

POLITECNICO DI TORINO

Master's Degree in MECHATRONIC ENGINEERING



Master's Degree Thesis

Robust Control of an Electronic Throttle Valve Actuator

Supervisors

Prof. Massimo SORLI

Ing. Filippo RIGOBON

Candidate

Daide NATALI

October 2022

Summary

This thesis aims to study different control techniques in order to design a robust control of an electronic throttle control system. In fact, in recent years, automotive industry requirements on emissions control, driveability and vehicle safety have become more stringent and, consequently, increasingly efficient engine control systems are required. On the other hand, the technological improvements of the control units allow the use of more complex control techniques.

The model-based control design is the methodology used to design and validate the control techniques considered in this work. Furthermore, to achieve robustness of the controllers, robust pole placement is used to design the different control system characteristic equations. The controllers were tested with MIL and HIL simulations. In particular, for HIL simulations, a test bench to simulate the real system in real-time was built.

A comparison study of the different control techniques was carried out using the results obtained in the previous validation phase. The compared control techniques were: the proportional-integral-derivative control, the integrative-proportional-derivative control, the discrete-time polynomial control, the linear state feedback control and the sliding mode control.

The analysis has highlighted that the PID control remains a suitable solution for the considered application, while, on the other hand, the IPD and SMC can improve the system in terms of transient and robustness performances.

Acknowledgements

Table of Contents

List of Tables	VII
List of Figures	VIII
Acronyms	XII
1 Introduction	1
1.1 Project overview	1
1.1.1 Digital twin	2
1.1.2 Robust actuator control	2
1.1.3 About Capgemini Engineering	3
1.2 Structure of the thesis	4
2 State-of-the-art	5
2.1 Electronic throttle control system	5
2.1.1 Hardware description	5
2.1.2 Throttle valve position control system	6
2.2 Ideal control system specification	7
2.2.1 Control system step response	8
2.2.2 Ideal control requirements	9
2.3 Current electronic throttle control technique	9
2.4 Motivation of the study	12
3 Modelling	13
3.1 Model-based control design	13
3.2 Models	17
3.2.1 Nonlinear model	17
3.2.2 Liner model	23
3.2.3 Reduced order linear model	24
3.2.4 Model parameterization and validation	26

4	Control techniques	30
4.1	Introduction	30
4.2	Control common features	30
4.2.1	Control design and validation procedure	30
4.2.2	Integrator anti-windup	34
4.2.3	Dither	34
4.3	Control design	36
4.3.1	PID	36
4.3.2	IPD	37
4.3.3	Discrete-time polynomial control	41
4.3.4	Linear state feedback control	46
4.3.5	Sliding mode control	51
5	Results	58
5.1	Introduction	58
5.2	HIL simulation details	58
5.2.1	Simulink test model	59
5.2.2	Experimental setup	60
5.3	Tests to assess the controllers performances	62
5.3.1	Step response	62
5.3.2	Drive cycle	63
5.3.3	Robustness test	64
5.3.4	Spring fault test	68
5.4	Obtained results	71
5.4.1	PID	71
5.4.2	IPD	76
5.4.3	Discrete-time polynomial control	82
5.4.4	Linear state feedback control with state observer	83
5.4.5	Boundary layer sliding mode control	89
5.5	Experimental results comparison	95
6	Conclusions	99
A	Manson’s gain rule	100
	Bibliography	101

List of Tables

3.1	Description of the model signals	18
3.2	Description of the model parameters	18
5.1	PID step response results	71
5.2	IPD step response results	77
5.3	RST step response results	82
5.4	LSF with state observer step response results	84
5.5	Boundary layer SMC step response results	89
5.6	Summary table of the obtained results	95
5.7	Summary table of the obtained results with point assigned	96
5.8	Control techniques scores	97

List of Figures

2.1	Throttle valve actuator	5
2.2	Throttle valve actuator components view	5
2.3	Throttle valve position control system	6
2.4	General control system step response	8
2.5	Proportional-integral-derivative control	10
2.6	PID with feed forward compensator	11
3.1	Graphical representation of the main steps of the model-based control design	15
3.2	Nonlinear model of the throttle valve actuator	18
3.3	Friction model angular speed-torque graph	19
3.4	Block scheme of the Pedersen and Dodds friction model	21
3.5	Block scheme of the throttle valve hard stops model	22
3.6	Block scheme of the return spring nonlinear model	22
3.7	Linear model of the throttle valve actuator	23
3.8	Simplified linear model of the throttle valve actuator	24
3.9	Reduced order linear model of the throttle valve actuator	25
3.10	Simplified reduced order linear model of the throttle valve actuator	25
3.11	Model parameterization scheme	27
3.12	Model validation with step response	28
3.13	Model validation with drive cycle	29
4.1	Model scheme of the closed-loop control system	31
4.2	Control design algorithm	33
4.3	An example of anti-windup applied to PI control	34
4.4	Control dither block diagram	34
4.5	System ramp response with and without dither control. Pulse amplitude equal to 2%	35
4.6	System ramp response with and without dither control. Pulse amplitude equal to 4%	36
4.7	PID with integrator anti-windup	37

4.8	IPD closed loop system with integrator anti-windup	38
4.9	Root locus of closed loop system using robust pole placement	40
4.10	Closed-loop system with RST block diagram	41
4.11	Root locus	44
4.12	Steady-state tracking error	44
4.13	Basic linear state feedback control block diagram	46
4.14	Linear state feedback control with state observer block diagram . .	47
4.15	Linear state feedback control with state observer and integrator for steady state error elimination block diagram	50
4.16	Linear state feedback control with state observer and integrator anti-windup block diagram	50
4.17	Double integrator plant	51
4.18	Basic sliding mode control scheme	51
4.19	Double integrator state trajectory	53
4.20	Boundary layer sliding mode control block scheme	55
5.1	Simulink test model	59
5.2	Experimental setup	60
5.3	Experimental setup block diagram	61
5.4	Step response parameters	62
5.5	Drive cycle signal	63
5.6	Drive cycle signals detail	64
5.7	Example of parametric variation for Monte Carlo analysis	66
5.8	Example of robustness test step response	67
5.9	Example of spring fault test response	69
5.10	Example of spring fault test duty cycle response	69
5.11	PID step response	71
5.12	PID step response duty cycle	72
5.13	PID drive cycle response	73
5.14	Detail of PID control system excited with sine waves at different frequency	73
5.15	Detail of the PID drive cycle response in small-signal operating mode	74
5.16	PID robustness test results	74
5.17	PID spring fault test response	75
5.18	PID spring fault test response duty cycle	76
5.19	IPD step response	76
5.20	IPD step response duty cycle	77
5.21	IPD drive cycle response	78
5.22	Detail of the IPD drive cycle response in small-signal operating mode	79
5.23	Detail of PID control system excited with sine waves at different frequency	79

5.24	IPD robustness test results	80
5.25	IPD spring fault test response	81
5.26	IPD spring fault test response duty cycle	81
5.27	RST step response	82
5.28	RST step response duty cycle	83
5.29	LSF with state observer step response	84
5.30	LSF step response duty cycle	85
5.31	LSF with state observer drive cycle response	85
5.32	Detail of the LSF with state observer drive cycle response in small-signal operating mode	86
5.33	Detail of the LSF control with state observer excited with sine waves at different frequency	86
5.34	LSF control with state observer robustness test results	87
5.35	LSF control with state observer spring fault test response	88
5.36	LSF control with state observer spring fault test response duty cycle	88
5.37	Boundary layer SMC step response	89
5.38	Boundary layer SMC step response duty cycle	90
5.39	Boundary layer SMC drive cycle response	91
5.40	Detail of the boundary layer SMC drive cycle response in small-signal operating mode	91
5.41	Detail of the boundary layer SMC control system excited with sine waves at different frequency	92
5.42	Boundary layer SMC robustness test results	92
5.43	Boundary layer SMC spring fault test response	93
5.44	Boundary layer SMC spring fault test response duty cycle	94
5.45	Difference between experimental and simulated step response of the different implemented control strategies	97

Acronyms

RAC

Robust actuator control

ETCS

Electronic throttle control system

MIL

Model-in-the-loop

HIL

Hardware-in-the-loop

PID

Proportional-integral-derivative

IPD

Integral-proportional-derivative

LSF

Linear state feedback

SMC

Sliding-mode control

DbW

Drive-by-Wire

ECU

Electronic control unit

PWM

Pulse width modulation

MBD

Model-based design

LTI

Linear-time invariant

Chapter 1

Introduction

1.1 Project overview

In recent years, requirements on emissions control, driveability and vehicle safety have become more stringent for several reasons including the environmental impact of motor vehicles and road safety. This has led to the replacement of traditional mechanical control systems with mechatronic control systems, e.g. throttle by wire, brake by wire and steer by wire. This technology is called Drive-by-Wire (DbW) and some of the advantages it brings are [1][2]:

1. **Weight reduction**, since DbW technology uses a lower number of mechanical moving parts compared to traditional mechanical control systems.
2. **Safety improvement**, DbW technology allows electronic safety systems to be added, e.g. electronic stability control, adaptive cruise control lane assist systems.
3. **Fuel efficiency and fewer carbon emissions**, electronic control systems make the motor more efficient for example by better controlling the opening of the throttle valve.

The thesis has been carried out in collaboration with the Capgemini Engineering S.p.A. company and concerns part of the Robust Actuator Control (RAC) project. The goal of the RAC project is the study of a formal control system development process for automotive Drive-by-Wire actuators. It is an iterative process which can be divided in two macro steps. The first phase can be summarized in plant modeling and model simplification, parameterization and validation. The second phase focuses on the control design for the considered system and it includes analysis and design of the controller based on the simplified model, simulation and

evaluation of closed-loop system performance, controller testing on the actual plant with hardware-in-the-loop. Throughout the development of the process, particular attention is paid to its flexibility, that is to the ability of the method to adapt to different types of electromechanical actuators.

The formal process was built step by step by studying an electronic throttle control (ETC) system. The method could be defined starting from any electromechanical actuator but the choice of ETC was made given the wide and diversified use of the system in the industrial field.

Although, the project consists of two phases, both fundamental to the achievement of the set objectives, only the second part of the project will be treated in details. However, a summary chapter of the first part will be provided for a complete understanding of the project and the process followed.

1.1.1 Digital twin

The first phase of the project is the construction of a digital twin of the actuator, a parameterized mathematical model of the system. The realized model represents a critical step as it will be a solid basis on which build the control logic.

The model must reflect the real dynamic behavior of the component, allow an easy re-calibration and simulate the variability of the parameters due to both the uncertainties of production and the aging of the components of the system. An important aspect of this phase is the elaboration of an accurate representation of the major non-linearities of the system. Consequentially, the developed digital twin is a non-linear model, particularly useful for estimating the performance of the control logic in model-in-the-loop simulation (MIL). Related to this, a linear model of the actuator was derived from the non-linear one for the purpose of designing and calibrating control strategies.

For the characterization and validation of the model, an actuator test procedure was developed.

1.1.2 Robust actuator control

The second step of the project is the design of a robust control logic for the actuator. This must be based on the model and consider all the critical issues of the system highlighted in the modeling phase. A high degree of control strength is required to ensure defined performance requirements and prevent the most common sources of failure.

During this phase, different control strategies were studied to understand what were the real advantages of more complex control logic compared to the current

state of the art. A comparative study of five different control strategies was made:

1. Proportional-integral-derivative control (PID);
2. Integral-proportional-derivative control (IPD);
3. Discrete-time polynomial control (RST);
4. Linear state feedback control (LSF);
5. Sliding-mode control (SMC);

The same process was followed for the design of the different controls. First, the controls were designed starting from the linear model of the actuator developed in the first part of the project. Subsequently, they were tested using the parameterized non-linear model. If the results obtained at this stage meet the control requirements, the controls are tested in real-time with the actual actuator.

The evaluation and comparison of performance was then assessed on the basis of step response, drive cycle response, robustness of the control action and the controller response in the event of a spring failure in the actuator operating phase.

1.1.3 About Capgemini Engineering

World leader in engineering and R&D services, Capgemini Engineering combines its broad industry knowledge and cutting-edge technologies in digital and software to support the convergence of the physical and digital worlds. Coupled with the capabilities of the rest of the Group, it helps clients to accelerate their journey towards Intelligent Industry. Capgemini Engineering has more than 55,000 engineer and scientist team members in over 30 countries across sectors including Aeronautics, Space, Defense, Naval, Automotive, Rail, Infrastructure & Transportation, Energy, Utilities & Chemicals, Life Sciences, Communications, Semiconductor & Electronics, Industrial & Consumer, Software & Internet.

Capgemini Engineering is an integral part of the Capgemini Group, a global leader in partnering with companies to transform and manage their business by harnessing the power of technology. The Group is guided every day by its purpose of unleashing human energy through technology for an inclusive and sustainable future. It is a responsible and diverse organization of over 340,000 team members in more than 50 countries. With its strong 55-year heritage and deep industry expertise, Capgemini is trusted by its clients to address the entire breadth of their business needs, from strategy and design to operations, fueled by the fast evolving and innovative world of cloud, data, AI, connectivity, software, digital engineering and platforms. The Group reported in 2021 global revenues of €18 billion.

1.2 Structure of the thesis

As mentioned in the previous sections, the main focus of the thesis is the study of a robust control technique for the intake valve and , in particular, the intake valve of an electronically controlled industrial diesel engine. Chapter 2 provides a general description of the actuator being studied and the current state of the art for its control. Since the design process is strictly based on the system models, Chapter 3 presents the design method followed and the models developed in the first part of the project. The use of models within the control design process is also explained. Chapter 4 presents the control techniques considered in this research and provides a theoretical explanation of their design. Chapter 5 focuses of the work carried out, showing the results obtained and comparing the control techniques considered. The last chapter, Chapter 6, contains a summary of the research and some ideas for future developments.

Chapter 2

State-of-the-art

2.1 Electronic throttle control system

2.1.1 Hardware description

In the automotive industry, one of the fruitful technologies that have emerged from the increasing regulations in terms of fuel economy, emission control, drivability and safety is the DbW technology that creates the electronic throttle control system (ETCS).

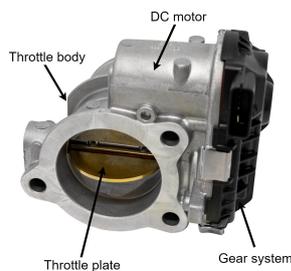


Figure 2.1: Throttle valve actuator

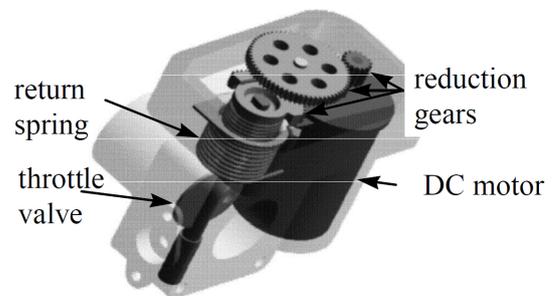


Figure 2.2: Throttle valve actuator components view

As shown in Figure 2.1 and Figure 2.2, the system is composed of a throttle plate connected with a pre-loaded spring and is driven by an electronic-controlled DC motor to regulate airflow in the intake manifold. The return spring is a fail-safe device, which ensures a safety recovery position for the throttle when no driving

torque is generated by the DC motor. The recovery position is denoted as limp home (LH) position. The throttle plate shaft is connected to the DC motor shaft through a pair of gear wheels. The transmission is used to amplify the torque delivered by the DC motor. The throttle plate position is measured by a potentiometer type sensor attached to the plate [3].

2.1.2 Throttle valve position control system

The aim of the electronic throttle valve position control system is to control the position (i.e., the angle) of the plate inside the throttle valve. The controller receives the desired angular position of the throttle valve from the engine control unit (ECU) and the feedback angular position from the potentiometer; then it compares the two input values and produces a pulse-width modulated output signal (PWM) for the DC motor to generate the amount of torque needed to bring the throttle valve to the target position [4].

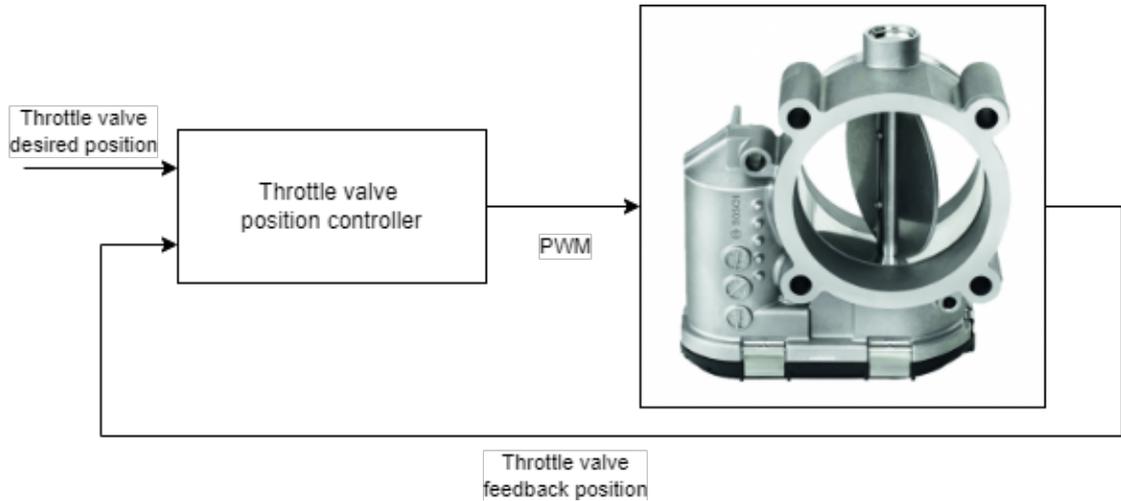


Figure 2.3: Throttle valve position control system

In modern vehicles, the throttle plate's desired angle is computed by the ECU. The calculation seeks to optimize the stoichiometric ratio in the combustion chamber, minimizing emissions and maximizing energy efficiency. It takes into account several factors such as the accelerator pedal position, engine speed and cruise control command. To achieve high energy performance the throttle valve needs to be moved very quickly during transition from the stoichiometric to the lean mixtures. Therefore, a throttle actuator with faster response and more precise control is demanded [3].

The research will not address the question of how the ECU calculates the reference angle. The main objective of this study is the design of a robust control that meets the closed loop system response requirements.

2.2 Ideal control system specification

Multiple system non-linearities (e.g. pre-loaded spring characteristic and friction phenomena in moving parts) and application constraints make throttle position control a complex problem. The dynamics of the desired closed loop system response has to be chosen properly. In fact, a dynamic with too much lag can impact the engine emissions, since it doesn't track the desired position for long period of time. On the other hand, too fast closed loop system response can wear down quickly the actuator gear system and DC motor.

In addition, very high robustness, that is the ability of the closed loop system to maintain almost the same performance even if the system parameters change, must be achieved by keeping project costs and delivery times low [4].

2.2.1 Control system step response

The output response of the control system is evaluated by stimulating the system with a step signal. An example of step response is shown in 2.4.

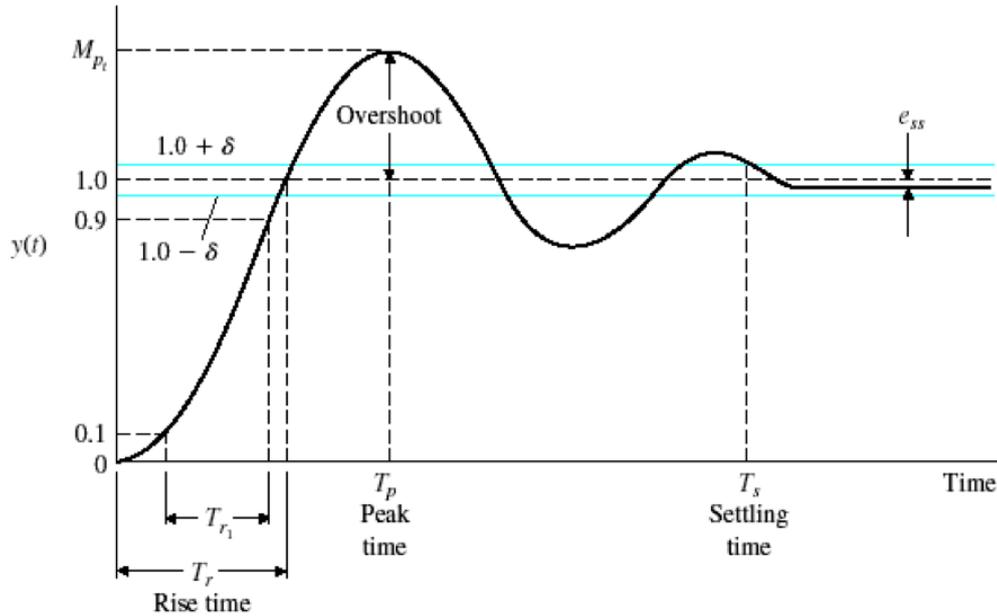


Figure 2.4: General control system step response

Specifically, the parameters evaluated in this project are:

1. **Settling Time T_s** , that is the time elapsed from the application of an ideal instantaneous step input to the time at which the system output has entered and remained within a specified error band. Typically the considered error band is 5%, 2% or 1%.
2. **Percentage Overshoot**, that is the maximum value M_{pt} minus the step value divided by the step value.
3. **Steady State Error e_{ss}** , that is the deviation of the output of control system from desired response during steady state.

2.2.2 Ideal control requirements

Engineering practice has shown that an electronic throttle control system should meet the following requirements [5] [4]:

1. **Settling Time** of the position control system step response should be less than 130 ms for any operating point and for any reference step change. This requirement may vary depending on whether it is preferred better performance or less wear of the motor and gear system of the actuator.
2. **The steady state error** should be less than 0.1 degree.
3. **No overshoot** of the step response is allowed; this requirement is particularly important for the large signal operating mode, in order to avoid hitting the mechanical stops at the extreme positions. Large signal operating mode refers to large variations in the input signal of the system.
4. **Low level of perturbations** in the throttle position signal and the commanded signal is required, in order to avoid excessive potentiometer and transmission wear, and motor losses.
5. **Robustness** of the control system with respect to variations of process parameters is required, which can be caused by production deviations, variations of external conditions (e.g., temperature) and aging.
6. **Safety in case of return spring fail** to allow low-performance engine operation instead of a engine shutdown or an unwanted acceleration.

2.3 Current electronic throttle technique

To achieve the performance objectives listed in the previous section, many studies have been carried out, both in the industrial and academic fields. Therefore, the solutions found are many and only what had been considered most relevant for the current study will be mentioned. In particular, the state of the art will be provided, for the application considered, currently adopted by the automotive industry, i.e. the PID with feed forward compensator.

The electronic throttle control system has essentially two operating mode, a large-signal and a small-signal ones. The difference between the two operating conditions depends basically on the height of the step input to the system. For large-signal operating mode the electronic throttle can be controlled by the well-known proportional-integral-derivative (PID) controller. However, its performance for the small-signal operating mode is affected by the friction and limp-home nonlinearities. For this reason, a feed forward compensator is usually added to

counteract the coil spring torque and to reduce the transient periods that would be too long, relying only on the integral action of the PID. In addition, a control dither is used to counteract the effect of the static friction [4]. This traditional control technique and its variants are the most used for this application and they are tuned by a trial and error or Zeigler-Nichols procedure. For a better understanding of the above mentioned control technique and control strategies considered in this project, a brief theoretical explanation will be provided in the following sections.

PID

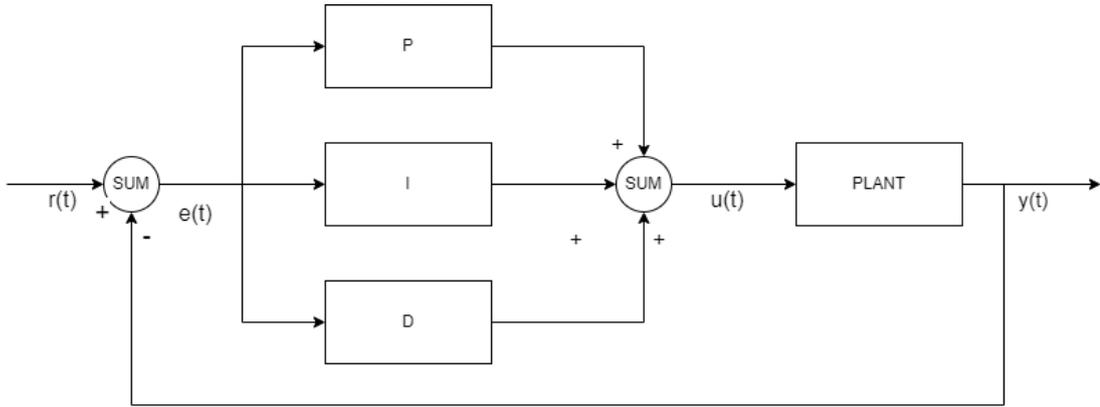


Figure 2.5: Proportional-integral-derivative control

A scheme of the PID controller is shown in Figure 2.5, it is responsible to ensuring that the process remains as close as possible to the desired value as possible regardless of various disruptions. It determines how much and how quickly corrections are applied by using varying amount of proportional (P), integral (I) and derivative (D) actions. Each block contributes a unique signal that is added together to create the controller output signal.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (2.1)$$

- $\mathbf{P} = K_p e(t)$, creates an output signal proportional to the magnitude of the error signal. However, the closer you get to the set point the less it pushes. Eventually, the process just runs continuously close to the set point but not quite there.
- $\mathbf{I} = K_i \int_0^t e(\tau) d\tau$, creates an output proportional to the duration in magnitude of the error signal. The longer the error and the greater the amount the integral output. As long as an error exists, integral action will continue.

- $D = K_d \frac{de(t)}{dt}$, creates an output signal proportional to the rate of change of the error signal. The faster the error changes, the larger the derivative output. Derivative control looks ahead to see what the error will be in the future and contributes to the controller output accordingly.

PID with feed forward compensator

A feed forward compensator aims to predict changes in system behavior and to compensate them adding to the plant a contribute to the control command input. In contrast with the feedback control strategies, that are error driven, the feed forward compensator depends only on the desired input of the control system. A block diagram of the PID with feed forward compensator is shown in Figure 2.6.

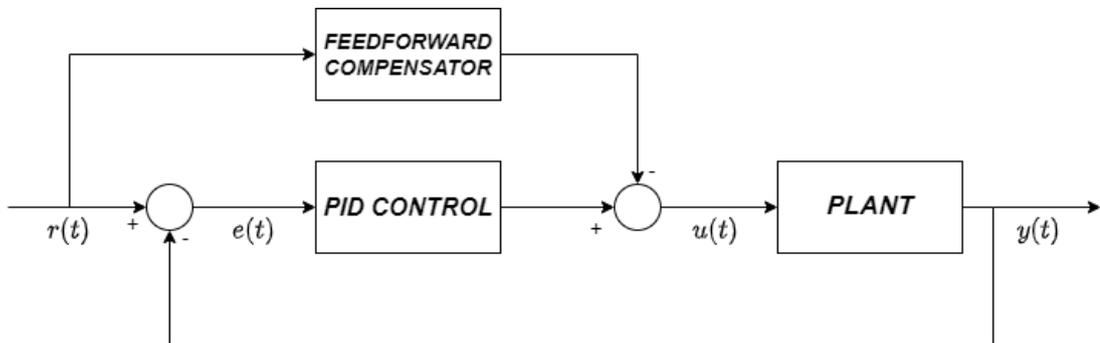


Figure 2.6: PID with feed forward compensator

The feed forward control action is closely related to the accuracy of the plant modelling phase of model-based control design. The more accurate the plant model is, with respect to the real system behavior, the more the feed forward control operates with faster response and less error. Although, in most cases models are not precise enough so a combination of feed forward and feedback control represents the optimal compromise.

In throttle valve position control system, the feed forward control is typically used to compensate the nonlinear effects of return spring and friction.

Since it has been already well studied in literature, in this study the PID control with feed forward compensator is not implemented while the difference between the simplest control strategy (PID) and the complex ones is enhanced.

2.4 Motivation of the study

Thanks to the continuous technological improvements, modern microcontrollers allow to deploy complex control strategies without increasing hardware costs. Therefore, the present work intends to study the control techniques used in vehicle power trains to control throttle valves position and verify any performance improvements due to the use of complex control techniques that the new microcontrollers available on the market allow to realize.

Chapter 3

Modelling

3.1 Model-based control design

Model-based design (MBD) is a mathematical and visual method to integrate all the development phases of a project into a continuous design cycle. Complex systems can be modeled by representing them in each subsystem and in the existing relationship between subsystems and the environment. These models have many applications in the design process, e.g. system simulation, stability analysis, and control algorithm design [6].

In a conventional development, a system engineer will typically define the entire system specification and submit it to software engineers as a design document, giving them the responsibility of turning those concepts into a fully functional solution. The biggest issue with this strategy is that the ideas that the system engineer presents in the specification document frequently diverge greatly from the software that is actually implemented. Since system engineers design and simulate the model using a model-based design tool and present the ideas behind the solution as a functioning model of the system, the model-based design (MBD) method solves this issue. Some highly developed model-based design tools provide cutting-edge capabilities for the automatic generation of source code that would mimic the simulated system, thereby accelerating time to market and lowering development costs.

