

POLITECNICO DI TORINO

**MASTER OF SCIENCE IN MECHATRONIC
ENGINEERING**



Master's Degree Thesis

**Characterization, control and testing of
an E/H actuator for the development of
an autonomous braking system**

Supervisors

Prof. CARLO NOVARA

Co-Sup. DAVIDE COLOMBO

Candidate

GIANLUCA DOTTA

DECEMBER 2022

Abstract

This thesis aims at designing an autonomous braking control implementable on a real L4 experimental autonomous vehicle. In particular, the main goals are characterizing, controlling and testing a pump to verify its capability to fulfill some desired braking performances. To these purposes, the Dexter electro-hydraulic (E/H) 1600 brake actuator has been used, which is commonly employed in trailer braking systems. In cooperation with the company Bylogix s.r.l., we have developed a brake-by-wire (BBW) solution where the pump, driven by the control unit, directly generates the hydraulic pressure within the brake circuit.

A tailor-made test-bench built in collaboration with the company has been used to characterize the actuator. This operation has been carried out by designing a proper acquisition system to allow experimental data collection in Simulink. The utilized device for the acquisition is the USB-6009 board by National Instruments (NI).

Due to the non-linearity of the actuator, a gain scheduling (GS) control technique has been adopted. The operating points have been divided in several regions within which has been possible to design a linear PID controller. Moreover, to handle the saturation of the command input, a back-calculation anti-windup technique has been employed.

The control algorithm has been developed in Simulink and then implemented and tuned on a CAN control unit through the Eagle programming interface included within the MRS Developer Studio software.

Later, both the pump and the controller have been mounted on the vehicle, where the control algorithm has been integrated and adapted to the real system.

Finally, different types of tests have been performed, both on the test bench and on the vehicle, to check the quality of the control algorithm under different operating conditions.

Acknowledgements

Vorrei ringraziare il Professor Carlo Novara per l'opportunità concessami e per avermi fornito con prontezza il suo aiuto quando ce n'è stato bisogno.

Un ringraziamento a tutta l'azienda Bylogix per avermi accolto e per aver messo a mia disposizione la strumentazione necessaria allo svolgimento della tesi.

Grazie inoltre al mio tutor, l'ing. Davide Colombo, per avermi dato la possibilità di lavorare a questo progetto sotto la sua guida e per avermi sempre consigliato .

Infine, un ringraziamento speciale alla mia famiglia, alla mia ragazza e ai miei amici, per avermi sempre sostenuto e supportato attraverso questo lungo percorso.

Table of Contents

List of Tables	VIII
List of Figures	IX
1 Introduction	1
1.1 Autonomous driving	1
1.1.1 Classification	2
1.1.2 Active safety systems	3
1.2 Braking system	4
1.2.1 Braking sequence	6
1.3 Brake-By-Wire	7
1.3.1 Pump-type EHB	8
1.3.2 Motor-type EHB	10
1.3.3 BBW control	11
1.4 Thesis objectives	12
2 System characterization	14
2.1 General setup	15
2.1.1 Test bench	15
2.1.2 Pressure sensor	16
2.1.3 Acquisition device	17
2.1.4 Calibration	18
2.2 Static characterization	20
2.2.1 Wiring connections	20
2.2.2 Voltage divider	21
2.2.3 Simulink acquisition diagram	22
2.2.4 Acquisition results	22
2.3 Dynamic characterization	25
2.3.1 Controlled-Area Network	25
2.3.2 Pulse-width modulated signal	27
2.3.3 Filtered voltage divider	28

2.3.4	Anti-aliasing filter	30
2.3.5	Wiring connections	31
2.3.6	Simulink acquisition diagram	31
2.3.7	Eagle diagram	33
2.3.8	Open-loop bandwidth	34
2.3.9	Quasi-static characterization	35
2.3.10	Non-linear behaviour	36
2.3.11	Acquisition results	37
3	System identification	39
3.1	Choice of the datasets	40
3.2	Linear models	40
3.2.1	ARX	41
3.2.2	OE	41
3.3	Nonlinear models	42
3.4	Observations	43
4	Control algorithm design	44
4.1	Controller choice	44
4.1.1	PID control	44
4.1.2	Gain Scheduling	45
4.2	Control architecture	48
4.3	Eagle implementation	50
4.3.1	Acquisition on PLC	50
4.3.2	PID implementation	52
4.3.3	GS implementation	53
4.3.4	Anti-windup technique	55
4.3.5	Tuning	57
5	Control algorithm validation	58
5.1	Acquisition interface	58
5.2	Reference response	60
5.2.1	Pre-braking phase	60
5.2.2	Step reference	61
5.2.3	Non-linearity handling	65
5.2.4	Sinusoidal and triangular references	65
5.2.5	Observations	66
5.3	Closed loop system bandwidth	67

6	Integration on the vehicle	69
6.1	Electrical harness	70
6.2	Control integration	73
6.2.1	PID re-tuning	73
6.2.2	Brake pedal role	73
6.3	Stationary vehicle results	74
6.3.1	Simulink	74
6.3.2	Oscillation phenomenon	76
6.3.3	Different braking profile	78
6.4	Moving vehicle results	80
6.4.1	Braking system homologation	83
7	Conclusions	85
7.1	Future works	86
A	Matlab code	88
A.1	Calibration	88
A.2	Static conditions acquisition	89
A.3	Static conditions data processing	89
A.4	Dynamic conditions data processing	91
A.5	System identification	91
A.6	Closed loop system bandwidth	94
B	Tables	95
B.1	Static characterization	95
	Acronyms	97
	Bibliography	102

List of Tables

2.1	Offset values	20
3.1	Best Fit percentages for ARX models (Appendix A.5)	41
3.2	Best Fit percentages for OE models (Appendix A.5)	42
3.3	Best Fit percentages for NLARX models (Appendix A.5)	42
4.1	K_p and K_i values for each operating region (V1)	57
5.1	CAN frames for testbench results	58
5.2	CAN Data Points for testbench results	59
5.3	K_p and K_i values for each operating region (V2)	61
5.4	Output response transient parameters on the testbench	64
5.5	K_p and K_i values for each operating region (V3)	65
6.1	K_p and K_i values for each operating region (V4)	73
6.2	Output response transient parameters on the vehicle	74
6.3	Final K_p, K_i and K_d values for each operating region, part A	77
6.4	Final K_p, K_i and K_d values for each operating region, part B	78
6.5	Parameters related to three different deceleration scenarios	81
B.1	Static characterization, upward sweep	95
B.2	Static characterization, downward sweep	96

List of Figures

1.1	Schematic view of driving task	2
1.2	Optimal slip ratio variation under different road surfaces	4
1.3	Components of a hydraulic dual-circuit car braking system	5
1.4	Ideal vehicle braking sequence	6
1.5	History of BBW systems	8
1.6	Structure of pump-type EHB	9
1.7	Structure of motor-type EHB	10
1.8	Dexter electro-hydraulic 1600 brake actuator	12
2.1	Hardware design verification table from ISO-26262	14
2.2	Test bench	15
2.3	Pressure sensor characteristic	17
2.4	USB-6009 acquisition device	18
2.5	Simulink calibration diagram	19
2.6	Static characterization test bench setup	20
2.7	Voltage divider	21
2.8	Simulink acquisition diagram in static conditions	23
2.9	Output pressure vs command voltage in static conditions	23
2.10	Output pressure vs absorbed power in static conditions	24
2.11	Standard CAN: 11-bit identifier	25
2.12	CAN I/O PLC waterproof	26
2.13	PWM modulated by a sinusoidal signal	28
2.14	Filtered voltage divider	29
2.15	Dynamic characterization test bench setup	31
2.16	Simulink transmission diagram in dynamic conditions	32
2.17	Simulink acquisition diagram in dynamic conditions	32
2.18	Eagle diagram for PWM generation	33
2.19	Open-loop bandwidth	34
2.20	Quasi-static characterization	35
2.21	Hysteresis behaviour	36
2.22	45% duty cycle step signal	37

2.23	Amplitude modulated sinusoidal wave at 0.5 Hz with duty cycle from 45 – 55% to 30 – 70%	38
2.24	Output pressure and input voltage: Uniformly distributed random signal at 1 Hz with duty cycle within 40 – 60%	38
3.1	V-model development flow, PIL phase	39
4.1	Gain scheduling operating range	45
4.2	Simulink diagram for control algorithm design	48
4.3	Simulink setup of the most general tested controller	49
4.4	Voltage to duty LUT	49
4.5	Acquisition diagram on Eagle	51
4.6	Feedback control system	52
4.7	Eagle diagram of PID controller	53
4.8	Eagle diagram interfacing with user code	53
4.9	Eagle diagram for anti-windup and PWM saturation	55
4.10	Anti-windup benefits on integrator term	56
5.1	Output response to a 20 bar step reference	60
5.2	Output response to a 20 bar step reference, with pre-braking	61
5.3	Output response to a 40 bar step reference, with pre-braking	62
5.4	Output response to a 60 bar step reference, with pre-braking	62
5.5	Output response to a 90 bar step reference, with pre-braking	63
5.6	Output response to a 110 bar step reference, with pre-braking	63
5.7	Output response to a 50 bar step reference, with pre-braking	64
5.8	Output response to a triangular reference with $T = 8$ s in the interval $(2 - 32)bar$, with pre-braking	65
5.9	Output response to a triangular reference with $T = 8$ s in the interval $(20 - 60)bar$, with pre-braking	66
5.10	Output response to a sinusoidal reference with $f = 50$ mHz in the interval $(25 - 35)bar$, with pre-braking	66
5.11	Bode diagrams of the closed-loop system	67
5.12	Output response in a neighborhood of the cutoff frequency	68
6.1	VeGA, autonomous vehicle	69
6.2	Pressure sensor electrical connections	70
6.3	CAN PLC electrical connections	71
6.4	E/H actuator electrical connections	72
6.5	Pedal logic implemented in Eagle	73
6.6	Output response to a 20 bar step reference, on the vehicle	75
6.7	Output response to a triangular reference with $T = 5$ s in the interval $(30 - 50)bar$, on the vehicle	75

6.8	Output response to a sinusoidal reference with $T = 5$ s in the interval (20 – 30)bar, on the vehicle	76
6.9	Oscillation phenomenon	77
6.10	PID derivative implementation in Eagle	78
6.11	Realistic braking profile	78
6.12	Braking test environment around Bylogix building	80
6.13	Soft braking results	81
6.14	Moderate braking results	82
6.15	Harsh braking results	82
6.16	Braking cycle for homologation	83

Chapter 1

Introduction

1.1 Autonomous driving

In recent years, the strictness of safety regulations in the automotive industry is increased, together with the growing customer demands for comfort functionalities. Therefore, this has propelled the development of intelligent vehicle technologies. These systems aim at realizing an autonomous car able to drive itself without collision by perceiving the surrounding environment [1].

Among the advantages of autonomous driving, the increased road safety has a noteworthy relevance. One of the most important parameters affecting road safety is the driver braking reaction time, which is the time elapsing from the appearance of a potential hazard up to the instant some evasive manoeuvre is initiated. [2] explains how the greatest factor affecting BRT is the expectancy. In particular, for high expectancy events, driver response is about 0.70-0.75 s, whereas for common unexpected events, such as brake lights, values increase to 1.25 s. Finally, for surprise events like path intrusions, 1.5 s are reached. These results are later criticized by [3], pointing out that, in urgent enough scenarios, latency can drop to 1 s even in surprise situations. Nevertheless, it is asserted that, whenever factors like old age, drugs, fatigue or distractions occur, reactions slow down even in urgent conditions. Indeed, [4] AASHTO establishes that a value of 2.5 s can comprise the capabilities of most drivers, by covering the 90th percentile of the measured reaction time.

Therefore, the introduction of autonomous driving systems allows to improve these latencies, since they are capable of performing with consistency under different operating conditions, being independent from the previously cited factors. Moreover, the ability of dealing with braking manoeuvres is included within autonomous driving system, which can apply braking inputs in order to appropriately handle both the longitudinal and lateral vehicle motions.

1.1.1 Classification

According to the Society of Autonomous Engineers in SAE J3016 [5], 6 different levels of autonomous driving are established. Firstly, it's necessary to introduce some definitions in order to understand how this classification works.

- Dynamic Driving Task (DDT) include all the real-time operational and tactical functions required to operate the vehicle in on-road traffic. According to SAE J3016 standards, operational functions include controlling lateral and longitudinal direction of the vehicle motion. DDT excludes strategical decisions involving trip planning, such as deciding whether, when and where to go or choosing the best routes to take.
- DDT fallback is the reaction by the user or by the ADS to either perform the DDT or achieve a minimal risk condition after the emergence of a DDT performance-relevant system failure or upon ODD exit.
- Object and Event Detection and Response (OEDR) is a subset of tasks of the DDT including monitoring the driving environment and executing a suitable response to objects and events.
- Operational Design Domain (ODD) are all those conditions (e.g. environmental) under which a given driving automation system is designed to operate.

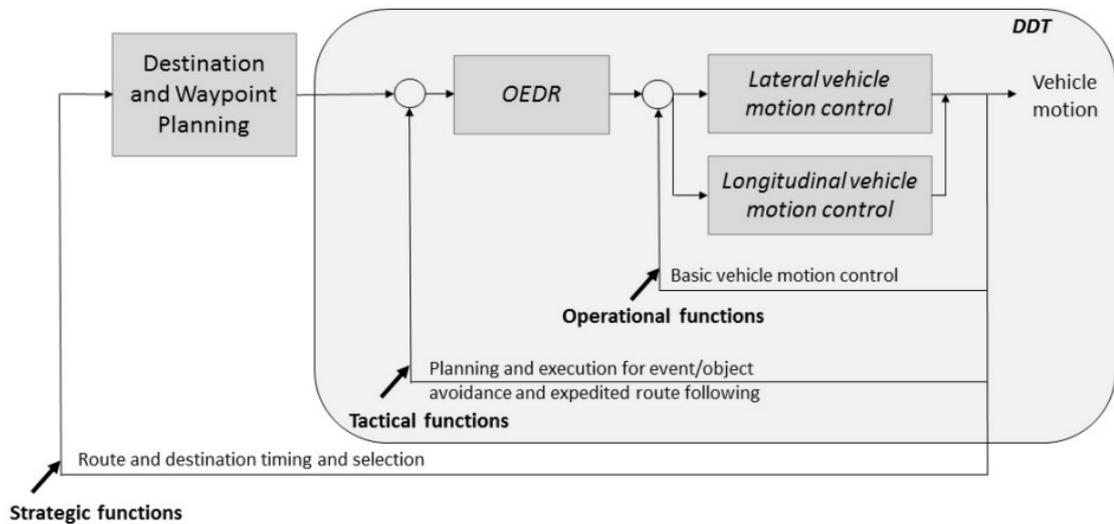


Figure 1.1: Schematic view of driving task

In figure 1.1 is described a schematic view of the driving task. It's now possible to define the taxonomy describing all the different levels.

- Level-0, no driving automation. The driver should perform the DDT for the entire time even if enhanced by active safety systems.
- Level-1, driver assistance. The ADS is able to perform only one among the lateral or the longitudinal vehicle motion control subtask, whereas the driver has to take care of what remains of the DDT.
- Level-2, partial driving automation. ADS executes the entire basic vehicle motion control, leaving to the driver the completion of the OEDR subtasks and the supervision of the driving automation system.
- Level-3, conditional driving automation. ADS is able to carry out the DDT in its entirety, with the expectation that the user is receptive and ready to suitably respond, in case ADS failures occurs.
- Level-4, high driving automation. ADS is responsible for both the ADS and the ADS fallback without any expectation of the user response to an intervention request. The ODD is still limited.
- Level-5, full driving automation. Under any condition, the ADS performs the entire DDT undertaking DDT fallback procedure if needed.

1.1.2 Active safety systems

According to [5], driving automation doesn't include active safety systems, since their triggering doesn't substitute driver role in performing part or the entirety of the DDT on a sustained basis. Indeed, active safety systems are all those ADAS which help avoiding or mitigating hazardous situations while driving a vehicle, by delivering a temporary intervention. In order to be quick in intervening, they constantly monitor both outside and inside vehicle conditions to perceive potential dangers.

For this reason, these kind of systems require good performances like stability, speed of response and reliability, both over time and under the most critical situations. Within the listing of all the conceived active safety systems, two of the major ones related to braking systems are ABS and AEB.

Anti-lock braking systems prevent brakes from locking the wheels when severe braking events occur or in presence of slippery road surfaces. By keeping rolling, the wheels don't lose adhesion with the road, allowing the driver to preserve control of the vehicle steerability.

$$\lambda = \frac{v - \omega R}{v} \tag{1.1}$$

ABS acts by controlling the wheel slip ratio 1.1 and aims at maintaining it in a stable region, where the tire-road friction coefficient results to be maximum. As

showed in figure 1.2 [6], the optimal slip ratio maximizing the grip coefficient varies with road conditions. Anyway, the majority of ABS work in a slip region between 0.1 and 0.3 which is fine for most situations.

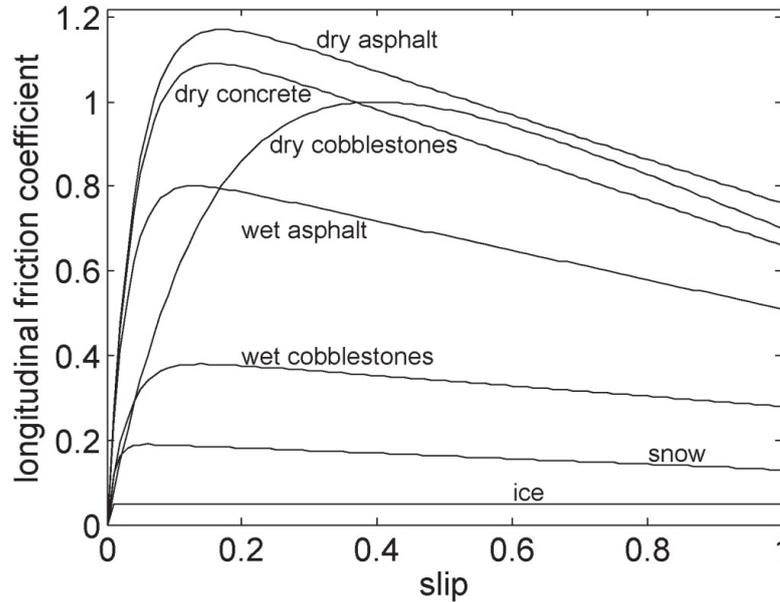


Figure 1.2: Optimal slip ratio variation under different road surfaces

Instead, AEB systems automatically start a braking manoeuvre whether a collision is imminent and the driver is too slow in reacting properly. If the impact can't be avoided, at least they allow to mitigate it.

1.2 Braking system

The braking system is defined by UNECE [7] as the combination of components whose function is to progressively reduce the speed of a moving vehicle or bring it to a halt, or to keep it stationary if it is already halted. It can be decomposed into three main parts:

1. Brake: the part which generates the force opposing to the vehicle motion.
2. Control: includes all the devices devoted to provide the energy transmission required for braking. Those parts can be actuated either by the driver or by the autonomous driving system.
3. Transmission: combinations of components comprised between brake and control and linking them functionally.

The braking system is one of the most critical systems for what concerns the active safety of a vehicle. Indeed, in order to guarantee reliability of braking functions, components redundancy is often employed. Law requirements impose braking systems to adopt a dual-circuit configuration, which can assume five possible combinations, according to the European standard DIN 74000 [8]. For instance, braking circuit of figure 1.3 leverages a II configuration which involves a front axle-rear axle split. The master pump actuates both the rear and the front brakes through two completely independent hydraulic circuits. Therefore, they have the opportunity to work together as service circuit or, whenever one of the two fails, as emergency circuit [9].

Generally, a braking system has to accomplish three different tasks: to completely arrest the vehicle, to modulate the vehicle speed and to maintain a vehicle still on a slope.

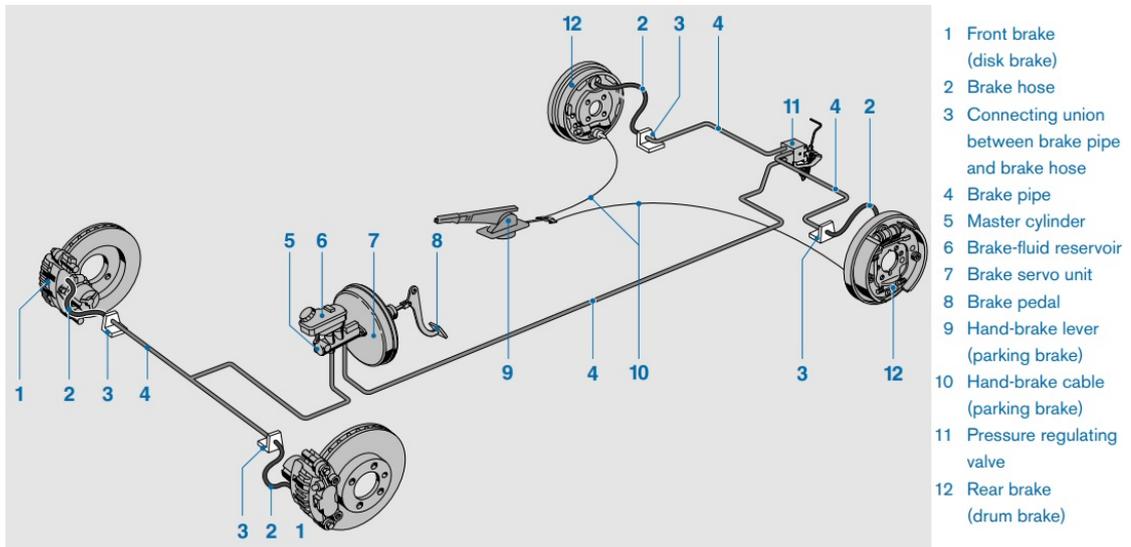


Figure 1.3: Components of a hydraulic dual-circuit car braking system

According to regulations, these functionalities has to be carried out by three different systems:

- The service braking system, whose purposes are to reduce the velocity or to safely halt the vehicle in a fast and effective way, whichever the speed and the load.
- The emergency or secondary braking system, which can perform the previous functions as well, and can operate even if a failure in the service brake occurs.
- The parking braking system, whose role is only to keep the vehicle in stationary position, even on inclined roads.

This dissertation focuses on a service braking system consisting of disc brakes and hydraulic transmission, whereas the brake control action aims at being delivered by the autonomous driving system, with the opportunity of the driver to take control of the braking manoeuvre.

1.2.1 Braking sequence

Braking sequence is introduced to define some parameters which are useful for determining the quality of the braking action.

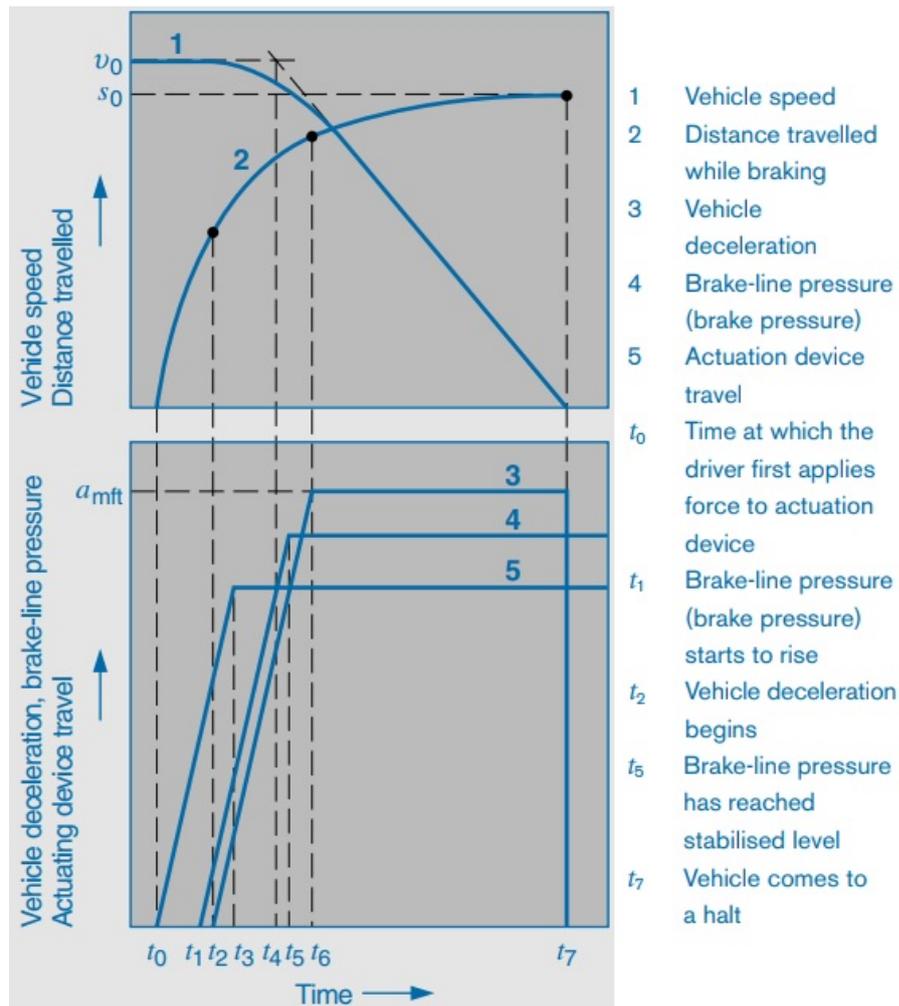


Figure 1.4: Ideal vehicle braking sequence

Referring to figure 1.4 [8], some quantities can be defined:

- Response time ($t_1 - t_0$): time elapsed from the instant at which force is applied

to the actuation device to the point at which pressure is first generated.

- Pressure build-up time ($t_5 - t_1$): time from the moment pressure is first produced to the point at which pressure reaches its highest level.
- Total braking time ($t_7 - t_0$): time from when force is applied up to the moment the braking force ceases. In case vehicle comes to halt, then it coincides with the instant at which the vehicle is first stationary.
- Total braking distance: distance travelled by a vehicle during the total braking time.

1.3 Brake-By-Wire

With the growth of electrical vehicles and the evolution of autonomous driving, also automobile braking systems have undergone some improvements. Traditional hydraulic braking systems have relentlessly become insufficient in satisfying the evolution of comfort, performance and safety requirements. BBW showed up in the scene, allowing the removal of some hydraulic components and by providing outstanding advantages. The vacuum booster is suppressed, and the driver pedal is decoupled from the wheel calipers, which, in EHB, remain hydraulically actuated and can be re-connected to the pedal in case of fault [10]. Indeed, in these systems, pedal stroke is perceived by a sensor, and the information travels through electrical signals to an ECU, which controls some mechatronic subsystems to perform the braking action. The pedal unit is formed by a system, called PFE, made of elastic bodies, which aims at generating the desired pedal feeling. Moreover in EHB, TMC is still used either as an intermediate transmission or as a failure backup system, depending on the configuration. Among the benefits provided by this technology [11] [12], there are high braking efficiency, fast and reliable response, high flexibility, improved pedal feeling, increased safety, possibility of independently control the braking force on the four wheels, increased service life, simple and compact structure and easy integration with active safety systems. Brake-By-Wire split in two main categories which are electro-hydraulic brakes and electro-mechanical brakes. Figure 1.5 [13] reports the historical development of this technology.

EMB systems are the most recent ones, owning the big advantage of a complete removal of all the hydraulic components, which increases the transmission efficiency and the environmental protection, due to the absence of risk of possible leakages. EMB works by driving reduction gears and a ball screw mechanism to directly push the pads to clamp the wheel disc, generating the friction braking torque [13]. Nevertheless, they also present some disadvantages, like the need of deep modification in the vehicle structure and the absence of current regulation fulfillment for what concerns brake system failure backup [14]. Indeed, although some companies and

universities have developed some prototypes, they haven't entered the market yet. EHB instead, by relying on traditional hydraulic braking systems, require less modifications than EMB and allow an easier implementation of the failure backup, thus making this technology the first approach to BBW.

Due to the high complexity of measuring the braking force acting on the wheels, EHB exploit closed-loop pressure control to indirectly regulate the braking force. The measure is performed through a pressure sensor, which is a critical component for what concerns the vehicle safety, since a failure can seriously affect the performances of the control algorithm. Redundancy solutions may improve the overall safety, like the insertion of a further sensor with a suitable algorithm, able to handle the two measurements.

