UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO DE ENGENHARIA INDUSTRIAL ELÉTRICA/ELETRÔNICA

ROBERTA ESPINDOLA VIEIRA

ADVANCED SIMULATION FOR CALIBRATION FOR RAIL PRESSURE GOVERNOR

FINAL PROJECT

CURITIBA
2011

ROBERTA ESPINDOLA VIEIRA

ADVANCED SIMULATION FOR CALIBRATION FOR RAIL PRESSURE GOVERNOR

Final Project of undergraduate course presented to the Industrial Engineering Electrical / Electronics program from Academic Department of Electronics – DAELN - Federal University of Technology - Paraná (UTFPR), as a partial requirement for the Engineer title.

Supervisor: Keiko Verônica Ono Fonseca
Local Supervisor: Alexander Pawlow

CURITIBA
2011

DEDICATION

This project is dedicated to my parents, Marcos and Rozilda, who taught me to fight for my dreams, who always gave me support to go ahead and, especially, who always wiped my tears, reach out and taught me to start over no matter how difficult could be. Without them this project would not be possible.

ACKNOWLEDGEMENTS

Dr. Keiko Verônica Ono Fonseca has been the ideal supervisor. Her advices and specially her encouragement helped me to develop this project and not give up even in the midst of many difficulties. I would also like to thank Alexander Pawlow who was always available to answer my questions and to help me to always find a new way to go ahead.

RESUMO

O *Common Rail System* (CR) é um sistema de injeção direta de combustível diesel em que a geração de pressão e a injeção são funções separadas, diferente dos outros sistemas de injeção. Hoje em dia o processo de calibração do CRS é executado manualmente, exigindo um alto nível de conhecimento de calibração de veículos e a repetição de inúmeros passos se algum erro é cometido. Além disso, não é possível definir um processo exato que leve a uma calibração considerada ótima. Por esse motivo, o principal objetivo deste projeto é desenvolver um método para o controlador do *Rail Pressure* que leve a melhor calibração possível, economizando custos e tempo. Para o desenvolvimento deste método é necessário primeiro um estudo de viabilidade e conhecimento detalhado sobre a teoria de controladores. Em seguida, a *Design of Experiment* (DoE) será usado. A DoE é um método utilizado para modelagem de sistemas desconhecidos baseados em dados, cujo objetivo é avaliar e otimizar o comportamento do sistema. O passo seguinte será executar as medições com um veículo utilizando a Unidade de Controle do Motor (UCM) e o programa INCA 6.2. Em seqüência, é necessário implementar um programa em MatLab para desenvolver critérios para a otimização. O resultado do calculo dos critérios desenvolvidos serão usados como entrada para o *Advanced Simulation for Calibration, Modeling and Optimization* (ASCMO) – ferramenta desenvolvida para modelagem de sistemas desconhecidos baseados em medições do comportamento de entrada/saída do sistema – e com este modelo é possível determinar o melhor parâmetro para controlador, achando assim a melhor calibração para o veículo. O sistema utilizado consistirá de uma UCM modelo EDC17, que será usada em um Citroen C4 (uma plataforma de demonstração, conectada através de *Controller Area Network* (CAN) e *Emulator Test Probe* (ETK). É necessário desenvolver o código para calcular os critérios e uma correta estratégia. A estratégia completa e a calibração devem ser feita em um segundo carro para concluir os benefícios deste novo método. Os resultados esperados são redução de tempo através da automação e DoE, assim como evitar retrabalho e reduzir custos. Para finalizar, é importante ressaltar que o trabalho de otimização e medição pode ser separado, portanto as medidas podem ser feitas por alguém sem conhecimento detalhado em calibração e apenas o processo de otimização necessitará de pessoal especializado.

Palavras-chave: *Rail Pressure*, Teoria de controladores, Calibração com simulação avançada

ABSTRACT

The CRS (Common Rail System) is an accumulator fuel-injection system where the pressure generation and the injection are decoupled in comparison to other injection systems. Nowadays CRS calibration process is performed manually: a good knowledge of calibrating the vehicle is needed and numerous steps needed to be repeated when an error is made. In addition, it’s not possible to define a process that leads to an optimum calibration. Hence, the main goal of this project is to develop a complete method to reach the best possible calibration by saving costs and time for the Rail Pressure Governor. In order to develop this system it is necessary to conduct a feasibility study (to be sure about the measurements reproducibility) and to get a deep knowledge about the governor theory first. The next step applies the Design of Experiment (DoE), a method for data-based modeling of unknown systems, with the goal to evaluate and optimize the system’s behavior. After that, measurements with the vehicle using the Engine Control Unit (ECU) and INCA 6.2 will be performed. Then, it is necessary to implement software to develop the criteria for an optimization in Matlab. The results of the criteria calculation will be the feed of ASCMO (Advanced Simulation for Calibration, Modeling and Optimization) – a tool for modeling the input/output behavior of unknown systems based on measuring data. With the model generated with ASCMO it is possible to determine the best parameters for the controller and, finally, find a good calibration for the vehicle. The system will consist of an ECU model EDC17 which will be used in a Citroen C4 - platform demonstrator, connected by a Controller Area Network and an Emulator Test Probe. A code needs to be developed in order to calculate the criteria and create a proper strategy. A complete test program and a calibration have to be done in another similar vehicle after the calibrating parameters in order to validate the results and conclude about the benefit of this new method. The expected results should save time through automation and DoE, as well as prevent reworks and reduce costs. At last, it is important to emphasize the fact that the work in optimization and measurement can be divided, so the measurement can be done by someone without a deep knowledge in calibration and just the optimization will need qualified knowledge.

Keywords: Rail Pressure, Governor Theory, Advanced Simulation Calibration

ABBREVIATIONS AND ACRONYMS

AccPed	Accelerator pedal
ASAP	Standardization of application systems
ASC	Advanced Simulation for Calibration
ASCMO	Advanced Simulation for Calibration, Modeling and Optimization
CAN	Controller Area Network
CCP	CAN Calibration Protocol
CPC	Coupled Pressure Control
CRI	Common Rail Injector
CRS	Common Rail System
DGS-EC	Diesel Gasoline System - Electronic Controls
DI	Direct Injection
DoE	Design of Experiment
D-SERAP	Dynamic Serial Application with additional programming
ECU	Engine Control Unit
EDC	Engine Diesel Control
EDOR	Experimental Design for Operating Range
ESB	Engineering SW Base system
IDI	Indirect Injection
INCA	INtegrated Calibration and Acquisition Systems
JTAG	Joint Test Action Group
MeUn	Metering Unit
PCV	Pressure Control Valve
PID	Proportional, integral and derivative
RAM	Random Access Memory
RMSE	Root Mean Square Error
ROM	Read Only Memory
RPG	Rail Pressure Governor
RP	Reference Page
SERAM	Serial Application with Additional Memory
SERAP	Serial Application
TDC	Top Dead Center
UIS	Unit Injection System
UPS	Unit Pump System
WP	Working Page

INDEX

1 INTRODUCTION

Nowadays the calibration of a Rail Pressure control is very expensive: it needs a long time and specialized people, and besides that, the calibration process is manually made. Trying to reduce the calibration time investigating new methods could be a solution to reduce time and costs of this calibration. Some studies about Advanced Simulation Calibration (ASC) had already been done before and the ASC using Design of Experiment (DoE) method is already successfully applied in other control systems. But nothing was doing yet about trying to use this method in the rail pressure control.

Hence, the motivation of this project was to investigate the use of ASC/DoE method to the rail pressure control. So, with duration of one year, a project started to be executed at BOSCH Group in the Diesel Gasoline System - Electronic Controls/ Engineering SW Base system (DGS-EC/ESB4).

If the results were applicable, they can bring many benefits for the company: through automation and DoE, this project can save time, prevent reworks and reduce costs. Besides that, it will be possible to divide the current process in optimization and measurement, so the measurement can be done for someone without a deep knowledge in calibration and just the optimization step will need an engineering work.

General Goal

Developing a complete method to reach the best possible calibration by saving costs and time for the Rail Pressure Governor (RPG).

Specific Goals

- Developing the best criterion to optimize the controller parameters of the rail pressure control on the vehicle.
- Developing an optimum final strategy:
  - Deciding optimum number of operation points.
  - Deciding optimum number of measurements points.
  - Analyzing how many measurements are necessary to each point.
- Integrating this method to other vehicle to analyze if it’s usable.

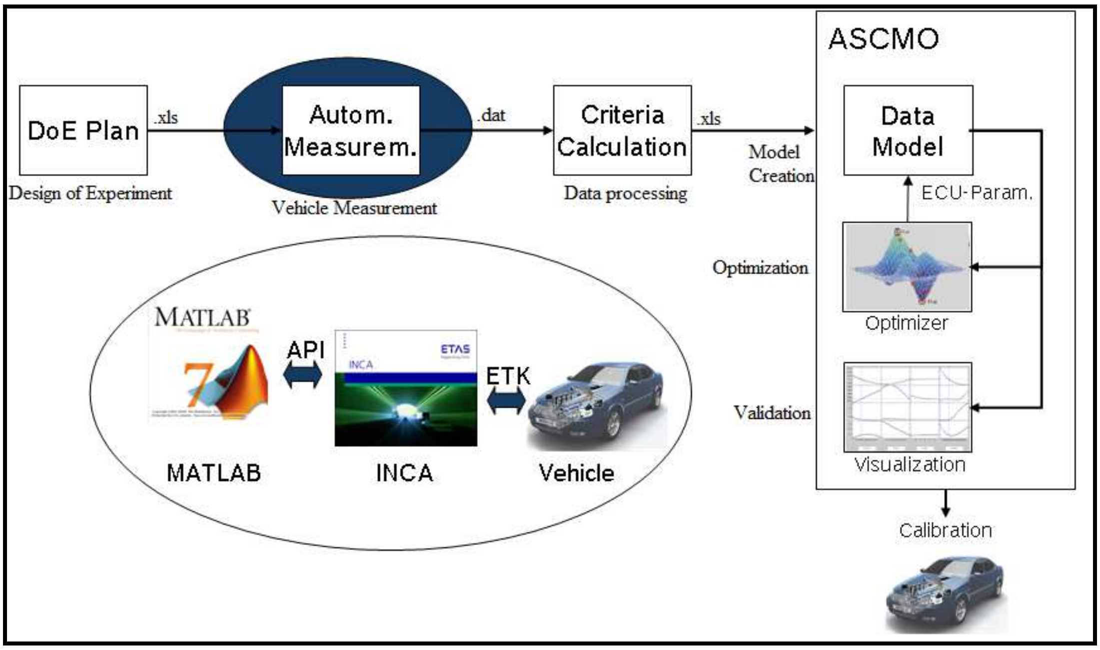


Figure 1.1-1 Diagram

The diagram shows all the methodology's steps. This work can be divided in four big steps: DoE Plan, Automated Measurement, Criteria Calculation and optimization using ASC. The development will start with the worst case, but using just one operation point. After analyzing the results, the method will be exanimate with more than one operation point and more than one project.