Model-based design steps

The main steps in model-based design approach are:

1. ***Plant modelling***

Data-driven or first-principles-based plant modelling are both possible.

System identification is one of the approaches used in data-driven plant modelling. By obtaining and processing raw data from a real-world system and using a mathematical procedure to identify a mathematical model, system identification identifies the plant model. The selected model can be used to conduct various types of simulations and analyses before being utilised to create a model-based controller.

In first-principles based modelling, the model block diagram is made up starting from the established differential algebraic equations which regulate plant dynamics. Physical modelling is a sort of first-principles based modelling where a model is made up of connected blocks that represent the physical components of the actual plant.

2. ***Controller analysis and design***

The dynamic properties of the plant model are identified using the mathematical model created in step 1. These qualities can then be used to design a controller.

3. ***Offline simulation and real-time simulation***

In this phase the performance of the developed controller are evaluated. This is done by simulating the controller with the LTI (Linear Time-Invariant) model, or the non-linear model of the plant. In this way, a time response of the system is obtained and an analysis of requirements, specifications and possible modeling errors can be made to quickly find any errors or inaccuracies.

The time response of the dynamic system to complex, time-varying inputs is investigated. This is done by simulating a simple LTI (Linear Time-Invariant) model, or by simulating a non-linear model of the plant with the controller. Simulation allows specification, requirements, and modeling errors to be found immediately, rather than later in the design effort. By automatically creating code for the controller created in step 2, real-time simulation is possible. This code can be installed on a specialised real-time prototyping computer, which can run it and manage how the plant operates.

4. ***Deployment***

This is best accomplished through code generation from the controller created

in step 2. Since it is unlikely that the controller will perform as well in practise as it did in simulation, an iterative debugging procedure is used, which involves examining data from the real target and changing the controller model. These iterative processes can all be carried out using model-based design tools in a single visual environment.

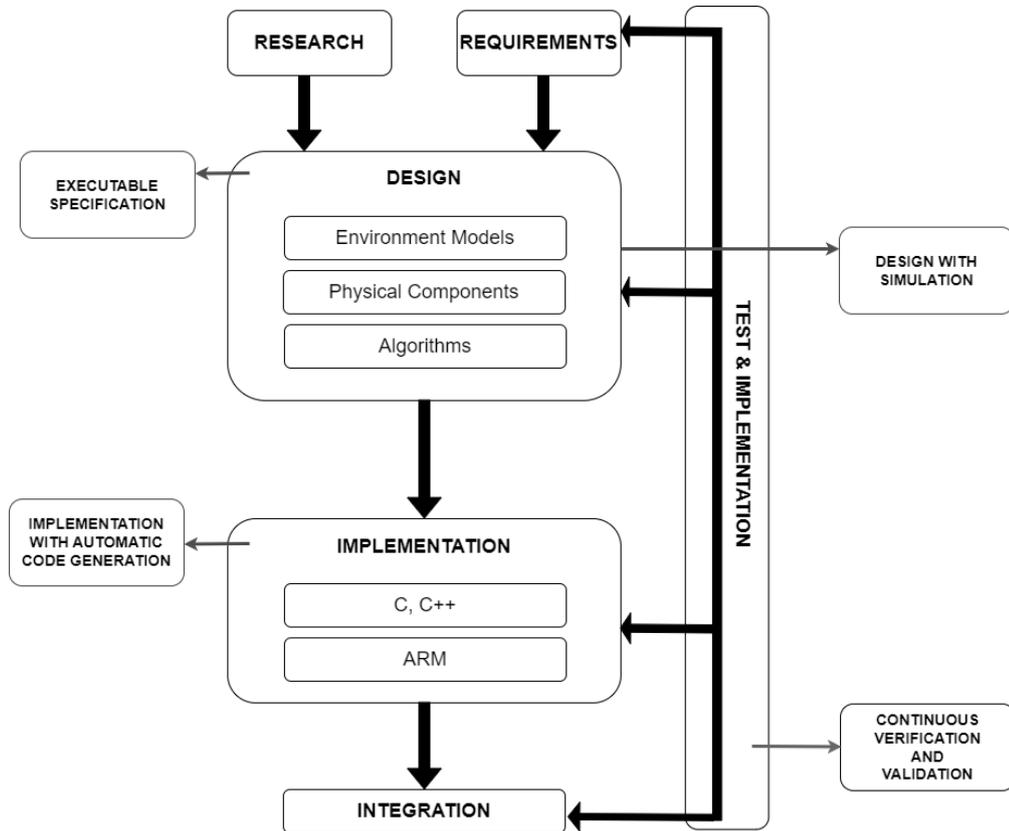


Figure 3.1: Graphical representation of the main steps of the model-based control design

Advantages

Model-based design has some benefits over the conventional method, including:

1. Model-based design offers a shared design environment that makes it easier for different (development) groups to communicate, analyse data, and verify systems.
2. When the time and expense effect of system modification are at their lowest, engineers can find and fix mistakes early in the design process.

3. Makes the found solutions more flexible and re-usable and allows for easier future project improvement.
4. Graphical modeling tools give a graphical modelling environment that is very general and uniform and reduce the complexity of model designs. They also make it easier for engineers to move the model between design stages and to visualise the overall system.
5. It is easier to evaluate the robustness of a control system since it is simple to observe the deviation of the control system response in case of disturbances on the plant and parameters variation that can be simulated.

Disadvantages

Model-based control design has also some disadvantage that is worth mentioning:

1. Often the time it takes to transfer processors to ecosystems can outweigh the temporal value it offers in the simpler lab based implementations.
2. The specific tools needed for the model-based control design are very expensive and they could contain errors due to automatic generation of the software that can only be detect during the testing phase.
3. Model-based design have to be modify and adapted according to each usage application.
4. The model is a partial representation of the reality and since some real elements are difficult to be simulated, the real behavior of the system may not match with the expected one.

3.2 Models

For the design of control strategies a model-based control design process was followed, as described in the previous Section 3.1. For this reason, before starting to discuss control techniques, it is necessary to show the results obtained in the first phase of the RAC project, that is, the phase of plant modeling. In particular, the nonlinear, linear and reduced order linear system models will be shown and a detailed explanation of their use in the control design process will be provided.

All block models were derived based on the equilibrium equations of the different subsystems of the plant. As explained in Section 1.1.1, only the results, and not the process that led to them, obtained at this stage will be analysed in the following sections.

3.2.1 Nonlinear model

The nonlinear model aims to predict the performance of control strategies as accurately as possible before they are tested experimentally. In order to obtain a good model accuracy combined with a good simulation speed of the same, only some of the nonlinearities studied during the analysis phase of the system have actually been considered and modeled:

- Static friction
- Coulomb friction
- Throttle valve hard stop
- Return spring preload

In Figure 3.2 below is shown the block diagram of the model.

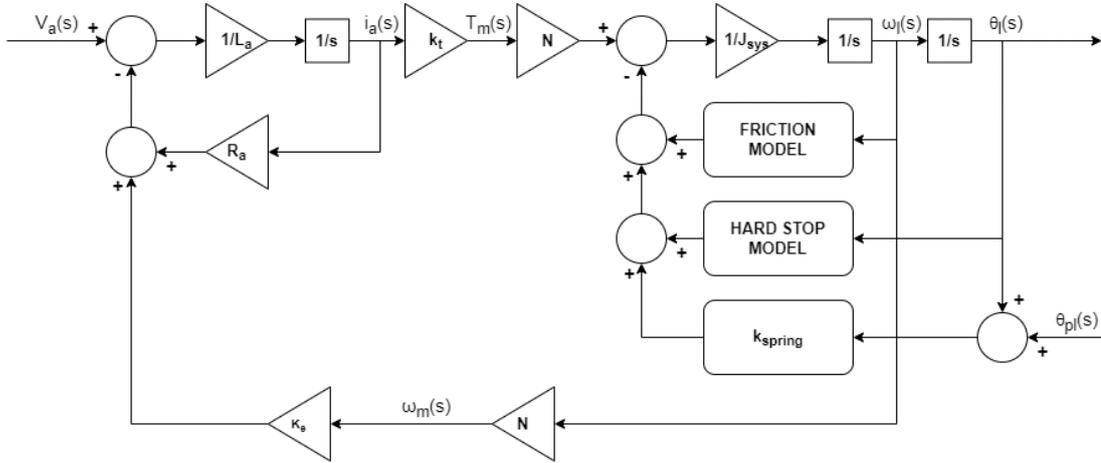


Figure 3.2: Nonlinear model of the throttle valve actuator

Where:

Signal	Description
$V_a(s)$	<i>DC motor armature voltage</i>
$i_a(s)$	<i>DC motor armature current</i>
$T_m(s)$	<i>Torque delivered by DC motor</i>
$\omega_l(s)$	<i>Throttle valve angular speed</i>
$\theta_l(s)$	<i>Throttle valve angular position</i>
$\omega_m(s)$	<i>DC motor angular speed</i>

Table 3.1: Description of the model signals

Parameter	Description
L_a	<i>DC motor armature inductance</i>
R_a	<i>DC motor armature resistance</i>
k_t	<i>DC motor torque constant</i>
k_e	<i>DC motor speed constant</i>
N	<i>Gear ratio</i>
J_{sys}	<i>System inertia</i>
k_{spring}	<i>Spring stiffness</i>
$k_{kinetic}$	<i>Viscous friction coefficient</i>

Table 3.2: Description of the model parameters

Friction model

To model the phenomenon of friction, present on the moving parts of the system, several models were considered. At the end of the analysis, it was decided to adopt the friction model of Pedersen and Dodds [4] because it was considered the most suitable to represent the phenomenon even at speeds close to zero.

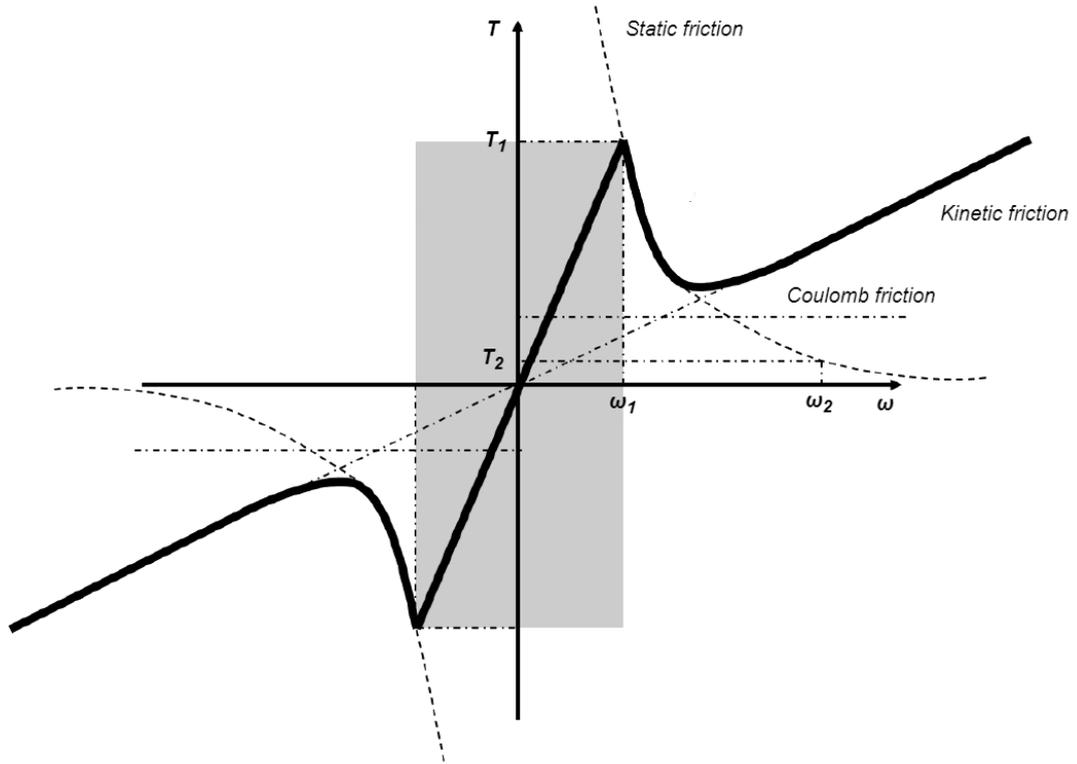


Figure 3.3: Friction model angular speed-torque graph

As shown in Figure 3.3, the friction torque always opposes the body's motion and, in this case, is modelled as the sum of several components that represent different aspects of the phenomenon depending on angular velocity:

1. **Static friction** T_s , not negligible for speeds close to zero but less relevant for higher speeds. To mimic the shape of the traditional friction model, the static friction component has been modelled around a rectangular hyperbola function. The advantage of using such a function is the reduced number of parameters to identify.

$$T_s = \frac{A}{\omega + |B|\text{sign}(\omega)} \quad (3.1)$$

$$A = T_1(B + \omega_1), B = \frac{T_1\omega_1 - T_2\omega_2}{\omega_2 - \omega_1}$$

To further simplify the parameterization, two parameters ω_1 and ω_2 were set initially to a constant value.

2. **Coulomb friction** T_c , which is a constant that takes into account the uniformity of the surfaces in contact.

$$T_c = \text{sign}(\omega)k_c \quad (3.2)$$

3. **Kinetic friction** T_k , which is directly proportional to the relative rotation speed among adjacent objects.

$$T_k = \omega * k_{kinetic} \quad (3.3)$$

To obtain the total friction torque T_{tot} , the sum of the three components has to be multiplied by a parameter that aims to make the transition from positive to negative velocities linear.

$$T_{tot} = (T_s + T_c + T_k)y_t \quad (3.4)$$

Where:

$$y_t = \begin{cases} \frac{|\omega|}{\omega_1}, & |\omega| \leq \omega_1 \\ 1, & |\omega| > \omega_1 \end{cases} \quad (3.5)$$

A block diagram of the model thus obtained is shown in Figure 3.4.

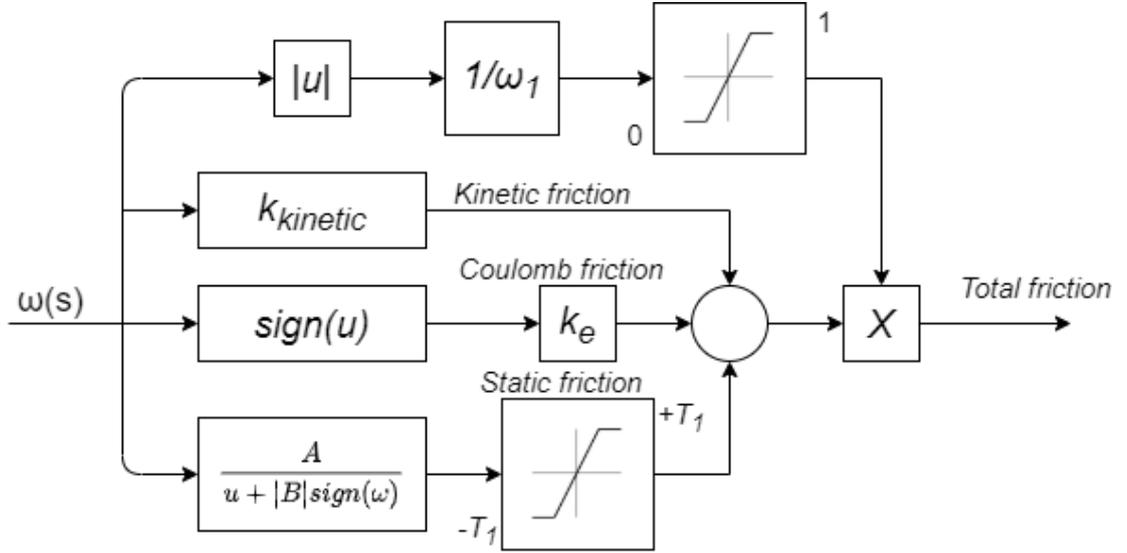


Figure 3.4: Block scheme of the Pedersen and Dodds friction model

Hard stops model

Hard stops are mechanical position constraints of the throttle valve. The model behaves like a dead-band, characterized by a very high gain at the extremes.

$$T_{hardstop} = \begin{cases} 0, & \theta_{l,min} \leq \theta_l \leq \theta_{l,max} \\ k_{hardstop}(\theta_l - \theta_{l,max}), & \theta_l > \theta_{l,max} \\ k_{hardstop}(\theta_l - \theta_{l,min}), & \theta_l < \theta_{l,min} \end{cases} \quad (3.6)$$

As can be seen from the Equation 3.6, the model depends on the angular position of the valve and includes three areas of interest. When the throttle valve is in a position between the upper and lower limit the hard stop torque $T_{hardstop}$ is zero. Instead, when the lower or upper position limit is exceeded, a high gain occurs $k_{hardstop}$ which produces a very high hard stop torque $T_{hardstop}$ that is opposed to that of the DC motor driving the throttle valve. So, in case of overshoot the position will straddle the lowest and maximum position, but only by a little margin if the hard stop gain $k_{hardstop}$ is sufficiently high.

A block diagram of the hard stop model is shown in the above Figure 3.5.

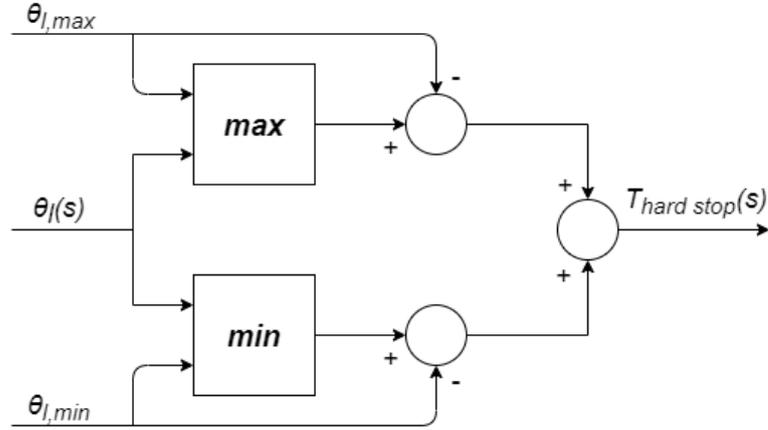


Figure 3.5: Block scheme of the throttle valve hard stops model

Return spring nonlinear model

To ensure a return to the limp-home position of the throttle plate (i.e. completely open since the considered system is an ETC for diesel application), the return spring of the actuator is preloaded. This means that the equilibrium point of the torques is not exactly on the limit position of the plate, but is slightly moved beyond the hard stop. Then at zero degree angular position the spring torque is different from zero. For this reason, the preload introduces a nonlinearity in the spring model.

$$T_{spring} = k_{spring}(\theta_l + \theta_{pl}) \quad (3.7)$$

A block diagram of the return spring nonlinear model is shown in Figure 3.6, where θ_l is the feedback angular position measured by the potentiometer, θ_{pl} is the preload expressed in terms of angular position and k_{spring} is the spring stiffness.

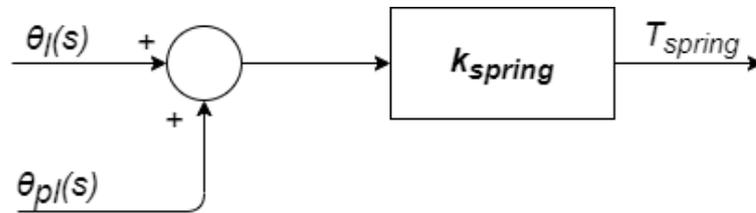


Figure 3.6: Block scheme of the return spring nonlinear model

3.2.2 Linear model

The linear model is a linearized version of the nonlinear model seen above (3.2.1). In particular, the hard stops of the butterfly valve and the preload of the spring are neglected, and the friction model, that in this case will have only the viscous component, is linearized.

A block diagram of the model is shown in Figure 3.7, where the signals and parameters are described, respectively, in Table 3.1 and in Table 3.2.

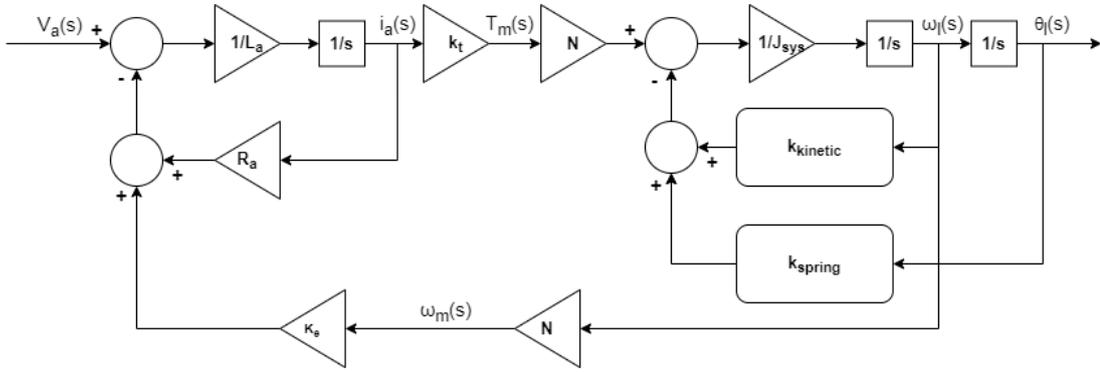


Figure 3.7: Linear model of the throttle valve actuator

Unlike the nonlinear model, the transfer function of the system can be derived for the linear model. Then, this will be used to design control techniques. The third order plant transfer function was obtained using the Manson's gain rule (Appendix A).

$$G_{plant}(s) = \frac{\frac{k_t N}{J_{sys} L_a}}{s^3 + s^2 \frac{(k_{kinetic} L_a + J_{sys} R_a)}{J_{sys} L_a} + s \frac{(k_{spring} L_a + k_{kinetic} R_a + k_t N^2 k_e)}{J_{sys} L_a} + \frac{k_{spring} R_a}{J_{sys}} L_a} \quad (3.8)$$

In order to avoid a lengthy argument in the next chapters the third order transfer function of the system will be represented in a simplified way (3.8):

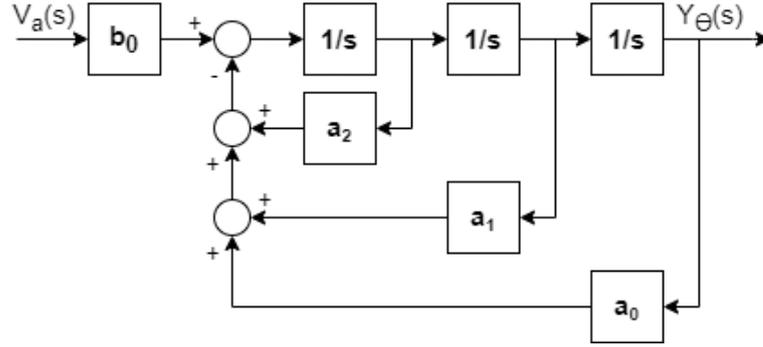


Figure 3.8: Simplified linear model of the throttle valve actuator

$$G_{plant}(s) = \frac{b_0}{s^3 + a_2s^2 + a_1s + a_0} \quad (3.9)$$

Where:

$$b_0 = \frac{k_t N}{J_{sys} L_a} \quad (3.10)$$

$$a_2 = \frac{k_{kinetic} L_a + J_{sys} R_a}{J_{sys} L_a} \quad (3.11)$$

$$a_1 = \frac{k_{spring} L_a + k_{kinetic} R_a + k_t N^2 k_e}{J_{sys} L_a} \quad (3.12)$$

$$a_0 = \frac{k_{spring} R_a}{J_{sys} L_a} \quad (3.13)$$

3.2.3 Reduced order linear model

The linear model of the third-order butterfly valve can be reduced to a second-order model by eliminating inductance. This is possible because the time constant is relatively small compared to the required control cycle adjustment time, which is set to the order of one hundredth of a second [4].

$$\tau_e = \frac{L_a}{R_a} \quad (3.14)$$

It should be noted that the presence of the feedback loop, introduced by the electromotive force, indicates that the electric time constant will not have exactly

the same value as that calculated from resistance and inductance, but will be of the same order.

A block diagram of the model is shown in Figure 3.9, where the signals and parameters are described, respectively, in Table 3.1 and in Table 3.2.

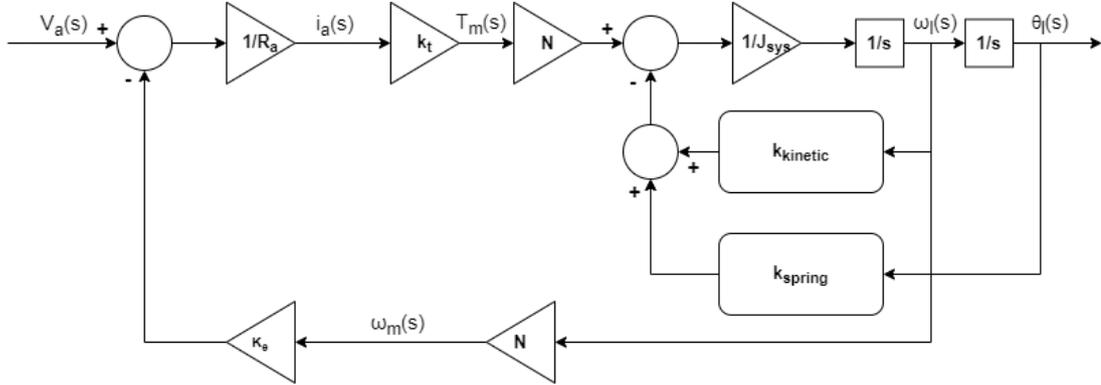


Figure 3.9: Reduced order linear model of the throttle valve actuator

$$G_{plant}(s) = \frac{\frac{k_t N}{J_{sys} R_a}}{s^2 + \frac{k_{kinetic} + \frac{k_t N^2 k_e}{R_a}}{J_{sys} s} + \frac{k_{spring}}{J_{sys}}} \quad (3.15)$$

In this way the model of the actuator will certainly be less accurate with respect to the third order model but, on the other hand, this reduce the design complexity and cost of control action.

In order to avoid a lengthy argument in the next chapters the second order transfer function of the system will be represented in a simplified way (3.10):

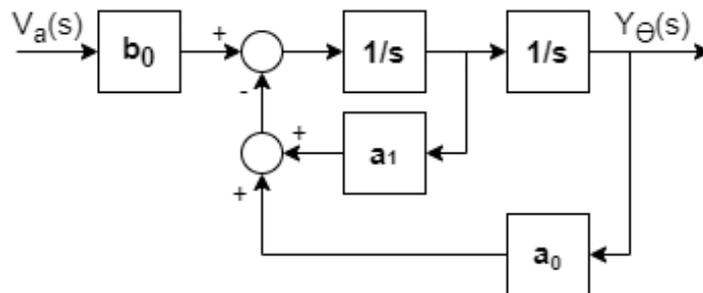


Figure 3.10: Simplified reduced order linear model of the throttle valve actuator

$$G_{plant}(s) = \frac{b_0}{s^2 + a_1s + a_0} \quad (3.16)$$

Where:

$$b_0 = \frac{k_t N}{J_{sys} R_a} \quad (3.17)$$

$$a_1 = \frac{k_{kinetic} + \frac{k_t N^2 k_e}{R_a}}{J_{sys}} \quad (3.18)$$

$$a_0 = \frac{k_{spring}}{J_{sys}} \quad (3.19)$$

3.2.4 Model parameterization and validation

Fundamental for the success of the model-based control design process are the parameterization and validation phases of the models. In these phases, in fact, the models are populated with the estimated values of the variable that compose them and their behavior, with respect to the real system, is estimated. The closer the model approaches the actuator's real behavior, the more precise the control design will be, resulting in better control action performance.

The three phases of modeling, parameterization and validation are iterative. This is because in the modeling phase the effect of some aspects of the real system can be underestimated or neglected. This, however, is only detected in the parameterization and validation phases where the expected results are not obtained. In this study this was experimented with the friction model. This, in fact, was reshaped because it was considered not representative enough of the real behavior of the system.

Model parameterization

In this step the model parameters are estimated. Two tools were used to parameterize the model.

- The following parameters were derived from the **actuator documentation**:
 1. Gear ratio
 2. DC motor armature resistance
 3. DC motor armature inductance
 4. Lower and upper limit hard stop positions

- The remaining parameters were estimated with **experimental tests**:
 1. Motor torque and speed constants
 2. System inertia
 3. Friction parameters
 4. Spring parameters

The objective of the experimental tests is to estimate the parameters of the nonlinear model by exciting the real system and the actuator model with the same inputs and comparing the output results. In particular, we want that the difference between the output signals of the actuator and the output signals of the simulated system tends to zero. A diagram of the described process is shown in Figure 3.11:

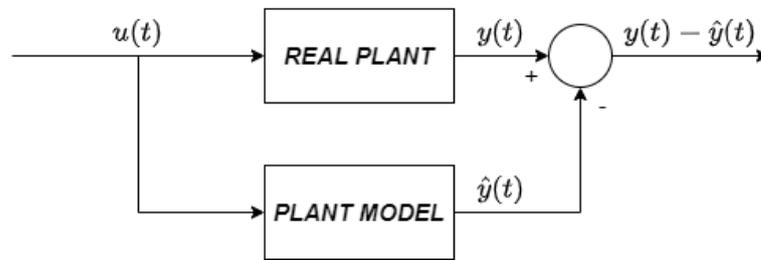


Figure 3.11: Model parameterization scheme

Where:

- $\mathbf{u}(t)$ is the reference position expressed in degree.
- $\mathbf{y}(t)$ is the real plant output vector:

$$\mathbf{y}(t) = \begin{bmatrix} i(t) \\ \theta(t) \end{bmatrix} \quad (3.20)$$

$i(t)$ is the DC motor armature current expressed in ampere and $\theta(t)$ is the throttle valve angular position expressed in degree.