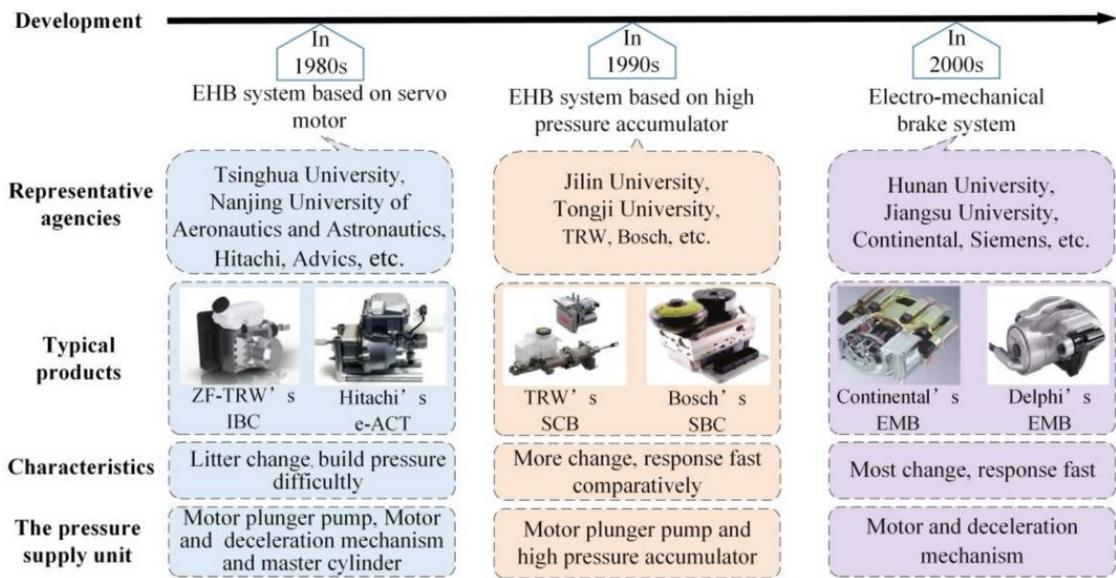


Figure 1.5: History of BBW systems

Since this dissertation focuses on a custom EHB BBW solution, further focus on the structural differences between the two possible EHB configurations is posed.

1.3.1 Pump-type EHB

From the 1990s to the early 2000s, motor technology was still immature [13]. In order to solve brake requirements on high pressure and fast performances, EHB systems using high-pressure accumulator as energy storage and supply device have been employed. These systems build up high pressure thanks to the motor plunger pump and save it in the accumulator in advance. When the brake starts, the high pressure is released by the ECU through a complex system of solenoid valves and it reaches the wheel cylinders.

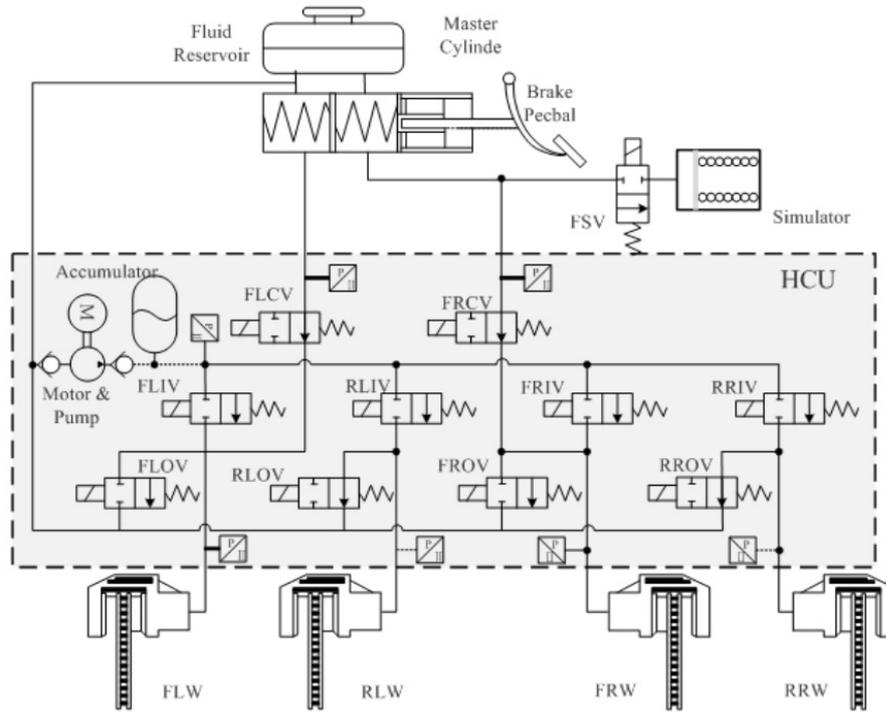


Figure 1.6: Structure of pump-type EHB

Figure 1.6 [15] shows a possible structure of a pump-type EHB system. During usual operating conditions, the control valves FLCV and FRCV are closed and the fail safe valve FSV is open, thus linking the pedal with the PFE. The TMC determines the pedal feeling only for the idle travel corresponding to the closure of the orifice which puts the first chamber in communication with the reservoir [10]. Under normal operating conditions, there are three possible scenarios [12]:

- Pressurization of the wheel cylinder. The inlet valves (FLIV, RLIV, FRIV, RRIV) are opened and the outlet valves (FLOV, RLOV, FROV, RROV) are closed, so that the high pressure fluid coming from the accumulator can enter the wheel cylinders.
- Decompression of the wheel cylinder. The inlet valves are closed, whereas the outlet valves are opened, so that the fluid can flow back to the oil tank through the low pressure pipes.
- Retention of the pressure level. If the cylinder pressure needs to be maintained, both inlet and outlet valves are closed.

The pump, the accumulator and the state of the valves are controlled by the HCU through a suitable control algorithm, depending on the pedal stroke.

If a fault occurs, the backup safety system goes into action [10]. This conditions is similar to the one of a classical hydraulic braking system without the vacuum booster. FLCV and FRCV valves, corresponding to the front axle, are opened, allowing to couple the pedal with the braking system. Instead, the FSV fail save valve is closed in order to avoid a significant pedal travel, which is given by the deformation of the elastic elements of the TMC due to the wheel caliper volume displacement.

Several pressure sensors are spread over the circuit to keep track of the system state in each point.

1.3.2 Motor-type EHB

The presence of the high pressure accumulator in the pump-type EHB has a negative effect on the service life, together with lower safety and reliability, due to the risk of possible leakages. For these reasons [16], EHB with motor as power source has become a good alternative solution, despite being invented in the 1980s. Another big advantage of this solution is to reduce the required amount of changes to the overall braking structure. Also in motor-type EHB, under normal operating conditions the pedal is isolated from the wheels calipers. In this case, instead of valves, a decoupling chamber is employed as a failure switching device. Moreover, as for the pump-type EHB, a haptic pedal feedback is given thanks to the PFE element.

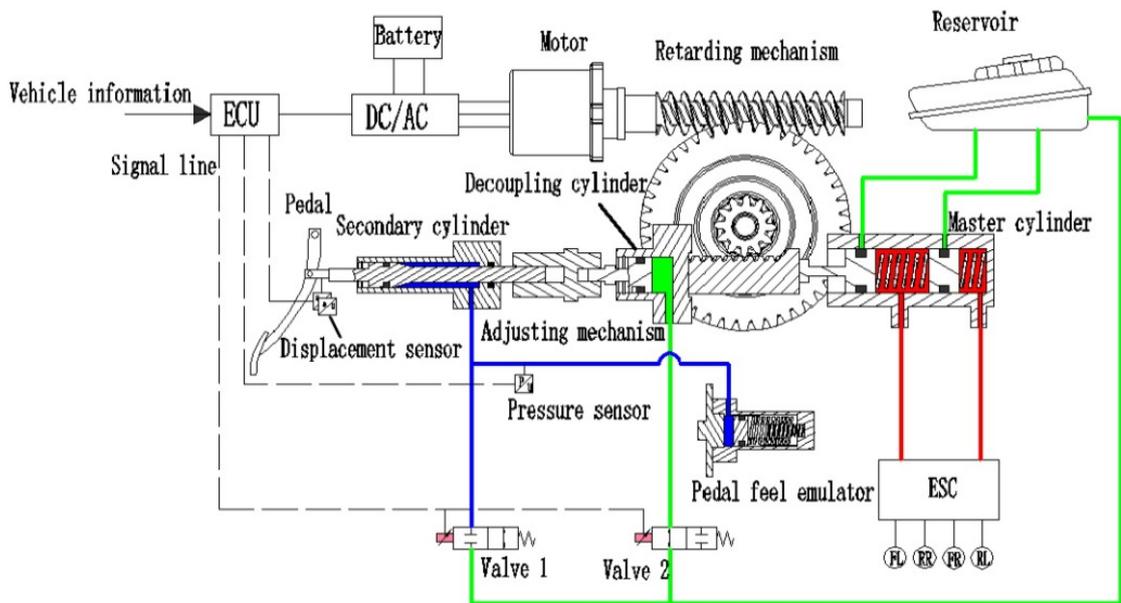


Figure 1.7: Structure of motor-type EHB

The working principle of this technology is based on an electro-mechanical actuator, consisting of an electric motor, which directly pushes the master cylinder through a linear motion mechanism to build hydraulic pressure [17]. With reference to figure 1.7, solenoid valve 1 is normally open, whereas solenoid valve 2 is normally closed. The figure represents the system configuration under normal operating conditions, where both valves are powered on. Valve 1 is closed so that the oil can flow from the secondary cylinder into the PFE to generate pedal feel. Valve 2 is opened to keep the wheel calipers decoupled from the pedal and the oil flows back to the reservoir.

If a failure in the motor occurs, valve 1 is opened to allow pedal traveling, whereas valve 2 is closed to create a rigid connection between the TMC and the driver pedal, which can directly control the wheel calipers. If also a leakage in the decoupling cylinder arises, it is compressed totally, in order to restore the coupling with the TMC. The only difference with respect to the previous scenario is that the pedal has to overcome idle travel in the decoupling cylinder, corresponding to the length of the chamber.

Although the evident structural differences between the two EHB technologies, they have something in common. Both the configurations are by-wire solutions, indeed the pedal stroke or force is perceived by a nearby sensor and the information is conveyed through an electrical signal to a dedicated ECU. This information is used to properly modulate the pressure into the braking system. Furthermore, as previously discussed, both of them have a backup system in case of emergency scenarios where a failure occurs. In these cases the driver can take control of the brake manoeuvre, by directly actuating the hydraulic braking system through the pedal.

1.3.3 BBW control

Braking manoeuvre is a short process, which requires the actuator to quickly respond to the pressure request coming from the ECU [12]. However, in control theory, fast response and precise control are two contradictory parameters. The pursuit of a fast response will inevitably provoke an excessive overshooting of the braking force. Therefore, it's necessary to find a good trade-off in order to balance braking comfort and responsiveness.

Moreover pressure control results to be a challenging and critical problem. Indeed, it's strongly influenced by nonlinearities such as friction and by uncertainties like temperature variation or brake pads wear [18].

According to the literature, there are several proposed pressure control algorithms which can enhance EHB performances, such as PID control [17], adaptive sliding mode control [19][18] or backstepping control [15]. The objective of this dissertation is to design a non-linear pressure control algorithm, which can be easily implemented

on the braking ECU while providing good performances.

1.4 Thesis objectives

This thesis aims at designing an autonomous braking system, by controlling an E/H actuator. The pump is eventually installed on a real L4 experimental autonomous vehicle owned by the company Bylogix s.r.l..



Figure 1.8: Dexter electro-hydraulic 1600 brake actuator

Although state-of-art BBW systems provide several advantages [12][11][20], they require considerable modifications to the vehicle structure in order to be implemented. Thus, [21] has proposed a solution which involves a suitably designed mechanical system exploiting an electric DC motor. The suggested solution aims at actuating the legacy preinstalled braking system by directly operating on its braking pedal. Despite this solution has provided good simulation results, it has stranded before being mounted on the vehicle, hence the necessity of a new solution. In this dissertation is introduced a particular EHB BBW system, which doesn't disrupt the vehicle structure. This solution exploits an E/H actuator, which can directly generate the pressure in the braking circuit. Unlike classical BBW, the BCU regulates the pump action on the base of the information provided by the autonomous driving unit instead of the driver. Thus, the pedal stroke is not encoded into a pressure request and no deep modification to the system is required. Indeed, conversely to classical BBW, all the hydraulic connection between pedal

and wheel calipers is preserved, and no PFE is added. The driver can take direct control of the braking manoeuvre in any moment by simply pressing the pedal, which disables the actuator. Hence, both the driver through the pedal and the autonomous driving system through the actuator have the opportunity of exerting pressure in the braking system.

The device utilized for this solution is the Dexter E/H 1600 brake actuator (figure 1.8, [22]), which is commonly employed in trailer braking systems. Being conceived for heavy load systems, it should be likewise able to provide the pressure needed to stop a car.

Indeed, after a static and dynamic characterization, a gain-scheduled PID pressure control algorithm with anti-windup compensation is designed and its performance are tested. Finally, all the system is integrated on the vehicle, where concluding tests are performed to verify the pump capability of fulfilling some desired braking performances, when leveraged for a vehicle application.

Chapter 2

System characterization

The characterization phase is an important stage for the study of the system behaviour. By stimulating the pump with different types of inputs and by properly measuring the outputs, it's possible to deduce some important information about the system. In particular, characterizing a system allows to identify possible nonlinearities and to bring out critical working conditions, which may even lead to hardware failures. Indeed, according to the hardware verification table (2.1) [23], static and dynamic analysis is one of the alternative entries recommended by ISO-26262, which is an international standard concerning functional safety of road vehicles. Thus, it's necessary to build up an adequate acquisition system, able to correctly interface with the actuator and to suitably stimulate it. In addition, this will be eventually convenient by the time the control algorithm has to be tested.

Methods and measures		According to req.	ASIL			
			A	B	C	D
1a	Hardware design inspection	6.4.4.1	+	+	++	++
1b	Hardware design review	6.4.4.1	++	++	++	++
2	Failure analysis, e.g FMEA, FTA	6.4.4.1	+	++	++	++
3a	Static analysis	6.4.4.1	+	+	+	o
3b	Static analysis AND Dynamic analysis	6.4.4.1	o	+	+	+
3c	Simulation	6.4.4.1	o	+	+	+
3d	Dynamic analysis AND « Worst case » analysis	6.4.4.1	o	o	o	+

Figure 2.1: Hardware design verification table from ISO-26262

2.1 General setup

2.1.1 Test bench

Together with Bylogix company, in order to perform the actuator characterization, a tailor-made test bench (figure 2.2) roughly replicating the real environment has been conceived .

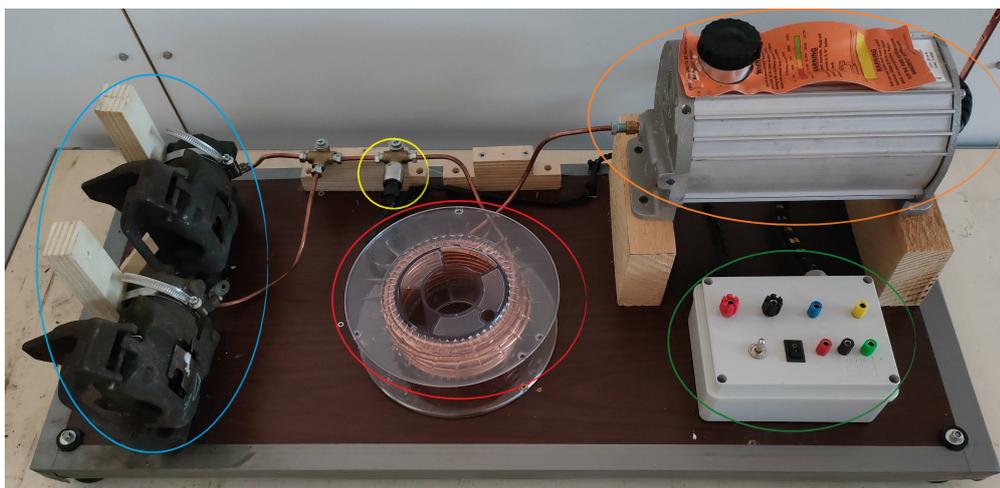


Figure 2.2: Test bench

With reference to figure 2.2, the elements highlighted with different colors are defined:

- Orange circle: E/H actuator under analysis and capable of delivering a maximum nominal pressure of 1600 psi, corresponding approximately to 110 bar. The pump is supplied with an ITECH IT6512C DC power supply which is able to deliver up to 80 V or 120 A for a total maximum power of 1.8 kW. In order to ensure having at least 12 V for the actuator supply, as required by [24], 14 V are always applied by the power supply. Since cables have their own resistance, some voltage drop may occur along the path in the presence of high current values.
- Red circle: Copper brake line approximately 10 m long which tries to replicate the compliance behaviour of car connections. According to [25], in the physiology discipline, compliance is the ability of an empty organ of increasing its volume when the internal pressure raises and the ability of resisting recoil to its original dimension when subjected to a compressing force. Brake lines are not perfectly rigid but show a certain level of compliance, thus it's necessary to model this behaviour on the test bench

by adding a copper coil between the pump and the calipers.

For what concerns the brake fluid, even though some air is always present in the circuit, it's possible to approximate it as not compressible. The actuator tank is filled with brake fluid, which is pumped within the braking lines and it comes back by inertia whenever the command is switched off.

- Yellow circle: Piezoresistive pressure sensor measuring the output pressure. In particular the chosen device is the SP150-M10x1 from Aviorace S.r.l. and it is able to measure pressure up to 150 bar, hence it is consistent with the actuator maximum nominal pressure.
- Blue circle: Brake calipers clamping a piece of steel simulating wheel discs.
- Green circle: Box with banana sockets for all the required connections and two switches to power on-off both the actuator and the sensor. In particular the upper red socket is the positive power supply of the actuator which, in standard conditions, is required to be 30 A 12 V, whereas the black socket is the ground reference voltage of the vehicle, as cited in the actuator datasheet [24]. The blue socket instead is connected to the command voltage, which is the input of our system, whereas, according to [24], the yellow socket should be connected to an emergency breakaway switch, which is currently unplugged on the test bench setup. Finally on the bottom row of the box, the red socket is the positive power supply of the sensor which is required to be 5 V, the black socket is the ground reference voltage to be connected to the previous one in order to have a common ground, and lastly, the green socket is the voltage output measurement coming from the pressure sensor.

The system under study results to be a SISO system, where the input is the command voltage able to control the actuator, whereas the output is the pressure delivered by the pump throughout the brake line.

2.1.2 Pressure sensor

According to the datasheet [26], the pressure sensor outputs a voltage which falls within the range 0.5 V – 4.5 V, which corresponds linearly to the pressure scale 0 bar – 150 bar, thus resulting in the characteristic represented in figure 2.3. In the following section, this characteristic is converted into a LUT within the Simulink schemes for the calibration and acquisition phases.

This sensor is supplied with the 5 V delivered by the USB-6009. According to [26] the maximum current absorbed by the sensor is equal to 5.5 mA, whereas according to [27] the maximum current delivered by the acquisition board is equal to 200 mA, thus making the USB-6009 compatible for supplying the sensor in the test bench.

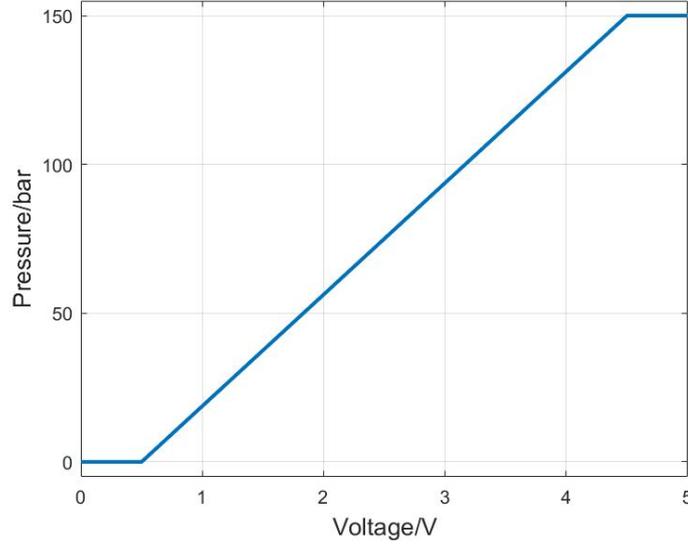


Figure 2.3: Pressure sensor characteristic

From the characteristic it's possible to obtain the related function by using the equation of a line through two points:

$$\frac{y - y_0}{y_1 - y_0} = \frac{x - x_0}{x_1 - x_0}$$

where $P_0 = (x_0, y_0) = (0.5, 0)$ and $P_1 = (x_1, y_1) = (4.5, 150)$ from which is obtained the final relationship:

$$y = x \cdot \frac{150}{4} - \frac{75}{4} \quad (2.1)$$

where x is the measured voltage and y the corresponding measured pressure.

2.1.3 Acquisition device

The acquisition device exploited is the USB-6009 from the NI and it is showed in figure 2.4 [28]. This board is capable of working in real time within the Simulink environment, after the installation of the DAQ toolbox and of the hardware support package for DAQmx devices. This device has 13 bits of resolution for single-ended measurements, thus resulting in 2^{13} quantization levels. Since in this device the maximum measurable voltage is equal to 10 V, the voltage gap between two adjacent quantization levels is equal to:

$$\Delta V = \frac{10}{2^{13}} = \frac{10}{8192} = 1.2 \text{ mV} \quad (2.2)$$



Figure 2.4: USB-6009 acquisition device

The sampling time of the analog input Simulink block, from the DAQ toolbox, has to be an integer multiple of the simulation step imposed within solver settings and it is going to be chosen accordingly in the following sections.

Instead, solver simulation step must be chosen considering that the smaller the size, the higher are the chances to detect fast changes, but as a drawback, it may slow down the simulation due to the higher number of output points produced. A fixed step with size 10^{-5} is chosen, together with automatic solver option.

2.1.4 Calibration

All the devices need to be calibrated by removing possible offsets, in order to obtain precise measurements. In particular, concerning the sensor, a pressure equal to 0 bar corresponding to 0.5 V has to be read, whenever the actuator is not active. Moreover, also the acquired input voltage needs to be equal to 0 V, whenever either the command is constant equal to 0 V or it is a PWM with a duty cycle equal to 0%. To do so, it's necessary to design a calibration Simulink scheme (figure 2.5) to be executed before the acquisition phase. Analyzing the elements in figure 2.5:

- The "Analog input" block allows Simulink to interface with the acquisition device USB-6009. Within the block the following settings have been chosen:
 - Asynchronous acquisition mode. Whenever the simulation starts, also the acquisition starts. Moreover, the simulation runs while data is acquired into a FIFO buffer.

- Both the acquisition channels have been setup in single ended configuration, since we are just measuring positive voltages with respect to ground.
 - Since in calibration no dynamic is considered, a sampling frequency $f_s = 5$ kHz is plenty.
 - A block size of 500 is chosen, where, according to MATLAB documentation [29], the block size is the number of data samples to read from the block output at each time step for each channel.
- The two "Unbuffer" blocks are necessary in order to unbuffer the chunk of data coming out from the analog input acquisition block.
 - The "LUT dynamic" block implements the linear region of the pressure sensor characteristics of figure 2.3. Within the block, the lookup method Interpolation-Extrapolation is exploited, in order to correctly compute the offset value.
 - "To workspace" blocks are used to transfer all the acquired data to the MATLAB script for data post-processing.

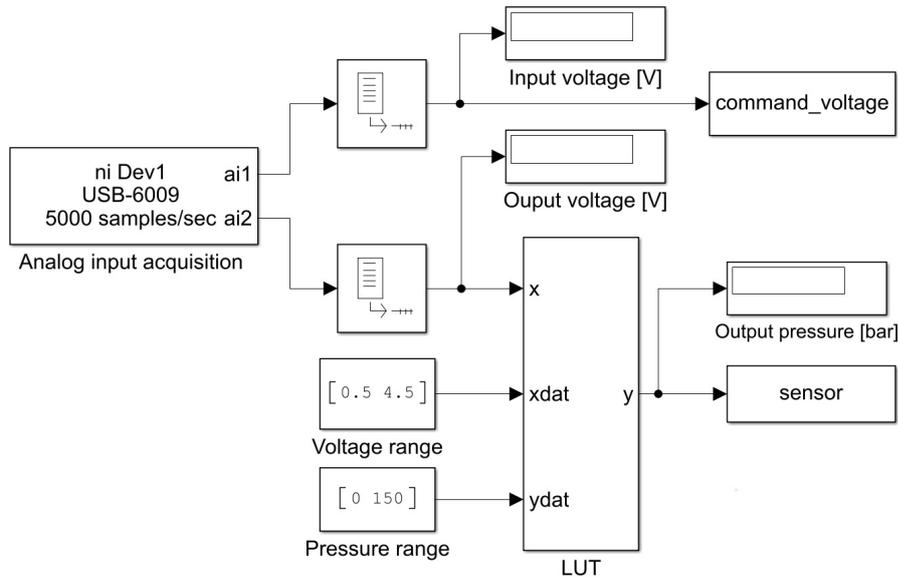


Figure 2.5: Simulink calibration diagram

The command input is coming from the EA-PS 2084-05B power supply which is able to deliver 84 V and 5 A for a maximum total power of 160 W. In this setup, power supplies of both the sensor and the actuator are switched on, and the constant command voltage coming from the sensor is set to 0 V.

2.2.2 Voltage divider

According to the board datasheet [27], the maximum positive measurable voltage on any pin is 10 V with respect to ground. Connecting a voltage greater than this value would result in a clipped output, therefore it's necessary to design a voltage divider (figure 2.7), which can allow to acquire voltages bigger than 10 V.

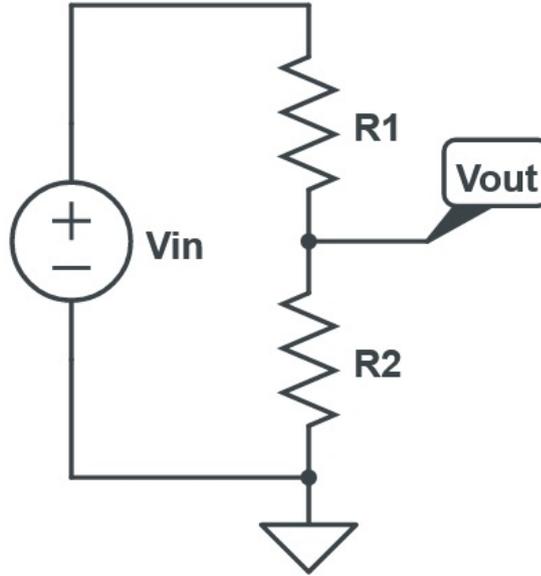


Figure 2.7: Voltage divider

Voltage divider resistors must be suitably designed. The chosen ones are able to dissipate 0.25 W before overheating and burning out. To avoid any kind of risk, it's safer to consider 0.1 W as maximum dissipated power. Moreover, must be considered that the bigger the resistor value, the lower will be the dissipated power $P_{max} = V_{max}^2/R$, as a drawback, the higher is the risk that it can collide with the internal resistance of the USB-6009, altering the measurement.

Considering the input impedance of the acquisition board equal to $R_L = 144 \text{ k}\Omega$, let's analyze what happens to the measured voltage whenever two equal ideal resistors of $1 \text{ k}\Omega$ are employed. $R_L = 144 \cdot R_2$ and R_L is parallel to R_2 , thus:

$$R_2 || R_L = \frac{R_2 \cdot R_L}{R_2 + R_L} = \frac{R_2 \cdot 144R_2}{R_2 + 144R_2} = \frac{144}{145}R_2 = 0.9931R_2$$

A load resistor of that dimension results in a reduction of R_2 resistor of almost 0.7%. But what happens to the measured voltage?

$$V_{out} = V_{in} \frac{0.9931R_2}{0.9931R_2 + R_1} = V_{in} \frac{0.9931}{0.9931 + 1} = V_{in} \frac{0.9931}{1.9931} = 0.4983V_{in}$$

from which is possible to compute the error with respect to the theoretical voltage:

$$error_{\%} = 100 - \left(\frac{0.4983}{0.5} \cdot 100 \right) = 0.34\%$$

The actual output of the voltage divider results to be 0.34% lower than the target voltage. This error is low enough to be considered negligible, hence making the 1 k Ω resistors suitable for the application.

