

POLITECNICO DI TORINO

---

Corso di Laurea Magistrale in Automotive Engineering  
Dipartimento di Ingegneria Meccanica e Aerospaziale



# Design and implementation of an Electric Power Steering system for a Formula Student Driverless vehicle

Candidato: **Raffaele Manca**

---

Anno Accademico 2019 - 2020

POLITECNICO DI TORINO

---

Corso di Laurea Magistrale in Automotive Engineering  
Dipartimento di Ingegneria Meccanica e Aerospaziale



Tesi di Laurea Magistrale

# Design and implementation of an Electric Power Steering system for a Formula Student Driverless vehicle

Politecnico di Torino:

**Prof. Nicola Amati** - Relatore  
**Prof. Andrea Tonoli** - Corelatore

Candidato:

**Raffaele Manca**

Supervisori:

**Salvatore Circosta**  
**Irfan Khan**

---

Anno Accademico 2019 - 2020



# Abstract

Electric power steering (EPS) systems represent a key technology for highly automated driving. Beside the ergonomical function of reducing the physical effort required to the driver during the steering manouver, EPS give the possibility of high precision control of the the steering system, furthermore the integration of the control unit into the vehicle electrical system paves the way to automated driving function.

Formula Student Driverless competition provides a unique opportunity to design and test autonomous control systems required in automated transport. Thus the aim of this master thesis is the design, control and implementation of an autonomous steering system for a Formula SAE Driverless vehicle. The requirements and regulations of Formula Student competition among with the constraints for the integration of the autonomous steering actuator with the steering system of the vehicle SC 19 developped by Squadra Corse in Politecnico di Torino, led to the layout chosen. It consists of the linear actuation of the rack by using a ball screw shaft. The ball screw nut is rotated using a belt transmission driven by a brushless DC electric motor.

The present work examines advantages and characteristics of the layout chosen. The attention has been focused on the design, tuning and implementation of the control of the BLDC motor. Numerical models of the motor and of the loads have been developped in Matlab/Simulink environment. Hence the PI current controller and the PID position controller has been designed and modeled in order to tune and optimize it for enhancing its performances. The complete model has been experimentally validated with the data recorded from motor sensors.

Furthermore the CANOPEN communication between the Autonomous steering system motor control unit and the car ECU is implemented and tuned for the autonomous driving mission. The last step of the project consists in setting up a test bench for the complete system testing and validation.



---

# Acknowledgements

I would like to express my great gratitude to Prof. Nicola Amati, supervisor of this thesis, for the opportunity of taking part to this project, for his trust in me and for his availability and constant support. I would also like to thank Prof. Andrea Tonoli for the valuable suggestions and for always addressing the work in the right direction.

A big thank to Dr. Salvatore Circosta and Dr. Irfan Khan, for their professionalism, for all the time dedicated to me in constantly supporting my activity. Thanks also to Dr. Stefano Feraco and Dr. Sara Luciani and to all the LIM (Laboratorio Interdisciplinare di Meccatronica) for making me feel part of a great team. Thanks to Mario Silvagni for the valuable help in configuring the Maxon motor.

A special thanks to my mother and father, for having always been there over this long course of the study, for listening to me in every difficult moment, for making me the person I am today.

Thanks to all the colleagues of Politecnico, who accompanied me in these years, Manlio, Niccolò, Eugenio, Gennaro and in particular Luca for having shared all the most memorable moments of this journey, inside and outside the university.

Thanks to the cultural association Tramoontana and to my life long friends Domenico, Paolo, Giuseppe, Andrea, Arianna, Pierfrancesco and Cristian for always supporting me.

Thanks to my second family of Via Isonzo 99, Francesco, Pierluigi, Mattia L., Mattia S., Andrea and Giovanni, for making me feel at home even in Turin.

Thanks to my guitar, to all the records listened to and to music for cheering me up every time I need it.

---

# Contents

<b>Acronyms</b>	<b>1</b>
<b>Symbols</b>	<b>4</b>
<b>1 Introduction</b>	<b>2</b>
1.1 Background . . . . .	2
1.2 State of art of Electric Power Steering system (EPS) . . . . .	4
1.3 Formula Student . . . . .	6
1.4 Motivation and Aim . . . . .	8
1.5 Thesis outline . . . . .	9
<b>2 Autonomous Steering System (ASS) Actuator</b>	<b>11</b>
2.1 Formula Student Regulations for the steering system . . . . .	11
2.2 SC19 and its steering system . . . . .	12
2.3 ASS working principle . . . . .	14
2.4 ASS chosen solution . . . . .	17
<b>3 Systems modelling</b>	<b>23</b>
3.1 BLDC motors . . . . .	24
3.2 Motor modelling . . . . .	26
3.3 Equivalent Inertia Computation . . . . .	28
3.4 Steering resistance torque considering tire/pavement friction . . . . .	30
3.5 Self aligning moment and steering resistance torque in dynamic conditions	33
3.6 Motor controller architecture . . . . .	35
3.7 PID controllers . . . . .	37
3.8 PI Current Controller . . . . .	39
3.8.1 PI Current Controller tuning . . . . .	40
3.9 PID Position Controller . . . . .	42
3.9.1 PID Position Controller tuning . . . . .	44
3.10 Complete system model . . . . .	45

<b>4</b>	<b>System Validation and Implementation</b>	<b>48</b>
4.1	Models validation . . . . .	48
4.1.1	PI Current Controller model validation . . . . .	48
4.1.2	PID Position Controller model validation . . . . .	50
4.2	CANopen communication . . . . .	52
4.3	CANopen communication implementation . . . . .	55
4.3.1	NMT communication implementation . . . . .	56
4.3.2	PDO communication implementation . . . . .	57
4.4	Test bench validation . . . . .	62
4.5	On-vehicle implementation . . . . .	64
<b>5</b>	<b>Conclusions and future development</b>	<b>66</b>
5.1	Future development . . . . .	66
<b>A</b>	<b>FSD regulations</b>	<b>69</b>

---

# Acronyms

<b>AS</b>	Autonomous System
<b>ASS</b>	Autonomous Steering System
<b>EBS</b>	Emergency Brake System
<b>BLDC</b>	Brushless Direct Current
<b>FSDV</b>	Formula Student Driverless Vehicle
<b>RES</b>	Remote Emergency System
<b>ASMS</b>	Autonomous System Master Switch
<b>DOF</b>	Degree Of Freedom
<b>PM</b>	Permanent Magnet
<b>PID</b>	Proportional Integral Derivative
<b>FFW</b>	Feedforward
<b>LIM</b>	Laboratorio Interdisciplinare di Meccatronica
<b>HTD</b>	High Torque Drive
<b>SAE</b>	Society of Automotive Engineers
<b>AD</b>	Autonomous Driving
<b>ADAS</b>	Advanced Driver Assistance System
<b>ABS</b>	Anti-lock Brake System
<b>ESP</b>	Electronic Stability Program
<b>EPS</b>	Electric Power Steering
<b>CV</b>	Combustion Vehicle

<b>EV</b>	Electric Vehicle
<b>DV</b>	Driverless Vehicle
<b>FSG</b>	Formula Student Germany
<b>R2D</b>	Ready To Drive
<b>IMU</b>	Inertial Measurement Unit
<b>CAN</b>	Controller Area Network
<b>CPU</b>	Control Program Unit
<b>LVMS</b>	Low Voltage Master Switch
<b>ASR</b>	Autonomus System Responisble
<b>GUI</b>	Graphical User Interface
<b>PMSM</b>	Permanent Magnet Synchronous Machine
<b>PWM</b>	Pulse-width modulation
<b>PPM</b>	Profile Position Mode
<b>HM</b>	Homing Mode
<b>PVM</b>	Profile Position Mode
<b>DOF</b>	Degrees of freedom
<b>CSP</b>	Cyclic Synchronous Position Mode
<b>CSV</b>	Cyclic Synchronous Velocity Mode
<b>CST</b>	Cyclic Synchronous Torque Mode
<b>ISO</b>	International Organization for Standardization
<b>LLC</b>	Logical Link Control
<b>MAC</b>	Medium Access Control
<b>RTR</b>	Remote Transmission Request
<b>PDO</b>	Process Data Objects
<b>SDO</b>	Service Data Object
<b>NMT</b>	Network Management Objects

**DBC**   Data Base CAN

**DLC**   Data Length Code

---

# Symbols

$T_{bs}$	Ball screw torque
$n$	Motor angular speed [rpm]
$A_{wire}$	Cross section of the wire
$B$	Magnetic flux density
$F_{rack}$	Required rack force
$\eta_{bs}$	Ball screw efficiency
$P_m$	Motor power
$v_{bs}$	Linear ball screw speed
$v_a$	Motor terminal voltage
$i_a$	Motor armature current
$L_a$	Terminal inductance
$R_a$	Terminal resistance
$e$	Back-EMF
$k_e$	Motor speed constant
$\omega$	Rotational speed [rad/s]
$J$	Moment of inertia
$r$	Damping coefficient
$T_{res}$	Resisting friction torque
$T_m$	Electromagnetic motor torque
$k_t$	Motor torque constant
$I_0$	No load current
$n_0$	No load speed
$s$	Laplace variable
$m_{s \rightarrow r}$	Equivalent mass fo the steering wheel reported to the rack
$m_r^*$	Equivalent mass of the rack
$b$	Longitudinal tie rod offset
$J_{r \rightarrow m}^*$	Equivalent inertia of the motor load
$\tau_{bs/belt}$	Transmission ratio of the ball screw/belt
$c$	Caster trail
$F_{y/z}$	Side force/Vertical load
$\mu$	Friction coefficient
$F_{rack}$	Required rack force

$g$	Gravity acceleration
$M_z$	Self alligning torque
$t$	Pneumatic trail
$M_{z0}$	Self alligning moment linear coefficient
$\alpha$	Tire sideslip angle
$l$	Vehicle length
$a$	Distance between front axle and center of gravity
$m_s$	Sprung mass
$m_{ns}$	Unsprung mass
$\beta$	Vehicle sideslip angle
$r$	Yaw rate
$\delta$	Steering angle
$k_P$	Proportional gain
$k_I$	Integrative gain
$k_D$	Derivative gain
$\tau$	Electric time constant



---

# List of Figures

1.1	EPS from Bosch with recirculating ball drive on the rack [9]	5
1.2	FSD vehicle during FSG 2019 [15]	7
1.3	FSD 2021 disciplines and points [15]	8
2.1	SC 19 Squadra Corse vehicle	12
2.2	SC19 steering system CAD [16]	14
2.3	Track layout of a trackdrive discipline (FSG 18) [7]	15
2.4	Autonomous System (AS) state machine [17]	16
2.5	ASS actuators example [15]	17
2.6	Bosch Rexroth ball screw	19
2.7	Autonomous Steering System SC 19 CAD	21
2.8	ASS top view CAD	21
2.9	ASS X-Section of the support structure CAD	22
2.10	ASS CAD details	22
3.1	Maxon motor and EPOS controller	23
3.2	Types of DC motors [15]	24
3.3	Three-phase inverter controlling BLDC motor [18]	25
3.4	Simulink model of the BLDC motor	27
3.5	CAD models for computation of the inertia of the steering wheel and column subassemblies	28
3.6	CAD models for computation of the inertia of the wheel	30
3.7	Kingpin geometry of a quarter car [19]	31
3.8	Simulink model of the friction torque	32
3.9	SC 19 self aligning moment	33
3.10	Motor resistant torque due to the self aligning moment for a sinusoidal steering manouvre at 30 km/h	35
3.11	Overview on the controller architecture	36
3.12	Step response characteristics [21]	38
3.13	Current regulator control architecture [20]	39
3.14	PI current controller Simulink model	40
3.15	Bode diagram of the closed loop PI current control system	41

3.16	Step response of the closed loop PI current control system . . . . .	42
3.17	Position regulator with feed forward, controller architecture [20] . . . .	42
3.18	Position PID controller Simulink model . . . . .	43
3.19	Step response of the closed loop PID position controller model . . . . .	45
3.20	Bode diagram of the closed loop PID position controller model . . . . .	45
3.21	Squadra Corse data - Steering angle as function of the rack displacement	46
3.22	Sine wave performances of the complete system model . . . . .	47
4.1	Regulation tuning in Epos Studio software screenshot . . . . .	49
4.2	Comparison between the behaviour of the maxon motor current and of the model following the same reference profile . . . . .	49
4.3	Regulation tuning (PID) in Epos Studio software screenshot . . . . .	50
4.4	Comparison between the Maxon motor actual position and the model behaviour following the same ramp profile . . . . .	51
4.5	Comparison between the Maxon motor actual position and the model behaviour following the same profile . . . . .	51
4.6	Test bench set up for motor controller tuning and validation . . . . .	52
4.7	Protocol layer interactions in CANopen communication [22] . . . . .	53
4.8	CAN communication, NMT slave states [22] . . . . .	55
4.9	Simulink screenshot of the CAN communication implementation for the ASS . . . . .	56
4.10	Device State Machine for enabling/disabling operations . . . . .	58
4.11	Epos Studio transmitting PDO mapping function screenshot . . . . .	59
4.12	Epos Studio receiving PDO mapping function screenshot . . . . .	60
4.13	Device state bits [23] . . . . .	61
4.14	Autonomous Steering Actuator test bench 1 . . . . .	63
4.15	Autonomous Steering Actuator test bench 2 . . . . .	63
4.16	Autonomous Steering Actuator test bench 3 . . . . .	64
4.17	ASS block scheme . . . . .	65
5.1	SC 19 on a FSD trackdrive scenario 1 . . . . .	67
5.2	SC 19 on a FSD trackdrive scenario 2 . . . . .	68

---

# CHAPTER 1

---

## Introduction

Before delving into the topic of this master thesis, the aim of this chapter is to provide an introduction to the dissertation in order to understand the background and the goals of the project along with the structure and methodology of the thesis.

An overview on the state of the art of autonomous vehicles and power steering systems and a description of the competition of Formula SAE Driverless will be provided, analysing the regulations and safety requirements for the systems. The Autonomous Steering System (ASS), along with the sensors, CPU and hardware elements necessary for the autonomous driving competition, have to be implemented and installed on the already existing vehicle *SC19 - Lucia*, developed by *Squadra Corse* team in Politecnico di Torino. For this reason the vehicle will be presented, focusing on its steering system; furthermore the mechanical design of the ASS actuator will be discussed before studying in deep the control.

### 1.1 Background

Since its conception in late 1800's, the automobile has seen a large change and development. Although the automobile was first invented and perfected in Germany and France, Americans quickly came to dominate the automotive industry in the first half of the twentieth century, introducing new mass-production techniques that became standard and Ford, Chrysler and General Motors emerged as *Big Three* of auto in 1920's. After World War II, European car makers started to meet growing demand, until the rise of Japan as the leading automaker by 80's [1].

Nowadays automotive industry is facing a big shift towards new mobility trends. The international consulting company *P.w.C.* defined the future scenario of mobility as *Eascy* - electrified, autonomous, shared, connected and yearly updated [2]. From this analysis emerges several trends for future mobility with a progressive increase of car-sharing services, autonomous mobility and with the introduction of new entrants (*Uber*,

Google, Apple) in the automotive industry. This scenario is driven by several forces and drivers like the demographic change, the urbanization, the increase of environmental awareness, the shift in habits and mobility preferences of younger generations and the rapid advance in connected vehicles technologies (V2V and V2I - vehicle to vehicle and vehicle to infrastructure). Another driver which is fundamental to take into account and which will influence the mobility in the current year is the Covid-19 pandemic outbreak. The imposed social distance is leading to the a decrease in the demand for public transportation and a big spread of micromobility services in the urban areas (such as bikes and electric scooter), furthermore the spreading of smart working in many companies is reducing the need of owning a car favoring the spread of car sharing services. In this scenario automation has a fundamental role infact by 2030 the widespread adoption of robo-taxis and autonomous commercial trucking is expected.

Aside from these trends and factors, the automation process is mainly due to the increase in active safety as top priority for car manufacturers. With 1.2 milion people dying on road accidents in 2019, road injuries is the top 10 cause of death worldwide [3] moreover research [4] states that driver-related behavioural factors contribute to the occurrence of 95% of all road accidents. Thus in the last years, in parallel to the improvement of passive safety technologies, active safety is a key element for modern vehicles. Features like ABS, ESP, Autonomous Braking System, Adaptive Cruise Control and Lane Keeping Systems are referred as Advanced Driver Assistance Systems (ADAS), this features expands from premium to mass offerings markets and they represent a key-point in the progressive transition towards Autonomous Driving (AD). Society of Automotive Engineers (SAE) defined five different levels for automation [5]:

- *Level 0 - No Automation.*
- *Level 1 - Driver Assistance.* ADAS systems assist drivers whom still has to be in total control.
- *Level 2 - Partial Automation.* The vehicle can control the speed and steering but the driver must be ready to take control at any moment.
- *Level 3 - Conditional Automation.* Autonomous vehicles are capable of driving themselves, but only under ideal conditions and with limitations.
- *Level 4 - High Automation.* Vehicles can drive themselves without human interactions (besides entering your destination) but will be restricted to known use cases.
- *Level 5 - Full Automation.* Level 5 capable vehicles should be able to monitor and maneuver through all road conditions and require no human interventions whatsoever, eliminating the need for a steering wheel and pedals.

Level 4 autonomy is expected to be disruptive and available between 2020 and 2022, while full autonomy with Level 5 technology is projected to arrive by 2030 at the earliest [6].

In this context the academic activities for developing an autonomous single seated race car capable to manage handling, acceleration and braking capabilities in a closed circuit for the Formula Student Driverless (FSD) competitions, provides a platform to develop and validate new technologies under challenging conditions. Self-driving race-cars provide a unique opportunity to test software required in autonomous transport, such as redundant perception, failure detection, and control in challenging conditions [7].

## **1.2 State of art of Electric Power Steering system (EPS)**

Electric power steering (EPS) is a key technology for highly automated driving. The main goal of power steering system is to help the driver in the steering maneuver trying to reduce the physical effort required specially at low speed, furthermore power steering systems are able to provide feedback of forces acting on the steering wheel so that the driver does not loose the driving sensation.

The first typology of servo power steering system were hydraulic power steering system and apperared in the market in the 50's, developed by General Motors. This system applies, as power source, the hydraulic pressure of a certain flow of oil, generated by a pump driven by the engine [8].

The first Electric Power Steering system has appeared during last decade of the past century on the Honda NSX. The power assistance is applied to rack and pinion steering boxes and operate through an electric motor providing the torque to the steering column or directly to the pinion or the rack. In this last application, generally the motor acts on the rack by using a recirculating ball drive and a belt drive (an alternative way to transimt the motor torque to the rack is the double pinion solution [9]). For this applications the motor is a simple direct current with permanent magnets. The electric motor is controlled by an electronic controller which provides the assistance torque basing on data about steering torque and angle from data recorded by specific sensors. Dealing with the control systems, the most widespread strategy is a two-layer fuzzy controller, in which the high level calculates the target position of the steering wheel to fit the vehicle to the desired route, while the low level computes the torque required to the motor to reach the target position of the rack . For example, Toyota's automatic parking system have taken this approach [10].

Aside of the ergonomical function of power steering for assisting the driver in steering and increasing safety, in the context of automated mobility EPS becomes the key tech-

nology for implemented autonomous driving. The integration of the electronic control unit into the vehicle electrical system paves the way to automated driving functions. In the last systems developed by Bosch, the controller is directly integrated with the motor in a modular built steering control unit. For automated driving, the system has a redundant layout including a second independently working electronic circuit, capable of providing, in the case of failure, at least the 50% of steering support in order to stop safely the vehicle. This EPS system enable driving till SAE-Level 5 of automation [9].



Figure 1.1: EPS from Bosch with recirculating ball drive on the rack [9]

For industrial applications, the tendency is to adopt PID control as a cost effective and easily tunable solution. Anyway this type of control may present a series of drawbacks dealing with the multi order nature of a steering system and can allow resonance frequencies affecting the stability or tracking performances of an EPS angle control [11]. Complementary work presented in [12] on PID control for front steer angle control gives an insight on the robustness drawbacks mentioned before and suggests the adoption of filters, antiwind-up and no zero structures in order to enhance the stability of the control systems.

Many researchers have reported the work on modelling and position control strategies of electric power steering systems for autonomous driving applications. In [13] Vehicle Automation and Chassis Systems department of *Daimler AG* presented an EPS model and the design of a controller in order to ensure accurate and robust following of the desired trajectories of front steering angle and smooth steering wheel movement. The steering model developed shows all the nonlinearities for the elastic elements, frictions and gear ratios. Thereafter the system has been linearized and controlled using a Linear

Quadratic Integrator controller as far as pole placement techniques. The performances of these controllers are checked in simulation environment, showing good tracking performances from both the two approaches presented and supports the use of a state space design approach for autonomous driving applications.

### **1.3 Formula Student**

Formula Student (FS) is an international competition for university students, it represents the Europe's most established educational engineering competition which uses motorsport to inspire students. It was born in the 1981 in the United States, when the Society of Automotive Engineers (SAE) started running its first Formula SAE program. The competition is based on the Formula SAE rules and guidelines [14]. The challenge consists of designing and constructing a single-seated formula style race car. This format provides an opportunity for students to improve engineering knowledge and capabilities to deliver a complex and integrated product in the environment of an interdisciplinary and international motorsport competition.

The competition consisted of a combustion (CV) and an electric (EV) class, by 2017 the driverless (DV) class has been introduced. Formula Student Driverless (FSD) takes place in the historic Hockenheimring circuit in Germany, but starting from 2018 also Formula SAE Italy and Formula Electric Italy along with Formula Student UK and Formula East have decided to follow the German event introducing a Driverless competition too. During the current year, due to the Covid 19, the events did not take place; the roadmap for next years competitions, presented by Formula Student Germany (FSG), expect to integrate by 2022 FS Driverless event directly into both FS Combustion and FS Electric events and to eliminate FS Combustion by 2023 [15].

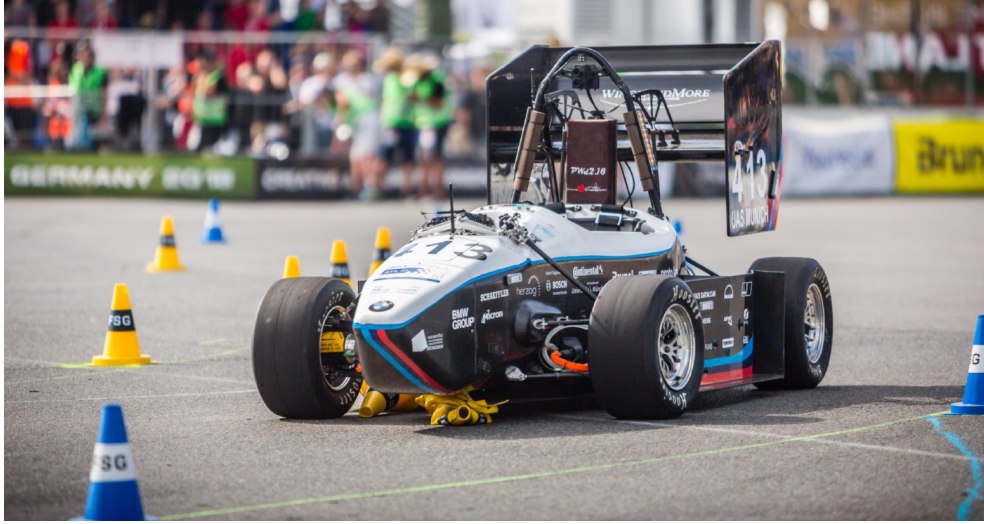


Figure 1.2: FSD vehicle during FSG 2019 [15]

The FSD competition is divided into static and dynamic disciplines. The team, apart from designing and developing a race-car meeting all the required technical aspects, has to present a business and marketing plan. A jury comprising experts from automotive and motorsport world and industry will evaluate the car according not only to its performances, but also to ergonomics, aesthetics and economic aspect such as the use of off-the-shelf components in order to make easier the replacement. The team which will earn the best overall score based on these design, performance, financial and marketing parameters will win the Championship. The sub-events for driverless vehicle are specifically for FSD, they focus on the capability of the autonomous vehicle to adapt to the tested driving scenario. The points for each discipline of the FSD are different from those of FSC and FSE, but the overall score is the same. All the cars have to attend a series of inspections for all autonomous systems (AS) and hardware elements such as the Emergency Brake System (EBS), in order to check the compliance with regulations and to be admitted to the competition. Furthermore each driverless vehicle must respect the base requirements for the combustion or the electric class and the vehicle must be able to be operated in both driverless and with driver modes [17].