- $\hat{\mathbf{y}}(t)$ is the simulated plant model output vector:

$$\hat{\mathbf{y}}(t) = \begin{bmatrix} \hat{i}(t) \\ \hat{\theta}(t) \end{bmatrix} \quad (3.21)$$

$\hat{i}(t)$ is the estimated DC motor armature current expressed in ampere and $\hat{\theta}(t)$ is the estimated throttle valve angular position expressed in degree.

- $y(t) - \hat{y}(t)$ is the difference between real and simulated plant outputs.

Parameter estimation is made by searching for the value of the parameters for which the output function $y(t) - \hat{y}(t)$ reaches a local minimum. The tool used to search for the local minimum of the output function from the scheme shown in Figure 3.11 is the MATLAB Simulink Parameter Estimator. To have a more accurate estimation of the parameters, these were not all estimated at the same time but a test procedure was followed aimed at stimulating only individual subsystems of the plant. In this way, it was possible to gradually derive the parameters of the model, reducing the computation time and increasing the accuracy of the estimate. To perform the tests it was necessary to build a test bench (shown in Section 5.2.2) and control the throttle valve in position. A low-performance PID is designed for throttle valve position control.

The tests performed on the throttle valve actuator will not be deepened because they are not topics of this study.

Model validation

Once acceptable results were obtained in the parameterization phase, the model was validated using signals different from those used for parameter estimation. In Figure 3.12 the step response of the valve in the two cases, the real and the simulated one, is compared.

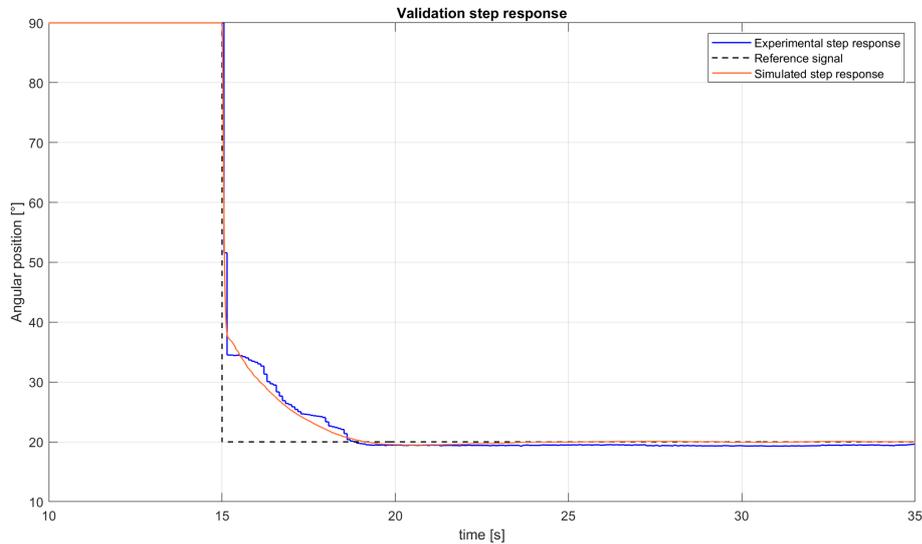


Figure 3.12: Model validation with step response

From the figure it can be noticed that the model response follows accurately the

real system response. The rise time and the settling time of the two response are comparable and neither has overshoot. The slow transient response is due to a low performance control action.

Another signal used to validate the model is the drive cycle, explained in Section 5.3.2.

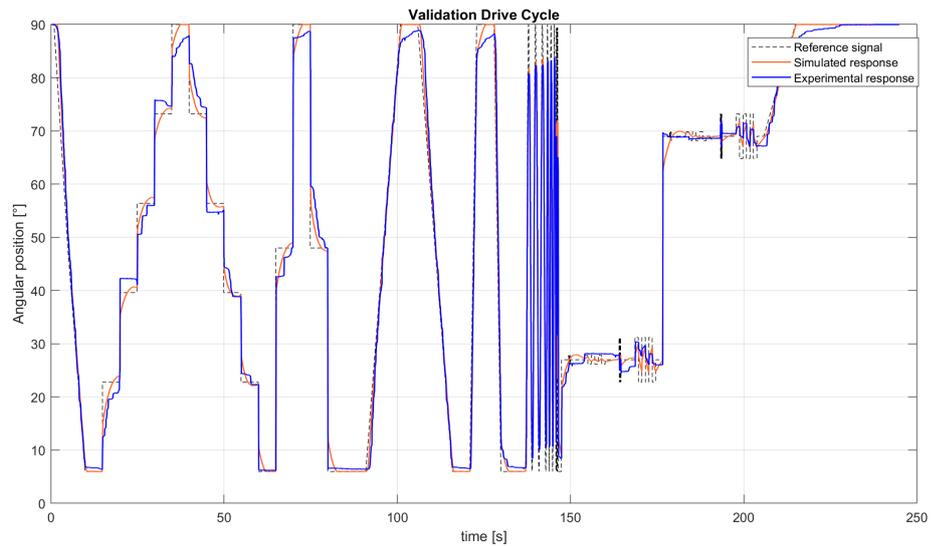


Figure 3.13: Model validation with drive cycle

Also in this case it is possible to notice as there is a small difference between the response of the model and that of the real system. A greater error between the two signals is appreciable in the final part of the test in which the small-signal operating mode of the system is tested. In this case, in fact, the effect of the nonlinearities of the system is more accentuated and therefore a greater level of detail is required in the modeling of these phenomena. For the application of the study the level of detail was considered acceptable.

Chapter 4

Control techniques

4.1 Introduction

Modern microcontrollers allow the implementation of more complex controls that provide better performance while maintaining hardware costs unchanged. For this reason, it is interesting to consider more sophisticated control strategies to evaluate their benefits in terms of performance compared to the current state of the art.

In this chapter, all topics related to position control of the throttle valve will be examined.

The study was conducted starting from simpler and more common linear control techniques, ending with nonlinear control ones. To avoid unnecessary complexity, the different controls considered will be analyzed explaining, for each of them, the critical issues and the possible solutions. The first paragraph is dedicated to the different control techniques common features so that repetitions within the chapter can be avoided.

Moreover, Mason's rule was used to derive the system's closed-loop transfer function. It was used for every controller design and an explanation of the method can be found in the Appendix A.

4.2 Control common features

4.2.1 Control design and validation procedure

The tools chosen to design and validate each control technique are MATLAB and Simulink. In particular, MATLAB scripts were used to determine the parameters of

the various types of control, while Simulink models were developed for the control system validation.

Since the control techniques are very different from each other, for each control it was necessary to develop a different MATLAB code, the theory behind which will be explained in the following sections. The same cannot be said for the validation. Indeed, it was possible to use the same model more than once by modifying only the control system. A diagram of the model scheme is shown in Figure 4.1 below:

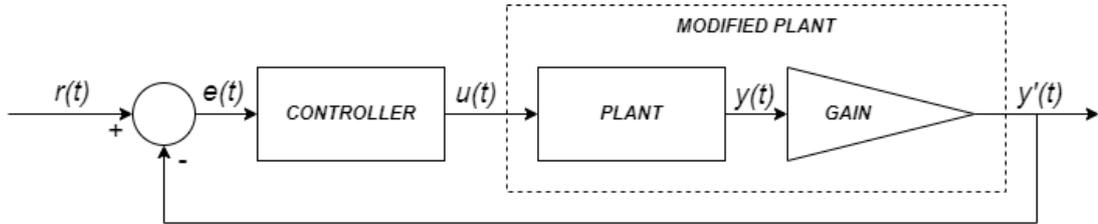


Figure 4.1: Model scheme of the closed-loop control system

where the model signal:

- $r(t)$ is the desired position and it is expressed in percentage;
- $e(t)$ is the error signal, i.e. the difference between the percentage desired position and the percentage feedback position. It is expressed in percentage;
- $u(t)$ is the plant command voltage expressed in Volt;
- $y(t)$ is the plant output and it is the feedback angular position of the throttle valve expressed in radians;
- $y'(t)$ is the feedback angular position of the throttle valve expressed as a percentage;

while the block:

- **CONTROLLER** represents the subsystem that contains the controller block diagram;
- **PLANT** is the subsystem that contains the plant models. To make the model more flexible, a variant subsystem was used. In this way it was possible to select the model to be simulated in the closed loop system using a specially defined variable, editable directly from MATLAB script;
- **GAIN** transforms the angular position expressed in radians in a percentage angular position. This gain was multiplied by the open-loop gain of the

transfer functions shown in Chapter 3 and must be considered in the control design:

$$GAIN = \frac{100}{\frac{84\pi}{180}} = 1.4661 \frac{\%}{rad} \quad (4.1)$$

where 84 is the angular stroke value of the actuator measured in degrees.

Not having an exact measurement of the position of the upper and lower mechanical limits but only an estimate found in the actuator documentation, gain has been added to allow easier and more flexible limit recalibration. To design the control techniques, the linear transfer (3.2.2) and reduced linear transfer (3.2.3) functions of the system have been modified by multiplying the open loop gain of the transfer function by the gain shown in the diagram.

$$b'_0 = b_0 GAIN \quad (4.2)$$

The validation algorithm used for control techniques is shown in Figure 4.2 below.

All simulations shown in the algorithm are performed in MIL using the combination of MATLAB and Simulink described above, where MIL stands for Model-in-the-loop simulation and it means that both the control technique and the plant exist entirely in native simulation tool.

The only exception is the last step where the control system is tested in HIL. A detailed description of the experimental setup used to perform this kind of test and a description of the test itself will be provided in Chapter 5.

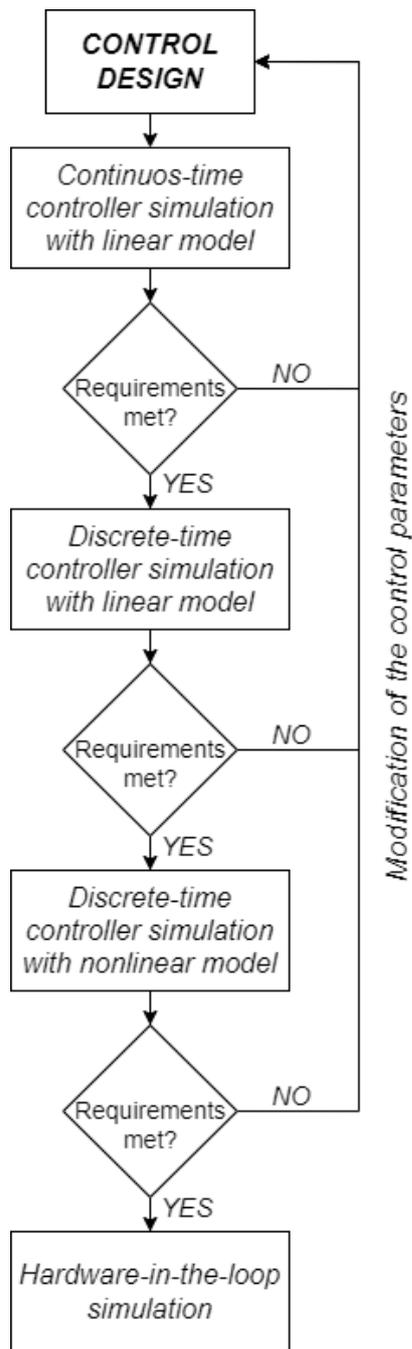


Figure 4.2: Control design algorithm

4.2.2 Integrator anti-windup

Due to the voltage limitations imposed on the EPU, the control signal is often cut in amplitude when the error between set and feedback or the control gains are too high. This results in an unexpected overshoot or undershoot in the step response or in an error at too high speed. To work around this problem, an anti-windup has been added, which is a control technique that provides a feedback branch that evaluates the error between filtered and unfiltered command signals and limits the contribution of the integrative branch of the control through a fixed gain. In this way the command signal is decreased just enough to stay below the limits imposed by the hardware.

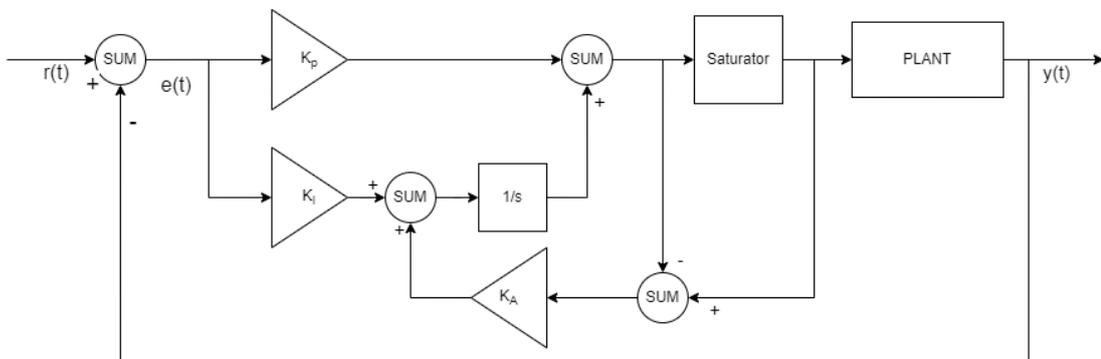


Figure 4.3: An example of anti-windup applied to PI control

4.2.3 Dither

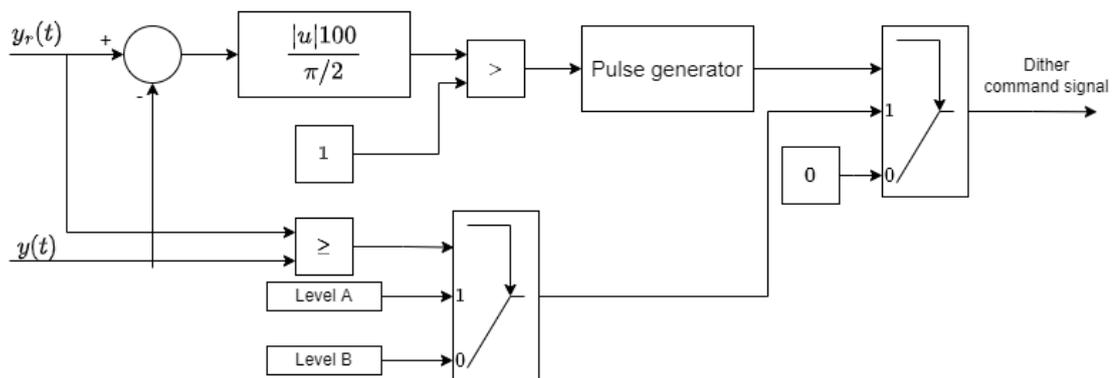


Figure 4.4: Control dither block diagram

Dither is a command signal which induces an alternating torque on the actuator, allows the system to be removed from static conditions and therefore to consider the static friction torque negligible. Indeed, the presence of static friction in the system causes a delay in the restart of the valve plate during the movement reversal phases or in the initial part of the step response. To reduce this effect and obtain a lower initial transient, a Dither signal has been added to the implemented control signal. The Dither signal generally has a constant frequency greater than zero and lower than a fixed value, whose sign depends on the sign of the error between set and feedback. In order to active the control signal, the mentioned error must be greater than a certain threshold (that is chosen by experimental test to be equal to 0.6%). In order not to affect the output signal, the frequency of the alternating torque control signal introduced by the Dither must be greater than the cutoff frequency of the signal itself.

At the beginning of the study it was imagined to add dither control to the design of each control technique. Unfortunately because of problems of delivery times, the dither has been tested only with the sliding mode control.

Unlike the other results that will be shown in the following chapter, the results of the tests performed to understand the action of dither have been reported below.

The test is executed exciting the system with a slow ramp signal, which aims to highlight the phenomenon of static friction. The dither command signal is a pulse duty cycle and was added to the controller one.

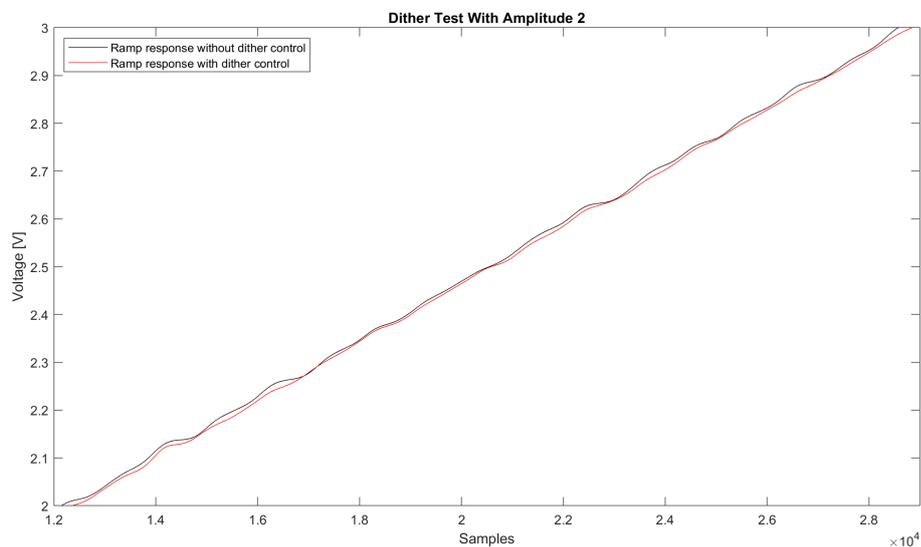


Figure 4.5: System ramp response with and without dither control. Pulse amplitude equal to 2%

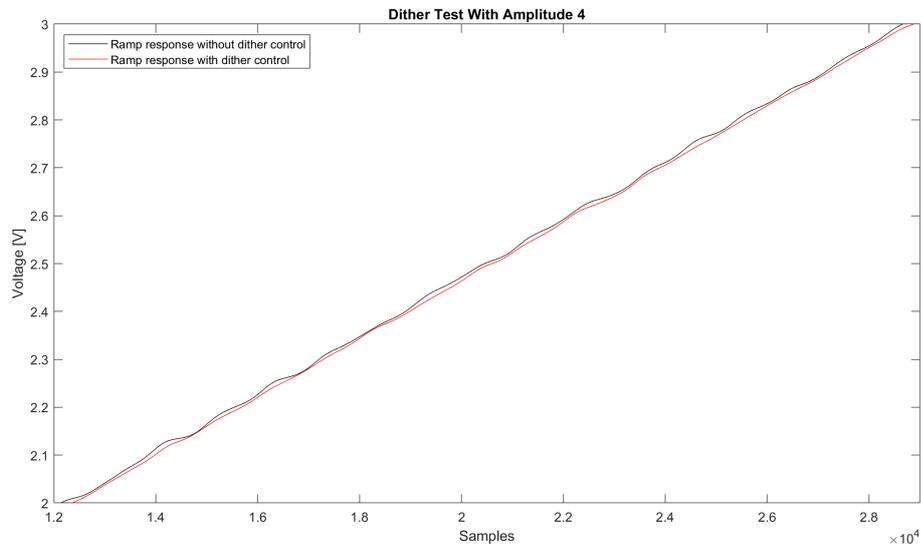


Figure 4.6: System ramp response with and without dither control. Pulse amplitude equal to 4%

The two graphs in Figure 4.5 and Figure 4.6 represent the control system response with and without the dither control implemented, respectively, with a pulse signal of amplitude of 2% and 4%.

From the graphs, Figure 4.5 and 4.6, can be noticed the presence of steps in the system response without dither. These are due to the stick-slip phenomenon. Moreover, it can be noticed how the steps are smoothed in presence of the dither control action and, in particular, that with a higher pulse amplitude of the control signal these are slightly smoother.

4.3 Control design

4.3.1 PID

The first control technique considered in the study is the simplest and most used in the industrial field, i.e. the PID controller. The choice to design a PID was made to compare the performance results obtained with more complex control techniques and make a critical analysis of the advantages and disadvantages in the implementation of the latter.

For this purpose, it was decided not to even include the feed forward compensator, described in Section 2.3, in order to point out the difference in complexity between

the solutions adopted even more. The only improvement made on the most traditional PID is the inclusion of an integrator anti-windup, which avoided system damage due to overvoltage.

For the control was followed the design algorithm described in Section 4.2.1 with the difference, compared to other control techniques, that to find the parameters was adopted a trial and error tuning procedure.

An explanation of the control strategy is shown in Section 2.3, while a scheme of the system with the anti-windup supplement is shown in Figure 4.7.

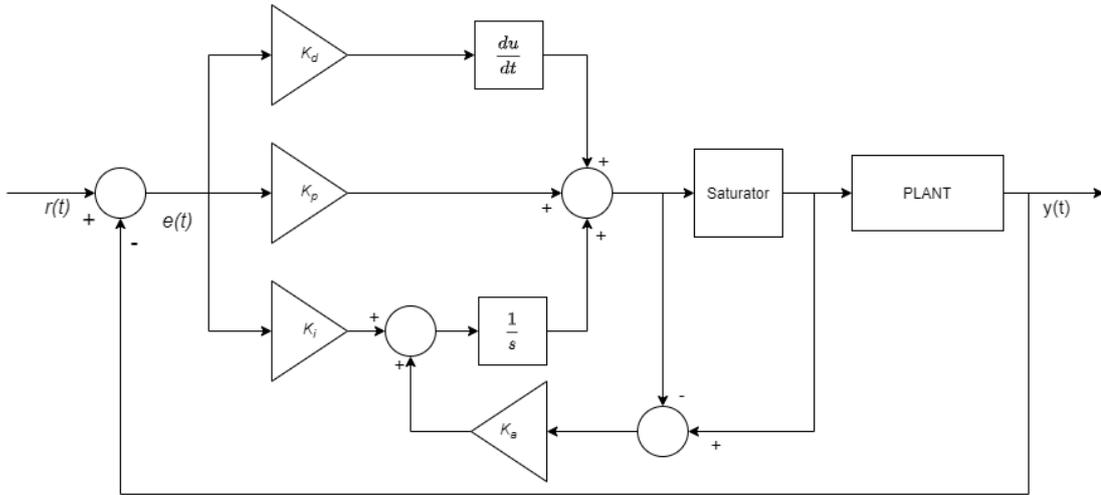


Figure 4.7: PID with integrator anti-windup

The parameters founded are:

$$\begin{cases} K_p = 4,4 \\ K_i = 3,3 \\ K_d = 0,08 \\ K_a = 0,11 \end{cases} \quad (4.3)$$

4.3.2 IPD

The IPD control is a modified version of the classical PID control in which the proportional and the derivative branches are triggered directly from the feedback path of the closed loop system. The following Figure 4.8 shows a scheme of the control system.

$$\frac{Y(s)}{Y_r(s)} = \left[\frac{\frac{1}{T_c}}{s + \frac{1}{T_c}} \right]^n = \left[\frac{\frac{1.5(1+n)}{T_s}}{s + \frac{1.5(1+n)}{T_s}} \right]^n \quad (4.6)$$

A *robust pole placement* can be performed since the number of adjustable parameters is equal to the order of the control closed loop transfer function. Despite the fact that, theoretically, the idea of closed loop poles vanishes upon closed loop with a nonlinear plant, robust pole placement is known to be thriving with plants that have continuous nonlinearities. A simple explanation of this is that the robustness is achieved through relatively high gains and the real implementation operates in the time domain.

To achieve high gains and to obtain the settling time specification without overshoot, $n - 1$ *dominant closed loop poles* should be placed at $-1/T_c$, according to Equation 4.4, with n replaced by $n - 1$. Thus:

$$T_c = \frac{T_s}{1.5n} \quad (4.7)$$

The magnitude of the remaining closed loop pole at $-1/T_f$ must be chosen so as to be faster than the dominant poles: this pole takes the name of *fast closed loop pole*. To guarantee a high degree of system stiffness, its value should be very high; although, in practice, its magnitude is limited by the sampling period h of the system. In fact, the characteristic polynomial's roots in the z -plane must be located and confirmed to fall within the unit circle in order to calculate the lower limit of T_f below which instability develops; however $T_f > 2h$ provides a pretty solid guidance [4].

Moreover, the fast closed loop pole must be positioned far enough away from the dominant poles to prevent the closed loop system response to deviate too much from its ideal specifications.

To solve this problem, a *minimum pole-to-pole dominant ratio* r_{ppmin} is used and it is defined as:

$$r_{ppmin} = \frac{\text{MinimumFastPoleMagnitude}}{\text{DominantPoleMagnitude}} = \frac{\frac{1}{T_{fmax}}}{\frac{1}{T_c}} \quad (4.8)$$

As a result, from Equation 4.8, another condition on the fast pole position can be derived:

$$T_{fmax} = \frac{T_c}{r_{ppmin}} \quad (4.9)$$

Therefore, the magnitude of the fast closed loop pole must comply with the following conditions 4.10:

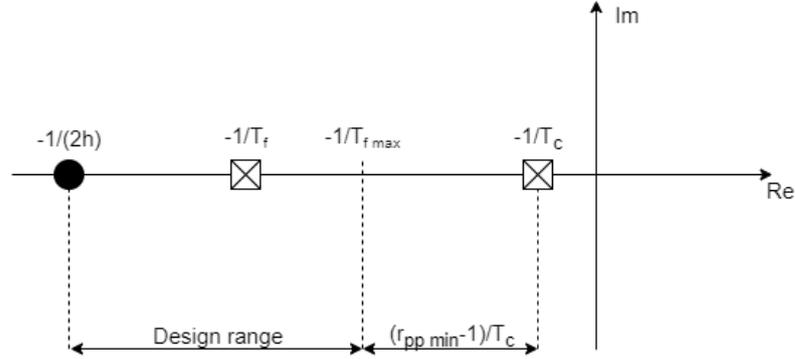


Figure 4.9: Root locus of closed loop system using robust pole placement

$$2h < T_f < (T_c/r_{ppmin}) \quad (4.10)$$

where r_{ppmin} depends on the number of dominant closed loop poles and fast closed loop poles. In this case the dominant poles are 2 and the fast pole is 1 so, from the table given in (Dodds 2013) and from calibration r_{ppmin} is chosen.

$$r_{ppmin} = 1.5$$

IPD parameters

At this point, it was possible to derive the IPD parameters (K_P , K_I , K_D) by comparing the characteristic polynomial and Equation 4.11.

$$(s + p)^2(s + q) = s^3 + (2p + q)s^2 + (p^2 + 2pq)s + p^2q \quad (4.11)$$

In Equation 4.11, p and q are the reciprocal values of T_c and the T_f , respectively.

The characteristic polynomial for the IPD closed loop system is the denominator of the system closed loop transfer function that is derived using the Mason's rule and the reduced order linear model of the system.

$$G(s) = \frac{b_0}{s^2 + a_1s + a_0} \quad (4.12)$$

The results is:

$$T(s) = \frac{b_0K_I}{s^3 + (a_1 + b_0K_D)s^2 + (a_0 + b_0K_P)s + b_0K_I} \quad (4.13)$$

Equating the two polynomials the following system of equations is given:

$$\begin{cases} K_P = \frac{p^2 + 2pq - a_0}{b_0} \\ K_I = \frac{p^2q}{b_0} \\ K_D = \frac{2p + q - a_1}{b_0} \end{cases} \quad (4.14)$$

4.3.3 Discrete-time polynomial control

RST or polynomial control is the only control technique designed directly in discrete time. It represents a linear control technique and uses the closed-loop transfer function of the system to perform pole placement. An explanation of the steps followed to carry out the control design is provided below.

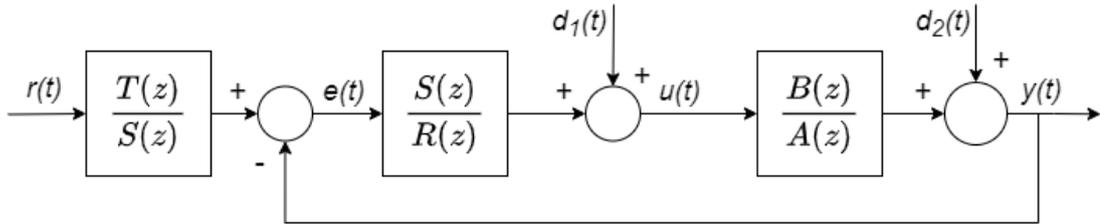


Figure 4.10: Closed-loop system with RST block diagram

- $F(z) = \frac{T(z)}{S(z)}$, $C(z) = \frac{S(z)}{R(z)}$: controller transfer functions;
- $G(z) = \frac{B(z)}{A(z)}$: plant model discretized transfer function;
- $r(t)$: reference signal;
- $e(t)$: error signal;
- $d_1(t)$, $d_2(t)$: disturbance signals;

- $\mathbf{u}(t)$: command signal;
- $\mathbf{y}(t)$: output signal;

For greater clarity in the design development the following definitions are provided:

- $R(z) = (z - 1)^{l_2} B^+(z) R'(z)$
- $S(z) = A^+(z) S'(z)$
- $B(z) = B^+(z) B^-(z)$
- $A(z) = (z - 1)^{l_1} A'(z) = (z - 1)^{l_1} A^+(z) A^-(z)$

Where + means stable zeros/poles, – means unstable zeros/poles, l_1 is the number of poles at the origin of the plant transfer function and l_2 is the number of poles at the origin of the controller transfer function.