Let's now verify if also the power conditions are satisfied. Referring to figure 2.7 and to the following voltage divider formula:

$$V_{out} = V_{in} \cdot \frac{R_2}{R_2 + R_1} \quad (2.3)$$

by considering two equal resistors of 1 k Ω each and a maximum voltage drop of 12 V, the maximum voltage over each resistor results to be $V_{max} = 6$ V. Thus, from Ohm's law 2.4:

$$V = R \cdot I \quad (2.4)$$

the maximum current passing through the resistor is $I_{max} = V_{max}/R = 6$ mA. Being power $P = V \cdot I$, the maximum dissipated power condition is therefore satisfied: $P_{max} = V_{max} \cdot I_{max} = 72$ mW \leq 100 mW.

Whenever the original input voltage is going to be computed back, in order to obtain precise results, the real resistor values are used: $R_2 = 982$ Ω and $R_1 = 980$ Ω . Therefore, by reversing the voltage divider formula 2.3, it's possible to obtain:

$$V_{in} = V_{out} \cdot \frac{R_2 + R_1}{R_2} = V_{out} \frac{1962}{982} \approx 1.998 \cdot V_{out} \quad (2.5)$$

2.2.3 Simulink acquisition diagram

Figure 2.8 represents the designed diagram for the acquisition phase in static conditions. The only difference with respect to the calibration diagram of figure 2.5 is the subtraction of the previously calculated offsets. Since during calibration the offset has been computed on resistor R_2 as well as the voltage here acquired, it is removed before obtaining back V_{in} from formula 2.5. The reverse voltage divider is applied after the acquisition, within the MATLAB code reported in appendix A.2.

2.2.4 Acquisition results

The E/H actuator is stimulated with command values in the interval 2.4 V – 12.2 V and the voltage is gradually changed with spans of 0.4 V. The measurements

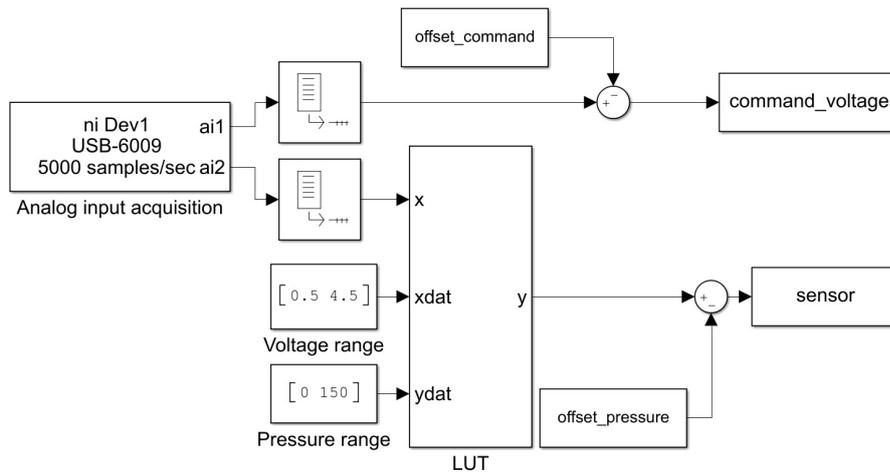


Figure 2.8: Simulink acquisition diagram in static conditions

are performed both by increasing the command up to the maximum value and by decreasing it to the minimum one. After a new voltage value is set, before performing the measurement, some time is waited to allow the pump to reach stable steady state conditions. Measurements are taken from 2.4 V, value at which the actuator starts pumping, up to 12.2 V, out of which no significant performance improvement can be observed.

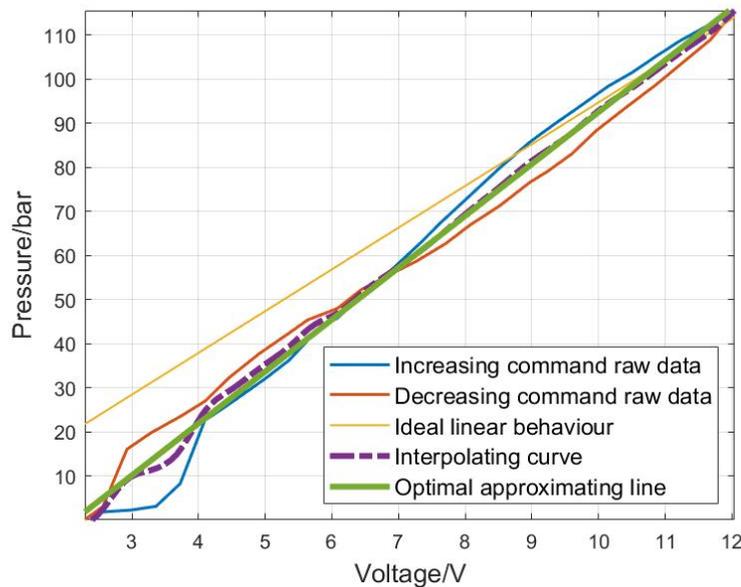


Figure 2.9: Output pressure vs command voltage in static conditions

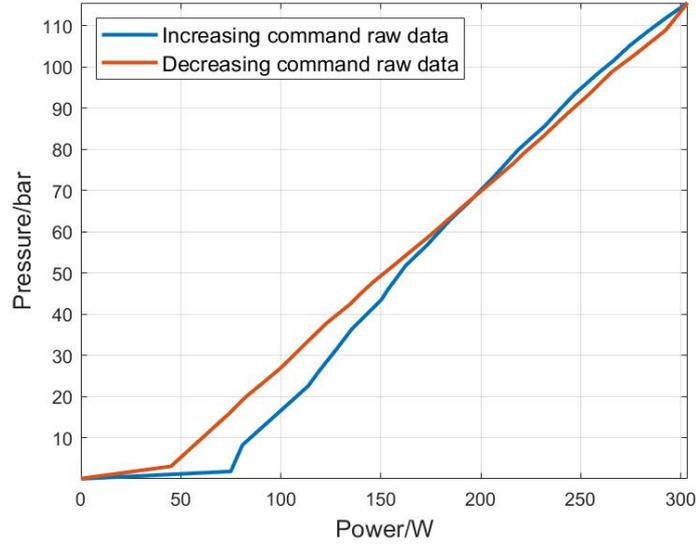


Figure 2.10: Output pressure vs absorbed power in static conditions

A total of 51 measurements are collected and they are reported in tables B.1 and B.2, both for the upward and the downward voltage sweep. Moreover, current measurements are performed by using a current probe, from which it's possible to derive the maximum absorbed power by the E/H actuator in static condition.

$$P_{max} = I_{max} \cdot V_{supply} = 25.28 \text{ A} \cdot 12 \text{ V} \approx 303.4 \text{ W} \quad (2.6)$$

By manipulating all these collected values (code reported in appendix A.3), it's possible to obtain figure 2.9. As previously observed, both the extremities of the characteristic are chosen on the base of the system behaviour. In particular, what happens outside of these limit values is a non-linear phenomenon of saturation. Indeed, as showed in figure 2.9, the characteristic starts from 2.4 V, since, up to that value, the pressure is kept equal to 0 bar and no current is absorbed. This issues the presence of a negative offset in the static characteristic, which is denoted by the difference with respect to the ideal linear behaviour passing through the origin. On the other extremity, by increasing the command over 12.2 V, the pressure is capped, due to the physical limitation of the actuator. Moreover, it can be noticed that increasing and decreasing pressures follow two different paths. Anyway, it's possible to derive an interpolation, whose path is close to both the curves and which is close to a linear behaviour. Indeed, by computing the optimal approximating line, is evident the proximity with the interpolating curve.

Finally, from figure 2.10, it's possible to notice that a consistent amount of power is absorbed by the actuator in the first phases, despite the pressure keeping low

values. This may be due to the initial effort, which is necessary to bring the pump at full regime.

2.3 Dynamic characterization

The dynamic analysis consists in imposing a time varying signal at the input of the system and by measuring the corresponding output. In this case, where the input is a PWM, the time varying signal is a modulating signal which is driving the duty cycle of the PWM wave.

2.3.1 Controlled-Area Network

The Controller Area Network is a ISO 11898 communication protocol which has been first developed to solve point-to-point wiring connections in vehicle applications. In particular, according to [30], CAN protocol is a multi-master serial bus which provides several advantages. For instance, it permits to move data with high transmission speed up to baud rates of 1 Mbit s^{-1} , and the transmitted data is very reliable and equipped with a robust error detection.

Information travels through twisted-pair cables in the form of differential voltage between CAN-H and CAN-L, making the communication strong against noise coupling [31].

Both the ends of the line are terminated according to standards with 120Ω resistors, which have the objective to avoid signal reflections by matching the characteristic impedance of the line.

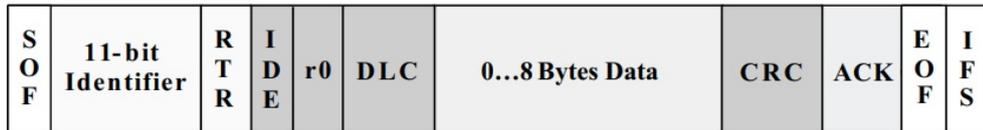


Figure 2.11: Standard CAN: 11-bit identifier

The protocol consists of sending messages which are identified by 11 bits, according to the standard frame. In figure 2.11 [32] is represented the message structure for the standard CAN. The meaning of the main bit fields are:

- SOF: Single dominant start of frame indicates the beginning of a message.
- 11-bit identifier to establish the priority of a message, the lower the binary value, the higher the priority.

- RTR: Remote transmission request is a single bit to identify whether the frame type is a data frame or a remote frame. A remote frame is a request of information from another node, hence this kind of frame doesn't contain the data field. This bit is always dominant ('0') in data frames.
- IDE: Single dominant identifier extension bit to mark that a standard CAN identifier is being transmitted.
- DLC: 4-bit data length code containing the total number of the transmitted data bytes.
- Data: A maximum amount of 64 bits can be transmitted.
- CRC: 16-bit for cyclic redundancy check to detect if corruption occurred during the transmission. The receiver computes the CRC and compares it with the CRC computed and received from the sender. If the two CRCs don't match, an error is issued.
- ACK: 2-bit where 1 bit is the acknowledgement bit and the other is a delimiter. Every time an accurate message is received, the recessive bit '1' is overwritten with a dominant one by the receiving node.
- EOF: End of frame 7 consecutive recessive bits marking the end of a message.



Figure 2.12: CAN I/O PLC waterproof

Figure 2.12 shows the CAN control unit where the control algorithm will be implemented. In particular the chosen device is the CAN I/O PLC waterproof

model 1.053.300.0000 from MRS Electronic. This device is chosen to dynamically characterize the actuator, with the purpose of approaching to the final setup on the vehicle.

In order to easily interface the test bench with the control unit, a dedicated box with banana sockets designed by Bylogix s.r.l. is employed. According to ISO 11898, a $120\ \Omega$ resistance is added at one terminal of the CAN line to allow correct operating conditions for the control unit. Moreover, KL15 ignition pin requires to be always activated through the dedicated switch on the box.

MRS Developer Studio software is employed for the programming of the CAN module, in particular this is carried out through the included graphical programming interface named Eagle. Since the final working environment of the actuator is a vehicle, from now on, the 500 kbits s^{-1} automotive standard baud rate is set within MRS Developer Studio environment. In order to interface the CAN PLC module with the laptop where Simulink is running, a PCAN-USB device from PEAK-System Technik GmbH company has been used. Finally, on Simulink the Vehicle Network Toolbox is utilized to interface the laptop with the CAN bus.

2.3.2 Pulse-width modulated signal

The type of signal chosen to pilot the pump is a PWM, which is a technique commonly employed to control analog circuit with the digital output of a microcontroller. PWM allows to encode an analog wave (modulating signal) into a digital signal (carrier) discretized in amplitude and it has the following expression:

$$v_{PWM}(t) = \begin{cases} V_H & \text{for } 0 \leq t < T \\ V_L & \text{for } T < t \leq T_{PWM} \end{cases} \quad (2.7)$$

from which it is possible to define the duty cycle $d = T/T_{PWM}$ driven by the modulating signal and the related mean value $V_{mean} = (V_H - V_L) \cdot T/T_{PWM}$. In figure 2.13 it's possible to observe a PWM whose duty cycle is modulated by a sine wave. A pulse width modulated signal is chosen over an analog signal due to the following reasons:

- Power loss consumption in switches is very low. No current passes through the switching device when it is off, whereas when it is on, almost the whole power is transferred to the load since there is nearly zero voltage drop across the switch. Being $P = V \cdot I$, consequently in both cases the dissipated power is close to zero.
- PWM works fine with digital controls due to their ability of switching on/off.
- The utilized CAN PLC controller doesn't have analog output pins, thus for hardware limitation it couldn't be possible to do otherwise.

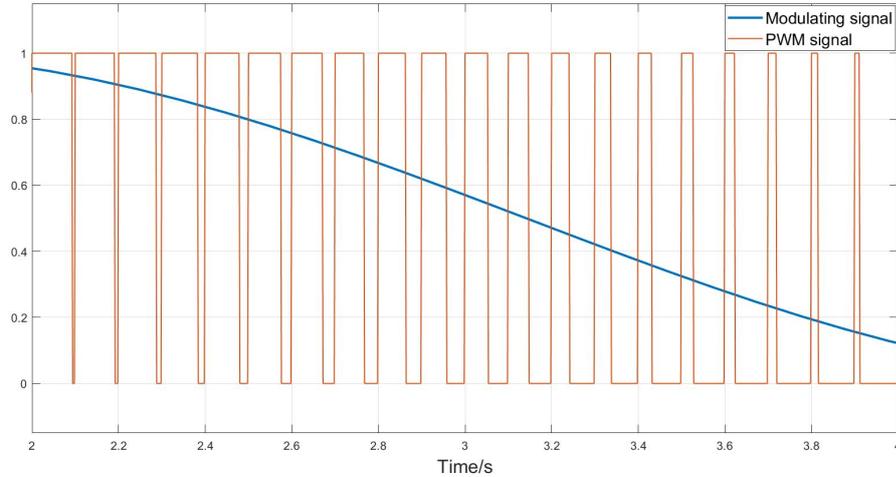


Figure 2.13: PWM modulated by a sinusoidal signal

The generated PWM has an high value which corresponds to the 14 V positive voltage of the power supply, whereas the low value corresponds to the ground reference voltage of 0 V. The signal can adapt its dynamics to the maximum acceptable command by the pump. Moreover, the generated PWM has a duty cycle accuracy of 0.1 %. Indeed, within Eagle environment, the maximum settable duty cycle value is equal to 1000.

In order to correctly generate the PWM signal, a modification is required. Since according to [33] the generated PWM is high side, a pull-down resistor of 1 k Ω is added at the output port of the control unit in order to obtain a clean PWM.

2.3.3 Filtered voltage divider

An analog low pass filter is designed in order to acquire the average value of the PWM command voltage. By extrapolating the average value of the PWM in time, it's possible to reconstruct the modulating signal which regulates the duty cycle. Moreover, as already explained in paragraph 2.2.2, there is a limit to the acquirable voltage by the USB-6009, hence making it necessary to implement also a voltage divider. The voltage divider is integrated with the filter in order to have a unique circuit (figure 2.14), capable of carrying out both the functions.

In order to properly design the circuit, some information has to be kept into account:

- The carrier frequency value of the generated PWM: 500 Hz. This is the frequency to be filtered out.

- The maximum bandwidth of the modulating signal which is going to regulate the duty cycle: apart from step signals, no more than 1 Hz for periodical waves. 1 Hz is chosen as limit frequency since, as shown in figure 2.19, a rough estimation for the actuator bandwidth can be 2 Hz. The modulating signal contains the information to be acquired, thus the filter cutoff frequency needs to be at least twice this value not to attenuate it.
- The input impedance of the USB-6009: 144 k Ω , which has to be at least 10 times higher than the impedance where voltage is measured.

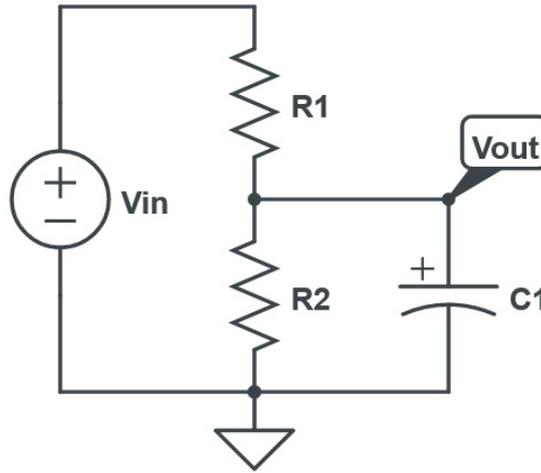


Figure 2.14: Filtered voltage divider

Let's analyze the frequency behaviour of circuit in figure 2.14. Since this circuit conciliates both the behaviour of a voltage divider and of a low pass filter, for convenience let's call it a filtered voltage divider. The transfer function in the Laplace domain results to be:

$$V_{out} = V_{in} \frac{\frac{1}{sC_1} || R_2}{R_1 + \frac{1}{sC_1} || R_2}$$

from which, by expanding all the calculations and by collecting the coefficient of the s-term, the final formula is obtained:

$$V_{out} = V_{in} \frac{R_2}{R_1 R_2 C_1} \cdot \frac{1}{s + \frac{R_1 + R_2}{R_1 R_2 C_1}} \quad (2.8)$$

Finally, from the above equation is immediate to extract the cut-off frequency of this circuit:

$$f_{cut} = \frac{R_1 + R_2}{2\pi R_1 R_2 C_1} \quad (2.9)$$

Both the capacitor and the resistor values are chosen by trial and error in order to obtain the best performances, ending up with $R_1 = R_2 = 10 \text{ k}\Omega$ and $C_1 = 2.2 \text{ }\mu\text{F}$, which yield $f_{cut} = 14.47 \text{ Hz}$. With these values, all the previously cited conditions are satisfied and the circuit is able to successfully isolate the modulating signal from the PWM.

Moreover, for low frequency values, being the capacitor impedance very high, C_1 acts as an open circuit. As a consequence, under these conditions, formula 2.8 simplifies into 2.3, going back to a voltage divider configuration.

2.3.4 Anti-aliasing filter

Aliasing is a dangerous phenomenon occurring in signal processing which causes signals at different frequencies to become aliases one of the other when sampled, hence making them indistinguishable and almost impossible to be correctly reconstructed [34]. This effect is caused by an undersampling condition during the acquisition phase, which generates an overlapping of the tails of the signal bandwidth. Defining the bandwidth of a signal as f_b , in order to properly acquire a signal, the Nyquist-Shannon sampling theorem must be satisfied, that is $f_s \geq 2 \cdot f_b$. Since in real systems signal bandwidth are not limited, during the analog to digital conversion, a low-pass anti-aliasing filter may be necessary to confine bandwidth in order to avoid aliasing effect. In most cases, the LPF is at least a second-order Butterworth filter [35].

According to [27], the employed acquisition board USB-6009 is not provided with an integrated anti-aliasing filter. Anyway it turned out to be unnecessary thanks to some precautions. First of all, during the acquisition phase a big enough sampling frequency $f_s = 5 \text{ kHz} \gg f_b$ is chosen.

However, the obtained data from the dynamic characterization are subsequently needed for the system identification phase, hence implying the sampling frequency can't be too high. A big f_s would load too much the identification algorithm and would generate a model which is strictly bound to the obtained data set, thus making it inadequate for any other set of data.

On the other hand, this oversampling allows to stay far enough from the system bandwidth, avoiding tail overlapping. Moreover, the filtered voltage divider utilized to filter out the carrier component acts as a very moderate anti-aliasing filter.

2.3.5 Wiring connections

In the following figure is represented the general setup of the test bench for the dynamic characterization:

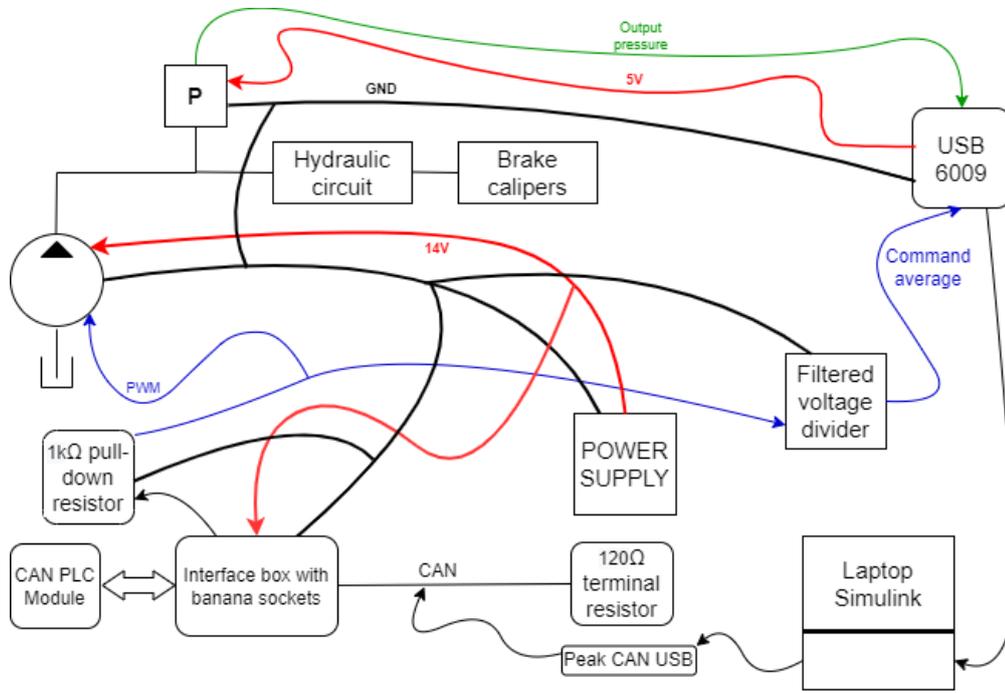


Figure 2.15: Dynamic characterization test bench setup

2.3.6 Simulink acquisition diagram

The following two figures represent the designed Simulink scheme for the acquisition phase in dynamic conditions. The first diagram in figure 2.16 represents the transmission data flow where the duty cycle is imposed. The second scheme in figure 2.17 instead represents the acquisition data flow where both the output pressure and the input average command voltage are acquired. Let's first analyze the blocks of diagram 2.16:

- Duty cycle modulating signal: it is in charge of generating the signal which modulates the duty cycle of the PWM. Within the subsystem, different types of signal are connected to a multiplexer where the output is chosen by a selector.
- Data type conversion: adapts the generated signal to a format which is acceptable by CAN protocol. The maximum transmitted duty cycle value is

1000, hence 10 bits are enough to encode all the possible values. Only positive integer values from 0 to 1000 are sent, this is why an uint conversion is chosen.

- Message packaging block: the message is packed according to the rules of figure 2.11. A 11-bits standard identifier type is chosen. Concerning the CAN identifier value, it must be coherent with the one inserted within CAN DB environment of MRS Developer Studio software. The hexadecimal number 100 is chosen, thus in this block is inserted the equivalent decimal value which is 256. Being the number of transmitted bits equal to 10, 2 bytes are enough to contain all the information. Finally a little-endian setup is chosen for byte order transmission.
- CAN configuration block: a bus speed of 500 kbit s^{-1} is set.
- CAN transmit block: message is transmitted periodically every 10 ms.

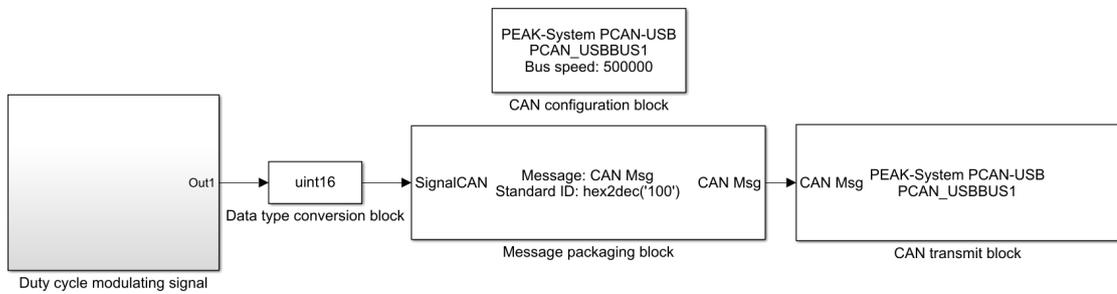


Figure 2.16: Simulink transmission diagram in dynamic conditions

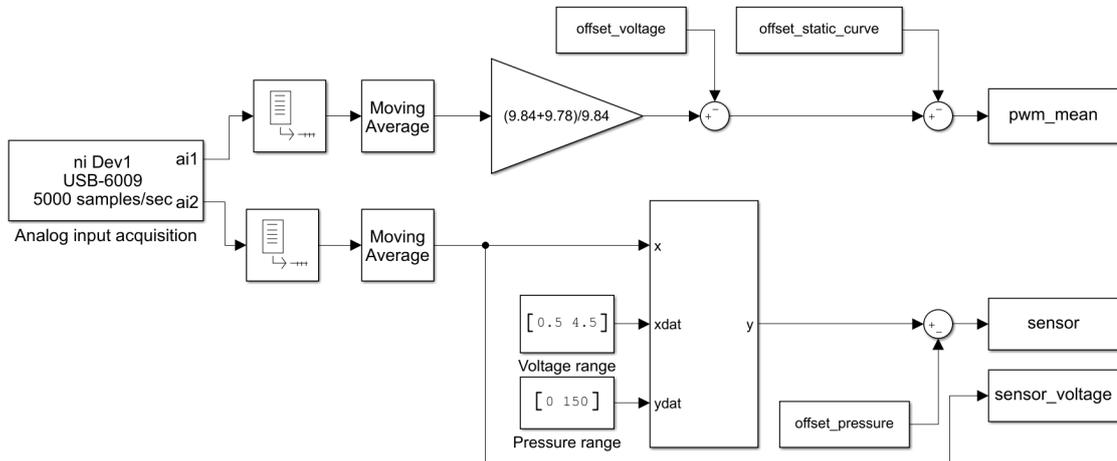


Figure 2.17: Simulink acquisition diagram in dynamic conditions

Concerning instead figure 2.17, few modifications are carried out from 2.8:

- The moving average block is inserted in order to smooth out the acquired signals. The average is performed over 50 values and this parameter is computed from certain variables. The sampling frequency of the DAQ toolbox is $f_s = 5$ kHz with a block size of 50, whereas the PWM duty cycle value is updated by the control unit every 0.01 s, $f_u = 100$ Hz. It is then possible to compute after how many samples the duty cycle is updated: $samples = f_s/f_u = 50$.
- The gain block is in charge of performing the reverse voltage divider operation by using the real values of the new 10 k Ω resistors.
- Offset static curve is the maximum voltage which doesn't make the pump switching on. Since the collected data has to be utilized to identify a suitable model of the actuator, this value is removed to eliminate the characteristic offset in order to simplify the identification. In this way an output pressure $p_{out} = 0$ bar corresponds to $v_{in} = 0$ V.

Moreover, the sampling frequency $f_s = 5$ kHz is updated. All the acquired signals are further smoothed out through another steep low pass filter. The related code is reported in appendix A.4.

2.3.7 Eagle diagram

Eagle is a programming interface developed by CadSoft, which was acquired by Autodesk. Together with the MCU flasher, the Eagle graphical user interface is included within MRS Developer Studio. The latter is a software which was developed by MRS company for the programming of their CAN products. Programming can be performed both graphically or in C code through the user interface. The following figure represents the design of the Eagle diagram for the generation of the PWM.

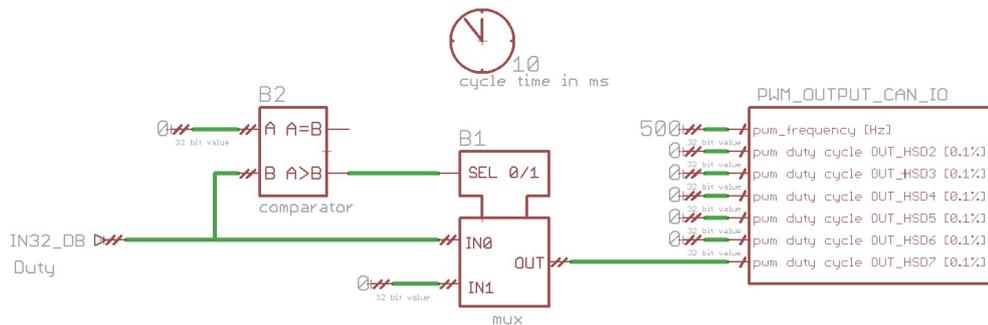


Figure 2.18: Eagle diagram for PWM generation

Duty cycle message is configured within CAN database environment following the choices previously decided in vehicle toolbox blocks. Comparator and multiplexer blocks are inserted in order to saturate the duty cycle in case it goes below 0. The maximum duty cycle value arriving through CAN from Simulink is equal to 1000, which corresponds to a PWM with 100 % duty cycle. Cycle time decides the time interval within which the graphical programming is executed and it is set to 10 ms. Finally, control unit pin A8 is chosen to generate the 500 Hz PWM.

