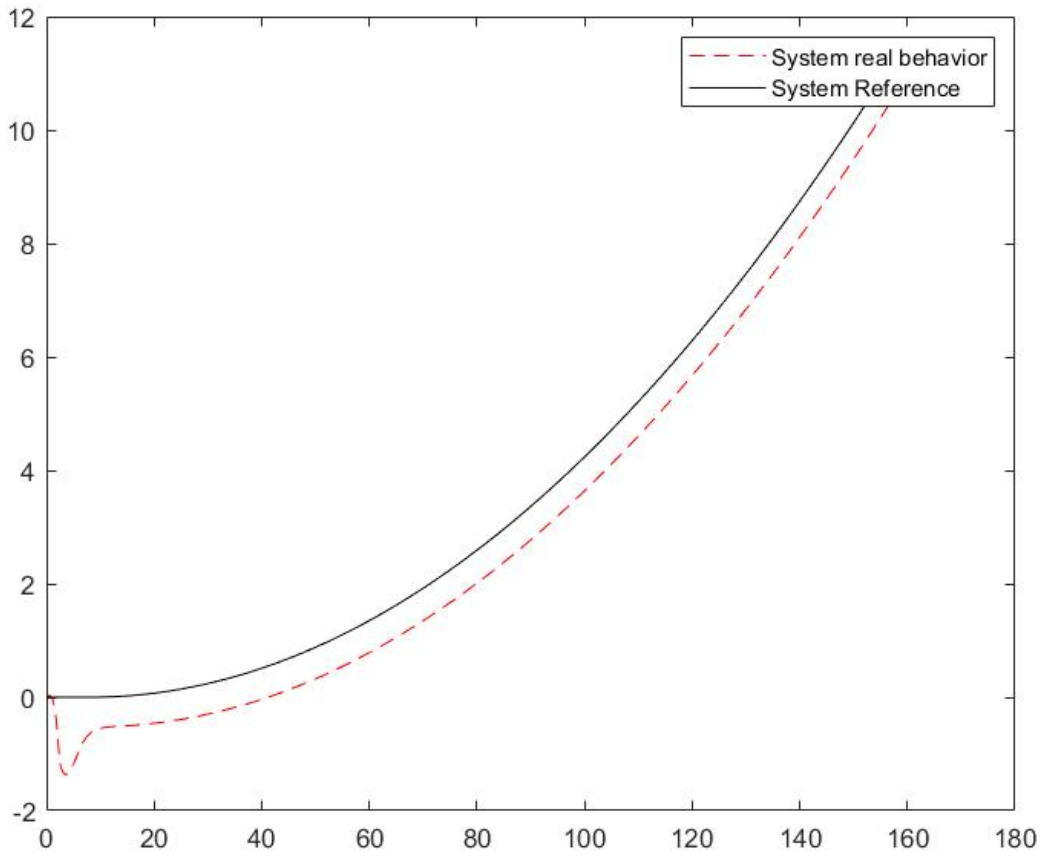


Figure 3.14: Trajectory that the vehicle would follow using the Ackerman controller design.

3.1.3.2 Controller Design by Blocks of the Lateral Dynamics

This section presents the controller design using the "control by blocks" method. The controller is using only the error dynamic for the y distance that corresponds to $e_1, \dot{e_1}$, and so, it is incomplete because it is not using the ψ dynamic for the $e_3, \dot{e_3}$. In further chapters we will retake this method using all dynamics. Starting with (3.62), we define the first block as:

$$E_1 = X_1 = e_1, \quad E_2 = X_2 = \dot{e}_1, \tag{3.72}$$

then, the derivative is

$$\dot{E}_1 = X_2 = -k_1 E_1, \tag{3.73}$$

and since we want to control E_1 using E_2, then we define our reference $E_{2,\text{ref}}$

$$E_{2,\text{ref}} = -k_1 E_1, \tag{3.74}$$

and its derivative

$$\dot{E}_{2,\text{ref}} = -k_1 E_2. \tag{3.75}$$

Now, we define the second block as:

$$E = E_{2,\text{ref}} - E_2, \tag{3.76}$$

deriving and substituting by the second row of (3.62), corresponding to $\ddot{e}_1$ and making it equal to $-k_2 E$ we have:

$$\begin{aligned} \dot{E} = \dot{E}_{2,\text{ref}} &- \left[0, -\frac{2C_{\alpha f} + 2C_{\alpha r}}{mV_x}, \frac{2C_{\alpha f} + 2C_{\alpha r}}{m}, \frac{-2C_{\alpha f} l_f + 2C_{\alpha r} l_r}{mV_x}\right] X \\ &- \left[\frac{2C_{\alpha f}}{m}\right] \delta - \left[\frac{-2C_{\alpha f} l_f + 2C_{\alpha r} l_r}{mV_x} - V_x\right] \dot{\psi}_{\text{des}} - g \sin(\phi) \\ &= -k_2 E. \end{aligned} \tag{3.77}$$

Solving (3.77) for δ, that will be our control law U,

$$\begin{aligned} U = \delta = &\left(\dot{E}_{2,\text{ref}} \quad + k_2 E - \left[\frac{-2C_{\alpha f} l_f + 2C_{\alpha r} l_r}{mV_x} - V_x\right] \dot{\psi}_{\text{des}} - g \sin(\phi)\right) \frac{m}{2C_{\alpha f}} \\ &- \left(-\frac{2C_{\alpha f} + 2C_{\alpha r}}{V_x} X_2 + (2C_{\alpha f} + 2C_{\alpha r}) X_3 + \frac{-2C_{\alpha f} l_f + 2C_{\alpha r} l_r}{V_x} X_4\right) \frac{1}{2C_{\alpha f}}. \end{aligned} \tag{3.78}$$

Then, we only have to substitute this U into the Algorithm 21 to test this control.

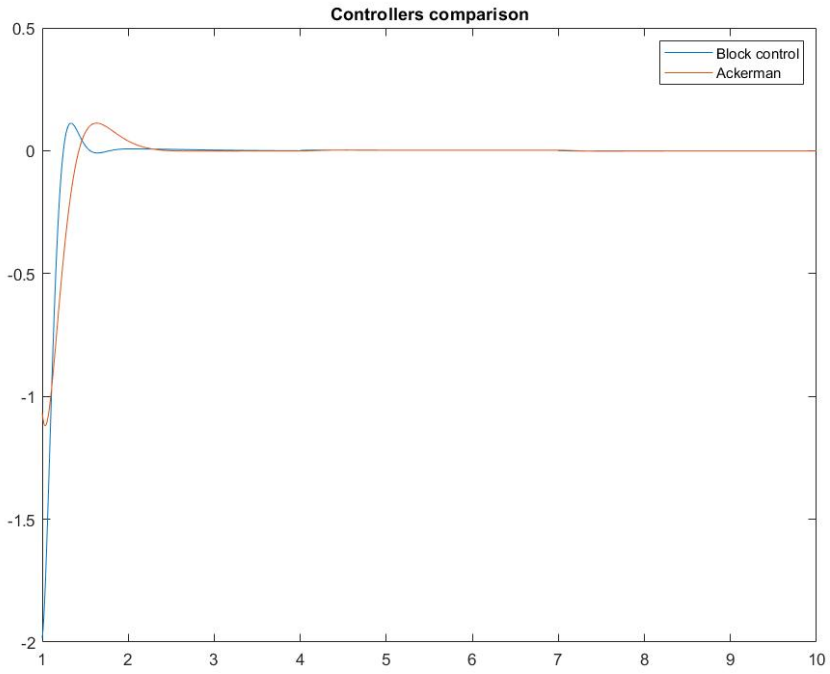


Figure 3.15: Comparison of the control laws for Ackerman (in red) and by blocks (in blue).

Implementing in simulation with initial conditions $X = [e_1, \dot{e}_1, e_2, \dot{e}_2]^T = [X_1, X_2, X_3, X_4] = [1, 1, 1, 1]$, we can see the results in Figs. 3.15-3.16. The simulation parameters are as follows: $C_{\alpha r} = 80,000$, $C_{\alpha f} = 80,000$, $I_z = 2873$, $V_x = 15\,\text{m/s}$, $l_f = 1.1\,\text{m}$, $l_r = 1.58\,\text{m}$, $m = 1573\,\text{kg}$, $R = 1000$, $\phi = \frac{\pi}{50}$, $g = 9.81\,\text{m/s}^2$. To convert to global coordinates based on the errors $e_1, \dot{e}_1, e_2, \dot{e}_2$ the equations (3.70) and (3.71) can be used. We can compare the results of both controller looking at Fig. 3.15 and Fig. 3.16. The results are similar, only with the transient state a bit different due to the differences in the gains of the block controller and the poles defined in the Ackermann controller; however, we can see that both remain with almost the same error in stable state Fig. 3.16. We see in Fig. 3.15 the control laws of both controllers.

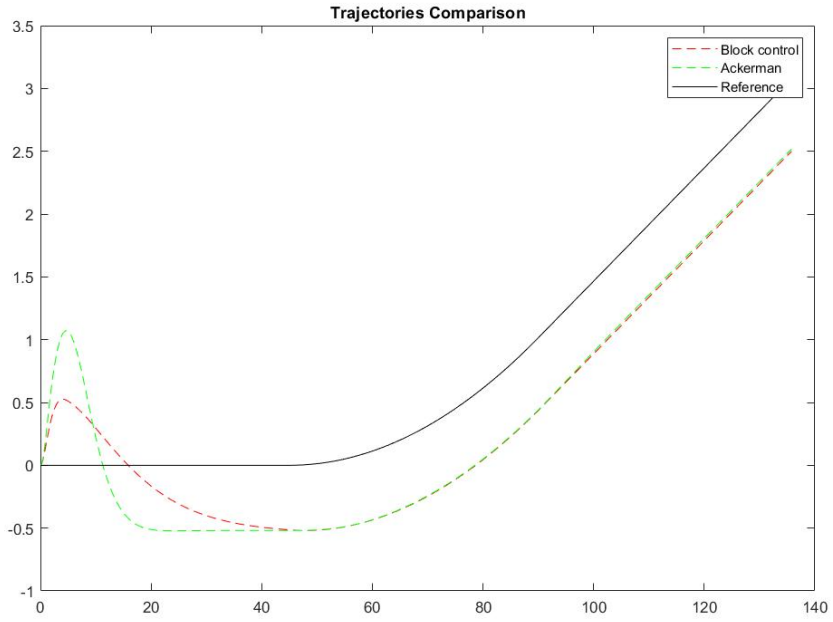


Figure 3.16: Comparison of the resultant trajectories between Ackerman (dashed green) and by block controller (dashed red). The black line is the reference.

3.2. Construction of an Electric Vehicle Model for Simulation

In this section, we present a vehicle model developed in Matlab®- Simulink® combining the physical dynamics of a vehicle seen in the last chapters, such as the tire dynamics, lateral and longitudinal dynamics and the dynamics of electrical machines such as BLDC and DC motors applied to the power of the vehicle and for the steering as well. Different controllers are applied for the vehicle speed and for the steering wheel simulating a keep lane system that follows a series of way-points specified in a text file that represents the coordinates in the terrain. The equations that describe vehicle physics in previous sections are written here again to facilitate the reading about how they are implemented in software blocks for simulation. In the references, the link to download the model can be found.

3.2.1 Introduction

This work is a set of several physical models put together and interacting each other to model the dynamics of a vehicle. There are plenty of similar works like the present. However, most of them are focus only on a specific part of the vehicle dynamics such as only the longitudinal or lateral, others in the suspension systems, the tires, the power, etc. and besides, they do not provide the source files to reuse them. As example we can mention some works such as [34] that implements parts of the longitudinal dynamics to be compared with a neural network. [21] uses the lateral model to create a new stability controller for electric vehicles. [1] models the lateral dynamics and implements fuzzy PID and linear quadratic control. [47] implements sliding mode control for lateral dynamics; this is expected to serve as reference for a future work with the additional feature of the electric motor dynamics for lateral and longitudinal dynamics as next steps for the present work. [15] is more related to the longitudinal dynamics plus the vertical dynamics that models the suspension system in an electrical vehicle; however, they do not model the dynamics of an electrical motor, rather they calculate the relation between the applied voltage and the vehicle speed. Some similar works as the present, such as the model examples of Simulink® can be obtained running the command ”sdl_car” having the Simscape® Driveline® toolboxes. These are very complete models well tested. However, the inconvenient of these models is that many blocks cannot be seen internally to see the actual implementation, and as opposite, in the present work all blocks can be seen internally and implement the theoretical models based in classic literature for academic purposes.

The focus of the present work is aimed to the electrical vehicles (EVs). The vehicle power considered for the longitudinal dynamic is an electrical motor that can be a DC motor or BLDC. For the lateral dynamics, an electric motor is considered as actuator for the steering wheel. The route path is taken from a file of way-points indicating the trajectory that must be followed by the vehicle. The actual coordinates followed by the vehicle are converted to global coordinates and compared with the desired trajectory in the way-points; this is useful to observe the performance of the lateral controller and longitudinal controller, that, in this model can be PID or error feedback with the Ackermann technique in the case of lateral controller and only PID for longitudinal controller. The utility of this model is to help in the analysis and design of controllers oriented to several aspects of

the EVs and autonomous (self-driving) vehicles. That is the reason of being free and available the model and source code in public space; see references.

3.2.2 Coordinates Conversion and Trajectory Errors.

In autonomous driving architectures, upper layers of the system pass a series of points, that are coordinates on the ground, to the control system that the vehicle must follow. That layers could be route planning, behavioral layer, motion planning, and local feedback control (Paden, et al., 2016). Commonly, this sequence of waypoints is obtained from a file, where each row is a set of values for coordinates and desired speed, that is $[X, Y, V_x]$. In this work, we did not use the V_x coming from this file; instead, we define a V_x profile along the trajectory to make it easier to control from the simulation. However, there is possible that this V_x from the file can be connected to the simulation anytime.

Based on the waypoint series, it is possible to calculate the error with respect to these points and the current coordinates vehicle calculated based on the vehicle dynamics. Consider an array with the coordinates of the points following the center of the lane in a road in the way as:

$$P_n = \begin{bmatrix} X_n & Y_n \end{bmatrix}^T, \tag{3.79}$$

where, X_n is the reference coordinate X_{ref}, Y_n is the reference coordinateY_ref, and n is the waypoint element number which maximum is the size of the array that shape the trajectory (P). In Fig. 3.17, it is possible to see the trajectory defined by the waypoints in the file used

Let us define P_q as the current position of the center of gravity (CoG) of the vehicle in the inertial reference system, and $P_k \in P$ is the closest point to P_q (CoG) in the list P. A k-nearest neighbor (kNN) search is performed for the given query point $P_q = \begin{bmatrix} X & Y \end{bmatrix}^T$, which is the current position of the vehicle, then

$$k = \text{kNN}(P_q) \in P, \tag{3.80}$$

where k is the closest point in the way-points list P to the actual position of the vehicle P_q.

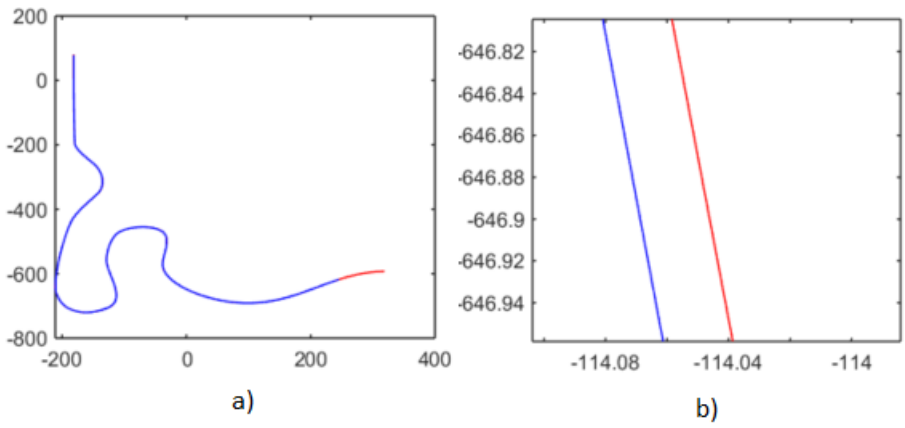


Figure 3.17: The trajectory of the vehicle (blue) over the reference path (red). Path in meters. a) vehicle trajectory. b) Close-up of the trajectory.

Then, the desired orientation is obtained as:

$$\psi_{\text{des}} = \text{atan2}(Y_{k+n} - Y_k, X_{k+n} - X_k), \tag{3.81}$$

which is computed by means of the closest point to CoG on the road and $n > 0$ points ahead. The parameter n can be tuned according to the noise in the points coming from the way-points of the trajectory-planning layer. In the present simulation, we used $n = 5$.

The coordinates in the inertial system must be converted to body-fixed system coordinates with P_k as the center and rotated by the angle defined by ψ_{des}. This transformation is defined as:

$$P_{\text{veh}} = M_{\text{rot}}(P_q - P_k), \tag{3.82}$$

with

$$M_{\text{rot}} = \begin{bmatrix} \cos(\psi_{\text{des}}) & \sin(\psi_{\text{des}}) \\ -\sin(\psi_{\text{des}}) & \cos(\psi_{\text{des}}) \end{bmatrix}.$$

In the Fig. 3.18, P_q is the current position of the CoG of the vehicle in the inertial system. P_k is the closest point of the way-points list to the CoG of the vehicle. P_{k-1} and P_{k+1} are the points before and after of the waypoint P_k. Here, P_k represents the origin of our new coordinates system which corresponds to the reference point in the road, and it is the closest point to the C.G of the vehicle. $\mathbf{P}_{veh}$ is a vector formed between the points P_q and P_k expressed in the vehicle coordinate system.



Figure 3.18: Desired trajectory and the error respecting the vehicle position.

Let define new error variables e_1 and e_2 as follows:

$$e_1 = \mathbf{P}_{veh,2}, \tag{3.83}$$

where e_1 the second element of $\mathbf{P}_{veh}$ which is the y distance from P_k to P_q and it means the lateral error, and

$$e_2 = \psi - \psi_{des}, \tag{3.84}$$

$$\dot{e}_1 = V_y + V_x e_2. \tag{3.85}$$

Now, a circle is fitted into the points P_{k-1}, P_k, and $P_k + 1$ using the function circfit() which take the points as input and returns x_c, y_c and R_c which are the coordinates of the center of the circle and its radius. After that, the inertial coordinates x_c, y_c are transformed to body fixed coordinates having P_k as reference point and rotating to the orientation of the road.

$$\begin{bmatrix} x_{cBF} & y_{cBF} \end{bmatrix} = \left(\begin{bmatrix} x_c & y_c \end{bmatrix} - P_k \right) \cdot M_{\text{rot}}. \tag{3.86}$$

Since R_c from circfit()[1] is always positive, we take the sign of y_{cBF} for R, and having R we obtain the desired angular speed of the orientation and the derivative in the orientation error e_2

$$R = R_c \cdot \text{sign}(y_{cBF}), \tag{3.87}$$

$$\dot{\psi}_{des} = \frac{V_x}{R}, \tag{3.88}$$

$$\dot{e}_2 = \dot{\psi} - \dot{\psi}_{des}. \tag{3.89}$$

[1] circfit - File Exchange - MATLAB Central (mathworks.com)

At this point, vector of errors can be constructed as $[e_1 e_1 e_2 e_2]$, and consider that the current position of the vehicle $P_q = [XY]$ is known. Now, the global coordinates in the inertial system (i.e. the coordinates from a GPS system) using the lateral and longitudinal velocities, can be calculated by taking the integral of them; however, since the lateral and longitudinal velocities(V_y, V_x), are from the vehicle orientation perspective, it is possible to integrate these velocities reorienting the vector according the vehicle orientation (ψ, the yaw angle).

Having V_x as longitudinal velocity, $V_y = yi$ as the lateral velocity, and ψ as the vehicle orientation, the velocity from the inertial system is

$$\mathbf{V}_{CG} = \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} V_x \\ V_y \end{bmatrix}, \tag{3.90}$$

and hence

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \int_0^t \mathbf{V}_{CG}\, dt. \tag{3.91}$$

Another way to express it, considering initial conditions different from 0, can be:

$$\begin{bmatrix} X - X_0 \\ Y - Y_0 \end{bmatrix} = \int_0^t \left(\begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} V_x \\ V_y \end{bmatrix} \right) dt, \tag{3.92}$$

where X_0 and Y_0 are the initial coordinates at $t = 0$. These global coordinates represent our GPS current position.

Next, in Fig. 3.19, is shown the Simulink® block that contains the code to get the errors. All equations, from eq. 3.79 to eq.3.89 are inside this block. It reads a file with the waypoints of the reference trajectory. Each time step the function knnsearch is called; this is intentional because we consider that this trajectory could change in real time since this is expected to be delivered by higher layer of the autonomous vehicle, lets say the trajectory planning layer. Algorithm 24 shows the code inside this block. Below the code is a detailed explanation of this.

Function Description:

The `GetReff` function calculates various error metrics and reference values needed for the lateral control of a vehicle as it follows a predefined path. Using the vehicle's current position, velocity,

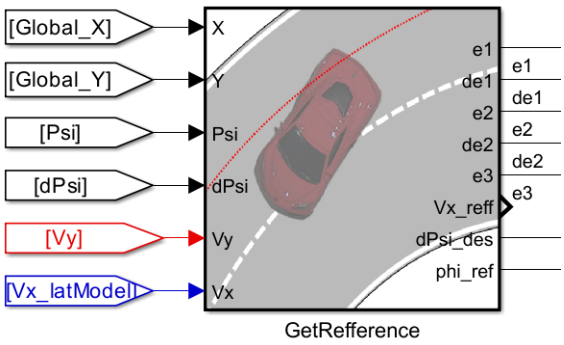


Figure 3.19: GetRef Simulink® block.

and orientation, the function computes lateral and longitudinal errors, desired yaw rates, and road curvature, which are essential for accurate steering control in autonomous driving applications.

Inputs:

- `X`: Global X-coordinate of the vehicle's center of gravity (C.G.).
- `Y`: Global Y-coordinate of the vehicle's C.G.
- `racetrackwaypoints`: Array with waypoints of the road, structured as $[X, Y, Vx]$, where Vx represents the reference velocity.
- `Psi`: Current yaw angle of the vehicle relative to the global coordinate system.
- `dPsi`: Current yaw rate or angular velocity of the vehicle.
- `Vy`: Lateral velocity of the vehicle.
- `Vx`: Longitudinal velocity of the vehicle.

Outputs:

- `e1`: Lateral position error, representing the perpendicular distance between the vehicle's C.G. and the closest point on the reference path.
- `de1`: Derivative of `e1`, combining lateral velocity and yaw angle error, which helps indicate lateral position change.
- `e2`: Yaw angle error, showing the difference between the vehicle's current yaw angle and the road's orientation at the closest point.
- `de2`: Derivative of `e2`, representing the difference between the current and desired yaw rates.
- `e3`: Longitudinal position error, showing the distance along the path between the vehicle and

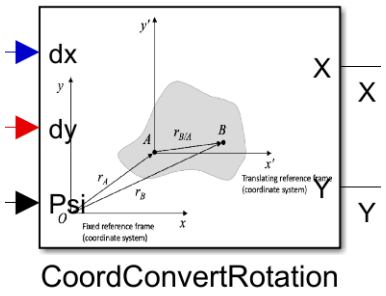


Figure 3.20: Coordinates conversion block.

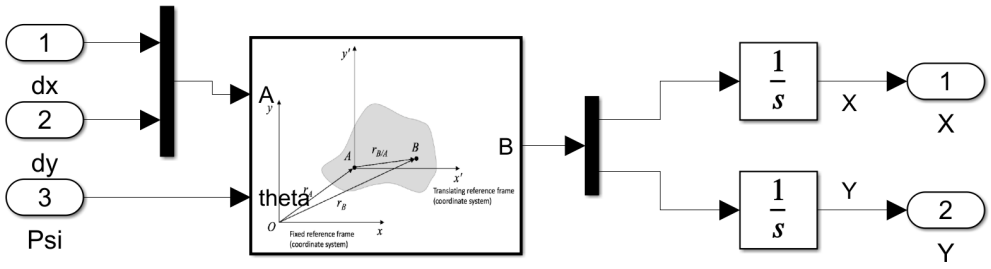


Figure 3.21: Coordinates conversion block inside.

the desired waypoint.

- `Vx_reff`: Reference longitudinal velocity at the closest waypoint on the path.
- `dPsi_des`: Desired yaw rate, calculated based on road curvature and vehicle speed.
- `phi_ref`: Road superelevation angle or banking, which accounts for the road's inclination due to curvature and speed.

The m-code of this block is described in Algorithm 24 in appendix, line by line.

Fig. 3.20 shows the implementation in Simulink® of the conversion of the coordinates from body-fixed to inertial reference frame. The equations implemented here are from eq.3.90 to eq.3.91. The block received the lateral and longitudinal velocities from the body-fix perspective and the angle of the orientation of the vehicle. It delivers the coordinates on the global inertial reference frame.

Fig 3.21 shows the internal blocks. here we can see the integral of the velocities once those are rotated. The integral, let us remember, are the blocks $\frac{1}{s}$.

Algorithm 25 shows the code to rotate the vector speed according the vehicle orientation. The equation implemented here is eq. 3.90.

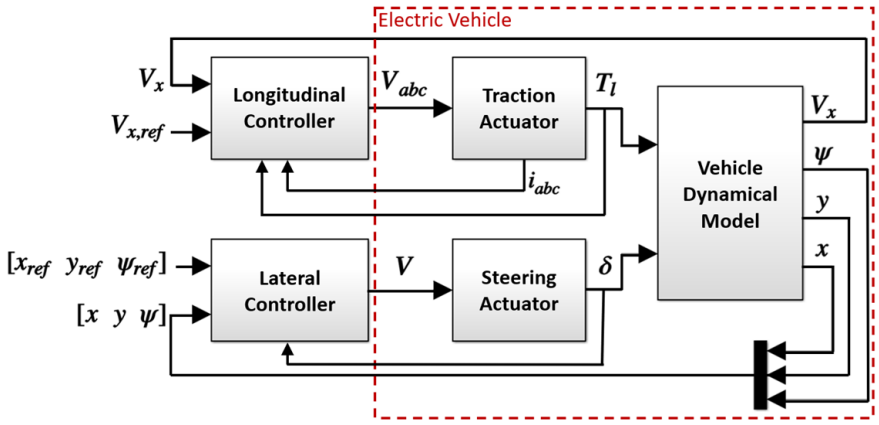


Figure 3.22: Overall control scheme for an electric vehicle.

3.2.3 The Electric Vehicle Model

The primary contribution of this chapter revolves around the development of a comprehensive model that encompasses dynamics often overlooked or unexplored in other models. The proposed model incorporates the coupling of the vehicle's dynamic model, lateral dynamics model, longitudinal movement model, and actuator models. Consequently, to achieve the desired trajectory for the vehicle, the overall control scheme depicted in Figure 3.22 is presented as part of this work.

The dynamic model proposed in this work confidently integrates various interconnected subsystems, namely the longitudinal dynamics, the lateral dynamics, and the dynamics of the vehicle tires, which are depicted in subsections 3.1 to 3.3, respectively. Moreover, our comprehensive model considers the intricate dynamics of the actuators inside the electric vehicle, which in this work a DC motor and BLDC motor are used; and their respective mathematical model is included in subsections 3.4 and subsections 3.5.

The seamless integration of these subsystems empowers the complete dynamic model to function as an advanced testing platform, enabling comprehensive assessment of controller performance in highly intricate and dynamic environments.

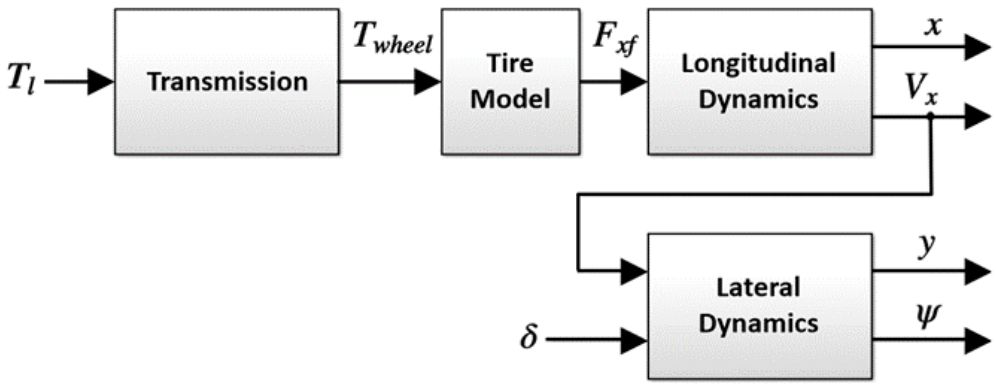


Figure 3.23: Vehicle dynamical model.

3.2.3.1 Longitudinal Dynamics

The longitudinal dynamics describe all the physics involved when the vehicle is shoved in the longitudinal direction by the tires. When model this, we can follow how the forces are generated from the motor, maybe a Spark-Ignition (SI) engine, maybe an electrical machine, that connects to a clutch or torque converter for SI engines or directly to a transmission gear for many electrical machines, and this transmits a torque to the wheels, and these generate the forces when the rubber of the tires and the road interacts. Besides these forces, the vehicle interacts with inertia, aerodynamic forces, gravity etc. Fig. 3.23 shows the elements involved in the vehicle longitudinal dynamics and it is explained in [72] and [28].

The components of the driveline are shown in Fig. 3.23 according to [72] together with the lateral dynamics. This is the path that the torque follows from its generation in the electrical motor (or combustion engine) to the wheels where the longitudinal force applied to the vehicle is generated with the friction of the rubber of the tires against the road. The components analyzed here are:

1) The transmission. This element converts the high speed of the crank shaft to a strong torque to the wheels coupling both making use of different sets of gears for spark-ignition (SI) engines, but for electrical motors the most common is to use only one size gear. 2) The wheels dynamics. This refers to the modeling of the forces involved in the tires due to the interaction of the rubber against the road, the rim, etc. 3) The longitudinal dynamic. The forces generated by the tires causes the movement of the vehicle, however, more forces are involved in this dynamic such as the air drag, the

inclination of the road, the rolling resistance, etc.

The longitudinal dynamics of the vehicle is defined as

$$\ddot{x} = \frac{F_{xf} + F_{xr} - F_{\text{aero}} - R_{xf} - R_{xr} - mg\sin\theta}{m}, \tag{3.93}$$

where the aerodynamic drag force is

$$F_{\text{aero}} = \frac{1}{2}\rho C_d A_F(\dot{x} + V_{\text{wind}})^2, \tag{3.94}$$

rear and front normal forces per wheel:

$$F_{zrw} = \frac{F_{\text{aero}}h_{\text{aero}} + m\ddot{x}h + mg\sin\theta + mgl_f\cos\theta}{N_r(l_f + l_r)}, \tag{3.95}$$

$$F_{zfw} = \frac{-F_{\text{aero}}h_{\text{aero}} - m\ddot{x}h - mg\sin\theta + mgl_r\cos\theta}{N_f(l_f + l_r)}, \tag{3.96}$$

front and rear rolling resistance are

$$R_{xf} + R_{xr} = f(N_f F_{zfw} + N_r F_{zrw}), \tag{3.97}$$

$$R_{xf} + R_{xr} = fmg\cos\theta; \tag{3.98}$$

where F_{xf} is the longitudinal tire force at the front tires, F_{xr} is the longitudinal tire force at the rear tires, F_{aero} is the aerodynamic drag force, R_{xf} and R_{xr} are the forces due to the rolling resistance at the front and rear tires,N_f, N_r are the number of tires in the front and rear axis, θ is the angle of inclination of the road on which the vehicle is traveling, m is the mass of the vehicle, g is the acceleration of the gravity, ρ is mass density of air, Cd is the aerodynamic drag coefficient, A_F is the frontal area of the vehicle, V_{wind} is wind velocity, h_{aero} is the height at which equivalent aerodynamic drag force acts, F_{zf} and F_{zr} are the front and rear normal tire forces.

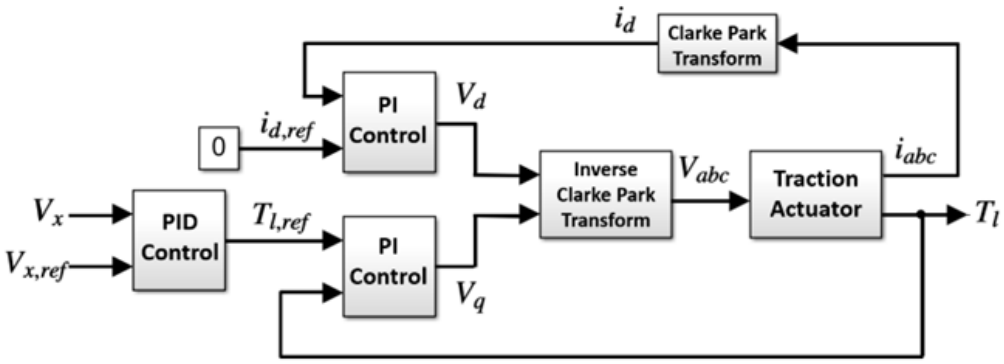


Figure 3.24: Longitudinal controller block diagram.

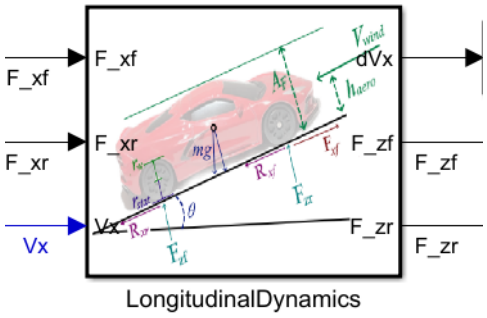


Figure 3.25: Longitudinal dynamics Simulink® block.

3.2.3.2 The Transmission

This block is a simple one. It only consists in the change of speed and torque generated form the motor and transmitted to the wheels based in a gear ratio. For SI engines with automatic transmissions we could add the torque converter dynamics shown in chapter 3.1.1.4. The transmission block receives the torque coming from the motor T_t, the angular speed of the wheels w_w, the gear ratio R, and as output we have the torque transmitted to the wheels T_wheels, and the angular speed of the motors shaft that connects the transmission w_t.

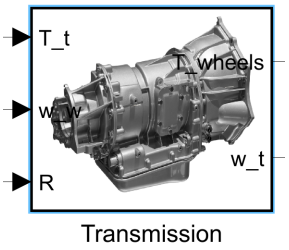


Figure 3.26: Simulink® block of the transmission gear.

3.2.3.3 The Tire Model

The tire model involves from the torque applied to the wheel that generates an angular speed and acceleration to the force generated by the friction with the ground. A key concept in this regard is the tire slip Eq. (3.100) which is the ratio of the difference of the vehicle actual velocity V_x and the linear speed of the wheels $r_{eff}\omega_w f$. This tire slip value is the input to model the actual tire force using the magic formula (MF) Eq. (3.99) that is an empirical equation that fits experimental data and its parameters depends on the conditions of the road such as if it is dry or wet, asphalt paved etc. and the characteristics of the tire [64] [28] [72]. The MF for tire force is defined as

$$F_{xf} = \tag{3.99}$$
$$F_{zf} D \sin\left(C \tan^{-1}\left[B\sigma_{xf} - E\left(B\sigma_{xf} - \tan^{-1}(B\sigma_{xf})\right)\right]\right),$$

where the F_{xf} represents the force produce by the tires and the σ_{xf} is the tire slip, D is the peak value (MF for tires). C is the shape factor, B is stiffness factor, E is the curvature factor. There are several forms to express the tire slip; In this case, for the simulation purpose, we will use the practical slip [64] [28] , where σ_{xf} is negative in deceleration and positive when it is in acceleration. The tire slip is

$$\sigma_{xf} = \frac{r_{\text{eff}}\,\omega_{wf} - \dot{x}}{|\dot{x}|}, \tag{3.100}$$

where the effective radius of the tire [72] r_{eff} is defined as

$$r_{\text{eff}} = \frac{\sin\left(\cos^{-1}\left(\frac{r_{\text{stat}}}{r_w}\right)\right)}{\cos^{-1}\left(\frac{r_{\text{stat}}}{r_w}\right)} r_w. \tag{3.101}$$

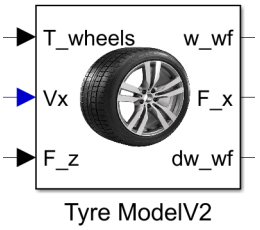


Figure 3.27: Simulink® block of the tire.

Once we get the force produced by the tires, we can calculate the dynamics of the driving wheels [72]:

$$I_w \dot{\omega}_f = T_{\text{wheel}} - r_{\text{eff}} F_{xf}. \tag{3.102}$$

For the non-driven wheels, we have:

$$I_w \dot{\omega}_f = -r_{\text{eff}} F_{xr}; \tag{3.103}$$

where the transmission dynamics and angular speeds are:

$$T_{\text{wheel}} = \frac{1}{R} T_t, \tag{3.104}$$

$$\omega_t = \frac{1}{R} \omega_w. \tag{3.105}$$

The symbols for the tire model are as follows: σ_{xf} is longitudinal tire slip, r_{eff} is the effective radius of the wheel, r_{stat} is the height from the ground to the axis of the wheel with load, ω_w is the angular velocity of the wheel. ω_{wf} is the angular velocity of the front wheels. I_w is the momentum of the wheel. T_{wheel} is the torque applied to the wheels coming from the engine. Nf, Nr Number of wheels in front and rear axles, R is the gear ratio and depends on the operation gear.

Below is the Simulink® block for the tire model Fig. 3.27. The equations contained in this block are the one from equations 3.99 to eq.3.105. T_wheels is the torque coming from the motor and transmission gear, Vx is the longitudinal speed, F_z is the normal force in the tire. As output, the module gives w_wf that is the angular speed of the wheel, F_x is the longitudinal force that the tire is creating and the one that is pushing the car forwards. dw_wf is the angular acceleration of the tire.