1.1 METHODOLOGY

The implementation of the proposed system will be done in following steps:

1st Step: Study of the problem:

The goal is to study about the benefits of this work for the customer, increasing the understanding of the importance and the necessity of this work. After that, a study about governor theory and the methods used at BOSCH Group to calibrate controllers needs to be done.

2nd Step: Feasibility' project study:

It's necessary to make an analysis about the reproducibility of the measurements. Reproducible measurements of the system behavior are a requirement to recognize calibration data influence.

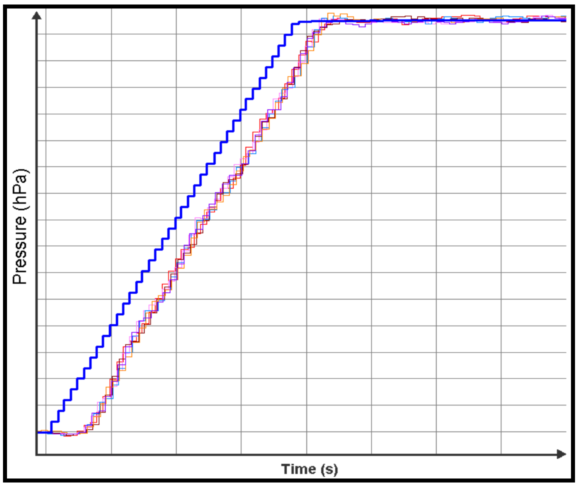


Figure 1.1-1 Reproducibility

3rd Step: Design of Experiment Plan:

Creation of an experiment plan according to statistical aspects. The goal of the DoE is to have a model-like description of unknown systems based on measuring data. It requires a test plan which is a file with parameters to be calibrated and all information necessary to do the initial measurements.

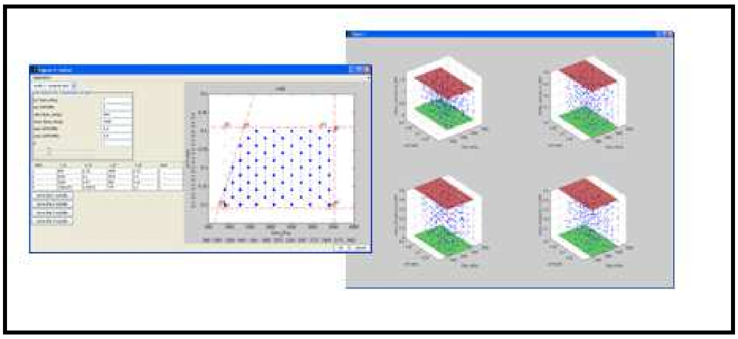


Figure 1.1-2 Design of Experiment

4th Step: Automated Measurement:

With the test plan ready, it is necessary to start measurements. All the measurements will be done on a proper track (Bosch - Boxberg and Schwieberdingen test tracks). A Citroen C4 will be the platform demonstrator: a DV6Mod Engine Version V3, 1.6l with CRI 2.5 1600 bar CPC System. To start this step it is necessary a complete knowledge about Bender tool (developed by Bosch to properly execute the measurement strategy).

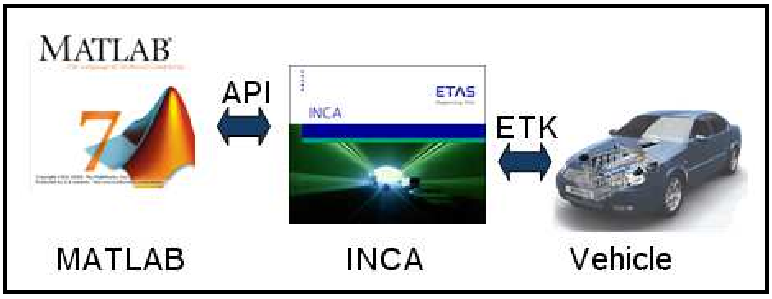


Figure 1.1-3 Automated Measurement

5th Step: Criteria Calculation:

This is a critical step since it is necessary find out which criterion could be the best to find a model that gives the best parameters to the calibration.

For that, it will be done a MatLab code to all criteria that could be useful. The code needs to be robust and precise because the signal shape can be very different from one vehicle to another and it is necessary cover up the maximum of cases. To define the criterion some models will be designed to describe the influence of criteria in the system, the result will be a Pareto-optimum solution. Then, with the Pareto model, a new round of measurements will be done to validate all models.

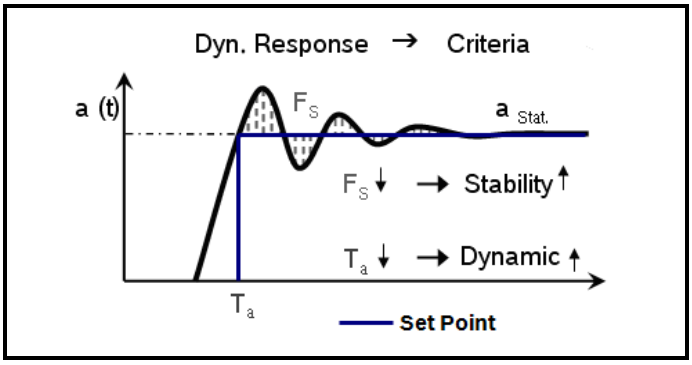


Figure 1.1-4 Criteria Calculation

6th Step: Optimization using ASC:

Defined the criterion calculation, the optimization using ASC step follows. For that, the tool so-called Advanced Simulation for Calibration, Modeling and Optimization (ASCMO) will be used. This step can be divided in three parts: data-model, optimization and visualization. With the results of the last step, the model will be generated. With this model, the optimization will be done using sliders, changing the ECU-parameters and looking to the criterion's result.

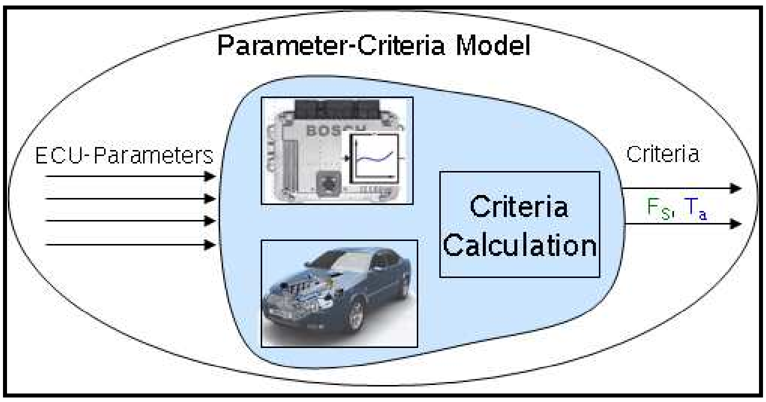


Figure 1.1-5 ASCMO

7th Step: Test Program:

A complete test program and a calibration in another vehicle have to be done after parameter calibration to validate results and conclude what is the benefit of this new method.

8th Step: Seminar presentation:

In this step the thesis will be concluded and a presentation will be done.

1.2 STRUCTURE OF DOCUMENT

This document is divided in four important subjects: Market Analysis, Theoretical basis, Development and Management.

Market Analysis will show what kind of clients this project intends to reach, who is already working in this area, and what kind of work was already done in this segment.

The chapter 3 is an introduction about the Theoretical basis, what is necessary to know to understand this work. First a brief explanation about how a diesel engine and the Common Rail System work. Second, what is an Engine Control Unit (ECU), and how the communication between the vehicle and software works (emulator test probe – ETK). Succeeding that, a description about PID controller (how it works, how can be calibrated, what is the effect of each controller and so on). After, the text clarifies about the Rail Pressure control. The next part will explain about all tools used to develop the project.

The chapter 4 presents the project's development. It shows how all the steps were developed since the creation of a test plan until the method's description. It describes what problems were founded in the way, and all tests and results.

The chapter 5 is about the management plan. An analysis of risk and costs will be described, and a business' plan is presented.

Finally, consideration discussion about this project and the achieved results, possible future work to be considered, next steps, a balance of results, the advantages that the new method brought to the calibration process, and the positive and negative points of this new approach.

2 MARKET ANALYSIS

Market analysis is the first step of a marketing planning. Before you start any business, it is important doing a study about the environment that your product or service will be consumed. Clients, concurrency and patent analysis are basic items that you should do in this step.

For a new company to be successful in the market, first of all it is necessary to analyze who are the potential clients, what they need or what necessity are you creating for them. The second point is 'who is the concurrent'. Identifying threats, opportunities, weaknesses and strength of competition is a key to success. And the last point is the patents' study. It is important to know if there are any patents that could compromise the right of usage of any technology or idea.

Following, all these analysis about market will be presented for the current project.

2.1 CLIENTS

All company in the automotive sector needs a team working in the calibration methods. It is not possible to sell a vehicle without a dataset that will enable a smooth operation of the vehicle.

Figure 2.1-1 Possible clients

Hence, every automotive company and its system suppliers are potential clients. Since the service here presented has as main goal the reduction of time and costs for the client, it is clearly an attractive business. The main point is finding ways to make the client comfortable and safe to invest in a very new service in the market. The advantages of the service here presented must be strong enough to make the potential client believe that investing in a new method (that can replace an existing and known one) is worthwhile. Figure 2.1-1 shows some of the potential clients.

2.2 COMPETITORS

Specialized companies in the calibration area, in general, are characterized as medium and large companies that often work exclusively for a particular automaker, but independently. Example of these companies are IAV, AVL and BEG. Following there is a brief description about these three companies.

The AVL[1] is a privately owned and independent for the development of powertrain with internal combustion engines, instrumentation and test systems. The areas where the company will generate competition in the service described here are the technologies of advanced simulation and instrumentation. In the calibration area the company offers software so-called AVL CAMEO, whose idea is very similar with the method described in this project. However, the AVL works with neural networks theory. The service here presented has the advantage of a new algorithm to complex system models, based on statistics process. AVL is a company present in the market since 1946, with a lot of experience with calibration, thus it will be a strong competitor.