<b>Disciplin</b>	<b>Points</b>
Business Plan Presentation	75
Cost and Manufacturing	100
Engineering Design	150
Acceeleration	75
Skid Pad	75
Autocross	100
Endurance	325
Efficiency	100
<b>Total</b>	<b>1.000</b>

Figure 1.3: FSD 2021 disciplines and points [15]

## 1.4 Motivation and Aim

Considering the big shift towards autonomous driving that automotive industry is facing, it is of utmost importance to optimize the control techniques and to increase the know how about driverless car in a research environment like Formula Student.

The state of the art of Electric Power Steering presents a series of commercially available systems with working principles comparable to the Steering Actuator developed during this project. The main difference is the field of application. The actuator designed will work for a fully autonomous vehicle able to run driverless in a closed environment. Thus the designed Electric Power Steering system has to deal with a series of design constraints for the field of application and for the regulations of the Formula Student Driverless Competition. For this reason the motivation behind this thesis project is not to innovate with the actuator itself, but to provide a suitable and optimized control strategy for the Electric Power Steering in the innovative field of application of fully autonomus driving.

Furthermore optimization practices based on virtual models grant the opportunity of analyzing different design versions, tuning the design parameters in a simulation environment in order to enhance the performances of the system.

The aim of this thesis project is established by taking into account of both all the necessities dictated by the accademic project of formula SAE driverless and also the industrial perspective of optimizing the control strategies for autonomous steering sys-

tems.

All the project has been conducted with an experimental attitude, the aim of the project is not only to design and optimize in a simulation environment a suitable autonomous steering system for a driverless vehicle application, but also to implement it. Thus the project includes the interface with the manufacturing company of the actuator and to deal with the hardware specifications for actual implementing the control systems.

Since the autonomous system has to be installed on the already existing vehicle SC 19 developed by Squadra Corse, during the entire project the driverless team cooperated with the Squadra Corse team in order to face all the design constraints of compatibility and reversibility of the driverless functions.

Within the driverless team itself the project is conducted with a cross functional attitude, in which each team member, in charge of a specific autonomous function or subsystem, collaborate with the team in order to reach the final common goal of designing and implementing a working prototype of a Formula SAE Driverless vehicle.

## 1.5 Thesis outline

After a first phase of finalization of the mechanical design and of components selection for the Autonomous Steering Actuator, the methodology adopted for the project mainly consists in modelling, tuning, validation and implementation of the systems.

Excluding the current introductory chapter which will provide an overview on the background and aim of the thesis, the structure of the dissertation is:

- in *Chapter 2 - Autonomous Steering System Actuator*, the layout chosen of the ASS will be presented and discussed taking into account all the design constraints. The Formula SAE regulations will be discussed among with the main features of the steering system of SC 19 Squadra Corse vehicle.
- *Chapter 3 - Control Systems modelling*, deals with the model design phase. A brief theoretical background about the BLDC motors and PID controllers will be provided. The models of the motor, PI current controller and PID positioning controller will be analysed in detail. The main load acting on the motor will be also modeled and discussed. Furthermore the tuning procedure of all the control parameters will be described.
- in *Chapter 4 - Systems validation and implementation*, the test bench for systems validation and testing will be set up. The implementation of the control architecture and of the CANOPEN communication will be deeply analysed.
- *Chapter 5 - Conclusions*. This last chapter will provide a conclusion to the dissertation discussing the future steps and developments of the project in order

to obtain a final complete Formula Student Driverless Vehicle compliant with the regulations of FSD competition.

---

## CHAPTER 2

---

# Autonomous Steering System (ASS) Actuator

Before analyzing the design and implementation of the control logic, it is important to provide an overview on the hardware actuator. The ASS has to fulfill a series of regulations and has to integrate with the already existing vehicle SC19 and its steering system in order to obtain the performances required.

### 2.1 Formula Student Regulations for the steering system

Formula Student 2020 rulebook [17] provides a set of guidelines for designing the system. Since FSDV must function as both a DV and a vehicle with driver, it should be compliant not only with the FSD regulations, but also with the general guidelines for the steering system and for the EV.

The regulations for steering system are and the additional guidelines for the driverless competition are reported in Appendix A.

To summarise, the keypoints to take into account in designing the steering systems are that the manual steering should directly actuate the front wheels by using only mechanical linkages and that the entire system must be mechanically attached to chassis of the vehicle by the means of connections clearly visible during the inspection phase. The "steer-by-wire" functionalities are allowed only for FSDV. It is important to state for FSD that the ASS actuation must start only when the vehicle is R2D, remaining active until the vehicle is at stand still after the R2D state was left, thus it is important to make sure that the actuator does not need to move for position referencing, as this will only be allowed while the vehicle is R2D. Since the vehicle can be operated in both

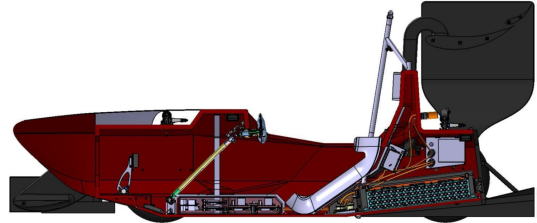
driverless and with-driver modes, manual steering must be possible as soon as ASMS is switched off and it is important to consider the impact of the steering forces generated by the ASS while manual driving [15].

## 2.2 SC19 and its steering system

The SC19 is the 7th electric car designed by *Squadra Corse*. It is equipped with four *AMK DD5 14 POW* electric motors installed in the uprights outboards, coaxially to the wheels. The motors reach a maximum speed of 20,000 rpm and provide 21 Nm of torque each at 8000 rpm. The car has a carbon fibre monocoque which allow to reduce the mass of the car of 7.5 Kg compared to the 2018 vehicle, reaching a kerb weight of 190 Kg, improving also the aerodynamic, dynamic performances and the torsional stiffness of the vehicle.



(a) SC19 Lucia



(b) SC19 Longitudinal cross section CAD

Figure 2.1: SC 19 Squadra Corse vehicle

Due to its compact size, precision and ease of repair and maintain, mechanical rack and pinion steering is the most widespread solution in FS cars. The steering system in SC19 is a mechanical rack and pinion with herringbone like gear profile. This gear profile lead to the advantages of not producing additional axial load like single helical gears and to balance the side thrust. SC19 steering rack is a customized solution starting from the *zRack* provided by the canadian manufacturer *Zedaro*, with a rack length of 264 mm in order to fit the reduced space in the front cross sectional area of the monocoque in the YZ plane [16]. The steer ratio of the system is of 4.3:1 with a steering wheel working angle from -90 to +90 degrees.

The technical specifications of the steering rack installed on the vehicle are reported in the table below:

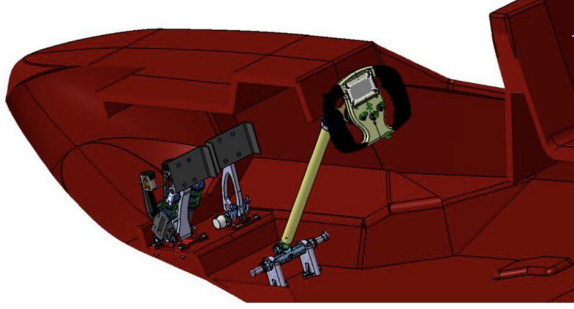
Since the power assist for steering system is not allowed in FS not driverless vehicles, the system has to be completely mechanical, thus the power effort necessary to steer the wheels made by the driver represents a design constraint in suspension design and

Table 2.1: Zedaro Zrack technical specifications:

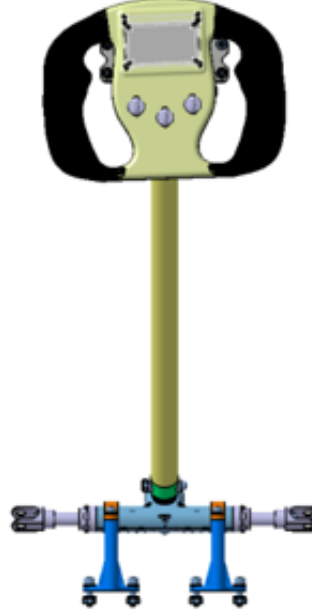
Modulus	$2\text{ mm}$
Pitch	$6.28\text{ mm}$
Primitive circle diameter	$27.8\text{ mm}$
Pressure angle	$20\text{ deg}$
Angular distance between two teeth	$26.4\text{ deg}$
Length of the contact patch	$8.8\text{ mm}$
contact ratio ( $\epsilon$ )	$1.40$
C-Factor	$85.5\text{ mm}$

hard points fixing. Furthermore the steering rack is equipped with a rotary position sensor that measures the angular position of the pinion, this signal can be used in the control loop of the ASS.

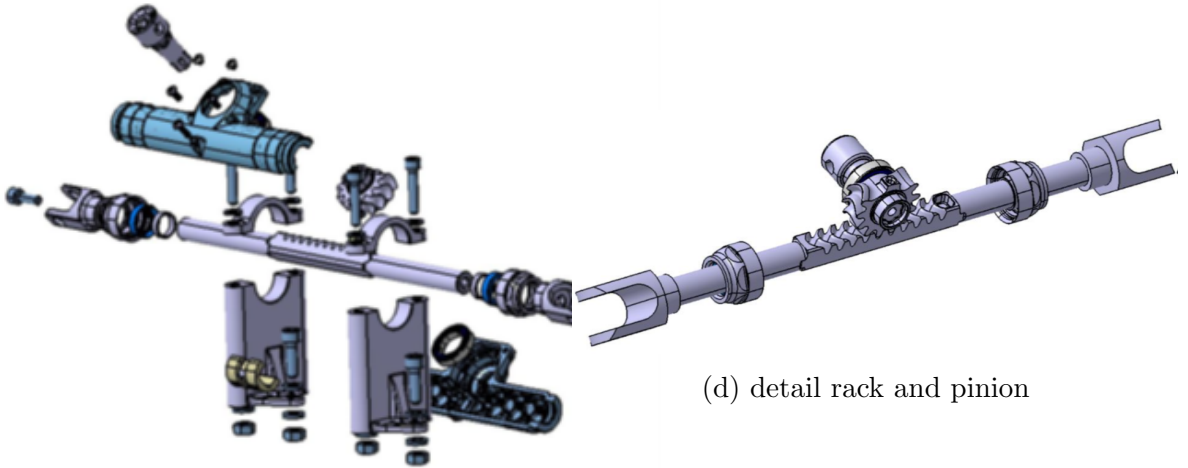
In fig. 2.2 the CAD models of the main subsystems for the steering system of SC 19 are reported.



(a) Isometric view of the steering system inside the cockpit



(b) Driver point of view



(c) Exploded view

(d) detail rack and pinion

Figure 2.2: SC19 steering system CAD [16]

## 2.3 ASS working principle

The purpose of the ASS in a FSDV is to drive the vehicle, along with the traction system, along the competition trackdrive. Furthermore the steering system is responsible for handling the vehicle in accident-avoidance manoeuvre to bring the car to a safe stop keeping the control of the yaw rate also during an emergency braking situation. The main race, the *trackdrive*, consists of completing ten laps around an unknown track made by a series of blue and yellow cones standardized by FSG. The track is a 500 m long closed circuit, with a width between 3 m and 5 m and it contains straights,

chicanes, harpines, multiple turns and decreasing radius turns. Before the starting of the race the track is unknown to all the participants [7].

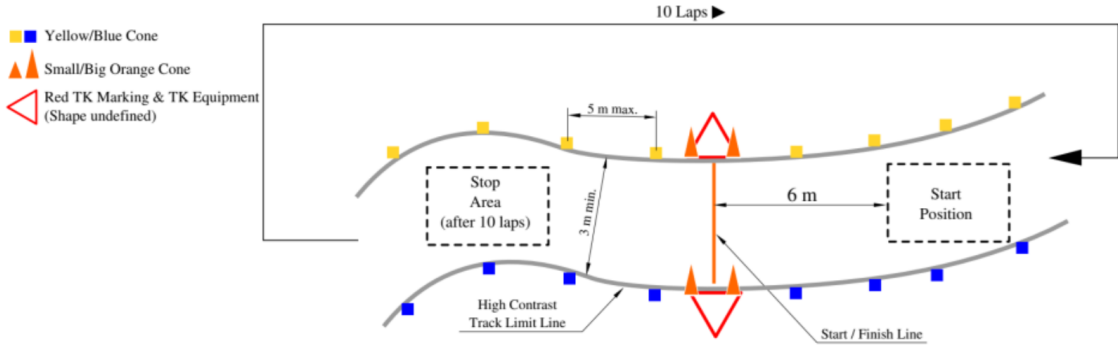


Figure 2.3: Track layout of a trackdrive discipline (FSG 18) [7]

In order to make the vehicle race fully autonomous, a FSDV is equipped with two main sensors for the environment perception: a 3D LiDAR placed on the front wing and a stereocamera. All the signals from these sensors and from the IMU system of the vehicle are processed on board on specific CPU and GPU units like the dSpace MicroAutoBox, in order to perform the perception and path planning tasks and to control the actuators.

In this context the two main actuators installed on the FSDV are the ASS and the EBS. The Emergency Brake has to be a passive system with a non-programmable logic, while the steering actuator communicates through the CAN bus with the dSpace to take as input the computed reference steering angle. Considering the ASS, it is connected to LVMS and to the ASMS which is in charge of defining the transitions between the different modes of operations required by the competition. The picture below gives a schematic representation of the FSDV modes.



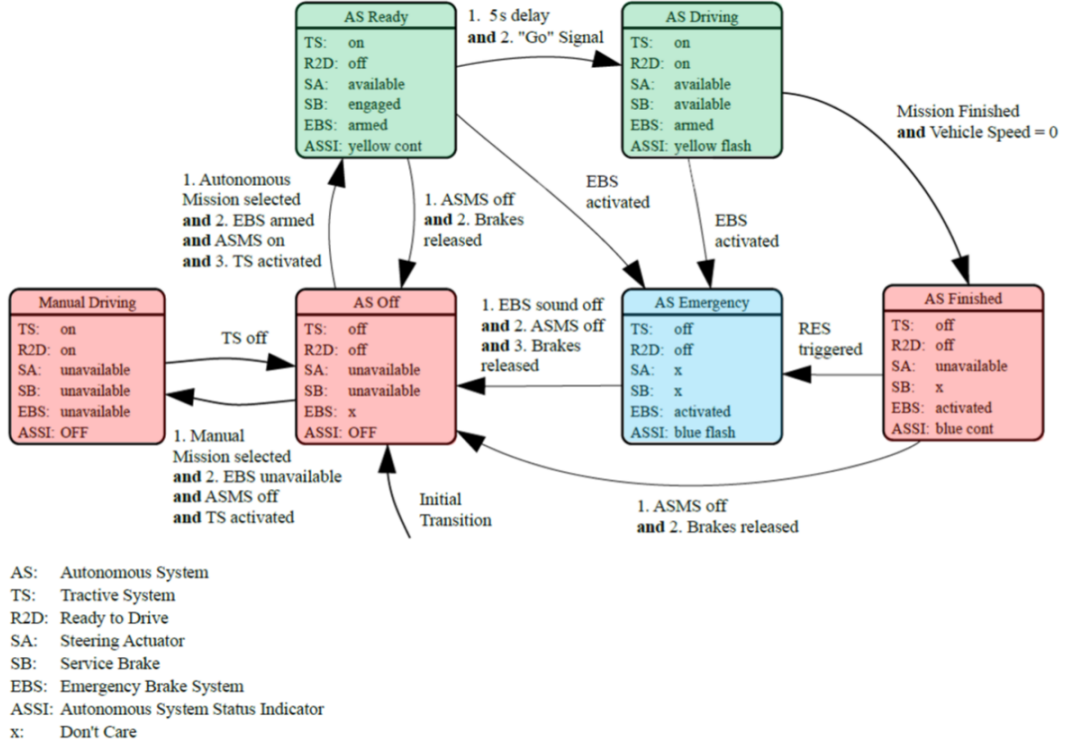


Figure 2.4: Autonomous System (AS) state machine [17]

As far as the ASS is concerned:

- The steering Actuator has to be available in case of:
  - the ASMS is ON and the vehicle enters in the state *Ready*.
  - The Autonomous System Responsible (ASR) triggers the Remote Emergency Switch (RES) giving the GO signal and the vehicle's speed is  $\neq 0$ .
- The Steering Actuator has to be unavailable in the case of:
  - Completion of the driving event, with ASMS state *ON* and vehicle speed=0.
  - The ASMS is switched to OFF.
  - The manual driving mode is selected.
- During the *Emergency* state, when the EBS is engaged, the SA availability is not mandatory (DV2.3.2 [17]). Anyway the SA can counter steer the vehicle during the emergency braking situation, in order to increase the vehicle stability counteracting the high yaw rate leading to a safe stop.

The controller of the State Machine, in charge of defining the different FSDV states and the availability of the actuators, should be programmed to actuate the Steering Actuator when the vehicle speed is different from zero in order to avoid the overloading of the actuator due to the maximum static sliding friction in turning the wheels when the vehicle is at rest.

## 2.4 ASS chosen solution

The three main solutions adopted for FSDV steering actuators are of the *hydraulic*, *pneumatic* or *electro-mechanical* type.

An hydraulic actuator has the same working function of a power-assyst steering system, the problem of this actuator technology is the lack of space inside the cockpit to accomodate a hydraulic pump driven by a motor and all the lines needed, for this reason hydraulic actuators are not so widespread for FSDV ASS. On the other hand, a pneumatic actuator is quite compact but the limited precision of this solution make it not suitable for a steering actuator application. Thus an electro-mechanical actuator represents the most adopted technology.

They are predominantantly of two types: *linear actuators* or *rotary actuators*. Both this two typologies are driven by an electric brushless DC motor or by a stepper motor, providing the translational/linear displacement or the rotational output required for the steering actuation. In the figure below (fig. 2.5) two examples of respectively rotary an linear steering actuators solutions adopted by two FSD teams are reported.

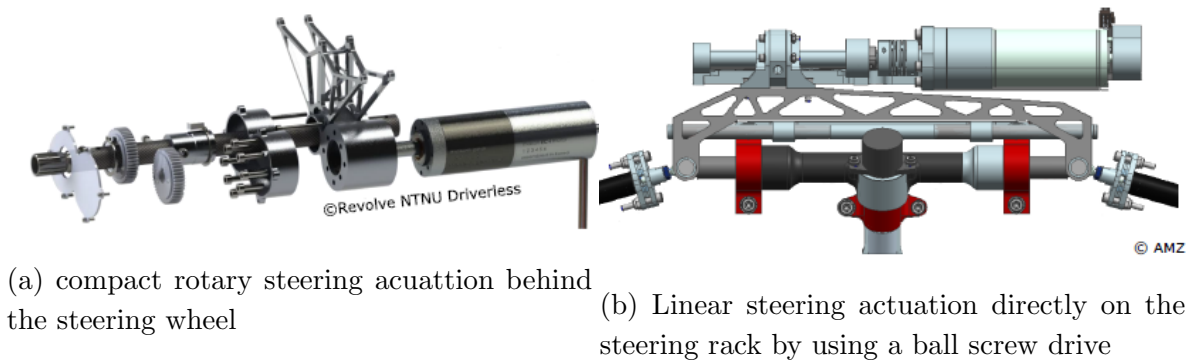


Figure 2.5: ASS actuators example [15]

The main requirements and constraints in choosing the most suitable ASS actuator layout for SC 19 are:

- Small rack assembly, rack length = 264 mm.
- Total rack travel required = 45 mm (considering a steering wheel working angle from -90 to +90 degrees).
- Actuation speed required = 45 mm/s (taking as target an actuation from full left to full right lock in one second, which represents a typical performance achieved by a driver that must be reproduced in the ASS).
- Reversibility is mandatory according to DV2.3.3 [17].

- Minimum load to turn the wheels in static condition (computed as the resisting torque of the wheel due to friction [16]):
  - For linear actuation = 700 Nm.
  - For rotary actuation = 15 Nm.
- Limited space availability for the actuator.
- Supply voltage = 12 V.

A *steering column actuation* layout is more complex compared to the rack actuation since it requires a series of modification of the SC 19 architecture in order to fulfill the regulation of reversibility and to accomodate the motor and the gearbox in an area of the vehicle with very limited space availability.

The *steering rack actuation*, on the other hand, can be placed directly beside the rack without any modification of the original assembly, furthermore the ball screw and the belt drive mechanism allow to use a smaller motor even without a gearbox, with respect to the rotary actuator solution.

In order to meet these requirements, the solution chosen for the SC 19 ASS consists in the linear actuation of the rack by using a ball screw shaft over which a ball nut is engaged. The power unit is a BLDC motor provided by *Maxon*. The nut is fixed and rotated using a belt drive in order to get the linear actuation through the screw, converted from the rotation of the Maxon motor.

The ball screw is of the *Rexroth* series provided by *Bosch*.