The closed loop transfer function of the control system is:

$$W(z) = F(z)W'(z) = \frac{T(z)}{A^+(z)S'(z)} \frac{B^-(z)S'(z)}{(z - 1)^l A^-(z)R'(z) + B^-(z)S'(z)} \quad (4.15)$$

and the objective is to design a controller $C(z)$ to place the roots of the closed loop characteristic polynomial $A_m(z)$ at given locations. The problem can be formulated as:

$$(z - 1)^l A^-(z)R'(z) + B^-(z)S'(z) = A_m(z) \quad (4.16)$$

$R'(z)$ and $S'(z)$ have to be derived knowing l , $A^-(z)$, $B^-(z)$ and $A_m(z)$.

Design phases

1. Discretize the plant transfer function and derive its poles and zeros. In this case, the transfer function used was the reduced order linear one, in order to obtain a simpler control transfer function and reduce the computation effort required. The sampling time used to discretize the plant is the same used in the MCU and it is equal to $SamplingTime = 0.001s$.

$$G(z) = \frac{0.0057(z + 0.9761)}{(z - 0.9987)(z - 0.9311)} \quad (4.17)$$

2. The second step to control design is the translation of the system transient requirements into damping ζ and natural frequency ω_n of a 2^{nd} prototype transfer function. It is expected that a suitable 2^{nd} order prototype system can

adequately explain the intended transient performance of a feedback control system:

$$W(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (4.18)$$

The transient requirements are those analysed in Section 2.2.2 and they can be expressed as functions of ω_n and ζ :

$$\begin{cases} \hat{s} = e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}} \\ t_{s,5\%} = \frac{\log(\frac{100}{\alpha})}{\omega_n\zeta} \end{cases} \quad (4.19)$$

where \hat{s} is the maximum percentage overshoot and $t_{s,5\%}$ is the settling time.

Inverting the Equations 4.19 the 2nd order prototype system parameters are obtained:

$$\begin{cases} \zeta = \frac{|\ln \hat{s}|}{\sqrt{\pi^2 + \ln^2 \hat{s}}} \\ \omega_n = \frac{3}{t_{s,5\%}\zeta} \end{cases} \quad (4.20)$$

3. Stable zero-pole cancellation between $C(z)$ and $G(z)$ can be done to reduce the controller design computation. So, using the dumping factor and the natural frequency found before, all the zeros and poles of the plant transfer function within the cardioid have to identified.

Observing the below Figure 4.11 can be noticed that the plant transfer function has two stable poles and one unstable zero. Furthermore, the cardioid is flattened because the requirements imposed on the overshoot are very stringent and consequently the damping factor is high.

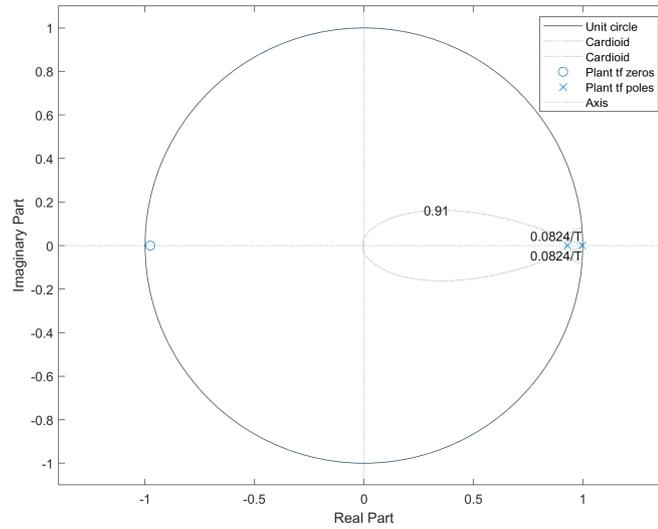


Figure 4.11: Root locus

- From the steady-state error requirement, which in this case is chosen to be zero, the number of closed loop poles at $z = 1$ of controller transfer function are derived. In particular, the final value theorem is used to perform steady-state analysis (Equation 4.21) and the results are summarized in the following Table 4.12:

$$\lim_{k \rightarrow +\infty} f(k) = \lim_{z \rightarrow 1} (z - 1)F(z) \quad (4.21)$$

$h \backslash \ell$	0 (step)	1 (linear ramp)	2 (parabolic ramp)
0	$\frac{ \rho }{1 + K_\ell}$	∞	∞
1	0	$\frac{ \rho }{K_\ell}$	∞
2	0	0	$\frac{ \rho }{K_\ell}$

Figure 4.12: Steady-state tracking error

From Table 4.12 $l = 1$, since it is required to have a zero steady-state tracking error with $h = 0$, i.e. step response. From l and the number of closed loop

poles at $z = 1$ of the plant transfer function, the number of closed loop poles at $z = 1$ of the controller transfer function can be derived:

$$l = l_1 + l_2 \Rightarrow l_2 = l - l_1 = 1 \quad (4.22)$$

5. From point 3 and point 4 the degree of the polynomials $S'(z)$, $R'(z)$ and $A_m(z)$ can be calculated, so the diophantine equation allows a unique solution:

$$\deg(S'(z)) = l + \deg(A^-(z)) - 1$$

$$\deg(R'(z)) = \deg(A'(z)) - \deg(B^+(z)) + l_1 - 1$$

$$\deg(A_m(z)) = l + \deg(A'(z)) + \deg(A^-(z)) + l_1 - \deg(B^+(z)) - 1$$

6. Based on the degree of the polynomial $A_m(z)$, number and position of the continuous time desired closed loop poles are chosen. A couple of complex values are chosen that depends on the dumping factor and the natural frequency of the 2^{nd} order prototype system function:

$$p_{1,2}^c = \sigma_0 \pm j\omega_0 = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2} \quad (4.23)$$

Once the continuous time poles have been chosen, they need to be mapped into their discrete time equivalent and this is done through the sampling transformation:

$$p_{1,2} = e^{T p_{1,2}^c} \quad (4.24)$$

Where p are the discrete time poles and T is the sampling time.

$A_m(z)$ is computed as:

$$A_m(z) = (z - p_1)(z - p_2) \quad (4.25)$$

7. At this point, with all the data computed in the previous steps, it is possible to solve the diophantine equation. This is done using a MATLAB script. At the end of the process the controller obtained is:

$$C(z) = \frac{0.2328(z - 0.9987)(z - 0.9311)}{(z - 1)(z - 0.8594)} \quad (4.26)$$

8. Usually, the design of $F(z)$ aims to:

- Place additional zeros in $W(z)$ in order to improve the controlled output response;

- Impose a suitable dc-gain to the closed loop transfer function $\mathbf{W}(z)$;
- Cancel the polynomial $\mathbf{S}'(z)$, in this way the slow zeros of $\mathbf{S}'(z)$ do not affect the closed loop response.

Although, in this case there are no additional faster poles so the transfer function of $\mathbf{F}(z)$ is a simple unit gain.

4.3.4 Linear state feedback control

Basic linear state feedback control

The model of the linearized plant can be represented by the equation of state:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}u, \mathbf{y} = \mathbf{c}^T\mathbf{x} \quad (4.27)$$

Where \mathbf{x} is the vector of states, \mathbf{u} is the plant command input, \mathbf{y} is the output of the system. \mathbf{A} represents the plant matrix, \mathbf{b} is the input matrix, and \mathbf{c}^T is the output matrix. Linear state feedback (LSF) is a control technique in which \mathbf{x} is feedback according to the law:

$$\mathbf{u} = \mathbf{y}_r\mathbf{r} - \mathbf{g}^T\mathbf{x} \quad (4.28)$$

Where \mathbf{g}^T is the vector of independent gains. The term \mathbf{r} represents the gain that is used to adjust the step response steady state error. A block diagram of the basic linear state feedback control is shown in Figure 4.13.

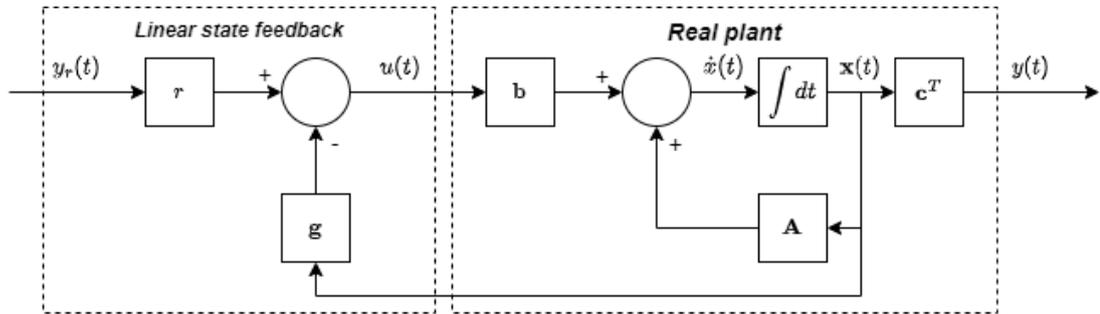


Figure 4.13: Basic linear state feedback control block diagram

With this control technique the desired closed loop transient response of the system can be reached as long as all the states of the real plant can be feedbacked. The control design can be done finding the characteristic polynomial of the closed loop

system and obtaining the control parameters by solving the system of equations derived by the equality of the polynomial and the polynomial found using the Dodds settling time formula of the same order. A critical parameter is the value of r which determines the step response steady state error.

Linear state feedback control with state observer

The basic linear state feedback control is based on the information about the states of the real plant. Most of the time this type of information is difficult to derive because it requires high instrumentation costs or it is impossible to measure the desired physical quantities. This is one of those cases because an online measurement of the DC motor armature current of the would be necessary for the operation of the control. To avoid this problem a state observer of the plant has been employed. Figure 4.14 shows the linear state feedback control with state observer block diagram.

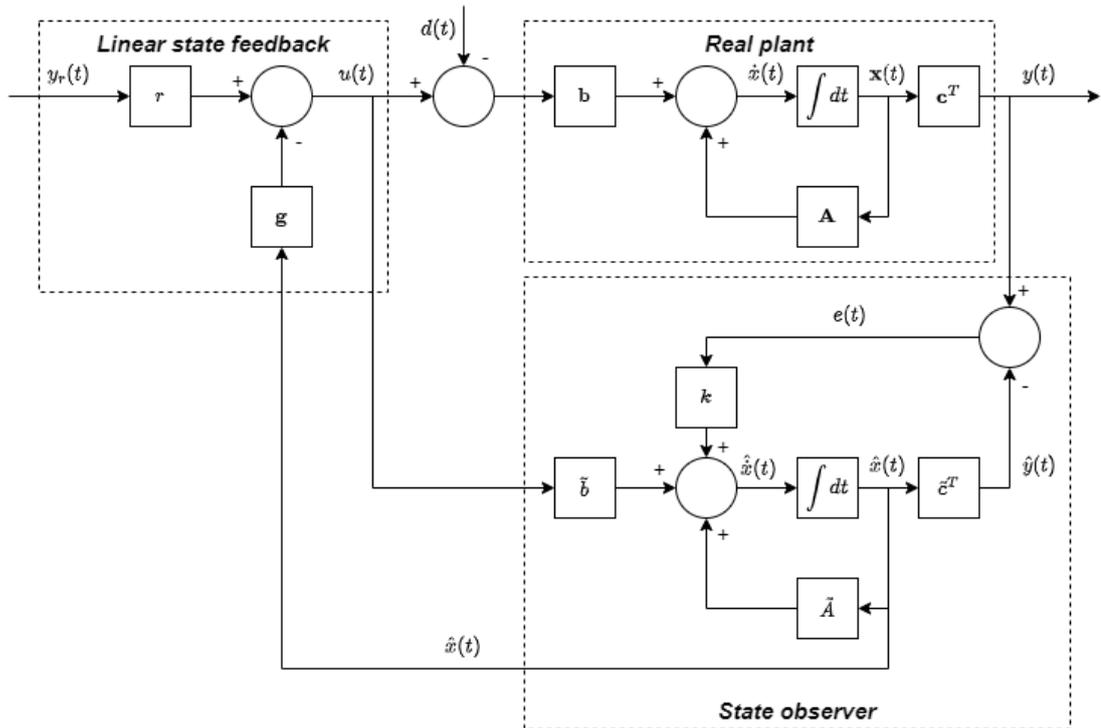


Figure 4.14: Linear state feedback control with state observer block diagram

where:

- \tilde{A} , \tilde{b} and \tilde{c}^T plant model matrices.

- $\hat{\mathbf{x}}(t)$ is the vector of estimated states.
- $\hat{\dot{\mathbf{x}}}(t)$ is the vector estimated states derivatives.
- $\hat{\mathbf{y}}(t)$ is the estimated output.

The state observer is built on a real plant model that receives the same control input as the actual plant and has its state adjusted to match the real plant's state using a correction loop that is activated by the error, $\mathbf{e}(t) = \mathbf{y}(t) - \hat{\mathbf{y}}(t)$ and applied to the integrator inputs of the plant model through a gain vector, \mathbf{k} . The error $\mathbf{e}(t)$ converged to zero if the correction loop gains are chosen properly, such that $\hat{\mathbf{x}} = \mathbf{x}$.

The correction loop aims to:

- Reduce the differences between model parameters and real plant parameters.
- Reduce the differences between the initial model and plant states.
- Reduce the effect of the external disturbances.

The set of equations that regulates the system are:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}u, \mathbf{y} = \mathbf{c}^T \mathbf{x} \\ \dot{\hat{\mathbf{x}}} = \tilde{\mathbf{A}}\hat{\mathbf{x}} + \tilde{\mathbf{b}}u + \mathbf{k}(\mathbf{y} - \hat{\mathbf{y}}), \hat{\mathbf{y}} = \tilde{\mathbf{c}}^T \hat{\mathbf{x}} \\ u = \mathbf{y}_r - \mathbf{g}\hat{\mathbf{x}} \end{cases} \quad (4.29)$$

Assuming $\tilde{\mathbf{A}} = \mathbf{A}$ and $\tilde{\mathbf{b}} = \mathbf{b}$, the state estimator error, $\boldsymbol{\epsilon} = \hat{\mathbf{x}} - \mathbf{x}$ shall be made to converge to zero from an arbitrary initial condition. This is demonstrated by subtracting the first two equations of the set that defines the system, obtaining:

$$\dot{\boldsymbol{\epsilon}} = [\mathbf{A} - \mathbf{k}\mathbf{c}^T]\boldsymbol{\epsilon} \quad (4.30)$$

So, if the gain matrix, \mathbf{k} , is chosen so that the eigenvalues of the matrix, $\mathbf{A} - \mathbf{k}\mathbf{c}^T$, have negative real part, then $\boldsymbol{\epsilon} \rightarrow \mathbf{0}$ as $t \rightarrow \infty$. To perform the pole placement the characteristic equation of the system has to be found and the settling time formula is used:

$$\det(s\mathbf{I} - [\tilde{\mathbf{A}} - \tilde{\mathbf{c}}\mathbf{k}]) = \left[s + \frac{1.5(1+n)}{T_{so}} \right]^n \quad (4.31)$$

where T_{so} is the observer settling time and its value is fundamental to achieve the desired closed loop step response starting from any arbitrary initial state estimation

error. To reach this goal the observer settling time is chosen as $T_{so} < T_s/5$. On the other hand, the parameter can't be too low because this would lead to errors in estimating the state due to measurements noise. The plant model considered to derive the characteristic equation is the reduced order linear one.

Moreover, in order to eliminate or reduce the steady state error, an integrator has been added to the forward path. A block diagram of the considered control system is shown in Figure 4.15.

Using the procedure described above, the control parameters were found:

$$\left\{ \begin{array}{l} g_2 = \frac{[(2p+q)-a_1]}{b_0} \\ g_1 = \frac{[(p^2+2pq)-a_0]}{b_0} \\ k_i = \frac{p^2q}{b_0} \\ k_0 = 2p_0 - a_1 \\ k_1 = p_0^2 - a_0 - a_1k_0 \end{array} \right. \quad (4.32)$$

where:

- $p = \frac{1.5(1+n)}{T_s}$ is the dominant pole.
- $q = pr_{ppmin}$ is the faster pole.
- $p_0 = \frac{1.5(1+n)}{T_{so}}$ is the observer dominant pole.
- $n = 2$ is the control system order.
- $r_{ppmin} = 38$ is the minimum pole-to-pole ratio.
- $T_{so} = \frac{T_s}{5}$ is the observer settling time.

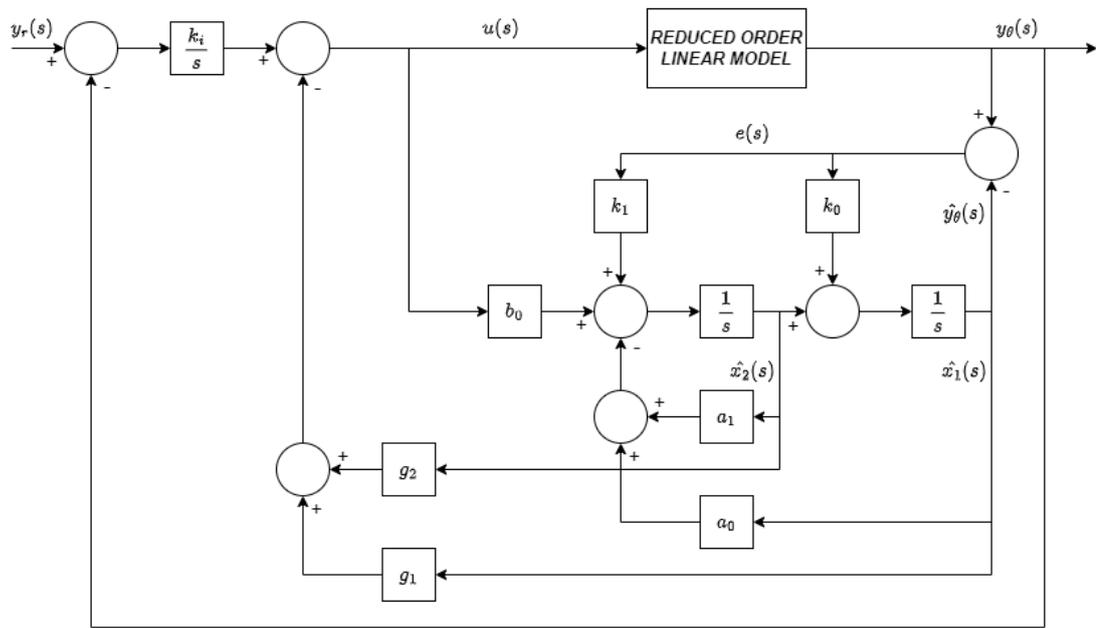


Figure 4.15: Linear state feedback control with state observer and integrator for steady state error elimination block diagram

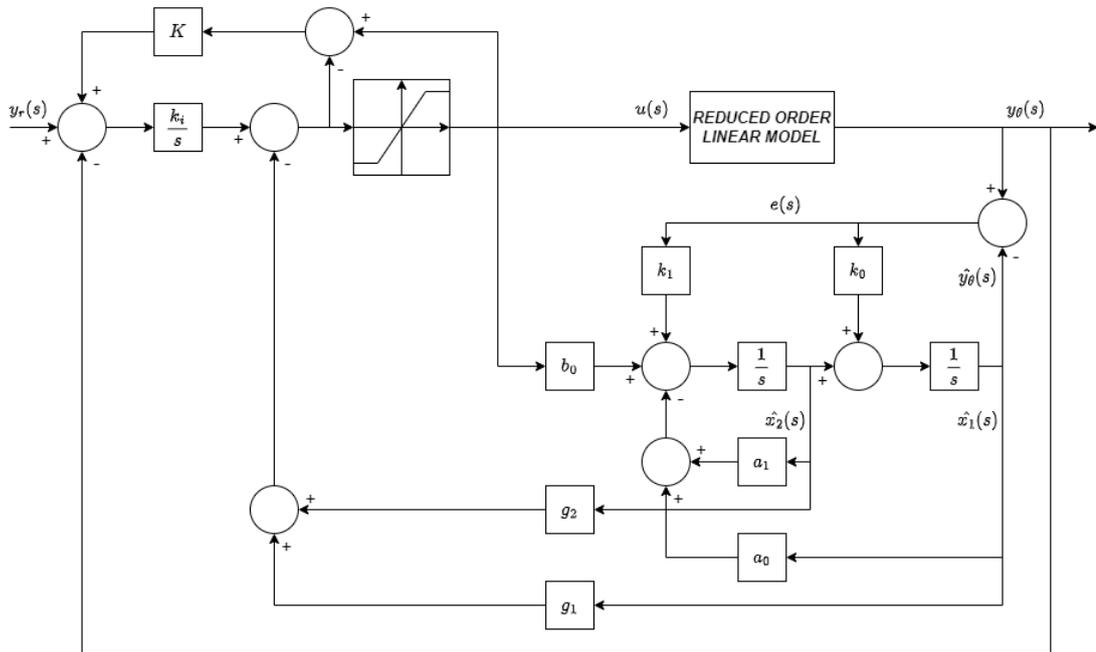


Figure 4.16: Linear state feedback control with state observer and integrator anti-windup block diagram

To prevent the control system to overshoot an integrator anti-windup is added to the control scheme and its gain \mathbf{K} is derived experimentally. A scheme of the modified control technique is shown in Figure 4.16.

4.3.5 Sliding mode control

Sliding mode control (SMC) is a discontinuous nonlinear control technique. The main advantage of this controller is its robustness to the variation of plant parameters and external disturbances.

Basic sliding mode control

To better understand the control technique, a practical example concerning the double integrator plant is provided.

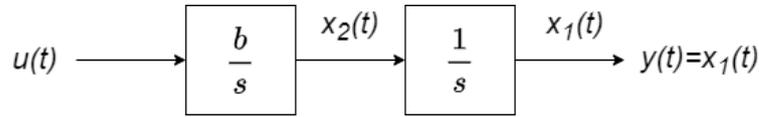


Figure 4.17: Double integrator plant

where $\mathbf{u}(t)$ is the command input to the plant and $\mathbf{x}_1, \mathbf{x}_2$ are the states.

The differential equations that describe the plant dynamics are:

$$\begin{cases} \dot{\mathbf{x}}_1(t) = \mathbf{x}_2(t) \\ \dot{\mathbf{x}}_2(t) = \mathbf{b}\mathbf{u}(t) \end{cases} \quad (4.33)$$

The first step to sliding mode control is to perform a bang-bang state control law for which the command input is switched between its extreme, i.e. $\mathbf{u} = \pm \mathbf{u}_{max}$.

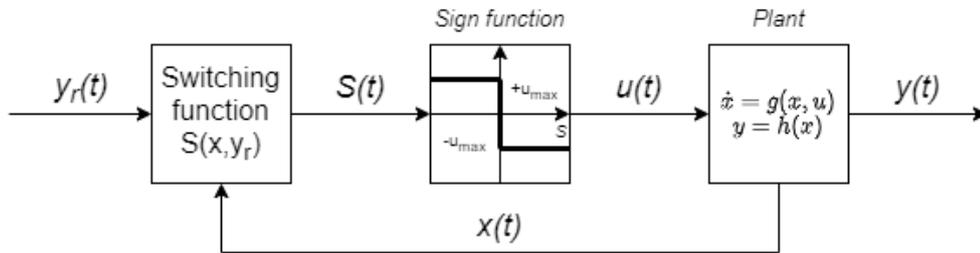


Figure 4.18: Basic sliding mode control scheme

The general form of the switching function shown in Figure 4.18 is:

$$\mathbf{S}(\mathbf{x}, \mathbf{y}_r) = \mathbf{W}_1(\mathbf{x}_1 - \mathbf{y}_r) + \mathbf{W}_2\mathbf{x}_2 + \cdots + \mathbf{W}_n\mathbf{x}_n \quad (4.34)$$

Where:

- \mathbf{x} are the plant states
- $\mathbf{W}_{1,2,\dots,n}$ are constants
- n is the number of plant states

In case of a double integrator plant, the switching function became:

$$\mathbf{S}(x_1, x_2, y_r) = \mathbf{W}_1(x_1 - y_r) + \mathbf{W}_2x_2 \quad (4.35)$$

To represent the switching behavior of $\mathbf{u}(t)$ in basic sliding mode control the following sign function is applied:

$$\mathbf{u}(t) = -u_{max}\mathit{sign}(\mathbf{S}(x_1, x_2, y_r)) \quad (4.36)$$

with:

$$\mathit{sign}(\mathbf{S}(x_1, x_2, y_r)) = \begin{cases} +1 & \mathit{for} \mathbf{S} > 0 \\ 0 & \mathit{for} \mathbf{S} = 0 \\ -1 & \mathit{for} \mathbf{S} < 0 \end{cases} \quad (4.37)$$

To visualise the states the controller goes through, a trajectory with $\mathbf{W}_1, \mathbf{W}_2 = \mathbf{1}$ is shown in Figure 4.19.

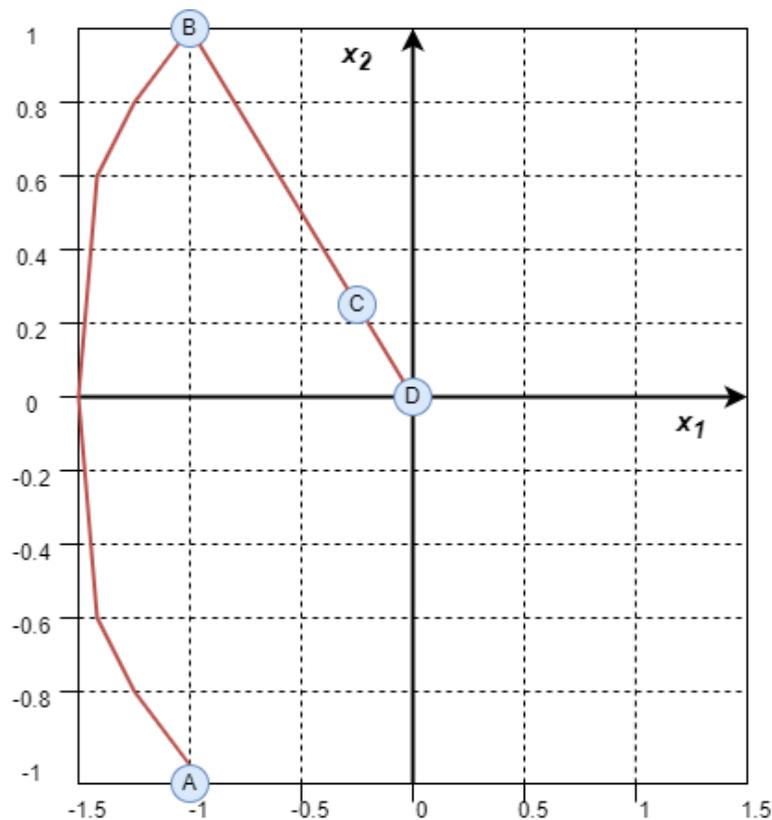


Figure 4.19: Double integrator state trajectory

Point A represents the initial condition of the system. Applying $\mathbf{u}(t) = \mathbf{u}_{max}$ the states moves to point B with no changes in the controller output. At point B the controller reaches the boundary layer and starts to switch the control action from $\mathbf{u}(t) = \mathbf{u}_{max}$ to $\mathbf{u} = -\mathbf{u}_{max}$ at, in theory, an infinity frequency following the switching boundary until point D is reached. In practice the switching frequency is not infinite and this causes the problem of chattering, i.e. the behavior of the states between the points C and D is not linear but oscillates causing a zigzag movement of the trajectory and a damage to the actuator. To avoid this problem, a filtering of the control action is necessary and will be explained when talking about boundary layer method.

The Equation 4.35 controls the system when the two states are on the switching boundary. So, the sliding mode control force the state to stay on the switching boundary and for this reason the closed loop system has a degree of freedom less than the plant.

The control has to control the output so that it follows the desired set, moreover,

this must be done with precise performance requirements and in addition reaching a certain degree of robustness. To do this the states in Equation 4.34 are replaced with the derivatives of the plant output $\mathbf{y}(t)$.

$$\mathbf{S}(\mathbf{y}_r, \mathbf{y}) = \mathbf{W}_1(\mathbf{y} - \mathbf{y}_r) + \mathbf{W}_2\dot{\mathbf{y}} + \mathbf{W}_3\ddot{\mathbf{y}} + \cdots + \mathbf{W}_n\mathbf{y}^{n-1} \quad (4.38)$$

where n is the degree of the plant transfer function.