Figure 2.2-1 Concurrent

With 4000 employees, IAV[2] is one of the leading engineering partners to the automotive industry whose main shareholder (50%) is Volkswagen group. Inside the company there is a sole department working on development of new processes and automation for engines and powertrain calibration. This company, as AVL, uses neural

---

1 AVL. Available in: <https://www.avl.com/c/document_library/get_file?uuid=52d82e99-285b-4077-ac07-3c3757621c81&groupId=10138> Accessed 13th September 2011.

2 IAV. Available in: <http://www.iav.com/en > Accessed 13th September 2011.

networks theory to model systems. Besides that, it has been working with calibration of diesel injection systems since 1989.

And finally, the Bosch Engineering GmbH[3] (BEG) is a modern engineering service provider and a wholly-owned subsidiary of Robert Bosch GmbH, present since 1999. Its services comprise specification, functional development, calibration and software integration including all tests and verifications.

2.3 CONSIDERATIONS

In this section was possible to verify that the service here presented can have a broad area of activity. Companies all around the world need partners to develop improvements in the calibration methods, which mean a lot of potential clients. Besides that, the project described here has an important difference in relation to the concurrency: instead of using neural networks, a new algorithm based on statistical theories too is the base of the modeling systems.

[3] BEG. Available in: < http://www.bosch-engineering.de/en/boschengineeringgmbh/overview/index.aspx> Accessed 15th September 2011.

3 THEORETICAL BASES

Following there is a brief explanation of all theoretical bases necessary to understand this project. How a diesel engine and a common rail systems work, what is an ECU, basics concepts of control engineering and PID controller, how the rail pressure is controlled by a PID and what tools were used for developing this project.

3.1 DIESEL ENGINE

A combustion engine is an engine where the heat and the combustion process are coupled. In the combustion, fuel's energy is converted to thermal energy and then to mechanical work. "This mechanical energy moves pistons up and down inside cylinders. The pistons are connected to a crankshaft, and the up-and-down motion of the pistons creates the rotary motion needed to turn the wheels of a car forward"[4].

In the same way as gasoline engine, a diesel engine can use a four-stroke combustion cycle: intake stroke, compression stroke, combustion stroke and exhaust stroke. Figure 3.1-1 shows how a four-stroke combustion cycle works.

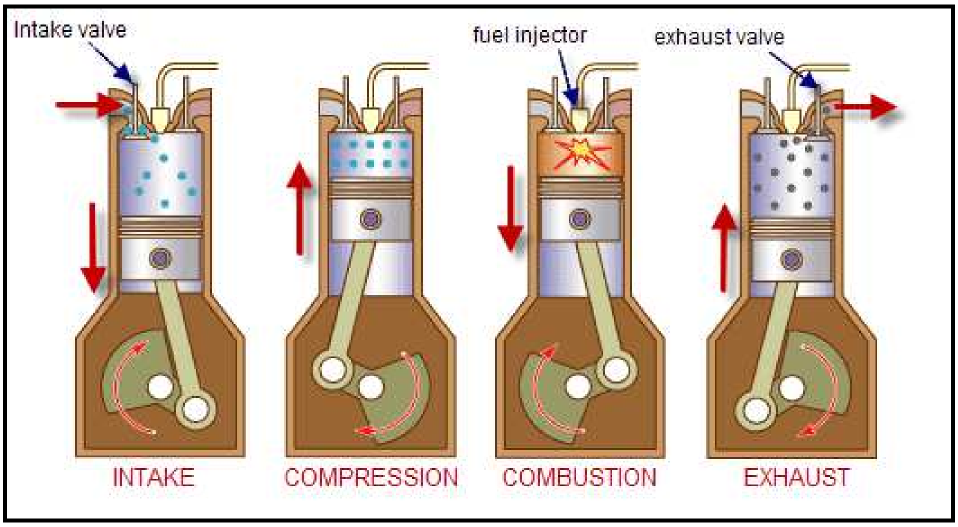


Figure 3.1-1 Four-stroke engine
Font: Adapted from Encyclopedia Britannica, 2007

First, the intake valve opens up, letting air and moving the piston down (INTAKE). Second, the piston moves back up and compresses the air (COMPRESSION). When the piston reaches the top, fuel is injected in a precise moment and ignited, dragging the piston back down (COMBUSTION). Lastly, the piston moves back to the top, eliminating the

[4] Font: How Stuff works < http://auto.howstuffworks.com/diesel1.htm >. 7th October 2011 accessed.

exhaust created from the combustion out of the exhaust valve (EXHAUST). Remember that it is the heat of the compressed air that ignited the fuel, not a spark plug as in a gasoline engine.

Diesel and gasoline engines have some important differences. One of these differences is about combustion process. In a gasoline engine, first fuel is mixed with air in an intake manifold, so it's compressed by pistons and ignited by sparks from sparks plugs. In a diesel engine, first the air is compressed, and thus the fuel is injected, no need of a spark plug. Nowadays, a new technology makes possible the gasoline engine have a direct injection too, that means it is not necessary an intake manifold, the fuel is injected directly in the cylinder.

However, the biggest difference is about controlling: gasoline has a quantity control; a diesel engine has a quality control.

In a gasoline engine, the ratio between fuel and air is always as constant as possible, but the quantity of fuel-air injected is changing, so the quality is almost constant, what is controlled is the quantity.

In a diesel engine, however, the quantity of air inside the cylinder is always the same (to each operation point), so quantity is almost constant, but the fuel quantity injected changes, which mean the quality of the mixture will be controlled.

Gasoline Engine	Diesel Engine
External fuel mixture formation	Internal fuel mixture formation
Spark ignition	Auto ignition
Quantity control	Quality control
Constant volume combustion	Seiliger process
Homogeneous mixture in combustion chamber	Inhomogeneous mixture in combustion chamber
Max. engine speed 7000 rpm	Max. engine speed 1500…5000 rpm
λ = 0,7…1,3	λ > 1 in normal operation
Compression ratio ε = 8…11	Compression ratio ε = 14…21
Compression pressure 10…16 bar	Compression pressure 25…55 bar
Compression temperature 350…450°C	Compression temperature 750…900°C
Max. combustion pressure 30…50 bar	Max. combustion pressure till 200 bar
Max. combustion temperature ~2500°C	Max. combustion temperature >2000°C
Exhaust gas temperature 600...800°C	Exhaust gas temperature 550...750°C
Specific fuel consumption 240…430g/KWh	Specific fuel consumption 160…400g/KWh
Effective degree of efficiency η_{eff} to 30%	Effective degree of efficiency η_{eff} up to 53% (stationary) or up to 43% (passenger car)

Figure 3.1-2 Comparison: Gasoline and Diesel engine[5]

Figure 3.1-2 summarize the mainly difference between gasoline and diesel engine.

Diesel engine's operation conditions depend on different process characteristic relations. Each operating point thus requires[5]:

- The correct fuel quantity,
- At the correct time for the correct duration,
- At the correct pressure, and
- Within the correct temporal course.

The injector on a diesel engine is one of the most complex components. It needs to resist the temperature and pressure inside the cylinder and deliver the fuel in a fine mist.

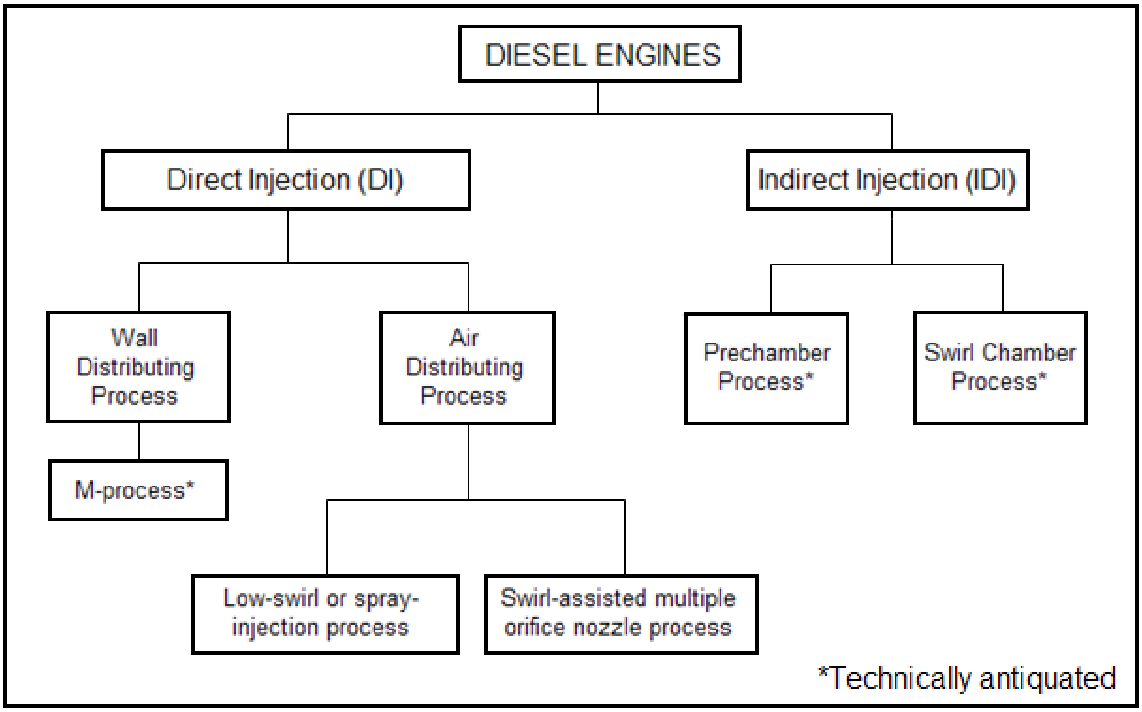


Figure 3.1-3 Combustion process
Font: BOSCH Group. Basic of diesel technology, Esslingen: Steinbeis Transferzentrum, 2008

The combustion process (Figure 3.1-3) can be divided in two: direct injection (DI) and indirect injection (IDI). DI principle summarizes the process for which there is no subdivision of the combustion chamber. Figure 3.1-4 shows a comparison of indirect and direct injection methods.