2.3.8 Open-loop bandwidth

During a dynamic characterization, the signals chosen to stimulate the system have to be chosen accordingly to its bandwidth. By stimulating the system with a chirp signal, it's possible to obtain a rough estimate of the system bandwidth, by looking at the trend of the output peak to peak amplitude.

By convention, it's possible to define as bandwidth the frequency where the output amplitude is attenuated by -3 dB with respect to the passband value. In particular -3 dB corresponds to an amplitude which is ≈ 0.707 the passband amplitude. As showed in the following figure, by measuring the peak to peak amplitude it's possible to obtain a behaviour which is similar to the magnitude Bode plot of the actuator transfer function.

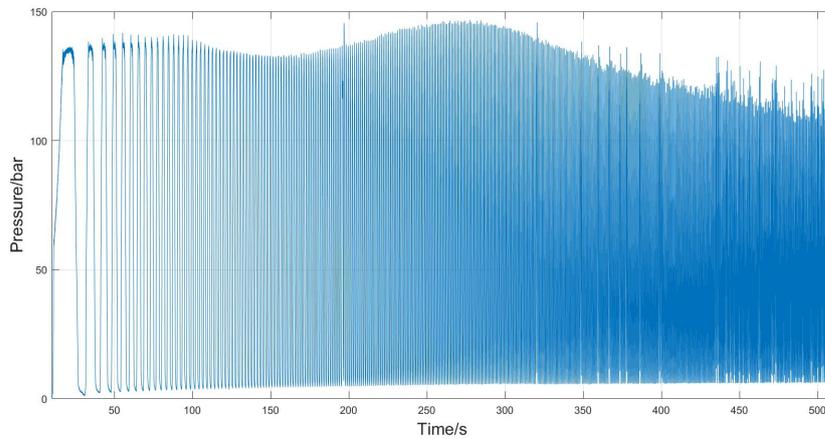


Figure 2.19: Open-loop bandwidth

The chirp signal used to obtain the figure above spans from 2 mHz up to 2 Hz. By comparing the peak to peak value with the initial low frequency amplitude, it's possible to obtain a rough bandwidth frequency $f_{bw} \approx 2\text{ Hz}$. With this value, it's now possible to choose the types of signal which can stimulate the actuator. Another thing which can be noticed from the figure is that the maximum measured

pressure is greater than 140 bar. This value is way higher than the maximum pressure measured in static conditions as depicted in figure 2.9.

2.3.9 Quasi-static characterization

A quasi-static characterization is a particular way of stimulating the system where the input variations happen very slowly such that the equilibrium holds at all times [36]. To actuate this very slow solicitation, a triangular wave of period $T_s = 1,000$ s is imposed as PWM modulating command voltage. The up and down sweep are carried out in different moments. In order to test different operating situations, the actuator is stressed out, by collecting the data both in cold and hot temperature conditions. The results are represented in the following figure.

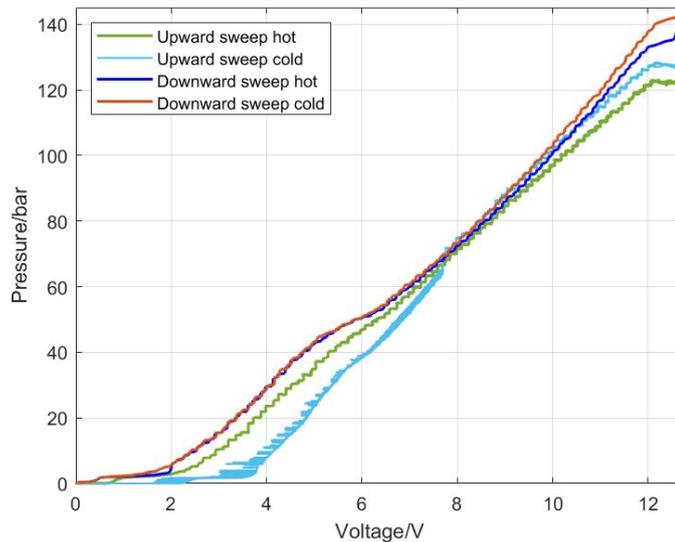


Figure 2.20: Quasi-static characterization

It can be noticed that each one of the four characteristics follows a different path, meaning that the pump behaviour is sensitive to its temperature variations, thus establishing a nonlinearity. Even the offset of the characteristic results to be affected by temperature alterations.

Switching from hot to cold doesn't considerably distort the curve for what concerns the downward sweep, whereas for the upward sweep there is a significant difference. Instead, in the region between 50 bar and 100 bar all the curves are almost coincident and linear and there is no substantial difference among them.

Moreover as previously noticed in the open-loop bandwidth analysis, the maximum pressure value is high for each of the four characteristics, compared to the static

one. This may be due to the fact that in dynamic conditions friction effects are easily overcome, thus resulting in an increase of the output pressure.

In conclusion, despite this sensibility to temperature variations, both the hot and cold states of this characterization are limit conditions. Cold values are maintained only for brief moments, since after few braking cycles the actuator tends to inevitably warm up. Instead, hot values are more likely to be achieved, but anyway they require very long working session and continuous operativity of the actuator. Moreover, it's preferable to avoid working under hot conditions, not to impair system parameters such as oil viscosity. Indeed, in a real scenario, the majority of the time the actuator works under warm conditions, whose characteristic can be considered an approximation of this four curves.

2.3.10 Non-linear behaviour

Along with temperature sensitivity, in order to evidence other nonlinear behaviours of the actuator, a particular dynamic characterization is performed. For the purpose, a full dynamics triangular wave at frequency 0.1 Hz is chosen. The obtained result is reported in figure 2.21.

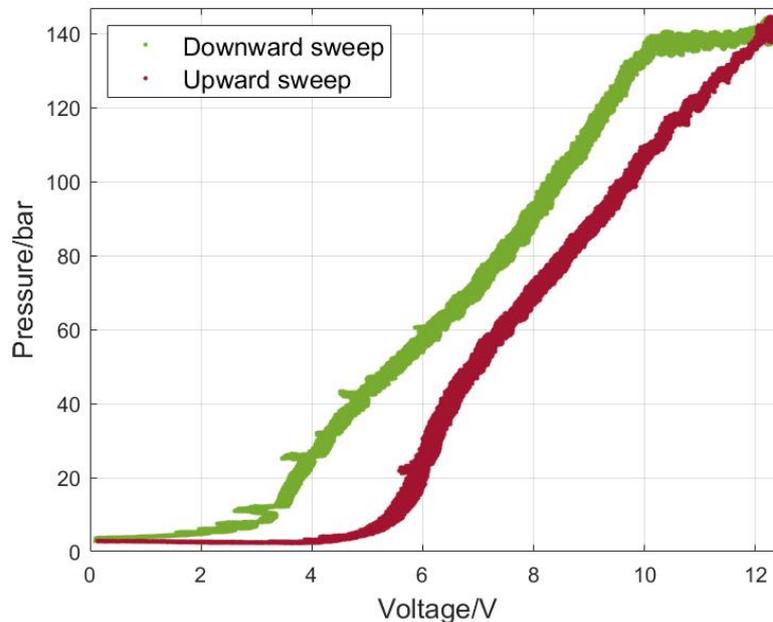


Figure 2.21: Hysteresis behaviour

By looking at the figure, a couple of nonlinearities are noticeable. The different paths followed by the upward and downward sweeps issues an hysteresis phenomenon.

This causes the system to behave differently, depending on how we approach a certain working point. The other observable nonlinear effect is the saturation at the extremities of the characteristic. Since it's lower than the pressure sensor saturation, the upward cap is due to the maximum pressure which can be provided by the pump in dynamic conditions. The bottom limit instead, as already stated in previous sections, is paired with the characteristic offset, due to the non-zero minimum command voltage required by the actuator to switch on.

2.3.11 Acquisition results

Different types of signal are chosen in order to modulate the duty cycle of the PWM and to stimulate the actuator under different operating conditions. In particular the chosen signals are:

- Step signals at different amplitudes.
- Chirp signals at different amplitudes.
- Amplitude modulated triangular waves.
- Random signals with uniform distribution.
- Amplitude modulated sine waves at different frequencies.

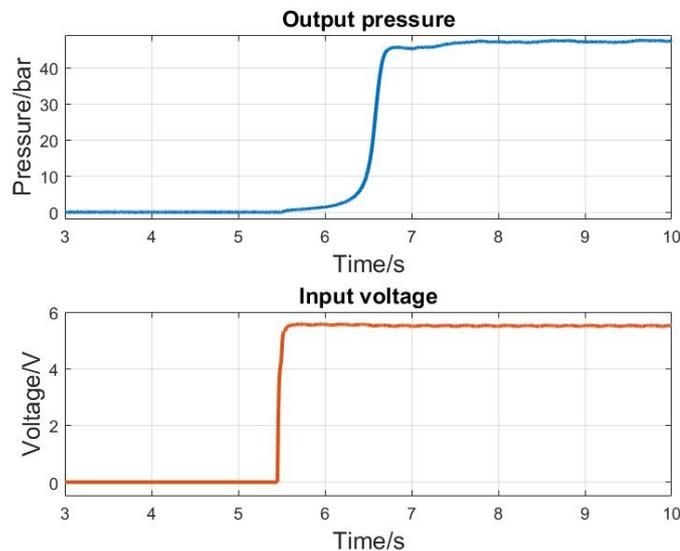


Figure 2.22: 45% duty cycle step signal

In figures 2.22, 2.23 and 2.24 are represented some of the cited signals together with their output pressure response.

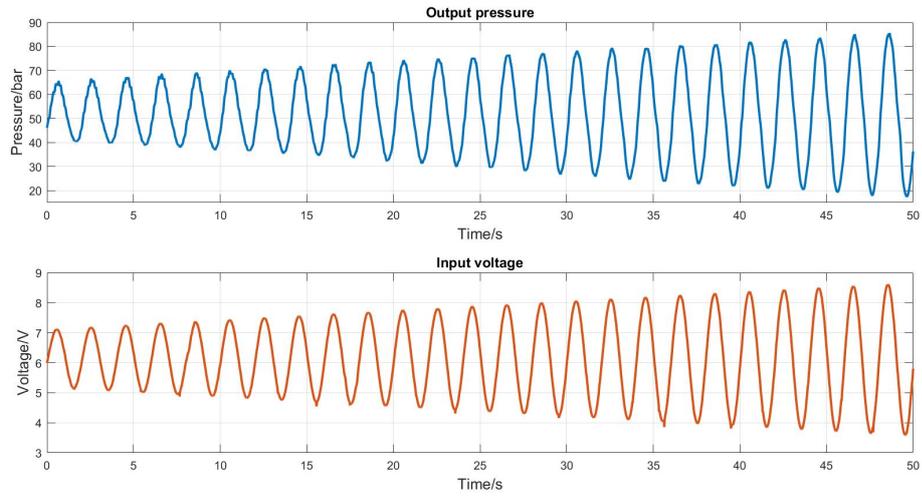


Figure 2.23: Amplitude modulated sinusoidal wave at 0.5 Hz with duty cycle from 45 – 55% to 30 – 70%

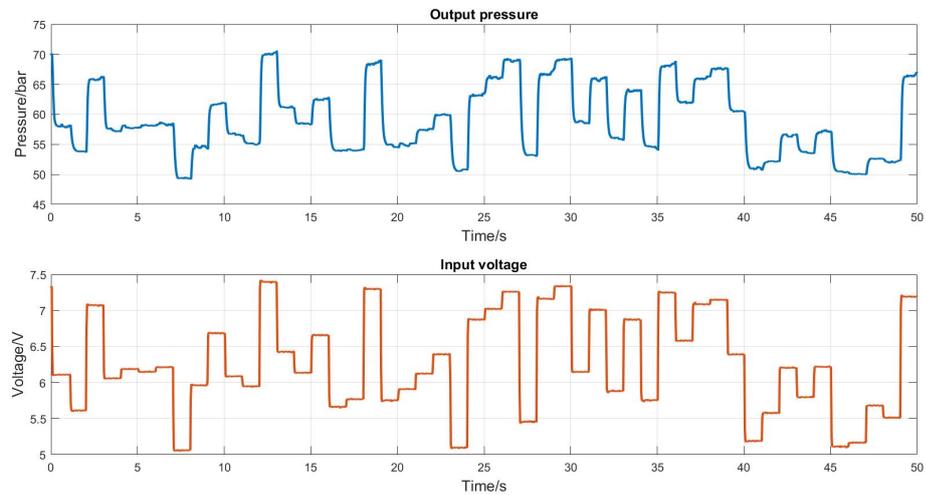


Figure 2.24: Output pressure and input voltage: Uniformly distributed random signal at 1 Hz with duty cycle within 40 – 60%

Chapter 3

System identification

System identification is a methodology which aims at building a mathematical model of a dynamic system, starting from the experimental measurements of its input and output signals [37].

A black box modeling approach is chosen, which means it doesn't exploit any knowledge about the system physical equations. Despite not being a scalable type of model, it has the big advantage that doesn't require any a priori system information to be built. The black-box technique aims at fitting the data regardless of a particular mathematical model structure.

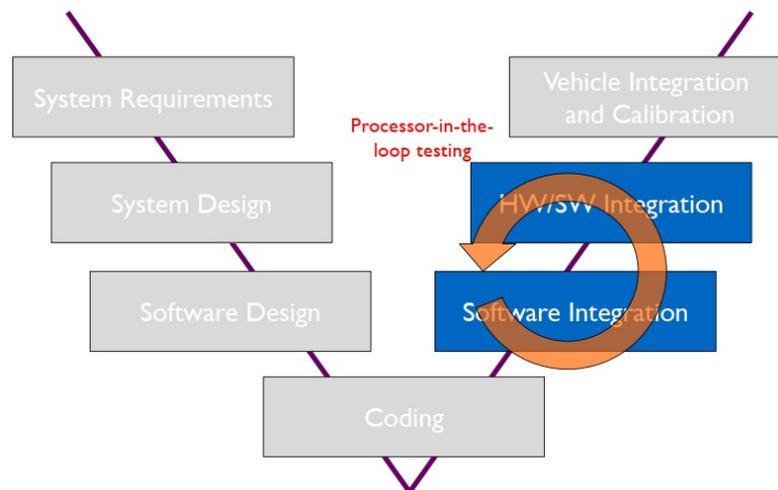


Figure 3.1: V-model development flow, PIL phase

Black-box modeling is usually performed by a trial and error procedure, where the parameters of the several structures are estimated and the results are compared. The standard approach is to start with simple models and then advance to more

complex ones. In this chapter, both linear and non-linear model structures are tested to verify their capability of fitting the system behaviour.

But why looking for a model? In this thesis, the main purpose of finding a model is to perform PIL. As shown in figure 3.1 [23], processor in the loop is one of the testing phases of the V-model development process employed in automotive and aeronautical applications. Specifically, it is a method which aims at validating a control algorithm, downloaded on the target hardware, over a model which emulates the system behaviour. This, not only allows to obtain an initial guess of the control parameters without the risk of damaging the actuator, but also speeds up the algorithm validation phase. Anyway, a final adjustment of the parameters will be required when moving to the real plant.

3.1 Choice of the datasets

All the types of dataset collected during the dynamic characterization phase are employed. 75% of the data is selected for estimation purposes, whereas the remaining part is chosen in order to validate the models.

Due to the high amount of samples, in order to reduce the effort of the system identification, over all the datasets a resampling operation of 100 Hz is performed. Moreover, further datasets are constructed by merging together the initial ones. Merging procedure allows the identified system to account for different dynamics at once, as if it were a single experiment. This is useful, especially in those situations where the system to be identified is nonlinear and it behaves differently depending on the operating regions.

The identified models are obtained both through the GUI of the system identification toolbox and through the code reported in appendix A.5.

3.2 Linear models

All the following models are designed for simulation purposes. This means that, during the estimation of the model parameters, the minimization process tries to reduce the simulation error between measured and simulated output. As a result, the estimation focuses on making a good fit for simulation of the model response with the current inputs [38].

The linear tested models are SISO polynomial discrete time models like ARX and OE. The quality of the models is evaluated by means of the Best Fit criterion [39], which is defined as:

$$BestFit = 1 - \sqrt{\frac{MSE}{\frac{1}{N - N_0} \cdot \sum_{t=N_0+1}^N (y(t) - \bar{y})^2}} \quad (3.1)$$

where,

$$MSE = \frac{1}{N - N_0} \cdot \sum_{t=N_0+1}^N (y(t) - \hat{y}(t))^2 \quad (3.2)$$

where N_0 is a suitable amount of samples after which the transient is over, $y(t)$ is the measured output and $\hat{y}(t)$ is the simulated output. Finally \bar{y} is defined as the arithmetic average of the measured output values.

This criterion defines how close is the simulated model output to the measured one. The most this value approaches to 1, the higher is the quality of the obtained model. A threshold of at least 90% is chosen to consider the model acceptable for all the validation datasets.

3.2.1 ARX

In the following table are reported some results yield by ARX models with different complexity and estimated over different datasets.

Table 3.1: Best Fit percentages for ARX models (Appendix A.5)

Estimation dataset	Model structure	Best Fit on validation datasets				
		data4v	data6v	data2v	data3v	data1v
data1e	ARX 441	Neg.	61.04%	55.95%	73.72%	72.52%
data6e	ARX 441	6.7%	64.34%	85.46%	68.41%	82.75%
data7e	ARX 221	28.98%	44.73%	Neg.	46.67%	26.36%
data11e	ARX 991	94.74%	29.9%	Neg.	21.36%	Neg.
data1m	ARX 991	17.97%	67.74%	60.31%	68.41%	57.5%
data2m	ARX 991	5.61%	61.9%	83.85%	67.57%	85.77%

3.2.2 OE

In the following table are reported some results yield by OE models with different complexity and estimated over different datasets.

Table 3.2: Best Fit percentages for OE models (Appendix A.5)

Estimation dataset	Model structure	Best Fit on validation datasets				
		data4v	data6v	data2v	data3v	data1v
data1e	OE 551	Neg.	59.95%	50.86%	72.65%	73.77%
data6e	OE 442	5.83%	63.66%	85.6%	68.38%	83.18%
data7e	OE 221	26.75%	43.93%	Neg.%	46.63%	44.62%
data11e	OE 553	95.33%	29.87%	Neg.	21.35%	Neg.
data1m	OE 332	Neg.	38.52%	69.25%	81.5%	75.17%
data2m	OE 661	Neg.	59.02%	82.42%	72.22%	88.15%

Neither the ARX nor the OE estimated models are capable of adequately fitting all the validation datasets.

3.3 Nonlinear models

Being linear models insufficient to describe the system behaviour, non linear ones like NLARX are tested. In table 3.3 are reported some results yield by NLARX models with different complexity and estimated over different datasets.

Table 3.3: Best Fit percentages for NLARX models (Appendix A.5)

Estimation dataset	Model structure	Best Fit on validation datasets				
		data4v	data6v	data2v	data3v	data1v
data1e	NLARX 551	29.98%	58.58%	63.11%	86.62%	80.51%
data6e	NLARX 551	22.99%	65.56%	78.54%	73.18%	83.86%
data7e	NLARX 331	25.35%	43.29%	Neg.	61.81%	Neg.%
data11e	NLARX 551	96.77%	Neg.	Neg.	3.75%	8.62%
data1m	NLARX 441	45.69%	62.98%	70.02%	89.49%	83.58%
data2m	NLARX 551	27.83%	52.19%	Neg.	78.68%	45.02%

The non-linearity estimators employed are wavelet networks, sigmoid networks

or linear functions, whereas the used regressor is a standard one.

3.4 Observations

Compared to linear models, NLARX models provide better performances, especially when validated on the same type of dataset used for estimation. Despite this, their Best Fit percentage does not reach the requirement on the threshold value previously established.

In conclusion, by observing all the obtained results, whenever a model properly fits a set of data, it behaves poorly for all the other ones. Neither the merging operation is able to improve the fit percentages up to the threshold value for all the validation datasets. Therefore, it's possible to conclude that none of the obtained model is adequate enough to describe the system behaviour in its entirety.

Usually, this type of non linear structures are able to grasp a wide range of models. Nevertheless, the difficulty in identifying the system may be due to some dynamics or external input which can't be captured by any of the model structures employed. Anyway, this does not impair the design of the control system. Observing the absence of the model, the tuning of the controller is directly carried out on the actuator. As a consequence, the time to perform this operation is dilated due to the slow interactions with the real system. A close attention must be paid during this phase to avoid damaging the pump.

Chapter 4

Control algorithm design

Until now the actuator has worked in open-loop, thus being subjected to any kind of uncertainty and external disturbance and not being able to properly satisfy the required braking performances.

Implementing a closed-loop feedback control system allows to solve the problem, both increasing the robustness and reducing the error between the measured output pressure and the requested one.

Generally, the Dexter DX Series brake actuator [24] should achieve performances when used along with the Dexter Predator DX2 inertia-activated electric brake controller.

In this chapter, taking into consideration the system non-linearity, a suitable custom made control algorithm is implemented.

The designed controller is a SISO system, where the input is the tracking error between the target pressure imposed by the autonomous driving unit and the actuator measured pressure. The produced output is a duty cycle value which modulates the PWM command signal.

4.1 Controller choice

4.1.1 PID control

Proportional-integral-derivative are a typology of linear controllers widely used in industrial applications due to their effectiveness. The design is based on an accurate tuning of three parameters. Starting from the standard mathematical expression in time domain of a parallel PID structure:

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

it's possible to derive the corresponding Laplace transform:

$$C(s) = K_p + \frac{K_i}{s} + K_d s$$

Since the control algorithm is going to be implemented on a PLC, the equivalent discrete transfer function is derived. By employing a Forward Euler discretization method and by modifying the derivative term into a LPF [40], the final result is achieved:

$$\frac{U(z)}{E(z)} = K_p + \frac{K_i T_s}{z - 1} + \frac{K_d N(z - 1)}{z - 1 + N T_s} \quad (4.1)$$

A PID controller is chosen for this application, thanks to its simplicity and to the ease of implementing it in the control unit software. Despite this, due to its nature, a single PID controller is not able to control the whole non-linear plant on its own.

4.1.2 Gain Scheduling

A gain scheduling is used to overcome the linearity of the PID controller, which alone is insufficient for controlling the whole operating region.

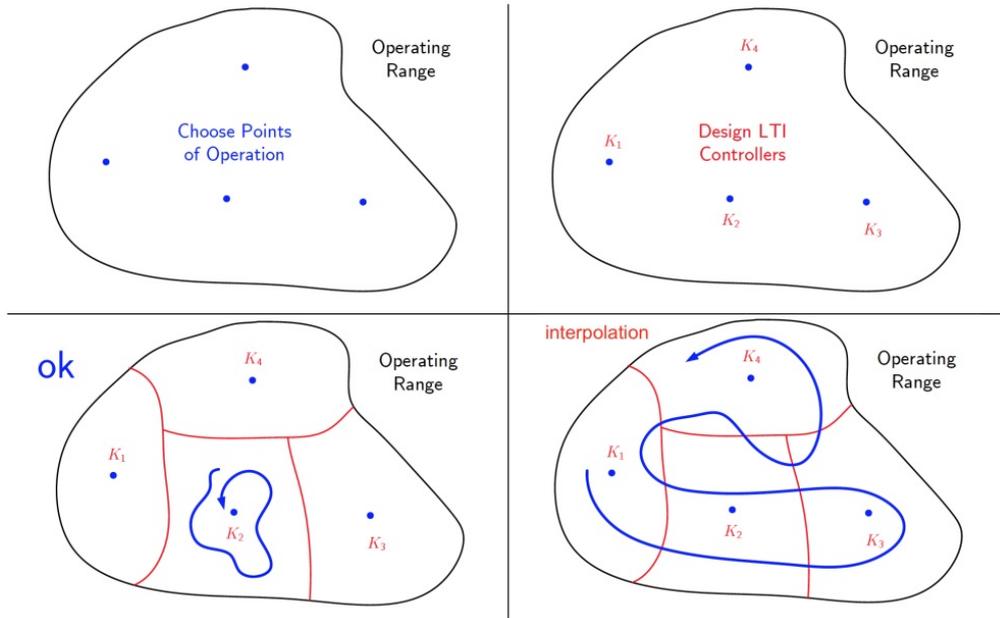


Figure 4.1: Gain scheduling operating range

Gain scheduling is a technique which was originally introduced for aerospace applications. The strength of this technique is the possibility to use different local

linear controllers to control a non-linear plant.

According to GS approach [41], several local linear models are obtained by linearizing the system around some equilibrium points. The chosen points are required to cover the majority of the system dynamics.

Afterwards, for each model, a linear feedback controller is implemented.

Finally, suitable interpolation functions are designed to obtain a unique global controller, capable of working on the entire region.

Unfolding the math behind this technique, the following generic system is considered:

$$\begin{aligned}\dot{x} &= f(x, u, w) \\ y &= h(x)\end{aligned}\tag{4.2}$$

where f and h are differentiable functions on suitable domains D_f and D_h , whereas $x \in \mathbb{R}^{n_x}$ is the state vector, $u \in \mathbb{R}^{n_u}$ is the command input and $y \in \mathbb{R}^{n_y}$ is the output. In this application, being the actuator a SISO system, u and y are scalar. In order to determine the operating points, a measured signal w , called scheduling variable, is defined, and a set of operating points is chosen:

$$\{\bar{w}_1, \dots, \bar{w}_N\} \subset D_w \subseteq \mathbb{R}^{n_w}$$

There are several possible choices for the scheduling variable and in this case the chosen variable is the setpoint of the feedback control system. Since the output follows the reference value, the setpoint is indirectly linked to the measured output. For each $\bar{w}_i \in D_w$, a pair (\bar{x}_i, \bar{u}_i) , satisfying the equilibrium point definition, has to be found:

$$f(\bar{x}_i, \bar{u}_i, \bar{w}_i) = 0, \quad i = 1, \dots, N.$$

From the nonlinear system 4.2, a new set of systems, linearized around each equilibrium point $(\bar{x}_i, \bar{u}_i, \bar{w}_i)$, can be written by defining the following relative variables:

$$\begin{aligned}\tilde{x}_i &\doteq x - \bar{x}_i \\ \tilde{u}_i &\doteq u - \bar{u}_i \\ \tilde{w}_i &\doteq w - \bar{w}_i \\ \tilde{y}_i &\doteq y - h(\bar{x}_i)\end{aligned}$$

Thus, the state-space representation of each local linear system is:

$$\begin{aligned}\dot{\tilde{x}}_i &= A(\bar{w}_i)\tilde{x}_i + B(\bar{w}_i)\tilde{u}_i + E(\bar{w}_i)\tilde{w}_i^e \\ \tilde{y}_i &= C(\bar{w}_i)\tilde{x}_i\end{aligned}\tag{4.3}$$

where superscript "e" identifies the components of w that are external input. The above matrices are defined as:

$$\begin{aligned}
 A(\bar{w}_i) &\doteq \left. \frac{\partial f}{\partial x} \right|_{(\bar{x}_i, \bar{u}_i, \bar{w}_i)}, & B(\bar{w}_i) &\doteq \left. \frac{\partial f}{\partial u} \right|_{(\bar{x}_i, \bar{u}_i, \bar{w}_i)} \\
 E(\bar{w}_i) &\doteq \left. \frac{\partial f}{\partial w^e} \right|_{(\bar{x}_i, \bar{u}_i, \bar{w}_i)}, & C(\bar{w}_i) &\doteq \left. \frac{\partial h}{\partial x} \right|_{(\bar{x}_i, \bar{u}_i, \bar{w}_i)}
 \end{aligned}$$

For each i -th LTI system 4.3, a linear local controller with fixed structure is designed. In this dissertation a PID technique is chosen.