In this figure 3.28 is shown the internal structure of the module with three modules that comprises

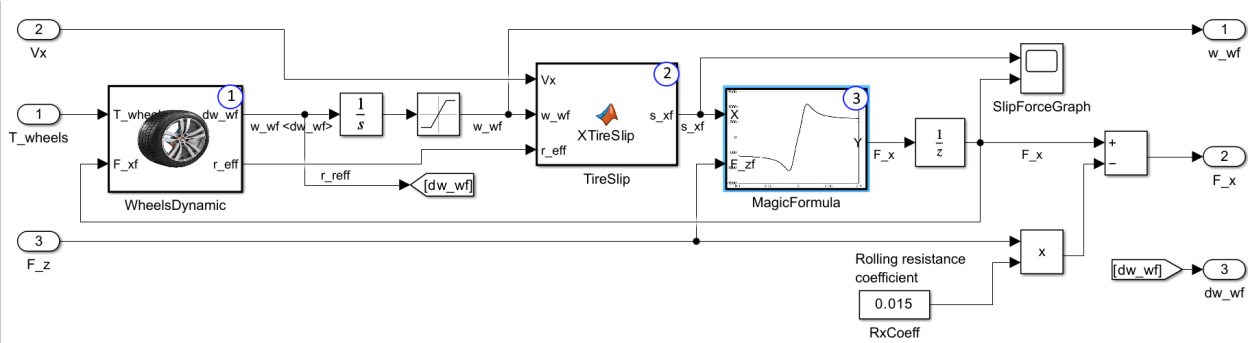


Figure 3.28: Block diagram in Simulink for the tire model.

1) the wheel dynamics, 2) the tire slip, and 3) the magic formula. Below are the code of the three sections.

3.2.3.4 Lateral Dynamics

The lateral dynamics cover the lateral motion and yaw angle dynamics of the vehicle. The analysis of these dynamics nowadays is used to develop control systems for lateral applications such as lane departure warning systems (LDWS), lane keeping systems (LKS), yaw stability control systems etc.

The main blocks for the lateral dynamics are shown in Fig. 3.29 where the blocks are the lateral controller, a BLDC motor as actuator which is in charge to position the tires angle, the lateral dynamic model, and the conversion of the coordinates from body-fixed to inertial reference frame. The mathematical model that describes the lateral dynamics of the vehicle is as follows, eliminating the simplification of small angle "$\tan\theta = \theta$" [72]:

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix}, \tag{3.106}$$

$$\frac{d}{dt}\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} \dot{y} \\ \frac{2C_{\alpha f}}{m}(\delta - \theta_{vf}) + \frac{2C_{\alpha r}}{m}(-\theta_{vf}) - \dot{\psi}V_x \\ \dot{\psi} \\ \frac{2l_f C_{\alpha f}}{I_z}(\delta - \theta_{vf}) - \frac{2l_r C_{\alpha r}}{I_z}(-\theta_{vf}) \end{bmatrix}, \tag{3.107}$$

where

$$\theta_{vf} = \tan^{-1}\left(\frac{\dot{y} + l_f\dot{\psi}}{V_x}\right). \tag{3.108}$$

The symbols are ψ heading angle (yaw angle), δ front steer angle, l_f and l_r are distances from the center of gravity (CoG) to the front and rear axis of the vehicle, $L = l_f + l_r$, $C_{\alpha r}$ and $C_{\alpha f}$ are cornering stiffness of rear and front wheels. The cornering stiffness is the ability of the tire to resist deformation while the vehicle is cornering. I_z is the momentum of inertia of the vehicle respecting the z axis, ${}_V x$ is the longitudinal velocity, m is the mass of the vehicle. For a more detailed information regarding the lateral dynamics please refer to [72], and [28].

The definition of the error variable of the position of the vehicle with respect to the center of the road, commonly used for keeping lane control applications, it is defined as:

$$\ddot{e}_1 = (\ddot{y} + V_x\dot{\psi}) - \frac{V_x^2}{R} = \ddot{y} + V_x(\dot{\psi} - \dot{\psi}_{\text{des}}), \tag{3.109}$$

where $\ddot{y}$ is the lateral acceleration, $\dot{\psi}$ is the angular velocity of the vehicle orientation, V_x is the longitudinal velocity, and R is the radius of the imaginary circle where the vehicle is turning around. The error variable e_2 corresponds to the desired orientation of the vehicle, defined as

$$e_2 = \psi - \psi_{\text{des}}. \tag{3.110}$$

Now, integrating Eq. 3.109 and considering V_x as constant, results in

$$\dot{e}_1 = \dot{y} + V_x(\psi - \psi_{\text{des}}), \tag{3.111}$$

where e_1 is the distance of the CoG of the vehicle from the center line of the road, and e_2 is the orientation error of the vehicle with respect to the road.

The image below Fig. 3.30 shows the complete scheme of the vehicle.

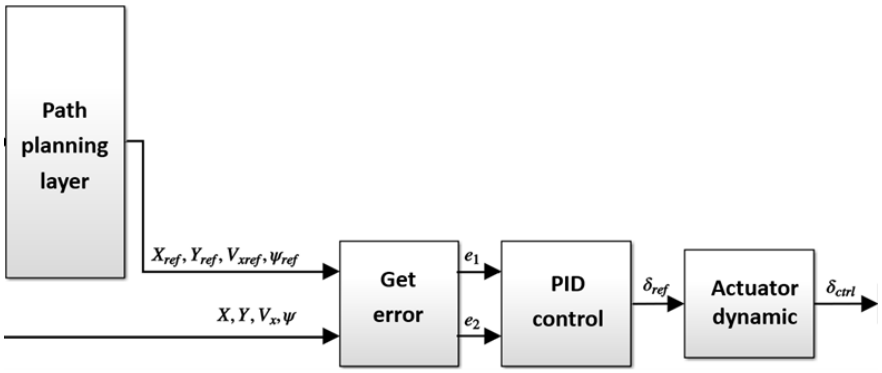


Figure 3.29: Lateral controller block diagram.

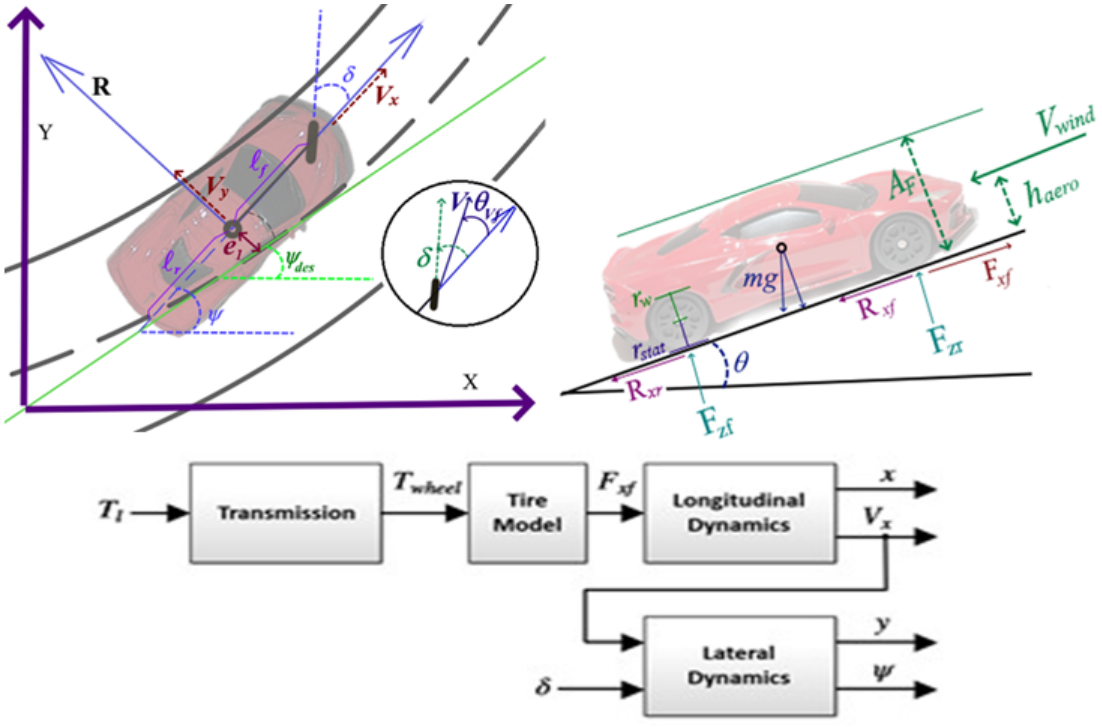


Figure 3.30: Longitudinal and lateral dynamic model.

3.2.3.5 Variant Blocks and the Complete Model

Here are presented the missing blocks in our model, but the ones that we want to put on test using the model. I refer to the controllers and actuators. Here we show the blocks that will serve as shell to test diverse algorithms. These blocks are defined in Simulink® as ”Variant” blocks. The controllers for the lateral and longitudinal dynamics will be defined here, and the actuators such as the electrical motors for the steering wheel or the propulsion as well.

Figure 3.31 and 3.32 shows the blocks for the longitudinal and lateral dynamics put together, respectively. Notice the feedbacks. Algebraic feedback loops require a memory with the value of one time step before. This memory can be implicit or explicit. To define explicitly this memory, we can add a block ”1/z” that means a time delay in the Z transform. If we do not use this block explicitly, Simulink® will put one hidden by us, however, some times Simulink® can not solve the loop by itself and then we need to forcibly specify it by hand. Dynamic loops are specify by the block ”1/s” that means the integral in the Laplace transform. This integral block solves the feedback loop and it is not needed to add the ”1/z” block. The model to run needs to specify the solver and the time step for the case of discrete solvers. The solver is the algorithm to solve the differential equations of the model, for example the Euler algorithm or the Runge-Kutta. Figure 3.33 shows the configuration of the solver which is defined as Fixed-Step and the selection of the algorithm will be chosen automatically by Simulink®. Take care of the fixed step size configuration. If the step size is too small, it will take too long to finish the simulation. If it is to big, the result will not be correct. If we are not using the electrical motors in the model, the 0.002 seconds is enough. However, if we introduce electrical motors, it will require to reduce the step size as smaller as 0.000002 seconds, otherwise the result can be unpredictable; this is because the model becomes a stiff problem which require a very small step size to be solved. Next is the description of the model.

Figure 3.31 shows the longitudinal blocks. The block marked with 1) is just a stimuli block to specify the desired velocity of the vehicle. 2) is a PID controller to control the speed defining the torque required to the block 3), this block will contain the electrical motor or SI motor to be chosen. This is a ”variant” block. 4) is the block transmission. 5) and 6) are the blocks for the front and rear tires respectively. Notice that the torque is divided by 2 at the input and multiply it by 2 in the

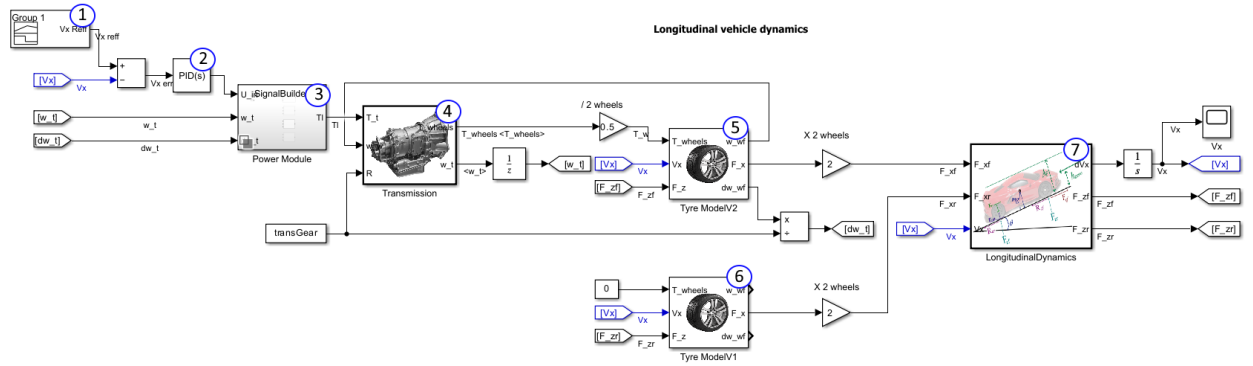


Figure 3.31: Simulink® blocks of the longitudinal dynamics.

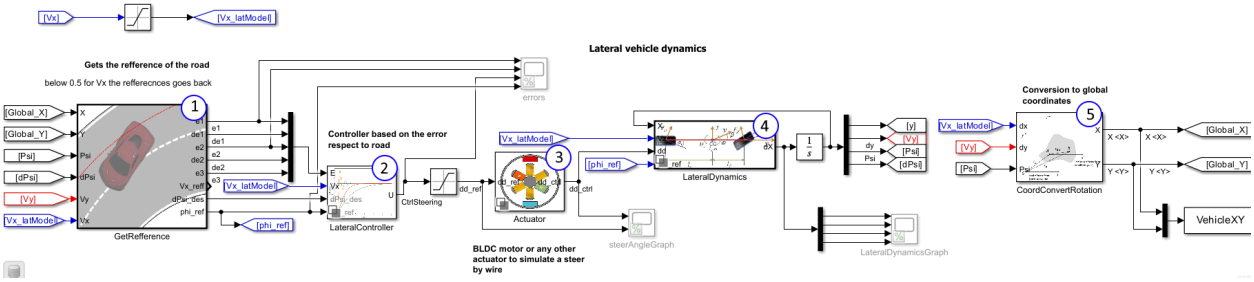


Figure 3.32: Simulink® blocks of the lateral dynamics.

output; this is because this is the model of one tire but we have only one motor generating the torque for the 2 drive wheels. 7) is the longitudinal dynamics.

Figure 3.32 shows the blocks for the lateral dynamics. 1) is the block that read the waypoints for reference and calculates the error respect our current position and orientation. 2) is a ”variant” block where we can choose different control algorithms for the lateral dynamics. 3) is a ”variant” block where we can choose the kind of actuator such as BLDC, a generic transfer function or not actuator at all. 4) is the lateral dynamics and 5) is the inertial global coordinates calculation block.

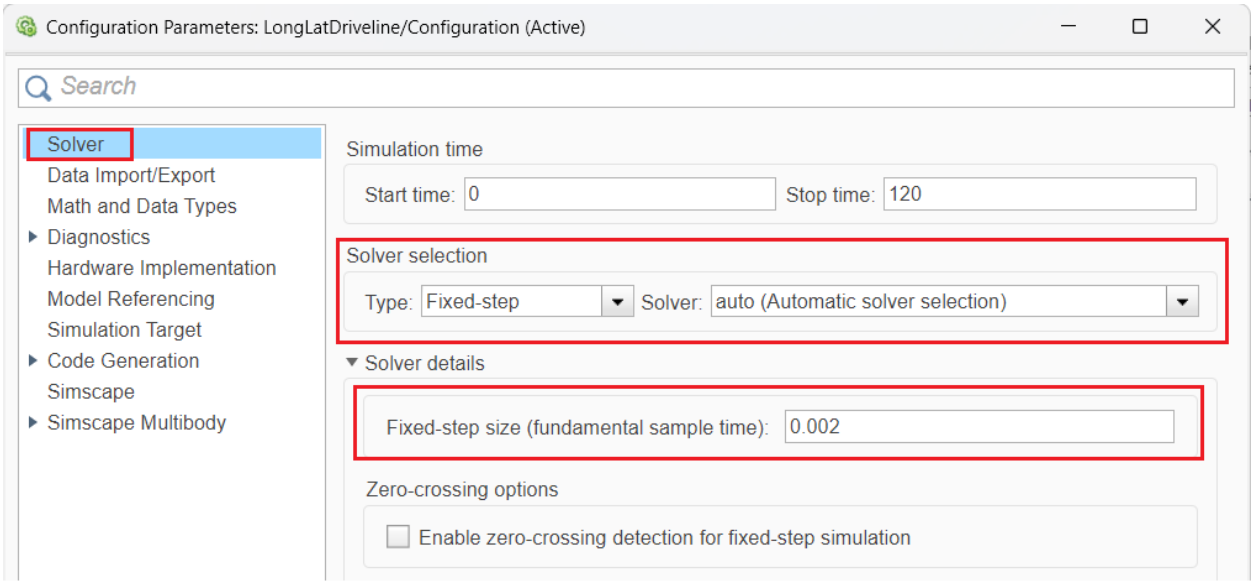


Figure 3.33: Configuration of the solver in Simulink®.

3.2.4 Simulations

In this section we show the simulations with everything put together. We run 120 seconds of simulation with a fixed-step size of 5E-6 seconds and solver ode3. particularly, the small step size required is due to the BLDC equations in the dq0 reference frame transformation. These equations, due to the response in the simulations, that needs a very small step size, could mean that they are ”stiff” differential equations [18].

Fig. 3.17 shows the trajectory of the vehicle chasing the path form by the way-points in the file. a) The blue line is the actual trajectory of the vehicle. The red line is the sequence of points of the file of way-points. In the b) graph is shown a close-up in a part of the route with an abrupt turn to see the error in the trajectory.

Fig.3.34 shows the simulation in the current and torque provided by the electrical motor BLDC used to power the vehicle. That is the motor connected to the transmission. Above in figure, I_abc represents that current. Below, is the torque in Nm, produced and delivered to the transmission. we see that while we reach the target speed, the torque is reduced. We can observe the current increasing as we reach the target vehicle speed, see Fig. 3.38 and Fig. 3.39.

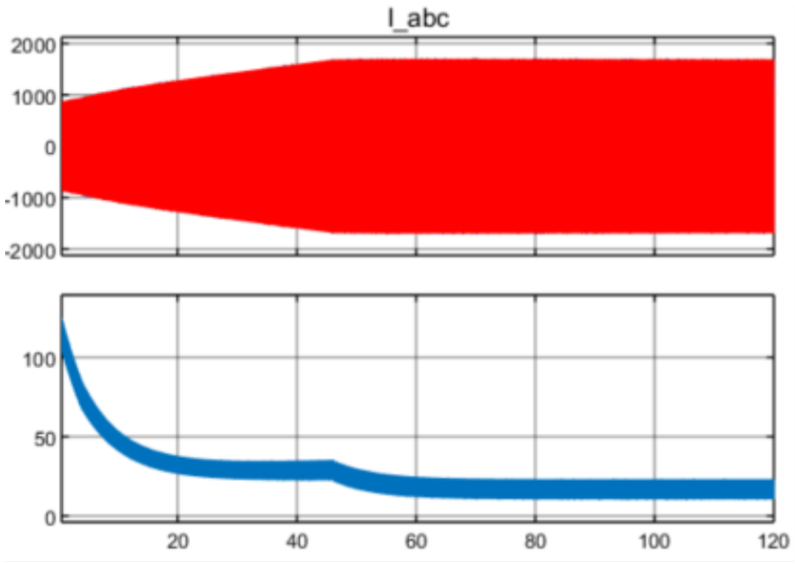


Figure 3.34: Current and torque generated by the BLDC for longitudinal dynamics. Horizontal is time is in seconds. Vertical is amperes.

Figs. 11-13 Show the measurements of the errors in the lateral dynamics and the response in current, and the angle of the BLDC that moves the steering. In Fig. 11, e1 is the error of the position of the vehicle against the center of the lane. e2 is the error in the yaw angle or vehicle orientation. CtrlSteering is the angle commanded by the lateral controller. Fig. 12 Is the angle of the motor BLDC used for steering. Fig. 13 Currents in amps. of the BLDC used for steering.

Fig. 3.38 shows the results of the simulation from the tire dynamics perspective. a) Above, the s_xf is the tire slip that is always close to 0. Below, F_x is the force that the tire generates that is the actual force that moves the vehicle. We can see the tire slip and the force delivered is reduces gradually until it stabilizes when the vehicle speed goal is reached. See Figs 3.39-3.40.

Figs. 3.39-3.40 show the longitudinal dynamics response. Fig. 3.39 is the actual vehicle speed along the 120 seconds of simulation. In Fig. 3.40 Vx_reff is the speed set-point along the simulation. We start from 10m/s to 16 m/s. Vx_err is the difference of Vx_reff minus Vx. PID_Controller1 is the control law coming from the PID, see Fig. 3.24. Tl is the torque delivered by the motor to the transmission.

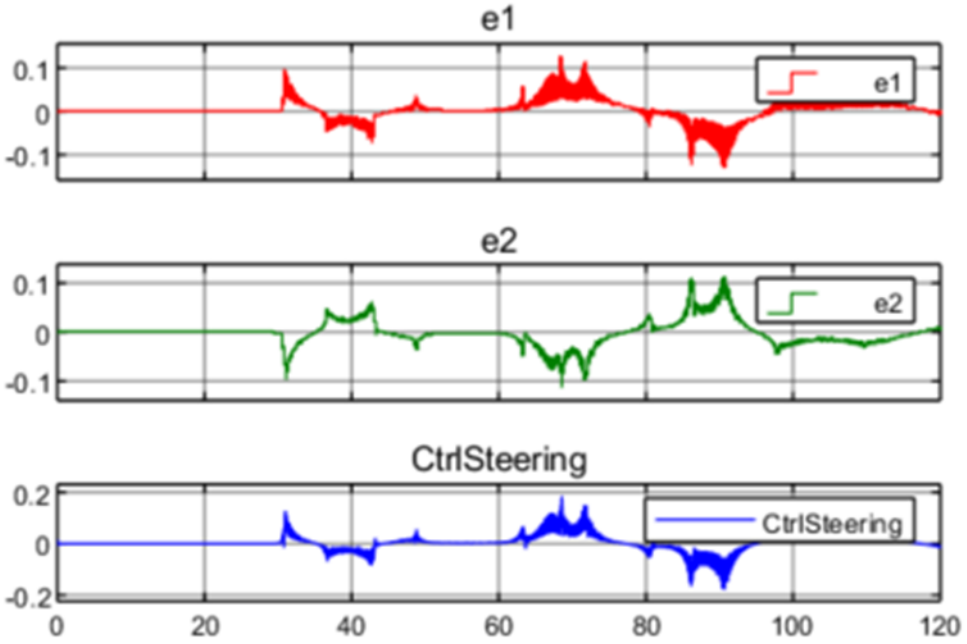


Figure 3.35: e1 is the error of the position of the vehicle against the center of the lane.

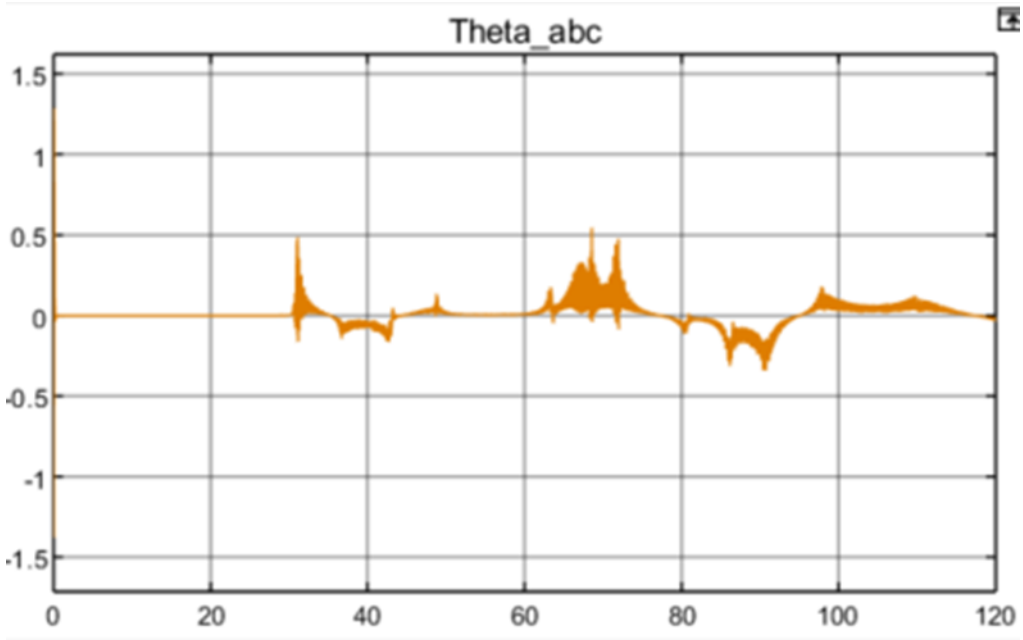


Figure 3.36: Angle of the motor BLDC used for steering in rads.

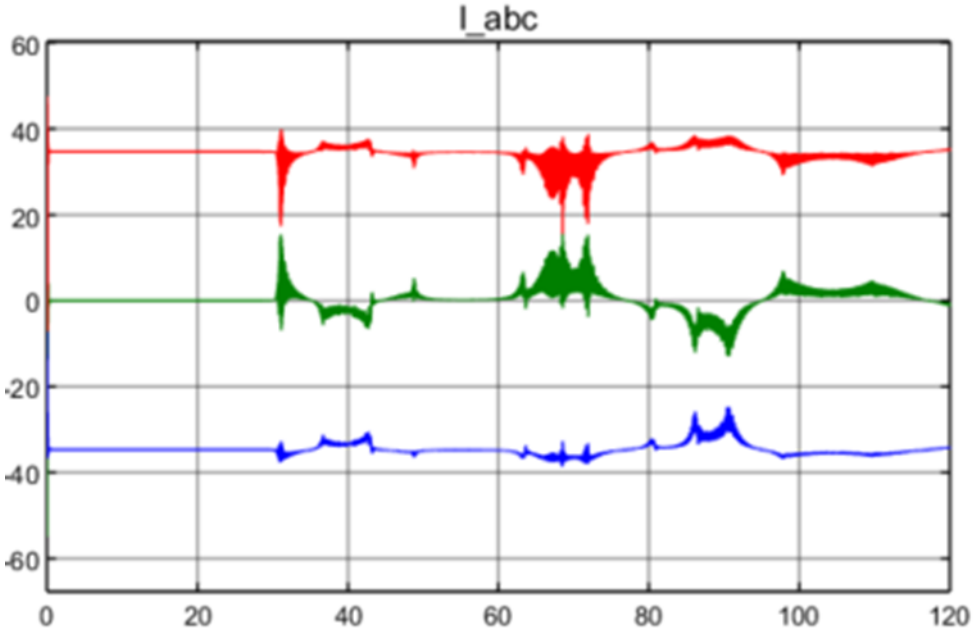


Figure 3.37: Currents in amps. of the BLDC used for steering. Horizontal is time in seconds. Vertical is amperes for the three phases.

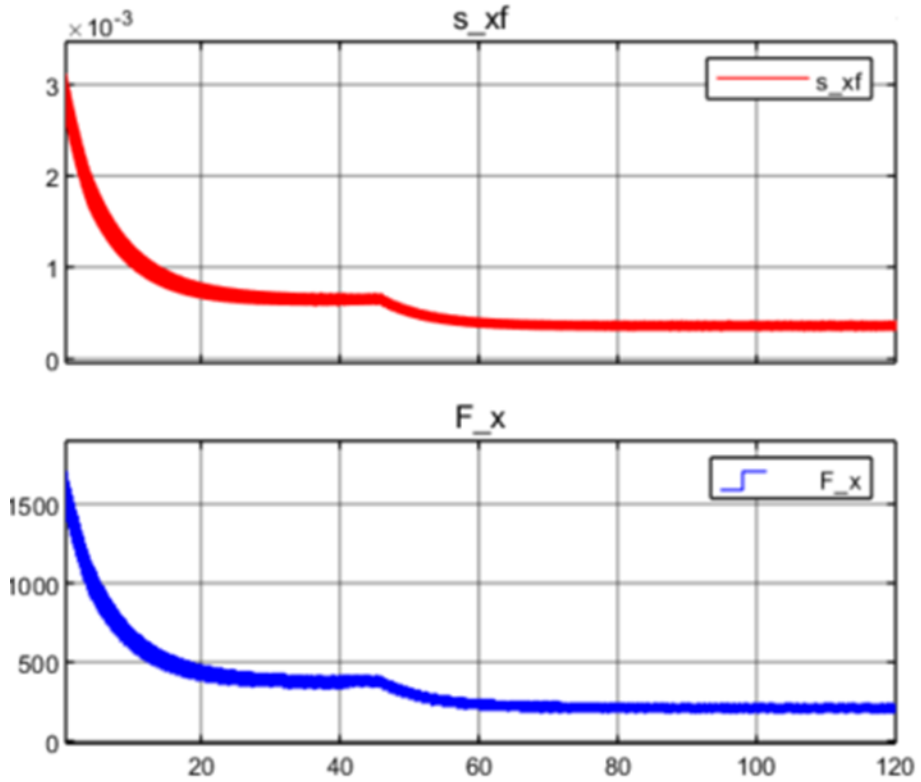


Figure 3.38: Tire dynamics simulation: Above is the tire slip. Below is the force in N produced by the tire.

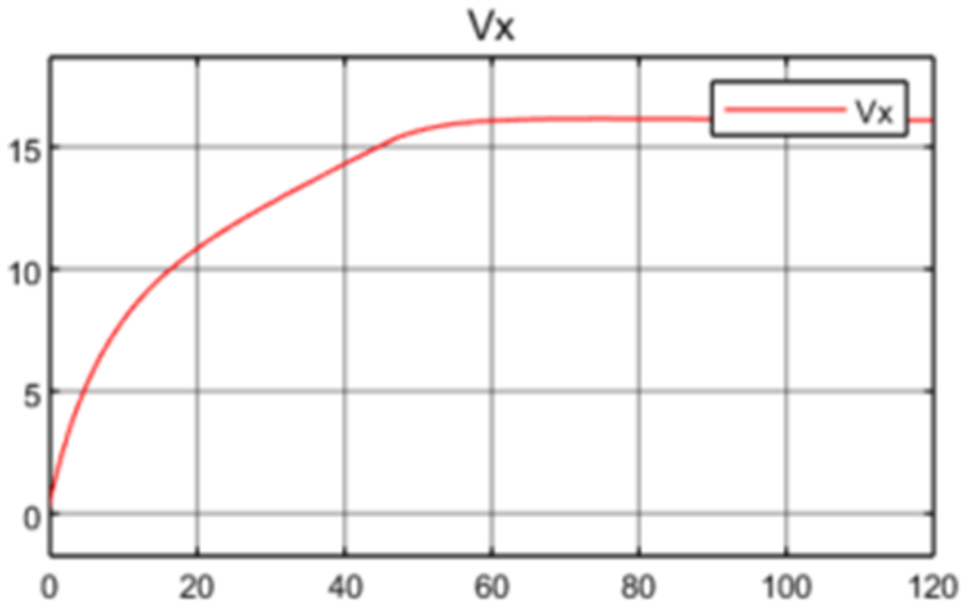


Figure 3.39: Longitudinal dynamics simulation: Vx is the actual vehicle speed during the 120 seconds of simulation.

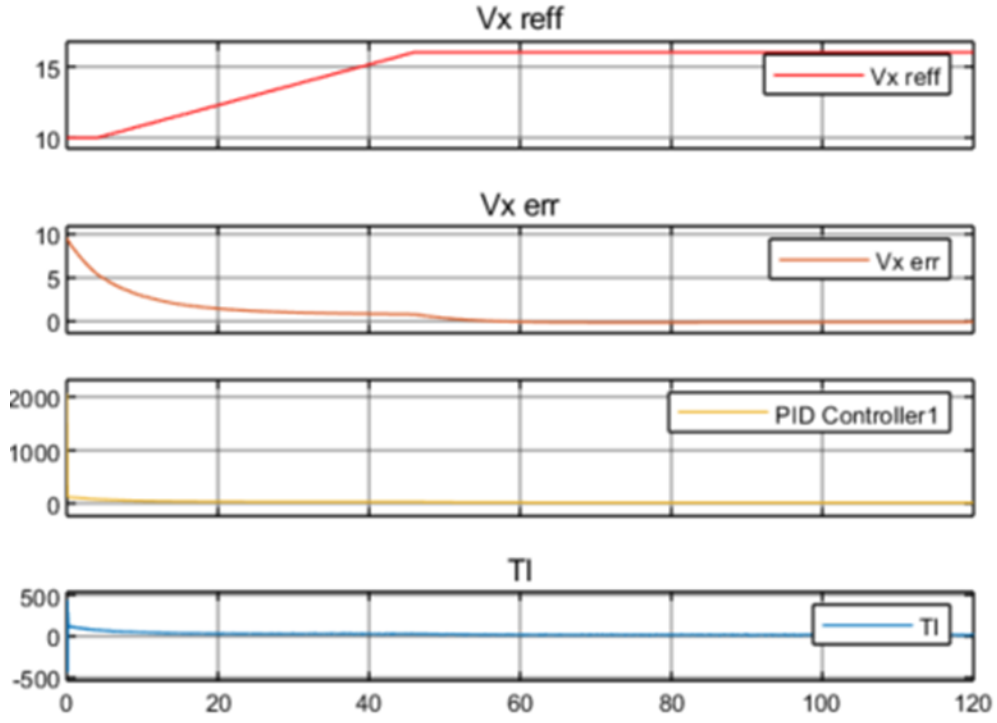


Figure 3.40: Longitudinal dynamics simulation: Vx_reff is the velocity that the longitudinal controller must follow. Vx_err is Vx_reff minus the actual velocity. PID_controller1 is the control law. Tl is the torque in Nm delivered by the electrical motor to the transmission.

TABLE 3.2. BLDC ELECTRICAL MOTOR TO POWER AND STEER THE VEHICLE

No.	Name	Sym	Unit	Val power	Val steer
1	Name model	-	-	Generic	Generic
2	R	R	Ω	0.0075	0.35
3	L	L	H	0.00183	0.001
4	M	M	H	0	0
5	J	J	Kg m^2	0.04	0.00018
6	P	P	-	4	4
7	b	b	N m s rad^{-1}	0.27285	0.0034
8	lambda	λ	V/rad/s	0.0836	0.0433
9	Max Torque	max T_l	Nm	430	N/A

TABLE 3.3. VEHICLE DATA

No.	Matlab Name	Sym	Unit	Val
1	lr	l_r	m	1.58
2	lf	l_f	m	1.1
3	transGear	R	-	0.1027
4	Iz	I_z	Nm	2873
5	m	m	Kg	2238.932
6	Nf	N_f	-	2
7	Nr	N_r	-	2

TABLE 3.4. ENVIRONMENT DATA

No.	Matlab name	Sym	Unit	Val
1	g	g	m/s^2	9.81
2	V_wind	V_{wind}	m/s	0
3	theta	θ	rad	0

TABLE 3.5. AERODYNAMICS DATA

No.	Name	Sym	Unit	Val
1	Mass density air	ρ	kg/m^3	1.225
2	Frontal area	A_F	m^2	2.341157
3	Drag area	$C_d A_F$	m^2	0.57599
4	Aero drag coeff	C_d	-	0.24
5	Height drag force	h_{aero}	m	0.65

TABLE 3.6. TIRE DYNAMICS DATA

No.	Name	Sym	Unit	Val
1	Name Model	-	-	Generic
2	C_ar	$C_{\alpha r}$	-	80000
3	C_af	$C_{\alpha f}$	-	80000
4	D	D	-	3
5	C	C	-	1.5
6	B	B	-	20
7	E	E	-	0
8	sh	S_h	-	0
9	sv	S_v	-	0
10	r_stat	r_{stat}	m	0.352
11	r_w	r_w	m	0.342

3.2.5 Conclusions

This model resulted useful to help design and test controllers for the longitudinal and lateral dynamics integrating the dynamics of electric motors as actuators to analyze the concept of electric vehicles. Despite that the simulation for 120 seconds took around 2 hours, the advantage of this model against the Simulink® default example, is that it uses only the mathematical concepts and can be analyzed internally. Data can be logged from inside any block and be used for training artificial neural networks (ANN) to create model predictive controllers based on ANNs of any part of the model. As next steps in this direction, we will create controllers based in sliding modes. The space vector pulse-width modulation (SVPWM) logic is already implemented in one block that works isolated, but it did not work integrated to the complete model yet, remaining for next steps too.

4. Sliding Mode Controllers for Electric Vehicles

4.1. Path Following Sliding Mode Controller for an Electric Vehicle Considering Actuator Dynamics

This study introduces a novel path-following controller tailored for electric vehicles equipped with a steer-by-wire system i.e. the steering angle of the vehicle was defined by an electrical actuator. The control objective is to force the proper steering angle of the vehicle which permits to follow a desired path. The system presupposes that an external algorithm, utilizing sensors data, provides the lateral movement references, while maintaining a steady longitudinal velocity for the vehicle. The proposed control scheme is based on a robust sliding mode steering controller to manage the vehicle's lateral movement. Furthermore, a Brush-Less DC (BLDC) motor is considered as the steering actuator which is controlled by a Field-Oriented Controller (FOC) which is based on four internal Proportional-Integral (PI) control loops for precise steering actuation. To assess the performance of the proposed control scheme, numerical simulations were obtained which demonstrates its effectiveness in achieving the control objective.

4.1.1 Introduction

Advanced driving assistance systems are evolving rapidly with the ultimate goal to achieve full self-driving vehicles which can be applied in many fields and industries such as transportation and mobility, surveillance, healthcare, agriculture, construction, manufacturing, among others. One of the advantages of these systems is the improvement of the safety of passengers and operators of machinery in roads since these systems do not get tired, or to commit unsafe behaviors or risky actions.

We can mention some automotive systems intended to keep the vehicle on track or to assist

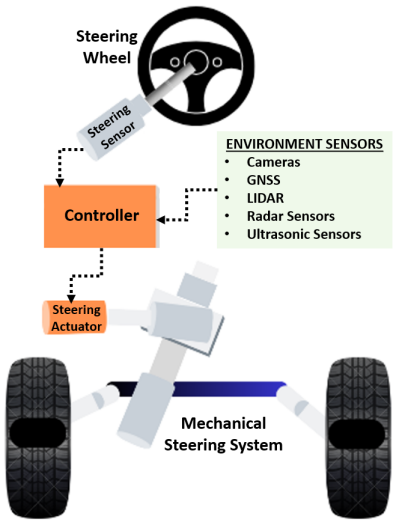


Figure 4.1: Overall control architecture for path following with different levels of autonomy.

the driver in this task. Some of these systems are known commercially as Lane Keeping Assist Systems (LKAS) whose objective is to help the driver to maintain the vehicle on its respective lane, the Lane Departure Avoidance (LDA) SAE L0 (No automation) which corrects the steering angle of the vehicle to prevent departure from a specific lane, the Emergency Lane Keeping System (ELKS) which is an adaptive application to reduce the effort required by the driver to keep his vehicle centered in the lane, the Lane Change System SAE L2 (partially automated system) which, after an initial command or confirmation by the driver, applies automatically steering to move the vehicle to an adjacent lane; and, of course, the full automation driving from SAE L4 to L5.