[5] Font: BOSCH Group. Basic of diesel technology, Esslingen: Steinbeis Transferzentrum, 2008

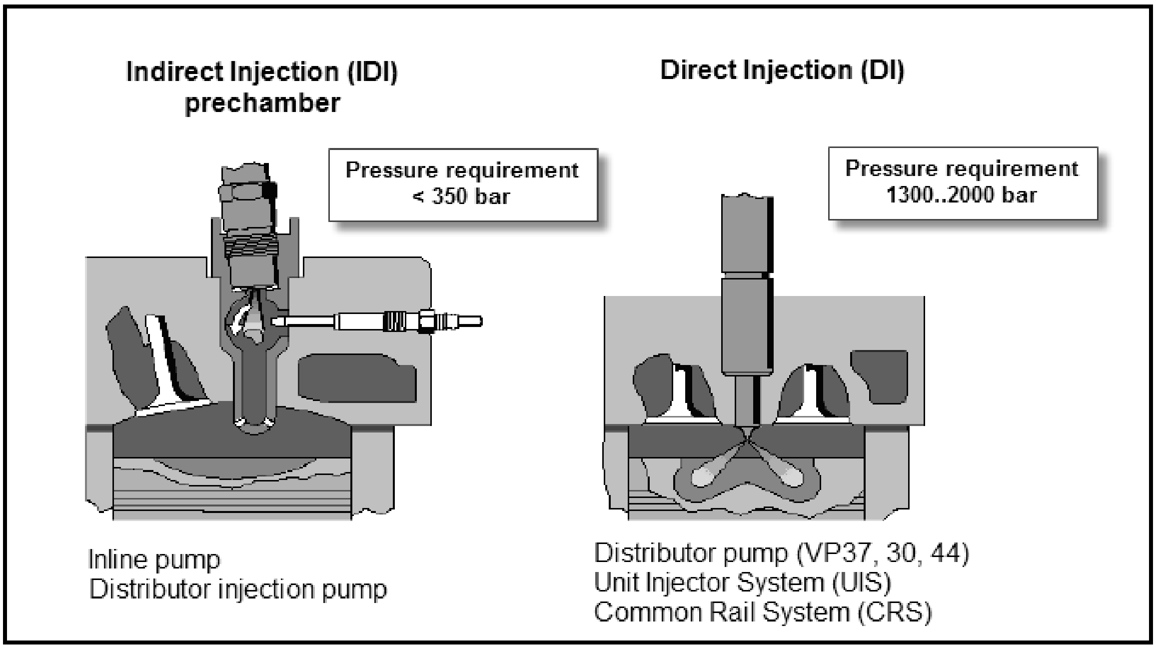


Figure 3.1-4 Direct and indirect injection principle
Font: BOSCH Group. Basic of diesel technology, Esslingen: Steinbeis Transferzentrum, 2008

The main difference between Direct and Indirect Injection is the layout of the injection system. The Indirect Injection System has a small swirl chamber above the cylinder, where the fuel is injected. The Direct Injection system has the injection nozzle fixed to the top of the combustion chamber; usually the piston has a crown shape in the top to create the needed swirl.

Unit Pump System (UPS), Unit Injection System (UIS) and Common Rail System (CRS) are examples of direct injections.

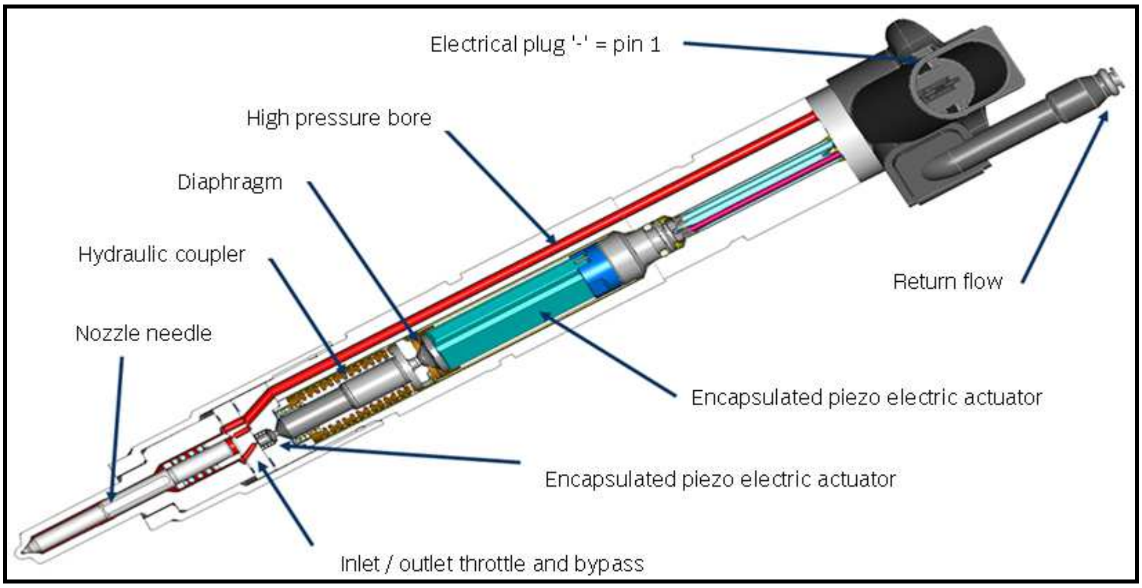


Figure 3.1-5 Common Rail Injection (CRS3)
Font: BOSCH Group. Injectors injection curves, Esslingen: Steinbeis Transferzentrum, 2008

UPS and UIS attain the highest injection pressure. These are time controlled diesel-injection systems, which consist of an injection pump and an injection nozzle with an integrated, fast-switching solenoid valve. The only difference between both is in their high-pressure component. Figure 3.1-5 shows one injector used in the CRS.

3.2 COMMON RAIL SYSTEM

A Common Rail is when the pressure generation and the injection are isolated from one another in the accumulator injection system. The injection pressure is created independent of the engine speed and the injection quantity. A high-pressure fuel accumulator provides the fuel for the injection, which means this system offers a lot of flexibility when performing the injection. Figure 3.2-1 shows a first generation common rail system mounted on a 4-cylinder engine. Figure 3.2-3 shows the common rail.

Figure 3.2-1 Common Rail System
Font: BOSCH Group. Introduction to the CR system, Esslingen: Steinbeis Transferzentrum, 2008

The high pressure is generated in two stages in a passenger car. First the low pressure pump (presupply pump) intake of the fuel from the vehicle tank and delivery to the high pressure pump. So the high pressure pump (Figure 3.2-2) compresses the supply fuel to a high pressure.

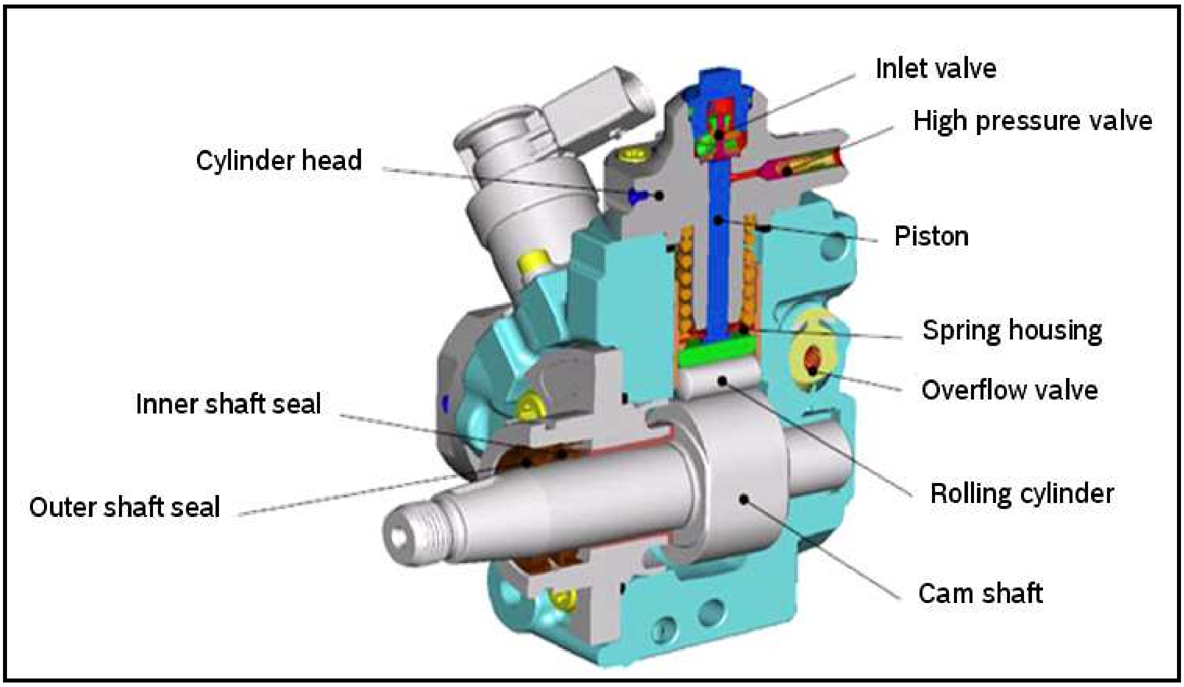


Figure 3.2-2 High pressure pump (CP4.2)
Font: BOSCH Group. Electronic control unit concept, Esslingen: Steinbeis Transferzentrum, 2008

The components of the low and high pressure system are low-pressure system (mechanical or electrical pre-supply pump, fuel filter with water separator and overflow valve, fuel metering unit) and high-pressure system (high pressure pump, high pressure accumulator/fuel distributor, pressure sensor, injectors, pressure control valve, pressure limiting valve and low limiter).