Starting from the required rack force  $F_{rack}$  it is possible to determine the required Torque at the ball screw  $T_{bs}$ , the motor output speed  $n$  (input for the ball screw) and the required motor power  $P_m$ , in order to select a suitable ball screw for the application:

$$T_{bs} = \frac{F_{rack}Lead}{2000\pi\eta_{bs}} = 397 \text{ mNm}$$

$$n = \frac{1000v_{bs}60}{Lead} = 1350 \text{ rpm}$$

$$P_m = \frac{F_{bs}v_{bs}}{\eta_{bs}} = 97 \text{ W}$$

The specifications of the Ball screw selected are:

Table 2.2: Selected ball screw specifications:

Required rack force ( $F_{rack}$ )	1000 N
Required linear actuation speed ( $v_{bs}$ )	0.045 m/s
Ball screw efficiency ( $\eta_{bs}$ )	0.8
Shaft diameter	8 mm
Threaded length	80 mm
Lead	2 mm

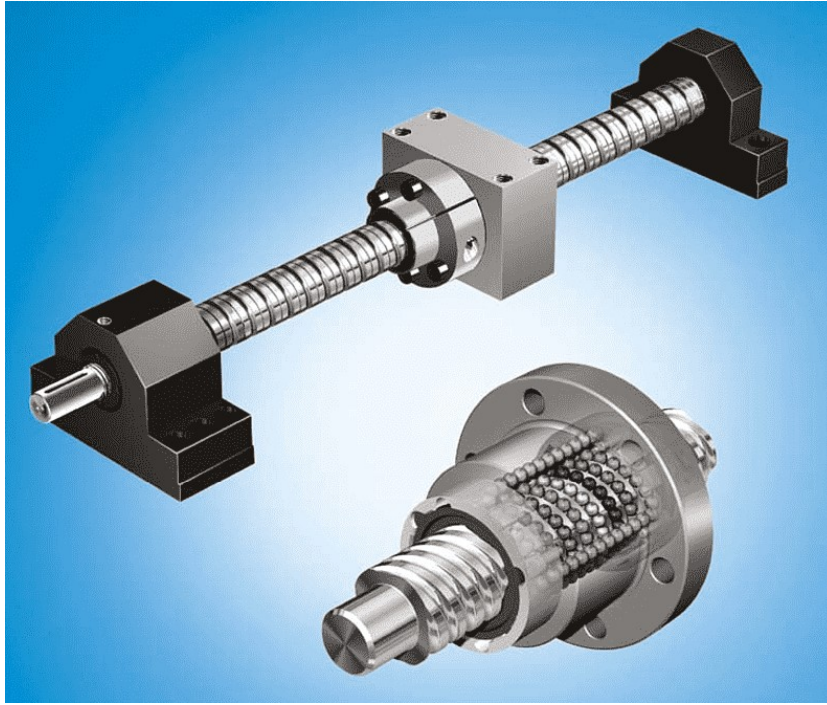


Figure 2.6: Bosch Rexroth ball screw

In order to satisfy these requirements, the motor selected is the *Maxon EC 60 Flat 150 W*. It presents a nominal speed of 3210 rpm and a nominal torque of 234 mNm. The secondary shaft over which the nut is engaged, is parallel to the original steering rack of SC 19, as can be seen from picture fig 2.10 (b), the mountings of the rack are modified in order to mount the motor, the belt drive and to accomodate the ball screw within a single assembly. The ends of the ball screw shaft (the secondary rack) are connected to two clevises which are rigidly coupled to the original rack ends. These clevises are connected on both ends to the steering tie rods, turning the front wheels. The ball screw drive is a reversible mechanism, allowing the FSDV to be driven in both driverless and with-driver mode without mechanically disconnecting the ASS, as stated in FSD regulations [17].

For the belt and pulley system, a standard HTD profile with a pitch of 3 mm and a

belt width of 9 mm has been selected; while the bearing is a standard *SKF Four-point contact ball bearing*.

The complete system including the custom support structure has been manufactured by *Officina Meccanica Massola* in Rivalta, Torino.

The list of components with the relative weight for the Autonomous Steering Actuator is reported in the table below:

Table 2.3: Components weight:

<i>Component</i>	<i>Weight [g]</i>
Ball screw (Bosch Rexroth)	100
Nut	55
Pulley/gears (2)	50
Bearing support	50
SKF Bearing	88
Maxon motor	350
End joints (2)	80
Sliding Bushing	10
Support structure	400
Seeger rings (2)	10
Toothed belt	20
Total weight	1213

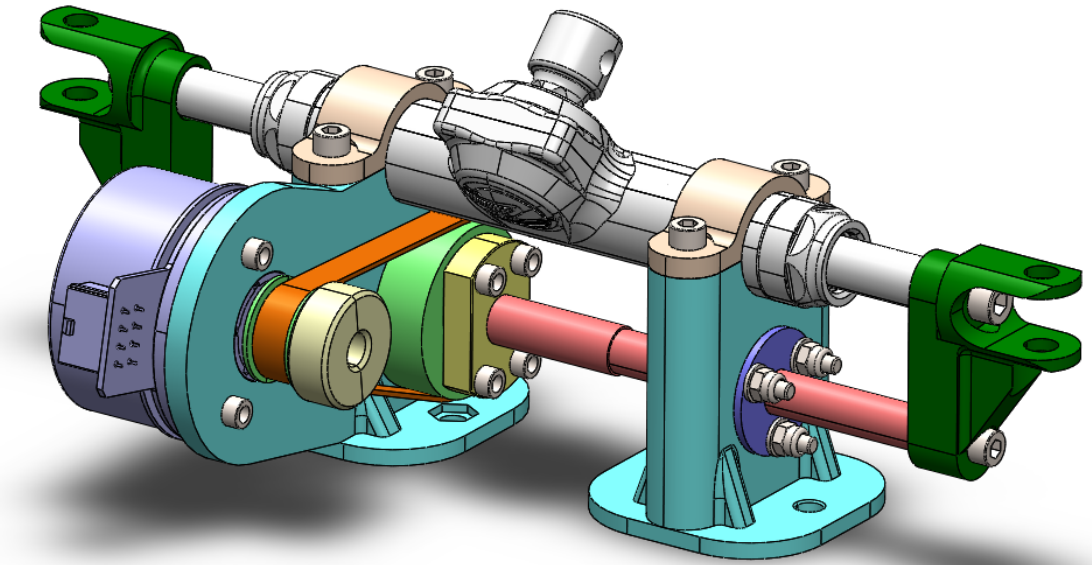


Figure 2.7: Autonomous Steering System SC 19 CAD

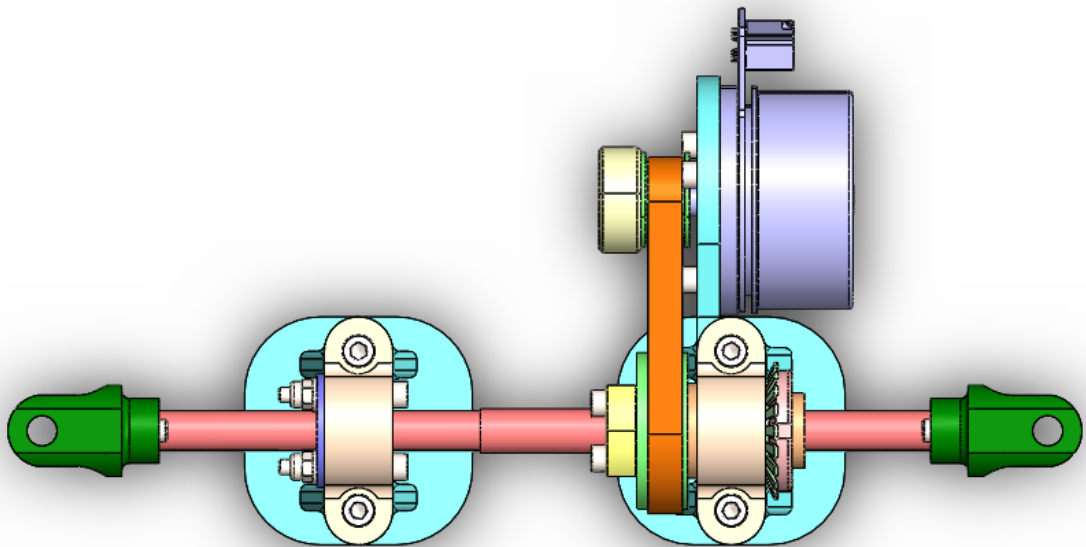


Figure 2.8: ASS top view CAD

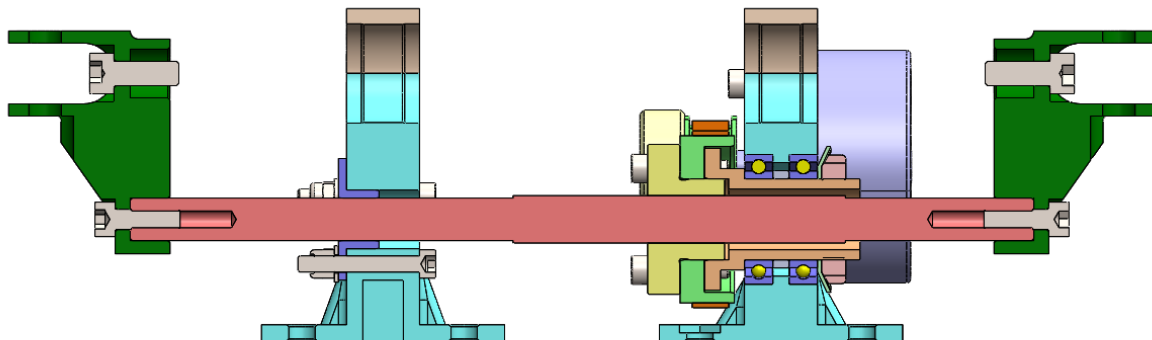
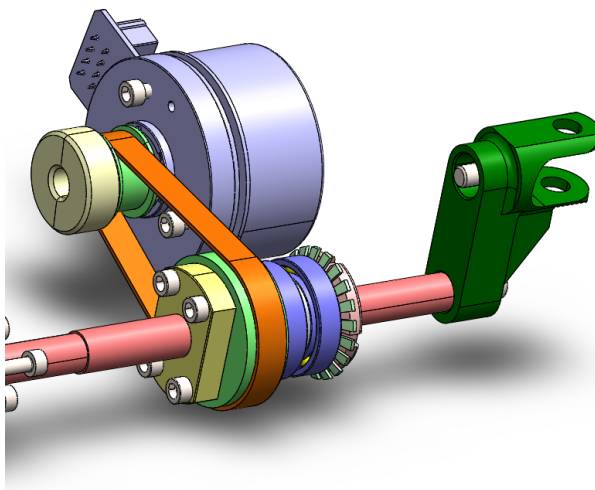
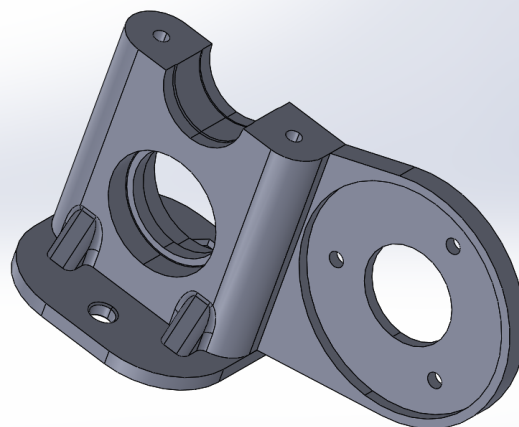


Figure 2.9: ASS X-Section of the support structure CAD



(a) ASS Ball screw with nut assembly and bearing CAD detail



(b) ASS support structure CAD detail

Figure 2.10: ASS CAD details



---

## CHAPTER 3

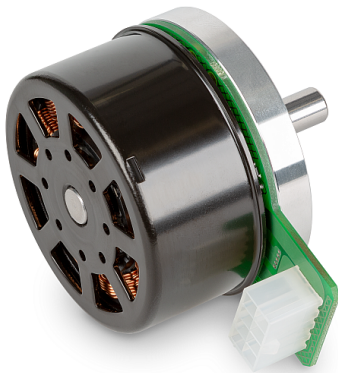
---

# Systems modelling

This chapter deals with the modelling of each subsystem in *Matlab/Simulink* environment. The main goal is to design, to model and to tune the controller of the ASS before implementing it on the actual hardware components in order to obtain the performances required.

The rack ball screw of the Steering Actuator is driven by the brushless DC motor *Maxon EC 60 flat*, for this reason before focusing on the model of the motor, an overview on BLDC motors and on how to control them will be provided.

The motor is connected via CAN to the digital positioning controller *EPOS 4 70/15* provided by *Maxon*. In the next sections the control strategies will be analyzed and modeled. The parameters found in the controller tuning and optimization phase in simulation environment will be finally implemented on the GUI software of the EPOS controller called *EPOS Studio*.



(a) Maxon EC 60 FLAT



(b) EPOS 4 70/15

Figure 3.1: Maxon motor and EPOS controller

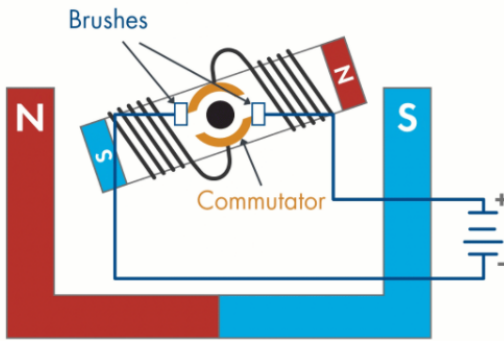


### 3.1 BLDC motors

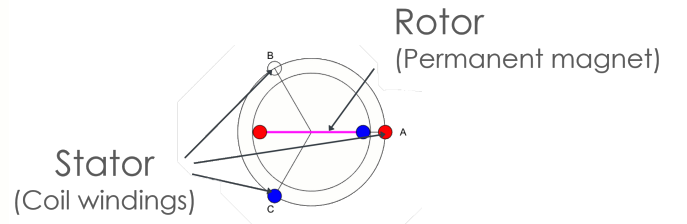
DC motors applications range from very simple devices to complex equipments like robotics. Brushless DC motors offer many advantages over brushed ones. BLDC motors present an higher efficiency and require lower maintenance, for these reasons they have replaced brushed motors in many applications. Both brushed and BLDC motors are based on a similar principle, in which the rotational motion is generated by the attraction and repulsion of magnetic poles of permanents and electromagnets. However, these two typology of electric motors are controlled in a very different way, since BLDCs require a controller able to convert a single DC power to three-phase voltage, while brushed motor are controlled only by regulating the DC power.

These differences are due to the operation principles of the two types of DC motors:

- *Traditional Brushed DC Motors:* the DC current passing through the coil windings of the rotor generates an electromagnet with poles interacting with the poles of the fixed permanent magnet, the stator, making the rotor spin. After half rotation of the rotor, to keep it spinning, the rotor poles are flipped by switching the polarity of the current in the coils. This process is called commutation and it is performed mechanically by the brushes. Due to the physical contact, brushes are subjected to wear.
- *Brushless DC Motors:* in BLDC motors the mechanical commutation is replaced by electronic commutation. In this type of motors the permanent magnets are installed in the rotor and the coil windings are the stator. The magnet arrangements can be different with different numbers of windings in the stator and multiple pole pairs in the rotor [18].



(a) Brushed DC motor working scheme



(b) BLDC motor working scheme

Figure 3.2: Types of DC motors [15]

*Permanent magnet synchronous machine*, or *PMSM*, have a similar structure of BLDC motors, since both have permanent magnets in the rotor and both are referred as synchronous motors in which the rotor is synchronized with the stator magnetic field, making the rotor turning at the same speed. The key difference between BLDCs and PMSMs is the back EMF (electromotive force) profile. BLDC motors show a trapezoidal back EMF shape and they are controlled by trapezoidal commutation, while PMSMs have sinusoidal shape back EMF and they require a field oriented control. This aspect determines the kind of algorithm necessary to optimally control it.

Considering the configuration of the rotor with a single pole pair and the stator made up of three coils spaced at 120 degrees, if a voltage is applied across two phases, the rotor starts to move in order to align itself with the stator magnetic field created. In this configuration, the coil pairs can be energized in six possible ways, thus commutating the current every 60 degrees, the motor will continuously spin, this type of control is the *six-step commutation*. The stator can be equipped with more pole pairs, requiring to perform the commutation more often at the right times with the correct phase, knowing the exact position of the rotor, measured by the Hall sensors.

Generally the phases in a BLDC are switched in order to have the angle between the rotor and the stator magnetic field close to 90 degrees, because at this angle the maximum torque occurs. In BLDC motors it is impossible to constantly achieve 90 degrees using six-step commutation, thus the angle between fields usually fluctuates between 60 and 120 degrees.

A *three-phase inverter* is used to control the phases during six-step commutation, switching between positive and negative current. The three-phase inverter keeps the motor rotating at constant speed, a varying motor speed can be achieved by adjusting the applied voltage or by performing pulse-width modulation (PWM).

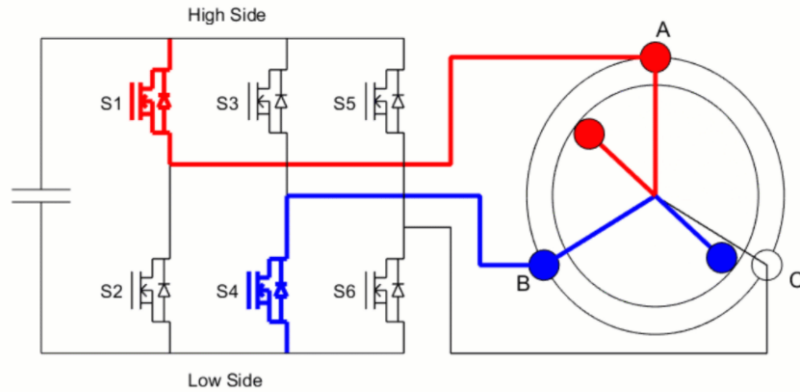


Figure 3.3: Three-phase inverter controlling BLDC motor [18]

### 3.2 Motor modelling

In order to perform the control strategy suitable for the performances required, which will be developed in the next sections, the BLDC motor can be modelled as a standard DC motor.

Neglecting the three phase system and the mutual inductance between the stator windings, the electrical dynamic model of the BLDC can be described as:

$$v_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + e(t)$$

Where  $v_a$  is the motor terminal voltage,  $i_a$  is the motor armature current,  $L_a$  is the terminal inductance phase to phase,  $R_a$  is the terminal resistance phase to phase and  $e$  is the motor back-EMF, given by:

$$e(t) = k_e \omega(t)$$

$k_e$  is the speed constant, measured in  $\frac{V}{rad/s}$ .

The mechanical dynamic equation of the system is:

$$J \frac{d\omega(t)}{dt} + r\omega(t) + T_{res}(t) = T_m(t)$$

where  $J$  is the inertia of the system which comprises both the inertia of the rotor and the inertia of the equivalent load of the entire system at the motor, which will be computed in the next sections of this chapter.  $r$  is the torque/speed gradient i.e. the damping coefficient of the system.

$T_{res}$  is the resisting torque generated by the static and dynamic friction between tire and road surfaces while  $T_m$  is the electromagnetic motor torque, given by:

$$T_m(t) = k_t i_a(t)$$

$k_t$  is the torque constant of the system, measured in  $\frac{Nm}{A}$ .

If the parameters are expressed in the metric system augmented for electric quantities, the torque constant  $k_t$  and the speed constant  $k_e$ , coincide numerically.

The damping coefficient is computed as no load torque divided by no load speed:

$$r = \frac{k_t I_0}{n_0}$$

$I_0$  is the no load current of the motor and  $n_0$  is the no load speed of the motor. All the parameters necessary to model the model are provided by *Maxon* for the specific BLDC motor in analysis and are reported in the table below.

Table 3.1: Maxon EC 60 Flat specifications

Resistance ( $R_a$ )	0.293 $\Omega$
Inductance ( $L_a$ )	0.279 mH
Torque constant ( $k_t$ )	52.5 mNm/A
Speed constant ( $k_e$ )	52.5 mV/(rad/s)
Torque/speed gradient ( $r$ )	$5.78 * 10^{-5}$ Nm/(rad/s)
Rotor inertia ( $J_{rotor}$ )	$8.1 * 10^{-5}$ Kg $m^2$
No load current ( $I_0$ )	497 mA
No load speed ( $n_0$ )	4300 rpm

By reporting the two equations in the Laplace domain it is possible to define the transfer function between the armature voltage as input and the rotor velocity as output:

$$\frac{\omega(s)}{V_a(s)} = \frac{k_t}{(R_a + sL_a)(r + sJ) + k_e k_t}$$

The system has been modeled on *Matlab/Simulink* in order to have the voltage as input and as output the armature current, the rotor velocity (from which, integrating, the angular position  $\theta(t)$  can be obtained) and the acceleration  $\dot{\omega}$  computed by the torque. The representation with this input and this output is needed for the control performed. The Simulink model without the resisting friction torque is reported below in fig.3.4.

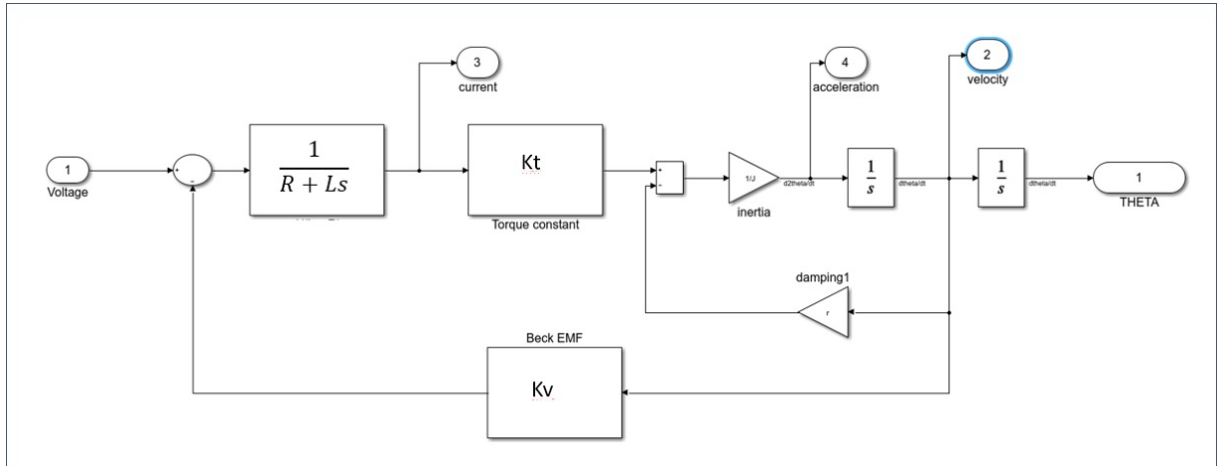


Figure 3.4: Simulink model of the BLDC motor

### 3.3 Equivalent Inertia Computation

The total inertia of the system is given by two contributes:

$$J = J_{rotor} + J_{load}$$

The inertia of the rotor  $J_{rotor}$  is tabulated in the Maxon datasheet, while the inertia of the load  $J_{load}$  can be computed as the equivalent inertia of the steering wheel, column, racks, ball screw and vehicle wheels reported to the motor.

The moment of inertia is computed with an energetic approach, considering the kinetic energy of the main subsystems moved by the motor. The rotational energy of the steering wheel and of the steering column rotating about their axes (considering that are linked by a double cardan homokinetic joint) is equal to the translational energy of the steering rack:

$$\frac{1}{2} J_s \omega_s = \frac{1}{2} m_{s \rightarrow r1} v_{r1}^2$$

Where  $J_s$  is the moment of inertia of the steering wheel, column and pinion subassembly about their axes. These value of Inertia are obtained by the vehicle CAD model in *Solidworks* as  $I_{zz}$  for the stering wheel and  $I_{xx}$  for the column, from the Inertia tensor provided by the software.