Dividing the Equation 4.38 by \mathbf{W}_1 :

$$\mathbf{S}(\mathbf{y}_r, \mathbf{y}) = (\mathbf{y} - \mathbf{y}_r) + \frac{\mathbf{W}_2}{\mathbf{W}_1}\dot{\mathbf{y}} + \frac{\mathbf{W}_3}{\mathbf{W}_1}\ddot{\mathbf{y}} + \cdots + \frac{\mathbf{W}_n}{\mathbf{W}_1}\mathbf{y}^{n-1} \quad (4.39)$$

Redefining the equation constants we obtain:

$$\mathbf{S}(\mathbf{y}_r, \mathbf{y}) = (\mathbf{y} - \mathbf{y}_r) + \mathbf{W}_1\dot{\mathbf{y}} + \mathbf{W}_2\ddot{\mathbf{y}} + \cdots + \mathbf{W}_{n-1}\mathbf{y}^{n-1} \quad (4.40)$$

To study the behavior of the system when it is close to the switching boundary the Equivalent Control Method is used. $\mathbf{u}_{eq}(t)$ is introduced and it is the short term average value of the rapidly switching physical control. By imposing $\mathbf{S} = \mathbf{0}$, from which $\dot{\mathbf{S}} = \mathbf{0}$ is derived, the solution for $\mathbf{u}_{eq}(t)$ is found:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (4.41)$$

where $\mathbf{x}(t)$ are the plant states and $\mathbf{u}(t)$ is command input. For $\mathbf{y}_r(t) = \mathbf{0}$ the switching boundary is

$$\mathbf{S}(\mathbf{x}) = \mathbf{W}_1\mathbf{x}_1 + \mathbf{W}_2\mathbf{x}_2 + \cdots + \mathbf{W}_n\mathbf{y}_n = \mathbf{W}^T \cdot \mathbf{x} = \mathbf{0} \quad (4.42)$$

So

$$\mathbf{W}^T[\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}] = \mathbf{0} \quad (4.43)$$

$$\mathbf{u}_{eq} = -[\mathbf{W}^T\mathbf{B}]^{-1}\mathbf{W}^T\mathbf{A}\mathbf{x} \quad (4.44)$$

Boundary layer sliding mode control

The boundary layer sliding mode control is the control technique implemented in this study. This control strategy aims to solve the chattering problem of the basic sliding mode control. The problem is solved by introducing a region formed by two saturation bands, one above and the other below the switching boundary, where the state continuously moves between the two bands keeping close to the set point on the sliding surface. In this way, the control action is smoothed. This is achieved

by replacing the switching function with a high gain transfer characteristic with saturation.

$$\mathbf{u}(t) = -\mathbf{u}_{max} \text{Saturation}[\mathbf{K} \cdot \mathbf{S}(y_r, \mathbf{y})] \quad (4.45)$$

where $\mathbf{u}(t)$ is saturated at $\pm \mathbf{u}_{max}$ when $|\mathbf{S}| < \frac{\mathbf{u}_{max}}{\mathbf{K}}$.

It can be noticed that if $\mathbf{K} \rightarrow \infty$ the region previously described became identical to the switching boundary and the state trajectory would follow that of the ideal sliding mode control. Although, this is impossible because, again, to achieve this an infinite switching frequency is required. Moreover, the high gain \mathbf{K} is finite and this causes a steady state error. To fix the issue an integrator is added to the forward path.

A block scheme of the control system is shown in Figure 4.20.

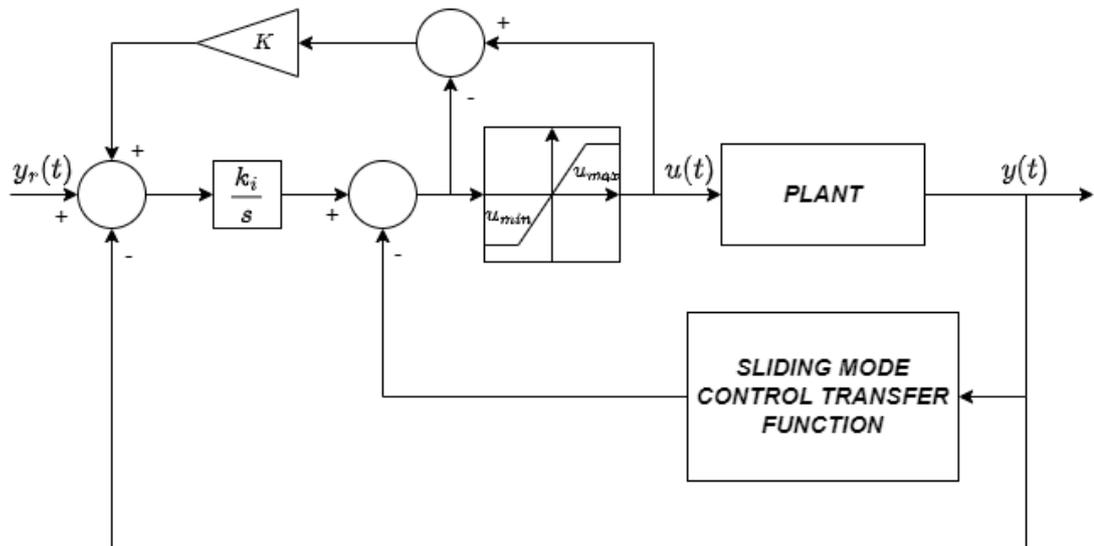


Figure 4.20: Boundary layer sliding mode control block scheme

Where:

- The **plant** is modelled using the linear model derived in Chapter 3.

Initially it was chosen to use the reduced order linear model to design the SMC. But during the experimental tests very low performance of the control system was detected. Then, the control was designed using the linear transfer function.

- According to Section 4.3.5 the **Sliding mode control transfer function** is:

$$\frac{s^2 + q_{c1}s + q_{c0}}{s^2 + q_{f1}s + q_{f0}} \quad (4.46)$$

- The slope of the linear part of the **saturator** is the gain **K**.

To control design, the closed loop control system transfer function is derived using Mason's gain rule.

$$G(s) = \frac{K_i K b_0 (s^2 + q_{f1}s + q_{f0})}{s^6 + (q_{f1} + a_2)s^5 + (q_{f0} + a_2 q_{f1} + a_1)s^4 + (a_2 q_{f0} + a_1 q_{f1} + a_0 + K b_0)s^3 + (a_1 q_{f0} + a_0 q_{f1} + K b_0 (q_{c1} + K_i))s^2 + (a_0 q_{f0} + K b_0 (q_{c0} + K_i q_{f1}))s + K_i K b_0 q_{f0}} \quad (4.47)$$

As the IPD control, the closed loop characteristic equation is designed using robust pole placement (explained in Section 4.3.2). The control system order in this case is $\mathbf{n} = \mathbf{6}$, so three dominant poles and three faster poles have to be placed according to Dodds settling time formula:

$$\begin{cases} \text{Dominant Poles} : p_c = \frac{T_s}{6} \\ \text{Faster Poles} : p_f = r_{ppmin} p_c \end{cases} \quad (4.48)$$

where the minimum pole-to-pole ratio is chosen $r_{ppmin} = 20$.

Then,

$$\begin{aligned} (s + p_f)^3 (s + p_c)^3 &= s^6 + 3(p_f + p_c)s^5 + [3(p_f^2 + p_c^2) + 9p_f p_c]s^4 \\ &+ [p_f^3 + p_c^3 + 9p_f p_c(p_f + p_c)]s^3 + [3p_f p_c(p_f^2 + p_c^2) + 9p_f^2 p_c^2]s^2 \\ &+ 3p_f^2 p_c^2(p_f + p_c)s + p_f^3 p_c^3 = 0 \end{aligned}$$

By equating the denominator of Equation 4.47 and Equation 4.3.5 and solving the

system of equations, the control parameters are derived:

$$\left\{ \begin{array}{l}
 q_{f1} = 3(p_f + p_c) - a_2 \\
 q_{f0} = 3(p_f^2 + p_c^2) + 9p_f p_c - (a_2 q_{f1} + a_1) \\
 K = \frac{p_f^3 + p_c^3 + 9p_f p_c (p_f + p_c) - (a_2 q_{f0} + a_1 q_{f1} + a_0)}{b_0} \\
 K_i = \frac{p_f^3 p_c^3}{b_0 K q_{f0}} \\
 q_{c1} = \frac{3p_f p_c (p_f^2 + p_c^2) + 9p_f^2 p_c^2 - (a_1 q_{f0} + a_0 q_{f1})}{K b_0 - K_i} \\
 q_{c0} = \frac{3p_f^2 p_c^2 (p_f + p_c) - a_0 q_{f0}}{K b_0 - K_i q_{f1}}
 \end{array} \right. \quad (4.49)$$

Chapter 5

Results

5.1 Introduction

In this chapter we will show the results obtained with the various control techniques in terms of step response, drive cycle response, robustness, and behavior in case of spring fault.

First we will show the experimental setup adopted to perform the tests in real-time both in terms of software and hardware. Then an explanation will be given of the evaluation criteria of the controls and in last place the results obtained will be shown and analyzed.

The chapter ends with a classification of the control techniques studied.

5.2 HIL simulation details

The HIL simulation test is the last step of the control development algorithm as shown in Figure 4.2. HIL stands for Hardware-in-the-loop simulation and it is a validation test. It is performed implementing the C-code of the control algorithm in the microcontroller (MCU) and the plant model on a real-time simulator. In this case, since the system is relatively simple, the real system was used directly, instead of simulating it with a real-time simulator. Therefore, it was necessary to build a test bench that would allow the real-time simulation of the system.

The test bench is composed by two major parts, i.e. the software and the hardware. The software part was created using a Simulink model while the experimental setup was built considering the operating requirements of the system.

5.2.1 Simulink test model

The C-code implemented on the MCU is directly generated by a Simulink model made specifically for HIL simulation testing. As in the case of MIL simulation models, this model can also be reused for all control techniques by setting up only the block diagram of the control strategy to be tested.

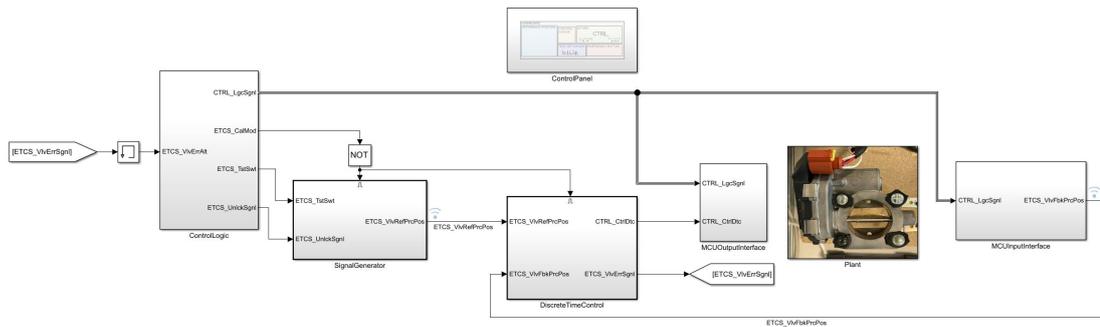


Figure 5.1: Simulink test model

The model consists of several subsystems:

- **ControlPanel subsystem**

It contains a dashboard from which you can set the model according to the test you want to perform and according to the control technique you want to validate. There is also a push button to impose a duty cycle of zero in case the control system is unstable during testing.

- **ControlLogic subsystem**

It manages the operating modes of the system. In particular, there are three different operating modes:

1. In *Normal mode* the controller manages the duty cycle to be sent to the DC motor according to the desired position signal and feedback position. It is the normal operating condition of the actuator.
2. *Calibration mode* is the first operational phase of the system. In this mode the duty cycle signal directed to the DC motor bypasses the control action. This signal has been elaborated experimentally and is used to bring the position of the valve to the two hard stops before switching to the normal mode of use of the actuator. In the period of time when the valve is at its extreme positions, the feedback signal is mediated and saved

to define the lower and upper limits of valve position in the condition of normal system use.

3. The *Blocked valve mode* is reached when the set position is not reached within a set time threshold. In this mode, first of all a signal is sent to try to unlock the target. If, however, the set position is not reached again, a zero duty cycle is imposed on the DC motor to prevent damage to the actuator due to, for example, too high currents.

- **SignalGenerator subsystem.** It contains the test signals described in the next Section 5.3.
- **DiscreteTimeControl subsystem.** It contains a variant subsystem with all block diagrams of discretized control techniques.
- **MCUOutputInterface & MCUInputInterface subsystems.** They contain drivers for communication between MCU and h-bridge and throttle position sensor (TPS) and MCU respectively.

5.2.2 Experimental setup

In this section the experimental setup used to perform the HIL simulation will be analyzed.

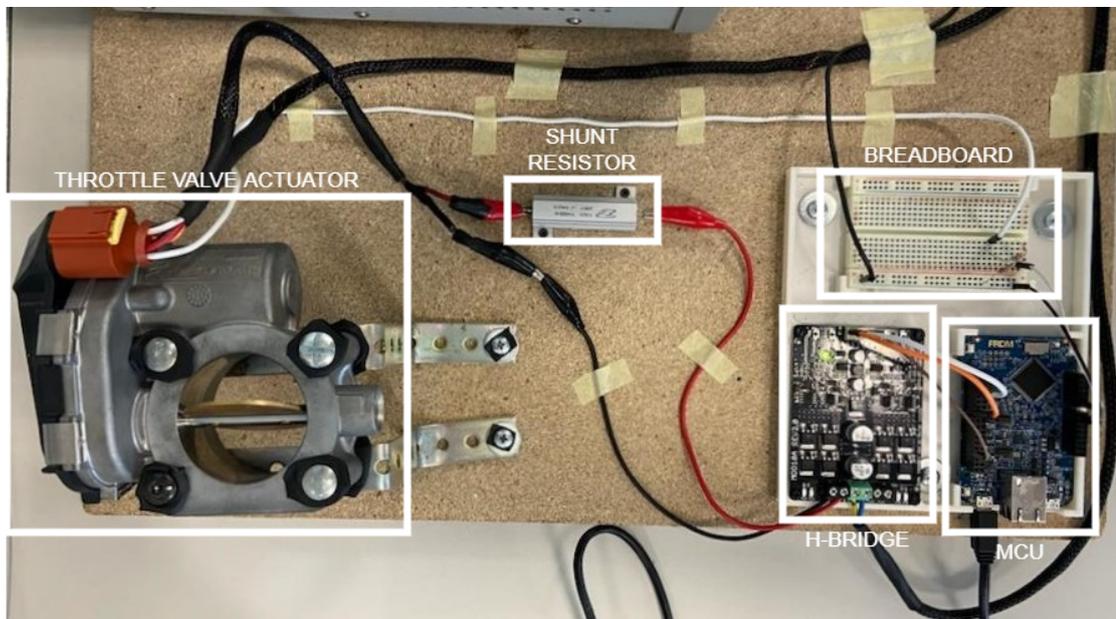


Figure 5.2: Experimental setup

In Figure 5.2 the test bench build is shown. In particular, the instrumentation that composes it are:

- The *MCU* receives and processes feedback data and according to control logic algorithm sends a PWM and a direction signal to the h-bridge.
- The *h-bridge* that amplifies the control signal in PWM from a value between $0 - 5V$ to a value between $\pm 15V$, to power the DC motor. It also allows an inversion of the current direction, i.e. an inversion of the plate motion, without changing the wiring.
- The *power supply* feeds at a constant voltage both h-bridge and TPS.
- The *breadboard* on which is implemented the voltage divider that linearly scales the voltage output of the TPS from $0 - 5V$ to $0 - 2.5V$, since the maximum voltage that the MCU can receive is $3.3V$ and receiving a voltage higher than this would result in position reading errors.
- The *shunt resistor* which is used only in the parameterization phase and it is used to measure the DC motor armature current. In control validation phase the shunt resistor has been short-circuited.

To a better understand of the components connections, a block scheme of the experimental setup is provided:

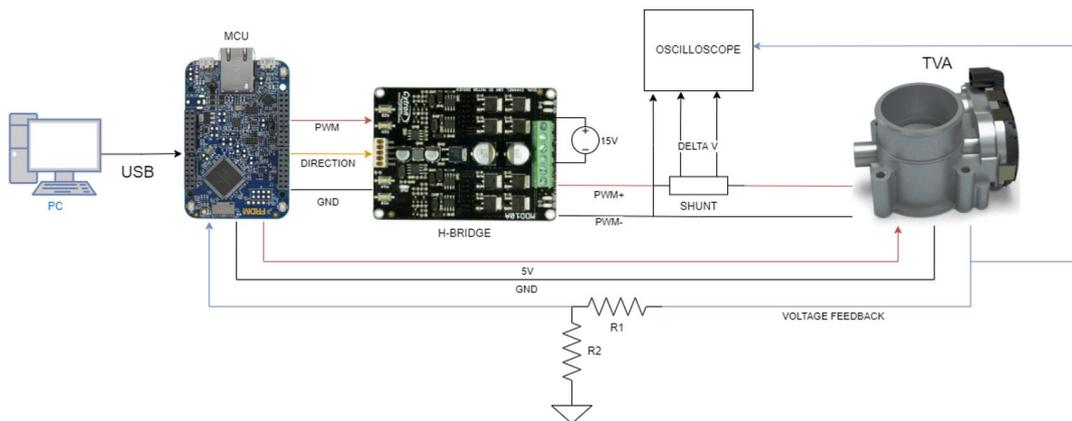


Figure 5.3: Experimental setup block diagram

5.3 Tests to assess the controllers performances

5.3.1 Step response

An explanation of the step response and its use in the validation of the control system was provided in Section 2.2.1 and Section 2.2.2 respectively. The section was added to make the argument smoother, provide details of the step signal used and the requirements used to validate the controls.

A graph of the step signal used is shown in figure below:

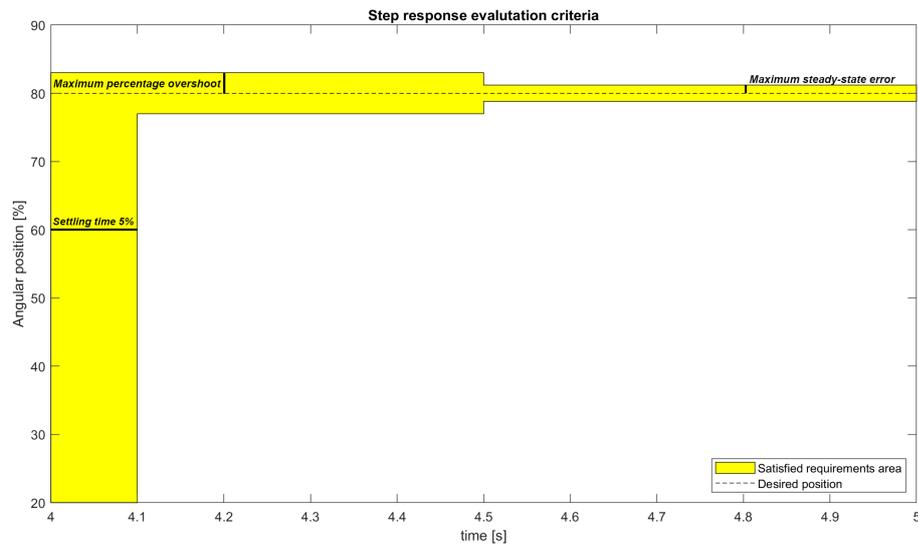


Figure 5.4: Step response parameters

As you can see from the Figure 5.4, the step is between a percentage position of **20%** and **80%**. This range was chosen to maintain safety margins from system hard stops and thus avoid system damage due to unexpected throttle valve overshoot.

The yellow zone of the graph is the area where the control system requirements are met. This zone is defined by three parameters:

- *Settling time 5%* = 100ms
- *Maximum percentage overshoot* = 5%.
- *Steady-state error* = 2%

Since the signal taken with the oscilloscope is disturbed, to more accurately consider the steady-state error, an average of the last 1000 measured values of the angular

position in the test was made, as the signal taken with the oscilloscope is disturbed. The step response is the minimum requirement to approve the developed control and gives an idea of the behavior of the system only in conditions of large-signal operating mode. A distinction between behavior in large-signal and small-signal operating mode is made with the second test, i.e. the system response to a drive cycle signal.

5.3.2 Drive cycle

We chose to call this test drive cycle response because stimulating the system with this type of signal is aimed at simulating different operating modes of the actuator on a moving vehicle. This, however, does not derive from data actually detected on a moving vehicle but has been elaborated virtually assembling different signals that aim to simulate different scenarios. A graph of the signal thus elaborated is shown in the following Figure:

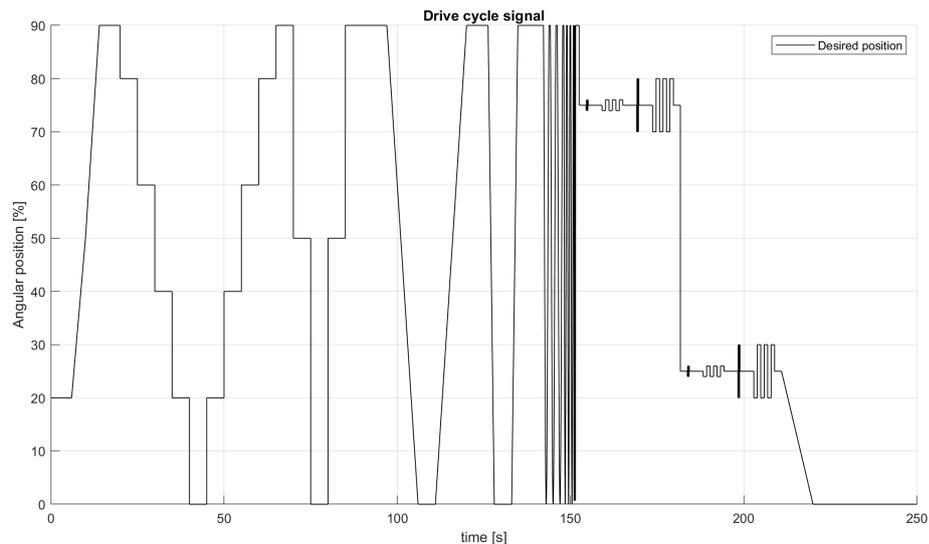


Figure 5.5: Drive cycle signal

As it can be seen from the Figure the signal consists of several sections. There are steps at different percentage position to stimulate the signal in both large-signal and small-signal operating mode, ramp signals to simulate a constant acceleration and deceleration of the vehicle and sine waves to assess the behaviour of the system at different frequencies.

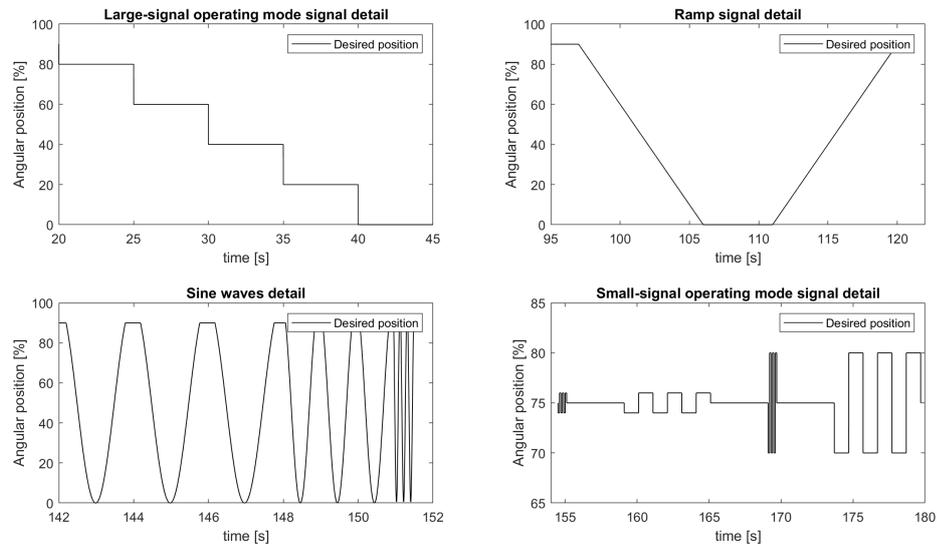


Figure 5.6: Drive cycle signals detail

The evaluation of control performance in small-signal operating is very important because it provides an indication of how well the control can cope with system nonlinearities. In this operating mode, in fact, the nonlinearities of the system, e.g. the static friction and the backlash of the gear system, acquire a great importance.

The observations made using this signal are not precise as in the case of step response but rather qualitative of the control behavior in the different situations imagined. It is also useful to make sure that the system does not go into instability if stimulated with a signal different from the classical step.

5.3.3 Robustness test

The objective of this test is to analyze the ability of the control technique to operate within the requirements even after a variation in the system parameters. The value of the parameters can vary for several reasons:

- Production uncertainties;
- Aging of the actuator components;
- Change in the environmental conditions;

For structural, costs and time reasons it was not possible to study in detail the variation of the parameters on the real system. For this reason, an alternative way of evaluating the robustness of control systems has been used.

The method used can be summarized in four steps:

1. Variation set definition.
2. Repopulation of linear model (Section 3.2.2) parameters with variation sets.
3. Step response of the closed-loop system with the reparameterized model and the block diagram of the control technique whose robustness has to be evaluated.
4. If the test is passed, the process is repeated starting from step 1 and generating the variation set with a higher standard deviation. In contrast, if the test fails, the highest standard deviation value is recorded.

A more detailed description of the process steps to execute the test will be provided but not before some considerations have been made:

- As the method is set it is not possible to associate any well-defined physical phenomenon to the variation of the parameters.
- The step response performance requirements for this test will be lower than the ones previously seen. This is because the objective of the test is not to evaluate the optimized behavior of the system but rather its correct functioning in case of deviation from the nominal conditions.

Parameters distribution through Monte Carlo analysis

For each parameter of the linear model, the variation around the nominal value is defined by a normal distribution, whose standard deviation is settled and the mean value is equal to zero.

Variation sets definition includes:

1. The random extraction of the parameter percentage of variation with respect to its nominal value within the normal distribution whose the standard deviation is defined.
2. The extraction is repeated to form a set of n values for each parameter, valid for a precise standard deviation. So, n sets of parameters will be randomly generated. All parameters vary at the same time. A set is composed by **1000** values.
3. The creation of the sets is repeated as the standard deviation increases, up to that imposed during the creation of the variation intervals.

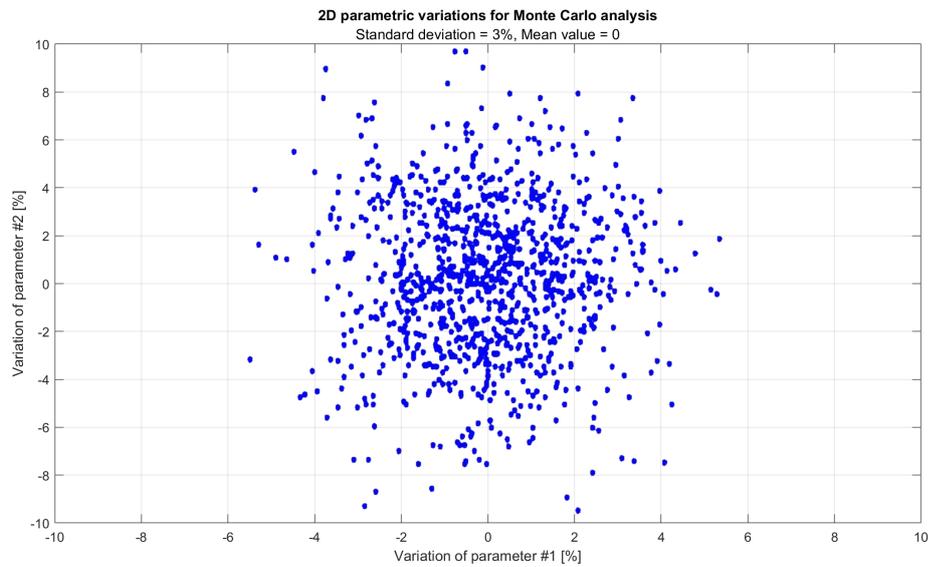


Figure 5.7: Example of parametric variation for Monte Carlo analysis

The model parameters that are altered are::

- Armature resistance;
- Armature inductance;
- Torque and speed constant;
- System inertia;
- Viscous friction coefficient;
- Coulomb torque;
- Static friction torque;
- Spring stiffness;
- Spring preload;

Step response of the closed-loop system with the reparameterized model

For each variation set, a system step response is performed to evaluate the output behavior against the control requirements.

To perform the step response the plate has been brought first to **20%** percentage position, then the test is performed with a step signal ranging from **20%** to **80%** percentage position. The new performance requirements are:

1. *Settling time 5% $T_s = 150ms$*
2. *Percentage overshoot = 5%*
3. *Maximum steady-state error $e_{ss} = 5\%$*

The three performance requirements define the so called "Satisfied requirements area", the yellow area in the graph of Figure 5.8.