These systems provide different levels of autonomy to the vehicle which are based on automatically controlling its inputs as steering angle, traction torque, braking, among others. Usually, a path following scheme for a vehicle considers the integration of a steer-by-wire system as depicted in Fig. 4.1 which represents a simplified version of the real system. In this case, the steering wheel does not have a physical connection to the mechanical steering system. Instead, a controller monitors the steering wheel using a steering sensor and determines how mechanical system is modified by means of a steering actuator. Furthermore, the controller receives information about the environment of the vehicle from a set of sensors as cameras, a global navigation satellite system (GNSS), LIDAR, radar and ultrasonic sensors. It is important to note that the performance of the global control scheme depends fundamentally on the accuracy and precision of the environment sensors and the algorithms

that permit to monitor the environment and, in particular, the road. This represents a very promising research area in the field of autonomous vehicle but it is not within the scope of the presented work.

The controller block in Fig. 4.1 is designed using different approaches. For instance, in [2] two nonlinear controllers were developed for autonomous vehicles, [75] which presents a path-following control algorithm for a vehicle. Some controller design techniques include adaptive control [98], optimal control schemes [23], [31], algorithms based on model predictive control (MPC) [73], [90], and controllers using artificial intelligence algorithms as reinforcement learning [84] and others such as or [29], [43], [33], [14]. On the other hand, other works were focused on taking advantage of the robustness of the sliding mode control (SMC) algorithms as any vehicle will encounter the effect of disturbances due to external and internal factors as wind, road irregularities, noise in sensors, model uncertainties and parameter variations. Sliding mode control is a nonlinear control technique that aims to compensate the discrepancies existing between the real plant and the mathematical model used to analyze and design a control algorithm. Essentially, the SMC consists of switching the inputs of a system at high frequency in order to force the states of the system to converge to a manifold, usually expressed as function of these states [86]. Moreover, the control law is designed to assure that the state of the system remains within the sliding surface in any future instant. Generally speaking, the control consists of two parts: one in charge to compensate the linear part of the system and the other designed to establish some robustness on the response of the system. The second part is usually modelled by means of a switching function and hence, this kind of control is also known as variable structure control (VSC). An important aspect of the system when enters into a sliding mode, known as the sliding motion, is that the order of its dynamics is reduced with respect to the original system. This approach is used in [77] and [59] where trajectory planning and tracking controllers for automotive vehicles were presented. In addition, a combination of sliding mode controller and neural networks were proposed in [36] for path following of autonomous vehicles. Moreover, in [95] and [91], robust sliding mode controllers were proposed with the inclusion of prediction capabilities and an extended state observer, correspondingly. Also, these control strategies which provide autonomy to automotive vehicles can be applied to other types of vehicles as tractors [46] and marine surface vehicles [93], [51].

None of the referenced works took into consideration the dynamical model of the actuators as part

of their control strategies. This is important because the features of the actuators as response time, sensitivity and bandwidth define the performance of the controller in a real application. Furthermore, neglecting the effect of the actuators could compromise the stability of the overall control scheme. In that regard, this work presents the designing and simulation of the model on Simulink of a path following controller for an electric vehicle considering the actuator's dynamics. It consists of two control loops: an inner loop for the steering actuator based on the FOC (Field Oriented Control) algorithm and the standard PI (Proportional Integral) controller; and an outer control loop based on sliding mode control. In section 2, the path following problem for an automotive vehicle is stated, and in section 3 the proposed overall control scheme is depicted which includes the lateral sliding mode controller (outer loop) and the inner control loop for the actuator. It is worth mentioning that the stability for the proposed lateral sliding mode controller in closed loop is demonstrated. In section 4, the simulations results for a predefined trajectory are presented which demonstrates the effectiveness of the proposed control scheme. Finally, in section 5 the conclusions of the presented work are outlined and future research paths for the project are established.

4.1.2 Path Following for an Automotive Vehicle

The control architecture of autonomous vehicle is commonly composed of the following layers [65]: the route planning layer, the behavioral decision making layer, and the motion planning layer. The first, is in charge of constructing and selecting a particular route to be followed by the vehicle among all the possibilities determined by the current position of the vehicle and a destination. The behavioral decision-making layer provides to the vehicle the ability to react to changes in its environment and the interaction with other vehicles, the status of the traffic lights, pedestrians, etc. Moreover, it must modify the behavior of the vehicle dynamically depending on the prediction of the motion of the other elements within its environment to avoid an accident or undesired event. In that sense, the output of this layer relates to a decision as stop, change lane, turn left or right, brake or accelerate, passing other vehicle, among others. Finally, the motion planning layer translates the decision taken by the previous layers into a path or trajectory that shall be followed by the vehicle. To this end, a control algorithm must determine a proper value for the inputs of the vehicle, so its states correspond to the desired path or trajectory. Precisely, the control algorithm proposed in this

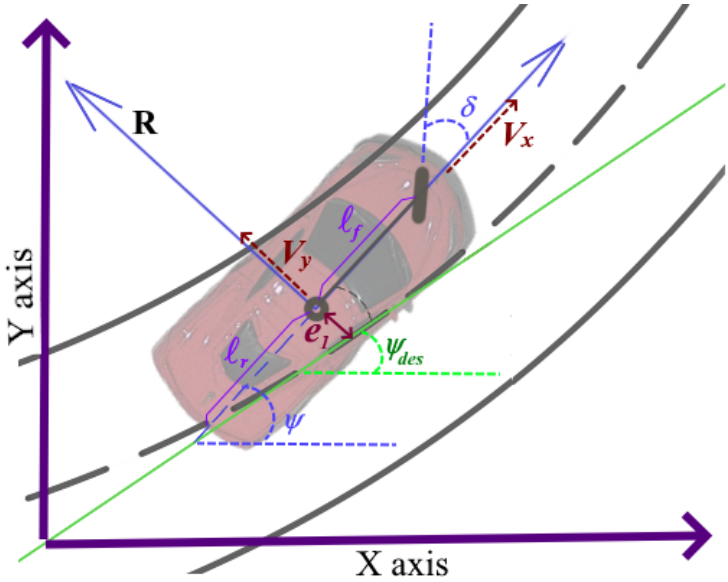


Figure 4.2: Parameters and variables related to the lateral vehicle's dynamics. R corresponds to trajectory instant radius.

work corresponds to the one depicted in the above mentioned autonomous vehicle architecture and, therefore, it is assumed that the desired path to follow was already defined by the upper layers of the architecture.

Most of the path following controllers are designed based on a mathematical model of the vehicle. There are two common versions for a model of the vehicle: one related to the kinematics of the vehicle and the other oriented to describe its dynamics. The first is only concerned to express the geometry defined during the motion of the components of the vehicle without considering the forces and torques that generated this motion, and the latter includes these forces and torques into its equations. The proposed controller considers a dynamical model of the vehicle, in particular, the bicycle model [71]. It consists of a bar that connects the front and rear tires as shown in Fig. 4.2 and it has two degrees of freedom which are the lateral position y and the orientation ψ of the vehicle.

In a path following scenario, the controller's objective is to force the pose (position and orientation) of the vehicle to track a desired pose defined by the path to follow. Let us consider an inertial coordinates system $\{X, Y\}$ and a vehicle-fixed coordinates system $\{x_0, y_0\}$ whose origin is located at the center of gravity (c.g.) of the vehicle and its x_0 axis is aligned with the longitudinal axis of the vehicle. Then, a reference path for the vehicle is expressed in the inertial coordinates

system which permits to define references for the coordinates of the vehicle and its orientation. It is assumed that these references are computed by an instrumentation system, from a higher layer in the software/hardware architecture of the vehicle, by means of data retrieved from sensors as cameras, IMU, LIDAR and ultrasonic sensors. Moreover, it is assumed that an external controller maintains the longitudinal velocity V_x of the vehicle in a constant value.

Within this scenario, the lateral dynamics of the vehicle with respect to the inertial frame, according to [71], can be defined as

$$\dot{X} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & a_{22} & 0 & a_{24} \\ 0 & 0 & 0 & 1 \\ 0 & a_{42} & 0 & -V_x + a_{44} \end{bmatrix} X + \begin{bmatrix} 0 \\ b_2 \\ 0 \\ b_4 \end{bmatrix} \delta + \begin{bmatrix} 0 \\ 0 \\ 0 \\ g \end{bmatrix} \sin\phi, \tag{4.1}$$

where $X = \begin{bmatrix} \psi & \dot{\psi} & y & \dot{y} \end{bmatrix}^T$, y is the distance from the c.g. of the vehicle to its center of rotation, ψ is the heading angle (yaw angle) of the vehicle, δ is the front wheel steering angle, V_x is its longitudinal velocity and ϕ is the road bank angle. The parameters of the model are defined as $a_{22} = -\frac{2C_{\alpha f}l_f - 2C_{\alpha r}l_r}{I_z V_x}$, $a_{24} = -\frac{2C_{\alpha f}{l_f}^2 + 2C_{\alpha r}{l_r}^2}{I_z V_x}$, $a_{42} = -\frac{2C_{\alpha f} + 2C_{\alpha r}}{mV_x}$, $a_{44} = \frac{-2C_{\alpha f}l_f + 2C_{\alpha r}l_r}{mV_x}$, $b_2 = \frac{2C_{\alpha f}l_f}{I_z}$ and $b_4 = \frac{2C_{\alpha f}}{m}$ where l_f, l_r are distances from the c.g. to the front and rear axis of the vehicle, $C_{\alpha r}, C_{\alpha f}$ are the cornering stiffness of rear and front wheels, I_z is the momentum of inertia of the vehicle with respect to normal axis of the reference system, m corresponds to the total mass of the vehicle, and g is the gravity constant.

4.1.3 Overall Control Scheme

The proposed control scheme is shown in Fig. 4.3 where X_{ref}, Y_{ref} and ψ_{ref} are the coordinates and orientation references for the vehicle, which correspond to the pose of the desired path. Furthermore, two control loops can be identified in the control scheme: the outer loop which receives a reference for the pose of the vehicle and generates a reference for the steering angle δ_{ref}; and the inner control loop which receives the steering angle reference and generates the voltages V_{abc} that excites the actuator. The proposed controller considers a BLDC motor as the steering actuator of the vehicle. Hence, the input of the actuator is a 3-phase voltage V_{abc} and its output is the current

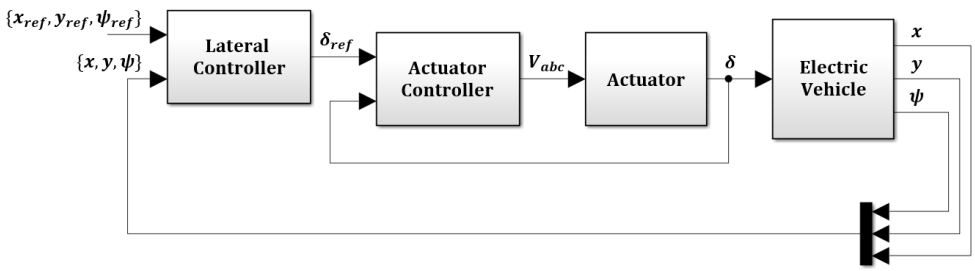


Figure 4.3: Overall control scheme.

steering angle δ of the vehicle.

To design the global control scheme, two control algorithms must be designed: a lateral controller that receives a pose reference for the vehicle, defined by X_{ref}, Y_{ref} and ψ_{ref}, and generates a desired steering angle δ_{ref}; and an actuator controller that receives δ_{ref} and generates the input voltage V_{abc} for the BLDC motor of the steering system. These controllers will be designed in the rest of this section.

4.1.3.1 Lateral Sliding Mode Controller

As mentioned before, the lateral controller receives the references X_{ref}, Y_{ref} and ψ_{ref} from the instrumentation sub-system of the vehicle. In addition, it also provides the current coordinates X,Y of the vehicle and its current orientation ψ. From these values, it is possible to define the error variables e_1 as the distance between the vehicle's c.g. and the center line of the road, and the error variable $e_2 = \psi_{ref} - \psi$. Then, an error variables vector $e = \begin{bmatrix} e_1 & \dot{e}_1 & e_2 & \dot{e}_2 \end{bmatrix}^T$ is established as depicted in Fig. 4.2. Hence, using (4.1) the error variables dynamics yields as follows,

$$\dot{e} = Ae + B\delta + L \tag{4.2}$$

with

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & 0 & 1 \\ 0 & a_{42} & a_{43} & a_{44} \end{bmatrix}, B = \begin{bmatrix} 0 \\ b_2 \\ 0 \\ b_4 \end{bmatrix}, L = \begin{bmatrix} 0 \\ l_1 \\ 0 \\ l_2 \end{bmatrix},$$

$a_{23} = -a_{22}V_x$, $a_{43} = -a_{42}V_x$, $l_1 = a_{44}\dot{\psi}_{ref}$ and $l_2 = (-V_x + a_{24})\dot{\psi}_{ref} + g\sin\phi$. According to this, the lateral controller objective is to design a control signal δ that stabilizes the system (4.2), i.e. that all the error variables e_1 and e_2 converge to zero in finite time. The controller design procedure, based on [76], and the closed loop stability analysis will be stated in the following theorem.

Theorem 4. *Consider the sliding mode control law*

$$\delta = -(SB)^{-1}[\eta sign(\sigma) + SAe + SL] \tag{4.3}$$

where $S_1 = \begin{bmatrix}\lambda_1\lambda_2\lambda_3 & \lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\lambda_3 & \lambda_1 + \lambda_2 + \lambda_3\end{bmatrix}$, $S_2 \in \mathbb{R}$, $S = \begin{bmatrix}S_1 & S_2\end{bmatrix} T$, $\sigma = Se$, *and the following conditions are fulfilled*

$$\eta > 0, \quad ||\dot{\psi}_{ref}|| \leq \Psi, \quad ||\phi|| \leq \Phi, \quad \lambda_3 \geq \lambda_2 \geq \lambda_1 > 0. \tag{4.4}$$

Then, the trajectories of the system defined by (4.2) and (4.3) are globally ultimately bounded by

$$||e|| \leq \frac{(||S_1|| + ||S_2||)[(|a_{24} - V_x| + |a_{44}|)\Psi + g\sin\Phi]}{\lambda_1}. \tag{4.5}$$

Moreover, if $\Psi = \Phi = 0$*, then the origin of the closed loop system is a globally asymptotically stable equilibrium point.*

Proof. First, a similarity transformation is defined as

$$T = \begin{bmatrix}\Delta \\ \Delta A \\ \Delta A^2 \\ \Delta A^3\end{bmatrix} \rightarrow z = Te = \begin{bmatrix}z_1 \\ z_2\end{bmatrix} \tag{4.6}$$

where $\Delta = \begin{bmatrix}0\ 0\ 0\ 1\end{bmatrix}\begin{bmatrix}B\ AB\ A^2B\ A^3B\end{bmatrix}^{-1}$, which converts the system (4.2) into

$$\begin{aligned}\dot{z_1}(t) &= A_{11}z_1(t) + A_{12}z_2(t) + L_1 \\ \dot{z_2}(t) &= A_{21}z_1(t) + A_{22}z_2(t) + \delta + L_2\end{aligned}, \tag{4.7}$$

with $\begin{bmatrix}A_{11} & A_{12} \\ A_{21} & A_{22}\end{bmatrix} = TAT^{-1}$, $\begin{bmatrix}L_1 \\ L_2\end{bmatrix} = TL$, $A_{11} \in \mathbb{R}^{3x3}$, $\{z_1, A_{12}, L_1\} \in \mathbb{R}^{3x1}$, $A_{21} \in \mathbb{R}^{1x3}$, and $\{z_2, A_{22}, L_2\} \in \mathbb{R}$. This transformation permits to simplify the designing procedure of the control law. Also, by means of (4.2), (4.4), and (4.6), a bound for L_1 can be obtained of the form

$$||L_1|| \leq \left\|T\right\| [(|a_{24} - V_x| + |a_{44}|)\Psi + g\sin\Phi] \tag{4.8}$$

which will represent an important definition as it allows to establish the values of the control gains that fulfills the control objective, as seen further. Then, a sliding surface is defined via a linear combination of the original error variables and, consequently, a linear combination of the transformed coordinates of the form

$$\sigma = Se = ST^{-1}z = \begin{bmatrix} S_1 & S_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}, \tag{4.9}$$

and, using (4.2) and (4.3), its derivative yields $\dot{\sigma} = -\eta sign(\sigma)$ which ensures the sliding mode occurrence in finite time as $\dot{\sigma}\sigma = -\eta||\sigma|| \leq 0$. During the sliding motion, we have $\sigma = \dot{\sigma} = 0$ and, from (4.9), we obtain

$$z_2 = -S_1 z_1. \tag{4.10}$$

Now, by means of (4.7) and (4.10), a reduced system is obtained as

$$\dot{z_1} = Mz_1 + L_1, \tag{4.11}$$

with $M = A_{11} - A_{12}S_1$. Then, using the value of S_1 defined in the conditions of the Theorem, the eigenvalues of the reduced system results in $\{-\lambda_1, -\lambda_2, -\lambda_3\}$, and considering $\lambda_1 < 0$, $\lambda_2 < 0$ and $\lambda_3 < 0$ the reduced system corresponds to a globally exponentially asymptotically stable dynamical system. Now, let us define a candidate Lyapunov function for system (4.11) as $V(z_1) = \frac{1}{2}z_1^T z_1$; whose derivative takes the form of

$$\begin{aligned} \dot{V}(z_1) =& z_1^T \dot{z}_1 = z_1^T(Mz_1 + L_1), \\ =& z_1^T M z_1 + z_1^T L_1, \\ \leq& -\lambda_1||z_1||^2 + ||z_1||\,||L_1||, \\ \leq& -||z_1||(\lambda_1||z_1|| - ||L_1||). \end{aligned} \tag{4.12}$$

Therefore, the trajectories of system (4.11) converge to a vicinity of its origin bounded by $||z_1|| \leq \frac{||L_1||}{\lambda_1}$. From (4.6), we have that $||e|| = ||T^{-1}||\,||z|| \leq ||T^{-1}||\,(||z_1|| + ||z_2||)$ which, using (4.8), (4.10) and $||z_1||$, turns into

$$||e|| \leq \frac{(||S_1|| + 1)[(|a_{24} - V_x| + |a_{44}|)\Psi + g\sin\Phi]}{\lambda_1}.$$

Finally, it is straightforward to demonstrate that, if $\Psi = \Phi = 0$, the previous inequality reduces to $||e|| = 0$. □

The lateral control law δ defined by the previous theorem represents a reference for the actuator controller of the vehicle, as shown in the following section.

4.1.3.2 Actuator Controller

A brush-less direct current (BLDC) motor is used as the steering actuator of the vehicle. Its dynamics ([9], [8], [54],[68]) are described by the following equations

$$\begin{aligned} \dot{I}_{abc} &= \frac{V_{abc} - RI_{abc} - e_{abc}}{L - M}, \\ \dot{\omega}_m &= \frac{T_e - T_l - b\omega_m}{J}, \\ T_e &= \frac{e_{abc}^T I_{abc}}{\omega_m}, \\ \dot{\delta} &= \frac{P}{2}\omega_m, \end{aligned} \tag{4.13}$$

where V_{abc} is the 3-phase input voltage, I_{abc} is the 3-phase current vector, e_{abc} is the 3-phase back-electromagnetic force, δ is the electrical rotor angle, ω_m is the angular velocity of the rotor in rad/s, P is the number of poles, R is the armature resistance, L is the self-inductance, M is the mutual inductance of the motor, b is the viscous friction coefficient of the motor's axle, T_e is the generated electrical torque, and T_l is the load torque experienced by the motor. It is important to note that δ corresponds to the steering angle of the vehicle, which represents the input signal for the vehicle's lateral dynamics system. In addition, these equations are valid under the following assumptions:

a) The motor is not saturated and should be operated with the rated current.

b) The resistances of the three stator phase windings are equal.

c) The self-inductance and mutual inductance of the motor are constant.

d) The switching of the semiconductor devices is ideal.

The BLDC motor equations in 4.13 are expressed in the a-b-c coordinates, which means that the voltage and currents signals of the motor are defined with respect to a static 3-phase reference frame

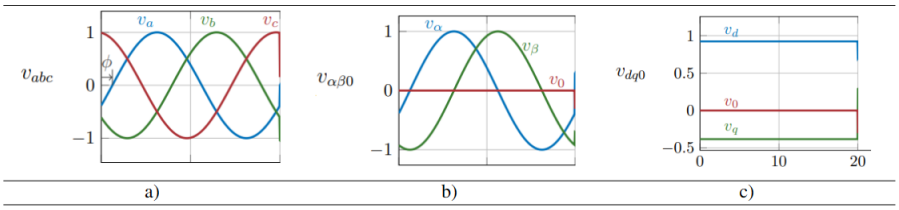


Figure 4.4: The three reference frames V_{abc},$V_{\alpha\beta0}$, V_{dq0}.

whose basic vectors are shifted in phase $\frac{2\pi}{3} rad$ with respect to each other as follows

$$\begin{aligned} v_a(t) &= V_a \cos(\omega_m t), \\ v_b(t) &= V_b \cos(\omega_m t - \tfrac{2\pi}{3}), \\ v_c(t) &= V_c \cos(\omega_m t + \tfrac{2\pi}{3}), \end{aligned} \tag{4.14}$$

However, designing a control algorithm in this reference frame is complicated. Hence, the Park-Clarke transformation, also known as the Direct-Quadrature-Zero transformation, is used to express the model of the motor in d-q-0 coordinates which simplify the control design procedure. This is because it permits to express the signals of the motor as constant values instead of sinusoidal waveforms as the reference frame is attached to the rotor of the motor. The d component of this reference represents the magnetic flux and the q component represents the torque-produced by the currents of the motor. As its name suggests, it is composed of two singular transformations: the Park and the Clarke transformations. The former result in the α-β-0 coordinates that correspond to a static frame aligned to one of the original basic vectors which reduces the significant components to only two [9]. This process is depicted in Fig. 4.4.

The Park-Clarke transform and inverse are defined as

$$PCT = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta) & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ -\sin(\theta) & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}, \tag{4.15}$$

and

$$PCT^{-1} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & \frac{1}{\sqrt{2}} \\ \cos\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta - \frac{2\pi}{3}\right) & \frac{1}{\sqrt{2}} \\ \cos\left(\theta + \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) & \frac{1}{\sqrt{2}} \end{bmatrix}. \tag{4.16}$$

The primary goal of FOC is to independently control the torque and flux in the motor, allowing

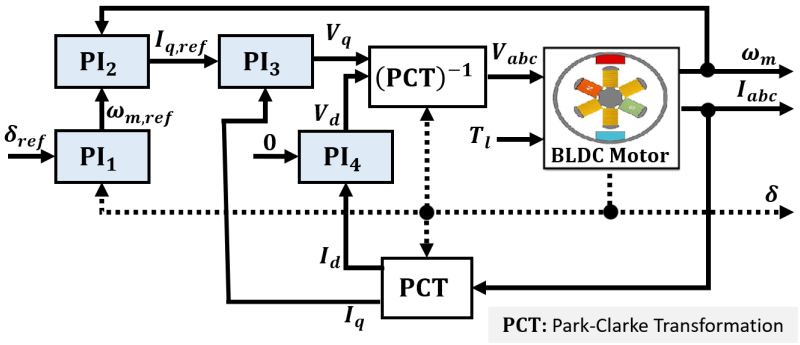


Figure 4.5: BLDC motor's angular position control scheme.

for better performance in terms of speed control, torque response, and efficiency. To this end, the control strategy consists in the following stages: first, the controller receives the reference for the angular position of the axle δ_{ref}; then, a PI controller defines a variable error between this reference and the current angular position of the motor δ and generates a reference for the angular velocity of the motor $\omega_{m,ref}$; after that, the second PI controller defines an error with the current angular velocity of the motor ω_m and generates a reference for the quadrature current of the motor $I_{q,ref}$; next, the third PI controller uses the actual quadrature current of the motor I_q to generate a desired quadrature voltage V_q to be applied to the motor and, at the same time, the fourth PI controller defines a desired direct voltage of the motor V_d in order to for the direct current I_d to remains in zero; finally, via the Park-Clarke transform, the V_{dq} voltages are converted to V_{abc} in order to be applied to the BLDC motor. It is worth to note that the Park-Clarke transform and its inverse, are used as necessary to transform the voltages and currents of the motor from the a-b-c coordinates to the d-q-0 coordinates and vice versa. This process is depicted in Fig. 4.5 which shows a classic FOC control scheme with an additional Proportional-Integral (PI) controller for the angular position of the motor. In the figure, the PCT block corresponds to the Park-Clarke transformation and the $(PCT)^{-1}$ block to its inverse.

Overall, the control strategy is composed of four PI controllers and their correspondent proportional and integral control gains. The first is used to control the angular position δ of the motor with gains $\begin{Bmatrix} K_{P,1} & K_{I,1} \end{Bmatrix}$, the second for controlling the angular velocity ω_m with gains $\begin{Bmatrix} K_{P,2} & K_{I,2} \end{Bmatrix}$, the rest to control I_q and I_d with gains $\begin{Bmatrix} K_{P,3} & K_{I,3} \end{Bmatrix}$ and $\begin{Bmatrix} K_{P,4} & K_{I,4} \end{Bmatrix}$, correspondingly.

The tuning process for all the PID controllers was based on a combination of the PID Tuning

toolbox from Matlab/Simulink and an additional fine tuning stage which was done manually. The tuning process order was as follows: first. PI_3 and PI_4 were tuned and the voltages V_q and V_d were stabilized; then, PI_2 was tuned and the angular speed was controlled; finally, PI_1 was tuned which controls the angular position of the motor. For more information regarding the field-oriented control and the $d-q$ reference frame, please refer to [9].

The simulation results and an analysis of the performance is shown in chapter 5 within the subsection with the same name of this controller.

4.2. Block Control with Super Twisting Sliding Modes for the Path Following of a Self-Driving Vehicle Considering Actuator Dynamics

This research presents the design of a robust nonlinear controller for the lateral dynamics of a self-driving car as it was published in [82]. It is based on the block control and the super-twisting sliding mode control techniques in order to effectively mitigates uncertainties and disturbances of the vehicle. The dynamic model of the system is composed of the standard bicycle dynamic model (not kinematic) for the vehicle and the dynamics of a BLDC motor connected to a steering rack system as the steering actuator. Moreover, the control scheme considers an inner loop for controlling the actuator position based on the Field Oriented Control (FOC) and the PID control approaches. The overall controller's performance is validated through its application to a mathematical model of a brushless direct current (BLDC) motor, acting as the actuator plus the steering rack dynamics and the lateral dynamic model of the vehicle. Measurements of voltages and currents are taken in both the abc and dq reference frames, the latter being commonly used in field-oriented control (FOC) technique. Additionally, the system's performance is evaluated in terms of trajectory tracking, orientation, and lateral deviation from the lane center.

4.2.1 Introduction

The automotive industry is rapidly advancing to the vehicle electrification and autonomous mobility, driven by goals such as the Paris agreement's mandate, which seeks to reduce the emissions due to the fossil fuels used by auto-motors. Autonomous mobility can offer benefits in this regard, as an example, autonomous systems can find the most efficient routes, drive with the most efficient speeds and accelerations to reduce energy consumption besides the safety improvements that non-human driving can bring. There are many approaches to tackle the complexity of an automotive self-driving system, however, most of them follow a similar generic structure for the system architecture based on layers, as shown in Figure 4.6 and 4.8. Specifically, Figure 4.6 depicts the standard control architecture for a steer-by-wire system with two control stages: the Lateral Controller designed

for the vehicle to follow a desired path, and an Actuator Controller (in this case, a BLDC motor) designed to generate the specific control inputs for the actuator. Among these layers, we can find the *route planning layer*, the *behavioral layer*, the *motion planning layer* and the *local feedback control layer* [66] [88]. The design of the latter is the objective of this work. The blue box with the label "Lateral Controller" is where our contribution is applied. It belongs to the *local feedback control layer* which is in charge of the control of the steering of the vehicle. The upper layers define a route, often given in the form of way-points that consist commonly of desired speed, and X,Y reference coordinates that the vehicle must follow. Diverse works have been developed for the described problem. For example, [60] intends to create a controller free of the physical model of the vehicle and uses conventional sliding mode control (SMC); it proposes to adapt the gains of the control at runtime. [99] implements a model predictive control that combines fuzzy logic and neural networks that adapt to different longitudinal speeds. However, its mathematical validity is still pending and under investigation. [43], [89],[97],[24], [29] and [74] use model predictive control (MPC) to solve the path-following problem considering the kinematic or dynamic lateral model of the vehicle and implementing a cost function. [83] and [92] use reinforce learning to tackle to path-following problem. [45] applies state-feedback control to an autonomous tractor with its trailer. It proposes a control algorithm and a path planning algorithm. [6] uses the kinematic model to compare several of the most common controllers for path-following such as "follow the carrot", "pure pursuit", "vector pursuit", "Stanley method", "MIT method", and "CF pursuit". [14] implements feed-forward control but using the planar two-track vehicle model. [32] implements a linear quadratic regulator (LQR) control technique, which is an optimization technique to solve the path-following problem where controller parameters are designed by minimizing a cost function. In [51], a control scheme is presented to solve the path following problem in three dimensions for a marine vehicle. Other works as [19] are based on the super-twisting algorithm (STA), which is a second-order sliding mode controller that presents several advantages over conventional SMC such as the reduction of the chattering effect, and the robustness against matched disturbances.[86] and [76]. Moreover,[78] applies two STA controllers in parallel: one for the lateral error dynamics and one for the yaw error dynamics. [37] apply the STA using a sliding surface formed by a combination of lateral speed and orientation errors. Both of them applied an additional method to find the parameters using neural networks or optimization methods. [7] used the STA in combination with an adaptive preview model

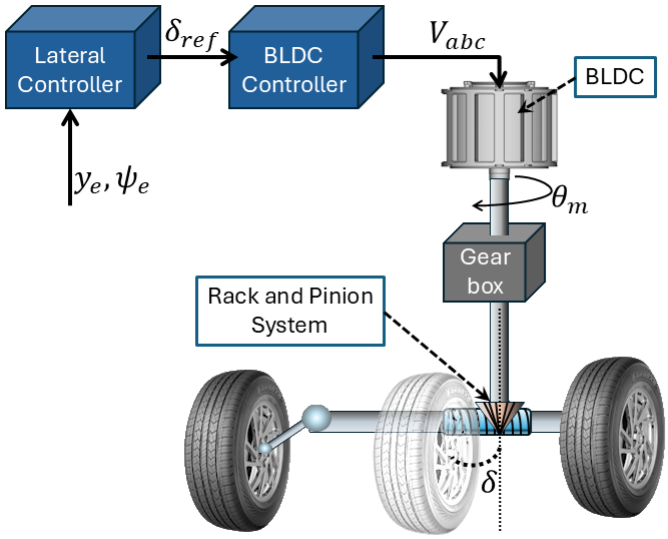


Figure 4.6: Overall control architecture for path following with different levels of autonomy.

and an additional low-pass filter to eliminate the chattering effect. However, they did not take into account the effects of actuator dynamics and used very simplified models for the vehicle. In this work, a combination of block control and integral STA is proposed for the path following problem in an electric autonomous vehicle. The combination of these control algorithms aims to increase the robustness of the global control scheme as this is a well known feature of sliding mode control techniques. Furthermore, the actuator dynamics, which is composed of a BLDC motor coupled to the steering system of the vehicle, is controlled by a PID control based on the field oriented control (FOC) scheme. This allows us to separate the actuator controller design from the design of the lateral controller of the vehicle. On the other hand, usually the parameters of electric motors are unknown or difficult to obtain, which makes the PID controller an excellent addition as its design and tuning do not require the model of the controlled system.

4.2.2 Path Following for a Self-Driving Automotive Vehicle

As explained briefly in the introduction section, self-driving vehicles implement the basic sections depicted in Fig. 4.8 where the *Route Planning Layer* is in charge of defining the route to follow from a starting point to a destination through the streets of a city. Once the route has been decided, the *Behavioral Layer* will be responsible for reacting to the external stimulus and will

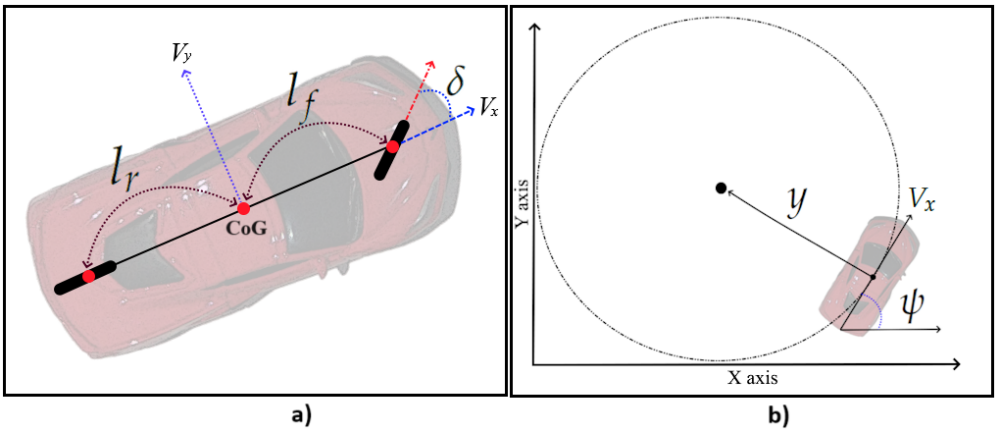


Figure 4.7: a) Bicycle Model. b) Components of the lateral dynamics.

be aware of the surroundings during the following of the established route. This layer decides the execution of primitive actions as change the lane, stop to wait for a pedestrian, reduce the vehicle speed, perform a maneuver to overtook another vehicle, etc. Then, these actions are translated by *Motion Planning Layer* which defines the physical trajectory that the vehicle must track but expressed in the vehicle's variables. This translated reference is received by the *Local Feedback Control Layer*, which generates the control input signals for the vehicle received by its actuators. The dynamical model used for the controller design corresponds to the standard bicycle model where the two frontal and rear tires are simplified as if those were only one, resembling a bicycle. This is depicted in Figure 4.7 where the variables and parameters of the vehicle's bicycle model are shown. In the figure, l_f and l_r correspond to the distances from the center of gravity (CoG) to the front and rear axles, respectively; δ is the orientation angle of the front wheel with respect to the body fixed frame; V_x and V_y represent the longitudinal and lateral velocities of the vehicle; y is the distance measured from the center of gravity (CoG) of the vehicle to the center of the circle that describes the trajectory of the vehicle; and ψ is the orientation angle of the vehicle with respect to the inertial frame defined by the X axis and Y axis.

A dynamical model for the vehicle can be obtained as [71]

$$\begin{aligned}
\ddot{y} &= \frac{2C_{\alpha f}\alpha_f + 2C_{\alpha r}\alpha_r}{m} + g\sin(\phi) - \dot{\psi}V_x, \\
\ddot{\psi} &= \frac{2l_f C_{\alpha f}\alpha_f - 2l_r C_{\alpha r}\alpha_r}{I_z}, \\
\alpha_f &= \delta - \theta_{vf}, \\
\alpha_r &= -\theta_{vr}, \\
\theta_{vf} &= \tan^{-1}\left(\frac{\dot{y} + l_f\dot{\psi}}{V_x}\right), \\
\theta_{vr} &= \tan^{-1}\left(\frac{\dot{y} - l_f\dot{\psi}}{V_x}\right),
\end{aligned} \tag{4.17}$$

and

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} V_x \\ V_y \end{bmatrix}, \tag{4.18}$$

where X, Y represent the coordinates of the CoG of the vehicle with respect to the inertial frame; α_f, α_r are the lateral slip angle for the front and rear tires; $C_{\alpha f}, C_{\alpha r}$ are the lateral stiffness coefficients for the front and rear tires, respectively; l_f, l_r are the distances measured from the CoG of the vehicle to the front and rear tire axis; m is the vehicle's mass; g is the gravity constant; ϕ is the road bank or the superelevation of the road; and I_z is the moment of inertia of the vehicle with respect to the vehicle's Z-axis. It is worth noting that this model avoids the use of the popular small-angle simplification $\tan\theta_{vf} \approx \theta_{vf}$.

4.2.3 Overall Control Scheme

Fig. 4.8 shows the basic architecture of a self-driving software stack. The route planning layer is responsible for determining the optimal path from the current location to the desired destination. This path is defined taking into account several conditions such as road types, distance, legal restrictions, traffic conditions, etc. Once the path is defined, it is passed to the next layer, the Behavioral layer, which decides how the vehicle should act according to different situations that the vehicle might encounter during the trip like the presence of a red traffic light, the presence of pedestrians, cyclists, decide if overtake or not, change a lane, etc. Based on the outcomes of this layer, the motion

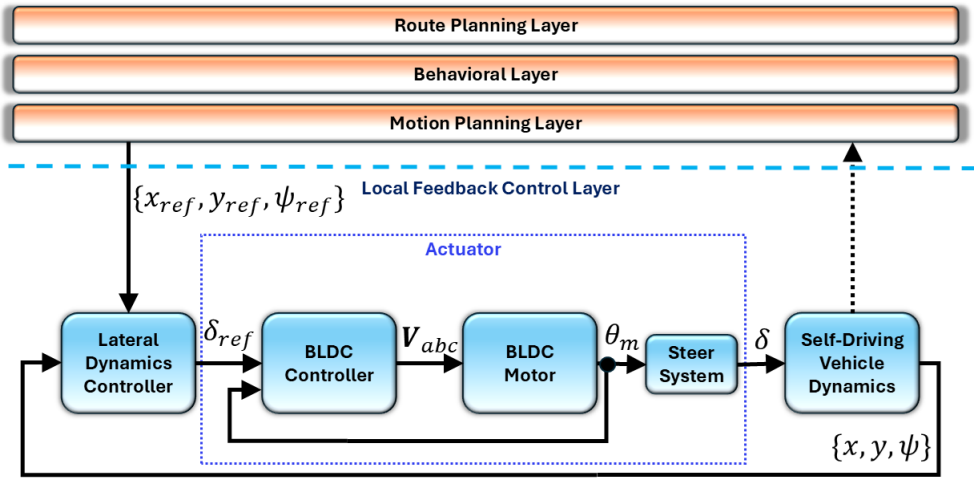


Figure 4.8: General control scheme for the local feedback control layer and overview of the software stack layer system for a self-driving vehicle.

planning layer translate these decisions to specific movements that the vehicle must perform; it involves creating a path that the vehicle must follow ensuring it is free of obstacles and ensuring that this movement will be comfortable for passengers. Now, this path is taken by the local feedback control layer, which is what this paper is about, and physically performs this predetermined trajectory, which is expressed in terms of the states of the vehicle. The task addressed by the proposed control scheme is complex considering the parameters (mass, inertia, friction, etc.) and nonlinearity of a standard automotive vehicle. The next subsection describes in detail the control methodology applied to synthesize the control law for the lateral dynamics of the vehicle.