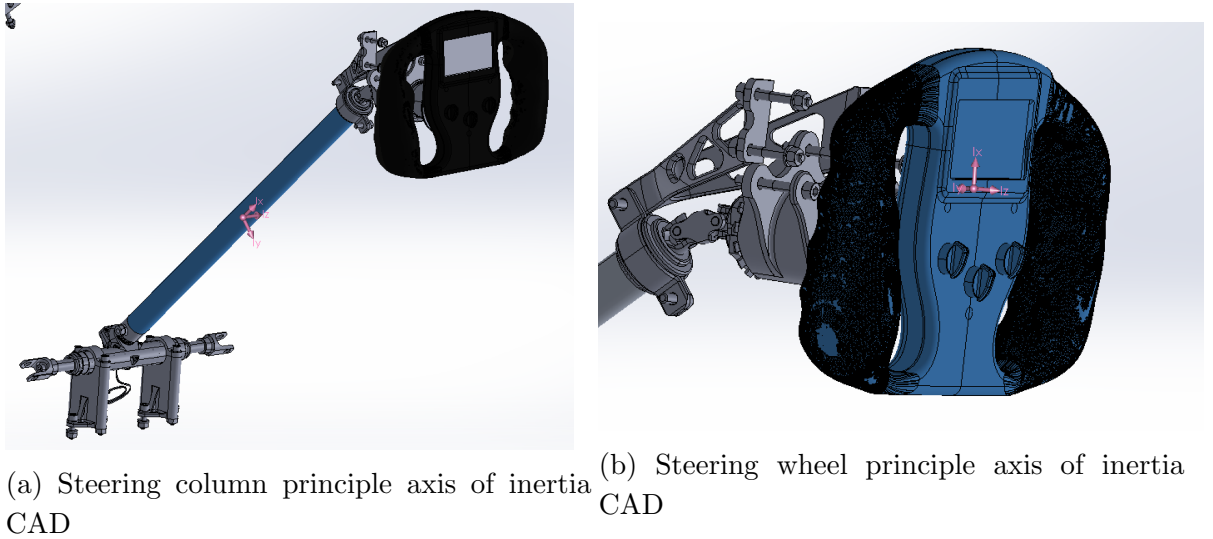


Figure 3.5: CAD models for computation of the inertia of the steering wheel and column subassemblies

Thus the equivalent mass of the steering wheel and column reported to the rack is:

$$m_{s \rightarrow r1} = \frac{J_s}{\tau_{s \rightarrow r1}^2}$$

$\tau_{s \rightarrow r1}$  is the ratio between the linear velocity of the rack and the angular velocity of the steering wheel. It refers to the C-Factor of the *Zedaro* rack installed on the SC 19.

The total travel of the rack is  $42.75\text{mm}$ . The steering wheel working angle is  $180^\circ$ , in this way for a complete rotation the displacement is  $85.5\text{mm}$  and the transimission ratio is  $\tau_{s \rightarrow r1} = 0.014$ .

The equivalent mass of the rack is:

$$m_{r1}^* = m_{r1} + m_{s \rightarrow r1} = 19.56\text{ kg}$$

Where  $m_{r1} = 65\text{g}$  is the mass of the rack available on the data-sheet.

The kinetic energy of the wheel rotating about the kingpin axle is equal to the kinetic energy of the ball screw rack of the steering actuator:

$$\frac{1}{2}J_w\omega_w = \frac{1}{2}m_{s \rightarrow r2}v_{r2}^2$$

The parameters are:

- $J_w$  is the moment of inertia of the wheel rotating about the kingpin axis. It includes the tire, the magnesium rim and the electric motor. The inertia is computed by the CAD model in Solidworks, by building up the kingpin axes. The motor is considered by using the Huygens-Steiner theorem.

$$J_w = I_{zz} + m_{mot}r_{mot}^2 = 0.4\text{ kgm}^2$$

$m_{mot} = 12\text{ kg}$  is the mass of the electric motor,  $r_{mot} = 13.12\text{ mm}$  is the normal distance between the center of mass of the motor and the kingpin axis.

- $\omega_w$  is the angular velocity of the wheel around the kingpin axis.
- $v_{r2}$  is the linear velocity of the ball screw.  $v_{r2} = \omega_w b$  where  $b = 49\text{ mm}$  is the longitudinal distance between the tie rod attachment point and the kingpin axis.

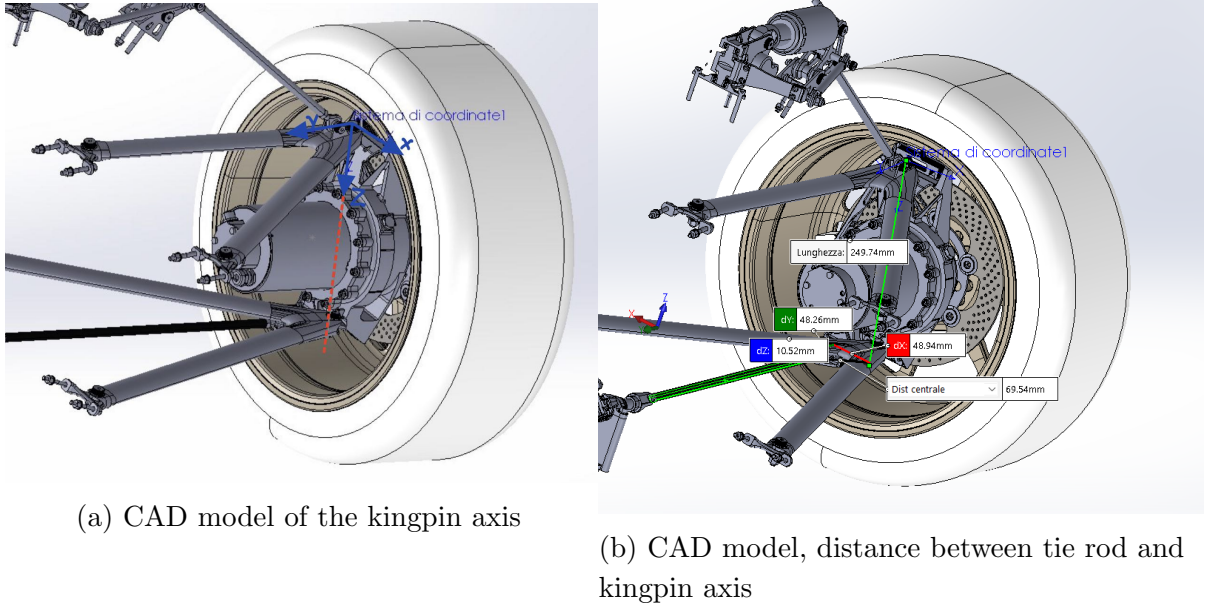


Figure 3.6: CAD models for computation of the inertia of the wheel

Hence the equivalent mass of the wheel at the ball screw is:

$$m_{w \rightarrow r2}^* = \frac{J_w}{b^2} = 160 \text{ kg}$$

The total equivalent mass of the ball screw is given by:

$$m_{r2}^* = m_{r1}^* + m_{w \rightarrow r2}^* + m_{r2} = 180.2 \text{ kg}$$

By equaling the kinetic energy of the rotor of the BLDC motor to the energy of the ball screw, the total equivalent inertia of the motor load  $J_{r2 \rightarrow m}^*$  is obtained:

$$\frac{1}{2} J_{r2 \rightarrow m}^* \omega_m^2 = \frac{1}{2} m_{r2}^* v_{r2}^2$$

$$J_{r2 \rightarrow m}^* = \tau_{bs}^2 \tau_{belt}^2 m_{r2}^* = 5.35 \cdot 10^{-6} \text{ kgm}^2$$

- $\tau_{bs} = \frac{v_{r2}}{\omega_{belt}} = 3.183 \cdot 10^{-4} \text{ m/rad}$  is the transmission ratio of the ball screw. The Lead is 2 mm of ball screw translation for a complete rotation.
- $\tau_{belt} \frac{\omega_{belt}}{\omega_m} = 0.5$  is the transmission ratio of the belt pulley between the motor and the ball screw.

### 3.4 Steering resistance torque considering tire/pavement friction

Dealing with the performances required to the rotary actuator, it is important to take into account the resisting torque due to friction between tire and ground.

This friction torque can be modeled in the worst case scenario of static friction, with stationary vehicle. Considering the data below:

- Kerb weight of SC 19 = 190 kg (considering the additional AS components and without the driver as stated in DV2.2.6 of Formula Student Rulebook [17]).
- Weight distribution of the car = 45:55.
- Quarter car mass of the front axle,  $m_{quarter} = 42.75 \text{ kg}$ .
- Longitudinal tie rod offset,  $b = 49 \text{ mm}$ .
- Front Caster angle = 6.9 deg.
- Mechanical/Caster trail  $c = 17.5 \text{ mm}$ .

Source: Race Car Vehicle Dynamics by Milliken

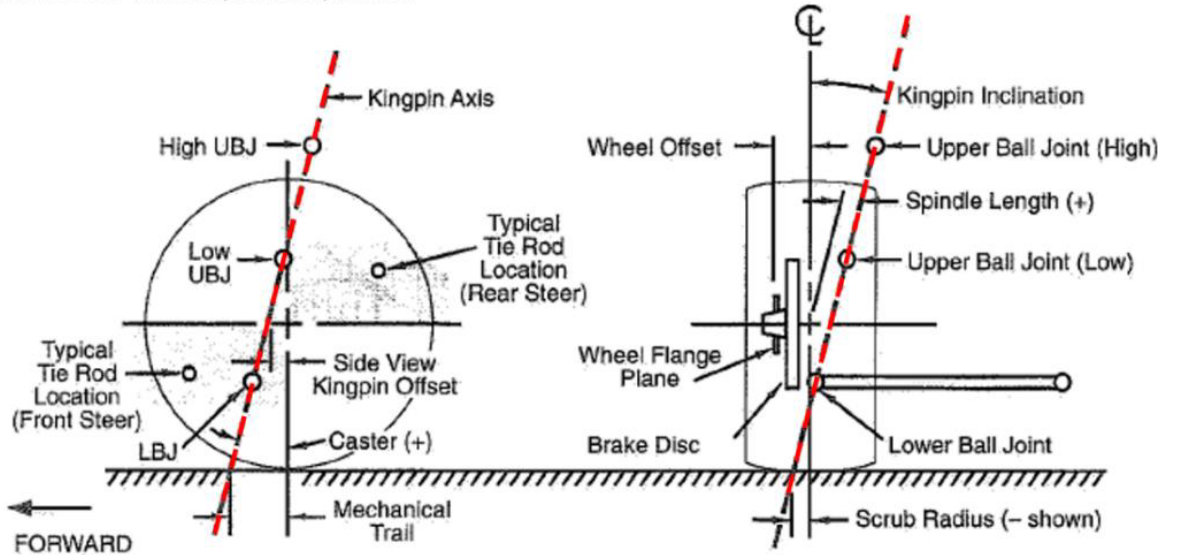


Figure 3.7: Kingpin geometry of a quarter car [19]

The input torque from the ground to the wheel is equal to the lateral push of the tie rod :

$$F_y c = F_{tierod} b = 26.42 \text{ Nm}$$

$F_y = \mu_y F_z$  is the side force on the wheel due to friction.

$\mu_y = 0.9$  is the equivalent static friction coefficient under normal driving conditions on dry asphalt.

$F_z = m_{quarter} g$  is the vertical load on the wheel.

The force acting on the rack is given by the sum of the forces of the tierod for the two front wheels.

$$F_{rack} = 2 \frac{F_y c}{b} = 259.08 \text{ N}$$



From this load it is possible to determine the torque on the ball screw:

$$T_{ballscrew} = \frac{Lead_{bs} F_{rack}}{2000\pi\eta_{bs}} = 0.255 Nm$$

$\eta_{bs} = 0.8$  is the efficiency of the ball screw and the *Lead* is 2 mm.

Considering the transmission ratio of the belt pulley, it is possible to obtain the final resisting torque at the motor:

$$T_{res} = \tau_{belt} T_{ballscrew} = 0.51 Nm$$

The resisting torque has been included in the mechanical equation of the motor model implemented in Simulink.

$$J \frac{d\omega(t)}{dt} + r\omega(t) + T_{res}(t) = T_m(t)$$

In order to consider both static and kinetic friction, a switch command has been used, triggered by the motor's velocity.

According to the study reported in [24], from the finite element analysis of tire and ground interaction to obtain steering friction force as function of the deflection angle of the tire at different running speeds, a negative exponential behaviour of equivalent friction coefficient with running speed emerges. This fitting function of the steering resistance torque equivalent friction coefficient is reported below, all the parameters have been empirically obtained from the study quoted in [24].

$$\mu_d = 0.451 e^{-0.4603 V} + 0.2376$$

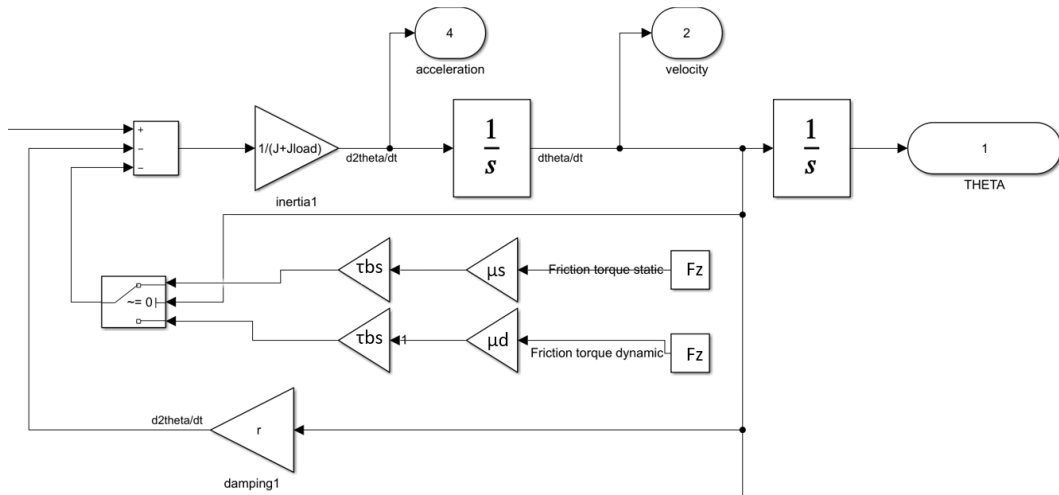


Figure 3.8: Simulink model of the friction torque

### 3.5 Self aligning moment and steering resistance torque in dynamic conditions

In dynamic conditions, due to the centrifugal effect a lateral force  $F_y$  is generated. This cornering force is directly linked with the compliance of the tire, that allows the tread to move, relatively to the center of the wheel, with the same velocity of the ground [25]. Thus the shape of the contact zone is distorted, generating a sideslip angle  $\alpha$ . The lateral force varies with alpha, with a linear behaviour between  $-4^\circ$  and  $4^\circ$  and it is not applied directly in the center of the contact patch of the tire. This longitudinal offset is the pneumatic trail  $t(\alpha)$ .

The self aligning moment is generated:

$$M_z = F_y t$$

This self aligning torque acts as a resistance torque to the steering and has to be modeled and taken into account in the model of the system as a load to the motor, in order to check the performances in dynamic conditions.

The vehicle's dynamic team of Squadra Corse evaluated the performances of the Pirelli 185/40R13 tires. In the following figure the self aligning torque on the sideslip angle is reported.

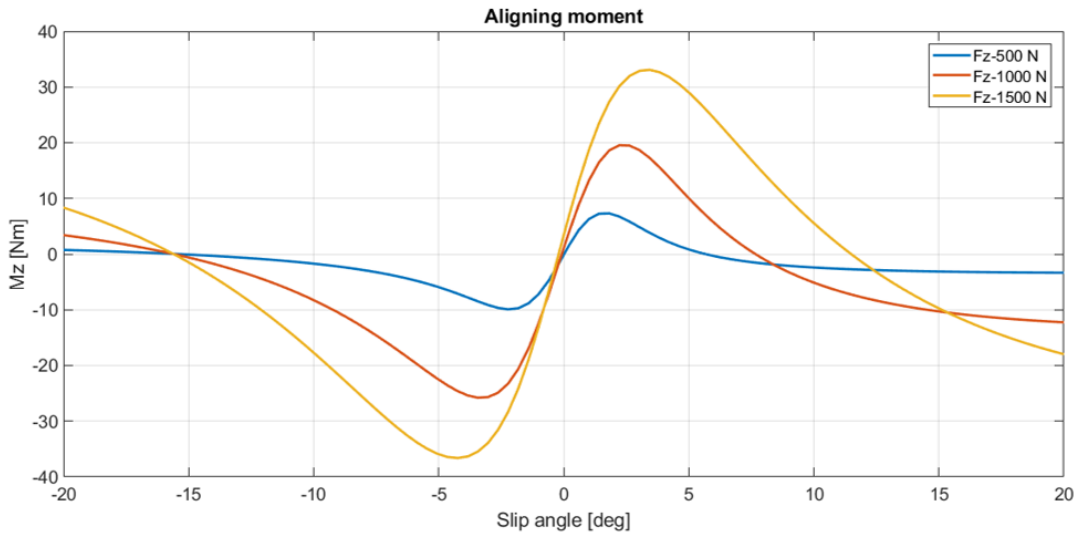


Figure 3.9: SC 19 self aligning moment

For  $\alpha$  between  $-2^\circ$  and  $2^\circ$ , the self aligning moment can be computed linearly as:

$$M_z = (M_{z0})_\alpha \alpha$$

Where the coefficient  $M_{z0}$  can be computed graphically as the slope in the origin at the vertical load.

Neglecting the aerodynamic terms, due to the low speed of the considered case, the vertical force acting on the front wheel can be computed as:

$$F_{z1} = \frac{1}{2} \left( \frac{m_s g (l - a)}{l} + m_{nsfront} g \right) = 590N$$

Where  $m_s = 190kg$  is the sprung mass of the vehicle;  $m_{nsfront}$  is the front unsprung mass;  $l = 1.525m$  is the length of the vehicle and  $a = 0.839m$  is the distance between the front axle and the center of gravity projection.

By computing the slope in the origin of  $M_z$  curve interpolating for the considered the load,  $M_{z0} = 390Nm/rad$ .

In order to determine the sideslip angle  $\alpha$  in a dynamic scenario, a 3 d.o.f. rigid vehicle model is adopted [25].

The main assumptions are:

- The motion of the vehicle is described only in the XY plane;
- The vehicle is an hyperstatic structure. The vertical tire-ground forces are undefined if the structure compliance is not accounted for. This is bypassed if the so called "bicycle" or monotrack model is adopted;
- the pitch, roll and the movement of sprung mass with respect to unsprung are neglected;
- The sideslip angle of the vehicle  $\beta$  and the steering angle  $\delta$  are small, thus the trigonometric functions and the equations of motion can be linearized.

The equations of motion (respectively along the y direction and about the z axes) describing the lateral dynamic of the vehicle are:

$$mV(\dot{\beta} + r) + m\dot{V}\beta = Y_{\beta}\beta + Y_r r + Y_{\delta}\delta + F_{ye}$$

$$J_z \dot{r} = N_{\beta}\beta + N_r r + N_{\delta}\delta + M_{ze}$$

$r$  is the yaw rate of the vehicle; the yaw moment of inertia of the vehicle is  $J_z = 88.3kgm^2$ ; the coefficients  $Y_i$  and  $N_i$  are the derivatives of stability, functions of geometric and aerodynamic parameters and of the cornering stiffness; The terms  $F_{ye}$  and  $M_{ze}$  indicate the generalized external forces that cannot be expressed as functions of  $\beta$  and  $r$ .

All the terms have been implemented on Matlab and the equation of motions have been expressed in state space form. The input of the system are  $F_{ye}$ ,  $M_{ze}$  and  $\delta$ . By inserting in Matlab the dynamic and the gain matrices representing the linearized equations of motion and by using the State Space Simulink block, the values of  $\beta$  and  $r$  can be obtained.

These values are necessary to determine the side slip angle of the front tires as [25]:

$$\alpha_1 = \beta + \frac{a}{V}r - \delta$$

By multiplying the sideslip angle by the coefficient of SC 19 tire  $M_{z0}$ , the self aligning torque is obtained. As for the resistance torque due to friction, the self aligning torque has to be included in the mechanical equation of the motor model in Simulink:

$$J \frac{d\omega(t)}{dt} + r\omega(t) + T_{res\ friction}(t) + T_{res\ Mz\ dynamic} = T_m(t)$$

The resistant torque due to the self aligning torque is determined by equalling it to the lateral push of the tie rod and by considering the force acting on the rack (for both wheels), the torque applied to the ball screw and the the belt transmission ratio.

$$T_{res\ Mz\ dynamic} = \frac{2\tau_{bs}\ Lead\ \frac{Mz}{b}}{2000\pi\eta_{bs}}$$

In the following figure the resistant torque due to the self aligning moment acting on the motor is reported for a sinusoidal steer manouvre ( $\delta = 5^\circ$ ) at a constant speed of 30 km/h.

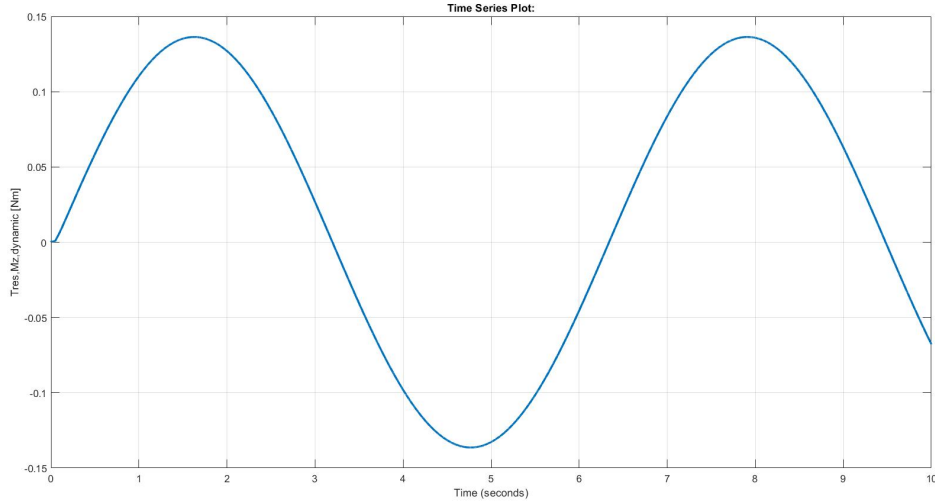


Figure 3.10: Motor resistant torque due to the self aligning moment for a sinusoidal steering manouvre at 30 km/h

It emerges that the values of the resistant torque due to friction in static conditions are larger than the dynamic resistance due to the self aligning torque, for this reason it is important to consider static steering as the worst case scenario for designing and tuning the autonomous steering system controller.

### 3.6 Motor controller architecture

The control strategy of the Maxon BLDC motor adopted, has been designed and modeled basing on the EPOS4 controller architecture [20].

The EPOS4 controller provides the possibility of working by using several operating modes, which are based on three control loops strategies:

- Position controller.
- Velocity controller.
- Current controller.

Position or velocity regulation is only used in operating modes based on position or velocity respectively. The current control loop is used in all the modes.