Being $K_i(z)$ the discrete TF of the i -th controller, the following control law is defined:

$$\begin{aligned}
 \tilde{u}_i &= K_i(z)\tilde{e}_i = K_i(z)e \\
 \tilde{e}_i &\doteq \tilde{r}_i - \tilde{y}_i = r - h(\bar{x}_i) - y + h(\bar{x}_i) = r - y = e
 \end{aligned}$$

where r is the reference signal and e is the tracking error. Each K_i controller is characterized by different values of its parameters $p_i \in \mathbb{R}^{n_p}$, which depend on the operating point \bar{w}_i :

$$K_i(z) \equiv K(p_i, z)$$

As showed in figure 4.1, each $K_i(p_i, z)$ controller works properly in some neighborhood of the i -th operating points. In order to obtain a unique controller, suitable interpolation function must be defined. Thus the global control law is defined as:

$$\begin{aligned}
 u &= \mathcal{I}_u(w) + e \cdot K(\mathcal{I}_p(w), z), \\
 \mathcal{I}_p(\bar{w}_i) &= p_i, & i &= 1, \dots, N, \\
 \mathcal{I}_u(\bar{w}_i) &= \bar{u}_i, & i &= 1, \dots, N.
 \end{aligned} \tag{4.4}$$

In this dissertation simple lookup tables with linear interpolation are chosen as interpolation functions. Depending on the operating region of the scheduling variable, the LUT returns the corresponding values of the controller parameters. Theoretical analysis shows that the global controller 4.4 works correctly on the whole domain D_f if two conditions are met:

- Variations of w are sufficiently slower compared to x ones. Here, being w the setpoint, the maximum variation frequency falls within the closed-loop system bandwidth.
- The number of operating points, which are going to be chosen, is sufficiently large to explore the majority of D_f .

4.2 Control architecture

Before the final implementation on the PLC, the control structure is devised and verified on Simulink. By analyzing the schematic in 4.2, four different subsystems are noticeable:

- The acquisition block where the output pressure and the average command voltage are acquired. It is the same as figure 2.17 except for the removal of the static curve offset block.
- The transmission block which transmits the duty cycle command value on the CAN bus. It is the same as figure 2.16.
- The reference generator block which is in charge of issuing different types of reference signals.
- The controller block containing all the elements employed for different trials of the control structure. An insight of this block is provided in figure 4.3.

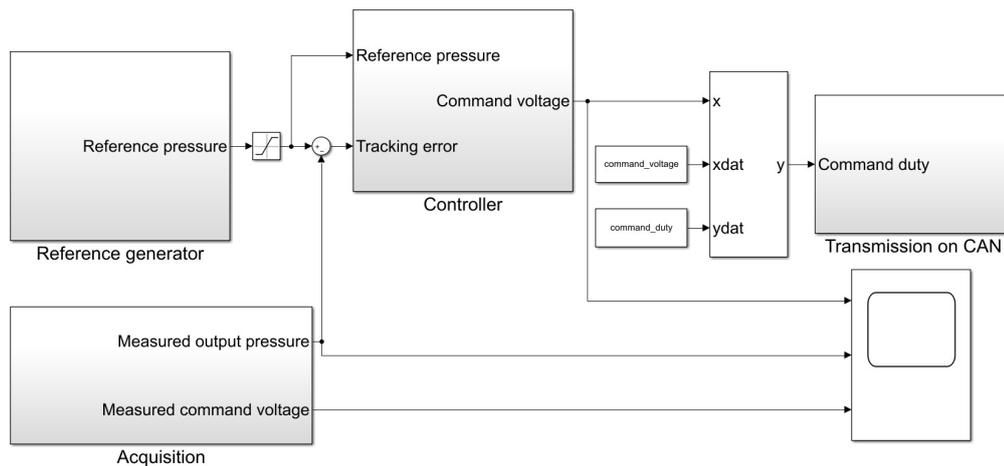


Figure 4.2: Simulink diagram for control algorithm design

Within the controller block, two configurations are tested to understand which one is able to yield the best results. In particular, transients performances are analyzed both in presence and absence of the feedforward term. Additionally, further tests are performed to understand which PID controller parameters are required to achieve good results.

The feedforward term is carried out through a look-up table which transforms the pressure setpoint value into a voltage one. The LUT implements a characteristic which is an interpolation of the four curves represented in figure 2.20.

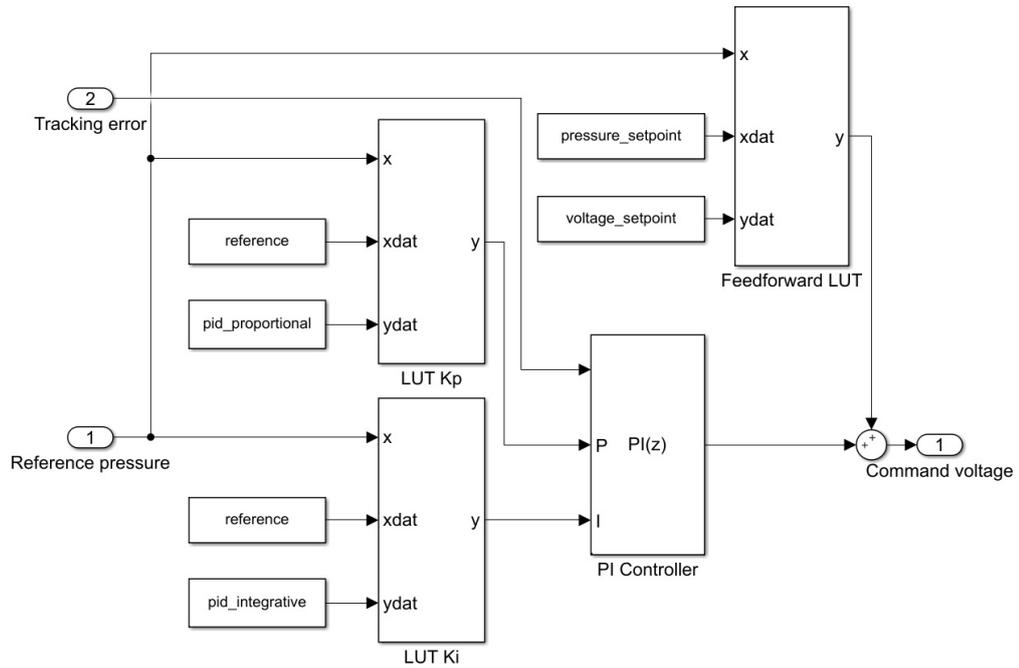


Figure 4.3: Simulink setup of the most general tested controller

Moreover, since the generated command input is a voltage, before sending the message on CAN, a further LUT is required in order to convert it into a duty cycle.

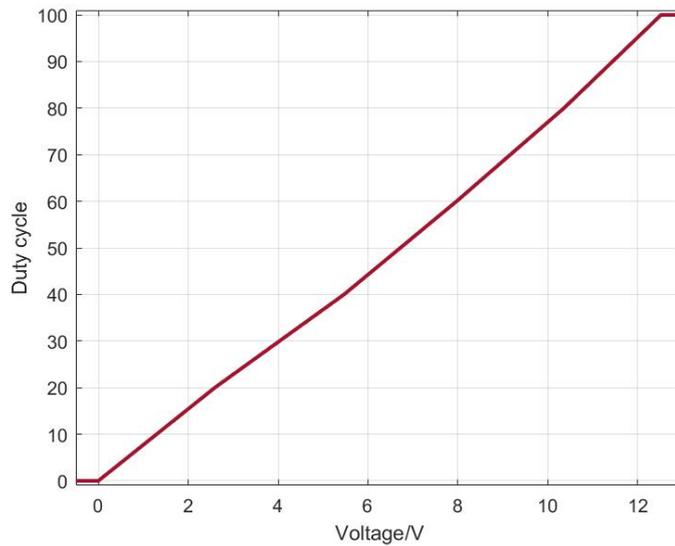


Figure 4.4: Voltage to duty LUT

Instead of using the formula $V_{mean} = (V_H - V_L) \cdot d$, some voltage-duty measurements are performed and all the points are linearly interpolated, resulting in the curve of figure 4.4.

Finally, after several tests, the control structure providing the best performances results to be the feedback Gain-Scheduled PI controller without the open-loop feedforward contribution. Established the structure, everything can be adapted to the control unit software, where eventually the parameters are tuned.

4.3 Eagle implementation

The Simulink elements, which need to be transferred to MRS and implemented in Eagle, are both the final structure of the controller and the acquisition blocks. Since the feedforward term is not present anymore, the voltage-duty LUT can be directly included into the controller.

4.3.1 Acquisition on PLC

Now that the acquisition task is devolved to the PLC, a new filtered voltage divider needs to be designed. According to [33], the internal resistance of the analog input pins $C3, \dots, C7$ is equal to 22.7 k Ω .

Together with the considerations performed in section 2.3.3, the new chosen values of the filtered voltage divider are:

$$R_1 = 5.078 \text{ k}\Omega \quad R_2 = 504 \Omega \quad C_1 = 47 \mu\text{F}$$

R_2 resistance is chosen this way to prevent overvoltage and to avoid measurement distortions due to the interaction with the PLC internal resistance. These values give rise to a new cut-off frequency:

$$f_{cut} = \frac{R_1 + R_2}{2\pi R_1 R_2 C_1} \approx 7.39 \text{ Hz}$$

In figure 4.5 is represented the designed Eagle acquisition diagram. Starting from the left, both the average command voltage and the output sensor voltage are acquired through the analog input channels. Input port blocks are followed by low pass filters averaging over 20 values. The value 20 is chosen by trial and error, considering the trade-off between the caused delay and the smoothness of the obtained signals.

All the voltages acquired by the analog input blocks are expressed in mV. Therefore, the new two points of the pressure sensor characteristic become $P_0 = (x_0, y_0) = (500, 0)$ and $P_1 = (x_1, y_1) = (4500, 150)$ which give rise to a new equation:

$$y = x \cdot \frac{150}{4000} - \frac{150}{8} = 0.0375 \cdot x - 18.75$$

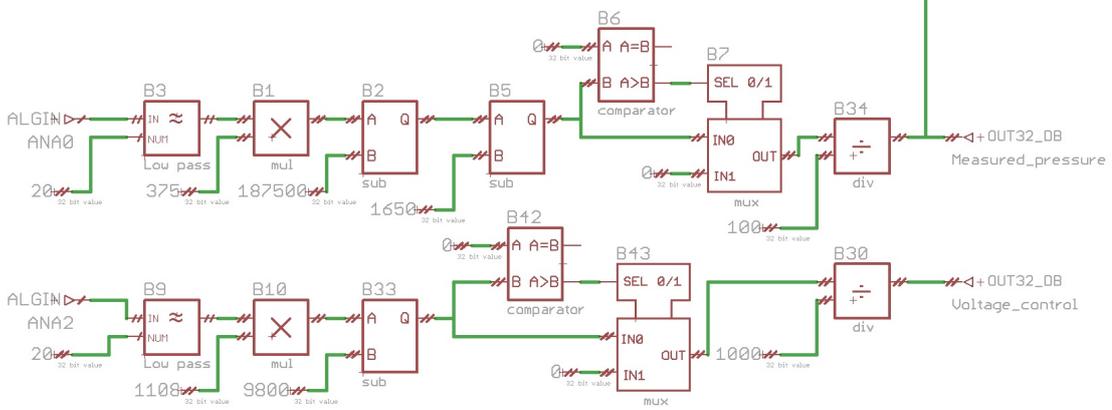


Figure 4.5: Acquisition diagram on Eagle

In order to avoid approximations within the programming interface, all the number are converted into integers. Being the signal dimensioned as 32 bit value, the software is able to handle positive integer numbers up to $2^{32} - 1$. Hence both side of the equation can be multiplied by 10000, giving rise to the final equation implemented in figure 4.5:

$$10000 \cdot y = 375 \cdot x - 187500$$

where y is the value of the measured pressure. Then, the offset pressure computed in table 2.1 is multiplied by 10000 and subtracted to y . Moreover, negative values are saturated to 0, by concatenating a comparator with a multiplexer. Saturation is required to prevent numerical error: a small negative number in Eagle becomes a big positive value when transmitted on CAN.

Finally, the pressure value is transmitted back to Simulink after being divided by 100, coherently with the accuracy decided in the CAN database reported in table 5.2.

Concerning the mean voltage acquisition, in order to deal with integer numbers, the new reverse voltage divider formula 2.5 becomes:

$$100 \cdot V_{in} = 100 \cdot V_{out} \frac{504 + 5078}{504} \approx 1108 \cdot V_{out}$$

This, together with the fact that voltages are expressed in mV, transforms the 0.098 V offset into 9800. Finally, after a saturation to 0, the voltage value is divided by 1000 to transmit it with the accuracy reported in table 5.2.

Cycle time is set to its minimum value equal to 10 ms, thus the acquisition sampling

frequency is equal to 100 Hz. Being the actuator bandwidth estimate ≈ 2 Hz, the Nyquist-Shannon theorem is satisfied.

Even if the acquisition task is now devolved to the PLC, due to the control unit inability of generating 5 V, the USB-6009 is still required to supply the pressure sensor.

4.3.2 PID implementation

The PI controller is implemented in Eagle by exploiting the elementary discrete integrator block z^{-1} . From control theory, it's possible to derive the closed loop transfer function of a feedback control system (figure 4.6):

$$\frac{Y(z)}{R(z)} = \frac{G(z)}{1 + G(z)H(z)} \quad (4.5)$$

where $Y(z), R(z)$ are the output and the reference z -transforms, whereas $G(z)$ is the feedforward transfer function and $H(z)$ is the feedback transfer function. The plus at the denominator comes from the negative sum at the feedback node.

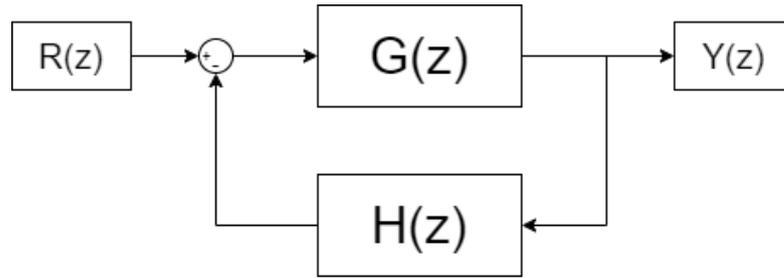


Figure 4.6: Feedback control system

Here, the transfer function to be carried out is $1/(z - 1)$, which has to be derived starting from the basic integrator block. Hence, a positive feedback with $G(z) = 1$ and $H(z) = z^{-1}$ is implemented:

$$\frac{Y(z)}{R(z)} = \frac{1}{1 - z^{-1}} = \frac{z}{z - 1}$$

which, divided by z , gives rise to the integrator term in forward Euler form, as showed in figure 4.7.

According to [42] and [43], a thumb rule for the choice of the controller sampling frequency is 10 times the value of the desired closed loop bandwidth. In this dissertation, the maximum frequency stimulating the system will be smaller than 2 Hz. Thus, choosing $f_s = 100\text{Hz}$ is a conservative choice which also matches Eagle cycle time. A division by 100 is performed to implement this value.

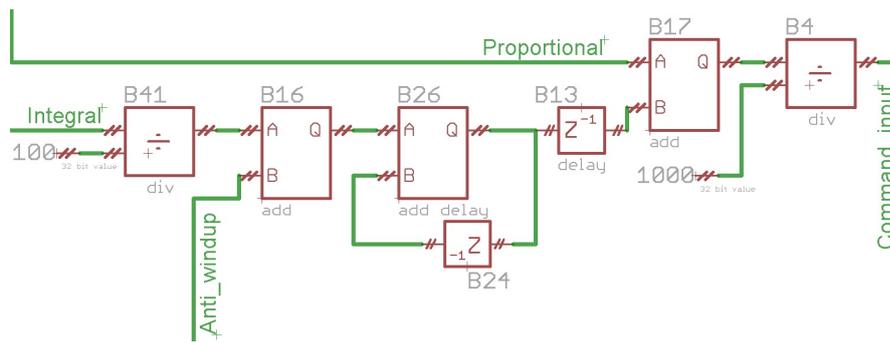


Figure 4.7: Eagle diagram of PID controller

Finally, since in the user code all PID parameters are multiplied by 1000, it's necessary to divide the final control input by the same number.

4.3.3 GS implementation

Gain Scheduling is mainly implemented within the user code file through C-programming language. The user code is executed synchronously with the code generated from the graphical interface.

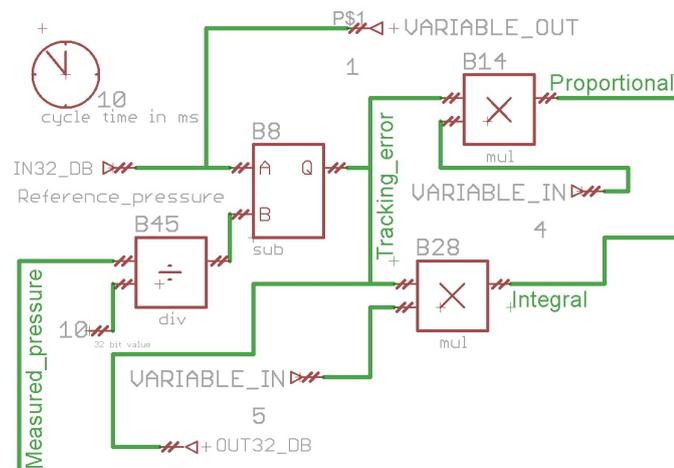


Figure 4.8: Eagle diagram interfacing with user code

The scheduling variable is extracted from Eagle environment and stored in the C-code, where it's employed to identify the working region. Suitable functions, implementing linear interpolation LUT, return the corresponding PID parameters to the graphical diagram, where they are multiplied by the tracking error.

```
1 | user_code.h
```

```

2 #ifndef _USER_CODE_H_
3 #define _USER_CODE_H_
4 //Select the hardware type
5 #define HW_TYPE HW_CAN_IO_V2
6 //Enable synchronous execution of graph code
7 #define GRAPH_ENABLE
8
9 user_code.c
10 void usercode(void)
11 {
12     //Scheduling variable: reference pressure,
13     //acquired from graphical interface.
14     int16_t reference = user_variable[1];
15
16     //Reference pressure with decimal precision,
17     //expressed as integer (multiplied by 10).
18     //Vector with all the chosen operating points.
19     int16_t arr_x[]={200,400,600,900,1100};
20
21     //PI controller for each operating point.
22     //Values multiplied by 1000 to deal with integer
23     //numbers.
24     int16_t arr_Kp[]={950,700,600,400,350};
25     int16_t arr_Ki[]={4300,4300,3800,3000,2500};
26
27     if(reference==0){
28         //Switch off the pump.
29         user_variable[4]=30000; //Kp proportional
30         user_variable[5]=30000; //Ki integral
31     }
32     else{
33         //Vector length.
34         uint8_t length_arr_x = sizeof(arr_x)/sizeof(
35         arr_x[0]);
36
37         //Implementing linear interpolation functions with
38         //LUT
39         //LUT_MODE_LIMIT: saturation outside of the end
40         //points.
41         //Returned Kp and Ki value: int16_t.

```

```

36     user_variable[4] = os_util_lookup1D(arr_x,
37     arr_Kp, length_arr_x, reference, LUT_MODE_LIMIT);
38     user_variable[5] = os_util_lookup1D(arr_x,
39     arr_Ki, length_arr_x, reference, LUT_MODE_LIMIT);
    }
}

```

Listing 4.1: User code implementing gain scheduling

4.3.4 Anti-windup technique

Windup is a phenomenon which can occur in PID controllers where the integral action is present. Whenever a big variation in the setpoint value arises, a big control action is required. However, due to the physical limits of the actuator, the command value is saturated, therefore the output takes more time to reach the setpoint.

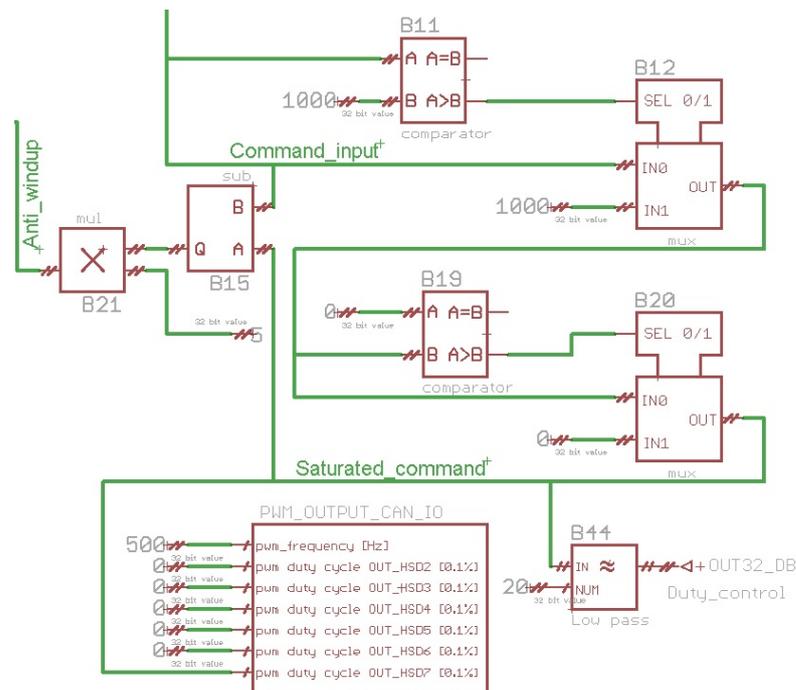


Figure 4.9: Eagle diagram for anti-windup and PWM saturation

When this happens the feedback loop is broken and the system runs as an open loop [35], due to the discrepancy between the controller and actuator outputs. Indeed the integrator, which is blind with respect to the actuator bounds, keeps accumulating in order to reduce the error. Even when the output approaches the

reference, the command input remains saturated, due to the large integral term. The direct consequences of this phenomenon are both big overshoots and long transient periods [44], before the integrator is unwound by errors in the opposite direction.

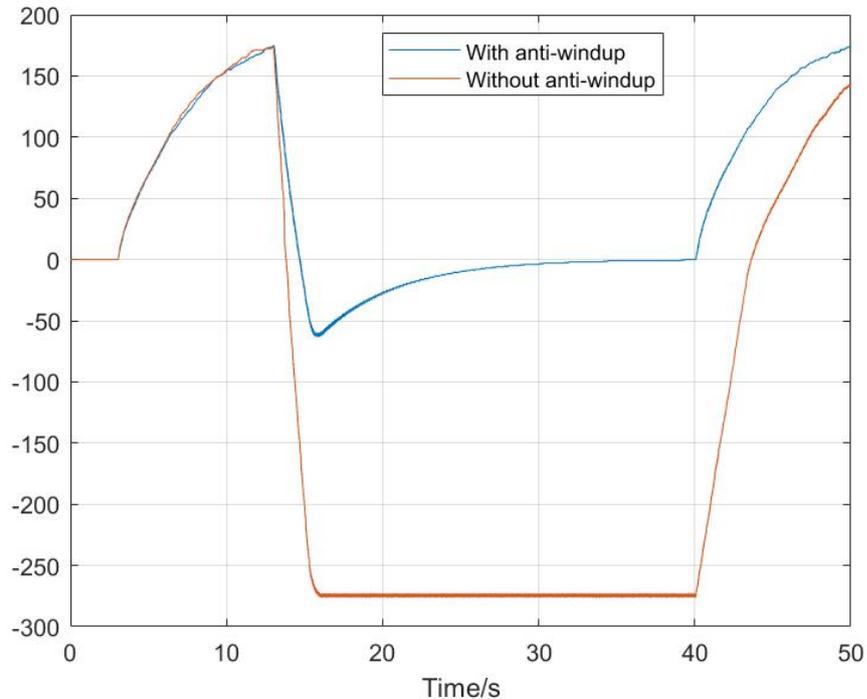


Figure 4.10: Anti-windup benefits on integrator term

Several techniques can be adopted to solve this problem, like gradually increasing the setpoint or preventing the integrator term to exceed some determined limits. In this dissertation a back-calculation anti-windup technique is employed. In [45], this method is named as tracking anti-windup and it's demonstrated its capability of significantly improving the step response.

Back-calculation method exploits a feedback loop to unwind the PID controller. In particular (figure 4.9), it measures the difference between the saturated and the measured command, multiplies it by a value $1/T_d$ and subtract the result to the integrator term. T_d is a suitable time constants which determines how fast the integrator is discharged. The smaller T_d , the faster the integral term is reset. Here, a value of T_d equal to 0.2 is chosen, indeed in figure 4.9 a multiplication by $1/T_d = 5$ is performed.

Figure 4.10 shows the practical effect of anti-windup algorithm on the integrator term whenever a negative step occurs. Without any contribution, the integral

term keeps charging to negative values and if a variation in the opposite direction appears, it needs to restore a big integral action before something happens. On the contrary, thanks to anti-windup, the integrator value is reset to 0, getting ready to future reference changes.

4.3.5 Tuning

In order to obtain some required transient performances, a fine tuning is carried out. For the purpose, ad hoc Simulink and Eagle diagrams are designed. In the literature, there are several techniques providing a good initial guess of PID parameters, which can be tweaked manually afterwards. Some of them are heuristic methods like Ziegler-Nichols and Coheen-Coon. Here, a simple trial and error procedure is followed, and for each local controller a couple (K_p, K_i) is found. The tuning is performed on the real plant, thus a careful attention is paid not to damage the actuator. In order to keep the system in a working region far from the upper saturation, a maximum pressure of 110 bar is chosen as maximum pressure reference, and therefore as scheduling variable. Finally, the obtained values, reported in table 4.1, are inserted within the C-code LUT.

Table 4.1: K_p and K_i values for each operating region (V1)

Parameter	Working point [bar]					
	0	20	40	60	90	110
K_p	30	0.95	0.7	0.6	0.4	0.35
K_i	30	4.3	4.3	3.8	3	2.5

Chapter 5

Control algorithm validation

Once the software development is completed, the algorithm has to be assessed. Validation phase aims at verifying whether the software meets the imposed requirements.

5.1 Acquisition interface

Control algorithm quality is verified through Simulink, where the reference pressure is generated and where all the relevant data are collected. Thus, two suitable set of signals has to be designed, one for the transmission frame and one for the receiving frame, showed in table 5.1.

Table 5.1: CAN frames for testbench results

Frame name	CAN ID [hex]	Extended	Send	Send max [ms]	Send min [ms]	DLC
Controller_to_Simulink	0x100	0	1	10	10	6
Simulink_to_Controller	0x200	0	0	0	0	2

In order to properly dimension the signals for the CAN communication, two parameters require to be evaluated: the desired resolution and the offset of the considered data range. With these two information, real values can be adapted to the CAN protocol. Finally, with the following formula, it's possible to obtain back the real values from CAN messages:

$$real_value = CAN_msg_value \cdot resolution + offset \quad (5.1)$$

Moreover, it's important to remind that CAN protocol is based on integer number transmission.

Table 5.2: CAN Data Points for testbench results

Variable name	CAN frame name	Bit start	Bit length	Data format
Reference_pressure	Simulink_to_Controller	0	11	0 (LE)
Duty_control	Controller_to_Simulink	0	10	0 (LE)
Voltage_control	Controller_to_Simulink	10	11	0 (LE)
Measured_pressure	Controller_to_Simulink	21	14	0 (LE)
Error	Controller_to_Simulink	35	12	0 (LE)

Considering the second row of 5.1, which is the frame transmitted by Simulink:

- Reference_pressure is the only message contained in this frame. Reference pressure values span from 0 up to 120 bar and a resolution of 100 mbar is desired. Hence, by reversing 5.1 formula, 1200 is the maximum value assumable by this message. Then, 11 bits are sufficient to include all the possible values

Switching to the first row of 5.1, there are all the signals received by Simulink:

- Duty_control spans from 0 to 100% and the desired resolution equals 0.1, which imply a total of 10 bits.
- Voltage_control spans from 0 to 12.5 V and the desired resolution equals 10 mV, which require a total of 11 bits.
- Measured_pressure spans from 0 to the maximum pressure readable by the sensor which equals 150 bar. A resolution of 10 mbar is required. 14 bits are chosen since they can cover values up to 16383.
- Error is the tracking error, which spans from negative values up to the maximum value of Duty_control. Hence, it's designed 1 bit longer than Duty_control, to account for the negative sign. Moreover, when received on Simulink, it's casted as signed integer.

Once each message is received on the respective environment, equation 5.1 is applied.

5.2 Reference response

Together with the company Bylogix, some target performances have been established. In particular, the required rise time for reaching the 90% of the reference pressure is $t_{r,target}^{(0-90)\%} = 0.5$ s. Moreover, overshoot $\hat{s}_{target} \leq 10\%$ and small settling time with few oscillations are preferred.

In the following figure is represented the output response of a 20 bar step reference.