4.2.4 Lateral sliding mode controller

This controller receives the reference for the position x_{ref}, y_{ref} and orientation ψ_{ref} of the vehicle and, along with the current state of the vehicle, calculates the lateral displacement error y_e and the orientation error ψ_e as depicted in Figure 4.9. The lateral displacement error y_e is computed based on an orthogonal line to the tangent line of the current position of the vehicle.

Then, defining auxiliary error variables as:

$$\boldsymbol{y_1(t)} = \begin{bmatrix} y_e \\ \psi_e \end{bmatrix}, \quad \boldsymbol{y_2(t)} = \begin{bmatrix} \dot{y}_e \\ \dot{\psi}_e \end{bmatrix}, \tag{4.19}$$

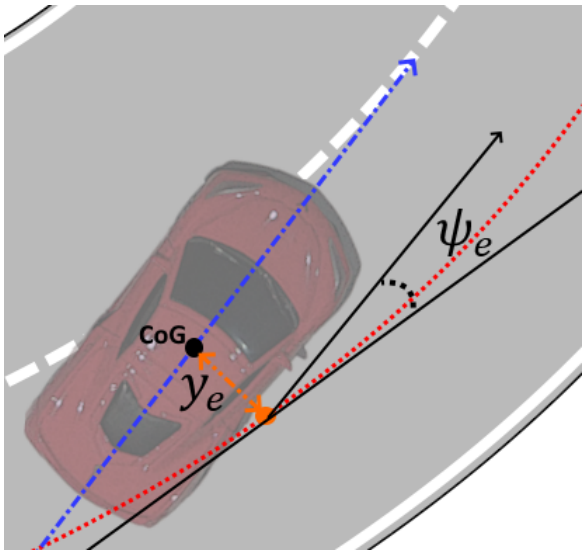


Figure 4.9: Lateral displacement error y_e and orientation error ψ_e of the vehicle with respect to the followed path (lane). The blue-dashed arrow represents the current orientation of the vehicle. The red-dotted line represents the followed path. The orange-dotted line represents the lateral error defined as the shortest distance between the CoG of the vehicle and the desired path.

we can define the dynamics of (4.19), by means of (4.17), of the form

$$\begin{aligned} \dot{\boldsymbol{y}}_{\mathbf{1}} &= \boldsymbol{y}_{\mathbf{2}}, \\ \dot{\boldsymbol{y}}_{\mathbf{2}} &= \boldsymbol{A}_{\mathbf{1}}\boldsymbol{y}_{\mathbf{1}} + \boldsymbol{A}_{\mathbf{2}}\boldsymbol{y}_{\mathbf{2}} + \boldsymbol{B}\delta + \boldsymbol{L} + \boldsymbol{\lambda}(\boldsymbol{y}_{\mathbf{1}}, \boldsymbol{y}_{\mathbf{2}}, \boldsymbol{t}), \end{aligned} \tag{4.20}$$

where

$$\boldsymbol{A}_{\mathbf{1}} = \begin{bmatrix} 0 & \frac{2C_{\alpha f}+2C_{\alpha r}}{m} \\ 0 & \frac{2C_{\alpha f}l_f-2C_{\alpha r}l_r}{I_z} \end{bmatrix}, \boldsymbol{A}_{\mathbf{2}} = \begin{bmatrix} -\frac{2C_{\alpha f}+2C_{\alpha r}}{mV_x} & \frac{-2C_{\alpha f}l_f+2C_{\alpha r}l_r}{mV_x} \\ -\frac{2C_{\alpha f}l_f-2C_{\alpha r}l_r}{I_zV_x} & -\frac{2C_{\alpha f}{l_f}^2+2C_{\alpha r}{l_r}^2}{I_zV_x} \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} \frac{2C_{\alpha f}}{m} \\ \\ \frac{2C_{\alpha f}l_f}{I_z} \end{bmatrix}, \boldsymbol{L} = \begin{bmatrix} l_1 \\ l_2 \end{bmatrix}, \tag{4.21}$$

with $l_1 = -(\frac{2C_{\alpha f}l_f-2C_{\alpha r}l_r}{mV_x} - V_x)\dot{\psi}_{des} + g\sin(\phi)$ and $l_2 = -(\frac{2C_{\alpha f}{l_f}^2+2C_{\alpha r}{l_r}^2}{I_zV_x})\dot{\psi}_{des}$. The term $\lambda(y_1, y_2, t) \in \mathbb{R}^{2\times 1}$ corresponds to the disturbances generated by uncertainties in the vehicle's parameters and sensors measurements, for example, lateral winds, terrain conditions that modifies the cornering stiffness $(C_{\alpha f})$, etc. This term is assumed to be unknown, but bounded, as

$$|\lambda(y_1, y_2, t)| < \Lambda. \tag{4.22}$$

Now, the following theorem establishes the proposed controller.

Theorem 5. *Any solution of the closed-loop system formed by the controller*

$$\delta = \delta_0 + \delta_1, \tag{4.23}$$

where

$$\begin{aligned} e &= -k_1 y1 - y_2, & \dot{v}_0 &= -k_{v0}\text{sign}(e), \\ \delta_0 &= -B^+(A_1 y_1 + (k_1 I_2 + A_2)y_2 + L - k_{u0}|e|^{1/2}\text{sign}(e) + v_0), \end{aligned} \tag{4.24}$$

$$\begin{aligned} \sigma &= e + z, & \dot{v}_1 &= -k_{v1}\text{sign}(\sigma), \\ \delta_1 &= -B^+(-k_{u1}|\sigma|^{1/2}\text{sign}(\sigma) + v_1), \\ \dot{z} &= A_1 y_1 + (k_1 I_2 + A_2)y_2 + L + B\delta_0, \end{aligned} \tag{4.25}$$

and the system (4.20) is ultimately bounded if conditions (4.22), $z(0) = -e(0)$ *and*

$$k_1 > 0 \tag{4.26}$$

$$k_{u0} > 0, \quad k_{v0} > 0 \tag{4.27}$$

$$k_{u1} > 2\Lambda, \quad k_{v1} > k_{u1}\frac{5\Lambda k_{u1} + 4\Lambda^2}{2(k_{u1} - 2\Lambda)} \tag{4.28}$$

are satisfied.

Proof. Let us define a sliding surface σ as in (4.25), then, its dynamics can be obtained using (4.20), (4.24) and (4.25) as

$$\begin{aligned} \dot{\sigma} &= BB^+(-k_{u1}|\sigma|^{1/2}\text{sign}(\sigma) + v_1) - \lambda(t, y_1, y_2) \\ \dot{v}_1 &= -k_{v1}\text{sign}(\sigma). \end{aligned} \tag{4.29}$$

The objective is to demonstrate that, in closed loop, the sliding surface converges to zero, and, consequently, the disturbance term $\lambda(t, y_1, y_2)$ is rejected. Now, propose a Candidate Lyapunov Function $V_1 = \Gamma_1 P_1 \Gamma_1^T$ with

$$\Gamma_1 = \begin{bmatrix} |\sigma|^{1/2}\text{sign}(\sigma) & v_1 \end{bmatrix}, \quad P_1 = \tfrac{1}{2}\begin{bmatrix} 4k_{u1} + k_{v1}^2 & -k_{v1} \\ -k_{v1} & 2 \end{bmatrix}; \tag{4.30}$$

its derivative results of the form $\dot{V}_1 = |\sigma|^{-1/2}(-\Gamma_1 Q_1 \Gamma_1^T + \lambda(t, y_1, y_2) q \Gamma_1^T)$ with

$$Q_1 = \tfrac{k_{u1}}{2}\begin{bmatrix} 2k_{u1} + k_{v1}^2 & -k_{v1} \\ -k_{v1} & 1 \end{bmatrix}, \quad q = \begin{bmatrix} 2k_{u1} + \frac{k_{v1}^2}{2} & -\frac{k_{v1}}{2} \end{bmatrix} \tag{4.31}$$

as $|BB^{+}| = 1$. Then, by means of (4.22), $\dot{V}_1$ yields

$$\dot{V}_1 \leq -\frac{k_{u1}}{2|\sigma|^{1/2}}\Gamma_1 \begin{bmatrix} 2k_{v1} + k_{u1}^2 - (\frac{4k_{v1}}{k_{u1}} + k_{u1}) & -k_{u1} - 2\Lambda \\ -k_{u1} - 2\Lambda & 1 \end{bmatrix} \Gamma_1^T \tag{4.32}$$

which is negative definite under conditions (4.28). Hence, σ and $\dot{\sigma}$ converge to 0 in finite time [55] which generates the equality $B\delta_1 = \lambda(t, y_1, y_2)$. It is worth to note that, as $z(0) = -e(0) \rightarrow \sigma(0) = 0$, and the sliding motion occurs from the initial instance.

After this, and using (4.24), the dynamics of the error variable e is reduced to

$$\begin{aligned} \dot{e} &= BB^{+}(-k_{u0}|e|^{1/2}\text{sign}(e) + v_0) \\ \dot{v}_0 &= -k_{v0}\text{sign}(e). \end{aligned} \tag{4.33}$$

Again, proposing the Candidate Lyapunov Function $V_0 = \Gamma_0 P_0 \Gamma_0^T$ with

$$\Gamma_0 = \begin{bmatrix} |e|^{1/2}\text{sign}(e) & v_0 \end{bmatrix}, \quad P_0 = \tfrac{1}{2}\begin{bmatrix} 4k_{u0} + k_{v0}^2 & -k_{v0} \\ -k_{v0} & 2 \end{bmatrix}; \tag{4.34}$$

its derivative results of the form $\dot{V}_0 = |e|^{-1/2}(-\Gamma_0 Q_0 \Gamma_0^T)$ with

$$Q_0 = \frac{k_{u0}}{2}\begin{bmatrix} 2k_{u0} + k_{v0}^2 & -k_{v0} \\ -k_{v0} & 1 \end{bmatrix}, \tag{4.35}$$

as $|BB^{+}| = 1$. If conditions (4.27) are satisfied, then, $e \rightarrow 0$ and $\dot{e} \rightarrow 0$ in finite time, and the dynamics of y_1 yields

$$\dot{y}_1 = -k_1 y_1. \tag{4.36}$$

Finally, using (4.26), the origin of the system y_1 is globally asymptotically stable. Therefore, the lateral displacement error y_e and the orientation error converge to 0, i.e., the objective of the lateral controller is fulfilled. □

4.2.5 Actuator Controller

Commonly, the actuator in steer-by-wire systems is a BLDC motor coupled to a steering rack system. The input of the BLDC motor is a three-phase voltage V_{abc} which generates a desired angular position δ_{ref} of the steering system as depicted in Fig. 4.8.

4.2.5.1 BLDC Motor Dynamics

The dynamic model of the actuator can be represented as [54],[68]:

$$\frac{d}{dt}\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} \frac{1}{L_s - M_s} & 0 & 0 \\ 0 & \frac{1}{L_s - M_s} & 0 \\ 0 & 0 & \frac{1}{L_s - M_s} \end{bmatrix} \left[\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} - \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} - \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \right], \tag{4.37}$$

where R_s is the stator resistance, assumed to be the same for the three phases a, b, c. L_s is the inductance in phases a, b, c. M_s is the mutual inductance between phases ab, ca, cb and we assume that all have the same value. $[e_a \quad e_b \quad e_c]^T = e_{abc}$ represents the back electromotive force (EMF). $[i_a \quad i_b \quad i_c]^T = i_{abc}$ are the currents of the three phases in the stator. $[V_a \quad V_b \quad V_c]^T = V_{abc}$ are the voltages in the three phases of the stator; below, we define e_{abc}

$$\begin{aligned} e_a &= \lambda_e \omega_m f_a \text{ where } f_a = f(\theta_e), \\ e_b &= \lambda_e \omega_m f_b \text{ where } f_b = f(\theta_e - \tfrac{2\pi}{3}), \\ e_c &= \lambda_e \omega_m f_c \text{ where } f_c = f(\theta_e + \tfrac{2\pi}{3}). \end{aligned} \tag{4.38}$$

For the torque and mechanical equations, we have:

$$T_e = (e_a i_a + e_b i_b + e_c i_c)/\omega_m = (\lambda_e f_a i_a + \lambda_e f_b i_b + \lambda_e f_c i_c), \tag{4.39}$$

where λ_e is the Back EMF constant in v/rad/s. θ_e is the electrical rotor angle. ω_m is the mechanical angular velocity of the rotor shaft in rad/s and, it is related to the electrical speed as $\omega_m = \frac{2}{P}\omega_e$. f_a, f_b, f_c are functions that have the same shape as e_a, e_b, e_c with a maximum magnitude of ±1; the three functions are the same but displaced 120º. The equation of motion, solved for T_l, that is, the torque load which is applied to the gearbox of the steering system is

$$T_l = T_e - b\omega_m - J\dot{\omega}_m. \tag{4.40}$$

The electrical angle speed, mechanical speed, and position are related as follows:

$$\omega_e = \dot{\theta}_e = \frac{P}{2}\omega_m, \tag{4.41}$$

$$\theta_m = \frac{2}{P}\theta_e, \tag{4.42}$$

where P is the number of poles, in our case it is 8.

4.2.5.2 Steering Rack System Model

The term T_l in (4.40) represents the torque transmitted to the steering rack system, whose dynamical model can be defined as [94]

$$J_s\ddot{\delta} + b_s\dot{\delta} = N_m T_l + \tau_a + \tau_f, \tag{4.43}$$

$$\delta = \frac{\theta_m}{N_m}, \tag{4.44}$$

where $\tau_a = -N_l F_z V_x \sin\delta$ is the self-aligning torque, $\tau_f = -N_l F_z \tanh\frac{\dot{\delta}}{\varepsilon}$ is the friction force of the tire over the ground, $F_z = \frac{mg}{2}$ is the normal force acting on the tire assuming the bicycle model, $\epsilon > 0$ is an approximation of the slope for the friction torque, N_m and N_l are transmission ratios, J_s and b_s are the equivalent moment of inertia and damping coefficient of the steering system, respectively.

4.2.5.3 Position Control Algorithm

To facilitate the controller design procedure of a BLDC motor, a transformation is applied from abc to dq coordinates of the currents and voltages of the motor. This is achieved by means of the Park-Clarke transform [8, 9] and its inverse, which are defined as

$$\begin{aligned} PCT &= \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta) & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ -\sin(\theta) & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}, \\ PCT^{-1} &= \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & \frac{1}{\sqrt{2}} \\ \cos\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta - \frac{2\pi}{3}\right) & \frac{1}{\sqrt{2}} \\ \cos\left(\theta + \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) & \frac{1}{\sqrt{2}} \end{bmatrix}. \end{aligned} \tag{4.45}$$

The actuator control strategy is depicted in Figure 4.10 which corresponds to the Field Oriented Control (FOC) approach based on PI (Proportional-Integral) controllers expressed in dq coordinates.

Hence, the general control objective is to force θ_e to follow $\theta_{e,ref}$ by means of V_{abc}. From (4.43) and (4.41), we obtain $\theta_{e,ref} = \delta_{ref} N_m \frac{P}{2}$, and the error variables for the PI controllers can be defined

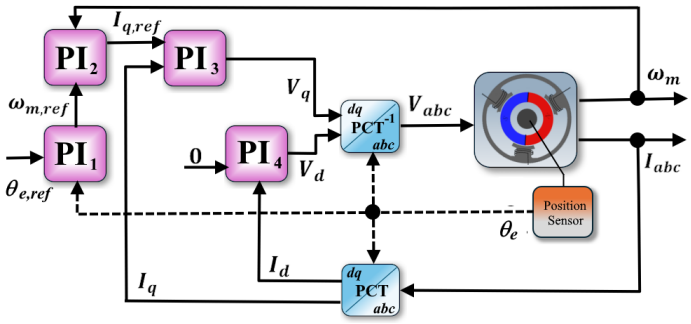


Figure 4.10: Control scheme of a BLDC motor.

as

$$e_{PI1} = \theta_{e,ref} - \theta_e, \quad e_{PI2} = \omega_{m,ref} - \omega_m, \quad e_{PI3} = I_{q,ref} - I_q, \quad e_{PI4} = -I_d. \tag{4.46}$$

Then, the execution of the actuator controller can be described as follows: first, the PI_1 controller computes e_{PI1} and generates $\omega_{m,ref}$; then, the PI_2 controller uses $\omega_{m,ref}$ to compute e_{PI2} and generates $I_{q,ref}$; after that, the PI_3 controller uses $I_{q,ref}$ to compute e_{PI3} and generates the V_q input voltage; finally, the PI_4 controller generates the V_d input voltage by means of the I_d current as its desired value is 0. The resulting V_{dq} voltages are transformed to V_{abc} using the Park-Clarke transform. The control gains for the four PI controllers were, firstly, obtained using the automatic tuner tool of Simulink and, then, applying a manual fine-tuning to improve the transient of the closed-loop system response in terms of settling time and overshoot. This is important as the internal closed-loop system defined by the actuators controller must have a lower constant time than the outer control loop defined by the vehicle's controller. This impacts also the required computational capacities of the ECU (Electronic Control Unit) where the embedded implementation of the control scheme will be deployed.

The simulation results and an analysis of the performance is shown in chapter 5 within the subsection with the same name of this controller.

4.3. Linear Block Control Plus Super Twisting Sliding Modes for Disturbance Rejection for the Path Following of a Self-Driving Vehicle Considering Actuator Dynamics (Variant)

4.3.1 Introduction

During the development of the control algorithm published in [82] and described in this dissertation in section 4.2, several variants were proposed and this is one of those and depicts how the concepts evolved. In [82], the super-twisting and integral sliding mode control (SMC) algorithms were introduced following Utkin's original formulation. In contrast, this section adopts the formulations proposed by Shtessel et al. [76]. One notable implication of this difference lies in the convergence proof of the STA algorithm: while [82] employed the Moreno–Osorio method [55], the Shtessel et al. formulation requires only the satisfaction of a direct inequality. While some of this section has similarities to the section 4.2, the mathematical expressions for the lateral dynamic are slightly different, and the resultant control law completely different. For more information about this control, see chapters 4.2.1, 4.2.2, 4.2.3, that are the *Introduction*, *Path Following for an Automotive Vehicle*, and *Overall Control Scheme* respectively.

4.3.2 Lateral sliding mode controller

For the design of the control, we take a modified dynamic bicycle model that give us the error respect to the road; that is, giving a reference path we obtain the distance of the displacement against our reference path, and the difference in the orientation between the vehicle and the road. In figure 4.9 is shown this idea. y_e is the distance from the CoG to the center of the lane; the red line (dotted) represents the reference path. ψ_e is the orientation error: that is, the angle between the black arrow (solid) which is parallel to the x body fixed axis of the vehicle and the tangent on the reference path in the point where the CoG should be.

Below, the model of the error expressed in two blocks:

$$\begin{aligned} \boldsymbol{y_1} &= \begin{bmatrix} y_e \\ \psi_e \end{bmatrix} \\ \boldsymbol{y_2} &= \begin{bmatrix} \dot{y}_e \\ \dot{\psi}_e \end{bmatrix} = \dot{\boldsymbol{y}}_\mathbf{1} \end{aligned} \tag{4.47}$$

where

$$\begin{aligned} \dot{\boldsymbol{y}}_\mathbf{1} &= \boldsymbol{y_2} \\ \dot{\boldsymbol{y}}_\mathbf{2} &= \boldsymbol{A e} + \boldsymbol{B}\delta + \boldsymbol{L} + \boldsymbol{\lambda} \end{aligned} \tag{4.48}$$

with

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}, \boldsymbol{e} = \begin{bmatrix} \dot{y}_e \\ \psi_e \\ \dot{\psi}_e \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \boldsymbol{L} = \begin{bmatrix} l_1 \\ l_2 \end{bmatrix}, \boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix},$$

and

$$\begin{aligned} a_{11} &= -\tfrac{2C_{\alpha f}+2C_{\alpha r}}{mV_x}, & a_{12} &= \tfrac{2C_{\alpha f}+2C_{\alpha r}}{m}, & a_{13} &= \tfrac{-2C_{\alpha f}l_f+2C_{\alpha r}l_r}{mV_x}, \\ a_{21} &= -\tfrac{2C_{\alpha f}l_f-2C_{\alpha r}l_r}{I_zV_x}, & a_{22} &= \tfrac{2C_{\alpha f}l_f-2C_{\alpha r}l_r}{I_z}, & a_{23} &= -\tfrac{2C_{\alpha f}{l_f}^2+2C_{\alpha r}{l_r}^2}{I_zV_x}, \end{aligned}$$

$$l_1 = -(\tfrac{2C_{\alpha f}l_f-2C_{\alpha r}l_r}{mV_x} - V_x)\dot{\psi}_{des} + g\sin(\phi), \quad l_2 = -(\tfrac{2C_{\alpha f}{l_f}^2+2C_{\alpha r}{l_r}^2}{I_zV_x})\dot{\psi}_{des},$$

$$b_1 = \tfrac{2C_{\alpha f}}{m}, \quad b_2 = \tfrac{2C_{\alpha f}l_f}{I_z}.$$

Additionally, to the model presented in (4.17), we added λ_1, and λ_2 that represent bounded unknown disturbances and non-linearities, assuming these are smooth and continuous. However, we assume we know the boundaries $|\lambda_1|$, and $|\lambda_2|$ representing the maximum disturbances in the y and ψ dynamics, respectively.

Theorem 6. *Consider the following sum of two controllers, one, δ_k intended to control the linear part which is the "known" part and the other, δ_u intended to compensate the "unknown" behavior due to non-linear disturbances:*

$$\delta_{ref} = \delta_k + \delta_u, \tag{4.49}$$

where

$$\delta_k = -(\boldsymbol{\gamma B})^{-1}\left[\boldsymbol{\gamma}(k_1\boldsymbol{y_2} + \boldsymbol{Ae} + \boldsymbol{L}) - k_2 e_b\right] \tag{4.50}$$

and

$$\delta_u = -k_u\sqrt{|S|}\textit{sign}(S) + v,\ \textit{with}\ \boldsymbol{\gamma} = \begin{bmatrix}\gamma_1 & \gamma_2\end{bmatrix}, \tag{4.51}$$

where

$$\dot{v} = \begin{cases} -\delta_u, & |\delta_u| > U_M \\ -k_v\textit{sign}(S), & |\delta_u| \leq U_M \end{cases} \tag{4.52}$$

and

$$S = e_{bn} - k_2 e_b(0) - k_2 e_b, \tag{4.53}$$

with

$$e_b \triangleq -\boldsymbol{\gamma}\left(k_1\boldsymbol{y_1} + \boldsymbol{y_2}\right), \tag{4.54}$$

$$e_{bn} \triangleq -\gamma\int_0^t (k_1\boldsymbol{y_2}(\tau) + \boldsymbol{Ae}(\tau) + \boldsymbol{L} + \boldsymbol{B}\delta_k)d(\tau), \tag{4.55}$$

*where γ_1 and γ_2 are proportionality constants for y_e and ψ_e respectively, e_{bn} is the nominal response of the system to the control δ_k and besides the conditions **C1** and **C2** are met, then the origin of the closed loop system in (4.47) is globally asymptotically stable.*

Proof. Having the two blocks in (4.48), $\dot{\boldsymbol{y}}_{\mathbf{1}}$ and $\dot{\boldsymbol{y}}_{\mathbf{2}}$, we want to control the first block $\dot{\boldsymbol{y}}_{\mathbf{1}}$ using $\boldsymbol{y_2}$ through the second block $\dot{\boldsymbol{y}}_{\mathbf{2}}$ and we will control this second block using δ. As we want in the first block to enforce a dynamic that causes to asymptotically converge to zero the errors $\boldsymbol{y_1} = \begin{bmatrix}y_e & \psi_e\end{bmatrix}^T$, we define our reference y_{2ref} this way:

$$\dot{\boldsymbol{y}}_{\mathbf{1}} = \boldsymbol{y_{2ref}} \triangleq -k_1\boldsymbol{y_1}, \tag{4.56}$$

and we define the error for the first block as

$$e_b = \boldsymbol{\gamma}(\boldsymbol{y_{2ref}} - \boldsymbol{y_2}),\ \text{with}\ \boldsymbol{\gamma} = \begin{bmatrix}\gamma_1 & \gamma_2\end{bmatrix}, \tag{4.57}$$

where γ_1 and γ_2 are proportionality constants that will be tuned by hand and are used to normalize the yaw angle error versus the y distance error, and then combining (4.56) and (4.57)

$$e_b = -\boldsymbol{\gamma}\left(k_1\boldsymbol{y_1} + \boldsymbol{y_2}\right); \tag{4.58}$$

we obtain the derivative of the error in the block, and we have

$$\dot{e}_b = -\boldsymbol{\gamma}\left(k_1\dot{\boldsymbol{y}}_{\mathbf{1}} + \dot{\boldsymbol{y}}_{\mathbf{2}}\right) = -\boldsymbol{\gamma}\left(k_1\boldsymbol{y}_{\mathbf{2}} + \dot{\boldsymbol{y}}_{\mathbf{2}}\right). \tag{4.59}$$

As we want the error e_b to vanish asymptotically, we force the derivative of the error e_b as

$$\dot{e}_b = -\boldsymbol{\gamma}\left(k_1\boldsymbol{y}_{\mathbf{2}} + \dot{\boldsymbol{y}}_{\mathbf{2}}\right) = -k_2 e_b, \tag{4.60}$$

and we expand (4.60) to obtain:

$$\dot{e}_b = -\boldsymbol{\gamma}\left(k_1\boldsymbol{y}_{\mathbf{2}} + \boldsymbol{A}\boldsymbol{e} + \boldsymbol{B}\delta + \boldsymbol{L} + \boldsymbol{\lambda}\right) = -k_2 e_b. \tag{4.61}$$

Now, we want to divide our control δ in two parts, one to reject the error for which dynamics we know δ_k and a second part to reject the error for which dynamics we do not know δ_u, in this way

$$\delta = \delta_k + \delta_u. \tag{4.62}$$

Then

$$\dot{e}_b = -\boldsymbol{\gamma}\left(k_1\boldsymbol{y}_{\mathbf{2}} + \boldsymbol{A}\boldsymbol{e} + \boldsymbol{B}\delta_k + \boldsymbol{B}\delta_u + \boldsymbol{L} + \boldsymbol{\lambda}\right) = -k_2 e_b, \tag{4.63}$$

we solve for δ_k and δ_u then

$$\delta_k + \delta_u = -(\boldsymbol{\gamma}\boldsymbol{B})^{-1}\boldsymbol{\gamma}\left(k_1\boldsymbol{y}_{\mathbf{2}} + \boldsymbol{A}\boldsymbol{e} + \boldsymbol{L} + \boldsymbol{\lambda}\right) + (\boldsymbol{\gamma}\boldsymbol{B})^{-1}k_2 e_b, \tag{4.64}$$

defining δ_k using all the known terms

$$\delta_k = -(\boldsymbol{\gamma}\boldsymbol{B})^{-1}\left[\boldsymbol{\gamma}(k_1\boldsymbol{y}_{\mathbf{2}} + \boldsymbol{A}\boldsymbol{e} + \boldsymbol{L}) - k_2 e_b\right], \tag{4.65}$$

leaving δ_u for the unknown terms that represents the disturbances and non-linearities, and that we want to vanish, so we want to force

$$\boldsymbol{\gamma}\boldsymbol{\lambda} + (\boldsymbol{\gamma}\boldsymbol{B})\delta_u = 0. \tag{4.66}$$

Now, as we want to conserve the linear dynamics of the control d_k, we will obtain the disturbances over the dynamic we want to force. For that, we will create our sliding surface according the integral sliding mode (ISM) technique [87][50] as follows:

We take (4.61) and (4.65). The left side of (4.61) will have the nominal response if we applied the control (4.65). Then, subtracting the nominal response from the actual states we want to force, we have

$$S = -k_2 e_b - k_2 e_b(0) - (-k_2 e_{bn}), \tag{4.67}$$

with e_{bn} as the nominal response of the system and $k_2 e_b(0)$ as the initial condition. We define

$$e_{bn} \triangleq -\gamma \int_0^t (k_1 \boldsymbol{y_2}(\tau) + \boldsymbol{A}\boldsymbol{e}(\tau) + \boldsymbol{L} + \boldsymbol{B}\delta_k) d(\tau), \tag{4.68}$$

which are the remaining terms in (4.64) once we removed the terms that are compensated by δ_k. Now, we apply STSMC theorem as described in [76] to (4.66) which guarantees the convergence to zero in finite time of the unknown terms giving as a result the equation (4.51), assuming $h(x,t) = \gamma\lambda$ and $g(x,t) = \gamma\boldsymbol{B}$ (the STSMC theorem in [76] is defined with $h(x,t)$ and $g(x,t)$).

Now, we proceed with the stability analysis. Assuming the next conditions are met:

C1 $k_1, k_2 > 0$.

C2

$$k_v > (\boldsymbol{\gamma B})^{-1} C \tag{4.69}$$

and

$$k_u > \sqrt{\frac{2}{(\boldsymbol{\gamma B})k_v - C}} \frac{(\boldsymbol{\gamma B} + C)(1+q)}{(\boldsymbol{\gamma B})(1-q)} \tag{4.70}$$

where

$$\begin{aligned} C &\geq |\boldsymbol{\gamma \dot{\lambda}}| \\ qU_M &> |\tfrac{\boldsymbol{\gamma \lambda}}{\boldsymbol{\gamma B}}|, \quad 0 < q < 1. \end{aligned} \tag{4.71}$$

Assuming the system (4.47) applying only the linear control δ_k and no disturbances $\lambda = 0$, the system is reduced to

$$\dot{y}_2 = -k_1 k_2 y_1 - (k_1 + K_2) y_2, \tag{4.72}$$

which has, for the homogeneous differential equation, the characteristic equation as

$$\eta^2 + (k_1 + K_2)\eta + k_1 k_2 = 0, \tag{4.73}$$

which indicates with two negative poles that in the absence of disturbances the system is asymptotically stable. Now, assuming disturbances $\lambda \neq 0$, we use the next Lyapunov energy candidate equation

$$V = \frac{1}{2} y_1^2 + \frac{1}{2} y_2^2, \tag{4.74}$$

then

$$\dot{V} = (1 - k_1 k_2) y_1 y_2 - (k_1 + k_2) y_2^2 + y_2 \lambda, \tag{4.75}$$

we assume $k_1 k_2 = 1$ to simplify, and if we assume that the disturbance is bounded to $|\lambda|$ we have that

$$\dot{V} \leq -(k_1 + k_2) y_2^2 + |y_2||\lambda|, \tag{4.76}$$

using the Young's inequality as follows

$$|y_2||\lambda| \leq \frac{(k_1 + k_2)}{2}|y_2|^2 + \frac{2}{(k_1 + k_2)}|\lambda|^2, \tag{4.77}$$

we have that

$$\dot{V} \leq -\frac{(k_1 + k_2)}{2}|y_2|^2 + \frac{2}{(k_1 + k_2)}|\lambda|^2, \tag{4.78}$$

then we can conclude that

$$\dot{V} \leq 0 \text{ if } y_2 > \frac{2}{(k_1 + k_2)}|\lambda|,$$

and hence the system do not diverge and stay in a region bounded by $\frac{2|\lambda|}{(k_1+k_2)}$.

We continue with the addition of the control δ_u that uses the STSMC algorithm applied to the surface (4.53) which represents the remaining error after applying δ_k to the system according the Integral SMC technique. We have that the derivative of this surface is

$$\dot{S} = \gamma B \delta_u + \gamma \lambda, \tag{4.79}$$

and applying the STSMC theorem meeting the condition **C2** then the two sliding mode $S = \dot{S} = 0$ will attract the trajectories in finite time, eliminating the disturbances. And the demonstration is complete. □

It is worth noticing that the super-twisting sliding mode control (STSMC) as defined by Arie Levant et al. in [76], describes some differences from the one described by Utkin et al. in [86]. We chose the first because of the direct application and the conditions to be met are defined as direct formulas.

Also, in practice, it is well known that an integral term in the control might cause a phenomenon called "windup" as explained in [61] and [44] and hence, here we add a simple anti-windup algorithm

to the integral term e_{bn} in (4.68) for the simulation and it is defined as

$$y = \zeta_w \tanh\left(\frac{u}{\epsilon_w}\right), \tag{4.80}$$

where ζ_w is the maximum value of the integral term u and ϵ_w is a parameter to define the slop of the tanh function and besides this function, the summation in the integral must be saturated to $\pm\zeta_w$.

4.3.3 Actuator and Its Controller

Commonly, for steer by wire systems or for electronic power steering (EPS), electrical machines are used as actuators to control the steering. In this paper, we chose the model of a brush-less direct current motor (BLDC), that is a type of trapezoidal surface permanent magnet (SPM) machine. The requirements for this motor are $12V$ input voltage per phase, maximum current of $90A$ with maximum peak of up to $200A$, torque of $2.3Nm$ at $250rad/s$. The dynamic model of the BLDC and rack system are the same than the formulations in section 4.3.3, see for more details. The simulation results and an analysis of the performance is shown in chapter 5 within the subsection with the same name of this controller.

5. Performance Analysis of the Proposed Controllers

In this section, the proposed controllers in section 4 are evaluated within a simulation in Matlab-Simulink that contains the vehicle dynamics and the dynamics of the actuators. Besides, an analysis of the performance is provided.

5.1. Path Following Sliding Mode Controller for an Electric Vehicle Considering Actuator Dynamics

5.1.1 Simulation Results

The parameters used for the simulation results presented in this section are defined in Table 5.1 and the gains of the overall control law are shown in Table 5.2. It is worth noting that the control gains of the lateral sliding mode controller were tuned according to the necessary stability conditions established by Theorem 1, and the gains of the actuator's controller were tuned heuristically. Furthermore, if the reader wishes to reproduce these results, the Simulink model can be downloaded from [20] or [79], and the configuration of the Simulink model is described in detail at Appendix 5.1.2.

The path following behavior of the proposed control scheme in the $x - y$ coordinates plane is shown in Fig. 5.1 for the whole experiment where the dashed blue line corresponds to the current position of the vehicle and the solid red line represents the reference trajectory. It can be noted that the performance of the closed loop system is satisfactory as the control objective is achieved even considering a constant longitudinal velocity of 22 m/s (79.2 km/h) and a constant road bank angle of 0.174533 rad (10 °). The experiments were carried out with different values of longitudinal velocity and road bank angle, however, only the results obtained with the extreme values were reported as we assume that these demonstrate better the capabilities of the proposed control scheme.

TABLE 5.1. PARAMETERS OF THE VEHICLE AND THE ACTUATOR

Symbol	Value	Symbol	Value
l_r	1.58 m	R	0.35 Ω
l_f	1.1 m	L	0.001 H
I_z	2873 $N \cdot m$	M	0 H
m	2238.932 Kg	J	0.00018 $kg \cdot m^2$
$C_{\alpha r}$	80000	P	4
$C_{\alpha f}$	80000	b	0.0034 $N \cdot m \cdot s \cdot rad^{-1}$
ϕ	0.174533 rad	T_l	3.0 $N \cdot m$
λ	0.0433 Wb	V_x	22 m/s

This can also be appreciated in Fig. 5.2 where a close-up of the previous figure is presented, and the small difference between the reference path and the position of the vehicle is more visible.

This path following error is depicted as well in the top of Fig. 5.3 where its maximum value during the whole experiment is 0.21 m and its average value corresponds to 0.033 m. Moreover, the heading angle error variable e_2 is shown in the middle of the figure where its maximum value is 0.09 rad and its average value is 0.019 rad. Also, the bottom of the figure depicts the steering angle of the vehicle which maximum value is 0.25 rad (14.32°).

It is important to emphasize that the bound of the resulting steering angle corresponds to the bound of the steering angle of a real automotive vehicle. This would be crucial if a real-time implementation of the proposed controller is considered.

Fig. 5.4 shows the states of the lateral dynamical model which corresponds to the distance to path y, the heading angle of the vehicle ψ, and their corresponding derivatives. In addition, Fig. 5.5 depicts the actuator's mechanical and electrical states.

The performance of the actuator's controller can be assessed by means of Fig. 5.6 where the steering angle δ and its reference are shown. Also, this performance can be interpreted from Fig. 5.7 where the error variables for the four internal PID controllers of the actuator are shown. It can

TABLE 5.2. CONTROL GAINS FOR LATERAL AND ACTUATOR CONTROLLERS

Symbol	**Value**	**Symbol**	**Value**
λ_1	-1	$K_{P,2}$	-0.39820378606854
λ_2	-2	$K_{I,2}$	-44.4991237925307
λ_3	-3	$K_{P,3}$	70
η	16	$K_{I,3}$	0.3
S_2	$1x10^6$	$K_{P,4}$	70
$K_{P,1}$	21.2472922845302	$K_{P,4}$	0.3
$K_{I,1}$	1497.59558537531		

be appreciated that the low frequency steady state of the four error variables converge to a vicinity of zero, thus ensuring the fulfillment of the control objective.

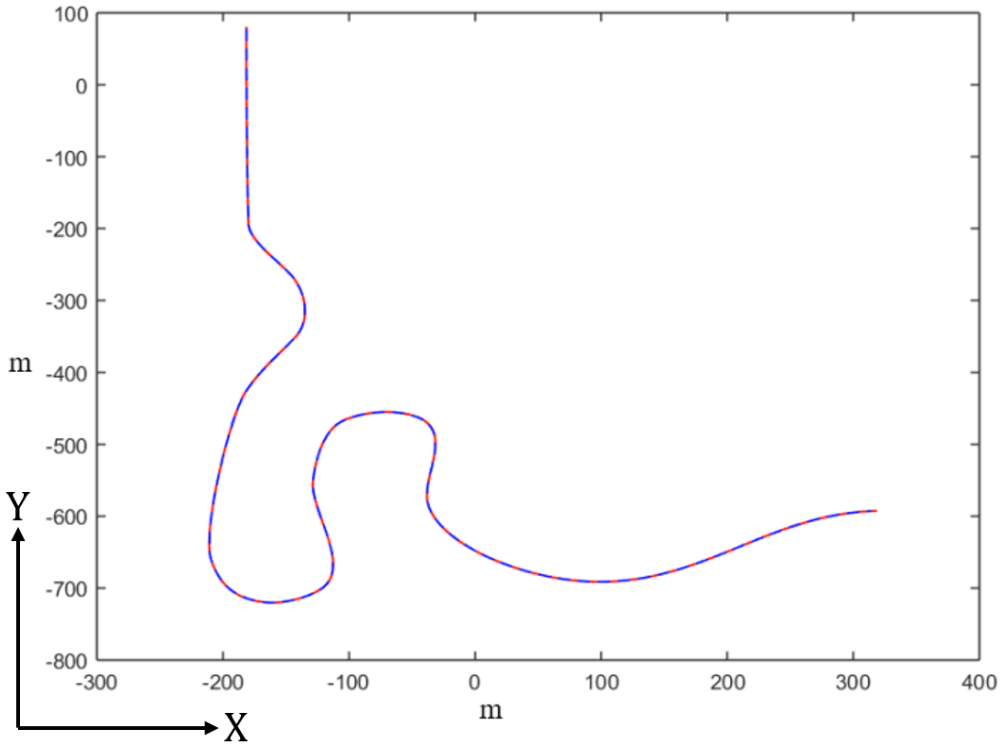


Figure 5.1: Results of the path following experiment where the current path (dashed blue line) and the reference trajectory (solid red line) are depicted.