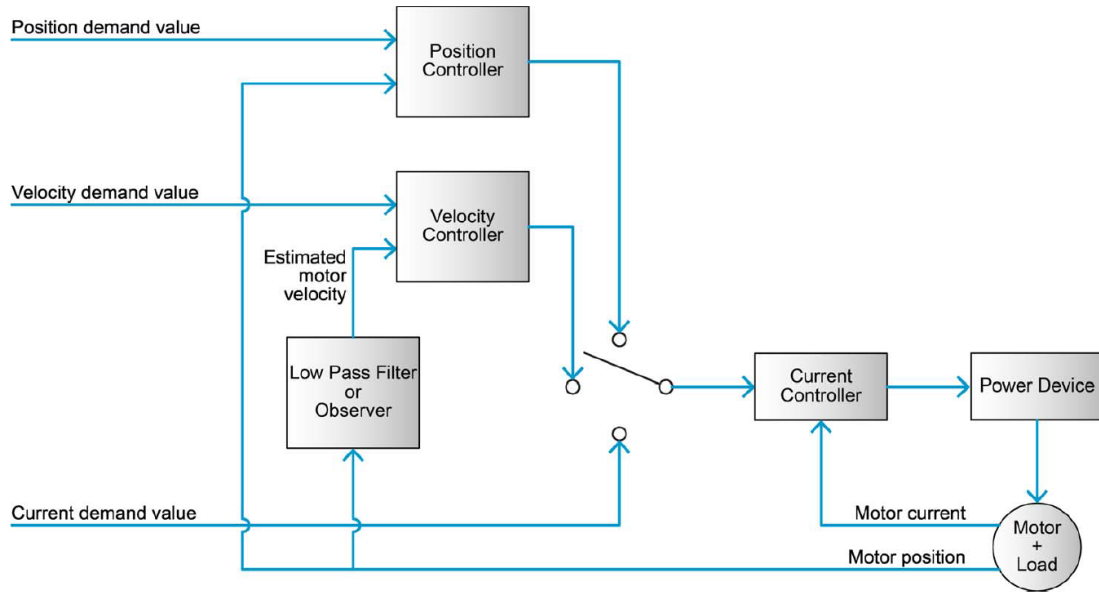


Figure 3.11: Overview on the controller architecture

The regulation method has to be chosen and tuned basing on the mode of operation selected between the ones available on the EPOS control unit:

- *Profile Position Mode* (PPM) - A target position is provided to the trajectory generator. It will generate a position demand value basing on the predefined trajectories according to the velocity and acceleration profile provided.
- *Homing Mode* (HM) - It refers to various procedures to reach the reference zero point position. These methods rely on the position sensor installed on the motor. The Maxon EC 60 is not equipped with an absolut digital incremental encoder, but only with a digital Hall Sensor.
- *Profile Velocity Mode* (PVM) - It controls the drive's velocity without particular focus on the position. It includes a velocity trajectory generator.

- *Cyclic Synchronous Position Mode (CSP)* - The trajectory generator is located in the control unit, and not in the drive device. It provides a target position cyclically and synchronously, controlling only the position and the torque. The velocity and acceleration is limited by the parameters of the motor setted in the startup configuration phase.
- *Cyclic Synchronous Velocity Mode (CSV)* - The target velocity is provided cyclically and synchronously, the drive controls only the velocity and the torque, not the position.
- *Cyclic Synchronous Torque Mode (CST)* - A target torque is provided to drive the device in cyclic synchronous manner. The torque control drives the power stage without focusing on velocity or position information.

The most suitable operating mode, for the ASS application, is the Cyclic Synchronous Position Mode. The dSpace will provide, at a given frequency, through the CAN Bus the target position to the EPOS, updating it with the data from the motor Hall sensor and from the Steering Encoder. Thus the EPOS will perform the position control loop providing the current set to the internal current control loop which will drive the motor inverter in order to reach the target position.

### 3.7 PID controllers

Before dealing with the modelling and tuning of the current and position controller it is important to give a basic theoretical background on PID controllers.

Proportional-Integral-Derivative (PID) control is the most common control algorithm used in industrial applications. This controller presents robust performances in a wide range of operating conditions. PID algorithm basically consists of three basic coefficient: proportional, integral and derivative which are tuned to optimize the response of the closed loop system.

In a typical control system, the *process variable* is the system parameter that needs to be controlled. In the application in analysis, the process variable is the angular position for the external control loop and the current set for the internal control loop. A sensor is used to measure the variable, and provide the feedback i.e. the actual position or actual current. The set point and the feedback process variable are then compared in the compensator in order to determine the actuator output needed to drive the plant. The closed loop control system has to be designed to minimize the disturbances affecting the process variable.

Control system performances are usually measured by applying a step function at the command input variable and by measuring the response. The response is determined by waveform characteristics: the *rise time* is the time taken by the system to go from

10% to 90% of the steady state value. *Percent overshoot* is the amount that the input overshoots the steady state expressed in percentage. *Settling time* is the time required for the input variable to settle within a 5% of the steady state value. The *steady state error* is the final difference between the process variable and the target point.

In order to improve the step response characteristics the PID response has to be optimized.

- The *Proportional* component depends on the difference between the target and the process variable. Increasing the proportional gain the speed of the control system response will increase. A too large proportional gain can lead to oscillations in the system.
- The *Integral* component sums the error term over time. Integral gain mainly affects the steady-state error, if too large can lead to the saturation of the system.
- The *Derivative* component causes the output to decrease if the process variable is increasing rapidly, its response is proportional to the rate of change of the process variable. If the feedback is affected by noise and the control loop is too slow, the derivative term can lead to instability in the system [21]. The main effect of the derivative term is to reduce the overshoot.

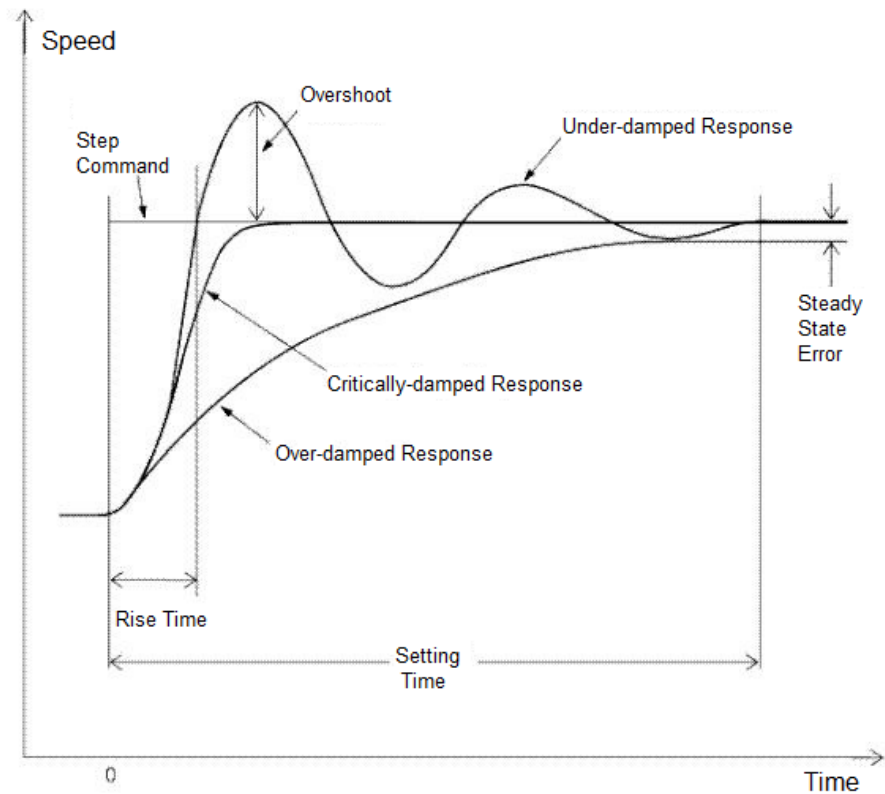


Figure 3.12: Step response characteristics [21]

The main tuning strategies adopted for PID controllers are the *pole placement technique* (which will be adopted in the current controller model described in the next section), the *loop shaping* and the *Ziegler-Nichols* method based on the oscillation method.

Beside these standard techniques, the tuning of PID controller can be performed by using automatic tuning tools or by tweaking the gains manually and checking the step performances.

### 3.8 PI Current Controller

During a movement with a driven system, the torque must be controlled. Therefore, as a principle regulation structure, the PI current based control has been implemented. The derivative term is not present in order to prevent the instability of the system due to the noise in the signal provided by the current sensor in the motor.

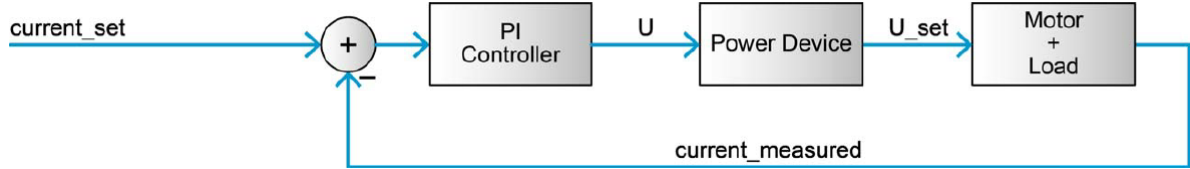


Figure 3.13: Current regulator control architecture [20]

In order to prevent degradation of the control performance when the control input is at the limit value for long time, an *anti-windup* algorithm is implemented. This algorithm is inserted in the controller model to avoid that the integral term of the PI controller takes values larger than the ones bound on the control input.

The *transport delay* of the current regulation loop is also taken into account.

The system is implemented on *Simulink*. The input is the current set obtained by the position control loop, the voltage set is provided to the motor/plant model as input. The anti-windup is modeled as a saturation taking into account the nominal voltage of the Maxon motor (24 V).



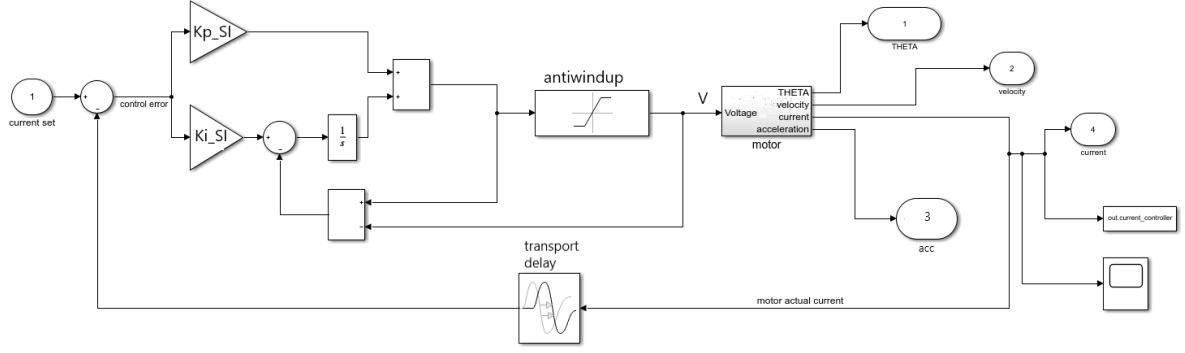


Figure 3.14: PI current controller Simulink model

- $KP_{SI}$  is the controller P gain expressed in SI units  $\frac{V}{A}$ .
- $KI_{SI}$  is the controller I gain expressed in  $\frac{V}{As}$ .
- The transport delay is of  $0.06\ ms$ .

The current controller parameters can be used in analytical or numerical simulations via the following transfer function:

$$C_{current}(s) = KP_{SI} + \frac{KI_{SI}}{s}$$

### 3.8.1 PI Current Controller tuning

The poles of a system affect the time response, thus the pole placement technique consists in choosing the controller gains ( $K_P$  and  $K_I$ ) so that poles of the closed-loop system meet given requirements. Since the system can be analysed as a linear canonical first order system, an algebraic method can be adopted to define precise relationships between poles and the shape of the response.

The aim of the pole placement is to make the poles of the motor ( $E(s)$ ) coinciding with the poles of the closed loop system ( $C(s)E(s)$ ).

The transfer function of the motor is:

$$E(s) = \frac{1}{1 + s\tau}$$

Where  $\tau = \frac{L}{R}$  is the electric time constant equal to the the windings inductance divided by the motor's resistance. The transfer function of the PI controller is:

$$C(s) = K_P + \frac{K_I}{s}$$

The closed loop system can be represented as:

$$\frac{V}{i_{ref}} = \frac{CE}{1 + CE} = \frac{(K_P + \frac{K_I}{s})(\frac{1}{s\tau})}{1 + (K_P + \frac{K_I}{s})(\frac{1}{s\tau+1})}$$

The pole of the motor is:

$$s = -\frac{1}{\tau}$$

Thus the poles of the closed loop system are equaled to the the poles of the motor in order to find the relation between the gains of the controller:

$$K_P = K_I\tau$$

With the integral term  $K_i$  it is possible to define the bandwith of the system.

Thus the gains defined are:

$$K_P = 2000 \frac{mV}{A}$$

$$K_I = 2100 \frac{mV}{Ams}$$

The frequency response and the time step response of the system are reported in the figures below. The performances obtained of the systems are consistent with the application.

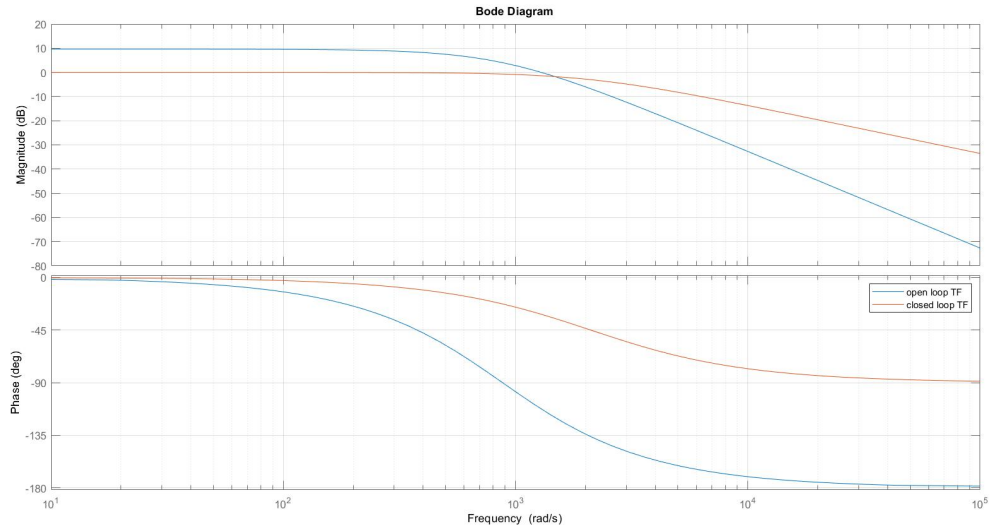


Figure 3.15: Bode diagram of the closed loop PI current control system

The cutoff frequency of the plotted systems (the first frequency where the gains drops below 3dB of its DC value) are of 605.7rad/s for the open loop system (the motor) and 2.09 10<sup>3</sup>rad/s for the closed loop system.

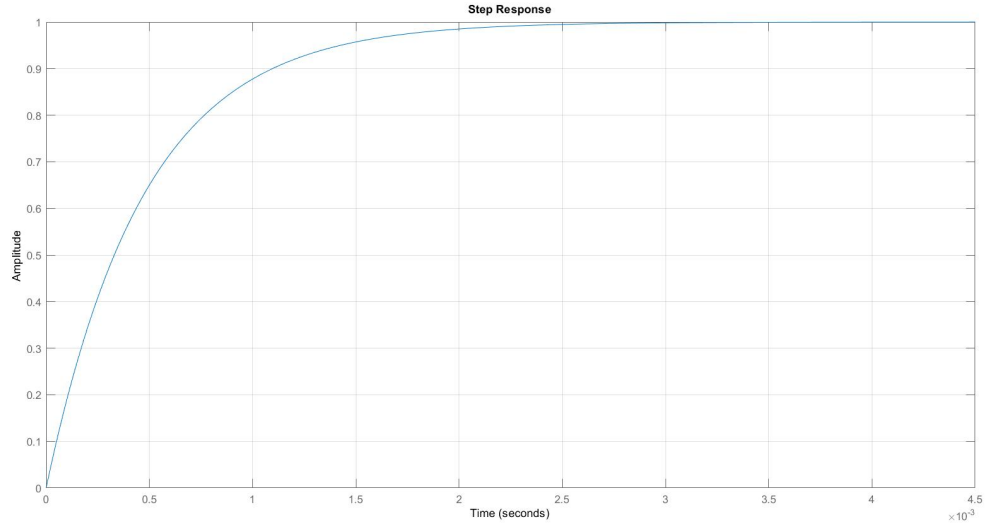


Figure 3.16: Step response of the closed loop PI current control system

### 3.9 PID Position Controller

In order to operate in cyclic synchronous position mode a positioning control loop is modeled and implemented, based on the subordinated current control. A reference position is provided to the controller which compare it with the actual position of the motor and computes the current set.

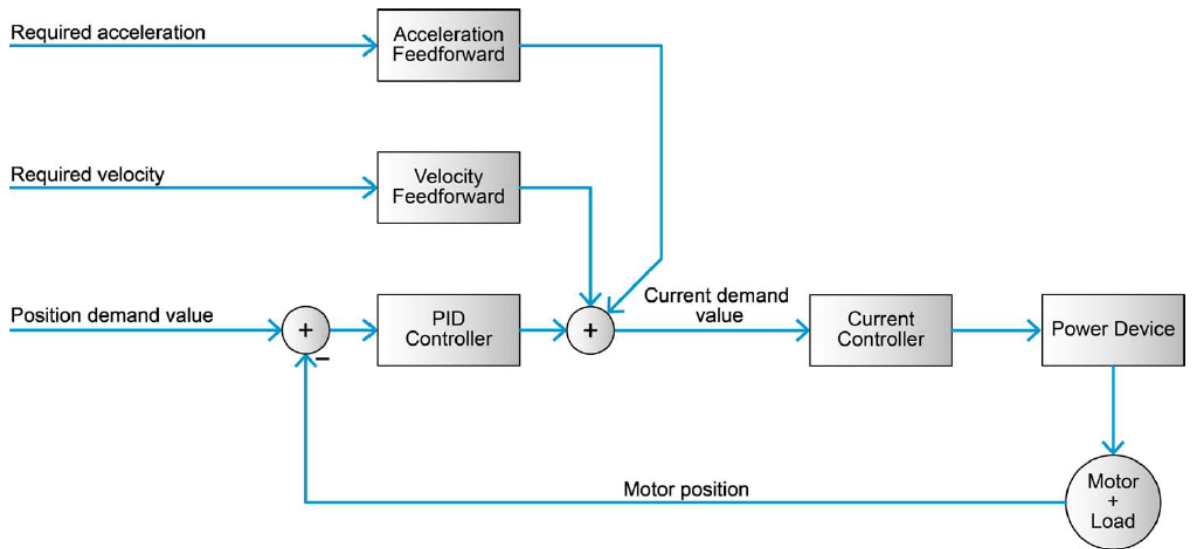


Figure 3.17: Position regulator with feed forward, controller architecture [20]

The position controller is implemented as PID controller. It is supplemented by *feedforward* control to improve the motion's system setpoint. Thereby, velocity feedforward serves for compensation of speed-proportional friction, while acceleration feedforward considers the inertia. Furthermore, the differential term of the PID controller signal is low pass filtered before adding to the proportional and integral part. This filtering stage is done to prevent differentiation of noisy measured motor position. The controller has been modeled on Simulink in order to tune it and analyze the performances. In this case the PID Simulink block has been used.

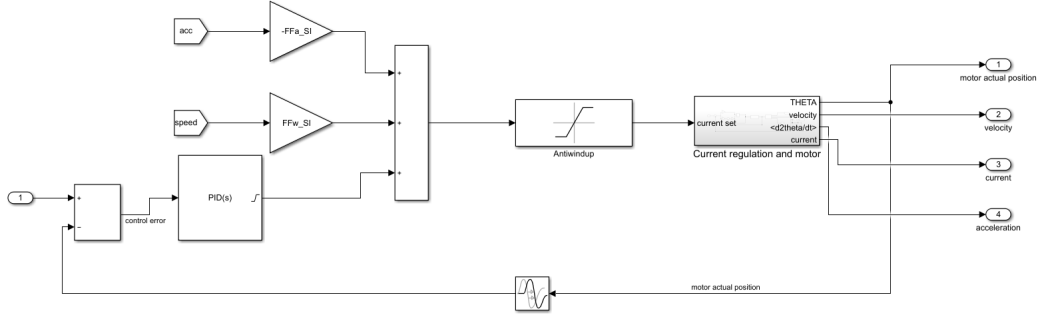


Figure 3.18: Position PID controller Simulink model

The main parameters of the controller loop are:

- $KP_{p_{SI}}$  is the position controller P gain expressed in  $\frac{A}{rad}$ .
- $KI_{p_{SI}}$  is the position controller I gain expressed in  $\frac{A}{rad \cdot s}$ .
- $KD_{p_{SI}}$  is the position controller D gain expressed in  $\frac{As}{rad}$ .
- $FF\omega_{SI}$  is the controller feedforward velocity gain expressed in  $\frac{As}{rad}$ . The purpose of the velocity feedforward is to provide additional current in cases in which the load increases with speed such as speed-dependent friction. The load is assumed to increase proportionally with speed. Thus the velocity feedforward coefficient can be defined as:

$$FF\omega = \frac{r}{k_t}$$

Where  $r$  is the damping proportional factor of the motor and  $k_t$  is the motor torque constant.

- $FFa_{SI}$  is the controller feedforward acceleration gain expressed in  $\frac{As^2}{rad}$ . The acceleration feedforward provides additional current in cases of high acceleration or high inertia. The acceleration feedforward coefficient is obtained as:

$$FFa = \frac{J}{k_t}$$

Where  $J$  is the inertia at the motor shaft and  $K_t$  is the torque motor constant.

- The *anti-windup* method is used to prevent saturation of the actuators. The limit is the nominal current of the motor (7.25A).
- The control loop transport delay is 0.4ms.

The transfer function that can be used in analytical or numerical simulations is:

$$C_{position}(s) = K_{pp} + \frac{K_{ip}}{s} + \frac{K_{dp}s}{1 + \frac{K_{dp}}{10K_{pp}}s}$$

### 3.9.1 PID Position Controller tuning

The tuning of the PID controller in position has been directly performed by using the *Tuning App* (transfer function based) in the Simulink block PID controller of the Control System Design Matlab add-on.

By using this tool it is possible to shape the step response by setting the robustness of the controller in order to obtain the PID gains. The plant is automatically linearized by the software.

The two parameters which can be optimized are:

- *The response time*, expressed in seconds.
- *The transient behaviour* of the system, making it more aggressive or more robust.

By increasing the aggressivity of the system and making its response faster, the PID gains are increased and the performances of the control system are enhanced. It is important to take into account that a too aggressive control can lead to saturation in current of the system, thus the performances have to be setted basing on the application in analysis.

The gains of the PID controller obtained by tuning are:

$$K_P = 1612.9 \frac{mA}{rad}$$

$$K_I = 1389 \frac{mA}{rad \cdot s}$$

$$K_D = 93.645 \frac{mA \cdot s}{rad}$$

The tuned system shows a *rise time* of 2.37 ms and an *overshoot* of 2.31%.

The step response and the Bode frequency response of the system are reported in the figures below, the performances obtained are consistent with the ASS application.