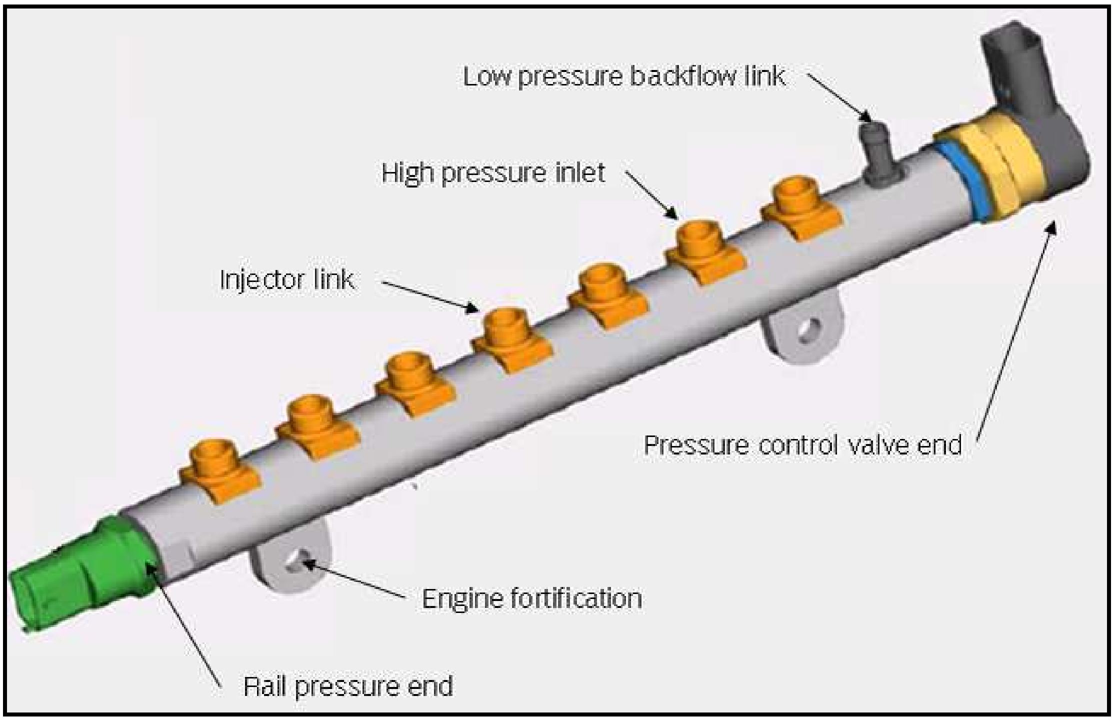


Figure 3.2-3 Common Rail
Font: BOSCH Group. Electronic control unit concept, Esslingen: Steinbeis Transferzentrum, 2008

Component	Description
High pressure pump (HPP)	Creates high pressure fuel for the injection
Common rail (CR)	Accumulates the fuel pressure
Metering unit (MeUn)	Controls the pump filling / rail pressure
Pressure control valve (PCV)	Controls the rail pressure
Rail pressure sensor (RPS)	Senses the rail pressure
Injector (CRI)	Injects the fuel into the cylinder

Figure 3.2-4 Rail system's main components

Figure 3.2-4 shows a brief description of the rail system's main components.

Figure 3.2-5 shows the hydraulic system overview of a possible rail system configuration, it's a dual actuator concept, that means the PCV and the MeUn are working together.

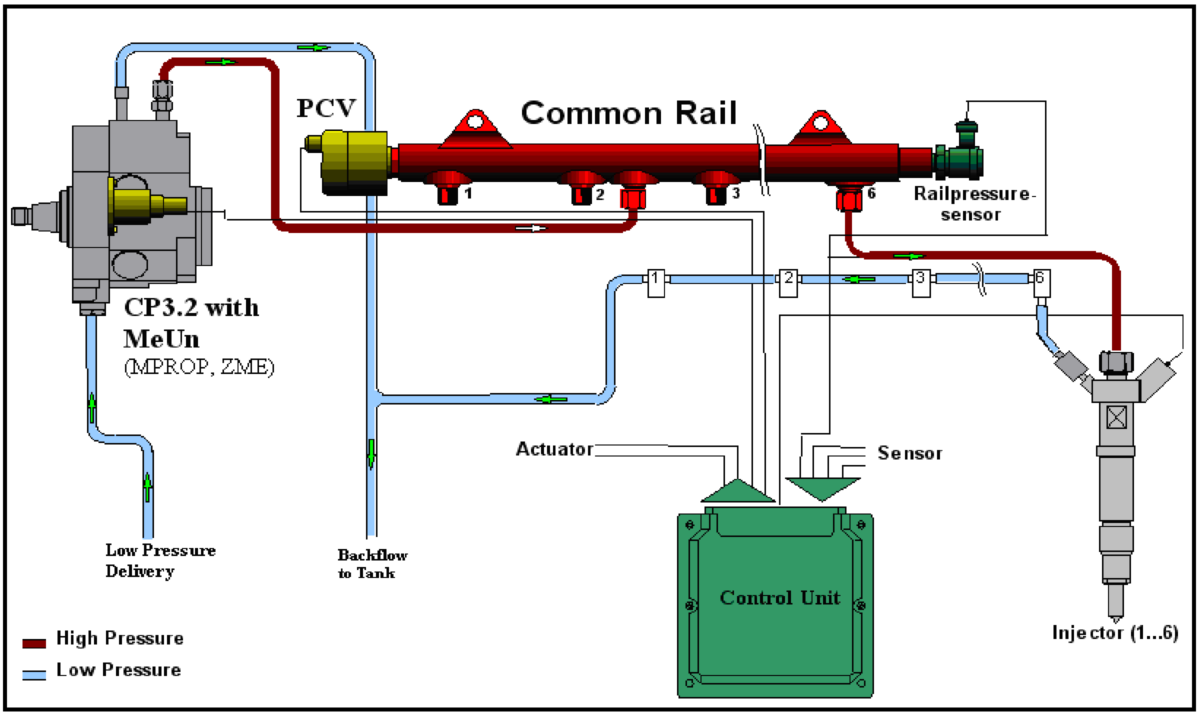


Figure 3.2-5 High and low pressure system
Font: BOSCH Group. Hydraulics, High Pressure System. Esslingen: Steinbeis Transferzentrum 2008.

The Common Rail System principle is a high-pressure generation and injection control independent of each other. The idea is accumulating fuel at any pressure level in the rail within the physical limits, so the desired injection quantity is always available. For each engine operation point it's possible select the high pressure, the injection quantity and the start of injection with large limits. This possibility is just in this kind of injection systems; the other systems are camshaft-driven systems and can only build up pressure if a cam initiates the pumping procedure.

Figure 3.2-6 shows the relationship between engine speed and pressure generation. It is possible observe that in CR-system the maximum pressure is available in a certain engine speed range.

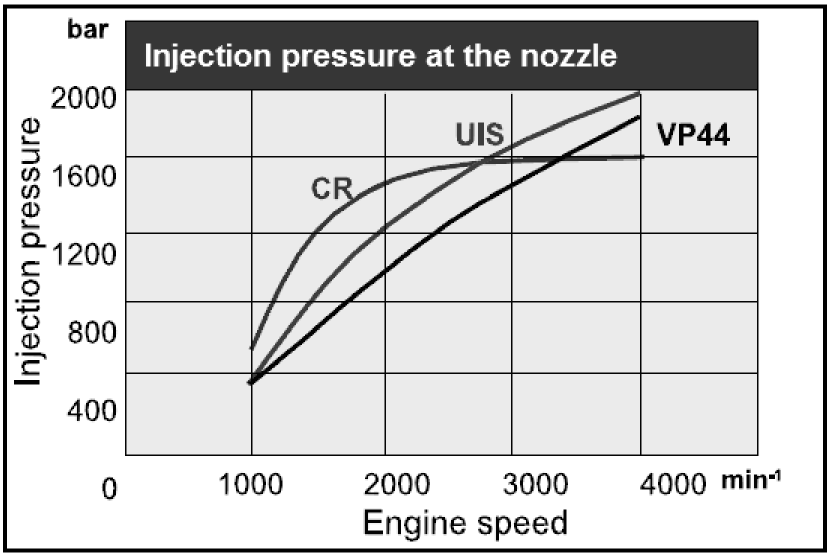


Figure 3.2-6 Comparison injection systems
Font: BOSCH Group. Basic of diesel technology, Esslingen: Steinbeis Transferzentrum, 2008

There are some different generations of the CRS, as Figure 3.2-7 shows. CRS Base (first generation) has a solenoid-valve and the high pressure pump compresses the largest possible fuel quantity each time the pump revolves. Using a Pressure Control valve (PCV) the excess quantity which depends on the operation point is discharged from the high pressure circuit and put again into the tank. This process results in loss of energy (the compressed fuel is dissipated as thermal energy). The maximum pressure that 1st generation can reach is 1400 bar.

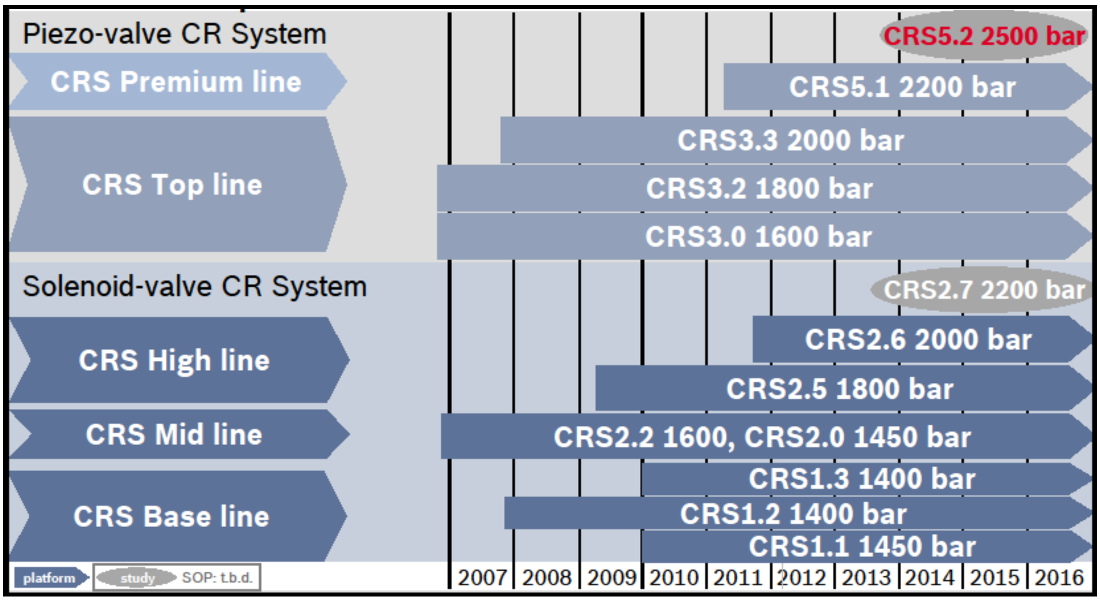


Figure 3.2-7 CRS' generations
Font: BOSCH Group. Development process at Bosch, 2009

The CRS Mid and High Line (second generation) have an optimized solenoid valve injector, multiple injections and a high pressure pump with quantity controlled in advantage of 1st generation. The maximum pressure is 1600 bar for Mid Line and 2000 bar for High line.