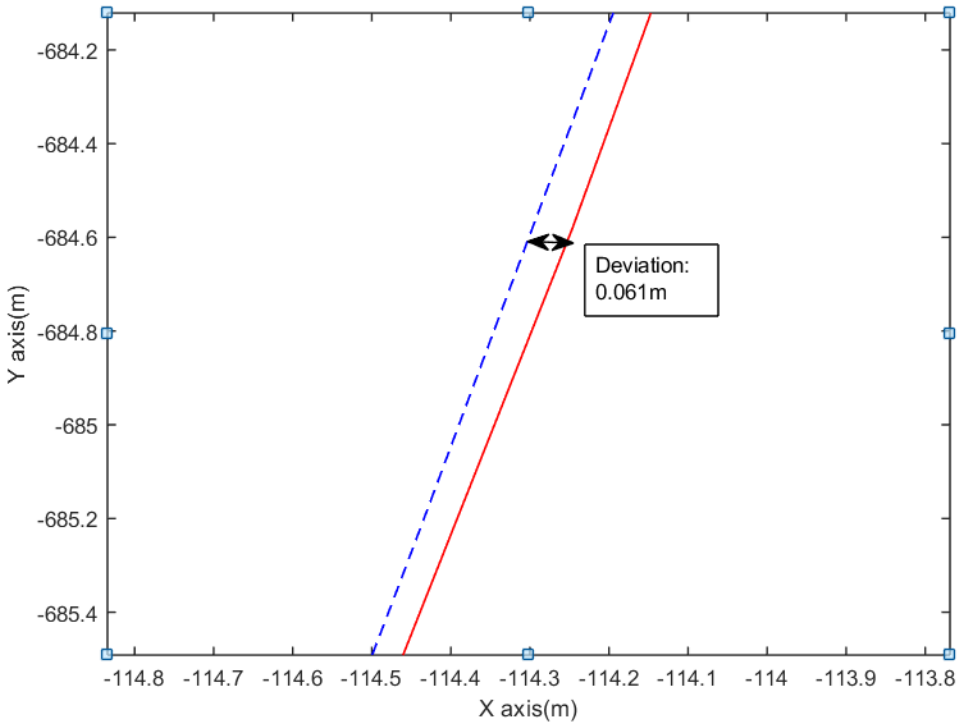


Figure 5.2: Close-up of the path following experiment depicted in Fig. 5.1 where the tracking error can be appreciated.

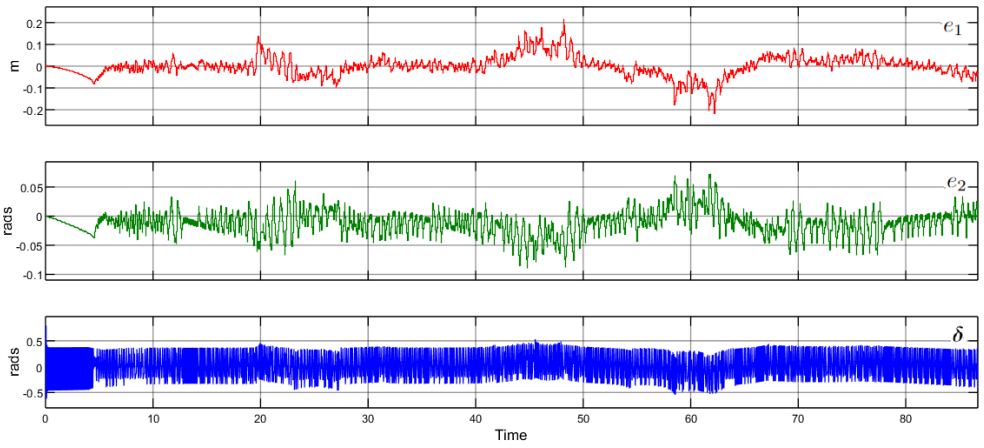


Figure 5.3: Variable errors e_1 (top) and e_2 (middle). Control signal δ (bottom) which is the position of the BLDC shaft that controls the steer angle.

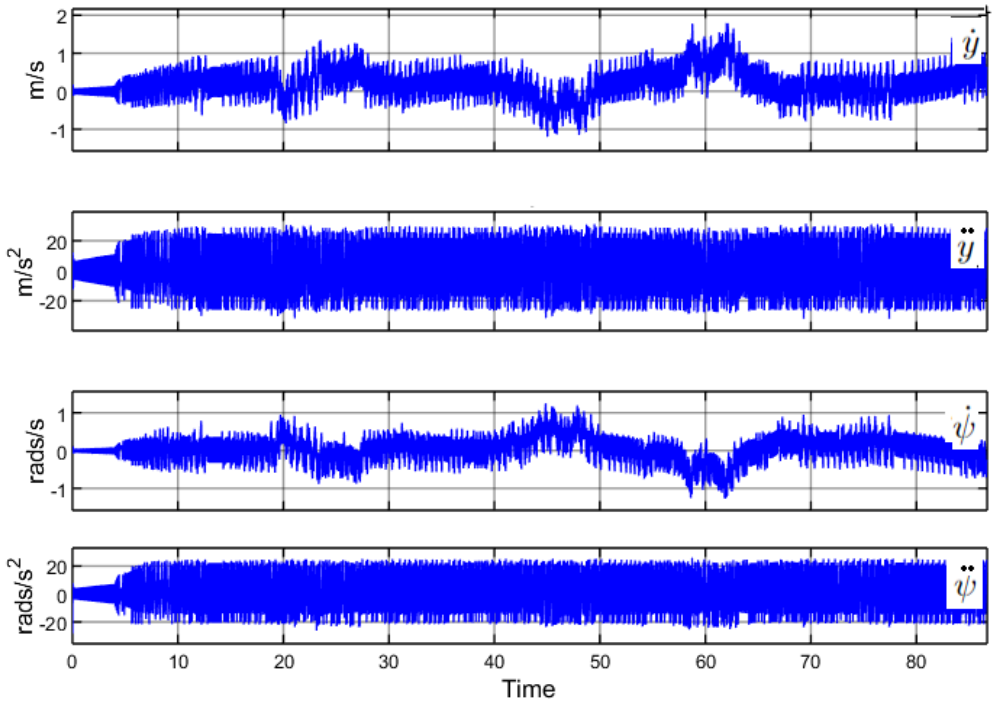


Figure 5.4: States of the lateral dynamics of the vehicle.

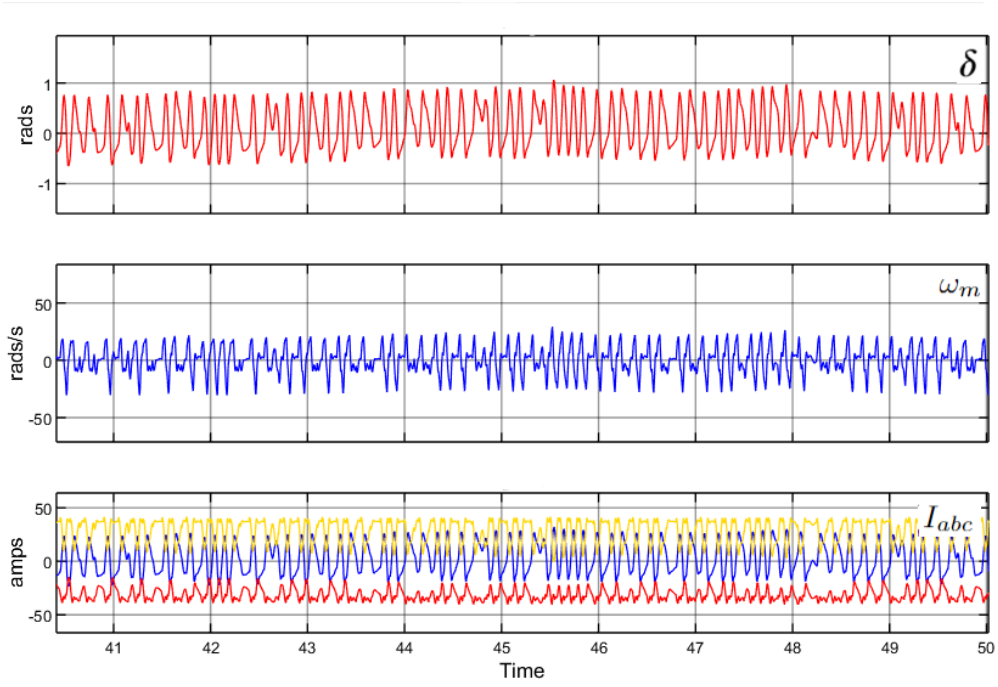


Figure 5.5: BLDC Internal states.

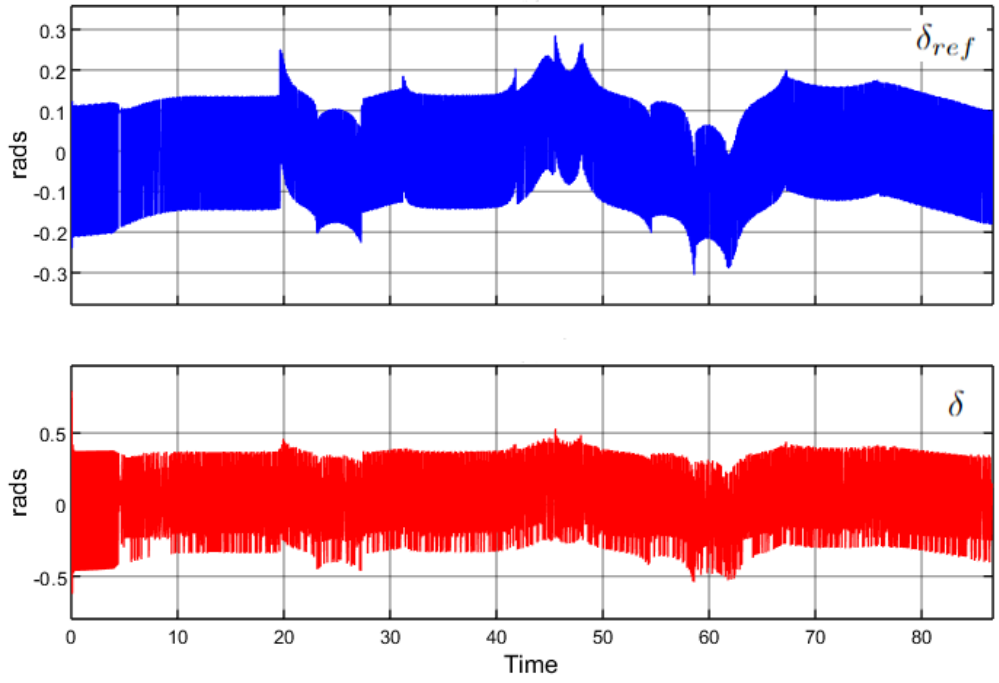


Figure 5.6: Steering angle δ and its reference δ_{ref}.

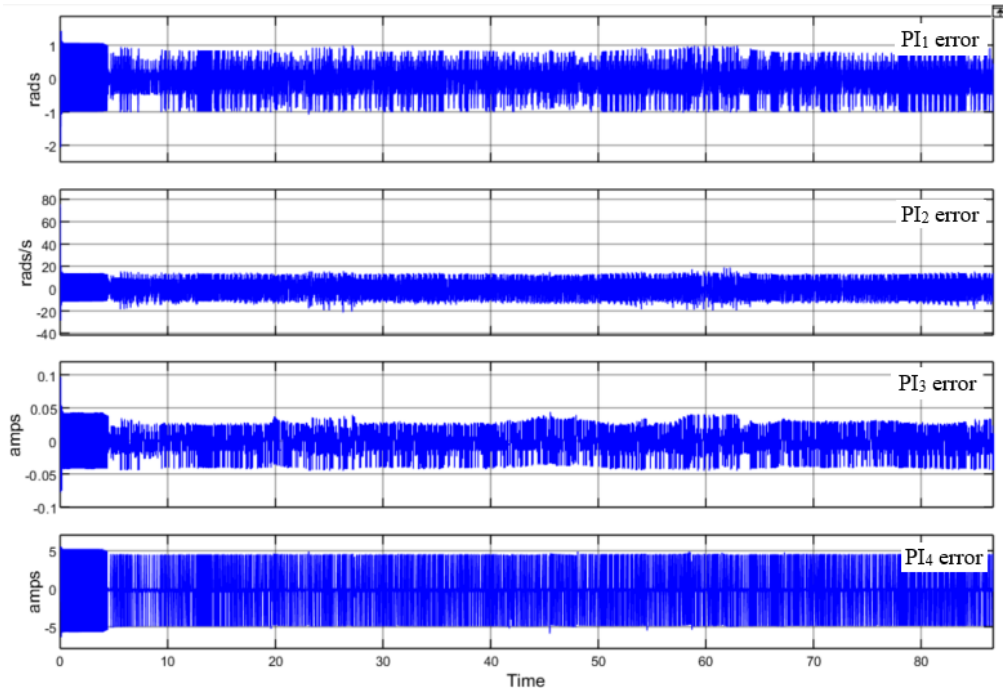


Figure 5.7: Errors of the 4 PIs in the BLDC control loop.

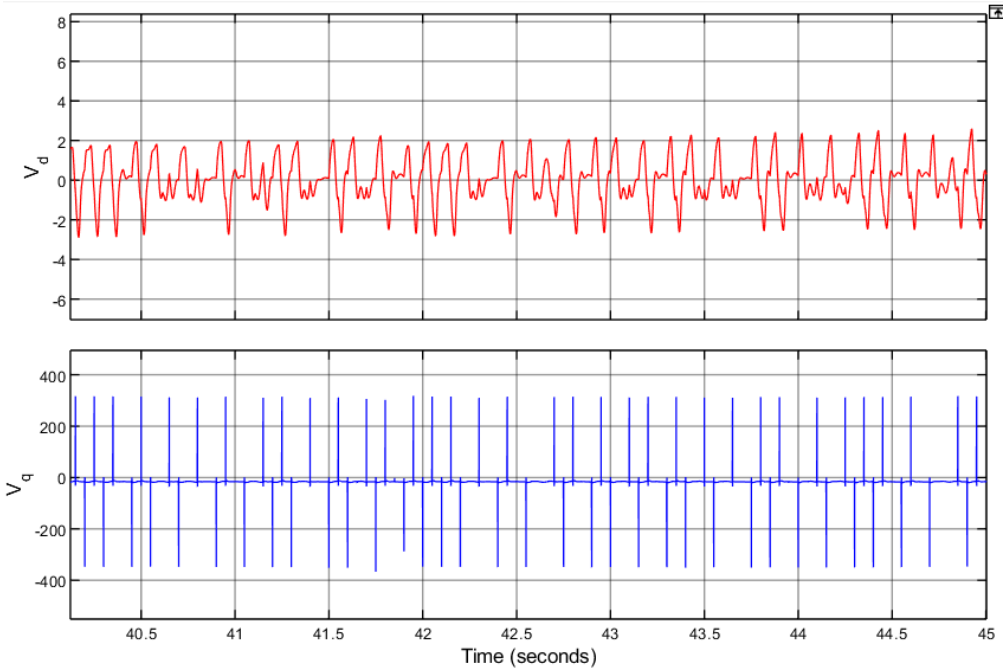


Figure 5.8: Input voltage for the BLDC motor in the dq reference frame.

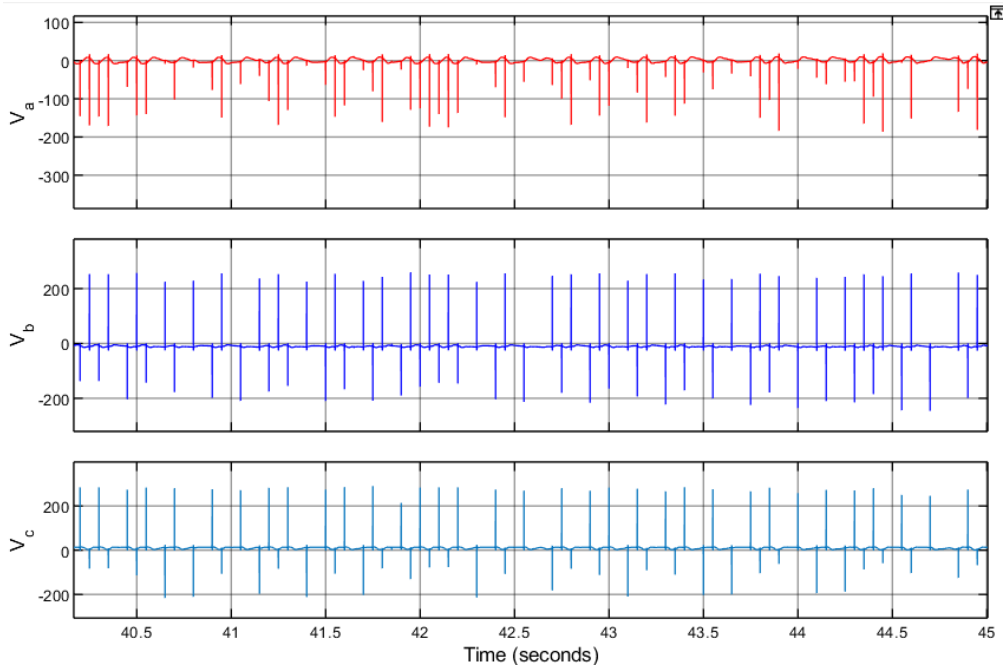


Figure 5.9: Input voltage for the BLDC motor in the abc reference frame.

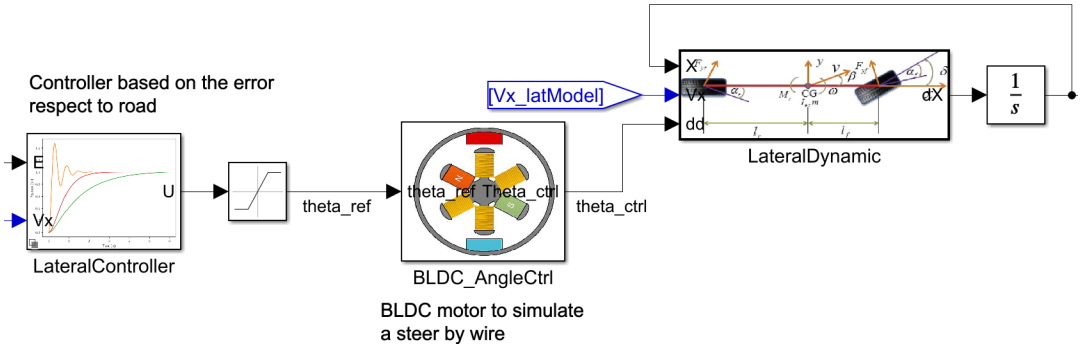


Figure 5.10: Lateral control blocks in Simulink model.

5.1.2 Configuration of the Closed-Loop System in Simulink

This section is intended to give specific hints in case the reader wishes to simulate the Simulink dynamical model defined in this work and which was used to obtain the corresponding reported results [20], [79]. The main blocks composing the Simulink model are depicted in Fig. 5.10.

First, it is required to set up the model to make it easier to run. The model consists of lateral and longitudinal dynamics. Since the longitudinal dynamics are not required, the upper section can be commented out, this will result in a lighter model and a reduced simulation period. Since we remove the longitudinal dynamics, we need to add the longitudinal velocity manually. For this, connect the ”Signal Generator” block, which is already contained in the model, to the label ”Vx_latModel”. You can manually specify the velocity profile you wish for the experiment. In our case this was set up to 22m/s (80km/h approx.).

In the left, the ”LateralController” block is a variant block. Choose the variant ”SMC Conventional V2” which is the implementation of the proposed controller. In the middle, the block ”Actuator” is a variant block where you can choose between ”BLDC”, ”generic transfer function” or ”No Actuator” as the corresponding model of the actuator. To replicate the reported simulations, the option ”BLDC” must be selected which converts the model into a stiff system. If the model is executed, the accuracy of the simulations will be severely decreased. To fix this issue, the model settings must be redefined. In particular, the ODE solver must be set to a Fixed-step Size solver with a sample time of 5E-6 s. The block to the right is the implementation of the lateral dynamics. This is a variant block too and, hence, it must be configured to ”Not simplified model”. This ends the

configuration process of the model.

5.1.3 Discussion

The simulation results presented in the previous section demonstrate that the designed control scheme achieves the path-following of the proposed reference in a satisfactory manner. This is depicted in Fig. 5.3 where a maximum lateral error of 0.2 m was presented during the experiment. It is worth noting that the experiment was carried out at a constant longitudinal speed of 80 km/h in comparison to [73] where the maximum test speed was 40 km/h. Moreover, a constant bank road angle was considered during the whole experiment which represents another factor to be counteracted by the control law. Additionally, the actuator dynamics were considered for the simulations which represents a more realistic implementation of the controller as the generated control signals corresponds to the voltage input of the electric actuator, which was assumed to be a BLDC electric motor according to most of the current commercial electric vehicles. To get this response, the actuator's current demand was up to 50 A in its peak value which falls within the limits of most of the available drivers for this type of motors. On the other hand, and due to the application of standard sliding modes, the control signals are characterized by the presence of high frequency components as shown in Fig. 5.8 and Fig. 5.9. Although this chattering effect is less visible in the mechanical variables of the vehicle, is still an undesired feature of the proposed controller as reduces the life cycle of the actuators. Despite this, the overall effectiveness of the control scheme is demonstrated by means of the presented simulations results.

5.1.4 Conclusions and Future Work

A control scheme based on sliding modes for an electric vehicle following a path was presented. In addition, the control schemes consider a PI-based FOC (Field-Oriented Control) for the steering actuator which was assumed as a BLDC motor. The general performance of the proposal was satisfactory as the control objective was fulfilled despite considerations as a constant 80 km/h longitudinal velocity and a constant rad bank angle. In comparison to other works, the inclusion of

the actuator dynamics represents a more realistic design for the control signal which corresponds to the voltage of the electric actuator. Despite the presence of high frequency components in the resulting control signal, the effectiveness of the overall control algorithm was demonstrated as a low bound for the lateral error and yaw angle error variables was attained.

Next steps of this work will be oriented to the inclusion High-Order Sliding Modes (HOSM) to the controller design to avoid the presence of high frequency components in the control signal. Moreover, the implementation of the designed control scheme on simulators as CARLA [12] or CarSim [13] to enhance the complexity and realism of the experimental scenarios and to include the effect of sensors and signal conditioning stages in the performance of the controller. Another improvement that will be implemented in future stages of the project is to consider the dynamic double-track model of the vehicle during the controller design process. The motivation is to increase the accuracy of the model, especially when the vehicle is operating at high speed and lateral tire slip is present during the experiments.

TABLE 5.3. PARAMETERS OF THE VEHICLE AND THE ACTUATOR.

Symbol	Value	Symbol	Value
l_r	1.58 m	b	$34 \times 10^{-4}\ N \cdot m \cdot s/rad$
l_f	1.1 m	P	8
I_z	2873 $kg \cdot m^2$	J_s	$24 \times 10^{-3}\ kg \cdot m^2$
m	2238.93 kg	b_s	$1.72 \times 10^{-3}\ N \cdot m \cdot s/rad$
$C_{\alpha r}$	8×10^4	N_m	4 (4:1 ratio)
$C_{\alpha f}$	8×10^4	N_l	6×10^{-5}
g	9.81 m/s^2	ϵ	0.1
V_x	18 m/s		
R	0.08 Ω		
λ_e	$3.33 \times 10^{-2}\ V \cdot s/rad$		
$L_s - M_s$	$1 \times 10^{-4}\ H$		
J	$1.8 \times 10^{-4}\ kg \cdot m^2$		

5.2. Block Control with Super Twisting Sliding Modes for the Path Following of a Self-Driving Vehicle Considering Actuator Dynamics

5.2.1 Simulation Results

Before proceeding with the simulation results, the authors want to mention that the readers can download the SIMULINK model corresponding to this work at [20] or [79]. It is a work in progress, but currently includes longitudinal dynamics, tire dynamics, lateral dynamics, and BLDC/DC motors as steering and traction actuators. Moreover, it contains the implementation of a variety of control algorithms for lateral dynamics. Table 5.3 shows the vehicle parameters used in the simulation experiments, which were obtained from [25] and [71]. On the other hand, Table 5.4 shows the controller gains, which were defined accordingly to the stability conditions defined in Theorem 5.

TABLE 5.4. CONTROL GAINS FOR LATERAL AND ACTUATOR CONTROLLERS.

Symbol	**Value**	**Symbol**	**Value**
k_1	$\begin{bmatrix} 30 & 6 \\ 6 & 6 \end{bmatrix}$	$K_{P,2}$	-15
k_{v0}	1	$K_{I,2}$	-75
k_{v1}	1	$K_{P,3}$	40
k_{u0}	1	$K_{I,3}$	80
k_{u1}	1	$K_{P,4}$	30
$K_{P,1}$	50	$K_{I,4}$	0
$K_{I,1}$	2	Λ	4

Fig. 5.11 shows the trajectory tracking performance for both controllers. During the reported experiments, it is assumed that $V_x = 18\ m/s = 65\ km/h$. This is a constraint on the dynamics of the vehicle, as it is not possible to follow a prescribed path with constant longitudinal velocity in a real scenario. However, $65\ km/h$ is an acceptable nominal velocity for an autonomous electric vehicle which allows us to validate the effectiveness of the proposed control scheme.

Furthermore, a time variable road bank ϕ (or superelevation of the road) was defined of the form

$$\phi = \tan^{-1}\left(\frac{V_x^2}{127R}\right) \tag{5.1}$$

where R is the radius of the curvature of the desired trajectory. Notice that this expression is a simplification of a real road bank which is usually more complex and takes into consideration many other elements. However, this version of the road bank allows us to recreate a more realistic version of the dynamics of an autonomous vehicle. The left part of the image depicts the global trajectory tracking and the right part shows a close-up of a segment of the trajectory at $t = 30s$. It can be seen that the control objective was achieved and, in addition, Table 5.5 shows that the average lateral displacement error is $\approx 0.0221m$, which is 10 times smaller than the resulting average error of [81].

Fig. 5.12 shows the lateral displacement error y_e and orientation error ψ_e during the experiment, which confirm the good performance of the controller. Moreover, the steering angle δ and its reference δ_{ref} are also shown. Compared to [81], the control input is no longer characterized by the

TABLE 5.5. RESULTS OF MAXIMUM AND MEAN VALUES FOR THE ERRORS Y_E AND ψ_E.

Variable	Maximum Value	Mean Value
y_e (m)	0.02976	0.00320
ψ_e (rads)	0.03154	0.00716

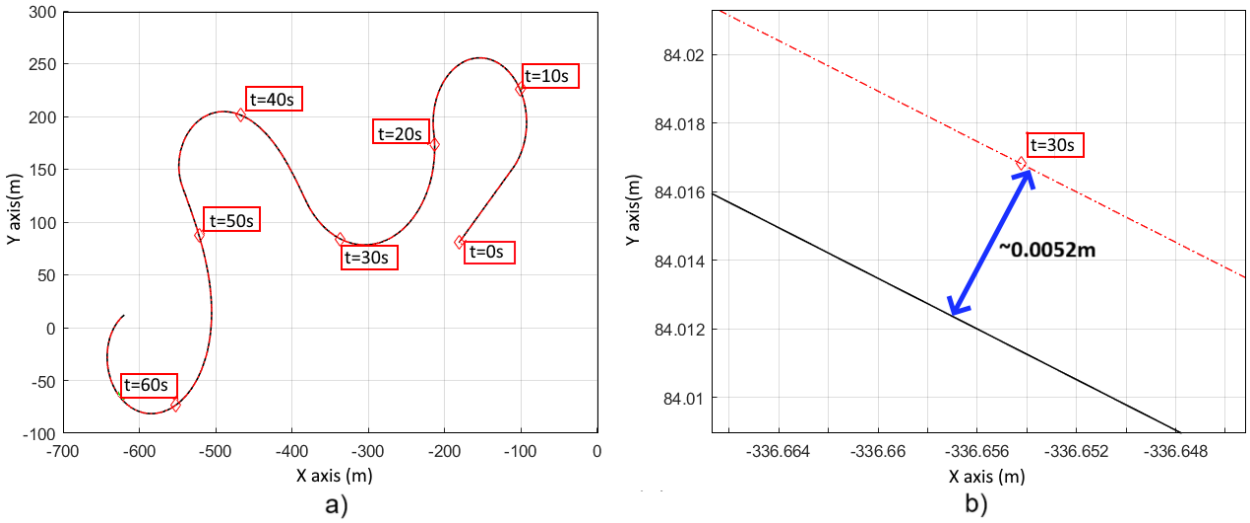


Figure 5.11: a) Complete trajectory tracking of the vehicle in closed-loop. b) Close-up at $t = 30$ s. The solid black line is the reference trajectory and the dashed red line the actual trajectory of the vehicle.

presence of chattering and, in addition, it is bounded by $0.12rad$ which represents a steering angle that is feasible to implement in a real steer-by-wire system for autonomous driving.

The lateral speed $\dot{y}$, lateral acceleration $\ddot{y}$, orientation velocity $\dot{\psi}$ and acceleration $\ddot{\psi}$ of the vehicle are shown in Fig. 5.13. All these variables remain within the expected values for a real automotive vehicle.

On the other hand, Figure 5.14 presents the error variables of the four internal PI controllers, which are part of the Field Oriented Control of the steering actuator. It can be observed that the error variables of the PI controllers for the actuator are characterized by high peaks and, in the case of the PI_4, a constant high frequency component. These undesirable effects are considerably attenuated by the low-pass filter nature of the electric motor and due to the saturation of its driver, which forbids the application of voltages greater than 24 V to the motor. This can be appreciated in

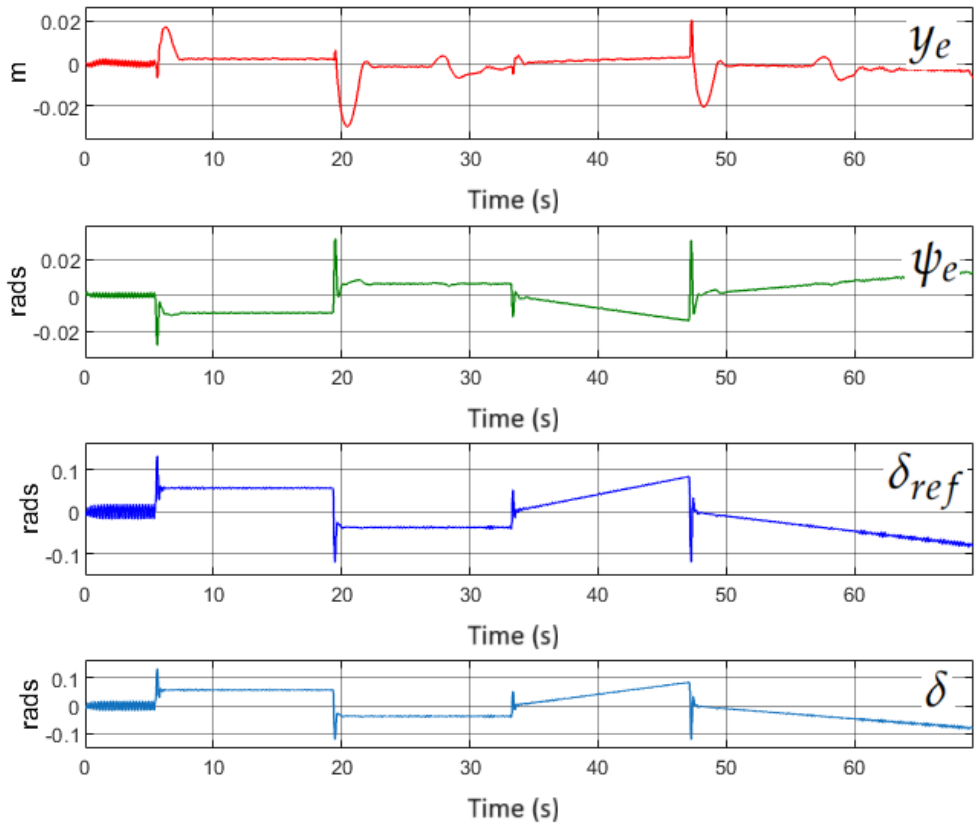


Figure 5.12: Lateral displacement error y_e, orientation error ψ_e, desired δ_{ref} and actual δ steering angle during the experiment

Figure 5.15, where the generated voltages are within the normal range of the considered electric motor. Besides that, the figure demonstrates the effectiveness of the controller and how it responds to abrupt changes in the reference δ_{ref}.

In order to test the robustness of the proposed controller, a second experiment was conducted where a disturbance term $\lambda(t)$ was added to the system dynamics, which was defined as

$$\boldsymbol{\lambda} = \begin{bmatrix} \frac{3}{5}\sin(\frac{\pi}{4}t) \\ \frac{\pi}{60}\sin(\frac{\pi}{8}t) \end{bmatrix}. \tag{5.2}$$

It is worth noting that, although this disturbance term was theoretically designed to test the robustness of the global controller, it represents perturbations on the lateral acceleration and in the yaw angular acceleration of the vehicle. Indirectly, this defines the effect of an external lateral force and a yaw torque on the vehicle dynamics.

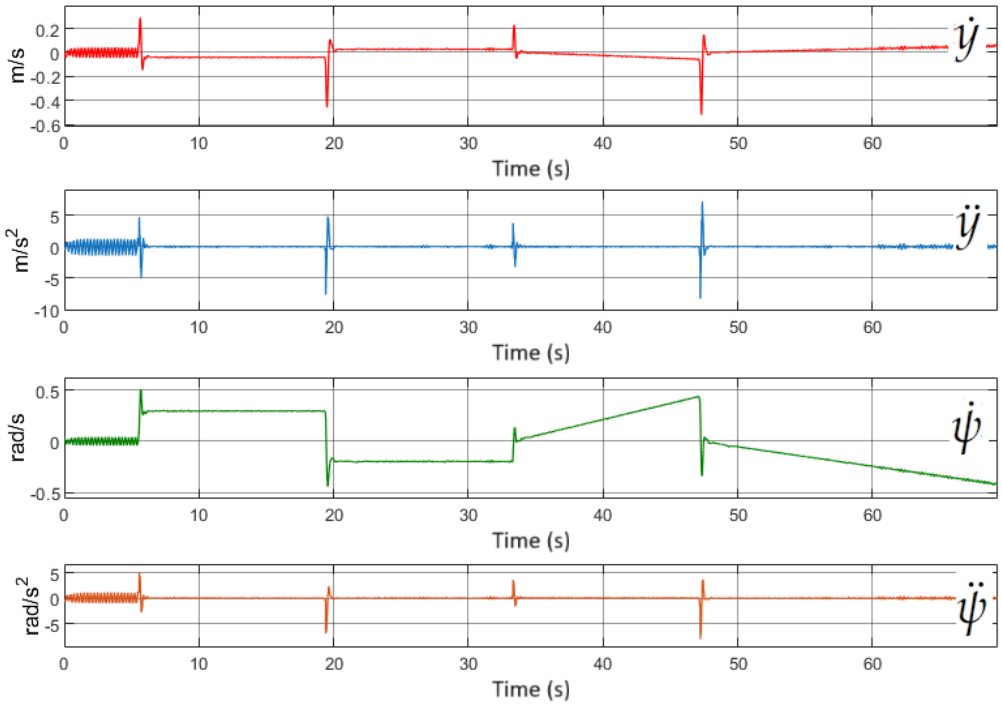


Figure 5.13: Velocities and accelerations for the lateral displacement and orientation of the vehicle.

The simulation results are shown in Fig. 5.16: in a) the error variables y_e and ψ_e are presented for both experiments: considering the disturbance term and neglecting it; and in b) the desired δ_{ref} and the actual steering angle δ are displayed. It can be noted that the performance of the controller is evidently affected by the disturbances, but not significantly as neither the magnitude nor the bandwidth of the variable errors were increased. This conclusion applies to the dynamics of the error systems and the input control signals of the overall control scheme. This strongly suggests the robustness capabilities of the proposed control scheme. Finally, a comparison between the performance of the control scheme designed in this work and the one proposed in [81] is shown in Figure 5.16. It can be seen that the magnitude of the control signal for the lateral controller is lower in the case of the controller proposed in this work. Moreover, the high frequency components in the control inputs and in the error variables are considerably reduced, which directly impacts the efficiency of the overall control scheme and the life cycle of the vehicle actuators.