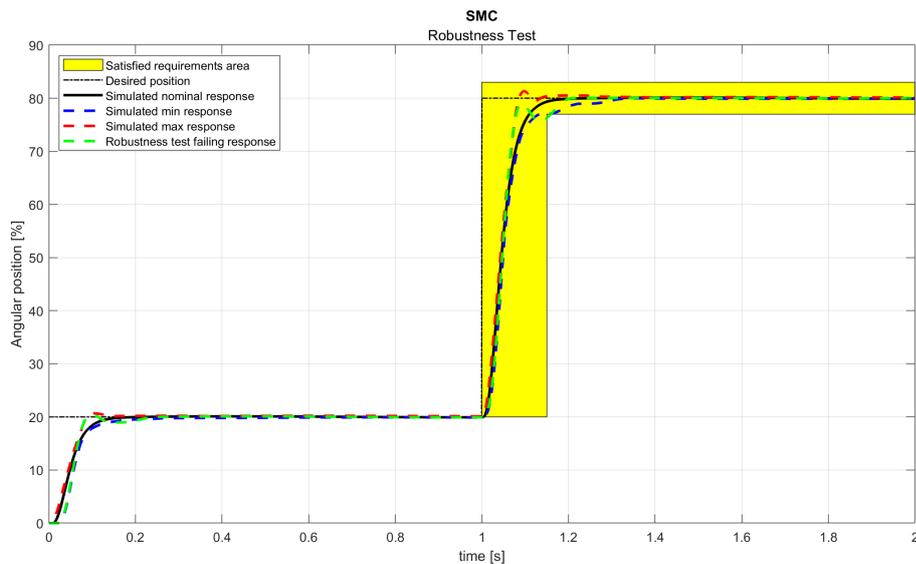


Figure 5.8: Example of robustness test step response

An example of the robustness test step response is shown in Figure 5.8.

Initially, the Monte Carlo analysis refers to a standard deviation of **1%**. For this standard deviation, 1000 step responses are made, each with a different set of parameters. This gives a homogeneous assessment of the variation associated with this standard deviation. The simulation continues by increasing the standard deviation of a unit until reaching a set of parameters which causes the failure of the test, i.e. the feedback curve exit the satisfied performance requirements area, it is represented by the green dotted line in Figure 5.8 that violate settling time requirement. The Figure 5.8 also shows a red dotted line, representing the maximum feedback excursion above the nominal value, and a blue dotted line, that is the maximum excursion of feedback below the nominal value.

Robust control classification

The classification of the most robust controls is based on the standard deviation characteristic of the parameter variation sets achieved during the simulations. The higher the standard deviation achieved for a control, the greater its robustness, this because it manages to operate in the requirements for a greater variation of the parameters.

5.3.4 Spring fault test

Another analysis that can be considered relevant with the concept of system robustness is the behavior of the system in the event of failure of the throttle valve return spring. Also in this case it was not possible to carry out the test physically removing the spring from the system because this would have led to a damage of the same. It was chosen, for this reason, to simulate the spring breaking bringing to zero its contribution of torque in the linear model.

In order to perform the test, the system is moved to the **50%** percentage position. At the time instant **2.5s** the torque contribution of the spring is set to zero and the behavior of the system is observed. The test is considered passed if the percentage position of the throttle valve remains within an acceptable error band between **49%** and **51%**. If the test is not passed, the time the control system takes to return the throttle plate to the acceptable error band is considered.

An example of the test response is shown in Figure 5.9, observing which it can be noticed that, as soon as the torque contribution is removed, the percentage position changes rapidly closing the plate. This is actually what it is expected, in fact, at the static equilibrium the controller drives the DC motor so that it contrasts exactly the spring torque that always tends to open the valve, but at the instant the spring torque became zero the DC motor torque contribution has no equal and opposite component and then it closes the plate.

At the same time, it can be notice that the control is aware of the error between the desired and the feedback position, so it reacts by returning the throttle plate to the set percentage position. In this situation the control will no longer be optimal because it was calibrated considering the spring contribution but it will allow a use of the vehicle at lower performance to for example to drive the vehicle in a safe place or to maintenance.

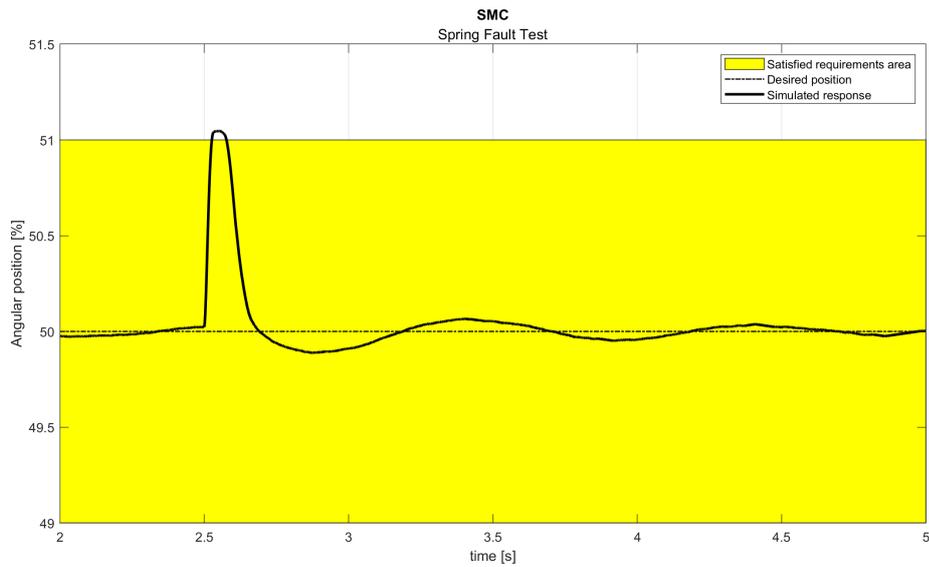


Figure 5.9: Example of spring fault test response

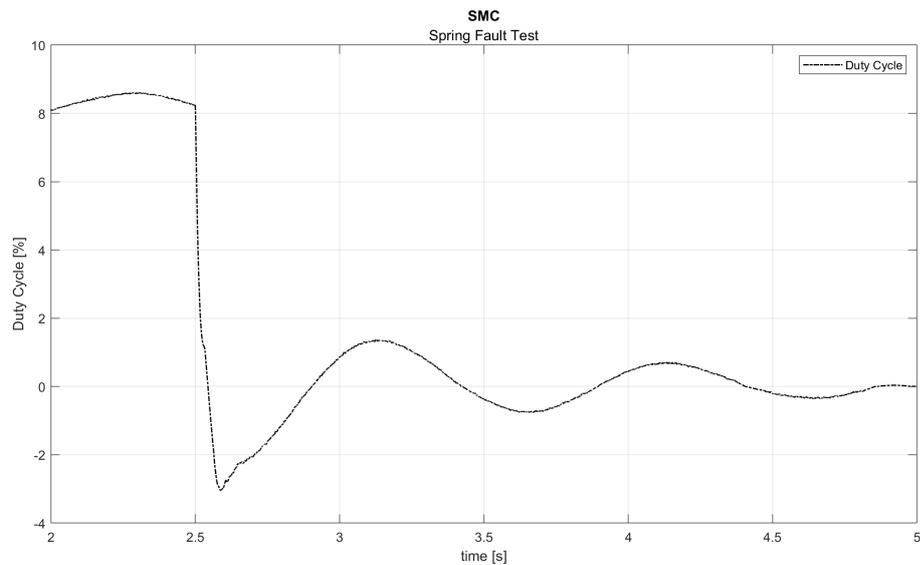


Figure 5.10: Example of spring fault test duty cycle response

The behavior described above is also confirmed by the graph that represents the duty cycle that is sent to the DC motor during the test, it is shown in Figure 5.10. It can be noticed, in fact, that the duty cycle before the spring break is positive,

it means that the DC motor is providing a torque that tends to close the valve. When the spring breaks, the duty cycle passes almost instantly to a negative value, indicating that the DC motor is providing torque to open the throttle valve and to return it to the set position. Another very important element to note is the almost zero duty cycle value once the steady-state is reached. This is justified by the fact that after the spring break the DC motor will no longer have to provide a torque to counteract the pull effect of the spring.

5.4 Obtained results

5.4.1 PID

Step response

The results of the step response test for PID control are shown in Figure 5.11 and in Table 5.1. All performance requirements are met and the behaviour of the simulated system represents accurately the behaviour of the real system.

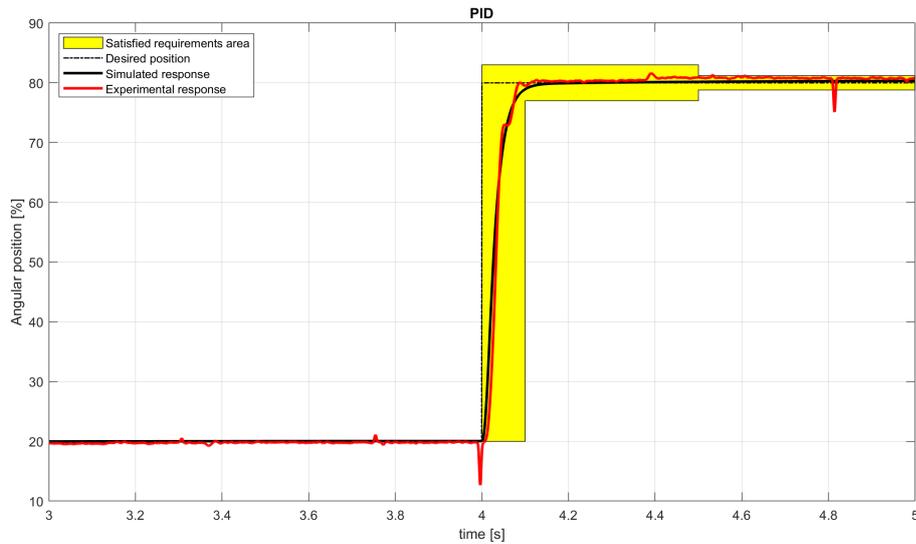


Figure 5.11: PID step response

	Simulated	Experimental
<i>Settling time 5% [s]</i>	0.0763	0.0745
<i>Percentage overshoot [%]</i>	0.0711	1.1263
<i>Steady-state error [%]</i>	0.2713	0.7219

Table 5.1: PID step response results

The graph in Figure 5.12 shows the time behavior of the duty cycle generated by the control and sent to the DC motor during the simulated step response. It can be noticed that the command control output saturates at the initial step time and remains constant for a small amount of time. This is due to the control action of the integrator anti-windup.

Proceeding with the analysis of the results, it will be noticed that the only control that saturates the control command output signal is the PID. The reason is to be found in the fact that, unlike the other control techniques, the control has been designed and validated with an additional resistance contribution in the DC motor armature circuit, i.e. the shunt resistor used for current measurement.

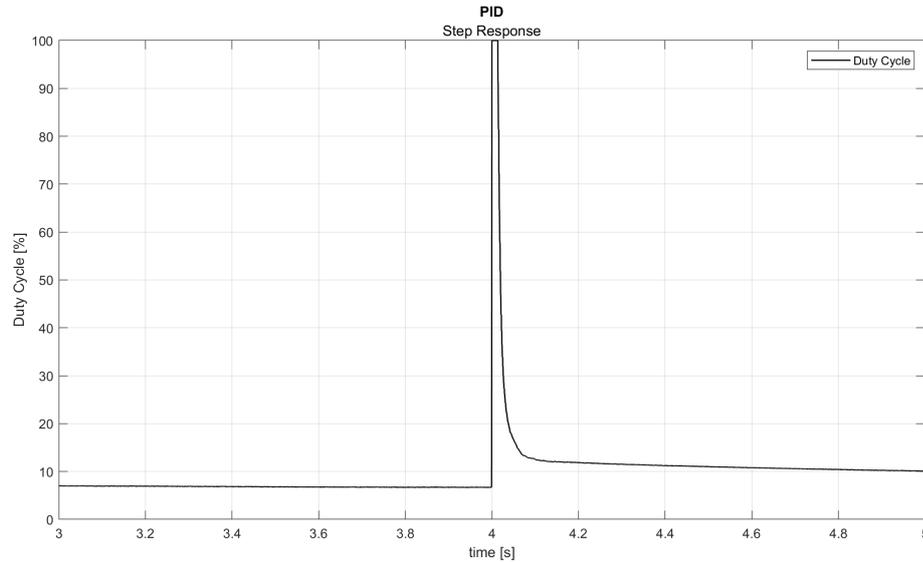


Figure 5.12: PID step response duty cycle

Drive cycle response

The three following figures represent the PID drive cycle and some of its details.

First of all, the simulated and experimental drive cycle response of the PID control are compared in Figure 5.13. The figure shows that the control system behaves very well in large-signal operating mode (time range: **20s – 90s**) and with the ramp input signals (time range: **90s – 140s**). Moreover, its behaviour is very accurate for high frequencies sine waves (time range: **140s – 180s**) input signals (Figure 5.14) while this is not true for all the designed controllers. This is an important aspect of the PID control technique because it means that the poles are well placed.

Going into detail, Figure 5.15 zooms on small-signal operating mode (time range: **180s – 210s** in Figure 5.13). As it can be seen, the control performances are degraded. This is because, as already pointed out, in this operating mode the effects of the nonlinear phenomena of the system becomes more relevant and the control fails to cope with the disturbances they produce.

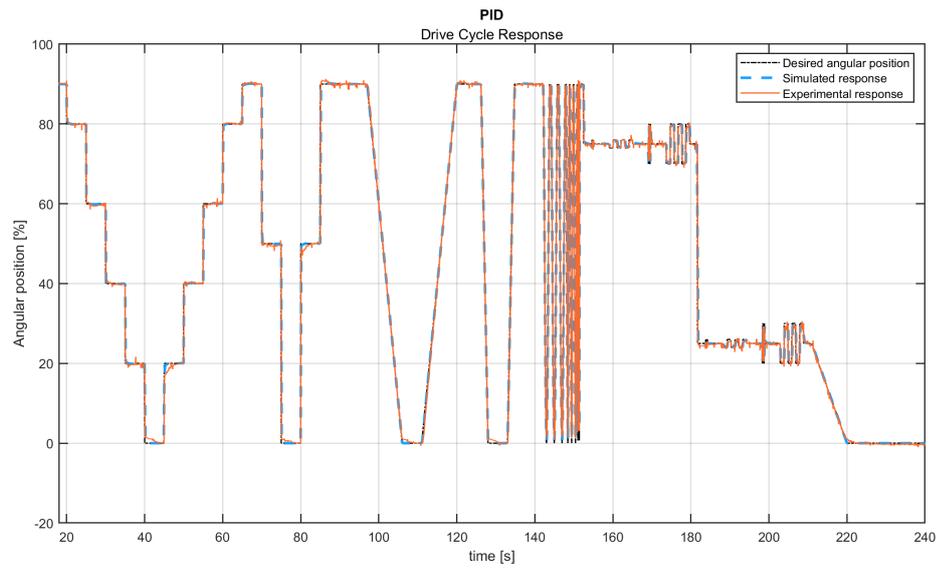


Figure 5.13: PID drive cycle response

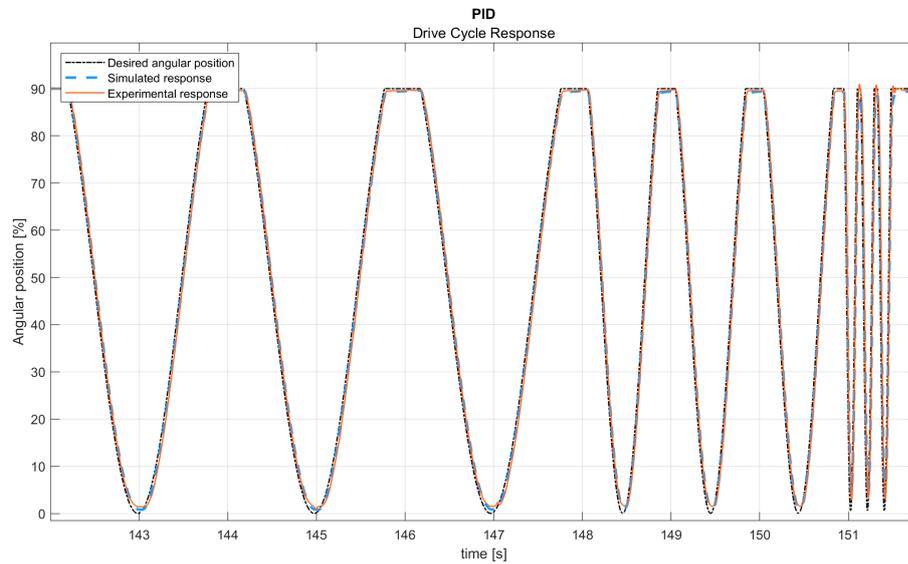


Figure 5.14: Detail of PID control system excited with sine waves at different frequency

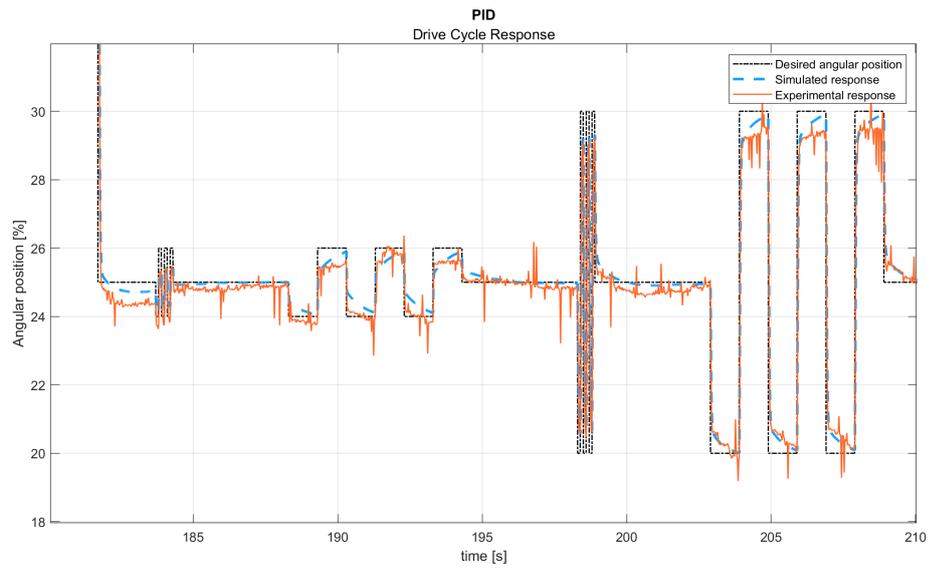


Figure 5.15: Detail of the PID drive cycle response in small-signal operating mode

Robustness test

A graphical representation of the results obtained with PID robustness test are shown in Figure 5.16.

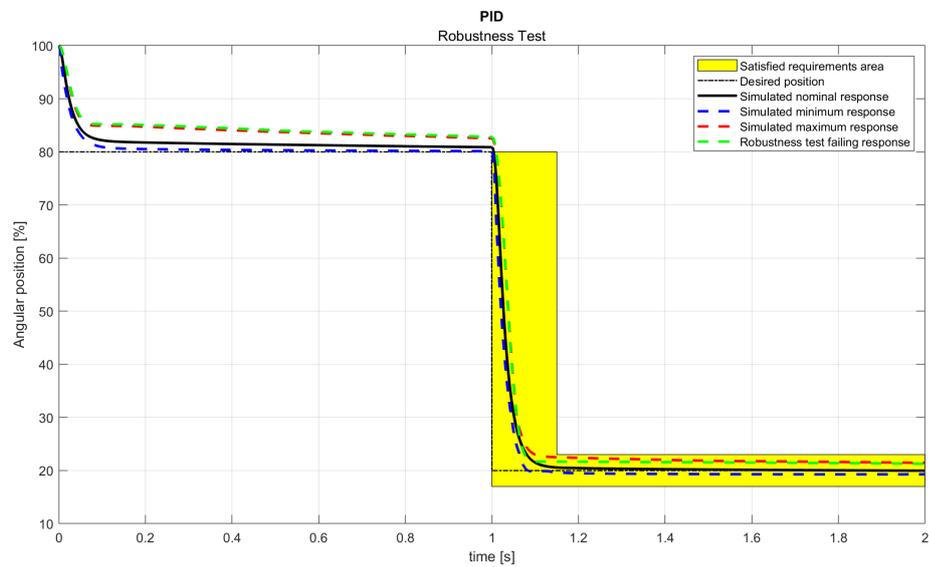


Figure 5.16: PID robustness test results

In this case, robustness test is failed when the PID fails to reach the steady state condition quickly enough during the first step signal so that there is a too high difference between the simulated signal with the nominal parameters (black line) and the simulated signal with varied parameters (green line) at the start time of the second test step.

The *Standard Deviation Index* obtained is equal to $\sigma\% = 18$, which is a better result than the one expected, especially, compared with the index gained with the nonlinear control techniques as shown in following sections.

Spring fault test

A graphical representation of the result obtained with PID spring fault test is shown in Figure 5.17.

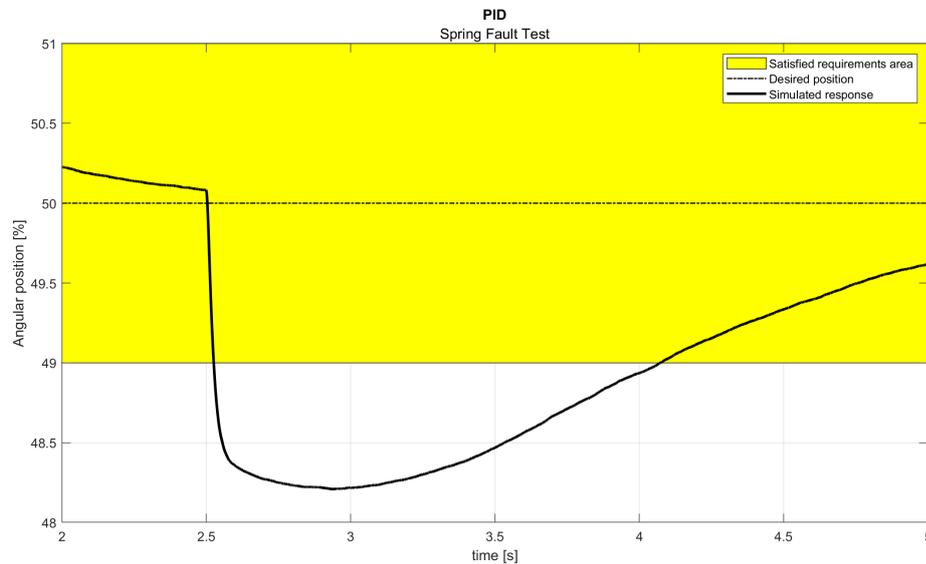


Figure 5.17: PID spring fault test response

Since after the time instant 2.5s the simulated response exceeds the yellow area, the test is considered **not passed**; the time the control system takes to return the throttle plate inside the acceptable error band (yellow area) is equal to $t_r = 1.57s$.

As expected, Figure 5.18 shows that, when the spring torque contribution is removed from the system, the duty cycle command signal tends to zero as the angular position tends to the steady state.

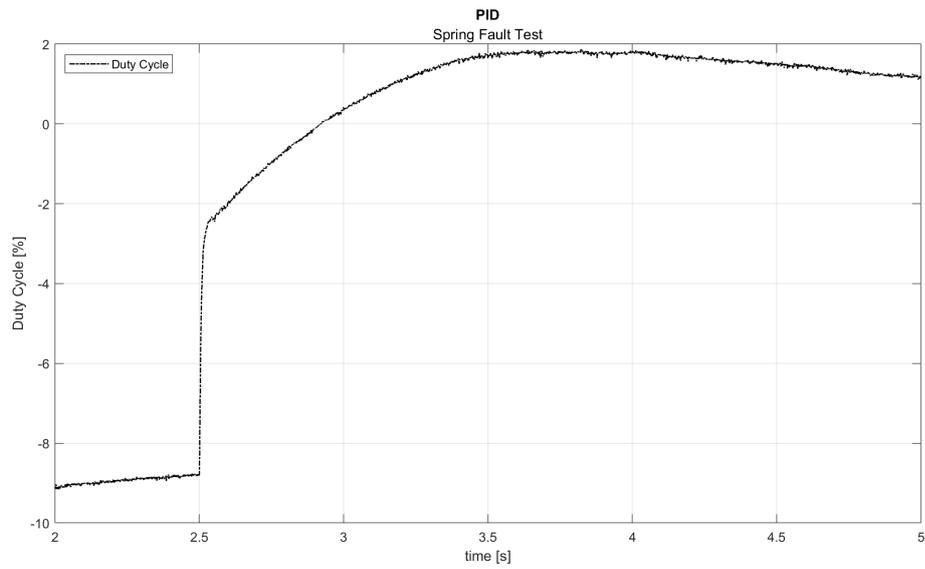


Figure 5.18: PID spring fault test response duty cycle

5.4.2 IPD

Step response

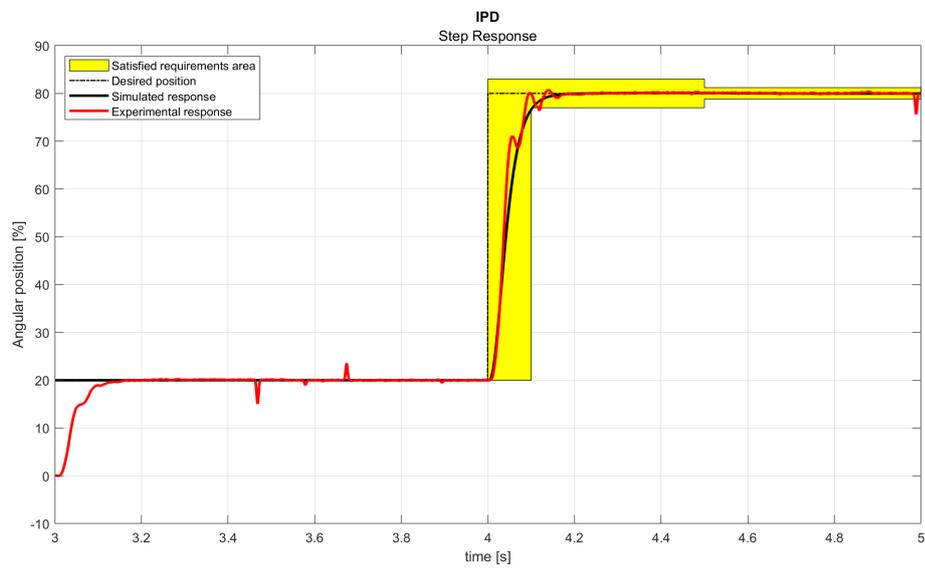


Figure 5.19: IPD step response

	Simulated	Experimental
<i>Settling time 5% [s]</i>	0.1030	0.0866
<i>Percentage overshoot [%]</i>	0.1527	0.8508
<i>Steady-state error [%]</i>	0.0311	0.0518

Table 5.2: IPD step response results

Figure 5.19 and Table 5.2 contain the results of the step response test for IPD control.

Not all performance requirements are met. In fact, the settling time 5% of the simulated response is slightly above its associated performance requirement (**0.1s**). On the other hand, this is not a big problem since the experimental response of the system respect all the performance requirements. The difference between simulated and experimental behaviors is due to IPD controller design which is performed with the reduced order linear model of the throttle valve that has a slower dynamic with respect to the real system.

Observing Figure 5.20 below, it can be assumed that the command control output signal doesn't saturate at the initial step time. Theoretically, this means that the controller can achieve better performance and can be optimized a little bit more. On the other hand, since the specifications are satisfied, the actuator can be controlled using less energy if the command control output signal does not saturate.

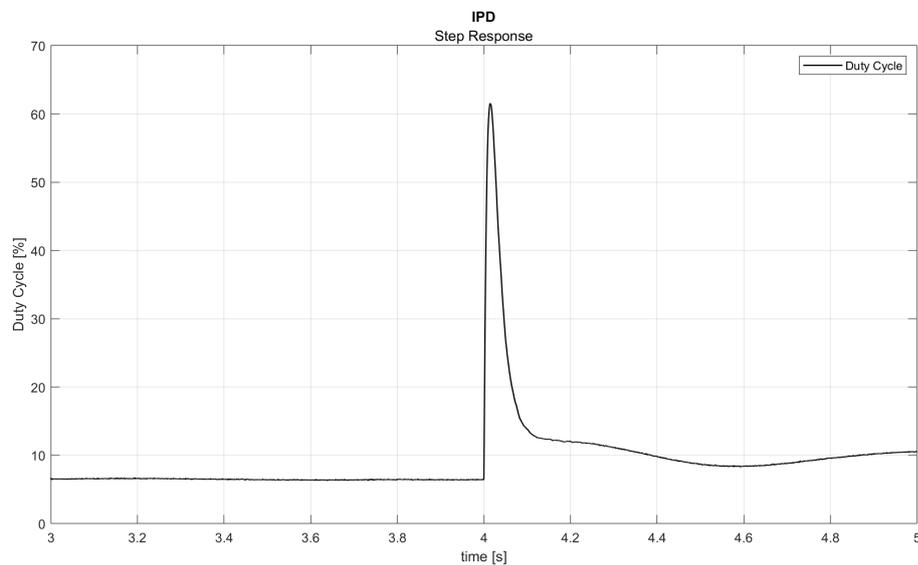


Figure 5.20: IPD step response duty cycle

Drive cycle response

The simulated and experimental drive cycle response of the IPD control are compared in Figure 5.21.