The CRS Top Line (third generation) uses a piezo-valve and it has improvement in the control functionality and high pressure pump. The new pumps are equipped with an intake-manifold-side quantity control via a metering unit. The power dissipation is reduced by avoiding the superfluous heating of the fuel. Figure 3.2-8 shows some differences between second and third generation.

Feature	CRI2	CRI3
Pressure	200 ...1600 bar	200 ... 2000 bar
Mass of moving parts (Needle)	15,5 g	4,3 g
Leakage	2,5 l/h	0 l/h
Needle velocity	v_{open} = 0,5 m/s v_{close} = 0,5 m/s	v_{open} = 0,8...1m/s v_{close} = 0,8...1m/s

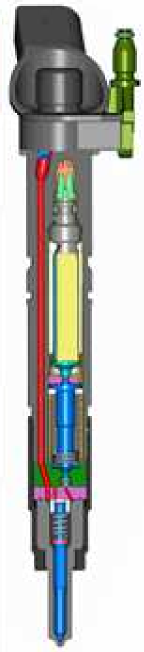


Figure 3.2-8 CRI2 x CRI3
Font: BOSCH Group. Development process at Bosch, 2009

> Piezo-valve: “In the 1st and 2nd generation the injection is controlled by a magnetic solenoid on the injectors. The hydraulic force used to open and close the injectors is transmitted to the jet needle by a piston rod. In the 3rd generation of Common Rail for passenger cars, the injector actuators consist of several hundred thin piezo crystal wafers. Piezo crystals have the special characteristic of expanding rapidly when an electric field is applied to them. In a piezo inline injector, the actuator is built into the injector body very close to the jet needle. The movement of the piezo packet is transmitted friction-free, using no mechanical parts, to the rapidly switching jet needles. The advantages over the earlier magnetic and current conventional piezo injectors are a more precise metering of the amount of fuel injected and an improved atomization of the fuel in the cylinders. The rapid speed at which the injectors can switch makes it possible to reduce the intervals between injections and split the quantity of fuel delivered into a large number of separate injections for each combustion stroke. Diesel engines become even quieter, more fuel efficient, cleaner and more powerful”. (SWEDESSPEED Available in: <http://www.swedespeed.com/news/publish/Features/printer_272.html> Accessed: 12th September 2011)

3.3 ECU

An engine control unit (ECU) is responsible for the control of the parameters engine (amount of fuel, starting of ignition, ignition timing and so on). The ECU monitors all inputs and outputs to have an effective control of the system managing fuel consumption and emission.

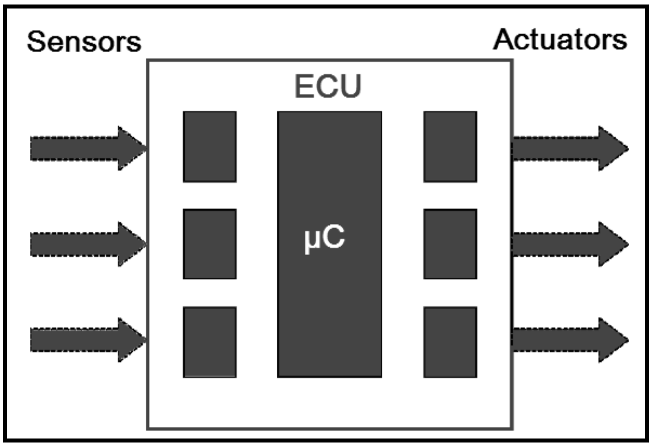


Figure 3.3-1 ECU

The processor needs memory to store the program and the data. To store the program and the permanent data is used a non-volatile memory (Flash). And to store measured values and calculation variables is used a Random Access Memory (RAM).

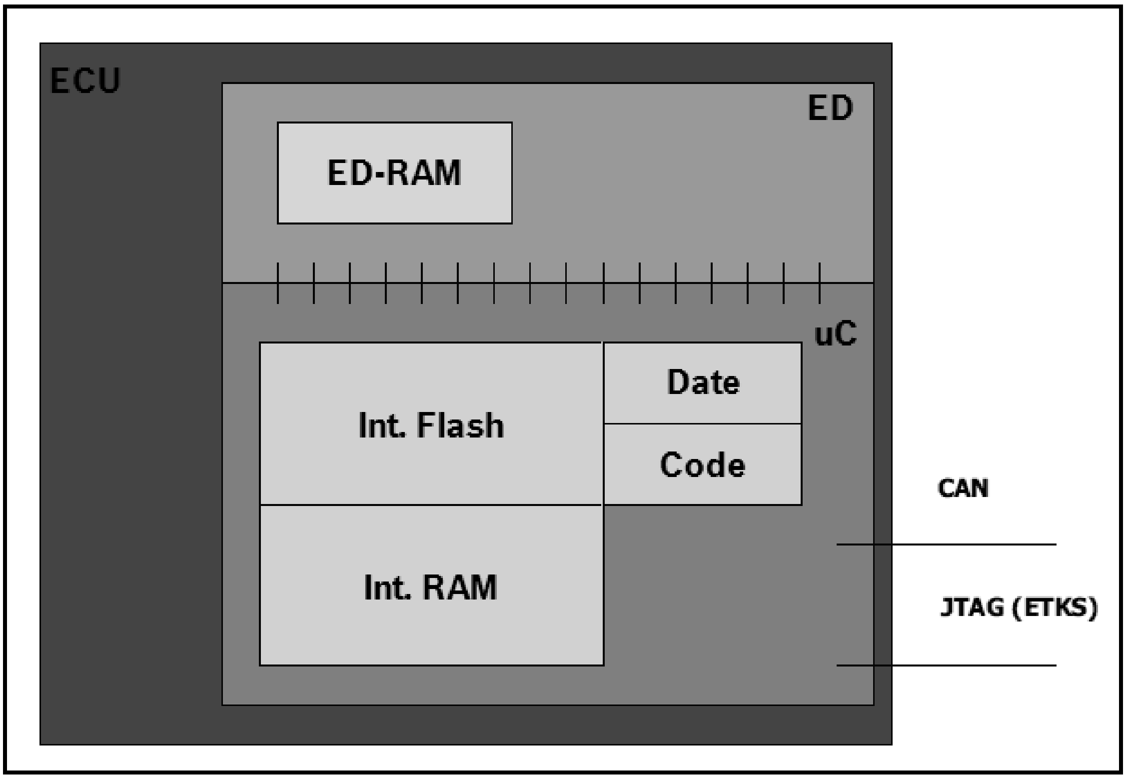


Figure 3.3-2 Control unit structure LEDA Light (EDC17)
Font: BOSCH Group. Electronic control unit concept, Esslingen: Steinbeis Transferzentrum, 2008

3.3.1 History of control unit development

In 1978 Bosch and AMI initialized the development of the control unit for diesel engines. The table below shows this development along the years.

Time	Manufacturer	Characteristic	Control Unit/Project
1978-81	Bosch/AMI Ben Hope	10 bits BUS 10 bits ADC 2 chip system	M1
1981-88	INTEL 8051	2 bits bus external 2 bits ADC Timer external	MSA1,M12,S1,M10 (John Deere)
1985-88	INTEL 8052	8051 with additional RAM and timer	MSA1 (series BMW)
1986-90	Siemens 80515	8 bits BUS, 8 bits ADC Timer and ADC integrated	M7(series Scania and Volvo) MSA6(series BMW and AUDI) LA3 (series MB)
1986-87	Motorola 6805	8 bits BUS ROM type	M102 MB add on-system (ARA)
1987	INTEL 8096/80C196	16 bits BUS 10 bit ADC High level language C	Commercial vehicle projects MS3, MS4,MS5,MSA11 passenger car (digital position controller)
1989	Siemens 80C517	8 bits BUS, CMOS, expanded periphery (basis 80515)	MSA8, MSA11, MSA12 (passengers car project with PI/HDK) CAN
1991	Siemens 80C167	16/32 bits BUS	MSA15 (passenger car control unit) EDC15 (passenger car control unit)
1993	Siemens 80C167	16/32 bits BUS	MS6.X (commercial vehicle control unit)
1996	Siemens 80C167	16 bits (expanded command set) internal dual port RAM internal ROM with 32 bits access	EDC15X (passenger car control unit)
2000	Motorola Power PC	32 bits RISC internal RAM, FLASH, CAN	EDC16 (passenger car control unit) EDC17 (commercial vehicle control unit)

Figure 3.3-3 Historical development
Font: Bosch Group. Electronic control unit concept, Esslingen: Steinbeis Transferzentrum, 2008 adapted

The system complex development is influenced not just by the growth in functions, as observed in the Figure 3.3-4. Due to the possibility of programming in higher-level languages, and the related structuring of software, more space is required for programs and for data. Besides that, the use of closed-loop instead of open-loop controls demands more storage space.

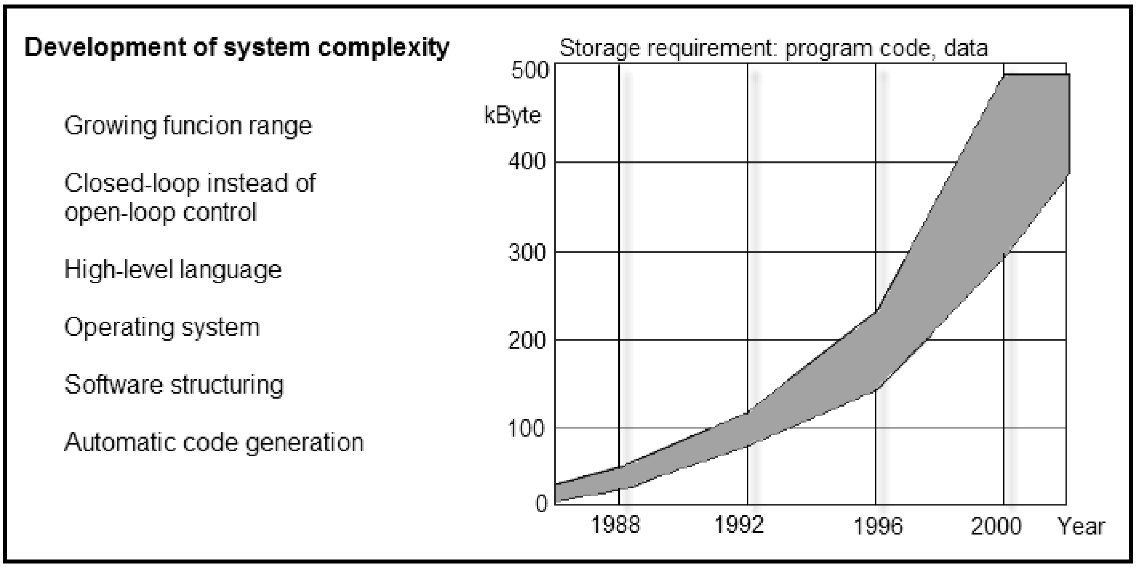


Figure 3.3-4 ECU storage requirement along the years
Font: Bosch Group. Electronic control unit concept, Esslingen: Steinbeis Transferzentrum, 2008

The control unit designation was derived from its main tasks. In the first control units the only task was to control the mass (M) of the diesel fuel to be injected, which is the reason they were called M1. Later, the start of injection (S) and the exhaust-gas recirculation (EGR) were also controlled using the control unit. After the release 15 of the control units, the designations were renamed to Electronic Diesel Control (EDC) in order to reflect the increasing importance of electronics in the field of passenger cars. With the new control unit platform, this designation was adopted for commercial vehicles as well. Figure 3.3-5 clarifies the control unit designation.