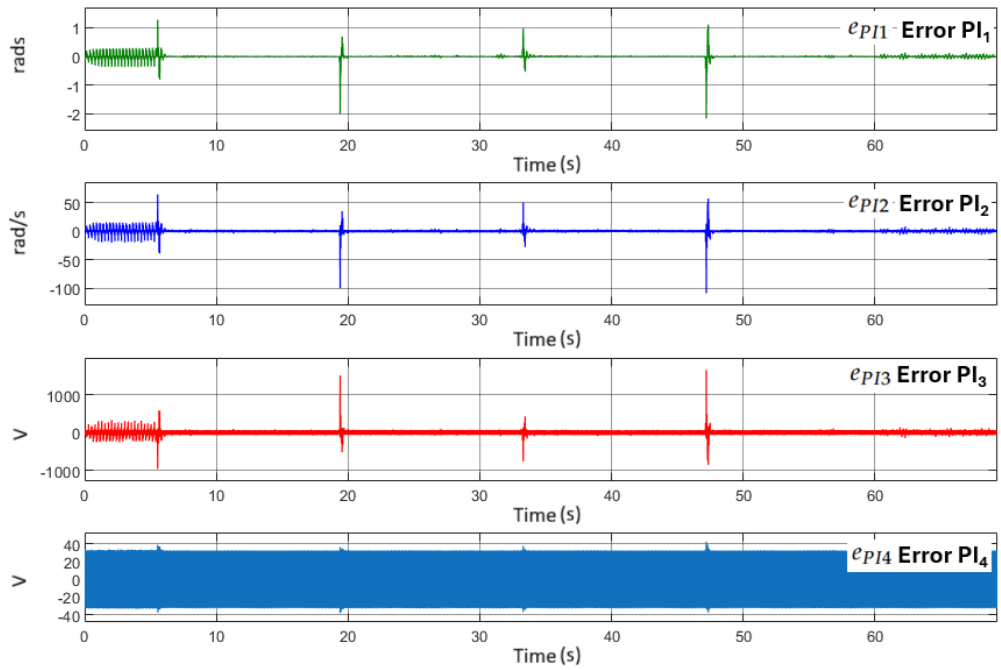


Figure 5.14: Error variables of the PIs controllers for the BLDC control loop as defined in 4.46.

5.2.2 Discussion

The simulations results have demonstrated the effectiveness of the proposed control scheme in terms of path following, robustness against matched perturbations, relative smoothness of the control signals, and its implementation feasibility. For instance, in comparison with the results reported in [43], the proposed controller reduced 50% the lateral displacement error during the simulations. On the other hand, related works as [60], [99], [89],[97],[24], [29] and [74], develop their respective control algorithms based on the kinematic or lateral dynamical model of the vehicle which represents a inconvenient when a real time implementation of the controller is necessary. Moreover, these works do not consider the dynamics of the actuators (usually electric motors) which modifies the dynamics of the overall vehicle. By discarding this factor, the probability of obtaining the same performance on a real vehicle are drastically reduced. Finally, the improvement against [81] is evident as the chattering effect was significantly reduced and the resulting smoothed control signals are more suitable for a real time implementation without compromising the integrity of the actuators.

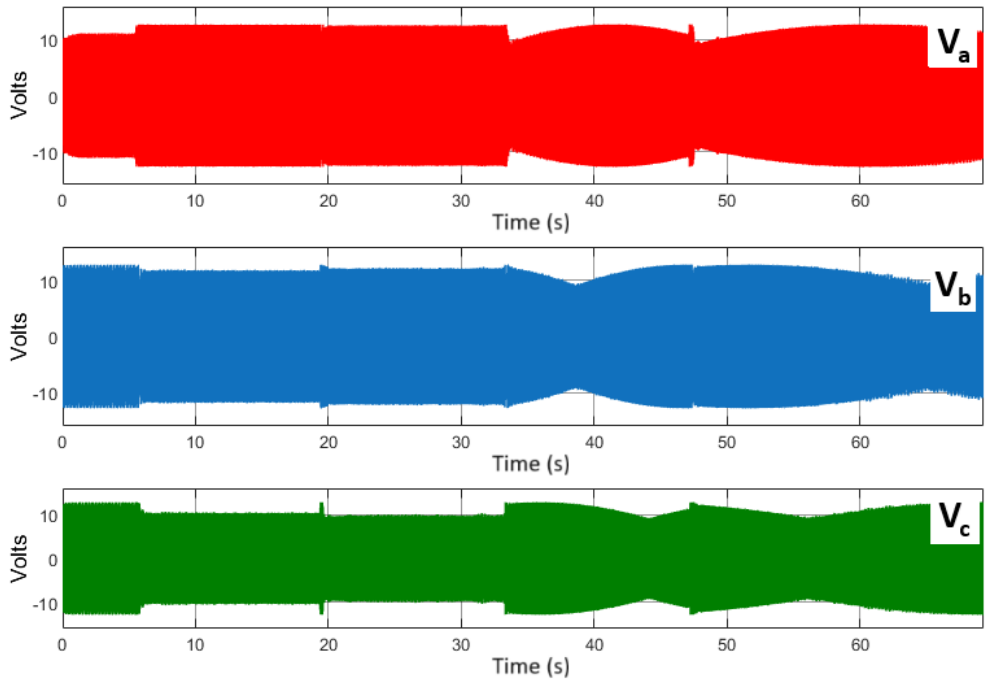


Figure 5.15: Voltages of phases a, b, c that feed the BLDC.

5.2.3 Conclusions and Future Work

In this work, a control scheme for the path following of an electric autonomous vehicle was developed and implemented in a simulation environment. It is based on the combination of three control techniques: the Block Control, Integral Sliding Modes, and the Super-Twisting Algorithm. The dynamical model of the electric vehicle is composed of the lateral dynamics of the vehicle, the electrical actuator dynamics and the steering rack dynamics. It is assumed that the longitudinal velocity of the vehicle is constant, but the road bank depends on the curvature radius of the path to follow. In addition, a disturbance term was added in the second simulation experiment to validate the robustness of the proposed control scheme. The proposed control scheme integrates the control signal's smoothness that characterizes the Super-Twisting Algorithm with the Block Control technique that allows controlling variables whose dynamics is not directly affected by the control inputs. According to the simulation results presented, the performance of the proposed control scheme is satisfactory as the path following of a desired trajectory, which is composed of a series of way points in the X-Y plane, was achieved. Moreover, the control inputs, which correspond to the steering angle of the steering rack system and the input voltages of its actuator, are consistent in

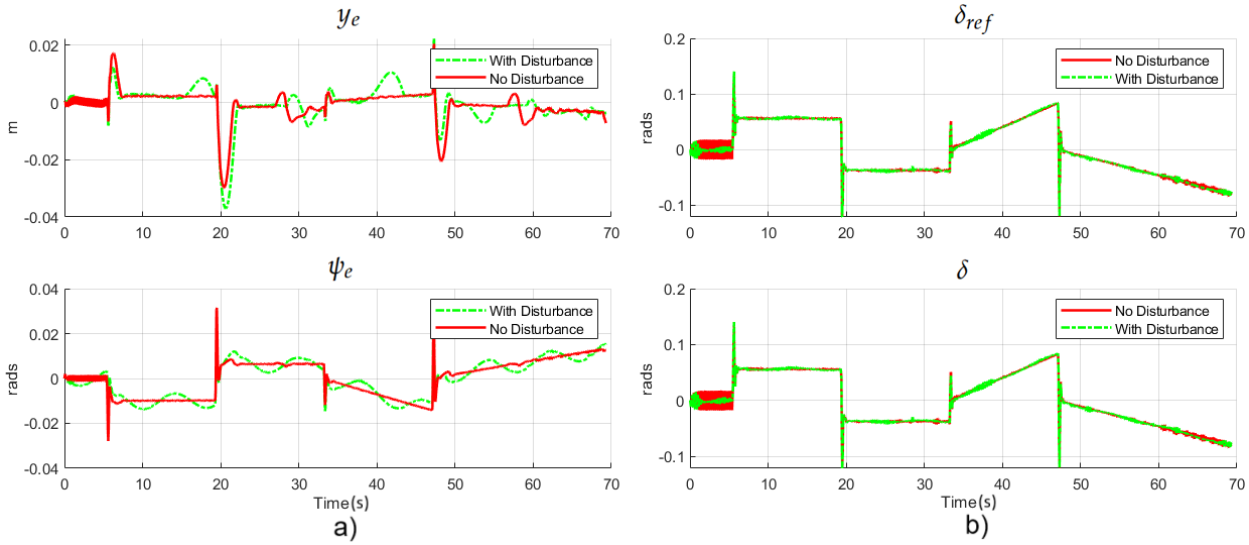


Figure 5.16: Error variables response of the proposed controller against matched disturbances. a) shows the closed loop response of the lateral displacement error y_e and orientation error ψ_e considering disturbances (dashed green lines) and without disturbances (red lines), and b) shows the corresponding steering control input.

terms of magnitude and bandwidth with the standard demand of the actuators of a real autonomous automotive vehicle. On the other hand, compared to the results obtained in [81], the performance of the proposed controller exhibits a significant reduction of chattering and high-frequency components in the control signals. Also, as shown in Table 5.5, the average lateral displacement error is 10 times smaller than the resulting average error in [81]. Furthermore, the magnitudes of the control inputs, steering angle, and input voltages of the actuator, feature lower magnitudes as well. On top of that, this closed-loop performance was obtained despite the presence of disturbances in the form of a time variant force and a torque applied directly to the vehicle's lateral displacement and orientation dynamics. The considered next steps for this work are the implementation of the proposed control algorithm car simulator such as CarSim [13] or CARLA [12] to validate its performance on a more realistic simulation environment. Also, further developments consider the integration of the longitudinal dynamics to the vehicle's model and its corresponding traction controller. This would improve the usability of the model for path following and stability evaluation. On the other hand, future implementation of the controller on a real prototype of an electric autonomous vehicle is considered where the integration of signal conditioning stages for sensors noise reduction will be crucial.

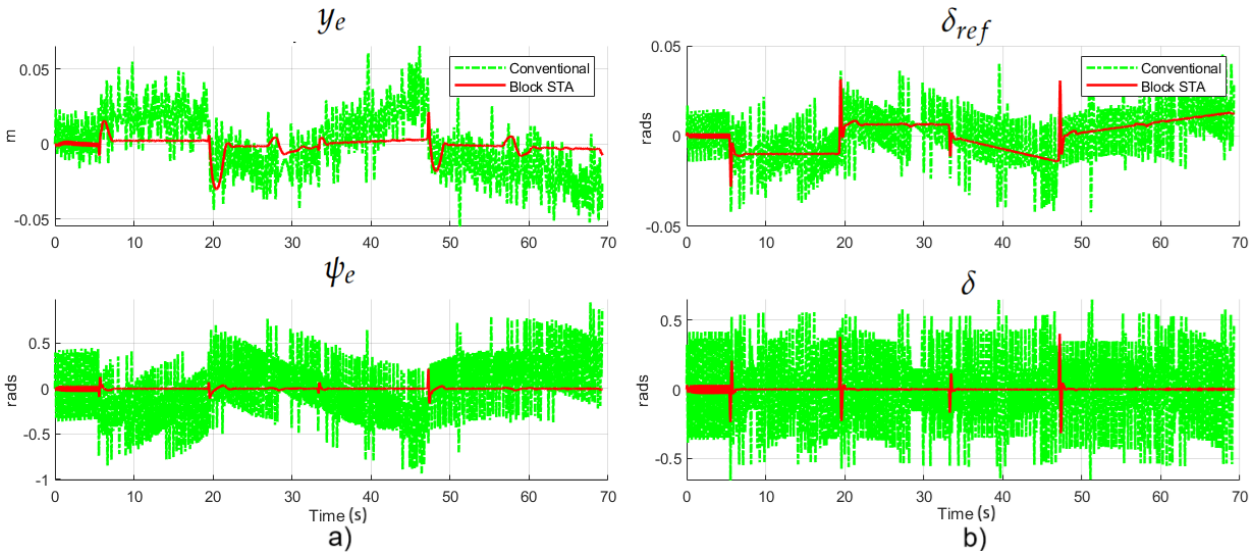


Figure 5.17: Comparison between the closed loop response of the controller proposed in this work (red lines) and the response of the controller in [81]. The improvement in terms of the control law magnitude and its high frequency components is clearly appreciated.

5.3. Linear Block Control Plus Super Twisting Sliding Modes for Disturbance Rejection for the Path Following of a Self-Driving Vehicle Considering Actuator Dynamics (Variant)

5.3.1 Simulation Results

First of all, it is worth mentioning that the reader can download the simulation used in this work from [20] or [79]. This model is continually improved. At the moment of writing these lines, the model has available the longitudinal dynamics, tire dynamics, lateral dynamics, besides several control algorithms for the lateral dynamics, and it contains models of DC motors and BLDC. This model takes a file with a list of waypoints and calculates the necessary errors required by the lateral controllers. Tables 5.6 and 5.7 show the parameters used for simulation. These parameters are basically the same as those used in [81] where the only change is the lateral controller and some parameters in the actuator. There, a conventional sliding mode technique was used, whereas here we used linear feedback control by blocks plus the super-twisting control technique. This is important to be able to compare the performances of the two control techniques.

Fig. 5.18 shows a small segment of the vehicle trajectory on the ground at a speed of 65km/h. The dimensions are in meters. The black line represents the desired trajectory coming directly from the way-points file. The dashed green line represents the actual trajectory of the vehicle (simulation) using the control $(\delta_k + \delta_u)$. The red line represents the trajectory using only the control δ_k. In Fig. 5.18 b) we see a close-up in $t = 30s$. Here, the deviation is 0.0221m aprx. which is smaller than the one depicted in [81] although this is not the same trajectory. We can corroborate from Table 5.8 that the maximum error is around 0.03278m for y_e while in [81] is around 0.2m, which represents an improvement in the performance regarding the maximum error.

The general results are in Table 5.8 where there is a comparison of performance when using only the linear control δ_k and the complete control $(\delta_k + \delta_u)$.

Fig. 5.19 shows the error in vehicle dynamics, besides, the control law δ_{ref} and the actual position of the steering δ. comparing these errors against [81] we can see a good improvement in

TABLE 5.6. PARAMETERS OF THE VEHICLE AND THE ACTUATOR.

Symbol	**Value**	**Symbol**	**Value**
l_r	1.58 m	b	$0.00034N \cdot m \cdot s \cdot rad^{-1}$
l_f	1.1 m	P	8
I_z	2873 $N \cdot m$	J_s	0.024 $kg \cdot m^2$
m	2238.932 Kg	b_s	0.00172 $N \cdot m \cdot s \cdot rad^{-1}$
$C_{\alpha r}$	80000	N_m	4 (4:1 relation)
$C_{\alpha f}$	80000	N_l	0.00006
g	9.81 m/s^2	ϵ	0.1
V_x	18 m/s		
R	0.08 Ω		
λ_e	0.0333 $V \cdot s/rad$		
$L_s - M_s$	0.0001 H		
J	0.00018 $kg \cdot m^2$		

the response. We can observe in the error and overall, the control signal of the steering δ_{ref} and the signal after the actuator δ, that we no longer have a strong chattering as we had in [81] and hence, this controller is feasible to be implemented in an actual steer by wire system for autonomous driving.

The image depicted in Fig. 5.20 presents the internal states of the vehicle dynamics that is the lateral speed $\dot{y}$, lateral acceleration $\ddot{y}$, angular speed of the yaw angle or vehicle orientation $\dot{\psi}$ and the angular acceleration of the vehicle orientation $\ddot{\psi}$.

Following with the actuator performance, Figures 5.21,5.22, and 5.23 depicts the error in each PI controller for the BLDC, the triphasic voltage that feeds the BLDC and the currents in the dq reference frame, respectively. Fig. 5.24 shows the dynamics of the rack system. We can observe the magnitudes of the self-alignment torque τ_a and the friction torque τ_f. In this image we can see as well, that the torque load T_l that receives the BLDC motor is no more than $|0.52Nm|$ along the whole trajectory. Fig. 5.25 shows an interesting chart, where the internal structure of the controller

TABLE 5.7. CONTROL GAINS FOR LATERAL AND ACTUATOR CONTROLLERS.

Symbol	**Value**	**Symbol**	**Value**
k_1	9	$K_{P,2}$	-15
k_2	8	$K_{I,2}$	-75
k_v	0.0065	$K_{P,3}$	40
k_u	0.0016	$K_{I,3}$	80
U_M	$\pi/8$	$K_{P,4}$	30
$K_{P,1}$	50	$K_{I,4}$	0
$K_{I,1}$	2	$\boldsymbol{\gamma}$	$[5\ 1]^T$

TABLE 5.8. RESULTS OF MAXIMUM AND MEAN VALUES FOR THE ERRORS Y_E AND ψ_E.

Variable	**Maximum Value**	**Mean Value**
Only linear control d_k		
y_e (m)	0.0428	0.0184
ψ_e (rads)	0.03051	0.007094
Control $(d_k + \delta_u)$		
y_e (m)	0.03278	0.004456
ψ_e (rads)	0.02923	0.007059

is decomposed. We can appreciate in the section of $(\delta_{ref} = \delta_k + \delta_u)$ that we do not have high frequencies oscillations – chattering – in the final control law; however, it can be seen in the section for $\dot{v}$ that the signal enters and exits the slide mode. This is expected since we had to make a trade-off. If the gains for the STA are too large, it starts to behave similar to the conventional SMC, with high-frequency components; if the gains are too small, the STA becomes unable to eliminate the disturbances.

Fig. 5.26 is an experiment to see how much the controller is capable to reject uncertainties. For this, run another simulation and, instead of defining zero values for λ, which represents the

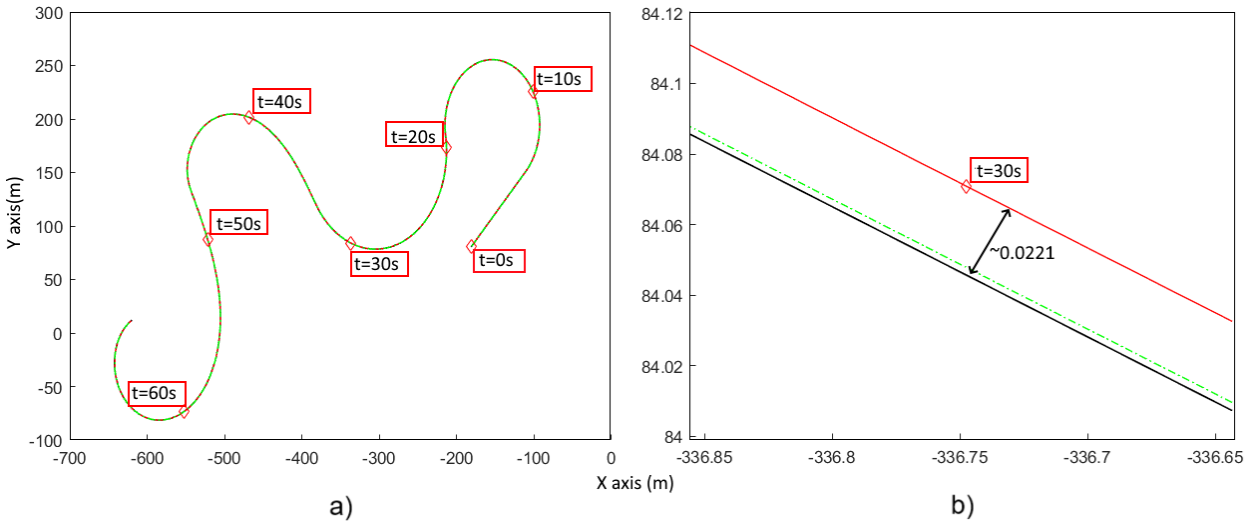


Figure 5.18: a) complete trajectory of the vehicle. b) close-up where the maximum deviation is shown. The black line is the reference trajectory. Dashed green line represents the actual trajectory of the vehicle using $(\delta_k + \delta_u)$. Solid red line is the trajectory using only δ_k.

uncertainties and non-linearities in (4.48), we define it as follows:

$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & = \frac{3}{5} \sin \frac{\pi}{4} t \\ \lambda_2 & = \frac{\pi}{60} \sin \frac{\pi}{8} t \end{bmatrix}. \tag{5.3}$$

There is no specific reason for these values beyond a consideration of the feasibility of the magnitude of the uncertainties and differences of behavior between the simulation and a hypothetical real vehicle. An additional element to take in consideration in this simulation is the determination of the road bank or superelevation of the road which corresponds to the variable ϕ in (4.17) or (4.48). Here, it was determined as follows:

$$\phi = \tan^{-1}\left(\frac{V_x^2}{127R}\right), \tag{5.4}$$

where R is the radius of the curve; see Fig. 4.7 b) where the distance y corresponds to R. Notice that this expression is an over simplification for the superelevation; usually the design of a road is more complex and takes into consideration many other elements; however, it was inserted here to test the performance of the control algorithm, not only with a fixed ϕ parameter but using a dynamic super-elevation according to the curves.

In the graph, Fig. 5.26, the superior graph on the left depicts the comparison in the behavior of the lateral errors y_e. The solid red line is the response to the controller $\delta_k + \delta_u$ without additional

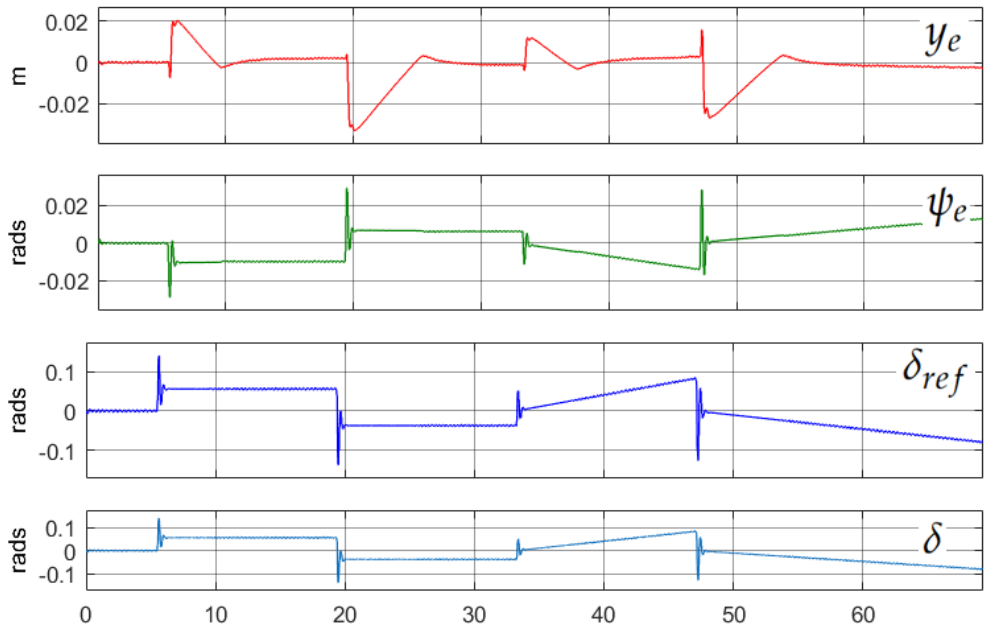


Figure 5.19: Errors in the vehicle dynamics.

disturbances. The blue dashed line is the response with the same controller adding the disturbances. Here we can observe that the behavior is almost unaffected. On the contrary, in the right side, we see the response to the control δ_k and we can observe that the blue dashed line is oscillatory, indicating that the disturbances are not being rejected at all. In chart c) and d) we can see that these are almost identical, indicating that the control $(\delta_k + \delta_u)$ is not being effective to the orientation error ψ_e. So we can conclude that the error caused by the uncertainty was considerably reduced in the y_e error; however, apparently the error in orientation ψ_e was not reduced by the effects of the super-twisting controller part δ_u.

5.3.2 Discussion

In this work, we develop an algorithm to control the steering angle based on the linear combination of two controllers, one based on error feedback which is linear, and the second one is an implementation of the super-twisting sliding mode control algorithm. This combination is in the hope that the linear controller compensates all the known dynamics while the non-linear, the super-twisting, compensates all unknown uncertainties and disturbances. We observed that effectively, the

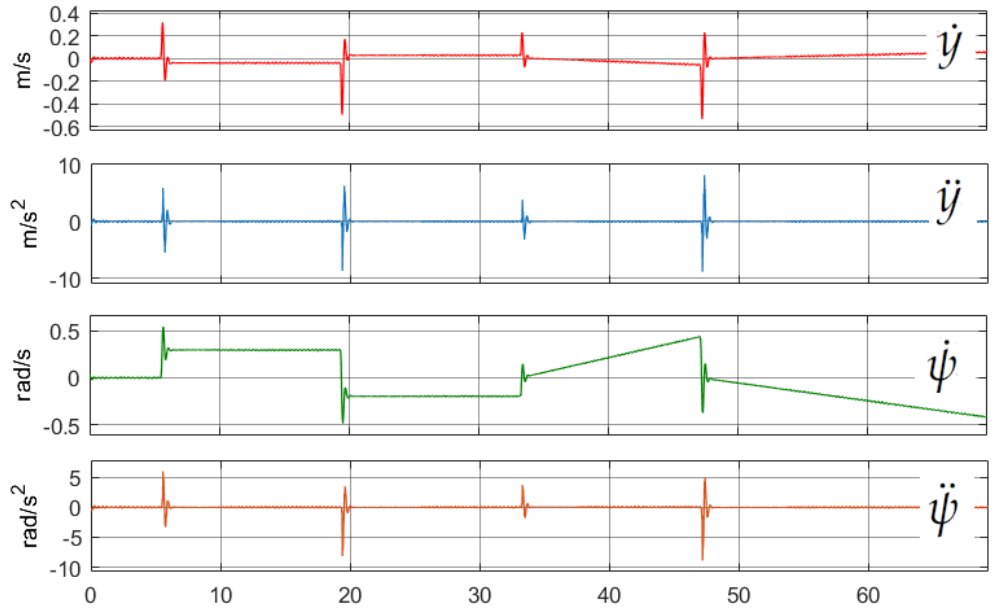


Figure 5.20: Internal states of the vehicle dynamics.

controller could significantly reduce the disturbances in the lateral distance error, although it did not have much effect in the orientation error disturbances. One big advantage of this algorithm against a conventional sliding mode developed in [81] is that the one developed here is feasible to implement, since it does not cause that much chattering as the conventional, which results impractical for many scenarios without a technique or procedure to cope with chattering. We took measurements of the behavior and impact of the controller on the actuator, in this case a BLDC electrical machine, and we could observe that the chattering in it is almost despicable. This is important since the presence of harmonics in the motor might reduce its efficiency.

5.3.3 Conclusions and Future Work

This controller algorithm resulted suitable to be implemented physically in an autonomous car prototype since it obtained good performance and minimum chattering. The performance can be improved even more by tuning the parameters or designing the linear part converting to a controller regular form the system and using the pole placement technique. Next steps to this work would be to implement this algorithm in a car simulator such as CarSim [13] or CARLA [12] to improve the

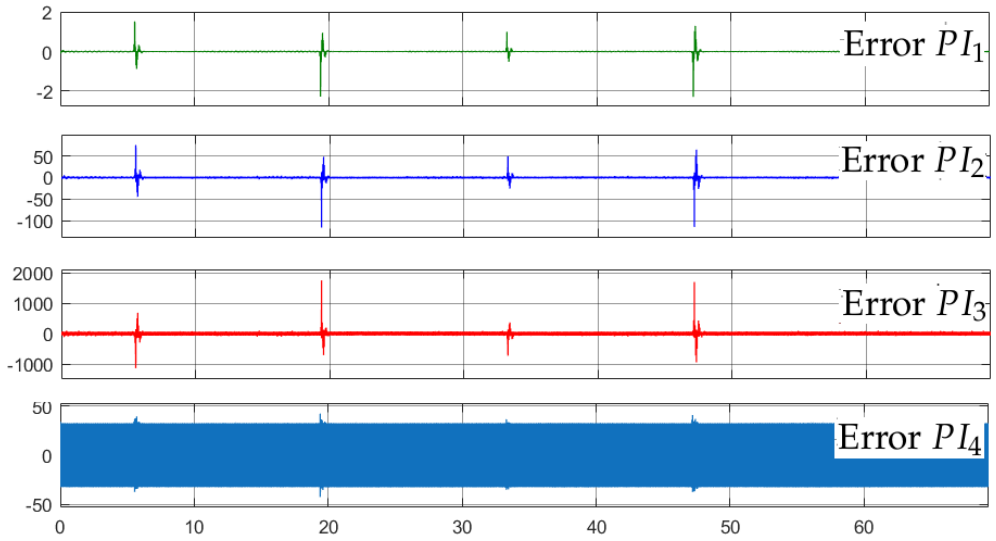


Figure 5.21: Errors in the PIs controllers for the BLDC control loop. The errors are defined in section 4.46.

realism of the simulations, since these simulators offer more precise physical models that are closer to the real life and it is the logical step before physical implementation, especially when a self-driving vehicle for testing is not available. Further developments should consider the longitudinal dynamics and define a controller for the vehicle traction and observe the impact in this lateral control since, for the development of the lateral controller, the assumption that the longitudinal speed is constant, was done. In the current simulation, we did not simulate the PWM that should be used in the variator to be able to control the current vectors of the BLDC to see the effects of this mechanism on the motor and in the controller; future work should consider the addition of this element in the simulations.

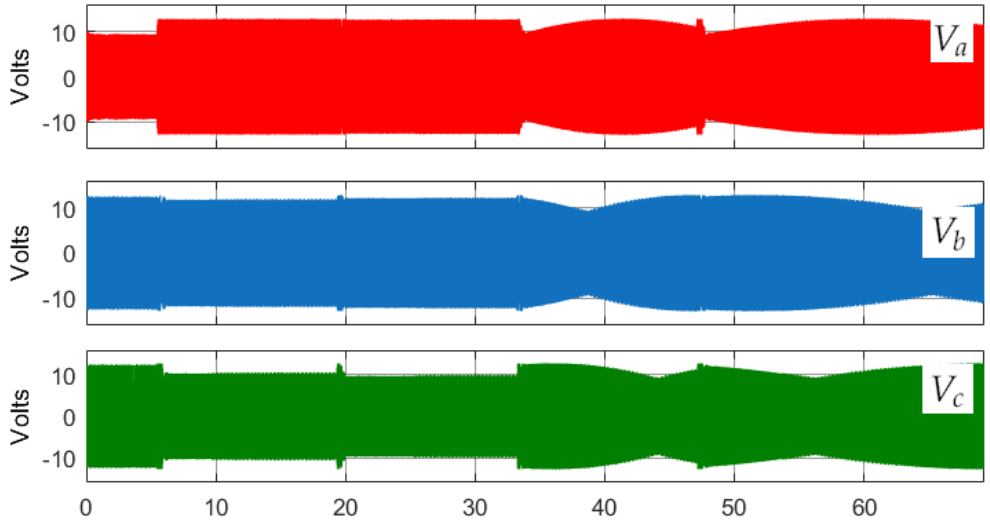


Figure 5.22: Voltages of phases a, b, c that feed the BLDC.

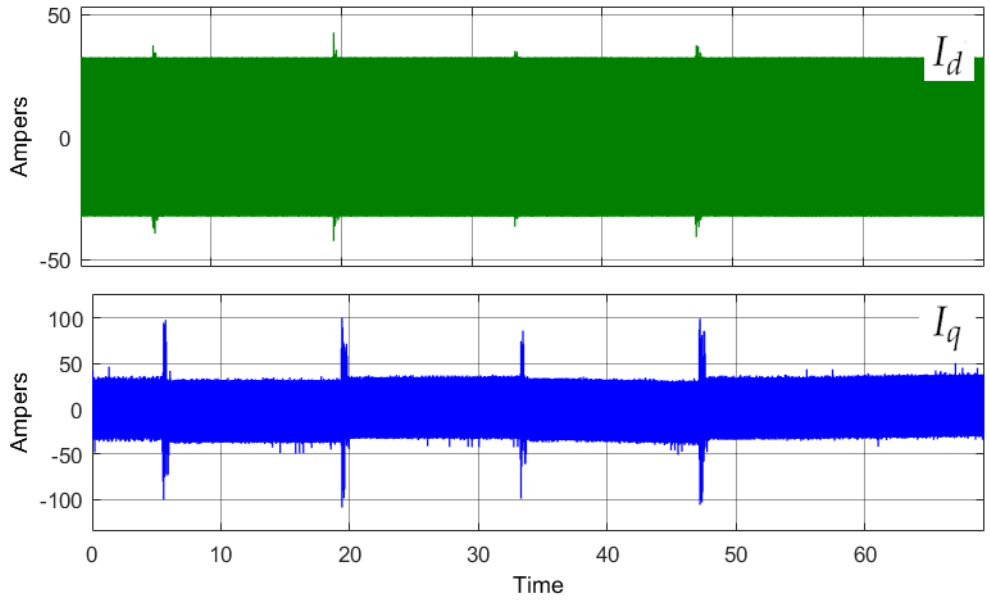


Figure 5.23: Currents of vectors in the rotational reference frame dq read from the BLDC.

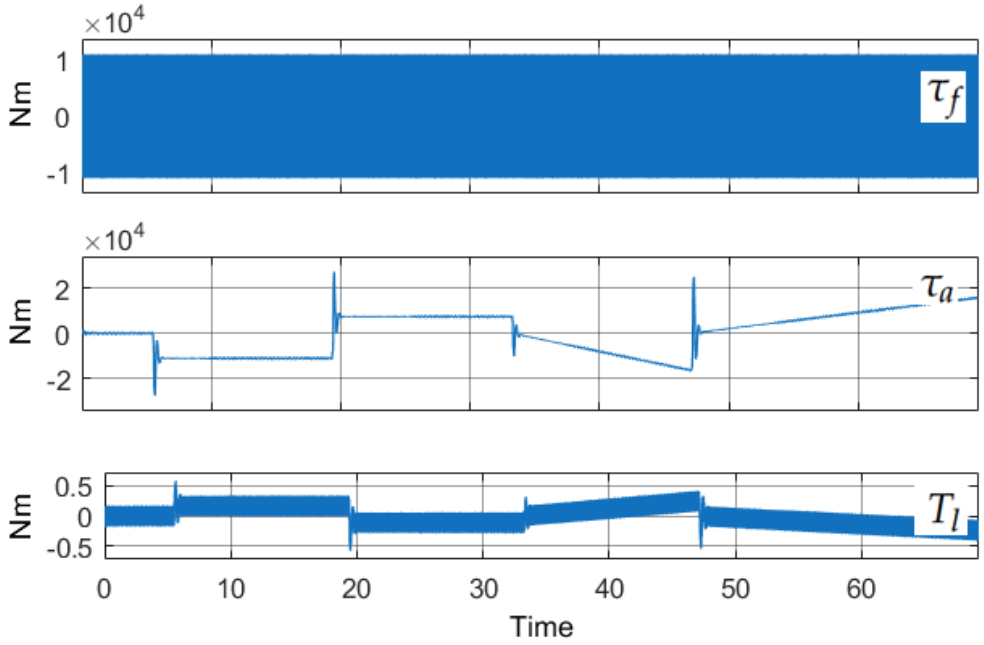


Figure 5.24: Forces in the steering rack system.

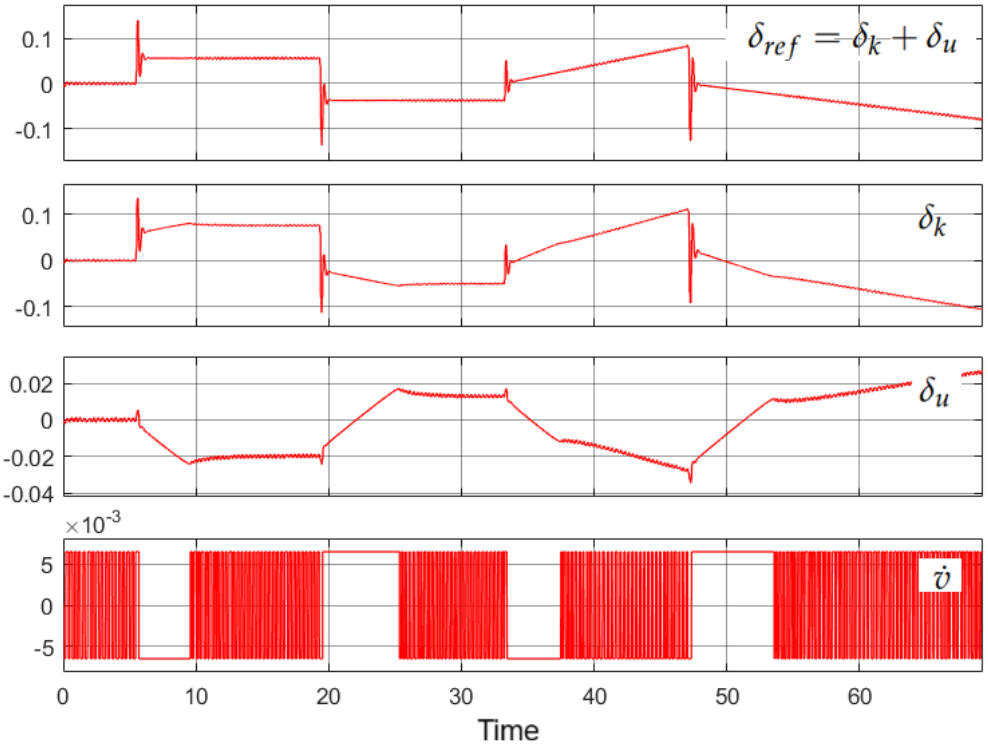


Figure 5.25: Internal structure behavior of the control algorithm.

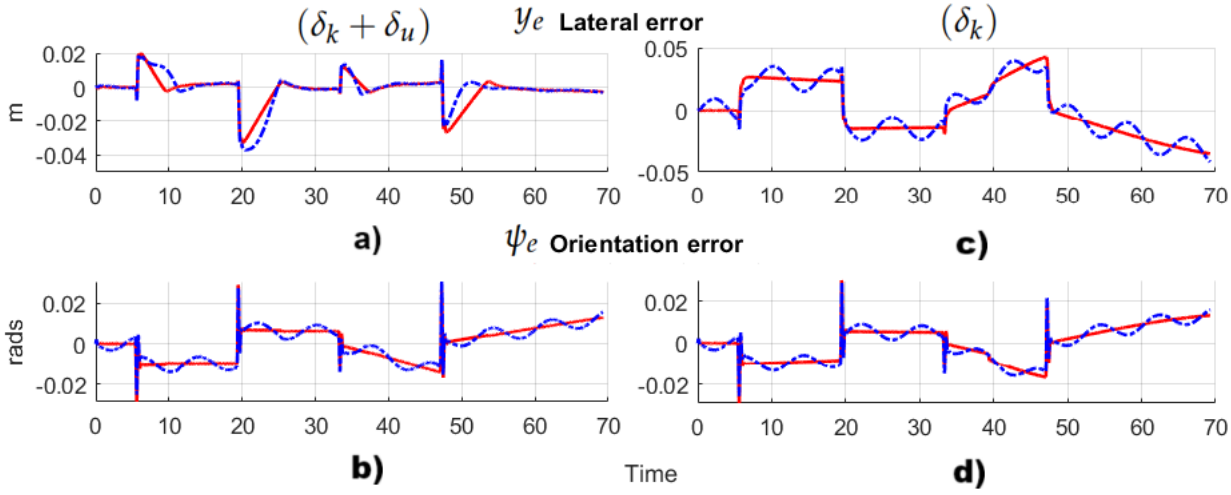


Figure 5.26: Error comparisons with additional disturbances. a) is a chart that represents the lateral error y_e with control $(\delta_k + \delta_u)$, solid red is without additional disturbances, dashed blue with disturbances. b) is the same than a) with ψ_e. c) control δ_k for y_e, solid red line no disturbances, blue dashed with disturbances. d) same as c) for ψ_e

General Conclusions

In this doctoral thesis, the problem of modeling and control of autonomous electric vehicles has been addressed from both an academic and industrial perspective. The increasing adoption of electric propulsion and advanced driver assistance systems has shifted the focus of the automotive industry toward electric actuators for steering and propulsion, making their accurate modeling and control a critical aspect of vehicle safety and performance.