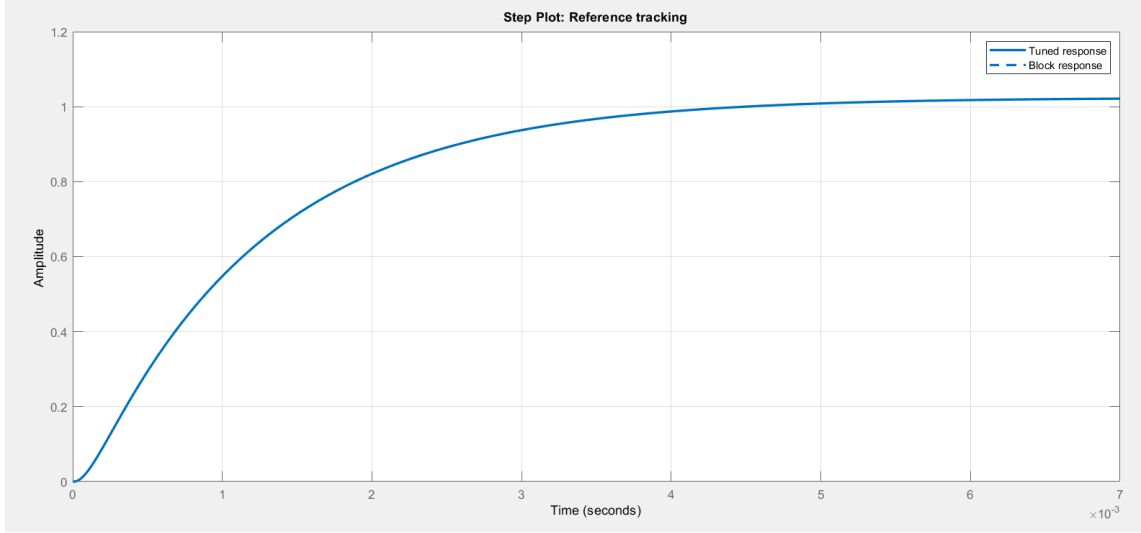


Figure 3.19: Step response of the closed loop PID position controller model

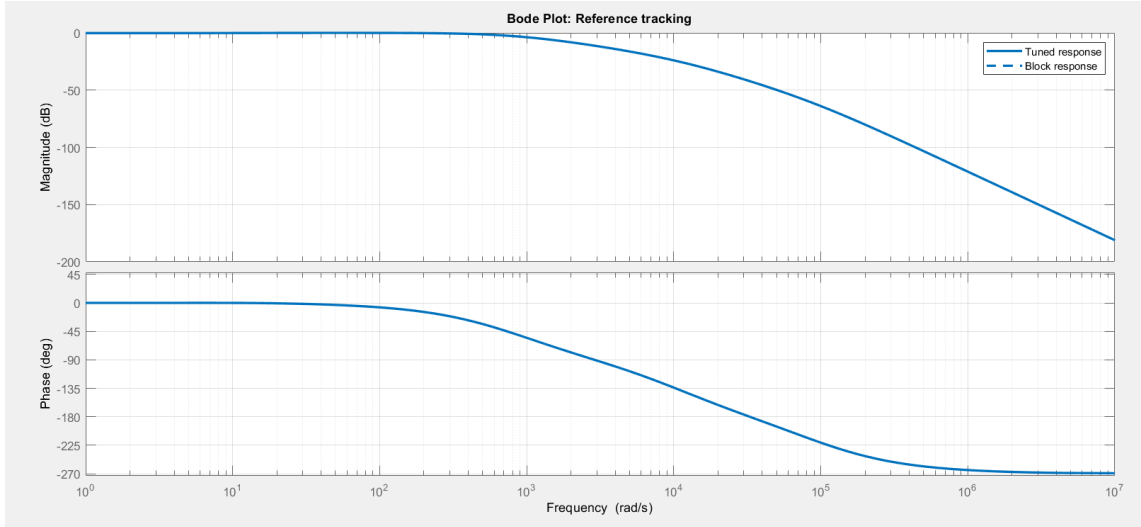


Figure 3.20: Bode diagram of the closed loop PID position controller model

### 3.10 Complete system model

The complete Autonomous Steering System, considering also all the mechanical sub-systems upstream the motor controller, have been modeled in Simulink.

The input to the control system is the reference steering angle at the wheels (the mean values considering a bicycle model of the vehicle in the path planning phase)  $\delta_w(ref)$ . This angle input is compared with the actual steering angle measured by the encoder sensor present on the steering column and divided by the steering ratio in order to obtain the actual  $\delta_w$ .

The steering ratio is computed by the data recorded from Squadra Corse. The toe angle is plotted in function of the rack displacement, by taking into account the C-Factor the steering ratio is obtained.

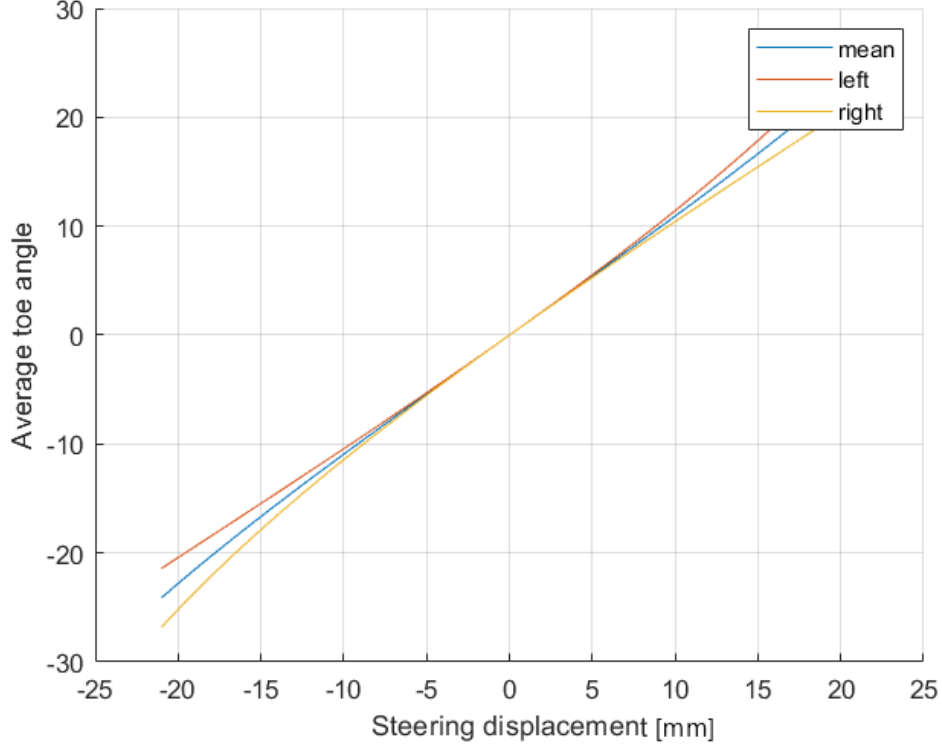


Figure 3.21: Squadra Corse data - Steering angle as function of the rack displacement

The input steering angle has to be converted in a target position of the motor  $\theta_{mot}$  by taking into account of all the kinematic ratios of the steering actuator system. The gains of the kinematic system are:

- *Steering ratio* = 4.29 (in order to pass from  $\delta_w$  to  $\delta_{sw}$ ).
- *C-factor* = 85.5/360mm per deg (To find the rack linear travel needed).
- Divided by: *Ball Screw Lead* = 2mm (Dividing by the Lead of the ball screw, the number of rotation of the nut are obtained).
- *Belt ratio* = 0.5

In this way the number of rotation of the motor needed to reach the target position are obtained. The Maxon Epos controller needs as angular position unit *inc* (the resolution of the Hall sensor per turn,  $42inc = 360deg$ ) thus the number of motor rotation must be multiplied by 42 to obtain the target position to be transmitted to the motor controller.

The CAN communication has been firstly simulated and modeled in a virtual channel by using the Car Communication Simulink toolbox in order to determine a suitable frequency of communication and its effect on the control performances. In the next chapter the CAN communication implementation will be discussed.

According to *DV 2.6.3*: *The inspection mission is defined by slowly spinning the drivetrain and actuating the steering system with a sine wave. After 25 s to 30 s the mission is finished and the transition to “AS Finish” must be initialized [17].*

For this reason a sine wave profile with as amplitude the maximum steering angle is used as input of the model in order to verify the performances and the compliance with the inspection test.

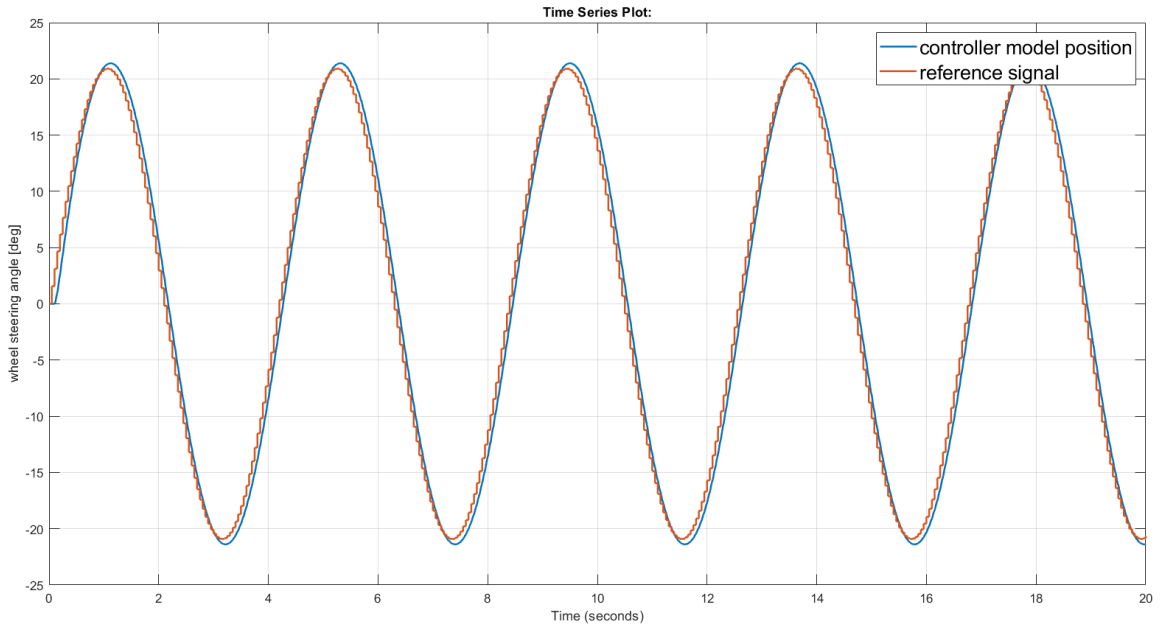


Figure 3.22: Sine wave performances of the complete system model



---

## CHAPTER 4

---

# System Validation and Implementation

The aim of this chapter is to describe the implementation on the physical actuator of the control strategy, of the communication and the description of the testing phase. The models described in the previous chapter, can provide the tuned parameters for the configuration of the Maxon motor. A test bench including the motor and the EPOS Positioning controller is set up for the validation of the models.

In the on-vehicle configuration, the Maxon motor will be connected via CAN to the dSpace Microautobox ECU. For this reason the CAN communication will be described among with all the procedure to configure and implement it.

The last section of the chapter deals with the description of the test bench of the actual actuator and the verification of its performances.

### 4.1 Models validation

The Simulink model of the motor and of the two control loops (current and position) defined and tuned in the previous chapter have been experimentally validated.

#### 4.1.1 PI Current Controller model validation

The validation of the model of the first control loop, including the motor and the PI current controller has been performed.

The PI gains obtained in the previous section of this thesis are manually inserted in the tool *regulation tuning* of the Maxon software *Epos Studio*. The motor has been connected to the PC by USB cable, in this way the motor can be directly tuned and controlled by using the Epos Studio software.

A screenshot of the regulation tuning phase and of the test signal applied in Epos Studio is reported in the fig. 4.1.

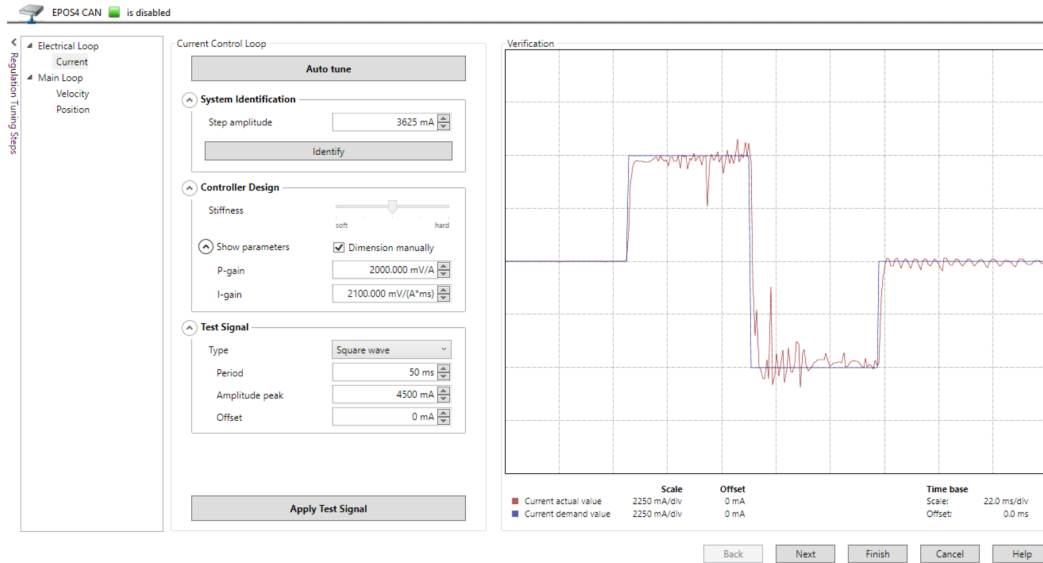


Figure 4.1: Regulation tuning in Epos Studio software screenshot

The data recording the behaviour of the current in the motor has been exported and compared with the performances of the controller-motor Simulink model, following the same reference profile.

As shown in fig. 4.2 the tuned model perfectly follow the reference current. The fluctuations and discrepancies of the current in the motor are mainly due to the resolution of the sensor.

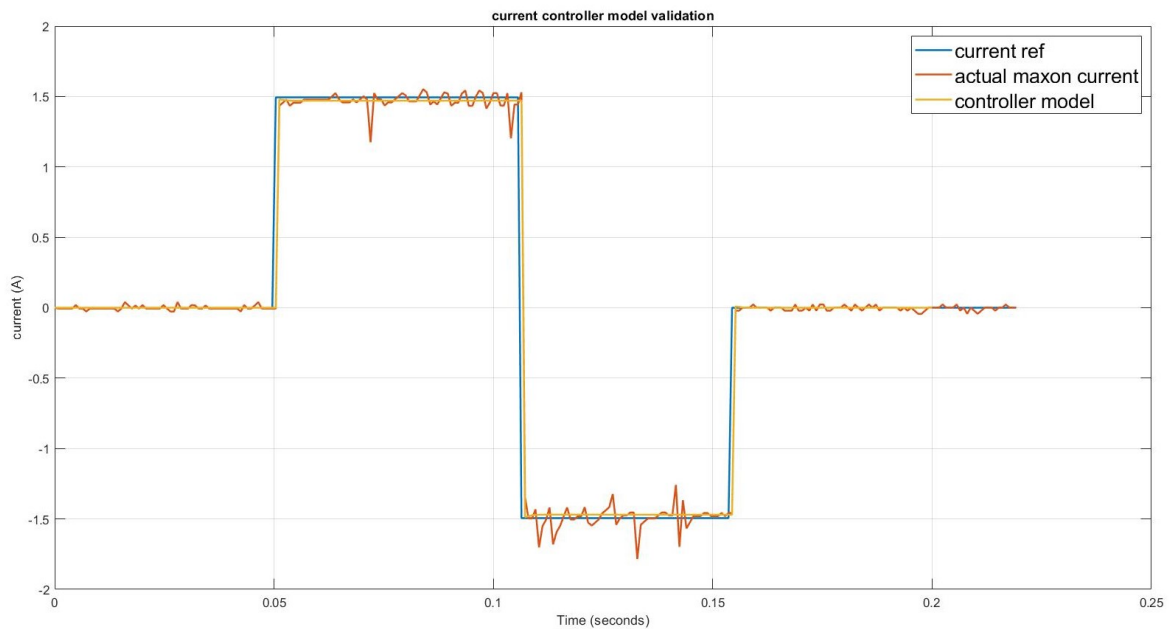


Figure 4.2: Comparison between the behaviour of the maxon motor current and of the model following the same reference profile

### 4.1.2 PID Position Controller model validation

For the validation of the external loop control stage (the PID in position), the same experimental procedure of the first loop is done.

The P, I and D gains obtained in Simulink and the feedforward gains obtained analytically have been imported in the *regulation tuning* tool of the Maxon software EPOS Studio.

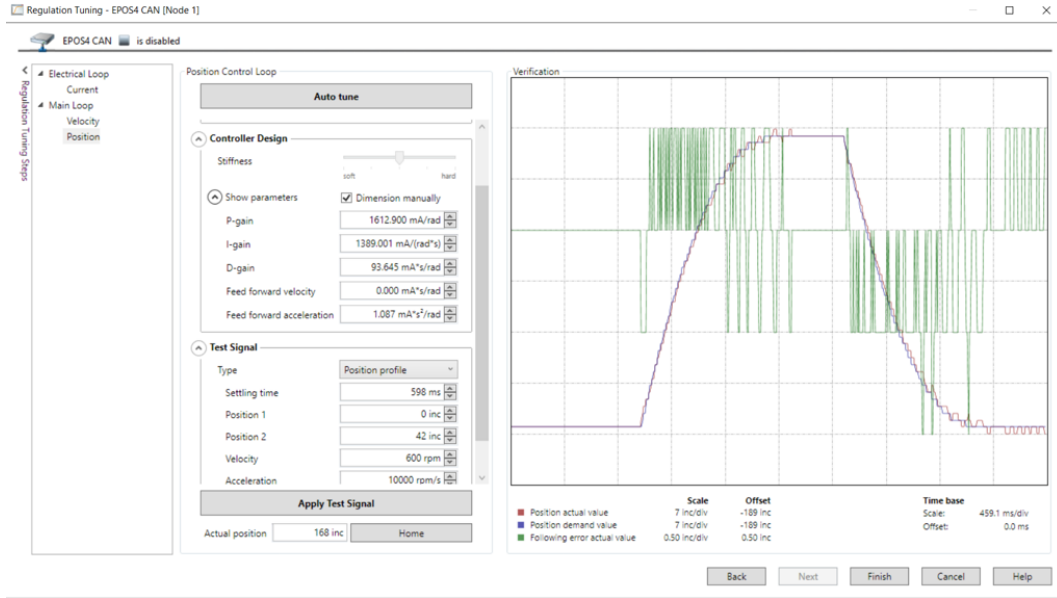


Figure 4.3: Regulation tuning (PID) in Epos Studio software screenshot

By using the Profile Position Mode in Epos Studio, a ramp signal (3000 rpm) is provided to the motor. The data recorded for the motor behaviour have been exported and compared with the behaviour of the model following the same profile. This procedure is also repeated for a random profile with a series of step.

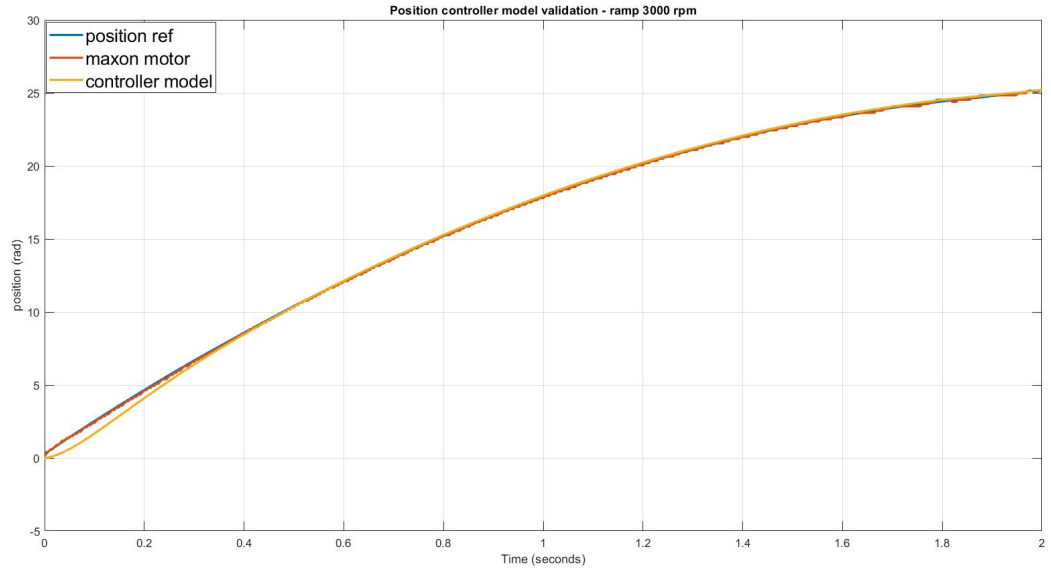


Figure 4.4: Comparison between the Maxon motor actual position and the model behaviour following the same ramp profile

In the ramp signal case, the model follows the reference profile and the behaviour is comparable with the actual motor.

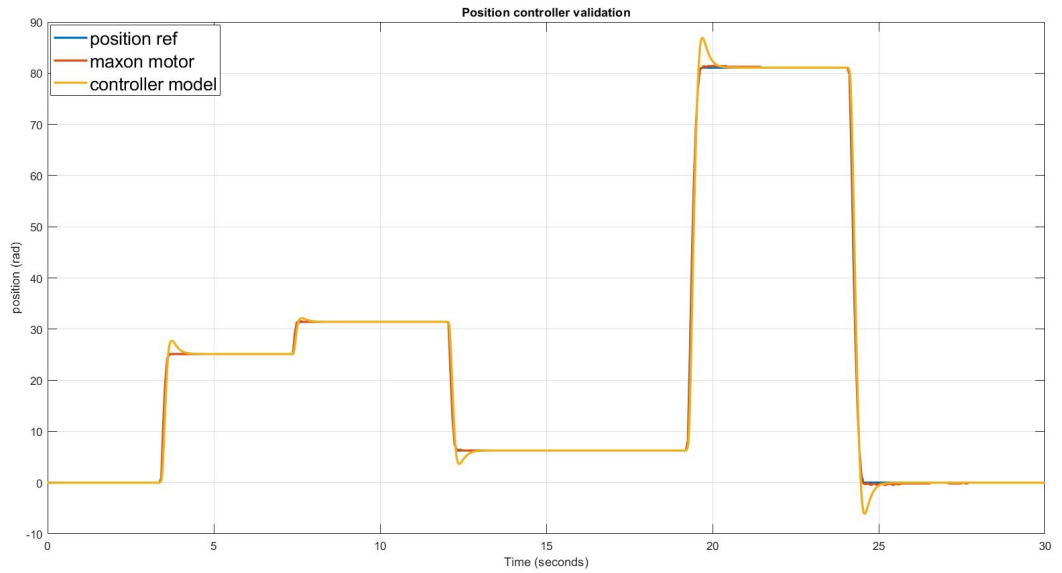


Figure 4.5: Comparison between the Maxon motor actual position and the model behaviour following the same profile

In the case reported in fig. 4.5 the model shows overshoot when very high amplitude

steps are applied. Anyway the performances are enough robust for the application and further increasing the aggressivity of the PID controller can lead to the saturation of the actuator.