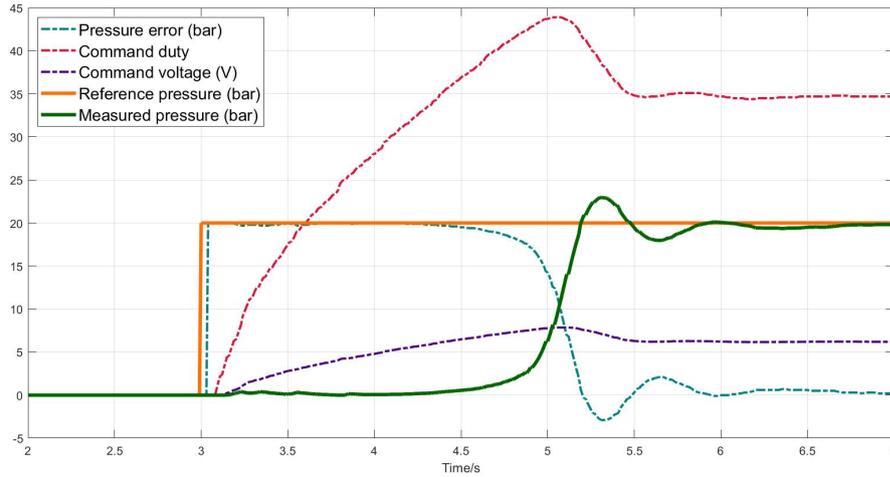


Figure 5.1: Output response to a 20 bar step reference

By measuring the transient parameters, it's possible to obtain an overshoot $\hat{s} = 14.85\%$ and a rise time $t_r^{(0-90)\%} = 2.170$ s. These attained values are really poor compared to the target ones.

5.2.1 Pre-braking phase

In order to improve the results previously achieved, a solution is required. Tuning the parameters to decrease the rise time is not effective, since it would result in a drastic worsening of the overshoot value. By looking at figure 5.1, it's possible to observe that, even if the command input starts increasing right after the step occurs, it happens really slowly.

Indeed, the huge reaction delay noticeable on the output response is eventually attributed to the possible presence of air in the test bench braking circuit.

To overcome this problem a solution is proposed. The latter involves the insertion of a pre-braking phase which aims at activating the pump, maintaining it in a "ready to react" state. The value chosen to switch on the pump is 2 bar, which is the smallest pressure the pump is able to hold at steady-state.

In order to have a fast response and to reach stable steady state performances, this value is inserted within the LUT and the PID is tuned accordingly. Thus resulting in the following updated table.

Table 5.3: K_p and K_i values for each operating region (V2)

Parameter	Working point [bar]						
	0	2	20	40	60	90	110
K_p	30	20	0.95	0.7	0.6	0.4	0.35
K_i	30	20	4.3	4.3	3.8	3	2.5

With these K_p and K_i values, the obtained overshoot for the 2 bar pre-step equals 9.5%, whereas the obtained rise time is around 1 s.

5.2.2 Step reference

In the following figures are reported some of the output step response for different reference values, with the insertion of the pre-braking stage. All these steps correspond to the working points previously chosen when designing the gain scheduled controller.

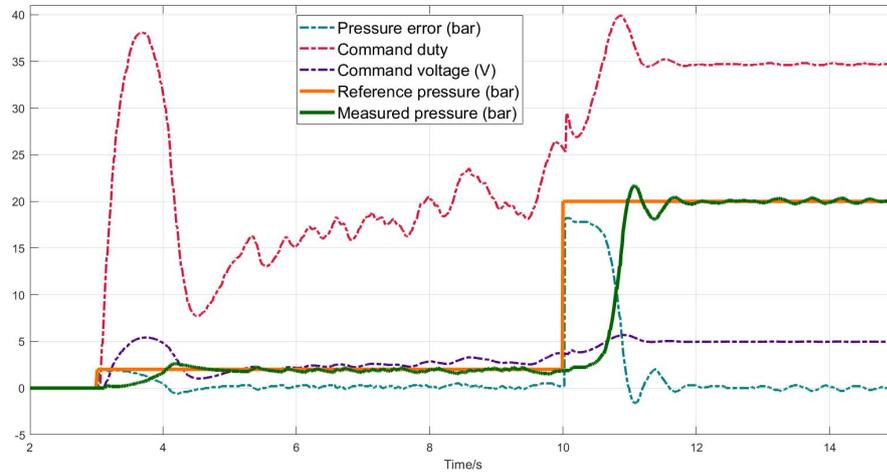


Figure 5.2: Output response to a 20 bar step reference, with pre-braking

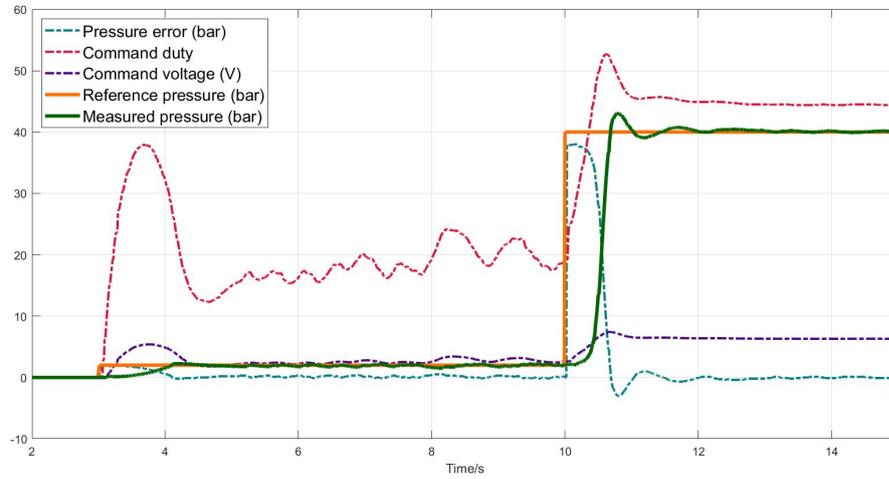


Figure 5.3: Output response to a 40 bar step reference, with pre-braking

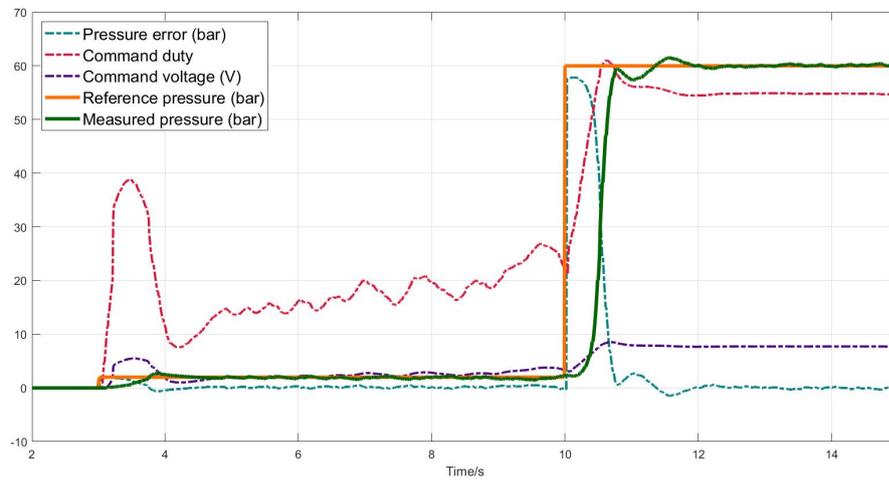


Figure 5.4: Output response to a 60 bar step reference, with pre-braking

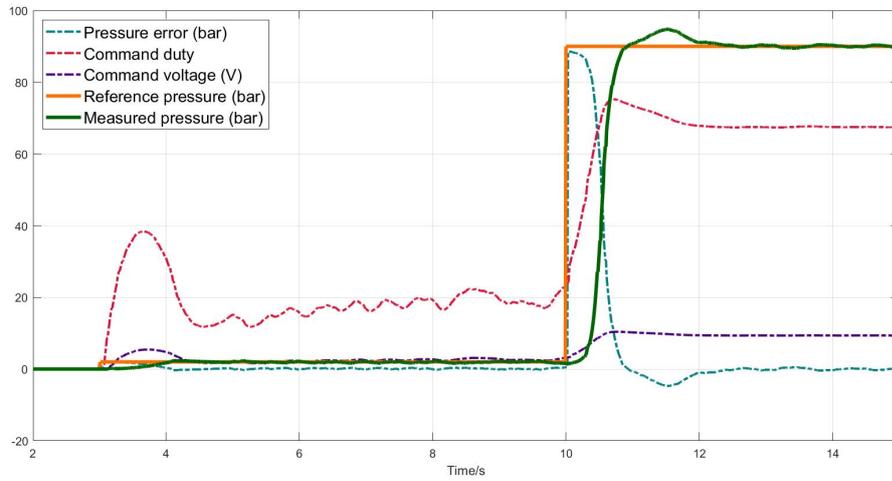


Figure 5.5: Output response to a 90 bar step reference, with pre-braking

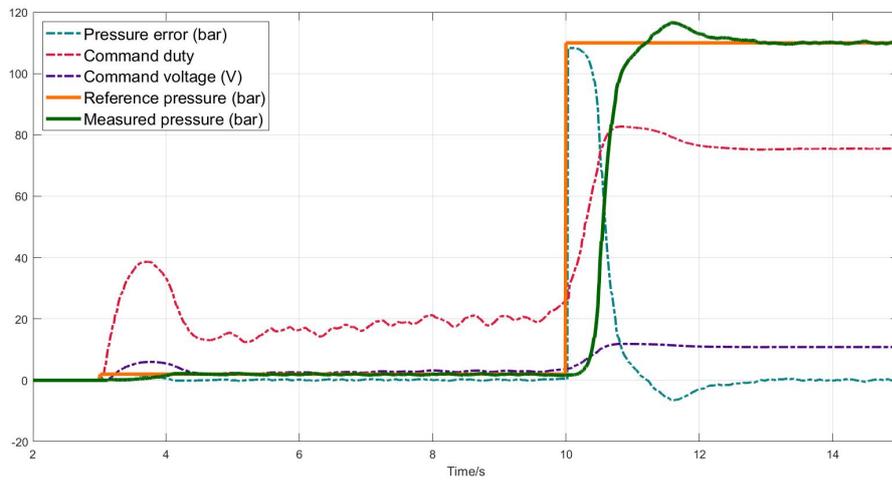


Figure 5.6: Output response to a 110 bar step reference, with pre-braking

Additionally, a reference value different from the working points is tested (Figure 5.7), to verify the functioning of the LUT linear interpolation.

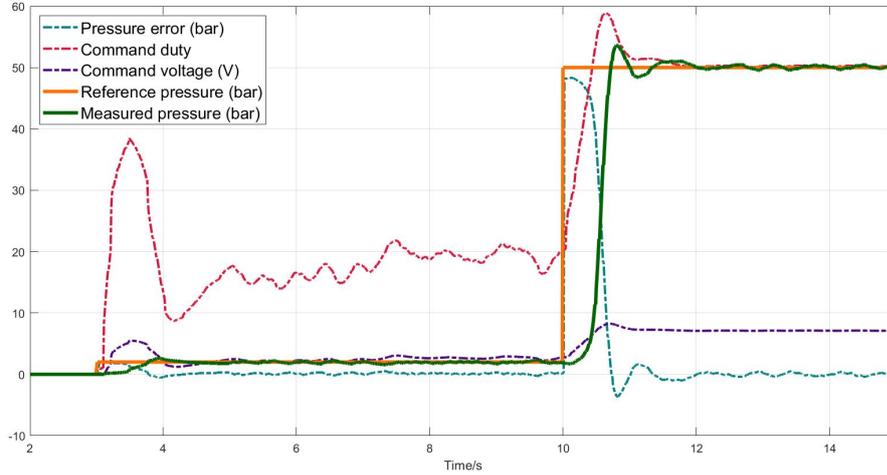


Figure 5.7: Output response to a 50 bar step reference, with pre-braking

Finally, the measured transient parameters of all the figures are resumed in table 5.4.

Table 5.4: Output response transient parameters on the testbench

Transient parameters	With pre-braking insertion						
	20 bar	20 bar	40 bar	50 bar	60 bar	90 bar	110 bar
Rise time [s]	2.170	0.926	0.644	0.667	0.666	0.717	0.815
Overshoot %	14.85	8.30	7.57	7.28	2.53	5.31	5.97

Even though the rise time requirement is not fulfilled, it's possible to notice a remarkable improvement with respect to the case without pre-braking. Now, the obtained rise time values are much closer to the target one. Moreover, all the obtained overshoots respect the 10% target value and few oscillations can be observed. Therefore, the whole obtained performances can be considered acceptable. The only drawback of this solution is that, when mounted on the vehicle, the actuator tends to slow down the car over time, due to the constant application of the 2 bar.

5.2.3 Non-linearity handling

Due to the non-linearity, performances degrade when the system works in a neighborhood of the bottom extremity of the characteristic. What can be noticed, is a substantial difference of the 2 bar PI parameter values with respect to the 20 bar ones. In order to overcome the problem, a further fitting of the look-up tables is performed in that region. In table 5.5 are reported the updated control parameters for all the working points.

Table 5.5: K_p and K_i values for each operating region (V3)

Parameter	Working point [bar]												
	0	2	3	4	6	8	12	16	20	40	60	90	110
K_p	30	20	10	5	2	1.2	1.1	1	0.95	0.7	0.6	0.4	0.35
K_i	30	20	15	12	10	8	5	4.6	4.3	4.3	3.8	3	2.5

5.2.4 Sinusoidal and triangular references

In real world applications, ideal steps, like the ones previously tested, never occur. In order to assess the controller behaviour in operating conditions closer to a real braking scenario, both triangular and sinusoidal reference signals are tested.

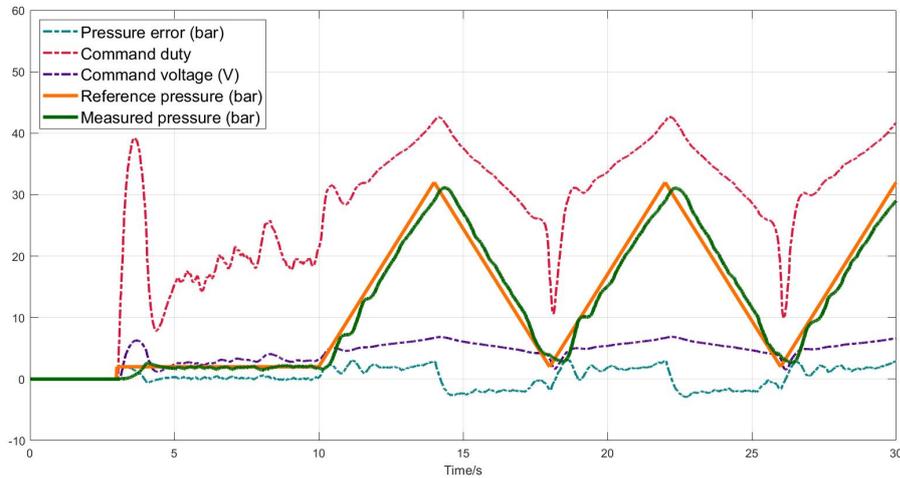


Figure 5.8: Output response to a triangular reference with $T = 8$ s in the interval $(2 - 32)$ bar, with pre-braking

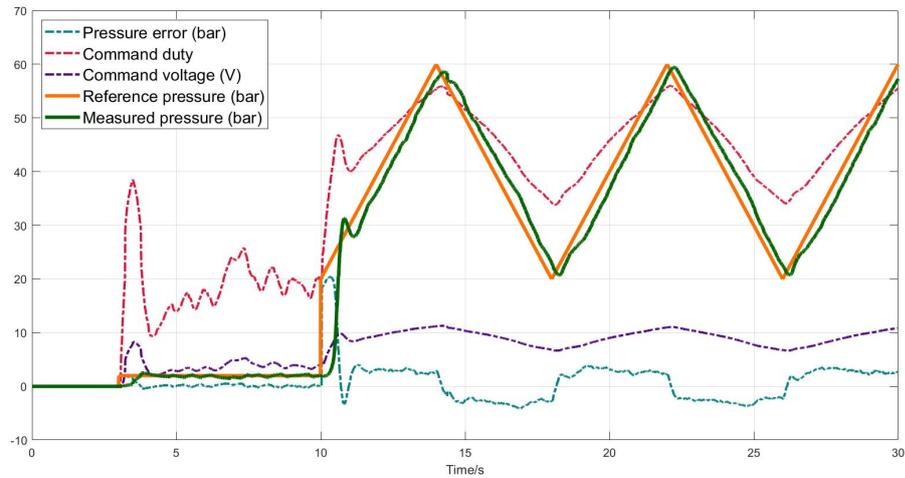


Figure 5.9: Output response to a triangular reference with $T = 8$ s in the interval $(20 - 60)$ bar, with pre-braking

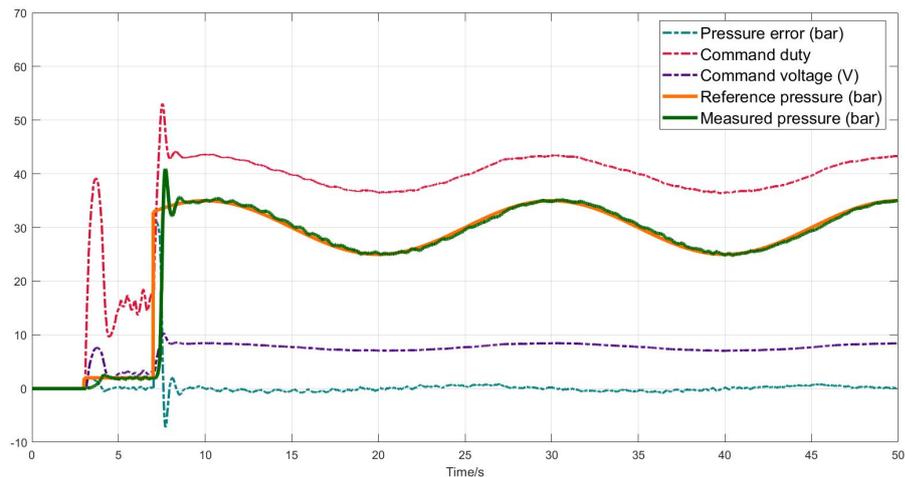


Figure 5.10: Output response to a sinusoidal reference with $f = 50$ mHz in the interval $(25 - 35)$ bar, with pre-braking

5.2.5 Observations

As showed in figure 5.8, the denser fitting in the bottom region has produced good performances. The output is able to chase the reference, even when crossing smaller pressure values. Overall, after the output is settled, both the sinusoidal and the triangular references are properly tracked (figures 5.8, 5.9, 5.10), when

their frequency fulfills bandwidth boundaries. Moreover, the error is always kept below a threshold of 3 bar, which can be definitely considered acceptable.

In conclusion, the obtained controller can successfully follow the required setpoint while achieving good performances, and therefore, it can be mounted over the vehicle.

5.3 Closed loop system bandwidth

To understand which is the frequency operating range of the closed loop system, a bandwidth computation is required. In particular, the Bode plots of the phase and of the module of the closed loop transfer function are determined empirically. The system is stimulated with sinusoidal reference signals at different frequencies and the corresponding output pressure is measured. Afterwards, each input-output couple is compared and information about gain and phase delay are extracted. In figure 5.11 is reported an estimate of the obtained diagrams, whose code is reported in appendix A.6.

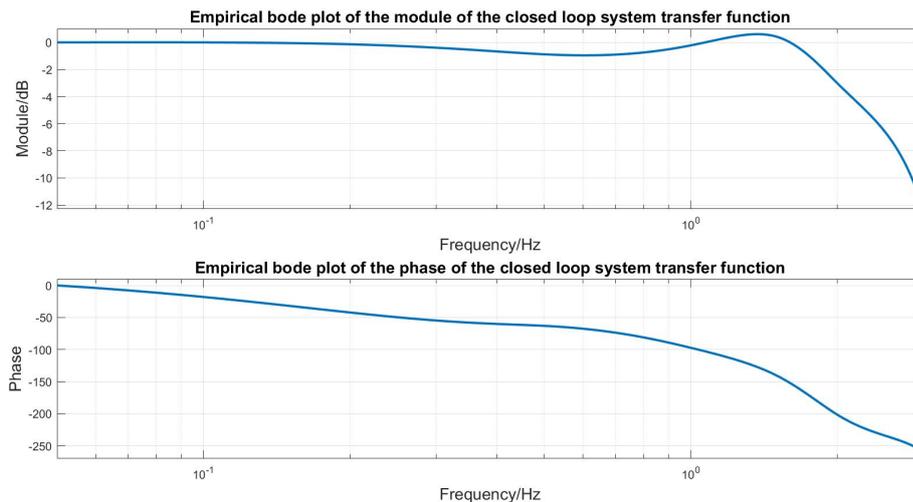


Figure 5.11: Bode diagrams of the closed-loop system

By looking at the graph, it can be noticed where the -3 dB cut-off frequency falls. A good guess for this value can be $f_{cut} \approx 2$ Hz, which is close to the value of the open-loop bandwidth.

In the following figure is represented the output response to a sinusoidal reference with a frequency close to the cut-off one. The output amplitude is roughly 0.707 times the input, whereas, concerning the phase, there is a delay of approximately 180° , coherently with the phase Bode plot of figure 5.11. When two signals are

subjected to this lag condition, they are said to be in phase opposition. Since this state must be avoided, the maximum suitable frequency which is chosen to enter the system is about 1 Hz. Greater values would result in a consistent reduction of the output amplitude, together with a substantial phase delay.

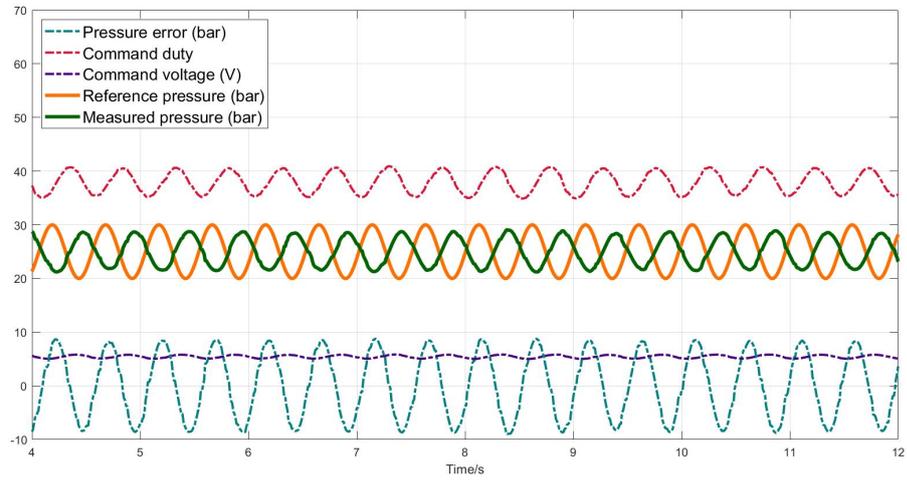


Figure 5.12: Output response in a neighborhood of the cutoff frequency

Chapter 6

Integration on the vehicle

Once the quality of the designed control algorithm is consolidated, the focus of the study transitions from the test bench towards the real plant.

Therefore, the actuator is mounted over an L4 experimental autonomous vehicle (figure 6.1), where the control algorithm is integrated and tested. The vehicle is a research and development project called VeGA, where a BEV Citroën E-Méhari is employed as base car. It belongs to the M1 vehicle category, which, according to the UNECE, includes all vehicles used for the carriage of passengers and comprising not more than eight seats in addition to the driver's one [46].



Figure 6.1: VeGA, autonomous vehicle

The company Bylogix s.r.l. has designed the electric and electronic architecture

of this car, allowing an easy integration of further features [47]. The autonomous driving architecture of this car currently lacks of an autonomous braking system, which needs therefore to be implemented. Broadly speaking, the adopted solution involves a custom brake-by-wire system, where the studied actuator can directly apply pressure in the braking circuit. As previously discussed, in classical BBW, braking effort is decided on the base of the pedal stroke measured by a suitable sensor, which transmits the signal to the HCU. Instead, in this solution, autonomous driving unit establishes the braking urgency by perceiving the surrounding environment and conveys the information to the HCU. The latter transforms the value into a reference pressure and employs it in the gain scheduled PID control algorithm previously designed. Thus, no implementation of the PFE is required and the hydraulic transmission between pedal brake and wheel calipers is preserved, allowing an easy integration with the legacy preinstalled braking system. Autonomous driving unit has control of the braking manoeuvre throughout normal operating conditions. Anyway, the driver can intervene through the brake pedal at any time, especially in case a failure in the actuator occurs, consequently working as a failsafe backup system. Indeed, a suitable logic is designed, allowing to disable the actuator whenever a pedal strain is detected. In conclusion, both the driver and the autonomous driving unit have the chance of exerting pressure, one at the time, in the same hydraulic circuit.

6.1 Electrical harness

The electrical harness is designed, in order to integrate in the vehicle the whole system, composed by the pressure sensor, the CAN PLC and the E/H actuator. All the wirings are identified by a number and a letter. The number are the mm^2 of the conductor section, whereas the letter stands for the color of the real wire.

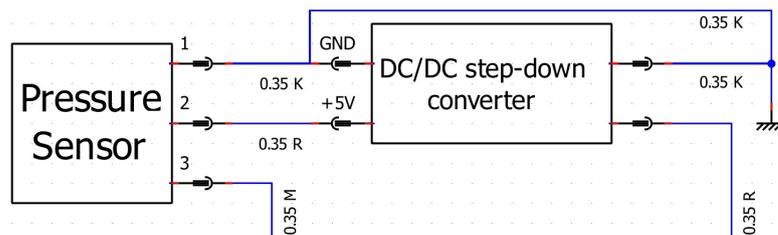


Figure 6.2: Pressure sensor electrical connections

Concerning the pressure sensor (figure 6.2), a DC/DC step-down converter is employed to transform 12V battery voltage into the required 5V power supply, which was previously provided by the NI-6009 acquisition device. The pressure measurement is provided as voltage value to the CAN PLC on pin C7.

For the control unit (figure 6.3), a total of 11 pins are leveraged. Pins B3 and B2 are connected to a temporary dedicated CAN bus with a terminal line resistance of $120\ \Omega$. On this CAN are going to travel both the provided reference and all the measured signals such as measured pressure, command duty and tracking error. Pins B8, B1 and C1, B6 are respectively the positive and negative of the battery voltage. Pin A8 generates the PWM command voltage, and as for the testbench, a $1\ \text{k}\Omega$ pull-down resistor is required.

In order to avoid useless current absorption when the car is switched off, both the actuator and the pressure sensor supplies are placed under vehicle ignition. This helps preserving the battery life from draining over time, and it's implemented through some relays devices which allow current transition only after vehicle ignition. Concerning the PLC, only the KL15 ignition key contact (Pin C8) is placed under vehicle ignition. Instead, power supply is always connected, thus causing shutdown current absorption when the device is turned off. Anyway, this choice allows the PLC to perform final operations, such as logging, before switching off, and eventually prevents bigger current leakages by disabling the device.

Instead, pin C5 is connected to a potentiometer of $10\ \text{k}\Omega$, which is going to be used as real-time user-driven pressure reference from the vehicle cockpit. Lastly, there is pin C6, which is connected to the brake pedal sensor to give the driver the opportunity of disabling the actuator at any time, by simply pressing the pedal.

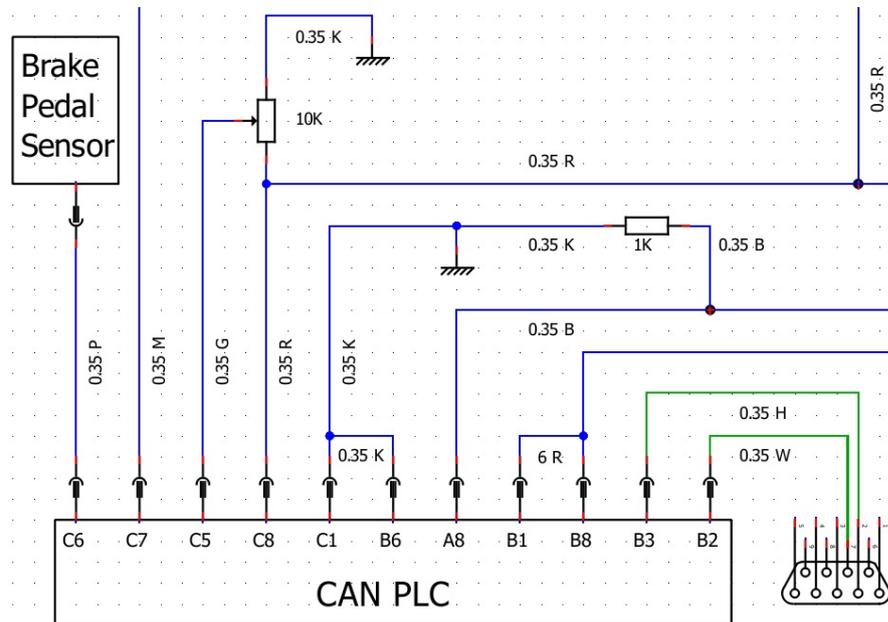


Figure 6.3: CAN PLC electrical connections

Regarding the actuator (figure 6.4), between the positive battery voltage and

6.2 Control integration

6.2.1 PID re-tuning

By altering the surrounding conditions of the actuator, a PID re-tuning is necessary in order to adapt the control algorithm to the new environment. The table of the new obtained parameters is reported hereafter (table 6.1). A maximum value of 80 bar is considered to be enough to completely lock the wheels, thus the parameters are tuned up to that value of the scheduling variable.