A review of the current literature reveals a clear trade-off between control sophistication and model fidelity. Many studies apply advanced nonlinear control techniques, such as sliding mode control, using overly simplified vehicle models, while others rely on complex models but adopt basic control strategies and neglect actuator dynamics. Such inconsistencies limit the transferability of research results to real vehicles and reduce their industrial applicability.

This research distinguishes itself by developing and employing comprehensive vehicle models for control design and validation, explicitly incorporating actuator dynamics associated with electric machines. Simplifications commonly introduced during simulation are deliberately avoided in order to better reflect realistic vehicle behavior as far as possible. The proposed control strategies are designed and evaluated using these models, enabling a more reliable assessment of performance, robustness, and practical feasibility.

Among the most noteworthy results obtained throughout this research, we highlight the following:

1. Three theoretical controllers for lateral dynamics based on sliding mode control (SMC): The first is a theoretical controller based on conventional SMC, converting the mathematical dynamic model into controllable companion form. The second and third combines the block control technique to compensate for known dynamics and adds a super-twisting SMC to eliminate disturbances or unknown dynamics. While the first controller is purely theoretical and impractical for real-world application due to high-frequency requirements, the second and third resolves

this issue, making them suitable for implementation in real systems.

2. Development of a testing framework: As a result of this research, a small testing framework was created. This framework consists of an executable model that includes the vehicle's lateral and longitudinal dynamics, as well as the dynamics of the tires and actuators, such as DC and BLDC electric motors. Additionally, the model allows users to create custom vehicle routes or add their own way-point files and modify all parameters to adapt the model to different types of vehicles. This executable model is publicly available for any researcher interested in testing their control systems.

Future plans include continuing to improve the physical models and implementing controls such as adaptive sliding modes or predictive controls for both lateral and longitudinal dynamics. In addition, efforts will be made to provide a framework that evaluates controls from the perspective of standardized safety, such as ISO 26262 and ISO 21448.

As a final reflection, this work highlights the structural divergence that often exists between academic research and industrial practice. While academia primarily aims at generating and disseminating new knowledge, industrial research is necessarily driven by competitiveness, time-to-market constraints, and intellectual property protection. As a result, relevant data and technical information required for applied research are frequently subject to confidentiality restrictions.

In the automotive domain, this limitation is particularly evident in the availability of realistic system parameters. Data such as tire characteristics, electric motor parameters, detailed vehicle specifications, and actuator performance limits are often inaccessible or only partially available outside industrial environments. This lack of transparency poses a challenge for academic research seeking to develop and validate models and control strategies with direct industrial relevance.

In parallel, industrial research activities are constrained by development costs and by the need to justify investments through a clear or short-term return. These constraints naturally limit the scope of exploratory or high-risk research, even when potential long-term benefits exist. Consequently, beyond the technical challenges posed by electric and autonomous vehicles, the interaction between academic openness and industrial constraints emerges as a relevant non-technical factor influencing

the pace of innovation in this field.

Recognizing these limitations provides important context for the contributions of this dissertation and underlines the value of research approaches that seek to bridge academic rigor and industrial applicability.

Conclusiones Generales

En esta tesis doctoral se ha abordado el problema del modelado y control de vehículos eléctricos autónomos desde una perspectiva académica e industrial. La creciente adopción de la propulsión eléctrica y de los sistemas avanzados de asistencia a la conducción ha desplazado el foco de la industria automotriz hacia los actuadores eléctricos empleados en la dirección y la propulsión, convirtiendo su modelado y control precisos en un aspecto crítico para la seguridad y el desempeño del vehículo.

Una revisión de la literatura revela un compromiso entre la sofisticación del control y la fidelidad del modelo. Muchos estudios aplican técnicas avanzadas de control no lineal, como el control por modos deslizantes, utilizando modelos de vehículo muy simplificados, mientras que otros emplean modelos complejos pero adoptan estrategias de control básicas y omiten la dinámica de los actuadores. Estas inconsistencias limitan la transferibilidad de los resultados a vehículos reales y reducen su aplicabilidad industrial.

Esta investigación se distingue por el desarrollo y uso de modelos de vehículo más completos para el diseño y validación de los controladores, incorporando explícitamente la dinámica de los actuadores asociados a las máquinas eléctricas. Las simplificaciones comúnmente introducidas durante la simulación se evitan deliberadamente para reflejar un comportamiento más realista del vehículo. Las estrategias de control propuestas se diseñan y evalúan empleando estos modelos, permitiendo una evaluación más fiable del desempeño, la robustez y la viabilidad práctica.

Entre los resultados más relevantes obtenidos a lo largo de esta investigación se destacan los siguientes:

1. Tres controladores teóricos para la dinámica lateral basados en control por modos deslizantes (SMC): El primero es un controlador basado en SMC convencional que convierte el modelo dinámico a una forma compañera controlable. El segundo y el tercero combinan la técnica de

control por bloques para compensar dinámicas conocidas e incorporan un SMC super–twisting para rechazar perturbaciones o dinámicas desconocidas.

2. Desarrollo de un entorno de pruebas: Como resultado de esta investigación se desarrolló un entorno de simulación ejecutable que incluye la dinámica lateral y longitudinal del vehículo, así como la dinámica de los neumáticos y de los actuadores, como motores eléctricos DC y BLDC. El modelo permite generar rutas personalizadas o importar archivos de puntos de paso, así como modificar los parámetros para adaptarlo a diferentes tipos de vehículos.

Como trabajo futuro, se contempla continuar mejorando los modelos físicos e implementar estrategias de control como modos deslizantes adaptativos o controles predictivos. Asimismo, se prevé el desarrollo de un marco de evaluación que permita analizar los controladores desde la perspectiva de normas de seguridad estandarizadas, tales como ISO 26262 y ISO 21448.

Como reflexión final, este trabajo pone de manifiesto la divergencia que con frecuencia existe entre la investigación académica y la práctica industrial. Mientras que la academia busca principalmente generar y difundir conocimiento, la investigación industrial está guiada por la competitividad, las restricciones de tiempo de comercialización y la protección de la propiedad intelectual. Como consecuencia, los datos relevantes y la información técnica necesaria para la investigación aplicada suelen estar sujetos a confidencialidad.

En el ámbito automotriz, esta limitación es especialmente evidente en la disponibilidad de parámetros realistas del sistema. Datos como las características de los neumáticos, los parámetros de los motores eléctricos, las especificaciones detalladas del vehículo y los límites de desempeño de los actuadores suelen ser inaccesibles o solo parcialmente disponibles fuera de los entornos industriales. Esta falta de transparencia representa un desafío para la investigación académica orientada al desarrollo y validación de modelos y estrategias de control con relevancia industrial.

De manera paralela, las actividades de investigación industrial están condicionadas por los costos de desarrollo y por la necesidad de justificar las inversiones mediante un retorno claro o de corto plazo. Estas restricciones limitan naturalmente el alcance de la investigación exploratoria o de alto riesgo, incluso cuando existen beneficios potenciales a largo plazo. En consecuencia, además de los

desafíos técnicos de los vehículos eléctricos y autónomos, la interacción entre la apertura académica y las restricciones industriales surge como un factor no técnico que influye en el ritmo de innovación en este campo.

Reconocer estas limitaciones proporciona un contexto importante para las contribuciones de esta tesis y subraya el valor de enfoques de investigación que buscan tender puentes entre el rigor académico y la aplicabilidad industrial.

Appendix

A. List of Internal Research Reports

- L. A. Torres-Romero, L. E. González-Jiménez, and R. Ruiz-Cruz, "A brief review on industrial standards and procedures for safety related automotive systems," Internal Report PhDEngScITESO-18-35-R, ITESO, Tlaquepaque, Mexico, Dec. 2018.
- L. A. Torres-Romero, L. E. González-Jiménez, and R. Ruiz-Cruz, "Introduction to physical automotive models and modern controllers," Internal Report PhDEngScITESO-19-05-R, ITESO, Tlaquepaque, Mexico, Dec. 2019.
- L. A. Torres-Romero, L. E. González-Jiménez, and R. Ruiz-Cruz, "Lateral vehicle dynamics and controllers design," Internal Report PhDEngScITESO-19-21-R, ITESO, Tlaquepaque, Mexico, Dec. 2019.
- L. A. Torres-Romero, L. E. González-Jiménez, and R. Ruiz-Cruz, "Longitudinal vehicle dynamics," Internal Report PhDEngScITESO-20-02-R, ITESO, Tlaquepaque, Mexico, Mar. 2020.
- L. A. Torres-Romero, L. E. González-Jiménez, and R. Ruiz-Cruz, "DC electric machines review and modeling for electric vehicles," Internal Report PhDEngScITESO-20-13-R, ITESO, Tlaquepaque, Mexico, Aug. 2020.
- L. A. Torres-Romero, L. E. González-Jiménez, and R. Ruiz-Cruz, "Longitudinal and lateral dynamic model of a vehicle using a electrical motor for power and waypoints for lateral control," Internal Report PhDEngScITESO-21-08-R, ITESO, Tlaquepaque, Mexico, Sep. 2021.
- L. A. Torres-Romero, L. E. González-Jiménez, and R. Ruiz-Cruz, "Electric machines for electric vehicles," Internal Report PhDEngScITESO-21-27-R, ITESO, Tlaquepaque, Mexico, Dec. 2021.
- L. A. Torres-Romero, L. E. González-Jiménez, and R. Ruiz-Cruz, "Space vector pwm for bldc motors," Internal Report PhDEngScITESO-22-03-R, ITESO, Tlaquepaque, Mexico, Mar. 2022.

- L. A. Torres-Romero, L. E. González-Jiménez, and R. Ruiz-Cruz, "PID control for the traction and steering of an electric vehicle considering BLDC motors as actuators," Internal Report PhDEngScITESO-22-07-R, ITESO, Tlaquepaque, Mexico, Jul. 2022.

- L. A. Torres-Romero, and L. E. González-Jiménez, "Brief analysis of the control design from safety of the intended functionality (SOTIF) perspective," Internal Report PhDEngScITESO-26-xx-R, ITESO, Tlaquepaque, Mexico, Mar. 2026.

- L. A. Torres-Romero, and L. E. González-Jiménez, "PID control for the traction and steering of an electric vehicle considering BLDC motors as actuators," Internal Report PhDEngScITESO-26-xx-R, ITESO, Tlaquepaque, Mexico, Mar. 2026.

- L. A. Torres-Romero, and L. E. González-Jiménez, "Performance comparison of controllers for lateral vehicle dynamics with BLDC as actuator for steering," Internal Report PhDEngScITESO-26-xx-R, ITESO, Tlaquepaque, Mexico, Mar. 2026.

- L. A. Torres-Romero, and L. E. González-Jiménez, "Vehicle dynamics and control simulation playground: the user manual," Internal Report PhDEngScITESO-26-xx-R, ITESO, Tlaquepaque, Mexico, Mar. 2026.

B. List of Publications

- L. Torres-Romero, R. Ruiz-Cruz, and L. González-Jiménez, ”Path-Following Sliding Mode Controller for an Electric Vehicle Considering Actuator Dynamics,” Machines Vol. 12, no. 4, p. 219, 2024.

- L. A. Torres-Romero and L. E. González-Jiménez, ”A Comparison of Sliding Mode Controllers for the Lateral Dynamics of an Electric Autonomous Vehicle,” in Proc. 2025 IEEE Int. Autumn Meeting on Power, Electronics and Computing (ROPEC), Ixtapa, Mexico, 2025, pp. 1–6, doi:10.1109/ROPEC68163.2025.11354035, ISBN: 979-8-3315-7673-8, ISSN: 2573-0770.

- L. A. Torres-Romero and L. E. González-Jiménez, ”A Block Controller with Integral Super-Twisting Algorithm for the Path Following of a Self-Driving Electric Vehicle Considering Actuator Dynamics,” World Electric Vehicle Journal, vol. 16, no. 12, p. 643, 2025. doi: 10.3390/wevj16120643

C. Source Code

C.1 Code for Section 02.01

Algorithm 1 RegularForm.mlx

```
format rational
syms A B Q R Tr
A = [1 1 1;0 1 3;1 0 1]
B = [0 1;1 -1;-1 0]
[Q R] = qr(sym(B))
Tr= flipud(Q') % we need the cero part in the superior part

B_ = Tr*B
A_ = Tr*A*inv(Tr)
% A_ matrix partition
A_11 = A_(1,1)
A_12 = A_(1,2:3)
A_21 = A_(2:3,1)
A_22 = A_(2:3,2:3)
```

Algorithm 2 LateralDynaRF_1.mlx

```
format rational
syms A B Q R Tr B1_2 B1_4 A22 A23 A24 A42 A43 A44
A = [0 1 0 0;0 A22 A23 A24;0 0 0 1;0 A42 A43 A44]
B = [0 0;B1_2 0;0 0;0 B1_4]
[Q R] = qr(sym(B))
Tr= flipud(Q') % we need the cero part in the superior part
rank(Tr)
B_ = Tr*B
A_ = Tr*A*inv(Tr)
% A_ matrix partition
A_11 = A_(1:2,1:2)
A_12 = A_(1:2,3:4)
A_21 = A_(3:4,1:2)
A_22 = A_(3:4,3:4)
syms M M11 M12 M21 M22 lambda_1 lambda_2 lambda
M = [M11 M12;M21 M22];
%%%
```

Algorithm 3 LateralDynaRF_2.mlx

```
%%%
```

Algorithm 4 LevantDiff.m

```
%************************************************
% Author: L.Arturo Torres-Romero
% based on
% Levant A. (1998). Robust exact differentiation via sliding mode
    technique.
% Automatica, 34(3), 379-384
%************************************************
function LevantDiff()
    clear all;
    dt = 0.01;
    [t, X] = ode45(@LevantDiffOde,[0:dt:15],[0 0]);

    figure;
    plot(t,X(:,2),'--',t,cos(t),'-.',t,sin(t),'-'); title('
       Differential comparison');
    legend('Levant Diff', 'Diff = cos(t)', 'Original = sin(t)', '
       Location', 'best');
end

%***************************************
% Levant differentiator
% ***************************************
function dX = LevantDiffOde(t,X)
    z_0=X(1);    z_1= X(2);

     %Diff Gains
     du_ref=z_1;
     lambda_u0=20;   lambda_u1=10;
     u_ref = sin(t);

    %Levant Differentiator
     d_z_0=-lambda_u0*sqrt(abs(z_0-u_ref))*sign(z_0-u_ref)+z_1;
     d_z_1=-lambda_u1*sign(z_1-d_z_0);

    dX = [d_z_0;d_z_1];
end
```

C.2 Code for Section 02.02

Algorithm 5 DCMotor.m

```
% L.Arturo Torres-Romero
% DC Motor simulation
function dX = fcn(X,u)
    ia = X(1); wm= X(2);
    Va = u(1); Tl = u(2);

    Ra = 0.193;       % Is the equivalent resistance.
    La = 3.83e-3;     % Is the inductance of the coil.
    Jm = 0.6;         % kg/m^2 is the moment of inertia or
                      % rotational inertia.
    bm = 2.632177;    % Damping torque coefficient.
    Kb = 2.332232;    % Angular velocity-Back EMF coefficent.
    KT = 2.1717;      % Current-torque coefficient.

    A = [-Ra/La -Kb/La;KT/Jm -bm/Jm];
    B = [1/La 0;0 -1/Jm];

    dX = A*X + B*u;
end
```

Algorithm 6 BLDCMotor.m

```
% L.Arturo Torres-Romero
% BLDC model
function dX = fcn(X,f,v,Tl)
    u = [v.*f;Tl];
    i_a = X(1); i_b = X(2); i_c = X(3); wm = X(4); theta = X(5);

    R = 0.7;          %hase resistance.
    L = 0.00383;      %Phase self-inductance.
    M = 0.003;        %Phase Mutual inductance.
    J = 0.6;          % Rotor inertia.
    P = 4;            % Number of poles.
    b = 0.0632177;    % Friction coefficient.
    lambda = 0.9;     %is the Back EMF constant in v/rad/s.

    A =[-(R/(L-M)) 0 0 -(lambda/(L-M))*f(1) 0;...
        0 -(R/(L-M)) 0 -(lambda/(L-M))*f(2) 0;...
        0 0 -(R/(L-M)) -(lambda/(L-M))*f(3) 0;...
        (lambda/J)*f(1) (lambda/J)*f(2) (lambda/J)*f(3) -b/J
           0;...
        0 0 0 P/2 0];

    B = diag([1/(L-M) 1/(L-M) 1/(L-M) -1/J]);
    B = [B;0 0 0 0];

    dX = A*X + B*u;   % model in the state-space
end
```

Algorithm 7 BLDCMotorFs.m

```
% L.Arturo Torres-Romero
% BLDC trapezoidal shape functions
function Fabc = fcn(X)
    Fabc = [0;0;0];
    theta = mod(X(5),2*pi);
    if (0<=theta)&&(theta<(pi/6))
        Fa = (6/pi)*theta;
    elseif ((pi/6)<=theta)&&(theta<(5*pi/6))
        Fa = 1;
    elseif ((5*pi/6<=theta))&&(theta<(7*pi/6))
        Fa = (6/pi)*(pi-theta);
    elseif ((7*pi/6)<=theta)&&(theta<(11*pi/6))
        Fa = -1;
    elseif ((11*pi/6)<=theta)&&(theta<(2*pi))
        Fa = (6/pi)*(theta-2*pi);
    else
        Fa = NaN;
    end
    if (0<=theta)&&(theta<(3*pi/6))
        Fb = -1;
    elseif ((3*pi/6)<=theta)&&(theta<(5*pi/6))
        Fb = (6/pi)*(theta-(2*pi)/3);
    elseif ((5*pi/6)<=theta)&&(theta<(9*pi/6))
        Fb = 1;
    elseif ((9*pi/6)<=theta)&&(theta<(11*pi/6))
        Fb = (6/pi)*(-theta+(5*pi)/3);
    elseif ((11*pi/6)<=theta)&&(theta<(2*pi))
        Fb = -1;
    else
        Fb = NaN;
    end
    if (0<=theta)&&(theta<(pi/6))
        Fc = 1;
    elseif ((pi/6)<=theta)&&(theta<(3*pi/6))
        Fc = (6/pi)*(pi/3-theta);
    elseif ((3*pi/6)<=theta)&&(theta<(7*pi/6))
        Fc = -1;
    elseif ((7*pi/6)<=theta)&&(theta<(9*pi/6))
        Fc = (6/pi)*(theta-(4*pi)/3);
    elseif ((9*pi/6)<=theta)&&(theta<(2*pi))
        Fc = 1;
    else
        Fc = NaN;
    end
    Fabc = [Fa;Fb;Fc];
end
```

C.3 Code for Section 02.03

Algorithm 8 PID_Controller.c

```
#define SAMPLING_TIME 0.001

float Integrate(float X)
{
 static float integral = 0;
 integral += X*SAMPLING_TIME;
 return integral;
}

float derivate(float X)
{
  static float prevX = 0;
  float derivative = 0;
  derivative = (X - prevX)/SAMPLING_TIME;
  prevX = X;
  return derivative;
}

float PID_Control(float error, float Kp, float Ki, float Kd)
{
  float ctrl = 0;
  ctrl = Kp*error + Ki*Integrate(error) + Kd*derivate(error);
  return ctrl;
}
```

Algorithm 9 dqBLDCModel.m

```
% L.Arturo Torres-Romero
function dX = fcn(X,f,v,Tl)
i_d = X(1); i_q = X(2); wm = X(3); theta = X(4);
i_dq = X(1:2);

lambda = 0.0433; %
R = 0.35;     %ohms armature resistance
L = 0.001;  %H armature inductance
M = 0.0000;  %H mutual inductance
J = 0.00018; % kgm^2 motor inertia
P = 4; % number of poles
b = 0.0034;     %Nms Hydraulic friction coefficient of motor rotor


M_LR = [-R/(L-M)  wm; -wm  -R/(L-M)];

% Park clarke transform eliminating the third element
PC_Te = (sqrt(2)/2)*[sqrt(3)*cos(theta)  sin(theta)  -sin(theta)
   ; ...
                     -sqrt(3)*sin(theta)  cos(theta)  -cos(theta)
                        ];

PC_T = PC_Te;

e_dq =(lambda*wm)*PC_Te*f;

v_dq = v;

di_dq = M_LR*i_dq + (1/(L-M))*v_dq - (1/(L-M))*e_dq;
T_e = (e_dq'*i_dq)/wm;
dwm = (T_e - Tl -b*wm)/J;
dtheta = (P/2)*wm;

dX = [di_dq;dwm;dtheta];

end
```

Algorithm 10 InvParkClarke.m

```
function Y_abc = fcn(X_dq, theta)
X_dq0 = [X_dq; 0];

%inverse Park-Clarke power invariant
iT_PC = sqrt(2/3)*[cos(theta)          -sin(theta)            1/
   sqrt(2); ...
                   cos(theta-2*pi/3)   -sin(theta-2*pi/3)     1/
                      sqrt(2); ...
                   cos(theta+2*pi/3)   -sin(theta+2*pi/3)     1/
                      sqrt(2)];
%

 %inverse Park-Clarke amplitude invariant
% iT_PC = [cos(theta)          -sin(theta)            1; ...
%          cos(theta-2*pi/3)   -sin(theta-2*pi/3)     1; ...
%          cos(theta+2*pi/3)   -sin(theta+2*pi/3)     1];



Y_abc =iT_PC*X_dq0;
end
```

Algorithm 11 ParkClarke.m

```
function Y_dq = fcn(X_abc, theta)

% Park clarke transform eliminating the third element
PC_T = (sqrt(2)/2)*[sqrt(3)*cos(theta)  sin(theta)  -sin(theta) ;
    ...
                    -sqrt(3)*sin(theta)  cos(theta)  -cos(theta)
                       ];

Y_dq =PC_T*X_abc;
end
```

Algorithm 12 SpaceVectorCalculator.m

```
function [n,v,theta_vs]= SpaceVectorCalc(V_link,Vs,theta_vs1)
v = [0 0];
theta_vs = mod(theta_vs1,2*pi);
theta = 0;
n = 0; % This is the quadrant of the vector. you should never get
    zero as output

if 0*(pi/180) <= theta_vs && theta_vs < 60*(pi/180)
   theta = theta_vs;
   n = 1;
elseif 60*(pi/180) <= theta_vs && theta_vs < 120*(pi/180)
   theta = theta_vs - 60*(pi/180);
   n = 2;
elseif 120*(pi/180) <= theta_vs && theta_vs < 180*(pi/180)
    theta = theta_vs - 2*60*(pi/180);
    n = 3;
elseif 180*(pi/180) <= theta_vs && theta_vs < 240*(pi/180)
    theta = theta_vs - 3*60*(pi/180);
    n = 4;
elseif 240*(pi/180) <= theta_vs && theta_vs < 300*(pi/180)
    theta = theta_vs - 4*60*(pi/180);
    n = 5;
elseif 300*(pi/180) <= theta_vs && theta_vs < 360*(pi/180)
    theta = theta_vs - 5*60*(pi/180);
    n = 6;
end


V_b = ((2*Vs)/sqrt(3))*sin(theta);
V_a = ((2*Vs)/sqrt(3))*sin(pi/3 - theta);

v = [V_a V_b];
```

C.4 Code for Section 02.05

Algorithm 13 EigenValPlacement.mlx

```
format rational
%% Symbolic definitions
% System matrices are kept symbolic to derive a general sliding
   surface
syms A B Q R Tr B1_2 B1_4 A22 A23 A24 A42 A43 A44 real

%% Original system definition
% State-space representation x_dot = A x + B u
A = [0 1 0 0;
     0 A22 A23 A24;
     0 0 0 1;
     0 A42 A43 A44]

B = [0;
     B1_2;
     0;
     B1_4]

disp("Original system matrices A and B defined symbolically")

%% QR decomposition of input matrix
% QR decomposition is used to construct a transformation
% that brings the system into regular form
[Q R] = qr(sym(B));

%% Coordinate transformation matrix
% The transformation matrix Tr is built from Q
% flip() is used to place the zero dynamics in the upper block
Tr = flip(simplify(Q)')
disp("Transformation matrix Tr constructed using QR decomposition
   ")

rank(Tr)

%% Transformed input matrix
% After transformation, B_ should have a specific structure:
% zeros in the upper rows and a non-zero term in the last row
B_ = Tr*B;
B_ = simplify(B_)
disp("Transformed input matrix B_ (regular form)")
```

Algorithm 14 EigenValPlacement2.mlx

```
%% Transformed system matrix
% A is transformed to the regular form using Tr
A_ = Tr*A*inv(Tr);
A_ = simplify(A_)
disp("Transformed system matrix A_ in regular form")

%% Partition of A_ matrix
% Regular form partition:
% x1_dot = A11 x1 + A12 x2
% x2_dot = A21 x1 + A22 x2 + B2 u
A_11 = A_(1:3,1:3)
A_12 = A_(1:3,4)
A_21 = A_(4:4,1:3)
A_22 = A_(4,4)

disp("A_ matrix partitioned into (A11, A12, A21, A22)")

%% Sliding surface parameterization
% Sliding surface is defined as:
% sigma = M x1 + x2
syms M M1 M2 M3 lambda_1 lambda_2 lambda_3 lambda
M = [M1 M2 M3];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Eigenvalue placement via polynomial comparison
% The reduced-order dynamics on the sliding surface is:
% x1_dot = (A11 - A12*M) x1
disp("Computing characteristic polynomial of sliding dynamics")

eqeigen = det(lambda*eye(3) - (A_11 - A_12*M));
eqeigen = simplify(eqeigen)
eqeigen = collect(eqeigen,lambda)

polyCoef = coeffs(eqeigen, lambda);
polyCoef = fliplr(polyCoef)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Desired characteristic polynomial
% Desired closed-loop dynamics on the sliding surface
disp("Defining desired characteristic polynomial")

desiredPoly = (lambda + lambda_1)*(lambda + lambda_2)*(lambda +
   lambda_3)
desiredPoly = collect(desiredPoly)
```

Algorithm 15 EigenValPlacement3.mlx

```
desiredCoefs = coeffs(desiredPoly, lambda);
desiredCoefs = fliplr(desiredCoefs)

%% Polynomial coefficient matching
% Coefficients are matched to solve for M
eq1 = desiredCoefs(1) == polyCoef(1);
eq2 = desiredCoefs(2) == polyCoef(2);
eq3 = desiredCoefs(3) == polyCoef(3);

system1 = [eq1;eq2;eq3];

sol = solve(system1,[M1 M2 M3]);
M_ = [sol.M1(1) sol.M2(1) sol.M3(1)]

disp("Sliding surface gain vector M obtained via pole placement")

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Sliding surface construction in original coordinates
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Final sliding surface:
% sigma = S x
S2 = eye(1);
S = S2*[M_ eye(1)]*Tr;

disp("Sliding surface matrix S:")
expand(S)

%% State and input symbols
syms n sigma_ x1 x2 x3 x4 u1 u2 sigma1 sigma2 real
X = [x1;x2;x3;x4];
U = [u1];

%% Sliding variable definition
SIGMA = S*X;
Sigma_1 = collect(simplify(SIGMA(1)),[x1;x2;x3;x4])

disp("Sliding variable sigma expressed in original state
   coordinates")

%% Reaching law
% Standard reaching condition:
% sigma_dot = -n * sign(sigma)
eqsolution = [-n*sign(sigma1)] == S*(A*X + B*U);
```

Algorithm 16 EigenValPlacement4.mlx

```
%% Solve for control input u
sol1 = solve(eqsolution(1),u1);
finalControl = simplify(sol1);
disp("!!!!! Final sliding mode control law u:")
finalControl = collect(finalControl,[x1;x2;x3;x4])
```

C.5 Code for Section 03.01

Algorithm 17 PureKinematicModel.m

```
function PureKinematicModel()
clear all;
[t, X] = ode45(@bicycleModelOde,[0:.01:30],[5 1 pi/20 0] );
figure('NumberTitle', 'off', 'Name', 'Kinematic model');
subplot(2,1,1);plot(t,X(:,4)); title('Steering wheel angle in
   rads');
subplot(2,1,2);plot(X(:,1),X(:,2));title('Trajectory XY of the
   vehicle');
end


function dX = bicycleModelOde(t,X)
dX = [0;0;0];
lf = 1;
lr = 1;
U = X(4);

dU = SteeringWheelRate(t);

V = 8;
Beta = atan(lr*tan(U)/(lr+lf));

dPhi = V*cos(Beta)*tan(U)/(lf + lr);

dX(1) = V*cos(X(3) + Beta); % X(x) = Phi
dX(2) = V*sin(X(3) + Beta); % X(x) = Phi
dX(3) = dPhi;
dX(4) = dU;
end


function  dU = SteeringWheelRate(t)
if t <= 5
  dU =  0.8*(pi/180); % half degree per second
else
    if t < 10
        dU= -1.6*(pi/180); % half degree per second
    else
        dU = 0;
    end
end

end
```

Algorithm 18 KinematicPsiCtrl_eFBK_solution.m

```
%sfind velocity in terms of wheel speed.
% l = lr + lf
% Wa Wc angular velocities of points C and A
% Ra Rc   distance OA OC
% Ww angular velocity of wheel
% r radius of wheel
% d angle of wheel delta
% V velocity of C.G.
% B slip angle beta
syms l Wa Wc Ra Rc Ww r d V B

B = atan((1/2)*tan(d));
Ra = (l*cos(B))/(2*sin(d-B));

%first version of V
%V = (l*Ww*r)/(Ra*tan(d)*cos(B));

% Second version of V
%V = l*Ww*r*sin(d)/(2*sin(B));
%V = simplify(V);

syms K e Psi_ref Psi t

dPsi = (V*cos(B)*tan(d))/l;

Psi_ref = (pi/4)*sin(t/20);
%Psi_ref = t/20;
dPsi_ref = diff(Psi_ref);

%e = Psi_ref - Psi;
de = dPsi_ref - dPsi;

f = K*e == de%  modo normal
%f = K*sign(e) == de;% modos deslizantes

solve(f,d)
```

Algorithm 19 DynaBicycleError.m part 1

```
function DynaBicycleError()

[t, X] = ode45(@bicycleModelOde,[0:.01:50],[0 0 0 0]);
figure;
subplot(1,2,1);plot(t,X(:,1));title('Lateral position error');
subplot(1,2,2);plot(t,X(:,3));title('Yaw angle error');
%subplot(3,1,3);plot(X(:,1),X(:,2));
%plot(X(:,1),X(:,2));
end
%*******************************************************
% BICYCLE MODEL OF LATERAL VEHICLE DYNAMICS
% Taken from Rajamani chapter 2.5 page 37
%           X(1) = e1       X(2) = de1      X(3) = e2        X(4)
    = de2
% e1 = lateral position error with respect road
% e2 = Yaw angle error  with respect road
function dX = bicycleModelOde(t,X) % pd polos deseados
dX = [0;0;0;0];
y = X(1);
dy = X(2);
Psi = X(3);
dPsi = X(4);
g = 9.81; % gravity
C_af = 1000; % cornering stiffness of front wheel
C_ar = 1000; % cornering stiffness of rear wheel
Vx = 12;
lf = 1.5;
lr = 1.5;
Iz = 3000; %momentum of inertia
m = 1000; % mass of the vehicle
R = 50; % radius of the curve of the road
d = 3.41*(pi/180);% (pi/50)*sin(t/5) - pi/100;     % angle of the
    orientation of the wheel related to the orientation of
   vehicle
phi = 0; %pi/50;
dPsi_des = Vx/R;
```

Algorithm 20 DynaBicycleError.m part 2

```
34 A22 = -(2*C_af + 2*C_ar)/(m*Vx);
35 A23 = (2*C_af + 2*C_ar)/m;
36 A24 = (-2*C_af*lf + 2*C_ar*lr)/(m*Vx);
37 A42 = -(2*C_af*lf - 2*C_ar*lr)/(Iz*Vx);
38 A43 = (2*C_af*lf - 2*C_ar*lr)/Iz;
39 A44 = -(2*C_af*lf^2 + 2*C_ar*lr^2)/(Iz*Vx);
40
41 B1_2 = 2*C_af/m;
42 B1_4 = 2*C_af*lf/Iz;
43
44 B2_2 = -(2*C_af*lf - 2*C_ar*lr)/(m*Vx);
45 B2_4 = -(2*C_af*lf^2 + 2*C_ar*lr^2)/(Iz*Vx);
46
47 A = [0 1 0 0;0 A22 A23 A24;0 0 0 1;0 A42 A43 A44];
48
49 B1 = [0;B1_2;0;B1_4];
50 B2 = [0;B2_2;0;B2_4];
51 B3 = [0;g;0;0];
52
53 dX = A*X + B1*d + B2*dPsi_des + B3*sin(phi);
54 end
```

Algorithm 21 StateFBK_Ctrl_lateralRoadError_XY_sim.m part 1

```
function StateFBK_Ctrl_lateralRoadError()
clear all;
[t, X] = ode45(@bicycleModelOde,[0:.01:20],[1 1 1 1 0 0 0 0 0]);
figure;
subplot(4,1,1);plot(t,X(:,1));
title('e1: lateral position error');
subplot(4,1,2);plot(t,X(:,2));
title('derivative of e1');
subplot(4,1,3);plot(t,X(:,3));
title('e2: Yaw angle error respect road');
subplot(4,1,4);plot(t,X(:,4));
title('derivative of e2');

figure;
a1 = plot(X(:,5),X(:,6),'red--');L1 = "System real behavior";
hold on;
a2 = plot(X(:,8),X(:,9),'k');L2 = "System Reference";
legend([a1,a2], [L1, L2]);
end

%*******************************************************
% BICYCLE MODEL OF LATERAL VEHICLE DYNAMICS
% Taken from Rajamani chapter 2.5 page 37
%           X(1) = e1       X(2) = de1      X(3) = e2        X(4)
    = de2
% e1 = lateral position error with respect road
% e2 = Yaw angle error  with respect road
function dX = bicycleModelOde(t,X) % pd polos deseados
dX = [0;0;0;0;0;0;0;0;0];
e1 =  X(1);
de1 = X(2);
e2 =  X(3);
de2 = X(4);
Psi_des = X(7);

g = 9.81; % gravity
C_af = 80000; % cornering stiffness of front wheel
C_ar = 80000; % cornering stiffness of rear wheel
Vx = 8;
lf = 1.1;
lr = 1.58;
Iz = 2873; %momentum of inertia
m = 1573; % mass of the vehicle
R = 1000; % radius of the curve of the road
phi = 0;% pi/50;
```

Algorithm 22 StateFBK_Ctrl_lateralRoadError_XY_sim.m part 2

```
46 if t > 1  % the first second the street is straight, after that a
       curve start
47     dPsi_des = Vx/R;
48 else
49     dPsi_des = 0;
50 end
51
52 A22 = -(2*C_af + 2*C_ar)/(m*Vx);
53 A23 = (2*C_af + 2*C_ar)/m;
54 A24 = (-2*C_af*lf + 2*C_ar*lr)/(m*Vx);
55 A42 = -(2*C_af*lf - 2*C_ar*lr)/(Iz*Vx);
56 A43 = (2*C_af*lf - 2*C_ar*lr)/Iz;
57 A44 = -(2*C_af*lf^2 + 2*C_ar*lr^2)/(Iz*Vx);
58
59 B1_2 = 2*C_af/m;
60 B1_4 = 2*C_af*lf/Iz;
61 B2_2 = -(2*C_af*lf - 2*C_ar*lr)/(m*Vx) - Vx;
62 B2_4 = -(2*C_af*lf^2 + 2*C_ar*lr^2)/(Iz*Vx);
63 A = [0 1 0 0;0 A22 A23 A24;0 0 0 1;0 A42 A43 A44];
64 B1 = [0;B1_2;0;B1_4];
65 B2 = [0;B2_2;0;B2_4];
66 B3 = [0;g;0;0];
67
68 % is it controlable?
69 % Controlability matrix
70 Mc = [B1 A*B1 A^2*B1 A^3*B1];
71 rankVal = rank(Mc);
72
73 % Pole placement
74 Pd =[-5-3i -5+3i -7 -10];
75 %Pd =[-2 -2 -2 -2];
76
77 H = (A- Pd(1)*eye(4)) * (A- Pd(2)*eye(4)) * (A- Pd(3)*eye(4)) * (
       A- Pd(4)*eye(4));
78
79 %ackerman formula
80 K = [0 0 0 1]*Mc^-1*H;
```

Algorithm 23 StateFBK_Ctrl_lateralRoadError_XY_sim.m part 3

```
82 %Global Coordinates calculation
83 %V = sqrt(Vx^2+de1^2);
84 G_dX_des = Vx*cos(Psi_des);
85 G_dY_des = Vx*sin(Psi_des);
86 G_dX = G_dX_des  - (de1*sin(e2 + Psi_des ) + e1*cos(e2 + Psi_des)
      *(de2 + dPsi_des));
87 G_dY = G_dY_des  + (de1*cos(e2 + Psi_des ) - e1*sin(e2 + Psi_des)
      *(de2 + dPsi_des));
88
89 U = -K*X(1:4);
90 dX(1:4) = A*X(1:4) + B1*U + B2*dPsi_des + B3*sin(phi);
91
92 dX(5) = G_dX;
93 dX(6) = G_dY;
94 dX(7) = dPsi_des;
95 dX(8) = G_dX_des;
96 dX(9) = G_dY_des;
97 end
```

Now, a brief explanation of the code:

- **Lines 1–3**: The function clears any existing variables and then simulates the dynamic bicycle model using `ode45`. The simulation runs over a time interval from 0 to 20 seconds, with time steps of 0.01 seconds. The initial conditions include lateral position error, yaw angle error, and their derivatives.
- **Lines 4–12**: The code generates the first figure, plotting four subplots:
  - `X(:,1)` represents the lateral position error (e_1).
  - `X(:,2)` shows the derivative of e_1.
  - `X(:,3)` represents the yaw angle error (e_2).
  - `X(:,4)` shows the derivative of e_2.
- **Lines 14–18**: The second figure compares the vehicle's actual trajectory (`X(:,5)` vs `X(:,6)`) with the reference trajectory (`X(:,8)` vs `X(:,9)`). A red dashed line represents the actual trajectory, and a black solid line shows the reference. A legend is added to distinguish the two.