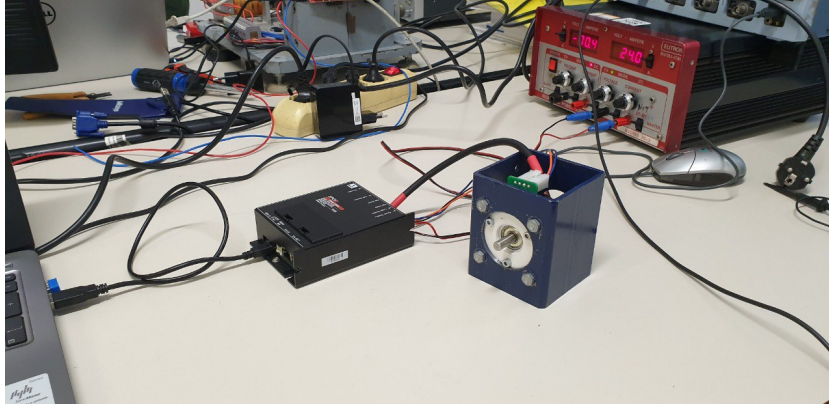


Figure 4.6: Test bench set up for motor controller tuning and validation

## 4.2 CANopen communication

CANopen is a communication protocol for embedded systems, generally used in automation. The CANopen communication concept can be defined as similar to the the ISO Open Systems Interconnection (OSI) Reference Model. In CANopen, only the layers above and including the network layer are implemented. Thus CANopen represents a standardized application layer and communication profile [22].

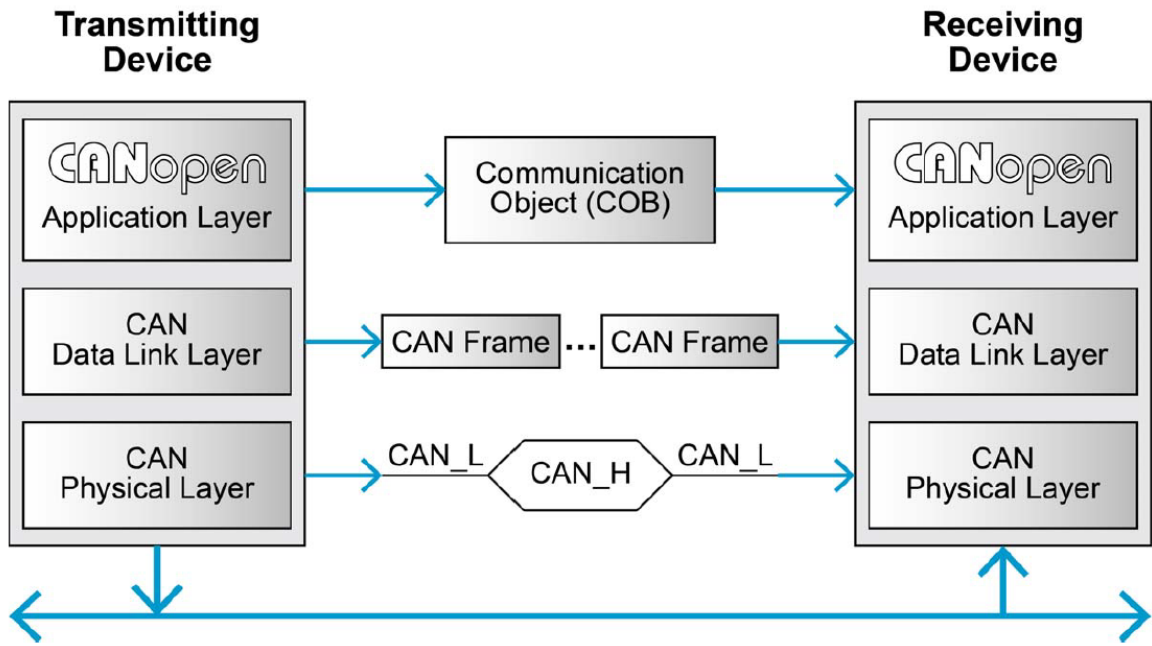


Figure 4.7: Protocol layer interactions in CANopen communication [22]

Dealing with the specific layers:

- *Physical Layer*

The networking system is based on the CAN serial bus. It assumes that the device's hardware features a CAN transceiver and a CAN controller as specified in ISO 11898. The physical medium is a differentially driven 2-wire bus line with common return.

- *Data Link Layer*

The Data Link layer is also standardized as CAN according to ISO 11898. Its services are implemented in the Logical Link Control (LLC) and Medium Access Control (MAC) sublayers. On the controller Epos 4, only Standard Frame Format is supported. It is important to take into account for this layer of the identifiers (COB-ID) and of the Remote Transmission Request (RTR) that can be used in asynchronous communication.

- *Application Layer*

This layer is standardized specifically for CANopen. In order to understand its features, it is possible to refer to the *Object Dictionary* which represents the most significant part of a CANopen device. It consists of a grouping of objects accessible via the network in an ordered predefined way. A 16-bit index is used to address all entries within the Object Dictionary. It can deal with simple variables, reporting directly the value, or to records and arrays which are managed by using Subindexes.

The Communication objects for CANopen differ for services and protocols. They can be classified in four main categories:

- *Process Data Objects (PDO)*

They are used for real time data transfer. PDO can be transmitted from a producer to a receiver or can be broadcasted. PDOs are transmitted in a non-confirmed mode.

The producer sends a Transmit PDO (TxPDO) with the specific identifier of the Receive PDO (RxPDO) of the consumer.

The number of PDOs that can be sent simultaneously depends on bus rate. In CAN communication the bus rate must be predefined and set equal for all the devices. Considering that the devices on the car are placed closely, the bus rate is fixed at 1 Mbit per second. The device supports so four PDOs.

CANopen communication allows three PDO message triggering modes:

- *Event-driven*. The message transmission is triggered by the occurrence of a specific object.
- *Polling by remote frames*. The transmission is asynchronous, triggered by an RTR frame.
- *Synchronized*. The PDO communication is synchronous, triggered by the periodic transmission of a SYNC message.

- *Service Data Objects (SDO)*

They are used to read/write access to entries of a device Object Dictionary. In this case the communication is confirmed.

- *Special Function Synchronization Objects (SYNC)*

They provide application-specific network synchronization and emergency messages. As the consumers receive the SYNC signal, they will commence carrying out their synchronized tasks. The SYNC consumers are the PDOs mapped as synchronously.

- *Network Management Objects (NMT)*.

They are used for network initialization, error control and device status message. The CAN network management is node oriented thus it is necessary to have one NMT Master.

NMT master send the NMT messages to switch the slaves from Pre-Operational State to Operational, according to the state scheme reported in the following figure. The PDO communication is allowed only in Operational State while SDO can only be managed in Pre-Operational state.

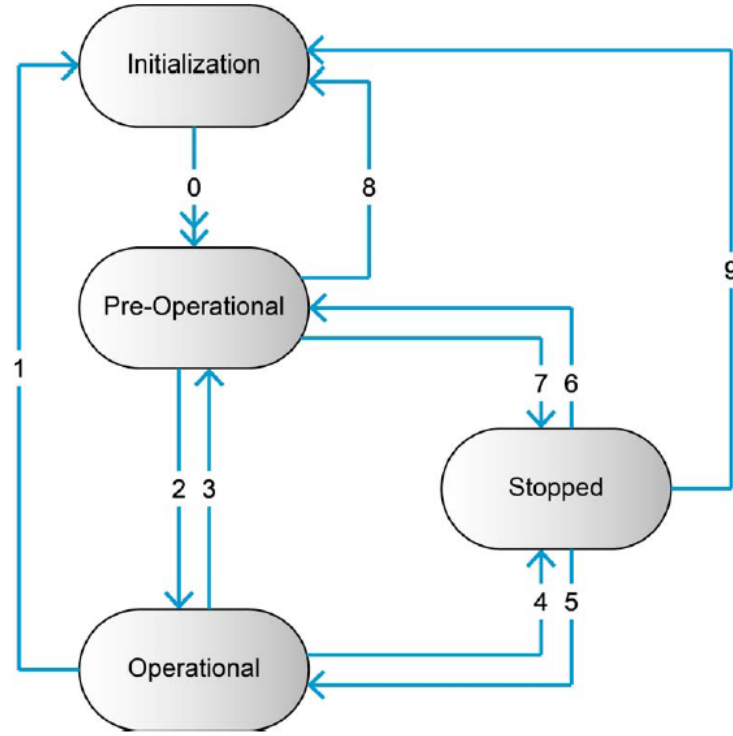


Figure 4.8: CAN communication, NMT slave states [22]

### 4.3 CANopen communication implementation

The communication between the EPOS motor controller for the steering actuator and the dSpace Microautobox ECU has been implemented and tested.

Firstly the State Machine for the CAN communication is implemented in Simulink, in order to optimize and to test it. In the final steps of the on-vehicle implementation the communication blocks along with all the other control algorithms have to be transcribed on the dSpace ECU. The EPOS controller is connected to the PC by using the CAN-USB interface *Kvaser Leaf Light*. The communication is real time checked by the CAN monitor software *Bus Master*.

In the application considered, the SDO communication is not implemented since the configuration relative to the limits for current, velocity, acceleration and position have been prior defined on the software EPOS Studio, thus only the real time (PDO) communication and the control messages (NMT) are needed.



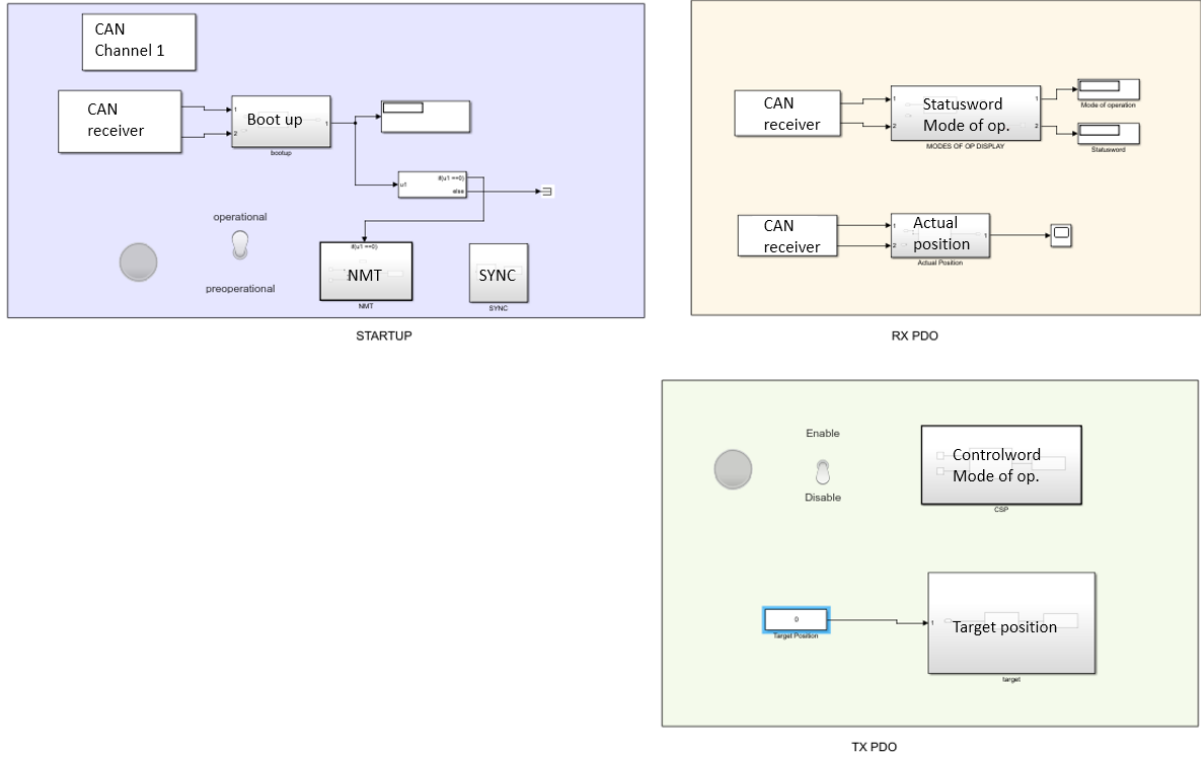


Figure 4.9: Simulink screenshot of the CAN communication implementation for the ASS

The Simulink blocks reported in fig. 4.9 send and receive the CAN messages by referring to a *DBC file* previously created.

The DBC file is an ASCII based translation file used to apply identifying names, scaling, offset and defining information, to data transmitted within a CAN frame. For any given CAN ID, a DBC file can identify some or all of the data within the CAN frame. The DBC file has been created by using the software *Vector CANdb++ Editor* and it contains all the identifiers and characteristics of all the messages processed, thus the transmitter or receiver Simulink block can directly manage data or signals.

In the next subsections the procedure necessary for setting up and operating the CAN communication for the ASS application is described.

#### 4.3.1 NMT communication implementation

As soon the device is turned on by applying the voltage set, it sends a *Boot Up* message. This message is 1 byte long and its COB-ID is *0x700* plus the Node ID of the device. An unique Node ID must be defined for all the devices within the CAN network. It results in the summed values of the stated DIP switches set to 1 (ON).

The Boot Up message contains a data equal to *0x00*. In the Simulink code implemented for the CAN communication, if the data of the Boot Up message is correctly received, the block managing the NMT communication will be triggered.

NMT messages are used to move between the different states reported in the scheme of fig. 4.8.

After the Boot Up initialization, the device automatically goes to Pre-Operational state, in which only the SDO communication is allowed.

To pass into Operational state and start the PDO communication, the *Start Remote Node Protocol* has to be sent. If it is needed to come back to SDO communication, it is necessary to send an *Enter Pre-Operational protocol* message.

The NMT message is 2 byte long with a COB-ID *0x00*. The data reported on the first byte indicates the function required, while on the second byte is reported the Node ID of the device. Sending a Node ID equal to 0, all the devices will perform that NMT function [20].

In the following table, the messages managed for NMT communication and listed on the DBC file are reported. All the values are expressed in hexadecimal.

Table 4.1: Boot Up and NMT messages

Name	COB ID	DLC	Data Byte 0	Data Byte 1
<i>Boot Up</i>	<i>0x701</i>	<i>1(unsigned)</i>	<i>0x00</i>	<i>/</i>
<i>NMT – Operational</i>	<i>0x00</i>	<i>2(unsigned)</i>	<i>0x01</i>	<i>0x01</i>
<i>NMT – Pre Operational</i>	<i>0x00</i>	<i>2(unsigned)</i>	<i>0x80</i>	<i>0x01</i>
<i>NMT – Stop</i>	<i>0x00</i>	<i>2(unsigned)</i>	<i>0x02</i>	<i>0x01</i>
<i>NMT – Reset</i>	<i>0x00</i>	<i>2(unsigned)</i>	<i>0x81</i>	<i>0x01</i>

### 4.3.2 PDO communication implementation

Once the device is entered in the Operational protocol, the PDO real time communication can start.

Before sending or receiving a PDO message, it is needed to configure the PDOs processed by using the *PDO Mapping* function of the software Epos Studio. Among all the PDOs available for the EPOS controller, only the PDOs necessary for the Cyclic Synchronous Position (CSP) mode are mapped.

Dealing with the CSP mode, before starting sending the target position and in order to let the motor rotating, it is necessary to perform an enabling phase defined by the state machine reported in fig. 4.10.

The states may be changed, enabling or disabling the operation, by using the data contained in the *Controlword*.

The state in which the device is, can be checked by reading the data included in the *Statusword* sent by the EPOS.

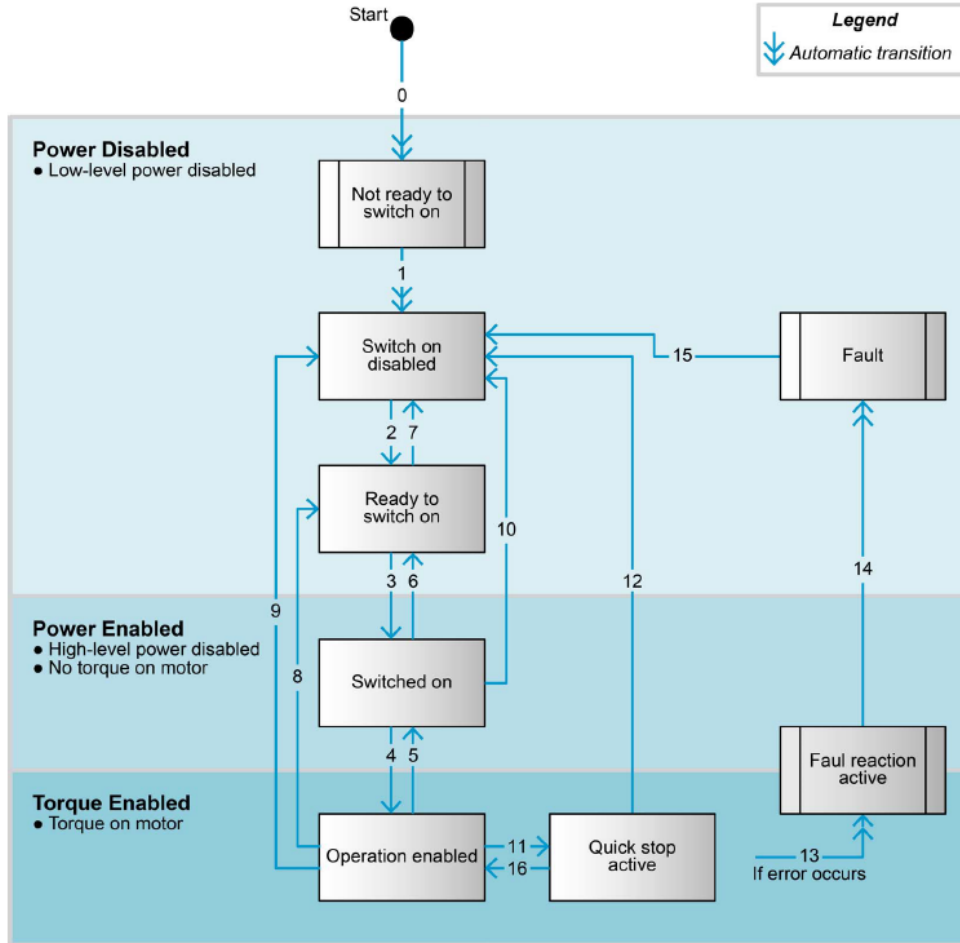


Figure 4.10: Device State Machine for enabling/disabling operations

Thus it is fundamental to take into account of controlword and statusword in mapping the receiving and transmitting PDOs and to configure all the messages needed for enabling/disabling the operations into the DBC file.

- *Transmit PDOs*

These PDOs are sent by the EPOS device controller towards the Master.

For the application analysed, only 3 PDOs are mapped on two COB - IDs.

The first TxPDO includes:

- *Position Actual Value* - It is a synchronous message sent according to the periodic sync message sent by the master. The data layer of this PDO contains the actual position of the motor measured by the embedded digital Hall sensor.

The second mapped TxPDO includes:

- *Statusword* - mapped on the first two bytes, it is asynchronously sent on data change. It reports the actual state of the device.

- *Modes of operation display* - One byte, sent asynchronously, indicating the selected actual mode of operation in the device.

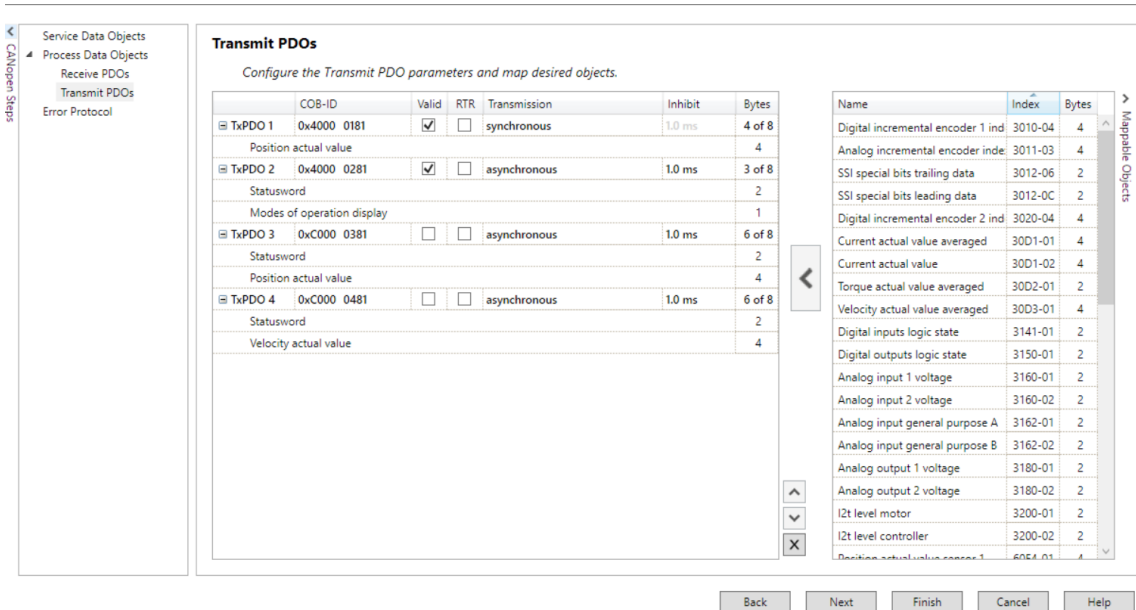


Figure 4.11: Epos Studio transmitting PDO mapping function screenshot

- *Receive PDOs*

Messages sent towards the EPOS controller.

In the ASS case, only 3 messages on two RxPDOs have been mapped.

The messages mapped in RxPDO1 are:

- *Modes of Operation* - Mapped on the first byte. By changing the data included in this PDO, the mode of operation can be selected. The transmission is asynchronous.
- *Controlword* - Mapped on two bytes, it is used to enable/disable operations and to pass from one state to another.

The message mapped in RxPDO2 is:

- *Target Position* - If the CSP mode is selected and enabled, when receiving this message, the motor will start to apply the torque to reach the target position indicated.