Table 6.1: K_p and K_i values for each operating region (V4)

Parameter	Working point [bar]										
	0	2	3	5	8	12	16	20	40	60	80
K_p	30	17	10	4	1.85	1.7	1.45	1.3	0.8	0.6	0.55
K_i	30	25	20	15	13.5	12.5	10	9.5	4.75	3.6	2.85

6.2.2 Brake pedal role

Although in standard operating conditions the autonomous driving unit has control of the braking manoeuvre, a driver intervention may be necessary, especially in hazard situations. The latter include possible failures, for instance either in the actuation device or in the pressure sensor, but also unforeseen scenarios which may lead the autonomous driving system not to brake in time. Under these conditions, the driver has the opportunity to disable the pump by simply pressing the pedal, thus working as a failsafe backup system. This is possible thanks to a suitable designed logic, which is implemented in the control unit firmware (figure 6.5).

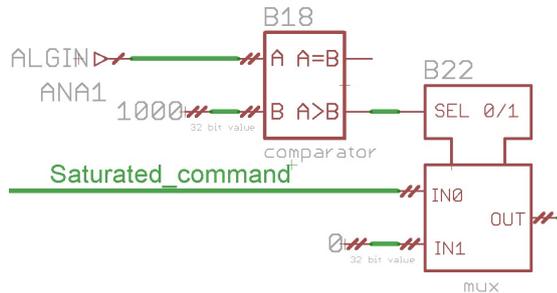


Figure 6.5: Pedal logic implemented in Eagle

Analog input port ANA1 (Pin C6) of the PLC is connected to the same switch which turns on the vehicle brake lights. Whenever a slight pedal strain is perceived, the switch is closed and a voltage of 12 V can be read. According to the PLC datasheet [33], the maximum readable voltage is 11.4 V. Anyway, the logic is designed in such a way that a 0 signal is selected by a multiplexer, upon reading a voltage greater than 1 V. A non-null value is chosen as threshold limit to avoid the risk of switching off the pump due to little voltages detection.

The designed logic is inserted in two positions of the Eagle diagram. One is placed in open loop before the PWM block (figure 6.5), such that it can override the duty control input and instantly disable the actuator. The other one is positioned in parallel to the reference pressure provided by the autonomous driving unit, such that it can rapidly discharge the integrator term, which otherwise would keep increasing. Although by pressing the pedal the vehicle is slowing down and consequently the AD-ECU is requiring less braking effort, being the sensor at the actuator output, the measured pressure is equal to 0. As a result, the integrator keeps rising, and for this reason it needs to be discharged by placing the logic also in closed loop.

This logic is designed in such a way that, if the autonomous driving unit is asking to brake the vehicle and the pedal is pressed, the pump switches off, but if the pedal is released and the target velocity is not reached yet, the pump switches on again, until the target speed is achieved. This implies that consecutive press-release actions during a braking manoeuvre must be avoided, since may lead the system to oscillate or even worse to an unstable state, due to the repeated change in the system rigidity.

6.3 Stationary vehicle results

6.3.1 Simulink

The control algorithm is first verified under stationary vehicle conditions.

Table 6.2: Output response transient parameters on the vehicle

Transient parameters	Reference pressure			
	20 bar	40 bar	60 bar	80 bar
Rise time [s]	0.370	0.457	0.348	0.617
Overshoot %	2.83	3.51	5.40	4.23

Steps of different amplitude and periodical signals are provided as reference through Simulink and the results are collected. In the following figures are represented some of the tested references. Moreover, a table including the transient parameters for different steps is reported above.

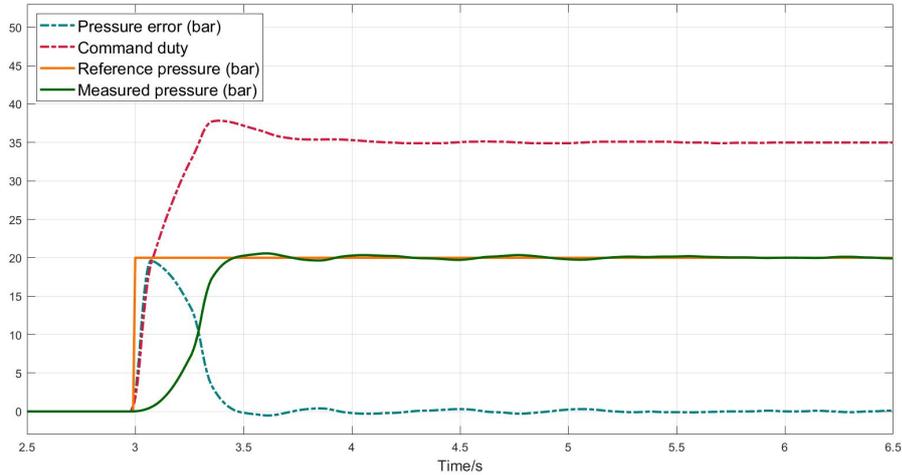


Figure 6.6: Output response to a 20 bar step reference, on the vehicle

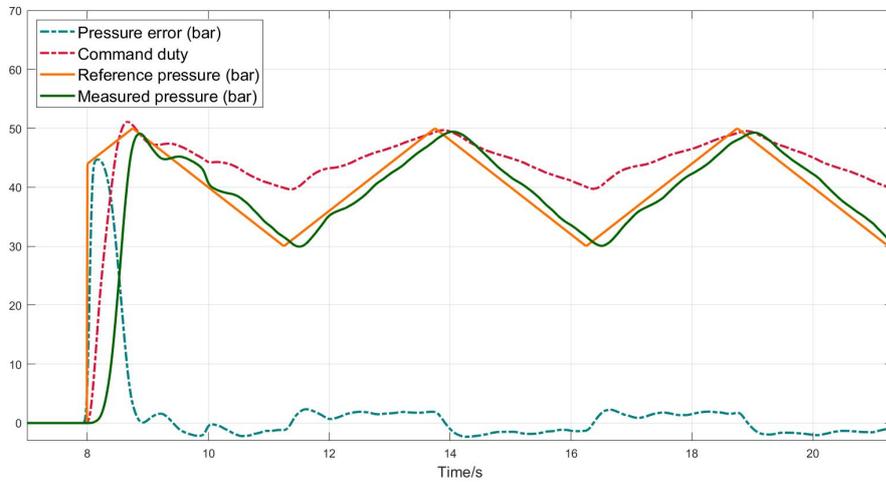


Figure 6.7: Output response to a triangular reference with $T = 5$ s in the interval $(30 - 50)$ bar, on the vehicle

By analyzing the obtained results, it can be observed that a remarkable improvement is achieved with respect to the test bench values. Concerning the tracking of

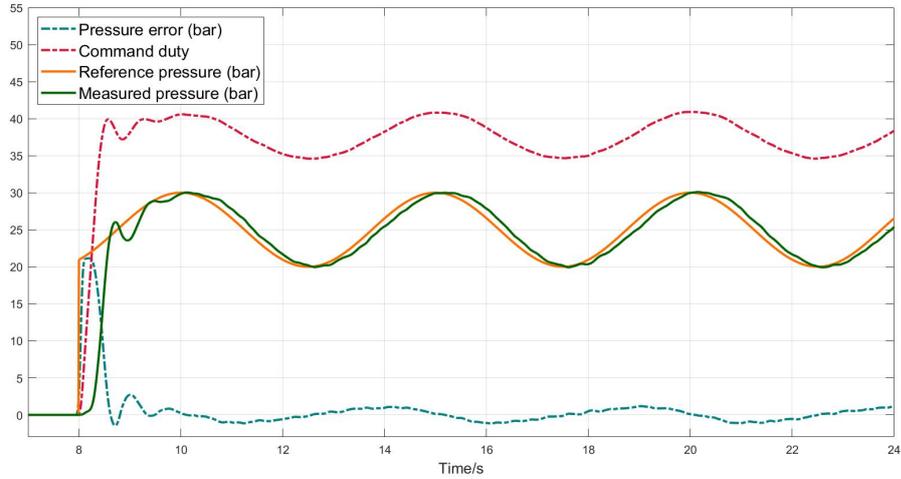


Figure 6.8: Output response to a sinusoidal reference with $T = 5$ s in the interval $(20 - 30)$ bar, on the vehicle

the periodical signals, the quality is preserved. Instead, by looking at the values of the table 6.2, not only all the overshoots are more than halved, but also the rising time have shrunk with respect to table 5.4. Almost each rise time is now below the 0.5s established as requirement and without even the need of inserting a pre-braking phase to activate the actuator. Most likely, an hypothetical reason for which test bench results were worse, could be the presence of air within the brake circuit. Air could cause a delay in the process of building up pressure due to its compressibility, differently from liquids. If this is the case, it might be the reason why all the system identification trials failed.

6.3.2 Oscillation phenomenon

In figure 6.9 is reported the oscillation phenomenon which can be observed whenever the output pressure is crossing the region below 15 bar. Oscillations are a dangerous occurrence which may lead the system to unstable conditions. Since the system has to be retained in a stable region, a derivative action is inserted in the control algorithm for the region between 3 bar and 12 bar. Derivative action has the advantage of reducing overshoot and oscillations, with the drawback of slightly worsening the rise time. The design of this new term is performed to gradually enter the unstable interval and to progressively fade when exiting the region.

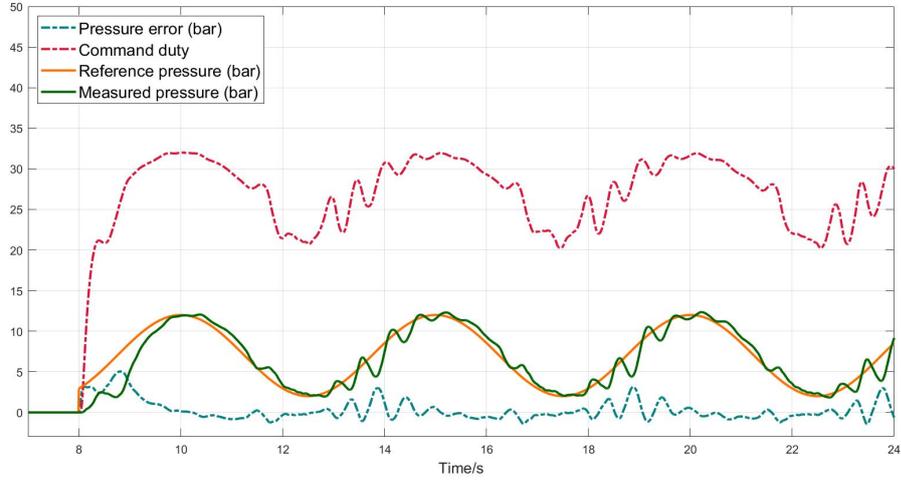


Figure 6.9: Oscillation phenomenon

Figure 6.10 reports the Eagle diagram in charge of performing the derivative action, where *variable_in_9* is the derivative coefficient computed through the LUT defined in the C-usercode. The schematic implements the derivative term of 4.1, but without the N filter parameter, thus resulting in the following discrete transfer function:

$$F_d(z) = \frac{K_d z - 1}{T_s z}$$

Finally, the concluding PID parameters used on the vehicle are reported below in two separate tables.

Table 6.3: Final K_p, K_i and K_d values for each operating region, part A

Parameter	Working point [bar]										
	0	2	3	4	5	6	7	8	10	12	16
K_p	30	20	2.2	1.8	1.7	1.9	2	1.9	1.9	1.8	1.4
K_i	30	25	20	17	15.5	15	16	14	13.5	12.5	10.5
K_d	0	0	0.02	0.04	0.07	0.04	0.03	0.02	0.02	0.02	0

Table 6.4: Final K_p, K_i and K_d values for each operating region, part B

Parameter	Working point [bar]			
	20	40	60	80
K_p	1.3	0.8	0.6	0.55
K_i	9.5	4.75	3.6	3
K_d	0	0	0	0

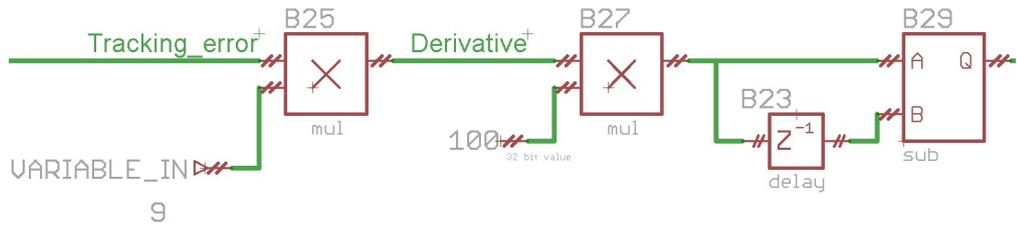


Figure 6.10: PID derivative implementation in Eagle

6.3.3 Different braking profile

Figure 6.11 reports the attempt of tracking a pressure reference signal closer to a real world scenario.

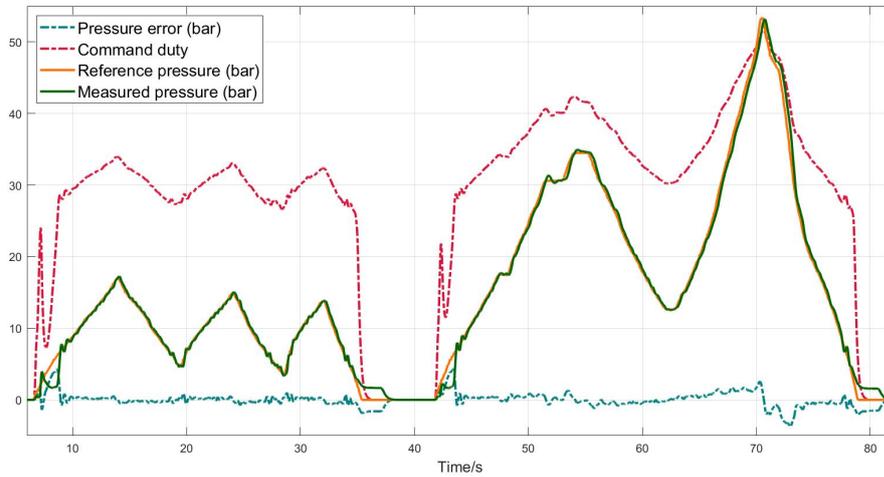


Figure 6.11: Realistic braking profile

This profile is issued through a 10kΩ potentiometer, which is connected to

the PLC thanks to the previously arranged electrical harness. The potentiometer voltage is read through an analog input pin, together with the reference pressure coming through the CAN bus. Then, within the C-usercode, in order to handle the reference selection, a suitable code is written, which is reported below.

By observing the profile, it's possible to notice that all the oscillations are canceled, however, as a straight consequence, there is a perceivable delay when the output has to track the reference in the unstable region. Indeed, although most of the time the overall tracking error is consistently kept below 1.5 bar, in a neighborhood of the critical region it increases up to ≈ 4 bar, still remaining reasonable.

Control algorithm tracking performance is verified by computing the RMSE, exploiting the square root of equation 3.2. For this specific profile, the obtained value is $RMSE \approx 0.928$ bar, which is definitely a good result.

```

1 user_code.c
2 //Potentiometer voltage acquisition through Analog
  Input pin.
3 uint16_t voltage_pot=user_variable [7];
4
5 //If potentiometer voltage is over 1V, use it as
  reference pressure.
6 if(voltage_pot>=1000){
7
8     //Restricted interval [1,11]V is encoded
9     int16_t arr_x1 [] = {1000,11000};
10    //Corresponding pressure range of interest
11    int16_t arr_ref_pressure [] = {0,800};
12
13    uint8_t length_arr_x1 = sizeof(arr_x1)/sizeof(
arr_x1 [0]);
14
15    //LUT to assign the reference depending on
  potentiometer knob rotation.
16    user_variable [6] = os_util_lookup1D(arr_x1 ,
arr_ref_pressure , length_arr_x1 , voltage_pot ,
LUT_MODE_LIMIT);
17 }
18 else{
19     //Pressure reference is read from CAN bus.
20     user_variable [6] = user_variable [8];
21 }

```

Listing 6.1: User code implementing gain scheduling

6.4 Moving vehicle results

Different testing are performed on the road with the vehicle in motion. The testing environment is the surrounding area of Bylogix building, which is reported in figure 6.12. The vehicle has to be tested in a closed private space, since, being an L4 experimental autonomous vehicle, it's not legally authorized to travel on public roads.



Figure 6.12: Braking test environment around Bylogix building

According to AASHTO [4], 3.4 m s^{-2} is the comfortable deceleration value for most drivers. Thus, referring to that threshold, three different scenarios are analyzed: a comfortable deceleration within the limit value, a comfortable deceleration outside the limit value, an uncomfortable deceleration outside the limit value. All the tests are carried out over slightly wet asphalt condition, hence performances may improve on a dry asphalt. In order to perform the measurements, the vehicle CAN bus is scanned, looking for the frame related to the vehicle speed. The located frame is the following and the read value is expressed ad Km/h :

Message_Address : $0x540$, *Signal* : *Vehicle_Speed*, *Bitstart* : 0, *Bitlength* : 8

Once the speed frame address is obtained, it's used to build a suitable CAN database on the software tool CANalyzer, together with the measurements of the tracking error, the duty control input and the measured output pressure, which all travel on a reserved CAN bus. Different trapezoidal reference pressure profiles are provided through the software, and then reacquired together with the just mentioned signals. In order to have an idea of the braking manoeuvres, it's

necessary to retrieve also the travelled space and instantaneous acceleration. Since, for this purposes, the vehicle odometry is able to provide a measurement whose resolution is too coarse, another solution is adopted. Vehicle speed is exported from CANalyzer software into Simulink, where it's derived to obtain the instantaneous acceleration, and it's integrated to obtain a rough estimate of the travelled distance. Finally, several parameters are measured from the obtained profiles and all the values are reported in table 6.5.

Table 6.5: Parameters related to three different deceleration scenarios

Braking type	Perceived braking	Braking time [s]	Braking distance [m]	Initial speed [m/s]	Deceleration max. [m/s^2]
Soft	Soft	5.202	15.909	6.389	-2.222
Medium	Soft	3.316	11.029	6.852	-3.889
Harsh	Harsh	2.248	6.846	6.637	-5.556

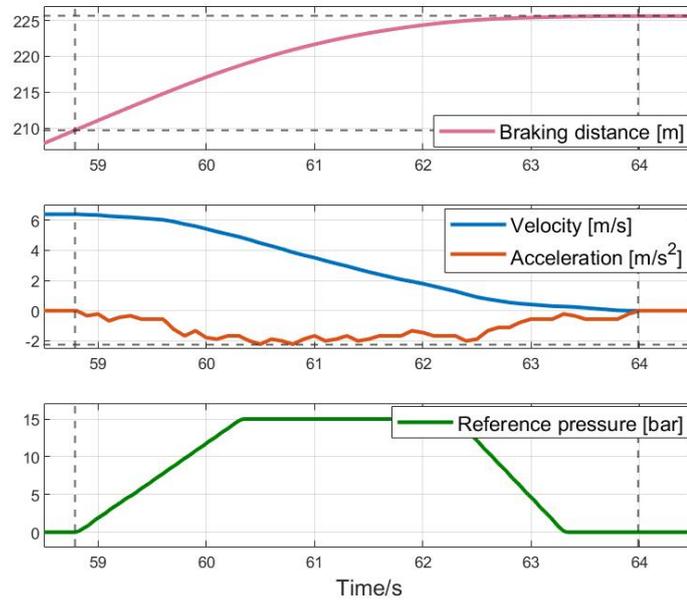


Figure 6.13: Soft braking results

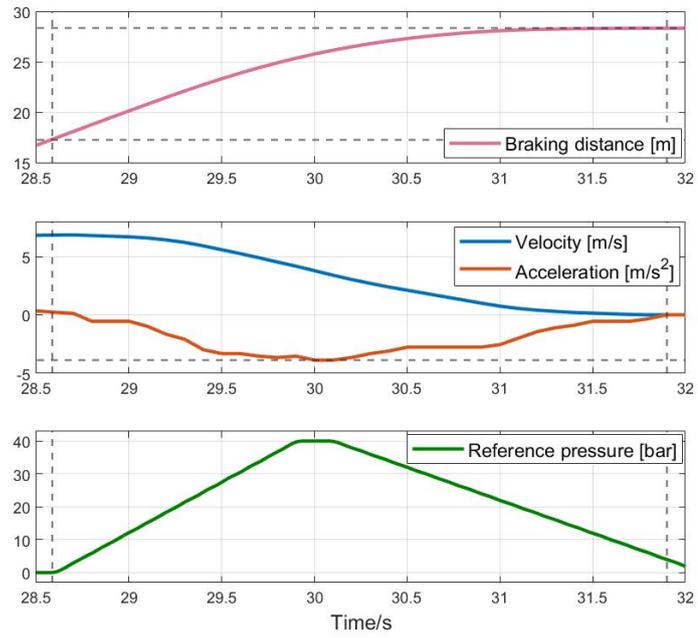


Figure 6.14: Moderate braking results

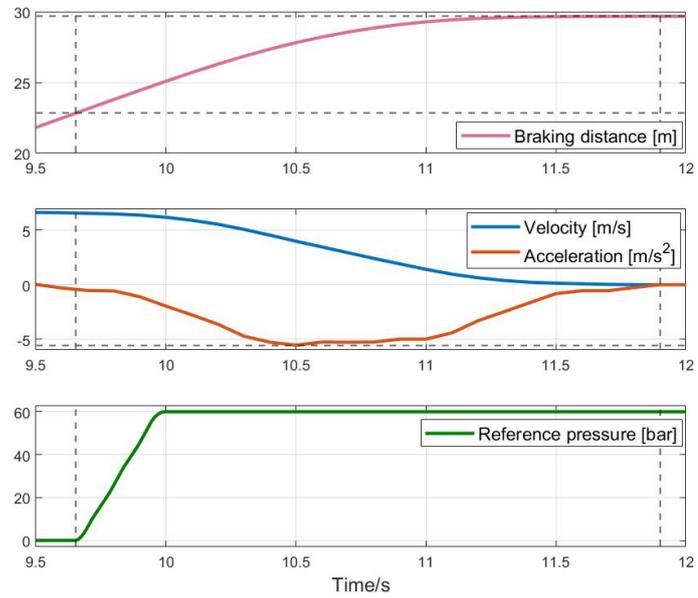


Figure 6.15: Harsh braking results

6.4.1 Braking system homologation

The following formula comes from the regulations of the Economic Commission for Europe of the United Nations and it is leveraged to homologate braking systems for vehicles of category M1 [48]:

$$s \leq 0.1v + \frac{v^2}{150} = 0.1v + 0.0067v^2 \quad (6.1)$$

where,

- s is the stopping distance in m
- v is the prescribed test speed in km/h

According to homologation rules, the prescribed test velocity has to be 80% of the vehicle maximum speed and \leq than 160 km/h. As showed in figure 6.12, due to the environment conformation, the maximum reachable speed is below 25 km/h. However, 6.1 is exploited as benchmark to have an indicative idea of whether the system can achieve standardized performances at least at lower speed.

In figure 6.16 is reported the result of a braking test where the vehicle initial speed is 22.5 km/h and the applied reference pressure reaches 70 bar in 0.1 s with a step pressure profile.

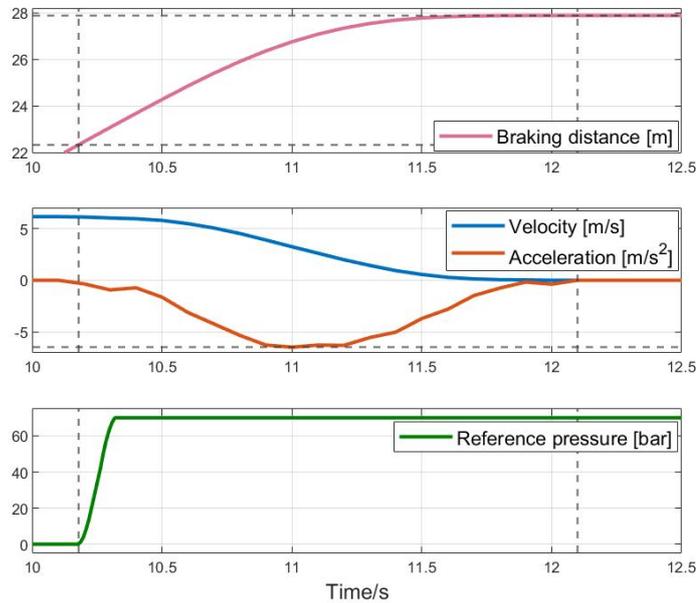


Figure 6.16: Braking cycle for homologation

The obtained braking distance measures ≈ 5.56 m, which results to be smaller than the benchmark value ≈ 5.62 m. Thus, the custom BBW system, composed of the actuator and of the designed control algorithm, is able to successfully fulfill homologation requirements at least at lower speeds.

Chapter 7

Conclusions

This thesis points at verifying whether the Dexter E/H 1600 actuator, commonly employed in trailer braking systems, can be successfully exploited to automate brake operations of a real vehicle. The developed solution consists in a custom brake by wire system, where the actuator directly generates the hydraulic pressure within the brake circuit under control of the braking control unit.

Before designing the control algorithm, a suitable acquisition environment is conceived, with the aim of studying the system behaviour under different conditions. All the collected datasets are leveraged to perform several attempts of system identification.

Afterwards, considering the system non-linearity, a gain scheduled PID controller with anti-windup is implemented. Generally, this actuator should achieve optimal performances when employing a controller of the same manufacturer. However, in this dissertation is showed how it's possible to accomplish good performances even with a custom made controller. Indeed, the results obtained on the test bench shows that satisfactory braking performances are met whichever the reference pressure. Finally, the BCU-pump system is mounted and integrated on the vehicle and final tests are performed, both stationary and on the road, under user-driven references. With respect to the testbench results, a remarkable improvement can be observed. Although the insertion of the derivative term has introduced a perceivable delay, satisfactory performance are achieved. Moreover, braking system homologation benchmark from UNECE is fulfilled under low speed conditions. Indeed, the system is capable of providing both a smooth and comfortable slowdown in standard circumstances and a fast deceleration in presence of emergency scenarios.

In conclusion, the actuator under study can be considered suitable for the vehicle application and, with the appropriate control algorithm, lays promising foundations for effectively handling the vehicle autonomous braking manoeuvres.

7.1 Future works

Based on the developed project, further improvements and studies can be carried out:

- A straightforward extension of the work is the integration of the control algorithm with the autonomous driving unit. Up to now, the reference pressure has been provided through CAN line by user driven sources. In order to let the AD-ECU handle the braking task, it has to be properly interfaced with the BCU. Primarily, some available addresses has to be identified in the CAN database, to be used as communication channel between the two ECUs. Then, since the AD-ECU issues a urgency braking percentage, it has to be suitably converted into the pressure range of interest. Final testing have to be performed to verify whether the AD-ECU can handle the autonomous braking manoeuvre over different scenarios.
- Although some trials have been performed, system identification has not been deepened in this dissertation. The possibility that air presence may have impaired the testbench measurements, requires an entire new group of datasets to be collected. Moreover, if the already tested polynomial discrete time models are not sufficient, a Linear Parameter Varying (LPV) modelling technique may be employed to derive a model. Linear local models are identified using data sets at corresponding working-points; then a final LPV model is obtained through interpolation of the linear ones [49].
- As observed from the results, despite a reduction of the oscillations, the insertion of the derivative term causes a worsening of the response time. Primarily, the $I_u(w)$ term of the GS global control 4.4 can be implemented to verify a potential performance enhancement, since it has been neglected in this dissertation. Moreover, further control algorithms can be tested, trying to achieve improved performances. In order to be suitable for the purpose, they have to be discrete non-linear model free controllers which can be implemented through C-code. For instance algorithms like Embedded Model Control (EMC) or Sliding Mode Control (SMC) can be tested out, although they require at least a simplified model of the plant. A valuable alternative could be purely Data-Driven Control (DDC) algorithms, which however are still confined to the research sphere and not well established yet.
- Furthermore, a comparison with the performance achieved by a Dexter controller can be realized, in order to further validate the quality of the developed control algorithm.