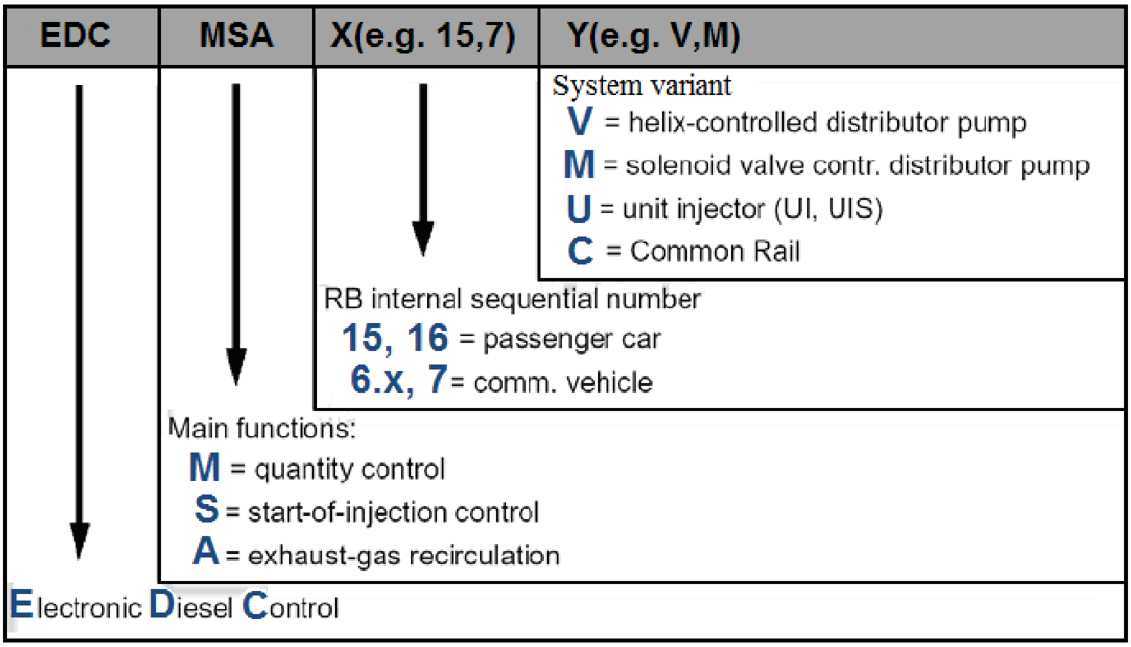


Figure 3.3-5 Control Unit Designation
Font: Bosch Group. Electronic control unit concept, Esslingen: Steinbeis Transferzentrum, 2008 adapted

3.3.2 Sensors and Actuators

Figure 3.3-6 shows a schematic overview of a Common Rail system with the three areas sensors, control unit and actuators.

An accelerator pedal sensor (AccPed) is used to monitor the accelerator pedal angle that corresponds to the torque demand. The AccPed is moved by the accelerator pedal using wiring (formerly it used a rod or a Bowden cable).

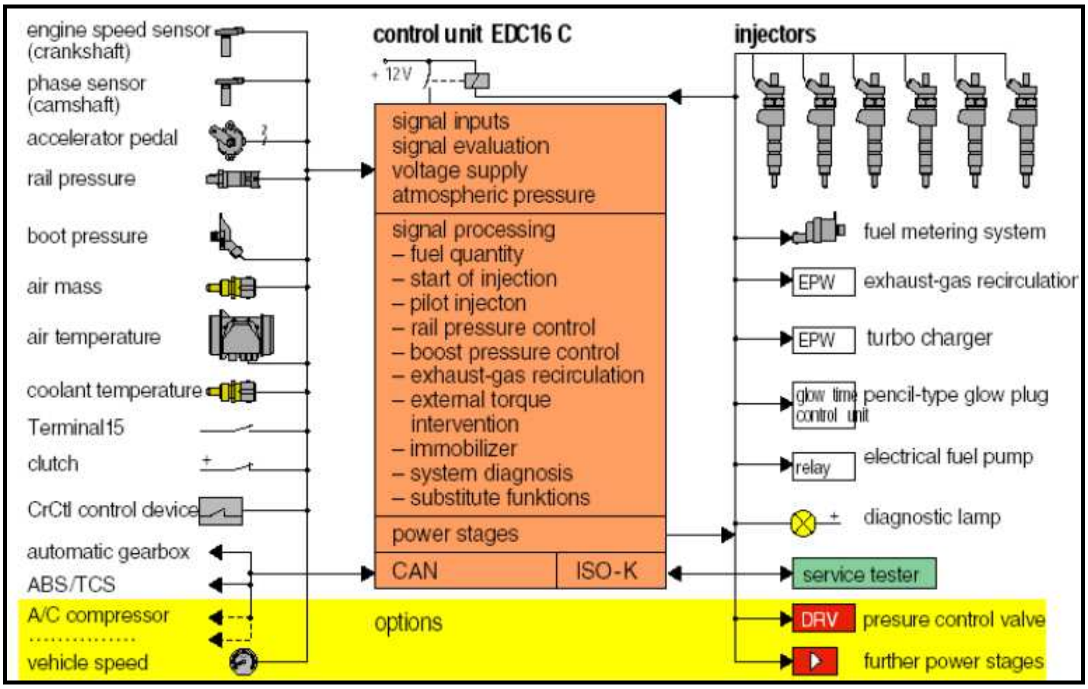


Figure 3.3-6 CRS overview

Font: BOSCH Group. Introduction to the CR system, Esslingen: Steinbeis Transferzentrum, 2008

The Temperature sensor is used for a lot of purposes and uses a simple temperature-dependent measuring resistor made of semiconducting material. Figure 3.3-7 shows a graphic with a relation between resistance and temperature, and the temperature range of some sensors.

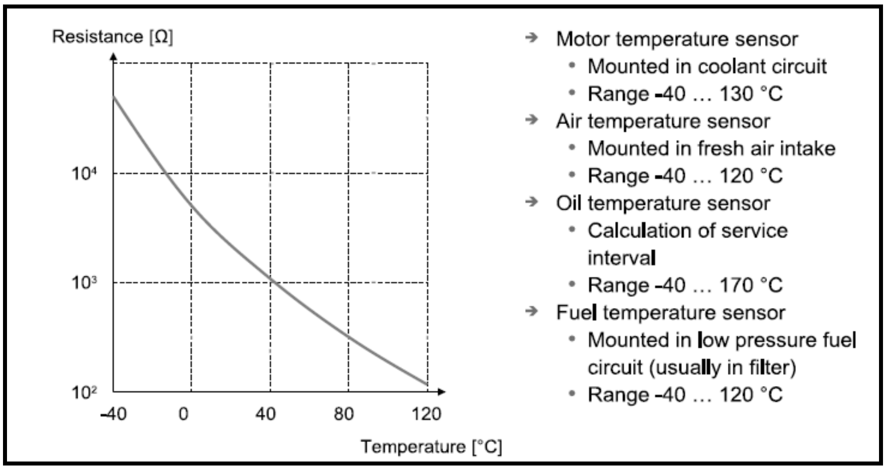


Figure 3.3-7 Resistance curve (NTC)

Font: Bosch Group. Electronic control unit concept, Esslingen: Steinbeis Transferzentrum, 2008 adapted

The engine speed sensor is used to measure the engine speed and to determine the crankshaft position or the position of the pistons.

The camshaft position indicates whether a piston moving towards top dead center (TDC) is currently in a compression stroke or an exhaust cycle. The phase sensor provides this information to the ECU. To implement this sensor the Hall Effect is used.

The Rail pressure sensor shown in Figure 3.3-8 measures the fuel pressure in the rail. The fuel pressure is controlled in the closed loop controller and deviations between setpoint and measured value are minimized using a pressure control valve (PCV) and a Metering Unit (MeUn).

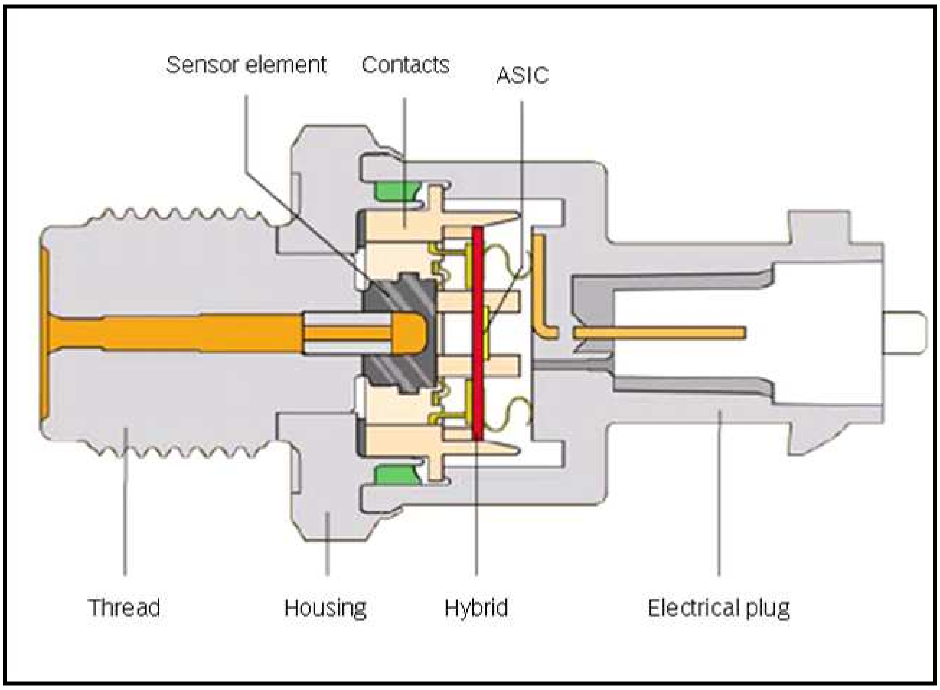


Figure 3.3-8 Rail Pressure Sensor
Font: Bosch Group. Electronic control unit concept, Esslingen: Steinbeis Transferzentrum, 2008

Besides these sensors, there are the boost pressure, hot film air mass and lambda sensor. Hot film air mass measures a partial flow of the current air mass flow through the air filter or through the measuring tube. The lambda sensor measures the fuel mixture formation (λ).