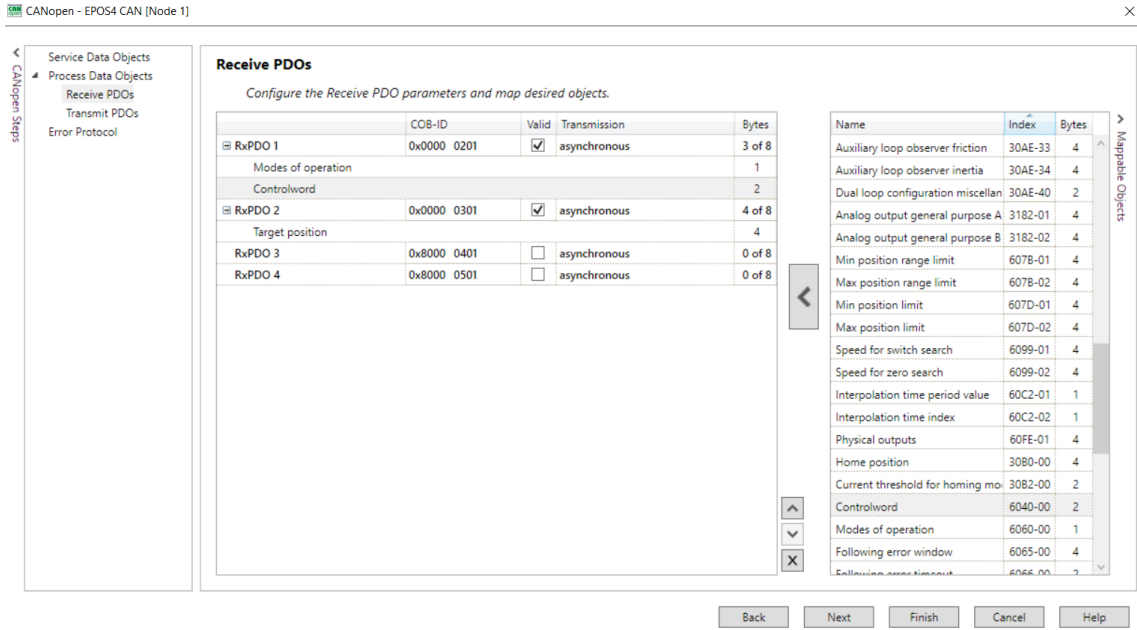


Figure 4.12: Epos Studio receiving PDO mapping function screenshot

The periodic SYNC message for synchronous communication is:

Table 4.2: SYNC message

Name	COB ID	DLC	Data Byte 0	Data Byte 1
<i>SYNC</i>	0x80	0	/	/

The following tables define the characteristics of the transmit PDOs mapped. The messages reported have to be written on the DBC file.

Table 4.3: Transmit PDO mapped

Name	COB ID	DLC	Data Byte 0	Data Byte 1
<i>TxPDO1 actual pos.</i>	0x181	4( <i>signed</i> )	<i>inc</i>	<i>inc</i>
<i>TxPDO2 statusword</i>	0x281	2( <i>unsigned</i> )	/	<i>LowByte</i>
<i>TxPDO2 mode of Op.</i>	0x281	1( <i>unsigned</i> )	<i>op.</i>	/

The actual state of drive is reported on the low byte of the Statusword according to the following table available on the EPOS Firmware Specifications manual [23].

State	Statusword [binary]	Description
Not ready to switch on	xxxx xxxx x00x 0000	Drive function is disabled
Switch on disabled	xxxx xxxx x10x 0000	Drive initialization is complete. Drive parameters may be changed. Drive function is disabled.
Ready to switch on	xxxx xxxx x01x 0001	Drive parameters may be changed. Drive function is disabled.
Switched on	xxxx xxxx x01x 0011	Drive function is disabled. Current offset calibration done.
Operation enabled	xxxx xxxx x01x 0111	No faults have been detected. Drive function is enabled and power is applied to the motor.
Quick stop active	xxxx xxxx x00x 0111	«Quick stop» function is being executed. Drive function is enabled and power is applied to the motor.
Fault reaction active	xxxx xxxx x00x 1111	A fault has occurred in the drive. Selected fault reaction is being executed.
Fault	xxxx xxxx x00x 1000	A fault has occurred in the drive. Drive parameters may have changed. Drive function is disabled.

Figure 4.13: Device state bits [23]

The receive PDOs characteristics necessary for building the DBC file are:

Table 4.4: Transmit PDO mapped

Name	COB ID	DLC	Data Byte 0	Data Byte 1
<i>RxPDO1 mode of Op.</i>	0x201	1( <i>unsigned</i> )	<i>op.</i>	/
<i>RxPDO1Controlword</i>	0x201	2( <i>unsigned</i> )	/	<i>LowByte</i>
<i>RxPDO2 target pos.</i>	0x301	4( <i>signed</i> )	<i>inc</i>	<i>inc</i>

The data to be sent in the Data Field of the Mode of Operation PDO, mapped on RxPDO1, in order to select CSP mode, is *0x08*.

The commands of the controlword have to be specified in the low byte of the controlword PDO. The function and the data of each command needed for states transition are specified in table 4.5.

Thus, as soon as the device enters in the Operational protocol, it automatically moves to the *Switch on disabled state*. By sending in the Controlword the *Shutdown command* (2 in the table) the device will be *Ready to Switch ON*.

Then, by sending through the Controlword the command number 4 *Switch on and Enable operation*, if the motor is in CSP mode, it will start to move for reaching the target position.

Table 4.5: Controlword command

Command	COB ID	DLC	Data Low byte	Transition
<i>Shutdown.</i>	0x201	2( <i>unsigned</i> )	0x0006	2/6/8
<i>Switch On/Enable</i>	0x201	2( <i>unsigned</i> )	0x000F	3/4
<i>Quick Stop</i>	0x201	2( <i>unsigned</i> )	0x000B	11

## 4.4 Test bench validation

The final step of the project is to set up a test bench for the validation of the Autonomous Steering System, in order to validate and test the performances of the actual components.

The actuator has been manufactured by *Officine Meccaniche Massola* according to the specifications provided during the design phase. The hardware components were available for test bench phase during the last week of the project.

Firstly the system has been assembled and carefully lubricated. The actuator is fixed on a wooden plank for experimental testing phase in the laboratory. During fixing and assembly phase it is important to guarantee the right alignment of the ball-screw holes. The BLDC Maxon motor has been placed in its apposite housing by adding a 3D printed plastic cover to protect the motor from dust during operations.

A series of test have been performed in order to check the performances of the actuator. The Maxon motor is connected to the EPOS Positioning controller unit with the apposite cables, the EPOS controller is connected to the PC via CAN Bus.

By using the CAN messages previously defined, step and random signals have been provided to the system. The response of the actuator is robust as expected.

The sine wave reference signal for inspection test (DV 2.6.3 [17]) is given as input in order to check the response of the system performing the complete travel. The actuator behaviour is compliant with the performances required.

The compliance of the ASS with requirement of reversibility (DV 2.3.3 [17]) has been also verified in the laboratory test phase before installing the actuator on the vehicle.

A series of picture of the Autonomous Steering System actuator in the test bench are reported below.



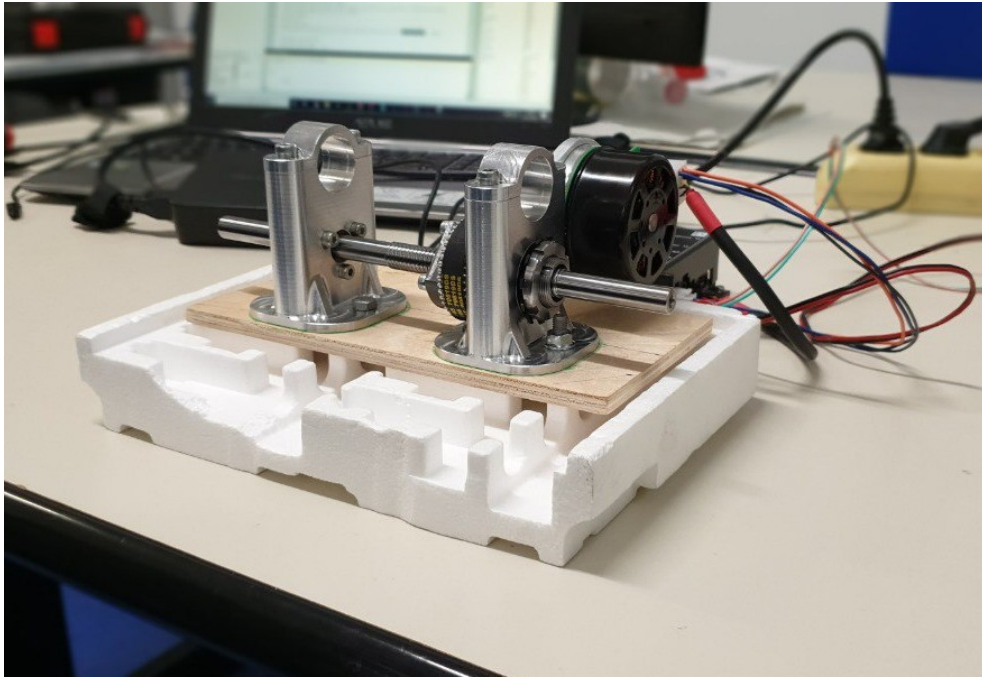


Figure 4.14: Autonomous Steering Actuator test bench 1

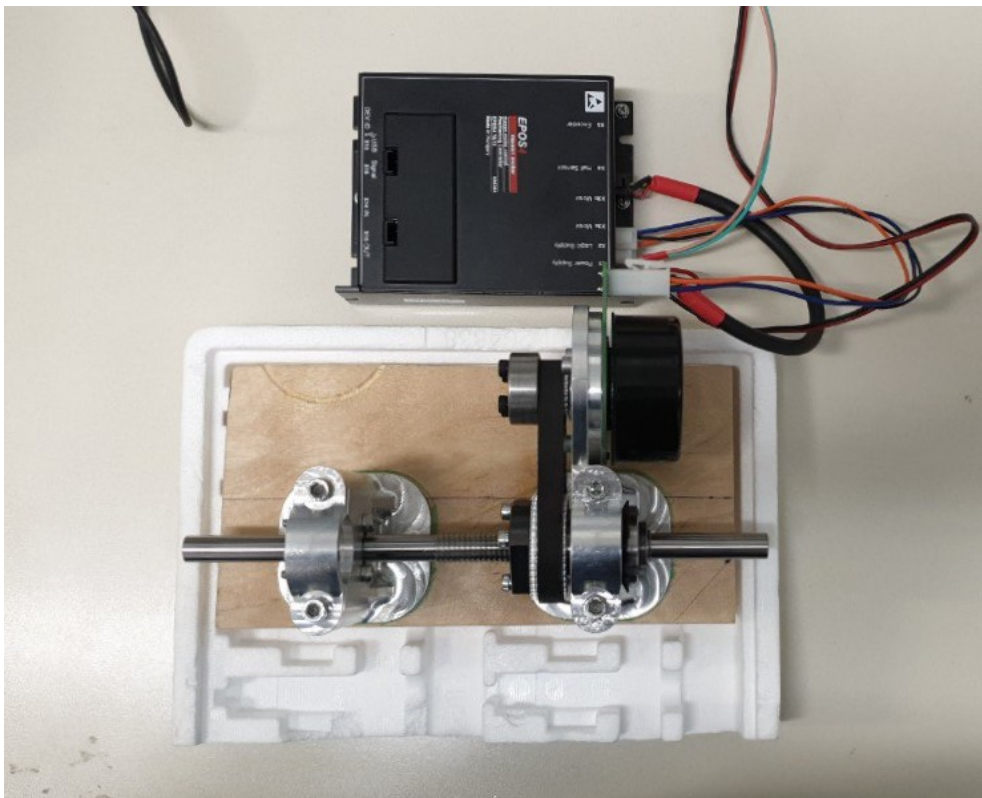


Figure 4.15: Autonomous Steering Actuator test bench 2



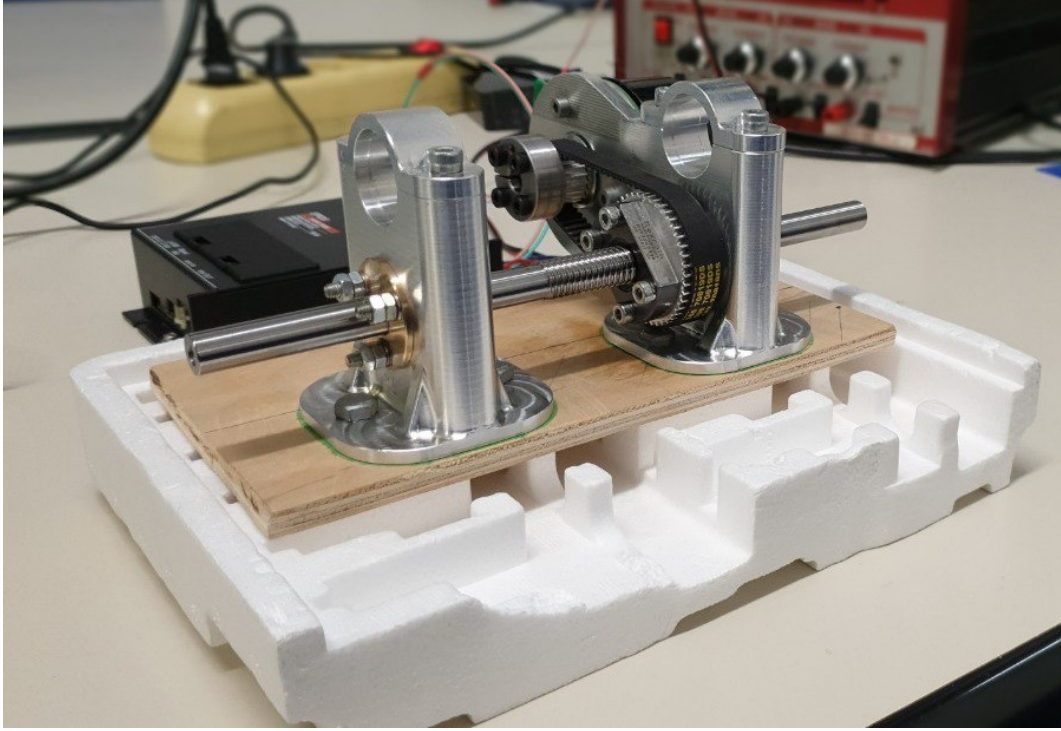


Figure 4.16: Autonomous Steering Actuator test bench 3

## 4.5 On-vehicle implementation

In order to implement the CAN communication and the ASS control on the vehicle, it is necessary to take into account of the integration between the EPOS 4 controller and the dSpace ECU.

The input to the system is the reference steering angle ( $\delta_{ref}$ ) computed on the dSpace by the positioning and path planning algorithm. This angle reference has to be converted into motor target position (in inc) by considering all the kinematic gains of the steering system and of the steering actuator ball screw and belt drive, as presented in section 3.10. The reference input is compared with the actual position of the steering wheel measured by the steering encoder sensor present on the steering column. This feedback is used to perform the homing procedure and provide the initialization value to get the zero position before starting moving. The value measured by the sensor is sent to the dSpace by CAN bus.

In fig. 4.17 the working principle of the complete system integrated with the steering encoder sensor is presented.

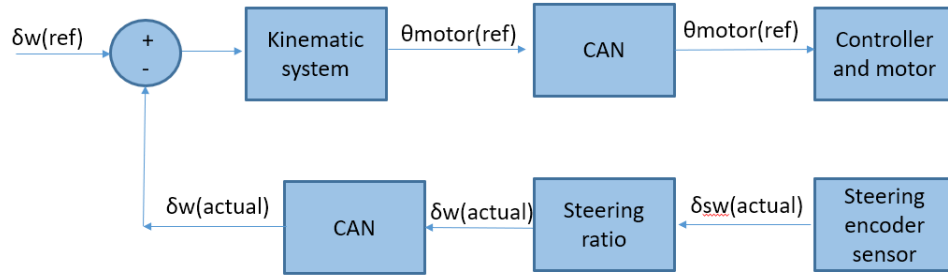


Figure 4.17: ASS block scheme

An alternative control strategy is to implement the positioning controller (presented in section 3.9) directly on the dSpace and to sent to the EPOS controller as input only the reference current set, thus only the PI current control is performed by the EPOS control unit. This approach can give more degrees of freedom in designing and tuning the positioning control.

---

## CHAPTER 5

---

# Conclusions and future development

Tying up the the dissertation presented, it is important to resume the main goals achieved during the thesis project. Starting from the analysis of the SC 19 steering system, FSD regulations and design constraints (limited space availability inside the cockpit), the most suitable design solution for the Autonomous Steering actuator system has been chosen and developed.

By the deatailed analysis of the control architecture for the Maxon BLDC motor provided by the EPOS 4 positioning controller, a reliable model of the motor and of the control system has been developed. Through the Simulink models it was possible to study an optimal tuning strategy for the design parameters of the controller taking into account a trade off between the perfromances needed in terms of response of the system and the possible saturation of the actuator.

The parameters obtained were then inserted in the actual controller and the models previously created have been experimental validated with the data recorded by the motor sensors. After the validation phase, the control of the actuator is implemented via CAN bus. The communication in CANOPEN protocol has been mapped, optimized and configured.

Once the actual actuator has been manufactured, the complete system has been controlled through CAN and the performances has been verified on a test bench. The complete system response is appropriate, the results obtained are in line with the expected ones from the models and the system is ready to be installed on the vehicle.

## 5.1 Future development

The project carried out within the context of this dissertation is only a stage of the complete pocess for the development of the Formula Driverless Vehicle. The scope and time frame of the thesis project has limited to the design, simulation work, control and communication implementation.

The future steps of the project are expected:

- The first step is to transfer the control and communication code developed in Matlab Simulink environment on the dSpace Microautobox ECU and to integrate them with the complete controller of the vehicle.
- To synchronize and optimize the integration of the control loop with the data provided by the steering encoder sensor in order to perform the homing procedure and to have a feedback on the actual behaviour of the steering system.
- To install the Autonomous system on the vehicle connecting all the mechanical linkages and the wirings with actual sensors and ECUs.
- The last step is to further optimize the control strategy of the system to deal with the actual FS competition conditions of an acceleration test and finally of a trackdrive discipline.



Figure 5.1: SC 19 on a FSD trackdrive scenario 1



Figure 5.2: SC 19 on a FSD trackdrive scenario 2

---

## APPENDIX A

---

### FSD regulations

The regulations and guidelines for designing the steering system are reported [17].

- T2.6.1 Steering systems using cables or belts for actuation are prohibited. [DV ONLY]  
This does not apply for autonomous steering actuators.
- T2.6.2 The steering wheel must directly mechanically actuate the front wheels.
- T2.6.3 The steering system must have positive steering stops that prevent the steering linkages from locking up. The stops must be placed on the rack and must prevent the tires and rims from contacting any other parts. Steering actuation must be possible during standstill.
- T2.6.4 Allowable steering system free play is limited to a total of  $7^\circ$  measured at the steering wheel.
- T2.6.5 The steering wheel must be attached to the column with a quick disconnect. The driver must be able to operate the quick disconnect while in the normal driving position with gloves on.
- T2.6.6 The steering wheel must be no more than 250mm rearward of the front hoop. This distance is measured horizontally, on the vehicle centerline, from the rear surface of the front hoop to the forward most surface of the steering wheel with the steering in any position.
- T2.6.7 The steering wheel must have a continuous perimeter that is near circular or near oval. The outer perimeter profile may have some straight sections, but no concave sections.
- T2.6.8 In any angular position, the top of the steering wheel must be no higher than the top-most surface of the front hoop.
- T2.6.9 The steering rack must be mechanically attached to the chassis.



T2.6.10 Joints between all components attaching the steering wheel to the steering rack must be mechanical and visible at technical inspection. Bonded joints without a mechanical backup are not permitted. The mechanical backup must be designed to solely uphold the functionality of the steering system.

T2.6.11 Rear wheel steering, which can be electrically actuated, is permitted if mechanical stops limit the range of angular movement of the rear wheels to a maximum of  $6^\circ$ . This must be demonstrated with a driver in the vehicle and the team must provide the equipment for the steering angle range to be verified at technical inspection.

The additional guidelines about the steering which are checked for a driverless vehicle are:

DV 2.3.1 Steering system actuation (movement) must only happen if the vehicle is R2D.

DV 2.3.2 The steering system may remain active during an emergency brake maneuver while vehicle is in movement.

DV 2.3.3 Manual steering must be possible without manual release steps (e.g. operating manual valves (dis-)connecting mechanical elements) while ASMS is switched “Off”.

---

## References

- [1 ] (2010). Automobile History, [Online]. Available: <https://www.history.com/topics/automobiles>.
- [2 ] Pwc global [Online]. Available: <https://www.pwc.com/en/industries/automotive/publications/eascy.html>.
- [3 ] [Online] Available: <https://www.stradeeautostrade.it/notizie/2020/incidenti-stradali-nel-mondo-12-milioni-di-morti-in-un-anno>.
- [4 ] E. P. & M. Moustaki, "Human factors in the causation of road trac crashes," European Journal of Epidemiology, p. 8, 2000.
- [5 ] SAE J3016 - Levels of Driving Automation.
- [6 ] Mc Kinsey Center for future mobility. [Online]. Available: <https://www.mckinsey.com/features/mckinsey-center-for-future-mobility/overview/autonomous-driving>.
- [7 ] Kabzan, Ehmke, Buhler, Gupta, *AMZ Driverless: The Full Autonomous Racing System*, ETH Zurich, Switzerland, May 2019.
- [8 ] Genta, Morello, *The Automotive Chassis Volume 1*, Springer, 2009.
- [9 ] Bosch Automated driving mobility soultions, [Online] Available: <https://www.bosch-mobility-solutions.com/en/products-and-services/passenger-cars-and-light-commercial-vehicles/steering-systems/electric-power-steering-systems/>.
- [10 ] Naranjo, Gonzalez, Garcia, de Pedro, *Electric Power Steering Automation for Autonomous Driving*, Instituto de Automatica Industrial, Madrid 2015.
- [11 ] Carriere, Caux, Fadel, *Optimal LQI Synthesis for Speed Control of Synchronous Actuator under Load Inertia Variations*, The International Federation of Automatic Control, Seoul,Korea 2008.
- [12 ] Govender, Khazaridi, Weiskircher, Keppler, Muller, *A PID and state space approach for the position control of an electric power steering*, 16th Stuttgart International Symposium Automotive and Engine Technology, 2016.



- [13 ] Govender, Muller, *Modelling and Position Control of an Electric Power Steering System*, Vehicle Automation and Chassis Systems, Daimler AG, IFAC, Berlin 2016.
- [14 ] About Formula Student, [Online]. Available: <https://www.imeche.org/events/formula-student/about-formula-student>.
- [15 ] 29/08/2020 FSG academy powered by Waimo.
- [16 ] Sarath Babu Nagarajan, *Design of autonomous steering actuator for Formula Student Driverless car*, Politecnico di Torino, 2019.
- [17 ] Formula Student 2020 Rulebook, [Online]. Available: <https://www.formula-student.de/all/2020/rules/FSRules-2020-V1>.
- [18 ] Mathworks, Introduction to Brushless DC motor control, [Online] Available: <https://it.mathworks.com/campaigns/offers/brushless-dc-motors-introduction.confirmation.html>.
- [19 ] William, Milliken, *Race Car Vehicle Dynamics*, SAE, 2014.
- [20 ] Maxon Group, *Epos 4 Positioning Controller Application Notes*, Edition 2019-11.
- [21 ] Goodwin, Graebe, Salgad, *Classical PID Control*, Control System Design, Prentice Hall PTR.
- [22 ] Maxon Group, *Epos 4 Positioning Controller communication guide*, Edition 2020-04.
- [23 ] Maxon Group, *Epos 4 Positioning Controller firmware specifications*, Edition 2020-04.
- [24 ] Cao, Tang, Jiang, Yin, Zhang, Huang, *Study on Low-Speed Steering Resistance Torque of Vehicles Considering Friction between Tire and Pavement*, Article, Applied Science, MDPI, 2019.
- [25 ] G. Genta, *Motor Vehicle Dynamics Modeling and Simulation*, World Scientific Publishing.