Now, the main part, **bicycleModelOde** is explained:

- **Line 27**: The `bicycleModelOde` function defines the dynamic bicycle model for the vehicle's lateral dynamics. The output `dX` is a 9-element vector that holds the derivatives of the state variables.
- **Lines 28–33**: The state derivative vector `dX` is initialized, and key state variables are extracted from the input vector `X`, such as lateral position error (e_1), its derivative ($\dot{e}_1$), yaw angle error (e_2), and its derivative ($\dot{e}_2$).
- **Lines 35–44**: Vehicle and road parameters are defined. These include the cornering stiffness of the tires, vehicle speed (V_x), distances from the center of gravity to the front and rear axles (l_f, l_r), mass, and yaw inertia of the vehicle.
- **Lines 46–50**: The desired yaw rate ($\dot{\Psi}_{des}$) is calculated. For the first second, the road is straight, so $\dot{\Psi}_{des}$ is set to zero. After 1 second, the vehicle enters a curve, and the desired yaw rate is based on the radius of the road's curve (R).
- **Lines 52–66**: The state-space matrices A, B_1, B_2, and B_3 are defined. These matrices represent the linearized system dynamics, which capture the relationships between lateral and yaw dynamics as well as external inputs like steering.
- **Lines 70–71**: The controllability matrix M_c is calculated, and its rank is checked to ensure the system is fully controllable.
- **Lines 74–80**: The Ackerman formula is used to compute the state feedback gain K. Pole placement assigns the poles of the closed-loop system to specific values (P_d), ensuring the system's desired stability.
- **Lines 84–87**: Global coordinates ($G_{\dot{X}}, G_{\dot{Y}}$) are calculated for both the actual vehicle trajectory and the reference trajectory. These global coordinates are used to compare the vehicle's actual behavior with the reference.
- **Lines 89–90**: The control input U is computed using the feedback control law ($U = -KX(1:4)$), and the state derivatives $dX(1:4)$ are updated based on the state-space equations.
- **Lines 92–96**: Global positions (`dX(5)`, `dX(6)`) and the desired yaw rate (`dX(7)`) are updated for comparison with the reference trajectory (`dX(8)`, `dX(9)`).

C.6 Code for Section 03.02

Algorithm 24 GetReference.m

```
function [e1,de1,e2,de2,e3,Vx_reff,dPsi_des,phi_ref] = GetReff(X,
   Y,racetrackwaypoints, Psi,dPsi, Vy, Vx)

PQ = [X,Y]; % current Intertial position of vehicle
P = racetrackwaypoints(:,1:2);
[k,D] = knnsearch(P,PQ);
k = k(1);
P_road1 = P(k-1,:);
P_road2 = P(k,:);
P_road3 = P(k+1,:);
Vx_reff = racetrackwaypoints(k,3);
Vx_reff = Vx_reff(1);
LookAhead = 5;
F = atan2( (P(k+LookAhead,2)-P(k,2)),(P(k+LookAhead,1)-P(k,1)));
psi_des = F(1);
P_veh = PQ - P_road2;
M_Rot = [cos(psi_des) -sin(psi_des); sin(psi_des) cos(psi_des)];
P_veh = P_veh*M_Rot;

e1 = P_veh(2);
e2 = Psi - psi_des; % error e2
de1 = Vy + Vx*(e2); % derivative of e1
e3 = P_veh(1);

% Obtainig the radius of imaginary circle of the curve
Xc = [P_road1(1);P_road2(1);P_road3(1)];
Yc = [P_road1(2);P_road2(2);P_road3(2)];
[xc,yc,R,a] = circfit(Xc,Yc);
C = [xc yc];
C = (C-P_road2)*M_Rot;
xc = C(1);
yc = C(2);
R = R*sign(yc);

eval = ~isfinite(R);
if eval
    R = 100000;
end

dPsi_des = Vx/R; % derivative of orientation refference
de2 = dPsi - dPsi_des; % derivative of e2

% sueperelevation of the road or banking
phi_ref = atan(Vx^2/(127*R));
```

Here, is an explanation of the source code for GetReference block:

- **Line 1**: Defines the function `GetReff`, which calculates the lateral, longitudinal, and yaw errors of a vehicle with respect to a reference path, as well as reference values for velocity and orientation needed for the vehicle's steering controller.
- **Line 3**: Sets `PQ` to the current position `[X, Y]` of the vehicle's center of gravity in the global (inertial) coordinate system.
- **Line 4**: Extracts only the X and Y columns from `racetrackwaypoints` into `P`, representing the coordinates of the track's waypoints.
- **Line 5**: Finds the closest waypoint to the vehicle's position using `knnsearch`, where `k` is the index of this closest point, and `D` is the distance to it.
- **Line 6**: Ensures `k` only keeps the first (closest) value if multiple results exist.
- **Lines 7-9**: Defines three consecutive waypoints near the vehicle: `P_road1` (the waypoint behind), `P_road2` (the closest waypoint), and `P_road3` (the waypoint in front), setting up for accurate calculations of curvature and orientation.
- **Lines 10-11**: Extracts the reference longitudinal velocity `Vx_reff` at the closest waypoint from `racetrackwaypoints`, selecting the first element to confirm the velocity value.
- **Line 12**: Sets the look-ahead distance `LookAhead` to 5, defining how far forward to check when calculating the road's orientation.
- **Line 13**: Calculates the desired road orientation `psi_des` based on the angular difference between the coordinates of the current waypoint and the look-ahead waypoint, using the `atan2` function.
- **Line 14**: Assigns the desired yaw angle `psi_des` to `F(1)`, ensuring the orientation aligns with the road ahead.
- **Line 15**: Defines `P_veh` as the vector from the closest waypoint `P_road2` to the vehicle's current position `PQ`.
- **Lines 16-17**: Creates a rotation matrix `M_Rot` using `psi_des` to convert the position into the body-fixed frame, centering on the vehicle's C.G. and aligning with the yaw angle.
- **Line 18**: Applies the rotation matrix to `P_veh`, rotating the position into the body-fixed coordinates.

- **Line 19**: Defines `e1` as the y-component of `P_veh`, representing the lateral distance error (from the vehicle's C.G. to the reference path).
- **Line 20**: Defines `e2` as the yaw angle error, calculated as the difference between the current yaw angle `Psi` and the desired yaw angle `psi_des`.
- **Line 21**: Calculates `de1`, the derivative of `e1`, incorporating the lateral velocity `Vy` and yaw error `e2` scaled by the longitudinal velocity `Vx`.
- **Line 22**: Defines `e3` as the `x`-component of `P_veh`, representing the longitudinal distance error from the C.G. to the desired waypoint along the `x`-axis in the vehicle's frame.
- **Lines 25-26**: Extracts three consecutive X and Y coordinates `Xc` and `Yc` to fit a circle representing the road curvature near the vehicle.
- **Line 27**: Applies the `circfit` function to calculate the circle's center `C = [xc, yc]` and radius `R`, indicating the curvature of the road segment.
- **Line 29**: Shifts `C` by subtracting `P_road2` and rotates it into the vehicle's body-fixed frame using `M_Rot`.
- **Lines 30-31**: Assigns the rotated coordinates of `C` to `xc` and `yc`, and adjusts the radius `R` sign based on the y-component to indicate curvature direction.
- **Line 34**: Uses `isfinite` to detect invalid values in `R`; if `R` is invalid, assigns a large value `100000` to approximate a straight line.
- **Line 38**: Computes the desired yaw rate `dPsi_des` by dividing the longitudinal velocity `Vx` by the curvature radius `R`, setting the reference angular velocity.
- **Line 39**: Calculates the derivative of `e2` as `de2`, defined as the difference between the current yaw rate `dPsi` and the desired yaw rate `dPsi_des`.
- **Line 42**: Calculates the road's superelevation angle or banking `phi_ref` using the formula $\arctan\left(\frac{Vx^2}{127R}\right)$, which incorporates the curvature radius and longitudinal velocity squared.

Algorithm 25 RotateVector.m

```
1 function B = RotateVector(A, theta)
2 B = [0;0];
3 R = [cos(theta) -sin(theta); sin(theta) cos(theta)];
4
5 B = R*A;
```

Algorithm 26 LongitudinalDynamics.m

```
function [dVx,F_zf, F_zr] = LongitudinalDynamics(F_xf,F_xr,Vx,
    rho, g, A_F, C_d, f, m, V_wind, theta, h_aero, h_gc, lf, lr,
    Nf, Nr)

F_aero = (1/2)*rho*C_d*A_F*(Vx+V_wind)^2;
dVx = (F_xf + F_xr - F_aero - m*g*sin(theta))/m;

F_zf = (-F_aero*h_aero -m*dVx*h_gc -m*g*h_gc*sin(theta) +m*g*lr*
    cos(theta))/(Nf*(lr+lf));
F_zr = ( F_aero*h_aero +m*dVx*h_gc +m*g*h_gc*sin(theta) +m*g*lf*
    cos(theta))/(Nr*(lr+lf));
```

Algorithm 27 Transmission.m

```
function [T_wheels, w_t]= fcn(T_t,w_w,R)

T_wheels = T_t/R;
w_t = w_w/R;
```

Algorithm 28 WheelsDynamic.m

```
function [dw_wf, r_eff]= fcn(T_wheels,F_xf, r_w, r_stat)

%r_w = 0.352; % radius of the tire. Pirelli Sottozero 3 P245/45
    WR19
%r_stat = 0.342; % static vertical deflection of the tire

alpha = acos(r_stat/r_w);
r_eff = (alpha*r_w)/alpha;

I_w = .9; % TODO define this
dw_wf = (T_wheels -(r_eff*F_xf))/I_w;
```

Algorithm 29 TireSlip.m

```
function s_xf = XTireSlip(Vx,w_wf,r_eff)

if Vx == 0
   s_xf = 0.1 * sign(w_wf);
else
   s_xf = (r_eff*w_wf -Vx)/abs(Vx); % acceleration
end

if s_xf > 1
   %s_xf = 1;
end
```

Algorithm 30 MagicFormula.m

```
function Y = XTireForce(X, F_zf)
% B stiffness factor
% C shape factor
% D peak value
% E curvature factor
% Sh horizontal shift
% Sv vertical shift

Sh = 0;% Sh horizontal shift
Sv = 0; %vertical shift

D = 3; % ym, maximum value Experimental from graph
C = 1.5;
B = 20;
E = 0;

x = X-Sh;
gamma = C*atan(B*x-E*(B*x-atan(B*x)));
y = F_zf*D*sin(gamma);
Y = (y - Sv);
```

Bibliography

[1] A. A. Ahmed, J. Santhosh, and F. W. Aldbea. “Vehicle Dynamics Modeling and Simulation with Control Using Single Track Model”. In: s.l., s.n. 2020, pp. 1–4.

[2] E. Alcala et al. “Comparison of two non-linear model-based control strategies for autonomous vehicles”. In: *Proceedings of the 24th Mediterranean Conference on Control and Automation (MED)*. Athens, Greece, June 2016, pp. 846–851.

[3] Panos J Antsaklis and Anthony N Michel. *Linear systems*. Vol. 8. Springer, 1997.

[4] Francisco Aparicio Izquierdo, Carlos Vera Alvarez, and Vicente Díaz López. “Teoría de los vehículos automóviles”. In: *Madrid. Universidad Politécnica de Madrid* (1995).

[5] ARSTECHNICA. *Motor technology from Model 3 helps Tesla boost Model S range 10%*. https://arstechnica.com/cars/2019/04/motor-technology-from-model-3-helps-tesla-boost-model-s-range-10/#:~:text=The%20Model%203%20debuted%20with,any%20windings%20or%20permanent%20magnets. [Online; accessed 15-August-2024]. 2019.

[6] S. Bacha et al. “Modeling and Control Techniques for Autonomous Electric and Hybrid Vehicles Path Following”. In: *Proceedings of the 2017 5th International Conference on Electrical Engineering - Boumerdes (ICEE-B)*. Boumerdes, Algeria, Oct. 2017, pp. 1–12.

[7] S. Bei et al. “Research on the Trajectory Tracking of Adaptive Second-Order Sliding Mode Control Based on Super-Twisting”. In: *World Electric Vehicle Journal* 13.141 (2022). DOI: 10.3390/wevj13080141.

[8] W. C. Bolton. *Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering*. UK: Pearson Education, 2018.

[9] B. K. Bose. *Modern power electronics & AC drives*. Prentice Hall, 2002.

[10] Bimal K Bose et al. *Modern power electronics and AC drives*. Vol. 123. Prentice hall Upper Saddle River, NJ, 2002.

[11] A Brouri et al. “Modelling and identification of switched reluctance machine inductance”. In: *Australian Journal of Electrical and Electronics Engineering* 18.1 (2021), pp. 8–20.

[12] CARLA. *Open-source simulator for autonomous driving research*. `http://carla.org`. Accessed on 10 November 2023.

[13] CarSim. *Mechanical Simulation*. `https://www.carsim.com/`. Accessed on 27 November 2023.

[14] R. De Castro et al. “Minimum-Time Path-Following for Highly Redundant Electric Vehicles”. In: *IEEE Transactions on Control Systems Technology* 24.2 (2016).

[15] K.-C. Chen and H.-Y. Shih. “Research and simulation of the electrical vehicle based dynamical system”. In: *Mathematical Problems in Engineering* 2015 (2015).

[16] CNET. *Tesla’s Model 3 motor may strain world’s supply of neodymium*. `https://www.cnet.com/roadshow/news/teslas-model-3-motor-neodymium/`. [Online; accessed 15-August-2024]. 2018.

[17] J. M. D. Coey. “Perspective and Prospects for Rare Earth Permanent Magnets”. In: *Engineering* 6.2 (Feb. 2020), pp. 119–131. DOI: `10.1016/j.eng.2018.11.034`.

[18] R. M. Corless et al. “On the LambertW function”. In: *Advances in Computational Mathematics* 5 (1996), pp. 329–359.

[19] Mohammad Dika, Michel Owayjan, and Roger Achkar. “Steering control of self-driving car using super-twisting sliding mode control”. In: *2023 Fifth international conference on advances in computational tools for engineering applications (ACTEA)*. IEEE. 2023, pp. 190–196.

[20] Vehicle Dynamics and Controller Simulink Model. *Vehicle Dynamics and Controller Simulink Model*. `https://github.com/L-Arturo-Torres-Romero/VehicleModel/tree/master/VehicleModel`. Accessed on 24 October 2023.

[21] M. T. Emırler et al. “Lateral stability control of fully electric vehicles”. In: *International Journal of Automotive Technology* 16 (2015), pp. 317–328.

[22] F. M. Favarò et al. “Examining accident reports involving autonomous vehicles in California”. In: *PLoS ONE* 12.9 (Sept. 2017), e0184952.

[23] B. Fernandez, P. Herrera, and J. Cerrada. "A Simplified Optimal Path Following Controller for an Agricultural Skid-Steering Robot". In: *IEEE Access* 7 (2019), pp. 95932–95940.

[24] B. Fernandez, P. Herrera, and J. Cerrada. "A Simplified Optimal Path Following Controller for an Agricultural Skid-Steering Robot". In: *IEEE Access* 7 (2019), pp. 95932–95940.

[25] *Five slippery cars enter a wind tunnel; one slinks out a winner*. Available online. Accessed: 10 June 2025. URL: https://www.tesla.com/sites/default/files/blog_attachments/the-slipperiest-car-on-the-road.pdf.

[26] FORBES. *Continental AG Intends To Spin Off Vitesco Technologies, Shelves Plans for Partial IPO*. https://www.forbes.com/sites/joecornell/2019/10/30/continental-ag-intends-to-spin-off-vitesco-technologies-shelves-plans-for-partial-ipo/#76366a783dae. [Online; accessed 15-August-2024]. 2019.

[27] E. A. Grunditz and T. Thiringer. "Performance Analysis of Current BEVs Based on a Comprehensive Review of Specifications". In: *IEEE Transactions on Transportation Electrification* 2.3 (Sept. 2016), pp. 270–289. DOI: 10.1109/TTE.2016.2571783.

[28] Massimo Guiggiani. "The science of vehicle dynamics". In: *Pisa, Italy: Springer Netherlands* (2014).

[29] N. Guo et al. "A Computationally Efficient Path-Following Control Strategy of Autonomous Electric Vehicles With Yaw Motion Stabilization". In: *IEEE Transactions on Transportation Electrification* 6.2 (2020).

[30] Richard Hammack. *Book of proof*. 2013.

[31] T. Hossain, H. Habibullah, and R. Islam. "Steering and Speed Control System Design for Autonomous Vehicles by Developing an Optimal Hybrid Controller to Track Reference Trajectory". In: *Machines* 10.420 (2022).

[32] T. Hossain, H. Habibullah, and R. Islam. "Steering and Speed Control System Design for Autonomous Vehicles by Developing an Optimal Hybrid Controller to Track Reference Trajectory". In: *Machines* 10 (2022), p. 420.

[33] C. Hu, R. Wang, and F. Yan. “Integral Sliding Mode-Based Composite Nonlinear Feedback Control for Path Following of Four-Wheel Independently Actuated Autonomous Vehicles”. In: *IEEE Transactions on Transportation Electrification* 2.2 (2016).

[34] S. S. James, S. R. Anderson, and M. D. Lio. “Longitudinal Vehicle Dynamics: A Comparison of Physical and Data-Driven Models Under Large-Scale Real-World Driving Conditions”. In: *IEEE Access* 8 (2020), pp. 73714–73729.

[35] H. K. Khalil. *Nonlinear Systems*. 3rd. Upper Saddle River, NJ: Prentice Hall, 2002.

[36] H. Kim and S.-C. Kee. “Neural Network Approach Super-Twisting Sliding Mode Control for Path-Tracking of Autonomous Vehicles”. In: *Electronics* 12.3635 (2023).

[37] Hakjoo Kim and Seok-Cheol Kee. “Neural Network Approach Super-Twisting Sliding Mode Control for Path-Tracking of Autonomous Vehicles”. In: *Electronics* 12.17 (2023), p. 3635.

[38] R. Krishnan. *Electric motor drives: modeling, analysis and control*. New Jersey: Prentice Hall, 2001.

[39] Jiří Lebl. “Basic analysis: Introduction to real analysis”. In: (2009).

[40] LEROY-SOMER. *D.C. motors 2 to 750 kW*. `https://www.leroy-somer.com/documentation_pdf/3805_en.pdf`. [Online; accessed 15-August-2024]. 2020.

[41] Arie Levant. “Quasi-continuous high-order sliding-mode controllers”. In: *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*. Vol. 5. IEEE. 2003, pp. 4605–4610.

[42] Arie Levant. “Robust exact differentiation via sliding mode technique”. In: *automatica* 34.3 (1998), pp. 379–384.

[43] J. Liang et al. “A Polytopic Model-Based Robust Predictive Control Scheme for Path Tracking of Autonomous Vehicles”. In: *IEEE Transactions on Intelligent Vehicles* (2023).

[44] Wei Liu and Zhenzhong Hao. “Anti-Windup Integral Terminal Sliding Mode Control of PMSM System Based on Improved Variable Speed Approximation Law”. In: *2025 8th International Conference on Advanced Algorithms and Control Engineering (ICAACE)*. IEEE. 2025, pp. 1–6.

[45] O. Ljungqvist et al. “A Path Planning and Path-Following Control Framework for a General 2-Trailer with a Car-Like Tractor”. In: *Journal of Field Robotics* 36 (2019), pp. 1345–1377.

[46] O. Ljungqvist et al. “A path planning and path-following control framework for a general 2-trailer with a car-like tractor”. In: *Journal of Field Robotics* 36 (2019), pp. 1345–1377.

[47] C. Lu, J. Yuan, and G. Zha. “Sliding mode integrated control for vehicle systems based on AFS and DYC”. In: *Mathematical Problems in Engineering* 2020 (2020).

[48] Luis.A Torres-Romero. *ControlBlocks/ElectricMotors/BLDCMotor.slx*. `https://github.com/L-Arturo-Torres-Romero/ControlBlocks/blob/master/ElectricMotors/BLDCMotor.slx`. [Online; accessed 15-August-2024]. 2020.

[49] The MathWorks. *Tire (Friction Parameterized)*. Tire-road dynamics given by magic formula coefficients - Simulink - MathWorks América Latina. Online. 2020. URL: `https://la.mathworks.com/help/physmod/sdl/ref/tireroadinteractionmagicformula.html` (visited on 03/04/2020).

[50] Gregory P Matthews and Raymond A DeCarlo. “Decentralized tracking for a class of interconnected nonlinear systems using variable structure control”. In: *Automatica* 24.2 (1988), pp. 187–193.

[51] P. Maurya et al. “A Path-Following Controller for Marine Vehicles Using a Two-Scale Inner-Outer Loop Approach”. In: *Sensors* 22.4293 (2022).

[52] T.J.E. Miller. “Optimal design of switched reluctance motors”. In: *IEEE Transactions on Industrial Electronics* 49.1 (2002), pp. 15–27. DOI: `10.1109/41.982244`.

[53] S. Mondal, A. Mitra, and M. Chattopadhyay. “Mathematical modeling and simulation of Brushless DC motor with ideal Back EMF for a precision speed control”. In: *2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. Coimbatore, India, 2015, pp. 1–5. DOI: `10.1109/ICECCT.2015.7225944`.

[54] S. Mondal, A. Mitra, and M. Chattopadhyay. “Mathematical modeling and simulation of brushless DC motor with ideal back emf for a precision speed control”. In: *Proceedings of the 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. Coimbatore, India, Mar. 2015, pp. 1–5.

[55] Jaime A Moreno and Leonid Fridman. “A survey on high-order sliding-mode control design using Lyapunov functions”. In: *IFAC-PapersOnLine* 56.2 (2023), pp. 25–36.

[56] F. Un-Noor et al. “A Comprehensive Study of Key Electric Vehicle (EV) Components, Technologies, Challenges, Impacts, and Future Direction of Development”. In: *Energies* 10.8 (2017), p. 1217.

[57] Colm J O’Rourke et al. “A geometric interpretation of reference frames and transformations: dq0, Clarke, and Park”. In: *IEEE Transactions on Energy Conversion* 34.4 (2019), pp. 2070–2083.

[58] Katsuhiko Ogata et al. *Modern control engineering*. Prentice Hall India, 2009.

[59] K. Oh and J. Seo. “Development of a Sliding-Mode-Control-Based Path-Tracking Algorithm with Model-Free Adaptive Feedback Action for Autonomous Vehicles”. In: *Sensors* 23.405 (2023).

[60] K. Oh and J. Seo. “Development of a Sliding-Mode-Control-Based Path-Tracking Algorithm with Model-Free Adaptive Feedback Action for Autonomous Vehicles”. In: *Sensors* 23 (2023), p. 405.

[61] Carlos MR Oliveira et al. “Vector control of induction motor using an integral sliding mode controller with anti-windup”. In: *Journal of Control, Automation and Electrical Systems* 27.2 (2016), pp. 169–178.

[62] Hans Pacejka. *Tire and vehicle dynamics*. Elsevier, 2005.

[63] Hans B Pacejka. “Spin: camber and turning”. In: *Vehicle System Dynamics* 43.sup1 (2005), pp. 3–17.

[64] HB Pacejka and Igo Besselink. “Tire and vehicle dynamics”. In: (2012).

[65] B. Paden et al. “A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles”. In: *IEEE Transactions on Intelligent Vehicles* 1.1 (2016), pp. 33–35.

[66] B. Paden et al. “A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles”. In: *IEEE Transactions on Intelligent Vehicles* 1.1 (2016), pp. 33–35.

[67] E. M. Pearce. “Integrated electric motor assembly”. US20150083505A1. Mar. 2015.

[68] P. Pillay and R. Krishnan. “Modeling, simulation, and analysis of permanent-magnet motor drives”. In: *IEEE Transactions on Industrial Applications* 25 (1989), pp. 265–273.

[69] P. Pillay and R. Krishnan. “Modeling, simulation, and analysis of permanent-magnet motor drives. II. The brushless DC motor drive”. In: *IEEE Transactions on Industry Applications* 25.2 (Mar. 1989), pp. 274–279. DOI: 10.1109/28.25542.

[70] Wach Piotr. *Dynamics and control of electrical drives*. Springer Science & Business Media, 2011.

[71] R. Rajamani. *Vehicle dynamics and control*. NY, USA: Springer Science & Business Media, 2011.

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

[73] R. Ritschel et al. “Nonlinear Model Predictive Path-Following Control for Highly Automated Driving”. In: *IFAC-PapersOnLine*. Vol. 52. 2019, pp. 350–355.

[74] R. Ritschel et al. “Nonlinear Model Predictive Path-Following Control for Highly Automated Driving”. In: *IFAC-PapersOnLine*. Vol. 52. 2019, pp. 350–355.

[75] P. Ritzer, C. Winter, and J. Brembeck. “Advanced path following control of an overactuated robotic vehicle”. In: *Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV)*. Seoul, Korea (South), June 2015, pp. 1120–1125.

[76] Y. Shtessel et al. *Sliding mode control and observation*. New York, USA: Springer, 2014.

[77] R. Solea and U. Nunes. “Trajectory Planning and Sliding-Mode Control Based Trajectory-Tracking for Cybercars”. In: *Integrated Computer-Aided Engineering* 14 (2007), pp. 33–47.

[78] Tamiru Takele, Tefera Terefe, and Sam Sun Ma. “Super Twisting Sliding Mode Controller for Trajectory Tracking Control of Autonomous Ground Vehicle System”. In: *International Conference on Advances of Science and Technology*. Cham: Springer Nature Switzerland, 2022, pp. 272–289.

[79] L. Torres-Romero. *L-Arturo-Torres-Romero/VehicleModel: Conventional SMC for lateral dynamic 2*. Zenodo. 2024. DOI: 10.5281/zenodo.10547313.

[80] L. Arturo Torres-Romero. "L-Arturo-Torres-Romero/ControlBlocks: Space-vector PWM for BLDC". In: (Jan. 2022). DOI: 10.5281/zenodo.5816154. URL: https://zenodo.org/badge/latestdoi/274325075.

[81] L.A. Torres-Romero, R. Ruiz-Cruz, and L.E. González-Jiménez. "Path-Following Sliding Mode Controller for an Electric Vehicle Considering Actuator Dynamics". In: *Machines* 12 (2024), p. 219. DOI: 10.3390/machines12040219. URL: https://doi.org/10.3390/machines12040219.

[82] Luis Arturo Torres-Romero and Luis Enrique González-Jiménez. "A Block Controller with Integral Super-Twisting Algorithm for the Path Following of a Self-Driving Electric Vehicle Considering Actuator Dynamics". In: *World Electric Vehicle Journal* 16.12 (2025), p. 643.

[83] J. Ultsch et al. "Reinforcement Learning-Based Path Following Control for a Vehicle with Variable Delay in the Drivetrain". In: *Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV)*. Las Vegas, NV, USA, Oct. 2020, pp. 532–539.

[84] J. Ultsch et al. "Reinforcement Learning-based Path Following Control for a Vehicle with Variable Delay in the Drivetrain". In: *Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV)*. Las Vegas, NV, USA, Oct. 2020, pp. 532–539.

[85] UNFCCC. *The Paris Declaration on Electro-Mobility and Climate Change and Call to Action*. https://unfccc.int/news/the-paris-declaration-on-electro-mobility-and-climate-change-and-call-to-action. [Online; accessed 15-August-2024]. 2015.

[86] V. Utkin, J. Guldner, and J. Shi. *Sliding mode control in electro-mechanical systems*. Boca Raton, USA: CRC Press, 2009.

[87] Vadim Utkin and Jingxin Shi. "Integral sliding mode in systems operating under uncertainty conditions". In: *Proceedings of 35th IEEE conference on decision and control*. Vol. 4. IEEE. 1996, pp. 4591–4596.

[88] G. Velasco-Hernandez et al. "Autonomous Driving Architectures, Perception and Data Fusion: A Review". In: *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*. Cluj-Napoca, Romania, 2020, pp. 315–321. DOI: 10.1109/ICCP51029.2020.9266268.

[89] T. M. Vu et al. “Model Predictive Control for Autonomous Driving Vehicles”. In: *Electronics* 10 (2021), p. 2593.

[90] T.M. Vu et al. “Model Predictive Control for Autonomous Driving Vehicles”. In: *Electronics* 10.2593 (2021).

[91] C. Wang, R. He, and Q. Xia. “Path following control for 4WID-EV based on extended state observer and sliding mode control considering yaw stability”. In: *Advances in Mechanical Engineering* 15 (2023), pp. 1–18.

[92] J. Woo, C. Yu, and N. Kim. “Deep Reinforcement Learning-Based Controller for Path Following of an Unmanned Surface Vehicle”. In: *Ocean Engineering* 183 (2019), pp. 155–166.

[93] J. Woo, C. Yu, and N. Kim. “Deep reinforcement learning-based controller for path following of an unmanned surface vehicle”. In: *Ocean Engineering* 183 (2019), pp. 155–166.

[94] Xiaohan Wu et al. “Robust Control of Electric Power Steering System Based on Aligning Torque Model”. In: *2024 IEEE 22nd International Conference on Industrial Informatics (INDIN)*. IEEE. 2024, pp. 1–6.

[95] Y. Wu et al. “Robust sliding mode prediction path tracking control for intelligent vehicle”. In: *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* 236 (2022), pp. 1607–1617.

[96] XIAOPENG. *Xpeng P7 specs*. `https://en.xiaopeng.com/p7/configuration.html`. [Online; accessed 15-August-2024]. 2020.

[97] Z. Xu et al. “A Model-Tuned Predictive Sliding Control Approach for Steering Angle Following of Full Self-Driving Vehicles”. In: *Proceedings of the Institution of Mechanical Engineers Part C: Journal of Mechanical Engineering Science* 0.0 (2024). DOI: `10.1177/09544062241233921`.

[98] X. Yuan, G. Huang, and K. Shi. “Improved Adaptive Path Following Control System for Autonomous Vehicle in Different Velocities”. In: *IEEE Transactions on Intelligent Transportation Systems* 21 (2020), pp. 3247–3256.

[99] X. Yuan, G. Huang, and K. Shi. “Improved Adaptive Path Following Control System for Autonomous Vehicle in Different Velocities”. In: *IEEE Transactions on Intelligent Transportation Systems* 21 (2020), pp. 3247–3256.

Index of Authors

A

Achkar, Roger, 153

Ahmed, A. A., 112

Alcala, E., 141

Aldbea, F. W., 112

Anderson, S. R., 112

Antsaklis, Panos J, 7, 11

Aparicio Izquierdo, Francisco, 93, XXVII

ARSTECHNICA, 26

B

Bacha, S., 153

Bei, S., 153

Besselink, Igo, 123

Bolton, W. C., 31, 32, 148, 162

Bose, B. K., 148, 149, 151, 162

Bose, Bimal K, 43, 62

Brembeck, J., 141

Brouri, A, 64

C

CARLA, 181, 189, 196

CarSim, 181, 189, 196

Castro, R. De, 141, 153

Cerrada, J., 141, 153, 187

Chattopadhyay, M., 38, 40, 148, 161, XXV

Chen, K.-C., 112

CNET, 38

Coey, J. M. D., 37, 38

Corless, R. M., 130

D

Díaz López, Vicente, 93, XXVII

DeCarlo, Raymond A, 167

Dika, Mohammad, 153

Dynamics, Vehicle, 171, 179, 182, 191

E

Emırler, M. T., 112

F

Favarò, F. M., 1

Fernandez, B., 141, 153, 187

FORBES, 26

Fridman, Leonid, 160, 164

G

González-Jiménez, L.E., 183, 186, 187, 189192, 196, XXXI

González-Jiménez, Luis Enrique, 152, 164

Grunditz, E. A., 26, 27

Guiggiani, Massimo, 88, 90, 120, 123, 126, XXVII

Guldner, J., 141, 153, 169

Guo, N., 141, 153, 187

H

Habibullah, H., 141, 153

Hammack, Richard, 7

Hao, Zhenzhong, 169

He, R., 141

Herrera, P., 141, 153, 187

Hossain, T., 141, 153

Hu, C., 141

Huang, G., 141, 153, 187

I

Islam, R., 141, 153

J

James, S. S., 112

K

Kee, S.-C., 141

Kee, Seok-Cheol, 153

Khalil, H. K., 7, 14, 17

Kim, H., 141

Kim, Hakjoo, 153

Kim, N., 141, 153

Krishnan, R., 38, 41, 148, 161

L

Lebl, Jiří, 7

LEROY-SOMER, 33

Levant, Arie, 80

Liang, J., 141, 153, 187

Lio, M. D., 112

Liu, Wei, 169

Ljungqvist, O., 141, 153

Lu, C., 112

Luis.A Torres-Romero, 41

M

Ma, Sam Sun, 153

MathWorks, The, 89

Matthews, Gregory P, 167

Maurya, P., 141, 153

Michel, Anthony N, 7, 11

Miller, T.J.E., 64

Mitra, A., 38, 40, 148, 161, XXV

Model, Controller Simulink, 171, 179, 182, 191

Mondal, S., 38, 40, 148, 161, XXV

Moreno, Jaime A, 160, 164

N

Nunes, U., 141

O

O'Rourke, Colm J, 44, 47

Ogata, Katsuhiko, 105

Oh, K., 141, 153, 187

Oliveira, Carlos MR, 169

Owayjan, Michel, 153

P

Pacejka, Hans, 88

Pacejka, Hans B, 88

Pacejka, HB, 123

Paden, B., 142, 153

Pearce, E. M., 26, 28, XXIV
Pillay, P., 38, 148, 161
Piotr, Wach, 50

R

Rajamani, R., 143, 144, 156, 182
Rajamani, Rajesh, 85, 90, 91, 93, 97, 98, 102, 103, 120, 123126, XXVII
Ritschel, R., 141, 153, 180, 187
Ritzer, P., 141
Ruiz-Cruz, R., 183, 186, 187, 189192, 196, XXXI

S

Santhosh, J., 112
Seo, J., 141, 153, 187
Shi, J., 141, 153, 169
Shi, Jingxin, 167
Shi, K., 141, 153, 187
Shih, H.-Y., 112
Shtessel, Y., 1214, 72, 78, 79, 146, 153, 164, 168, 169
Solea, R., 141

T

Takele, Tamiru, 153
Terefe, Tefera, 153
Thiringer, T., 26, 27
Torres-Romero, L., 171, 179, 182, 191
Torres-Romero, L. Arturo, 61
Torres-Romero, L.A., 183, 186, 187, 189192, 196, XXXI
Torres-Romero, Luis Arturo, 152, 164

U

Ultsch, J., 141, 153
UNFCCC, 25
Un-Noor, F., 26, 27, 31, 37, XXIV
Utkin, V., 141, 153, 169
Utkin, Vadim, 167

V

Velasco-Hernandez, G., 153
Vera Alvarez, Carlos, 93, XXVII
Vu, T. M., 153, 187
Vu, T.M., 141

W

Wang, C., 141
Wang, R., 141
Winter, C., 141
Woo, J., 141, 153
Wu, Xiaohan, 162
Wu, Y., 141

X

Xia, Q., 141
XIAOPENG, 26
Xu, Z., 153, 187

Y

Yan, F., 141
Yu, C., 141, 153
Yuan, J., 112
Yuan, X., 141, 153, 187

Z

Zha, G., 112

Subject Index

$\alpha\beta$ reference frame, 43

A

Ackerman controller, 105

Ackermann model, 97

Active Steering Assist, 96

advanced driving assistance, 84

aerodynamic drag, 83, 87, 88, 121

Anti-lock Braking System, 88

autonomous driving, 26

B

back EMF, 31

bicycle model, 97, 99, 100, 143, 238, 239

BLDC, 37

Blind Spot Detection, 96

C

Clarke transformation, 43, 46

CoG, 97–99, 103, 104, 113, 114, 126

combustion engine, 83, 85, 94, 120

control by blocks, 109

Control law signal, 68

Control theory, 66

control theory, 7, 68, 80, 100

cornering stiffness, 126, 144, 239

D

DC motor, 31

domain architecture, 21

dq reference frame, 43, 53

dynamic model, 101, 104, 119, 125

E

electric motor, 26, 67

electromagnetic torque, 32, 40

Electronic Stability Control, 88

Emergency Steering Assist, 96

Error signal, 67

F

Feedback elements, 67

Field-oriented Control, 52, 53

G

greenhouse gas, 25

H

heading angle, 98, 99, 126

I

inertial reference frame, 98, 118, 125

ISO 21448, 202, 206

ISO 26262, 202, 206

K

kinematic model, 97, 98, 101

Kirchhoff's voltage law, 31

L

Lane Departure Warning, 96

Lane Keeping Assist, 96

lateral error, 104, 115, 180, 181

Levant differentiator, 16, 80

longitudinal vehicle dynamics, 84

longitudinal velocity, 102–104, 116, 118, 126, 171, 179, 180, 242, 243

Lyapunov function, 16–18, 147

Lyapunov stability, 17, 18

M

magic formula, 87–89, 123, 125

mass density of air, 88, 121

O

orientation error, 104, 126

Output signal, 68

P

Park transformation, 43

Park-Clarke transformation, 45, 47, 54, 149, 150

permanent magnet synchronous motor, 61

PID, 36, 53, 112, 128, 131, 150, 172

plant, 67

PMSM, 26, 29, 43, 52

PMSRM, 26

Q

QR decomposition, 11

R

Rear Cross-Traffic Alert, 96

Reference signal, 67

regular form, 11–13

rolling resistance, 86–88

S

sliding mode controller, 141, 142, 171

slip ratio, 87, 88

software define vehicle (SDV), 23

space-vector, 58

stability, 68

steering angle, 98, 99, 105

T

tire model, 123, 124

Tire Pressure Monitoring System, 88

torque converter, 85

Torque Vectoring System, 88

Traction Control System, 88

Traffic Jam Pilot, 96

transmission, 85, 86, 91, 92, 120, 122–124, 128, 130, 131

V

vehicle dynamics, 2, 83, 84, 96, 112, 113

W

wheel dynamic, 86, 125