- Concerning safety redundancy, is recommended the insertion of an additional pressure sensor with a suitable control algorithm able to manage both the measurements. This allows to handle hazardous situations, where a failure of the principal sensor occurs.
- As observed during the characterization, temperature variations may affect system behaviour and parameters such as oil viscosity, hence a thermal analysis can be performed to study the actuator dependency on temperature.

Appendix A

Matlab code

A.1 Calibration

```
1 %Calibration cycles
2 n=10;
3
4 %Inizialization to 0 of the vectors I have to fill
5 offset_pressure_vector=zeros(n,1);
6 offset_command_vector=zeros(n,1);
7
8 for i=1:n
9     tsim=10;
10    sim('Calibration')
11
12    %Storing in another variable the values coming from Simulink
13    pressure(:,1)=sensor.data(1,1,:);
14    command(:,1)=command_voltage.data(1,1,:);
15
16    %Saving mean value in a vector
17    offset_pressure_vector(i,:)=mean(pressure);
18    offset_command_vector(i,:)=mean(command);
19 end
20
21 %Average of the offset values obtained
22 offset_pressure=mean(offset_pressure_vector);
23 offset_command=mean(offset_command_vector);
```

A.2 Static conditions acquisition

```

1 %Loading offset values and setting simulation time
2 load offset.mat offset_command offset_pressure
3 tsim=6;
4
5 %Initializing vectors to zero
6 command_mean_v=zeros(3,1);
7 pressure_mean_v=zeros(3,1);
8
9 %Three cycles of measurements
10 for i=1:3
11
12     sim('Measurement')
13
14     %Mean value over all the data collected in a single simulation
15     %Reverse formula of voltage divider, to obtain original V_in
16     command_mean_v(i,:)=mean(command_voltage.data(1,1,:)*1962/982);
17     pressure_mean_v(i,:)=mean(sensor.data(1,1,:));
18
19 end
20
21 %Mean value of the three measurement cycles
22 command_mean=mean(command_mean_v);
23 pressure_mean=mean(pressure_mean_v);

```

A.3 Static conditions data processing

```

1 %Loading data previously imported from excel file
2 load data_set_1
3
4 %Defining straight lines passing through minimum and maximum points
5 v=linspace(min(command_voltage),max(command_voltage),...
6     length(command_voltage)); %Command voltage
7 p=linspace(min(output_pressure),max(output_pressure),...
8     length(output_pressure)); %Measured pressure
9 c=linspace(min(12*absorbed_current),max(12*absorbed_current),...
10    length(12*absorbed_current)); %Absorbed power
11
12
13 %Find position of the maximum value
14 pos_max=find(command_voltage==max(command_voltage));
15 %Sorting in ascending order the decreasing sweep values

```

```

16 ordered_p=zeros(pos_max,1);
17 ordered_v=zeros(pos_max,1);
18 for i=1:pos_max
19     ordered_p(i,:)=output_pressure(end+1-i);
20     ordered_v(i,:)=command_voltage(end+1-i);
21 end
22 %Interpolating ascending and descending command values over 100
    points
23 base=linspace(min(command_voltage),max(command_voltage));
24 p1=interp1(command_voltage(1:pos_max),output_pressure(1:pos_max),...
25     base,'spline');
26 p2=interp1(ordered_v,ordered_p,base,'spline');
27 %Doing the mean value of the two interpolated curves
28 av=zeros(100,1);
29 mat=[p1,p2];
30 for i=1:100
31     av(i,:)=mean(mat(i,:)); %Final single interpolated curve
32 end
33
34
35 %Find an initial estimate of the gradient gain_K
36 j=1;
37 while (base(:,j)<=7.1)
38     pos_K=j;
39     j=j+1;
40 end
41 gain_K=av(pos_K,:)/(base(:,pos_K)-base(:,1));
42 %Find an initial estimate of the y-intercept q1
43 %y-y0=m*(x-x0) -> y=mx-mx0+y0 -> q=-mx0+y0
44 q1=-gain_K*base(:,pos_K)+av(pos_K,:);
45 %Optimization problem
46 opt = @(par) fnc(par(1),par(2),base,av);
47 %Using as inisial guess the values previously computed
48 m_guess=gain_K;
49 q_guess=q1;
50 %Minimizing output of fnc function
51 [par_min,fval] = fminsearch(opt, [m_guess,q_guess]);
52 %Optimal parameters for the best approximating line
53 m_opt=par_min(1);
54 q_opt=par_min(2);
55
56
57 function [out] = fnc(m,q,base,av)
58     %Line equation depending on parameters to be optimized
59     line=m*base+q;
60     %Function to be minimized: standard deviation
61     out=sum(((av-line).^2)/length(av));
62 end

```

A.4 Dynamic conditions data processing

```

1 Fs=5000; %Sampling frequency
2 Nfir=10; %Filter order
3 Fst=10; %Cut-off frequency
4
5 %Filter design
6 firf=designfilt('lowpassfir','FilterOrder',Nfir,'CutoffFrequency',Fst
7     , 'SampleRate',Fs);
8 %Average input voltage
9 pwm_mean_vector_f=filtfilt(firf,pwm_mean_vector);
10 %Output pressure
11 pressure_vector_f=filtfilt(firf,pressure_vector);
12 %Output voltage
13 sensor_voltage_vector_f=filtfilt(firf,sensor_voltage_vector);

```

A.5 System identification

```

1 %Load all estimation datasets
2 load 'Chirp_duty_300_to_800_01mHz_1Hz_estimation.mat'
3 load 'Chirp_duty_400_to_600_01mHz_1Hz_estimation.mat'
4 load 'Ramp_up_down_250_to_850_to_300_700_10mHz_set1_estimation.mat'
5 load 'Random_uniform_duty_300_700_1Hz_estimation.mat'
6 load 'Random_uniform_duty_400_600_1Hz_set0_estimation.mat'
7 load 'Random_uniform_duty_400_600_1Hz_set1_estimation.mat'
8 load 'Sine_amp_mod_450_550_to_300_700_05Hz_estimation.mat'
9 load 'Sine_amp_mod_450_550_to_300_700_5mHz_estimation.mat'
10 load 'Step_duty_0_to_300_set0_wo_offset_estimation.mat'
11 load 'Step_duty_0_to_300_set1_wo_offset_estimation.mat'
12 load 'Step_duty_0_to_300_set2_wo_offset_estimation.mat'
13 load 'Step_duty_0_to_450_set0_wo_offset_estimation.mat'
14 load 'Step_duty_0_to_450_set1_wo_offset_estimation.mat'
15 load 'Step_duty_0_to_450_set2_wo_offset_estimation.mat'
16 load 'Step_duty_0_to_600_set0_wo_offset_estimation.mat'
17 load 'Step_duty_0_to_600_set1_wo_offset_estimation.mat'
18 load 'Step_duty_0_to_600_set2_wo_offset_estimation.mat'
19
20 %Sampling period
21 Ts=1/5000;
22
23 %Creating iddata object for each dataset
24 data1e = iddata(chirp_0_p_out_f,chirp_0_v_in_f,Ts);
25 data2e = iddata(chirp_1_p_out_f,chirp_1_v_in_f,Ts);

```

```

26 data3e = iddata(ramp_1_p_out_f,ramp_1_v_in_f,Ts);
27 data4e = iddata(random_2_p_out_f,random_2_v_in_f,Ts);
28 data5e = iddata(random_0_p_out_f,random_0_v_in_f,Ts);
29 data6e = iddata(random_1_p_out_f,random_1_v_in_f,Ts);
30 data7e = iddata(sin_amp_mod_1_p_out_f,sin_amp_mod_1_v_in_f,Ts);
31 data8e = iddata(sin_amp_mod_0_p_out_f,sin_amp_mod_0_v_in_f,Ts);
32 data9e = iddata(step_300_set_0_p_out_f,step_300_set_0_v_in_f,Ts);
33 data10e = iddata(step_300_set_1_p_out_f,step_300_set_1_v_in_f,Ts);
34 data11e = iddata(step_300_set_2_p_out_f,step_300_set_2_v_in_f,Ts);
35 data12e = iddata(step_450_set_0_p_out_f,step_450_set_0_v_in_f,Ts);
36 data13e = iddata(step_450_set_1_p_out_f,step_450_set_1_v_in_f,Ts);
37 data14e = iddata(step_450_set_2_p_out_f,step_450_set_2_v_in_f,Ts);
38 data15e = iddata(step_600_set_0_p_out_f,step_600_set_0_v_in_f,Ts);
39 data16e = iddata(step_600_set_1_p_out_f,step_600_set_1_v_in_f,Ts);
40 data17e = iddata(step_600_set_2_p_out_f,step_600_set_2_v_in_f,Ts);
41
42 %Resampling estimation datasets to 100Hz
43 data1e = resample(data1e,1,50);
44 data2e = resample(data2e,1,50);
45 data3e = resample(data3e,1,50);
46 data4e = resample(data4e,1,50);
47 data5e = resample(data5e,1,50);
48 data6e = resample(data6e,1,50);
49 data7e = resample(data7e,1,50);
50 data8e = resample(data8e,1,50);
51 data9e = resample(data9e,1,50);
52 data10e = resample(data10e,1,50);
53 data11e = resample(data11e,1,50);
54 data12e = resample(data12e,1,50);
55 data13e = resample(data13e,1,50);
56 data14e = resample(data14e,1,50);
57 data15e = resample(data15e,1,50);
58 data16e = resample(data16e,1,50);
59 data17e = resample(data17e,1,50);
60
61 %Example of possible merged datasets
62 %Chirp, triangular, random, modulated sine and step signals
63 data1m=merge(data3e,data4e,data5e,data6e,data7e,...
64             data9e,data12e,data15e);
65 %Only random signals
66 data2m=merge(data4e,data5e,data6e);
67
68 %Load all validation datasets
69 load 'Chirp_duty_300_to_700_01mHz_1Hz_validation.mat'
70 load 'Random_uniform_duty_400_600_1Hz_validation.mat'
71 load 'Step_duty_0_to_300_set3_wo_offset_validation.mat'
72 load 'Step_duty_0_to_450_set3_wo_offset_validation.mat'
73 load 'Step_duty_0_to_600_set3_wo_offset_validation.mat'
74

```

```
75 %Creating iddata object for each dataset
76 data1v = iddata(chirp_2_p_out_f, chirp_2_v_in_f, Ts);
77 data2v = iddata(random_3_p_out_f, random_3_v_in_f, Ts);
78 data3v = iddata(ramp_1_p_out_f, ramp_1_v_in_f, Ts);
79 data4v = iddata(step_300_set_3_p_out_f, step_300_set_3_v_in_f, Ts);
80 data5v = iddata(step_450_set_3_p_out_f, step_450_set_3_v_in_f, Ts);
81 data6v = iddata(step_600_set_3_p_out_f, step_600_set_3_v_in_f, Ts);
82
83 %Resampling validation datasets to 100Hz
84 data1v = resample(data1v, 1, 50);
85 data2v = resample(data2v, 1, 50);
86 data3v = resample(data3v, 1, 50);
87 data4v = resample(data4v, 1, 50);
88 data5v = resample(data5v, 1, 50);
89 data6v = resample(data6v, 1, 50);
90
91 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ARX Models %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
92
93 %Imposing Simulation Focus as objective of the identification.
94 opt_m=arxOptions('Focus', 'Simulation');
95 %Choosing model order
96 Order=[5 5 1];
97 %Estimating ARX models
98 model_arx=arx(data2m, Order, opt_m);
99
100 %Defining initial conditions for validation
101 opt_c=compareOptions('InitialCondition', 'e');
102 figure,
103 %Comparing ARX model with validation datasets using Best Fit
104 compare(data1v, model_arx, inf, opt_c);
105
106 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% OE Models %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
107
108 %Choosing model order
109 Order=[6 6 1];
110 %Estimating OE models
111 model_oe=oe(data1e, Order);
112
113 %Defining initial conditions for validation
114 opt_c=compareOptions('InitialCondition', 'z');
115 figure,
116 %Comparing OE model with validation datasets using Best Fit
117 compare(data6v, model_oe, inf, opt_c);
118
119 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% NLARX Models %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
120
121 opt_m=nlarxOptions;
122 %Imposing Simulation Focus as objective of the identification.
123 opt_m.Focus='simulation';
```

```

124 %Turn on estimation progress display
125 opt_m.Display='on';
126 %Choosing model order
127 Order=[5 5 1];
128 %Estimating NLARX models with standard regressor
129 model_nlarx1 = nlarx(data2m,Order,'wavenet',opt_m);
130 model_nlarx2 = nlarx(data2m,Order,'sigmoidnet',opt_m);
131 model_nlarx3 = nlarx(data2m,Order,'linear',opt_m);
132
133 %Defining initial conditions for validation
134 opt_c=compareOptions('InitialCondition','e');
135 figure,
136 %Comparing NLARX model with validation datasets using Best Fit
137 compare(data1v,model_nlarx1,inf,opt_c)

```

A.6 Closed loop system bandwidth

```

1 %Utilized frequencies
2 frequency = [0.05,0.1,0.5,1,1.5,2,2.5,3];
3
4 %Measured peak-to-peak amplitudes
5 p_measured = [10,10,9.04,9.75,10.5,7.07,4.7,2.43];
6 %Input peak-to-peak amplitude
7 p_reference = 10;
8 %System gain computation and conversion in dB
9 module = 20*log10(p_measured/p_reference);
10
11 %Measured time delays
12 distance_pp = [0,0.5,0.35,0.27,0.26,0.28,0.26,0.24];
13 %Computation of corresponding phase delays
14 phase = -(360*distance_pp).*frequency;
15
16 %Choice of the query points where to interpolate
17 frequency_q = linspace(0.05,3,1000);
18 %Interpolation of module values
19 module_q = interp1(frequency,module,frequency_q,'spline');
20 %Interpolation of phase values
21 phase_q = interp1(frequency,phase,frequency_q,'spline');

```

Appendix B

Tables

B.1 Static characterization

Table B.1: Static characterization, upward sweep

Voltage imposed [V]	Voltage acquired [V]	Measured pressure [bar]	Current absorbed by the power supply [A]
2.4	2.377	0.001	0
2.6	2.558	1.795	6.25
3	2.998	2.245	6.28
3.4	3.362	3.072	6.34
3.8	3.724	8.205	6.73
4.2	4.093	22.539	9.47
4.6	4.453	26.214	9.93
5	4.995	31.994	10.72
5.4	5.357	36.223	11.27
5.8	5.810	43.494	12.52
6.2	6.081	45.778	12.77
6.6	6.533	51.831	13.54
7	6.894	56.745	14.42
7.4	7.346	63.106	15.42
7.8	7.617	67.250	16.18
8.2	8.069	73.579	17.24
8.6	8.519	79.898	18.22
9	8.970	85.767	19.34
9.4	9.332	89.761	19.97
9.8	9.693	93.567	20.59

10.2	10.144	98.413	21.53
10.6	10.505	101.546	22.19
11	10.868	105.273	22.88
11.4	11.229	108.808	23.65
11.8	11.679	112.534	24.52
12.2	12.039	115.610	25.28

Table B.2: Static characterization, downward sweep

Voltage imposed [V]	Voltage acquired [V]	Measured pressure [bar]	Current absorbed by the power supply [A]
12.2	12.039	115.610	25.28
11.8	11.678	108.986	24.37
11.4	11.228	103.363	23.16
11	10.866	98.826	22.12
10.6	10.412	93.602	21.21
10.2	9.961	88.292	20.19
9.8	9.601	83.084	19.25
9.4	9.239	79.148	18.47
9	8.968	76.598	18.02
8.6	8.518	71.291	16.96
8.2	8.068	66.887	16.07
7.8	7.707	62.702	15.25
7.4	7.256	58.603	14.45
7	6.896	55.870	13.87
6.6	6.445	52.283	13.13
6.2	6.084	47.995	12.22
5.8	5.632	45.346	11.72
5.4	5.361	42.426	11.22
5	4.909	37.718	10.22
4.6	4.458	32.266	9.26
4.2	4.098	26.937	8.33
3.8	3.737	23.560	7.64
3.4	3.286	19.749	6.84
3	2.927	16.034	6.21
2.6	2.566	3.018	3.75
2.4	2.296	0.069	0

Acronyms

AASHTO

American Association of State Highway and Transportation Officials

ABS

Anti-lock Braking System

ADAS

Advanced Driver Assistance System

AD-ECU

Autonomous Driving Electronic Control Unit

ADS

Automated Driving System

AEB

Autonomous Emergency Braking

ARX

Autoregressive Exogenous

BBW

Brake-By-Wire

BCU

Braking Control Unit

BEV

Battery Electric Vehicle

BRT

Braking Reaction Time

CAN

Controller Area Network

DAQ

Data Acquisition

DB

Data Base

DC

Direct Current

DDT

Dynamic Driving Task

DIN

Deutsches Institut für Normung

DLC

Data Length Code

ECU

Electronic Control Unit

E/H

Electric/Hydraulic

EHB

Electro-Hydraulic Brake

EMB

Electro-Mechanical Brake

GS

Gain Scheduling

GUI

Graphical User Interface

HCU

Hydraulic Control Unit

I/O

Input/Output

ISO

International Standardization Organization

LE

Little Endian

LPF

Low Pass Filter

LPV

Linear Parameter-Varying

LTI

Linear Time Invariant

LUT

LookUp Table

MATLAB

MATrix LABoratory

MCU

MicroController Unit

NI

National Instruments

NLARX

NonLinear Autoregressive Exogenous

ODD

Operational Design Domain

OE

Output Error

OEDR

Object and Event Detection and Response

PFE

Pedal Feel Emulator

PI

Proportional Integral

PID

Proportional Integral Derivative

PIL

Processor In the Loop

PLC

Programmable Logic Controller

PWM

Pulse-Width Modulation

RMSE

Root-Mean-Square Error

SAE

Society of Autonomous Engineers

SISO

Single Input Single Output

TF

Transfer Function

TMC

Tandem Master Cylinder

UNECE

Economic Commission for Europe of the United Nations

USB

Universal Serial Bus

VeGA

Veicolo a Guida Autonoma

Bibliography

- [1] Jo Kichun, Junsoo Kim, Dongchul Kim, Chulhoon Jang, and Myoungcho Sunwoo. «Development of Autonomous Car-Part II: A Case Study on the Implementation of an Autonomous Driving System Based on Distributed Architecture». In: *IEEE Transactions on Industrial Electronics*. Vol. 62.8. Aug. 2015, pp. 5119–5132 (cit. on p. 1).
- [2] Marc Green. «How Long Does It Take to Stop? Methodological Analysis of Driver Perception-Brake Times». In: *Transportation Human Factors*. Vol. 2(3). 2000, pp. 195–216 (cit. on p. 1).
- [3] Heikki Summala. «Brake reaction times and driver behavior analysis». In: *Transportation Human Factors*. Vol. 2(3). Sept. 2000, pp. 217–226 (cit. on p. 1).
- [4] American Association of State Highway and Transportation Officials. *A policy on geometric design of highways and streets*. Washington D.C., 2001 (cit. on pp. 1, 80).
- [5] *SAE J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. SAE International, Sept. 2016 (cit. on pp. 2, 3).
- [6] R. Hoseinnezhad and A. Bab-Hadiashar. «Efficient Antilock Braking by Direct Maximization of Tire–Road Frictions». In: *IEEE Transactions on industrial electronics*. Vol. 58.8. Aug. 2011, pp. 3593–3600 (cit. on p. 4).
- [7] Economic Commission for Europe of the United Nations. *Uniform provisions concerning the approval of vehicles of categories M, N and O with regard to braking*. 2016 (cit. on p. 4).
- [8] Konrad Reif. *Brakes, Brake Control and Driver Assistance System: Function, Regulation and Components*. Springer Vieweg Wiesbaden, 2014 (cit. on pp. 5, 6).
- [9] Giancarlo Genta and Lorenzo Morello. *The Automotive Chassis, Volume 1: Components Design*. Springer Nature Switzerland, 2020 (cit. on p. 5).

- [10] N. D'alfio, A. Morgando, and A. Sorniotti. «Electro-hydraulic Brake Systems: Design and Test through Hardware-in-the-loop Simulation». In: *Vehicle System Dynamics*. Vol. 44 Sup.1. 2006, pp. 378–392 (cit. on pp. 7, 9, 10).
- [11] Ren He and Daxing Huang. «Automatic Testing Method for Static Pressure Drive Performance of Electronic Hydraulic Brake System». In: *Thermal Science*. Vol. 24 N.3A. 2020, pp. 1529–1536 (cit. on pp. 7, 12).
- [12] Xiaoxiang Gong, Weiguo Ge, Juan Yan, Yiwei Zhang, and Xiangyu Gongye. «Review on the Development, Control Method and Application Prospect of Brake-by-Wire Actuator». In: *Actuators*. Vol. 9 N.1. 2020 (cit. on pp. 7, 9, 11, 12).
- [13] Dexiang Li, Cao Tan, Wenqing Ge, Jin Cui, Chaofan Gu, and Xuwen Chi. «Review of Brake-by-Wire System and Control Technology». In: *Actuators*. Vol. 11 N.3. 2022 (cit. on pp. 7, 8).
- [14] Shi Biaofei, Xiong Lu, and Yu Zhuoping. «Pressure Estimation of the Electro-Hydraulic Brake System Based on Signal Fusion». In: *Actuators*. Vol. 10 N.9. 2021 (cit. on p. 7).
- [15] Jian Zhao, Dongjian Song, Bing Zhu, Zhicheng Chen, and Yuhang Sun. «Nonlinear Backstepping Control of Electro-Hydraulic Brake System Based on Bond Graph Model». In: *IEEE Access*. Vol. 8. 2020, pp. 19100–19112 (cit. on pp. 9, 11).
- [16] Chengcheng Ma, Xuanyao Wang, Tongliang Xu, Jianye Wang, and Peng Feng. «Research on a novel electro-hydraulic brake system and pedal feel control strategy». In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*. Vol. 0 N.0. 2022 (cit. on p. 10).
- [17] Zhuoping Yu, Songyun Xu, Lu Xiong, and Wei Han. «An Integrated-Electro-Hydraulic Brake System for Active Safety». In: *SAE 2016 World Congress and Exhibition*. 2016 (cit. on p. 11).
- [18] Wei Han, Lu Xiong, and Zhuoping Yu. «Braking pressure control in electro-hydraulic brake system based on pressure estimation with nonlinearities and uncertainties». In: *Mechanical Systems and Signal Processing*. Vol. 131. 2019, pp. 703–727 (cit. on p. 11).
- [19] Lu Xiong, Wei Han, and Zhuoping Yu. «Adaptive sliding mode pressure control for an electro-hydraulic brake system via desired-state and integral-antiwindup compensation». In: *Mechatronics*. Vol. 68. 2020 (cit. on p. 11).
- [20] V.R. Mamilla and M.V. Mallikarjun. «Control of Electro-Mechanical Brake with Electronic Control Unit». In: *International Journal of Electronic Engineering Research*. Vol. 1 (3). 2009, pp. 195–200 (cit. on p. 12).

- [21] Raffaele Valentini. «Controlling car braking by means of electromechanical actuation on the pedal». Master's Degree Thesis. Torino, Italy: Politecnico di Torino, Dec. 2021 (cit. on p. 12).
- [22] Trailer & Caravan Parts Online. *Dexter E/H Brake Actuator 1600 PSI image*. [https://www.trailerpartsnz.com/product_info.php/dexter-electrichydraulic-brake-actuator-1600-psi-p-3764](https://www.trailerparts.nz.com/product_info.php/dexter-electrichydraulic-brake-actuator-1600-psi-p-3764). Online; accessed 06 November 2022. 2022 (cit. on p. 13).
- [23] Massimo Violante. *Model based software design lecture notes*. Politecnico di Torino, Mar. 2020 (cit. on pp. 14, 40).
- [24] *DX series electric-hydraulic actuator service manual*. Elkhart, Indiana, United States: Dexter Axle Company, Feb. 2021 (cit. on pp. 15, 16, 44).
- [25] Wikipedia. *Compliance (physiology) definition*. [https://en.wikipedia.org/wiki/Compliance_\(physiology\)](https://en.wikipedia.org/wiki/Compliance_(physiology)). Online; accessed 27 September 2022. 2021 (cit. on p. 15).
- [26] *SPxx-M10x1 Datasheet*. Vignola, Modena, Italy: Aviorace S.r.l., Sept. 2022 (cit. on p. 16).
- [27] *USB-6008/6009 features*. Austin, Texas, United States: National Instruments, Sept. 2022 (cit. on pp. 16, 21, 30).
- [28] AliExpress. *NI USB-6009 image*. <https://it.aliexpress.com/item/4000796671086.html>. Online; accessed 25 November 2022. 2022 (cit. on p. 17).
- [29] MathWorks. *Analog input block documentation*. https://it.mathworks.com/help/daq/analog_input.html. Online; accessed 27 September 2022. 2022 (cit. on p. 19).
- [30] Hanxing Chen and Jun Tian. «Research on the Controller Area Network». In: *2009 International Conference on Networking and Digital Society 2*. Guiyang, China, May 2009, pp. 251–254 (cit. on p. 25).
- [31] Steve Corrigan. *Introduction to the Controller Area Network (CAN)*. Tech. rep. SLOA101B. Dallas, Texas: Texas Instruments, May 2016 (cit. on p. 25).
- [32] MRS Electronic. *CAN I/O PLC Waterproof image*. <https://www.mrs-electronics.com/products/detail/can-io-plc-waterproof>. Online; accessed 19 November 2022. 2022 (cit. on p. 25).
- [33] *Datasheet CAN I/O and PLC WP 1.053*. Rottweil, Germany: MRS Electronic GmbH and Co. KG, Oct. 2022 (cit. on pp. 28, 50, 74).
- [34] Wikipedia. *Aliasing definition*. <https://en.wikipedia.org/wiki/Aliasing>. Online; accessed 26 November 2022. 2022 (cit. on p. 30).

- [35] Karl Johan Åström. *Control System Design lecture notes*. University of California, Santa Barbara, 2002 (cit. on pp. 30, 55).
- [36] Matthew Schwartz. *Statistical mechanics lecture notes*. Harvard University, Mar. 2021 (cit. on p. 35).
- [37] MathWorks. *System identification overview*. <https://it.mathworks.com/help/ident/gs/about-system-identification.html>. Online; accessed 14 October 2022. 2022 (cit. on p. 39).
- [38] MathWorks. *ARX Options*. <https://it.mathworks.com/help/ident/ref/arxoptions.html>. Online; accessed 14 October 2022. 2022 (cit. on p. 40).
- [39] Michele Taragna. *Estimation, filtering and system identification lecture notes*. Politecnico di Torino, Mar. 2020 (cit. on p. 40).
- [40] Scilab. *Discrete-time PID controller implementation*. <https://www.scilab.org/discrete-time-pid-controller-implementation>. Online; accessed 24 October 2022. 2022 (cit. on p. 45).
- [41] Carlo Novara. *Nonlinear control and aerospace applications lecture notes*. Politecnico di Torino, Mar. 2021 (cit. on p. 46).
- [42] Luca Bascetta. *Automatic control lecture notes*. Politecnico di Milano, 2018 (cit. on p. 52).
- [43] Alessandro De Luca. *Automatic control lecture notes*. Università di Roma “La Sapienza”, 2010 (cit. on p. 52).
- [44] Youbin Peng, Damir Vrancic, and Raymond Hanus. «Anti-windup, bumpless, and conditioned transfer techniques for PID controllers». In: *IEEE Control Systems Magazine*. Vol. 16(4). Aug. 1996, pp. 48–57 (cit. on p. 56).
- [45] C. Bohn and D.P. Atherton. «An analysis package comparing PID anti-windup strategies». In: *IEEE Control Systems Magazine*. Vol. 15.2. Apr. 1995, pp. 34–40 (cit. on p. 56).
- [46] Economic Commission for Europe of the United Nations. *Consolidated Resolution on the Construction of Vehicles (R.E.3)*. 2014 (cit. on p. 69).
- [47] Bylogix s.r.l. *VeGA, Autonomous Vehicle*. <https://www.bylogix.it/2020/07/14/vega-autonomous-vehicle/>. Online; accessed 15 October 2022. 2020 (cit. on p. 70).
- [48] Economic Commission for Europe of the United Nations. *Uniform provisions concerning the approval of passenger cars with regard to braking*. 2015 (cit. on p. 83).
- [49] Yucai Zhu and Zuhua Xu. «A Method of LPV Model Identification for Control». In: *IFAC Proceedings Volumes, Volume 41, Issue 2*. Seoul, Korea, July 2008, pp. 5018–5023 (cit. on p. 86).