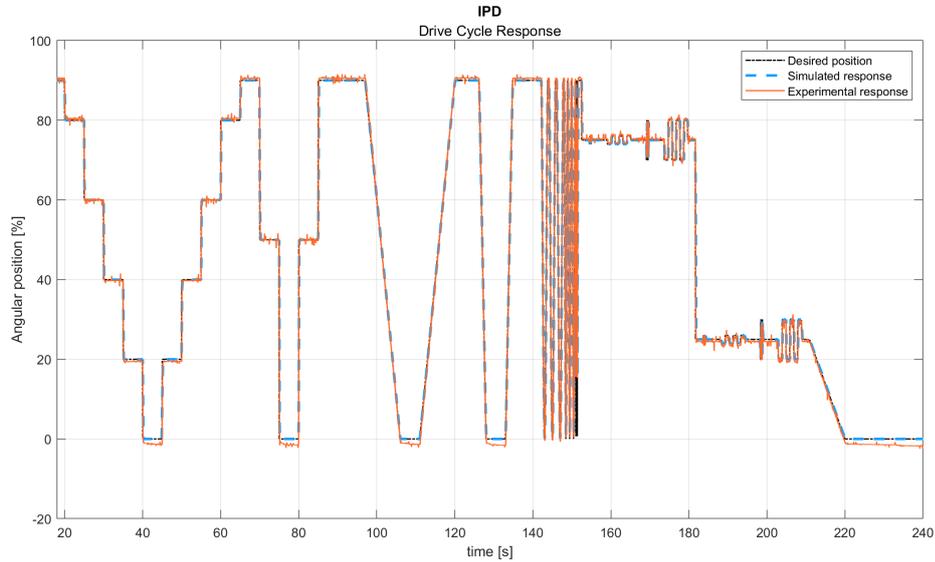


Figure 5.21: IPD drive cycle response

As shown in Figure 5.21, the desired, simulated and experimental drive cycle response are very close to each other throughout the test. Some imperfection can be noticed near the angular position of 0% , which corresponds to the hard stop upper limit. The error, in this case, may be due to potentiometer calibration: if the angular position value exceeds the calibration range, the acquired data could be incorrect.

Considering the the system behavior in small-signal operating mode, the controller reaches, theoretically, better performance with respect to PID controller which is appreciable from the simulated response represented in Figure 5.22. Unfortunately, due to system nonlinearities, the experimental response does not match well enough the simulated response in small-signal operating mode.

In Figure 5.23, it can be noticed that the control system behaves like a low-pass filter reducing the amplitude of the sine waves at higher frequency. Moreover, since the real system dynamic is faster than the one of the nonlinear model, the experimental response is more accurate in following the input sine waves with respect to the simulated one.

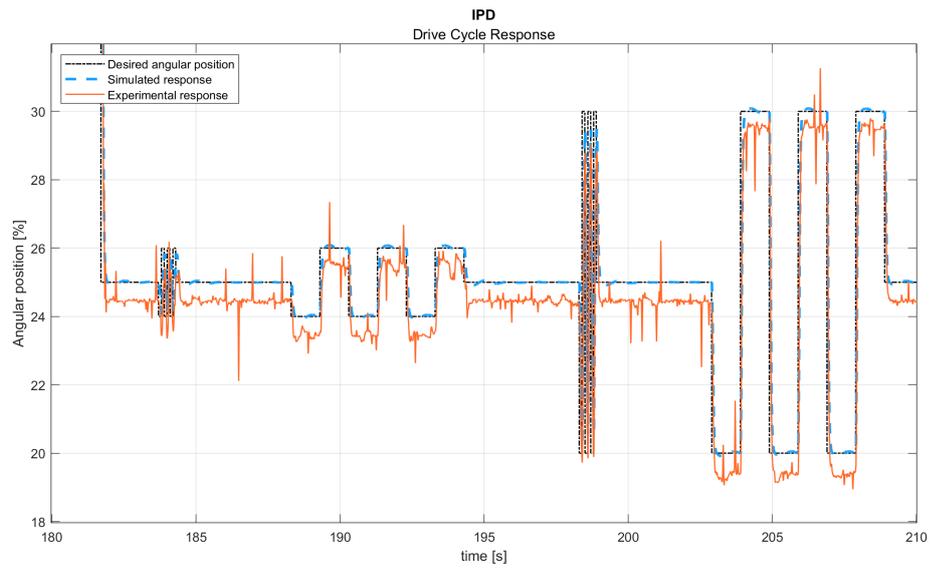


Figure 5.22: Detail of the IPD drive cycle response in small-signal operating mode

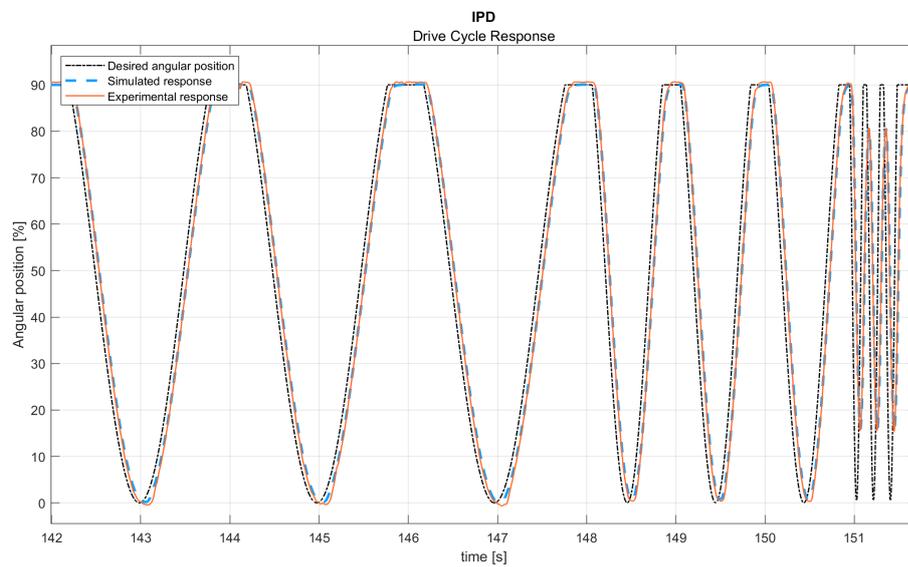


Figure 5.23: Detail of PID control system excited with sine waves at different frequency

Robustness test

A graphical representation of the results obtained with IPD robustness test are shown in Figure 5.16.

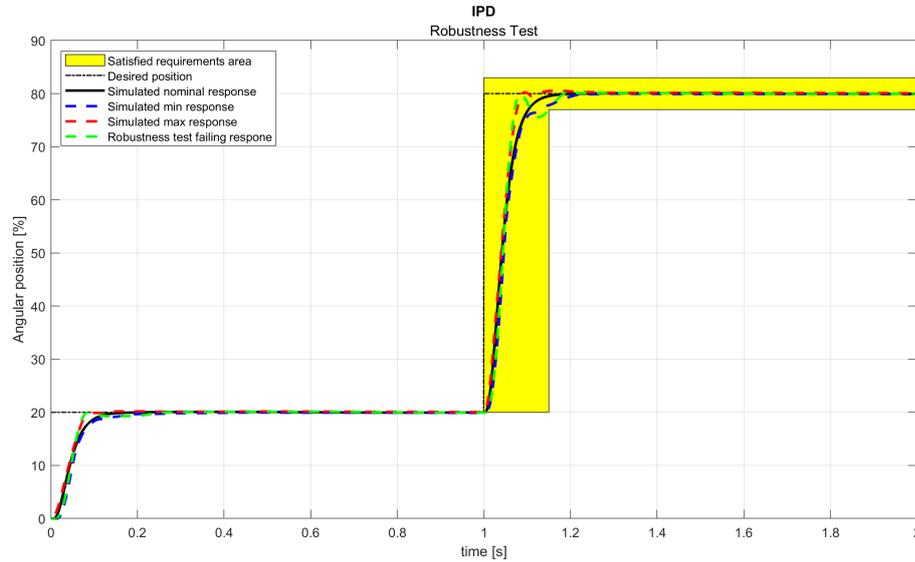


Figure 5.24: IPD robustness test results

In this case, robustness test is failed when the combination of the varied parameters bring the **settling time 5%** of the control system out the associated performance requirement.

The *Standard Deviation Index* obtained is equal to $\sigma_{\%} = 22$.

Spring fault test

A graphical representation of the result obtained with IPD spring fault test is shown in Figure 5.24.

The test is considered **passed** because the simulated response remains in the satisfied requirements area. The time the control system takes to return the throttle plate to the acceptable error band is equal to $t_r = 0s$.

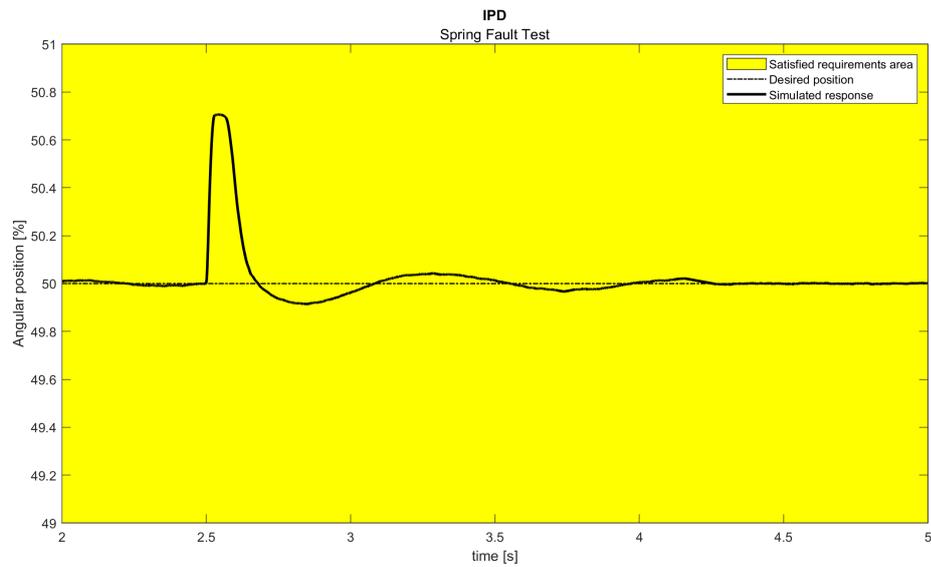


Figure 5.25: IPD spring fault test response

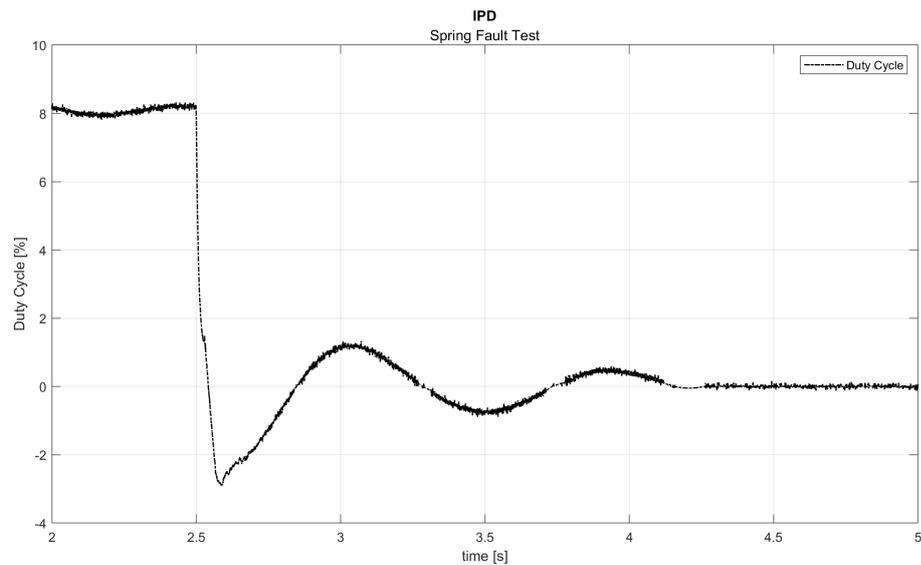


Figure 5.26: IPD spring fault test response duty cycle

From Figure 5.26, it can be noticed that, as expected, when the spring torque contribution is removed from the system, the duty cycle command signal tends to zero as the angular position tends to the steady state.

5.4.3 Discrete-time polynomial control

Step response

The results of the step response test for RST control are shown in Figure 5.27 and in Table 5.3.

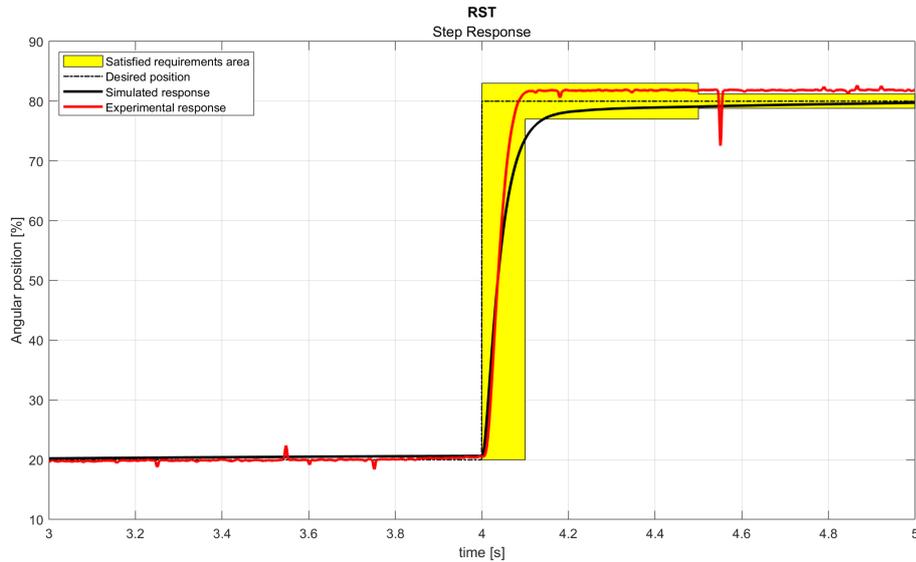


Figure 5.27: RST step response

	Simulated	Experimental
<i>Settling time 5% [s]</i>	0.1421	0.0732
<i>Percentage overshoot [%]</i>	<i>N.A.</i>	<i>N.A.</i>
<i>Steady-state error [%]</i>	0.5514	1.8450

Table 5.3: RST step response results

Not all the performance requirements are satisfied. In particular, a marked difference between the simulated and the experimental response behaviours can be observed. This leads to complications in the control validation, as the simulation is not reliable in order to predict the system behavior when the real system is tested. Moreover, the two behaviors are opposed: the simulated one is very slow and reaches the desired steady state position, while the experimental one is faster but the percentage overshoot and the steady state error are not acceptable. In particular, the percentage error can't be computed since the system doesn't reach the desired steady state.

Figure 5.28 shows that the control action saturates at the initial step time which justifies the low control system settling time.

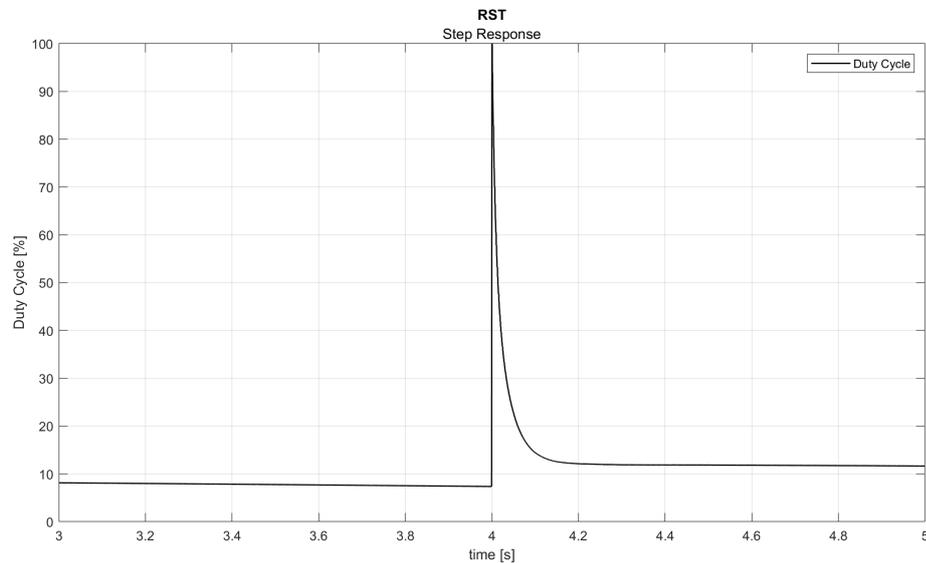


Figure 5.28: RST step response duty cycle

Even if the control has very high transient performances, the control was deemed to have failed to perform its function. The control technique should be redesigned starting from the beginning, perhaps considering the linear model instead of the reduced order linear one. Due to delivery time reasons, this was not possible to be performed. It was preferred to concentrate the remaining time on designing the nonlinear control techniques, i.e. linear state feedback and sliding mode controls.

5.4.4 Linear state feedback control with state observer

Step response

Figure 5.29 and Table 5.4 show the results of the step response test for LSF control with state observer.

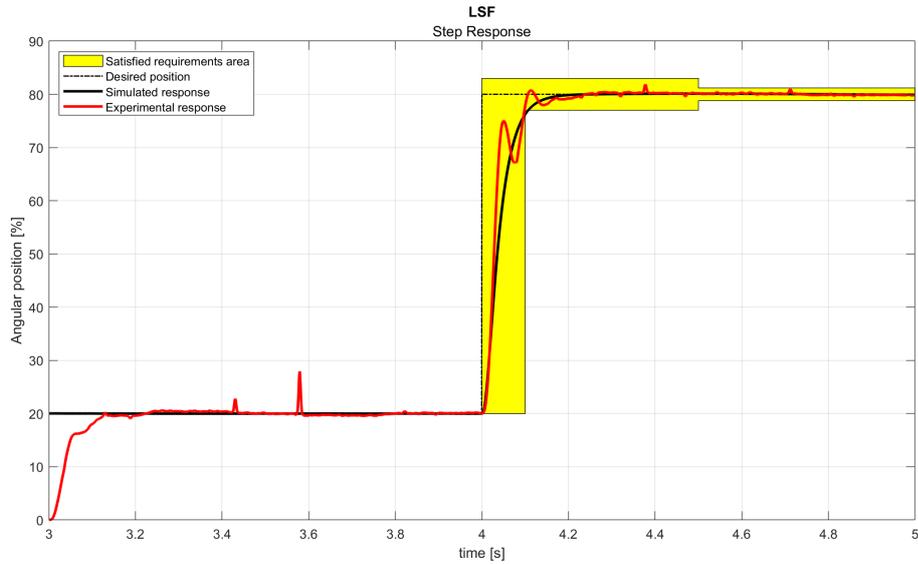


Figure 5.29: LSF with state observer step response

	Simulated	Experimental
<i>Settling time 5% [s]</i>	0.1068	0.0981
<i>Percentage overshoot [%]</i>	0.1226	2.3358
<i>Steady-state error [%]</i>	0.0324	0.0206

Table 5.4: LSF with state observer step response results

Even in this case, not all performance requirements are met. The settling time 5% of the simulated response is slightly above its associated performance requirement, which is not a big problem since the experimental response of the system respect all the performance requirements. The difference between the simulated and experimantal behavior depends on the the different dynamic of the reduced order linear model of the throttle valve, used for LSF controller design, and the real plant.

The simulated duty cycle control command output, as shown in Figure 5.30, reaches almost **80%** at the initial step time. This means that the controller performances are not optimised in terms of power consumption since the IPD and the boundary layer SMC reach almost the same performances but with a lower amount of duty cycle request.

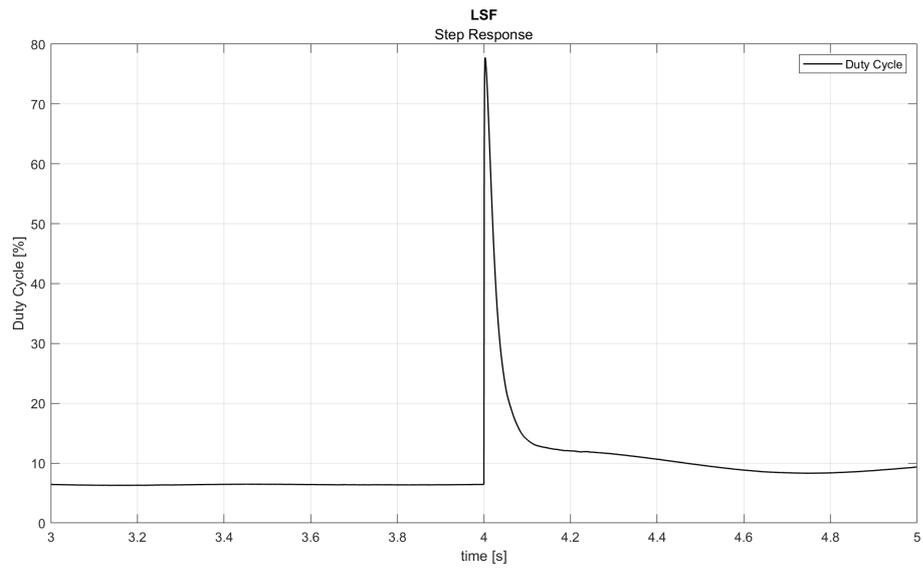


Figure 5.30: LSF step response duty cycle

Drive cycle response

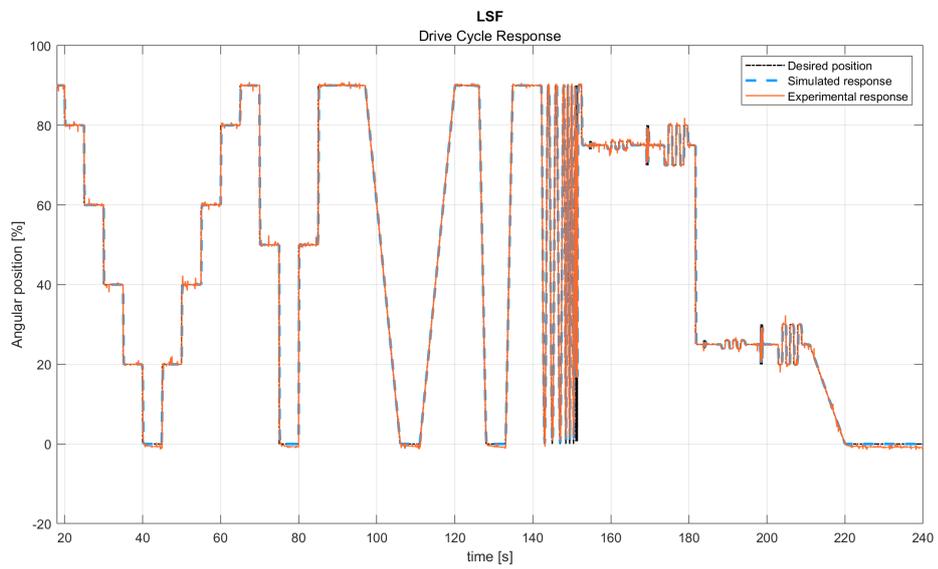


Figure 5.31: LSF with state observer drive cycle response

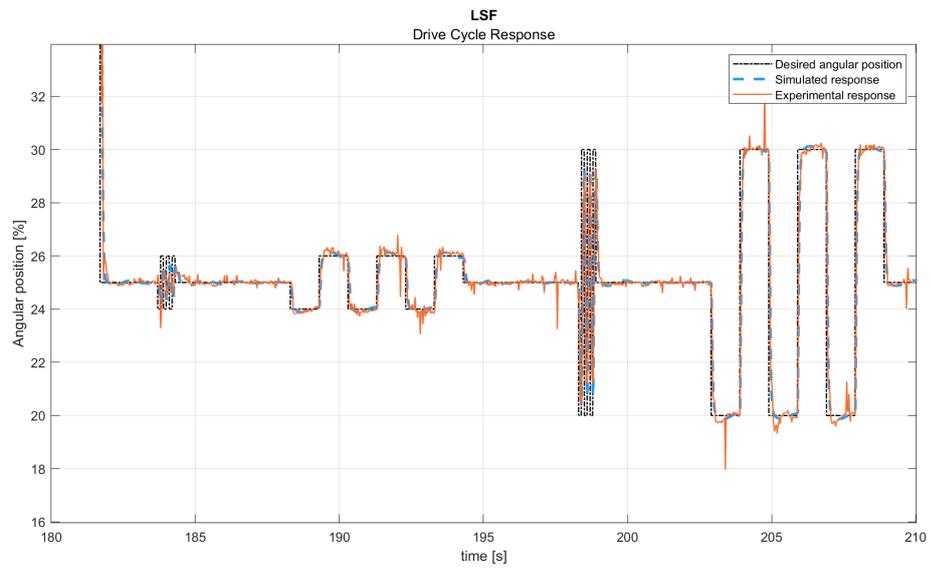


Figure 5.32: Detail of the LSF with state observer drive cycle response in small-signal operating mode

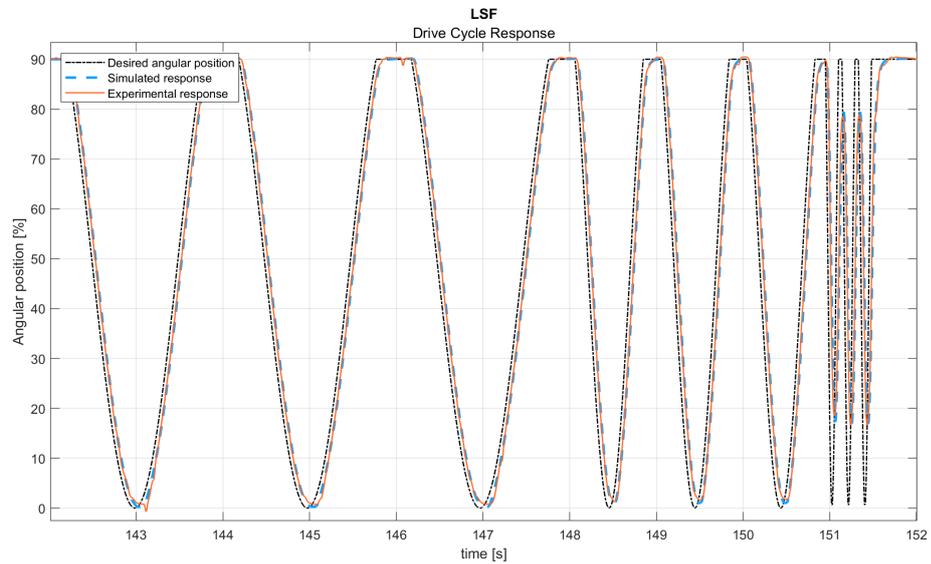


Figure 5.33: Detail of the LSF control with state observer excited with sine waves at different frequency

The simulated and experimental drive cycle response of the LSF control with state observer are compared in Figure 5.31.

In general, it can be noticed how the control system behaves very well in all the operating modes that the drive cycle signal excite. In particular, it is very accurate for small-signal operating mode, as can be seen in Figure 5.32.

Excited with sine waves the control system behaves like a low-pass filter and it can be noticed from Figure 5.33.

Robustness test

A graphical representation of the results obtained with LSF control robustness test are shown in Figure 5.34.

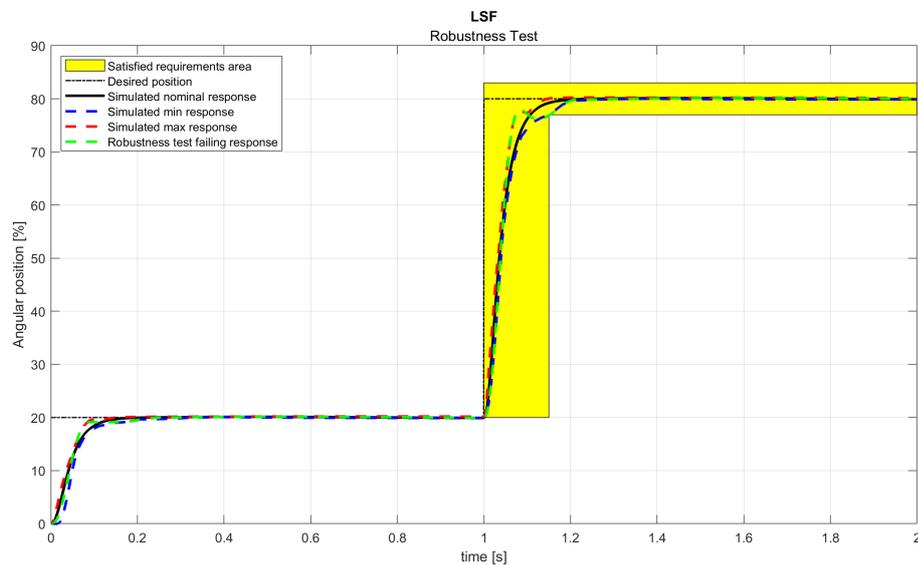


Figure 5.34: LSF control with state observer robustness test results

In the shown case, robustness test is failed since the combination of the varied parameters bring the **settling time 5%** of the control system out the associated performance requirement (green line).

The *Standard Deviation Index* obtained is equal to $\sigma\% = 15$.

Spring fault test

A graphical representation of the result obtained with LSF control spring fault test is shown in Figure 5.34.

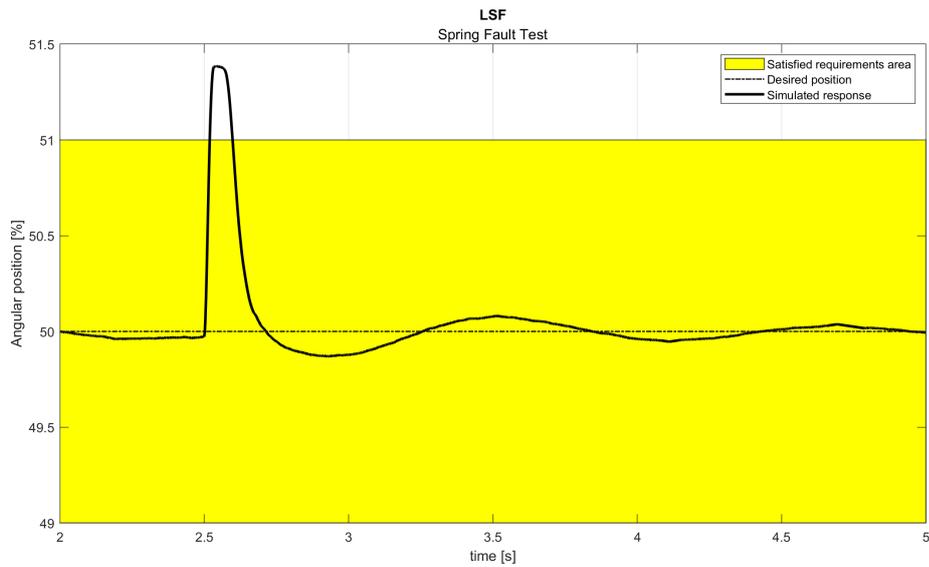


Figure 5.35: LSF control with state observer spring fault test response

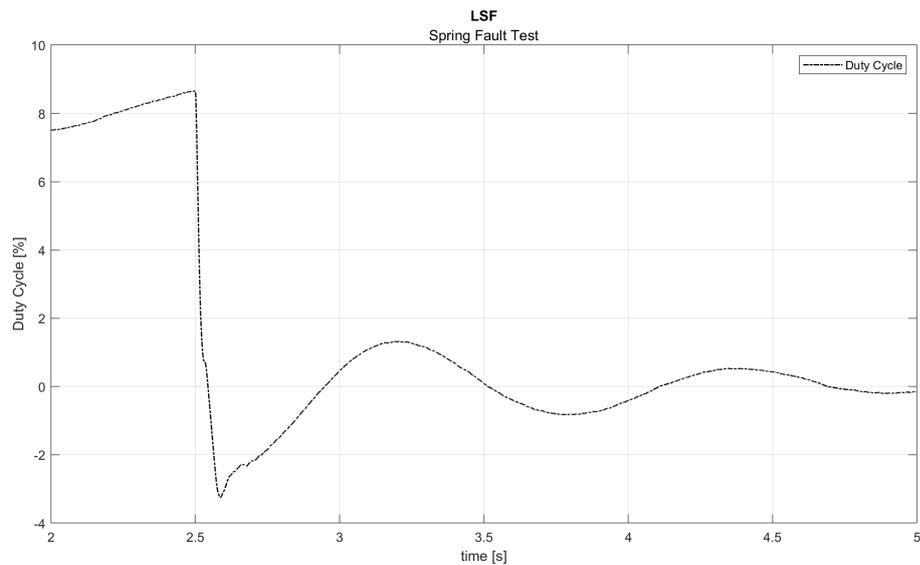


Figure 5.36: LSF control with state observer spring fault test response duty cycle

Observing Figure 5.35, it is evident that the test can be considered **not passed** and the time the control system takes to return the throttle plate to the acceptable error band is equal to $t_r = 0.09s$, which is less than the system settling time

requirement. For this reason, even if the test is not passed, the behavior of the control system in case of spring fault is considered acceptable.

As occurred for PID and IPD, Figure 5.36 shows that even for LSF with state observer control technique, when the spring torque contribution is removed from the system, the duty cycle command signal tends to zero as the angular position tends to the steady state.

5.4.5 Boundary layer sliding mode control

Step response

The results of the step response test for boundary layer SMC are shown in Figure 5.37 and in Table 5.5.

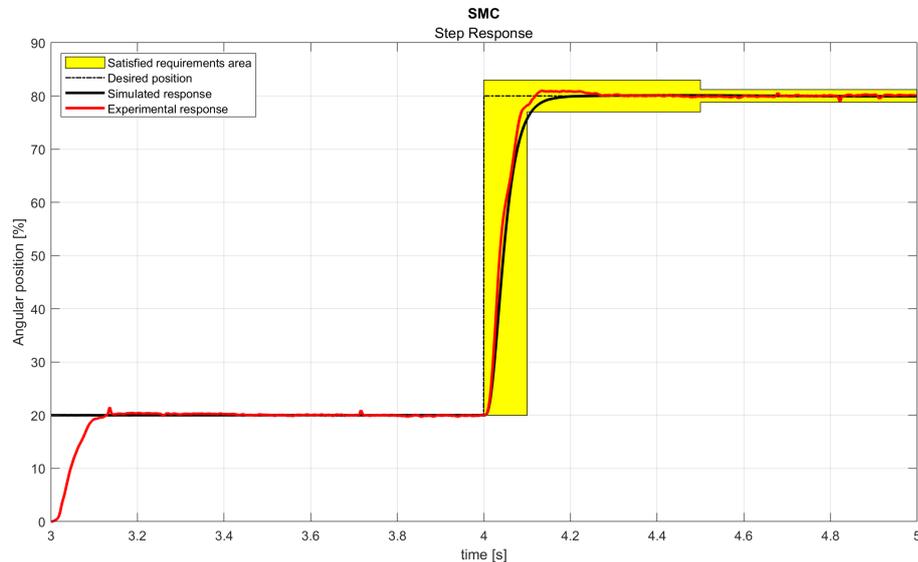


Figure 5.37: Boundary layer SMC step response

	Simulated	Experimental
<i>Settling time 5% [s]</i>	0.1092	0.0867
<i>Percentage overshoot [%]</i>	0.1575	1.2466
<i>Steady-state error [%]</i>	0.009	0.0243

Table 5.5: Boundary layer SMC step response results

Not all performance requirements are met, in fact, the settling time 5% of the simulated response is slightly above its associated performance requirement. On

the other hand, this is not a big problem since the experimental response of the system respect all the performance requirements.

The simulated duty cycle control command output, as it can be noticed in Figure 5.38, reaches almost **60%** at the initial step time. For almost the same performance, this is the control strategy that requires the minimum amount of duty cycle command control output.

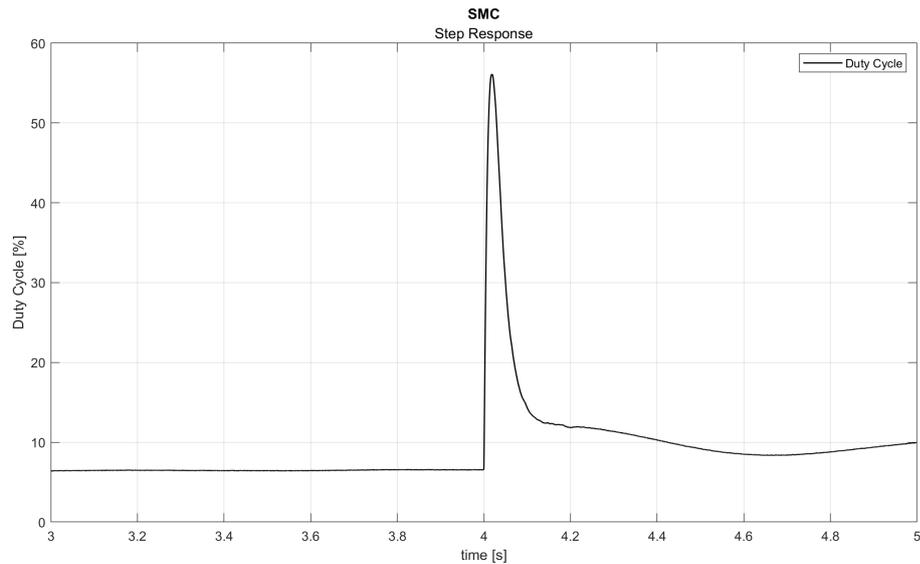


Figure 5.38: Boundary layer SMC step response duty cycle

Drive cycle response

The simulated and experimental drive cycle response of the boundary layer SMC are compared in Figure 5.39.

The control system behaves very well in all the operating modes that the drive cycle signal excite. In particular, it is very accurate for small-signal operating mode, as can be seen in Figure 5.40.

Excited with sine waves, the control system behaves like a low-pass filter as can be noticed in Figure 5.33.

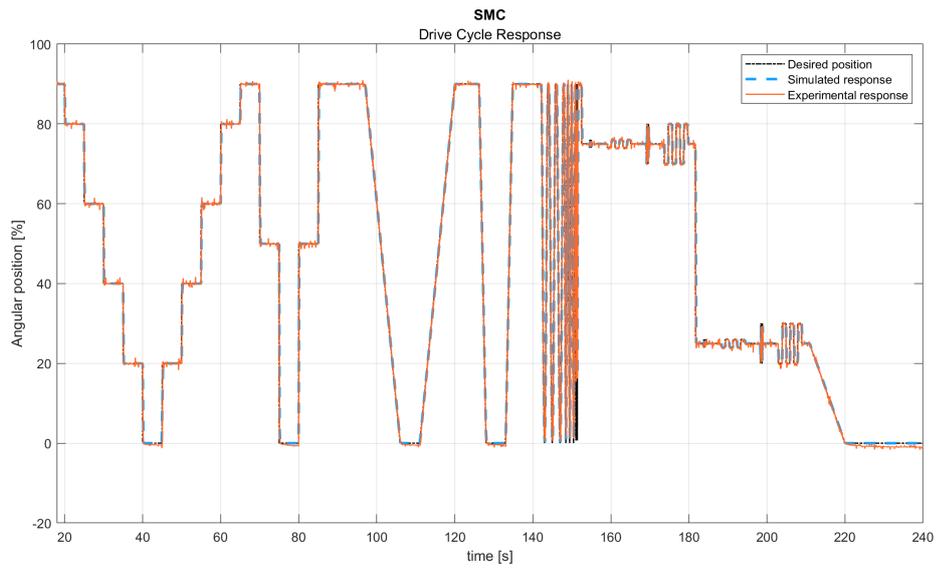


Figure 5.39: Boundary layer SMC drive cycle response

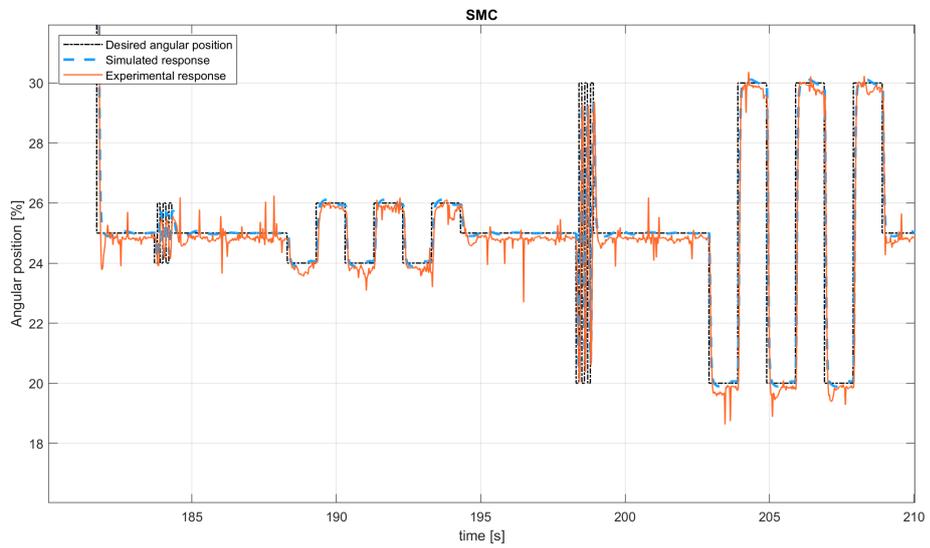


Figure 5.40: Detail of the boundary layer SMC drive cycle response in small-signal operating mode

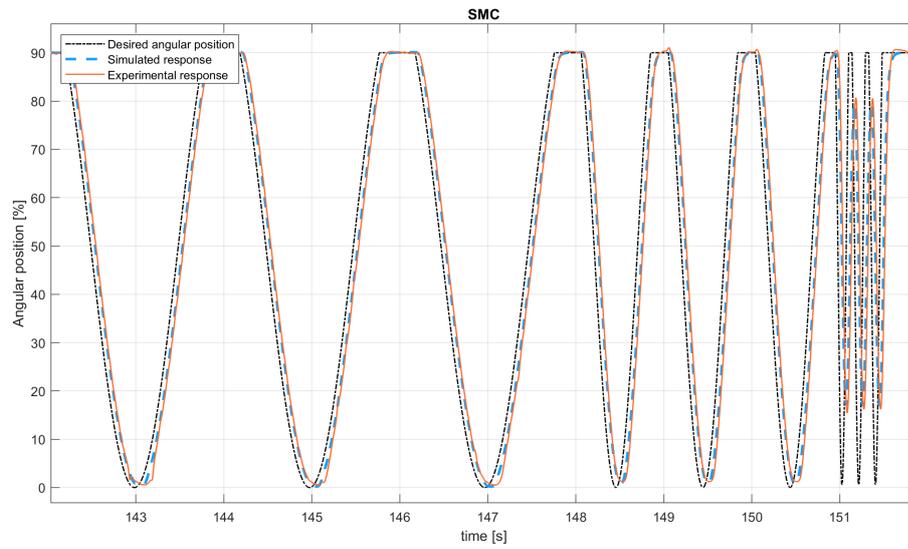


Figure 5.41: Detail of the boundary layer SMC control system excited with sine waves at different frequency

Robustness test

A graphical representation of the results obtained with boundary layer SMC robustness test are shown in Figure 5.42.

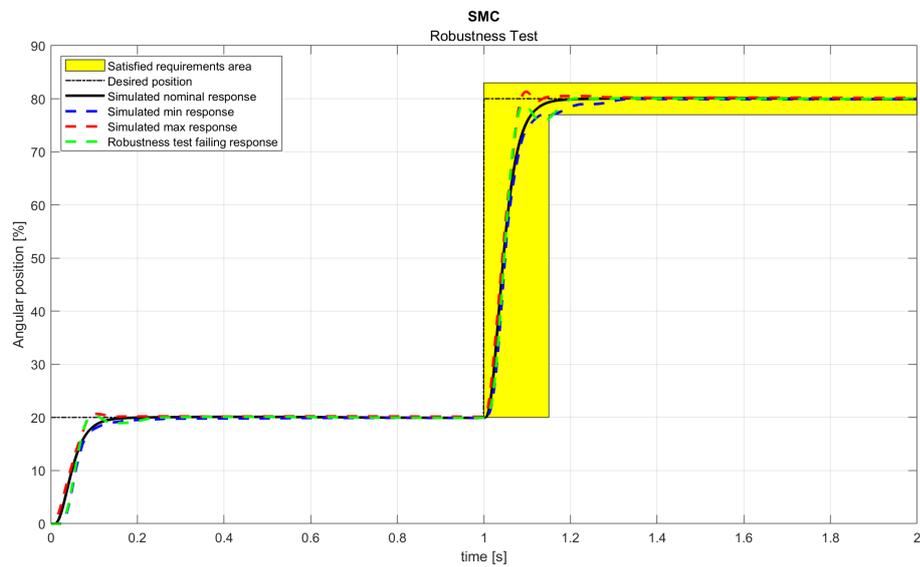


Figure 5.42: Boundary layer SMC robustness test results

As the previous control technique analyzed, boundary layer SMC robustness test is failed since the combination of the varied parameters bring the settling time **5%** of the control system out the associated performance requirement.

The *Standard Deviation Index* obtained is equal to $\sigma\% = 21$.

Spring fault test

A graphical representation of the result obtained with boundary layer SMC spring fault test is shown in Figure 5.42.

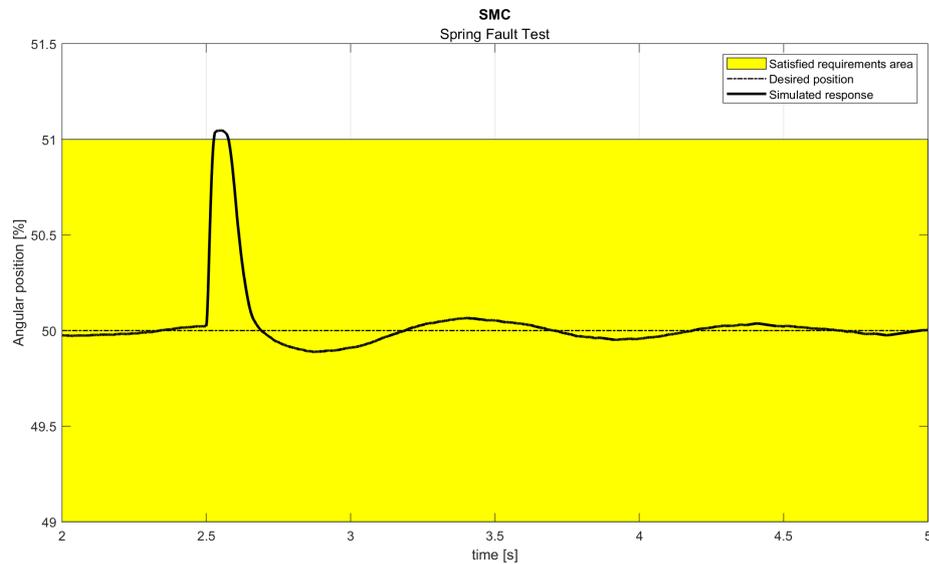


Figure 5.43: Boundary layer SMC spring fault test response

Observing Figure 5.43, it is clear that the test is **not passed** and the time the control system takes to return the throttle plate to the acceptable error band is equal to $t_r = 0.07s$. Because the system settling time requirement is higher than this value, even if the test is not passed, the behavior of the control system in case of spring fault is considered acceptable.

From Figure 5.44, it can be noticed that, as expected, the duty cycle command signal, when the spring torque contribution is removed from the system, tends to zero as the angular position tends to the steady state.

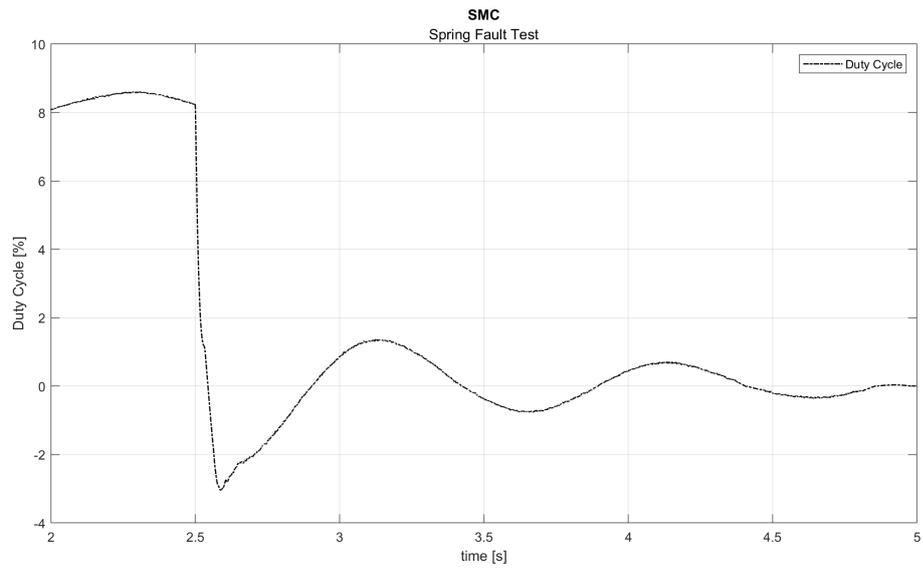


Figure 5.44: Boundary layer SMC spring fault test response duty cycle

5.5 Experimental results comparison

A summary table of simulated and experimental obtained results is provided below:

	PID	IPD	RST	LSF	SMC
<i>Settling time 5% [s]</i>	0.0745	0.0866	0.0732	0.0981	0.0867
<i>Percentage overshoot [%]</i>	1.1263	0.8508	-	2.3358	1.2466
<i>Steady-state error [%]</i>	0.7219	0.0518	1.845	0.0206	0.0243
<i>Standard deviation index [%]</i>	18	22	0	15	21
<i>Spring failure time [s]</i>	1.57	0	2.19	0.09	0.07

Table 5.6: Summary table of the obtained results

Simulated settling time 5%, percentage overshoot and steady state error of the different control systems are not included in Table 5.6 because considered not relevant in this phase of the analysis.

Discrete time polynomial control results are shown only in order to give a complete view of the results data. However, this technique will be not further analyzed since its results do not match the set requirements.

In order to classify the different control techniques, starting from the obtained data a points based system is been elaborated. In particular, specific thresholds were considered in order to divide the performance results in "not acceptable" (NA), "acceptable" (A) and "good" (G), to which 0, 1 and 2 points were associated, respectively. The following tables show how the points were assigned to each performance requirement. The only exception is represented by the standard deviation index. In this case, the associated point is the normalization of the index itself (5.1).

1. *Settling time 5%*

Condition	Threshold	Point
NA	>0.1s	0
A	<0.1s AND >0.08s	1
G	<0.08s	2

2. *Percentage overshoot*

Condition	Threshold	Point
NA	>2%	0
A	<2% AND >1%	1
G	<1%	2

3. *Steady state error*

Condition	Threshold	Point
NA	>1%	0
A	<1% AND >0.1%	1
G	<0.1%	2

4. *Normalized standard deviation index*

$$\sigma_{\%,norm} = \frac{2}{MaxValueReached} \sigma_{\%} \quad (5.1)$$

5. *Spring fault test*

Condition	Threshold	Point
NA	Not Passed AND $t_r > 0.1s$	0
A	Not Passed AND $t_r < 0.1s$	1
G	Passed	2

The table below (Table 5.7) shows the score assigned to each control technique for each performance requirement. Then, for each control technique the points are summed and the total score is registered.

	PID	IPD	RST	LSF	SMC
<i>Settling time 5% [s]</i>	2	1	2	1	1
<i>Percentage overshoot [%]</i>	1	2	-	0	1
<i>Steady-state error [%]</i>	1	2	0	2	2
<i>Standard deviation index [%]</i>	1.63	2	-	1.36	1.9
<i>Spring failure time [s]</i>	0	2	0	1	1
SUM	5.63	9	2	5.36	6.9

Table 5.7: Summary table of the obtained results with point assigned

Then, control techniques are rearranged based on obtained scores:

IPD	9
SMC	6.9
PID	5.63
LSF	5.36

Table 5.8: Control techniques scores

Because of its insufficient performances, RST control technique is not considered.

Since Table 5.8, the following conclusions can be derived: the highest score is achieved by IPD control technique, which presents good transient performances combined with high control system robustness. Furthermore, this is the only analyzed control technique which has passed the spring fault test. Comparable results are achieved by the boundary layer SMC technique. What it is important to highlight is that PID and LSF have gained quite high similar scores. Since LSF control technique is more complex than PID, the latter is preferable for the considered application.

The below Figure 5.45 shows the difference between the simulated and experimental step response signals.

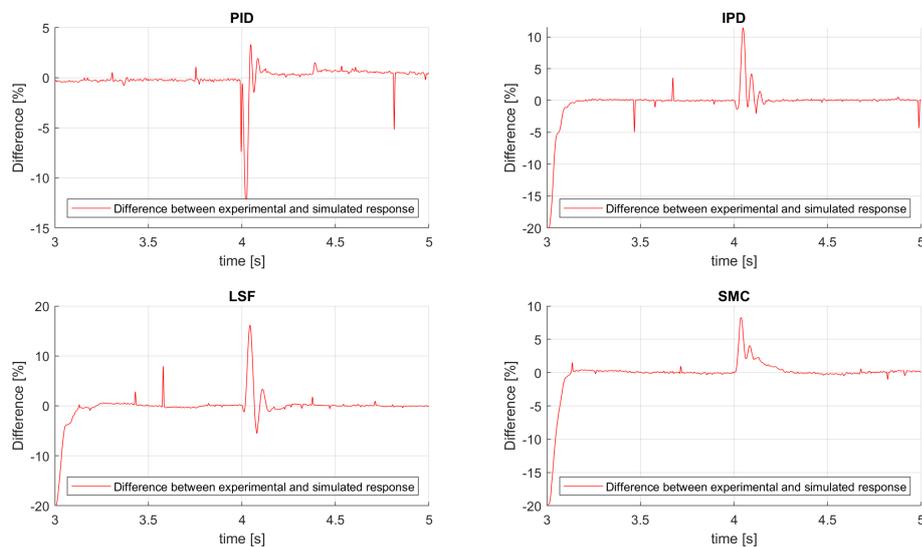


Figure 5.45: Difference between experimental and simulated step response of the different implemented control strategies

The difference between the model and the real system depends on the accuracy with which the model represents the physical system and on the real system parameters changes with respect to the nominal ones. Since the accuracy of each model is the same, as their design is, the control robustness is affected only by the difference between nominal model parameters and real system "varied" ones. Then, Figure 5.45, which represents the error between the real and the simulated systems, indicates the robustness of the control system.

The above Figure 5.45 also confirms results shown in Table 5.7: SMC and IPD are the most robust control techniques implemented.

The applied classification is based on the considered application, the throttle valve. For this reason, it has not to be considered absolute.

Chapter 6

Conclusions

The purpose of the study, since technological improvements allow it, is the design of the control of an electronic throttle control system that enhances the quality of the control system and its performances. Then, the control techniques that can be used in vehicle power trains in order to control the electronic throttle valve actuator have been analyzed to understand if it was possible to improve the current state of the art constituted by the PID control. This leads to two major objectives:

1. The definition of an easy to adapt robust control design process;
2. The design of a robust controller, specific for the considered system.

The control design process was defined step by step following the model-based control design guidelines. Starting from these, four control techniques, i.e. IPD, RST, LSF and SMC, were developed and compared with the most common and used, PID control technique.

A test bench to validate the different control strategies was built and the results obtained from the validation tests were used to compare and classify the different control techniques.

The results demonstrated that all control techniques are suitable for the considered application, except for the polynomial control which did not meet the requirements. Moreover, it was observed that the IPD and the boundary layer SMC controllers gained the better performances, both in terms of robustness and transient response. However, since the PID control has a simpler architecture and good performances, it remains a good solution for the electronic throttle control system.

Appendix A

Manson's gain rule

Mason's Gain Formula is a formula that uses the signal flow graph to determine the transfer function of a linear system. It was used to derive the closed-loop transfer function of the control systems.

$$T(s) = \frac{Y_{out}}{Y_{in}} = \frac{1}{\Delta} \left[\sum_{k=1}^N P_k \Delta_k \right] \quad (\text{A.1})$$

Where:

- P_k = k -th forward path
- Δ_k = $1 -$ (Gain of non-touching loops of the k -th forward path)
- Δ = $1 -$ (Sum of individual loops gain) + (Non-individual loops gain)

Bibliography

- [1] Wikipedia. *Drive by wire* — *Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/w/index.php?title=Drive%20by%20wire&oldid=1066247655>. [Online; accessed 07-September-2022]. 2022 (cit. on p. 1).
- [2] Carlo Rossi, Andrea Tilli, and A. Tonielli. «Robust control of a throttle body for drive by wire opening of automotive engines». In: *Control Systems Technology, IEEE Transactions on* 8 (Dec. 2000), pp. 993–1002. DOI: 10.1109/87.880604 (cit. on p. 1).
- [3] Robert Loh, Witt Thanom, Jan Pyko, and Anson Lee. «Electronic Throttle Control System: Modeling, Identification and Model-Based Control Designs». In: *Engineering* 05 (Jan. 2013), pp. 587–600. DOI: 10.4236/eng.2013.57071 (cit. on p. 6).
- [4] Jacob Pedersen. «Model Based and Robust Control Techniques for Internal Combustion Engine Throttle Valves». PhD thesis. Dec. 2013, pp. 25–29 (cit. on pp. 6, 7, 9, 10, 19, 24, 39).
- [5] J. Deur, Danijel Pavković, Nedjeljko Peric, and M. Jansz. «An Electronic Throttle Control Strategy Including Compensation of Friction and Limp-Home Effects». In: vol. 40. July 2003, 200–206 vol.1. ISBN: 0-7803-7817-2. DOI: 10.1109/TIA.2004.827441 (cit. on p. 9).
- [6] Wikipedia. *Model-based design* — *Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/w/index.php?title=Model-based%20design&oldid=1081446169>. [Online; accessed 16-September-2022]. 2022 (cit. on p. 13).
- [7] Stephen Dodds. «Settling time formulae for the design of control systems with linear closed loop dynamics». In: (Jan. 2008) (cit. on p. 38).