3.3.3 Interfaces and transmission protocols

The interfaces provided in the control unit are:

- CAN
- JTAG (calibration control unit for ETKS)
- K-line (not in all Projects)
- Internal control unit address/data bus

The Controller Area Network (CAN), which was developed for implementation in vehicles, is a serial communication interface. It is used in the vehicle for the networking of electronic control units.

This approach enables a data exchange between several control units over the network, and thus coordinates their functions more effectively. Besides that, a network with serial bus systems requires fewer lines and plug contacts to the control unit than the conventional cable harness.

The Joint Test Action Group (JTAG) interface is required for communication via the serial emulator test probe (ETK). ETKS is only available with an emulator device because there is no emulator RAM available for emulation.

The third interface is so-called K-line. It serves to exchange data between the control unit and the diagnostic test unit for measurement and testing purposes, but can also be used for calibration. The K-line interface is standardized according to ISO 9141.

Interface	Protocol	Functionality	Comments
CAN	Serial CCP XCP KWP 2000	Diagnosis Tester Adjust Measure	- Limited performance, dependent on free transfer bandwidth and message framework - Standard
ETKS (JTAG)	Serial	Adjust Measure	- High Performance Interface - EDRAM must be available
K-line	Serial KWP 2000	Diagnosis Tester Adjust Measure	- Not available in all EDC17 ECUs - Limited performance (susceptibility to interfaces depends on transfer rate)
Address/data bus	Parallel access	Adjust Measure	- High performance interface - Modified ECU casing - Tool Specific

Figure 3.3-9 Overview of hardware interface

Table above shows a summary of the various interfaces, their protocols, and their functionality. The suitable interface can be selected according to the calibration process and the control unit resources available.

The CAN Calibration Protocol (CCP) is used for serial information transmission via the CAN bus between the calibration tool and the control unit. It is part of the standardization of application systems (ASAP1) standard.

XCP is a further development of CCP. It is used for the CAN interface. The functionality here is the adjustment and the measurement.

3.3.4 Calibration Process

In order to meet the various demands, two different calibration approaches have been developed: serial and parallel calibration.

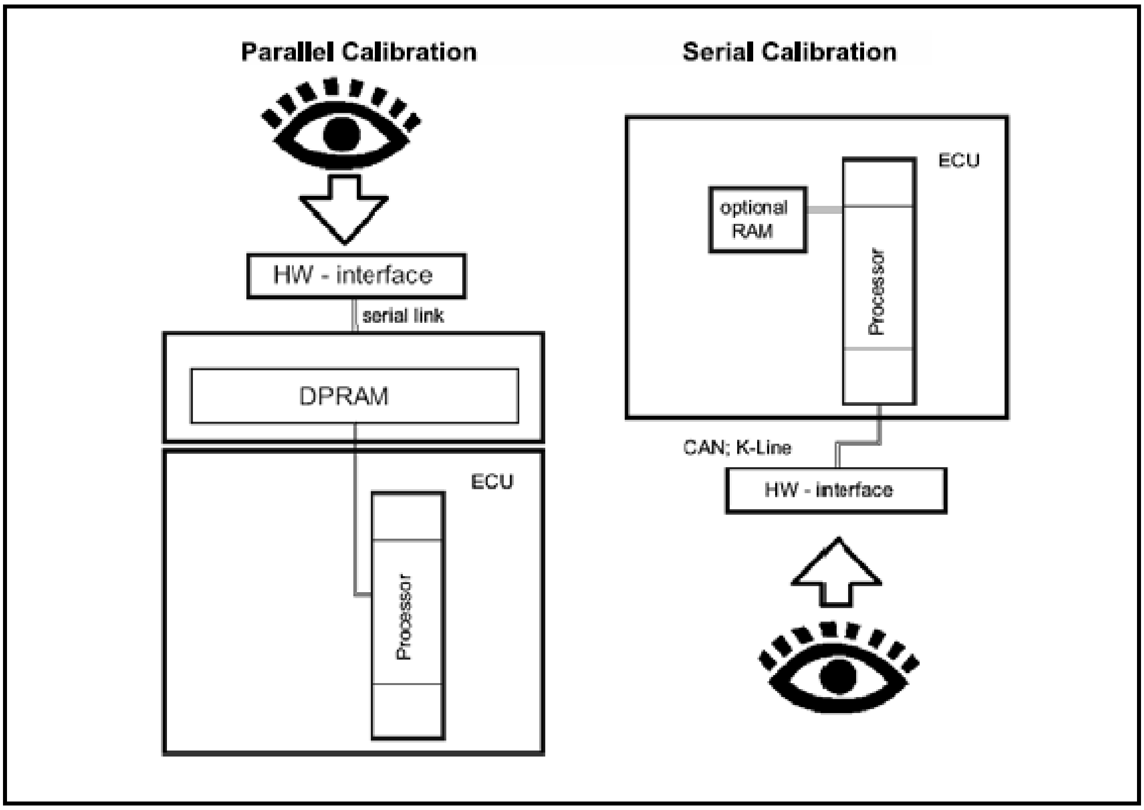


Figure 3.3-10 Comparison serial/parallel calibration
Font: Bosch Group. Electronic control unit concept, Esslingen: Steinbeis Transferzentrum, 2008 adapted

3.3.4.1 Serial Calibration

In a serial calibration the data is manipulated and the measured data is acquired by communication with the processor. In this case, the data to be calibrated are modified using one of the serial interfaces (CAN, JTAG or K-line). There are three possibilities for a serial calibration:

1. Serial Application with Additional Memory (SERAM) - full data emulation.

2. Dynamic serial application with additional programming (D-SERAP) - partial data emulation.

3. SERAP - adjusting by flashing only

If the size of calibration data exceeds the size of RAM, then the changed data on the working page will be saved partly to the RAM. This concept is also called partly emulation. ECU includes code and data, which represent the reference page. The working page is either the available RAM of the ECU or an additional memory (e.g. Emulation Device EDRAM). By calibration with the concept of partly data emulation, the working page must be written to the ECU-Flash very often. This makes some space in the limited RAM available for the calibration for further changes. This concept shows that not all of the calibrated data are available at the same time.

With SERAM calibration an additional memory (RAM) can be used in the control unit. When the hardware is initialized, the data from the Flash is copied into the RAM, whereas FLASH represents the reference page and RAM represents working page. In this method is possible switch over between working page and reference page during the calibration process. Copying data from the working page in the RAM to the reference page in the Flash is only possible using the Flash functionality.

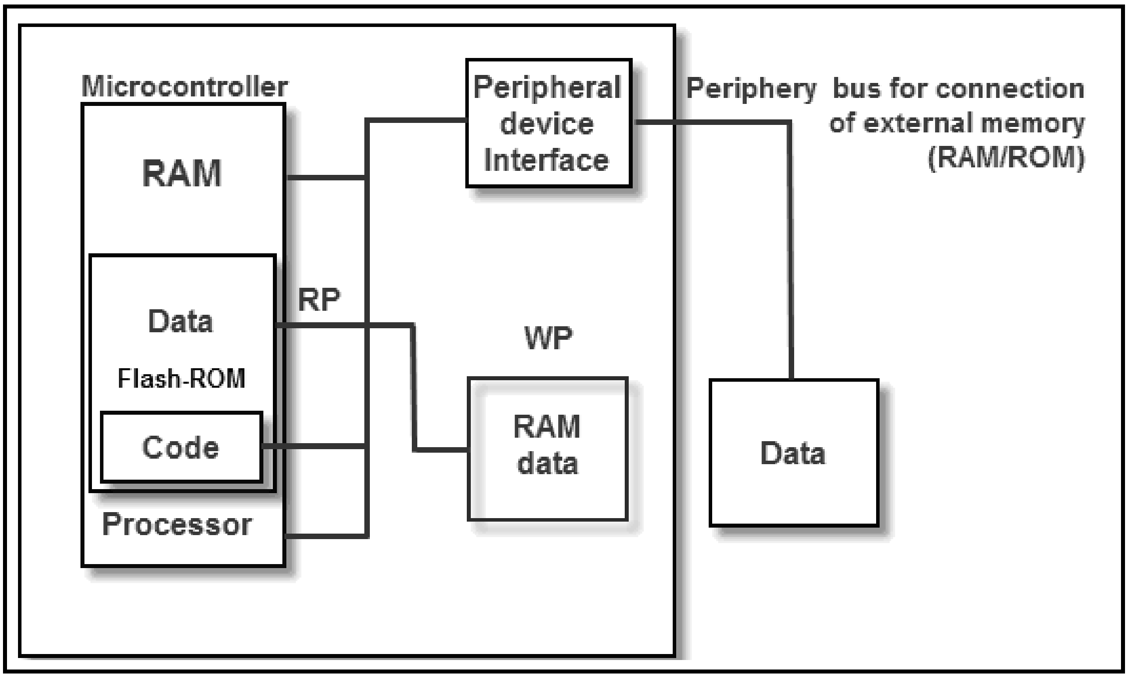


Figure 3.3-11 SERAM
Font: Bosch Group. Electronic control unit concept, Esslingen: Steinbeis Transferzentrum, 2008 adapted

With D-SERAP, an area of free memory in the control unit RAM is used, which is not allocated to variables. This is used as an emulation memory. The free memory area is reserved for the calibration. The data is combined to groups. One pointer is used to point at each of these groups. The pointers, in turn, are stored in a table. During calibration, the free memory area stores a second pointer table and the variables to be adjusted (fixed values, curves and maps). This area forms an image of the working page - the reference page still is in the Flash.

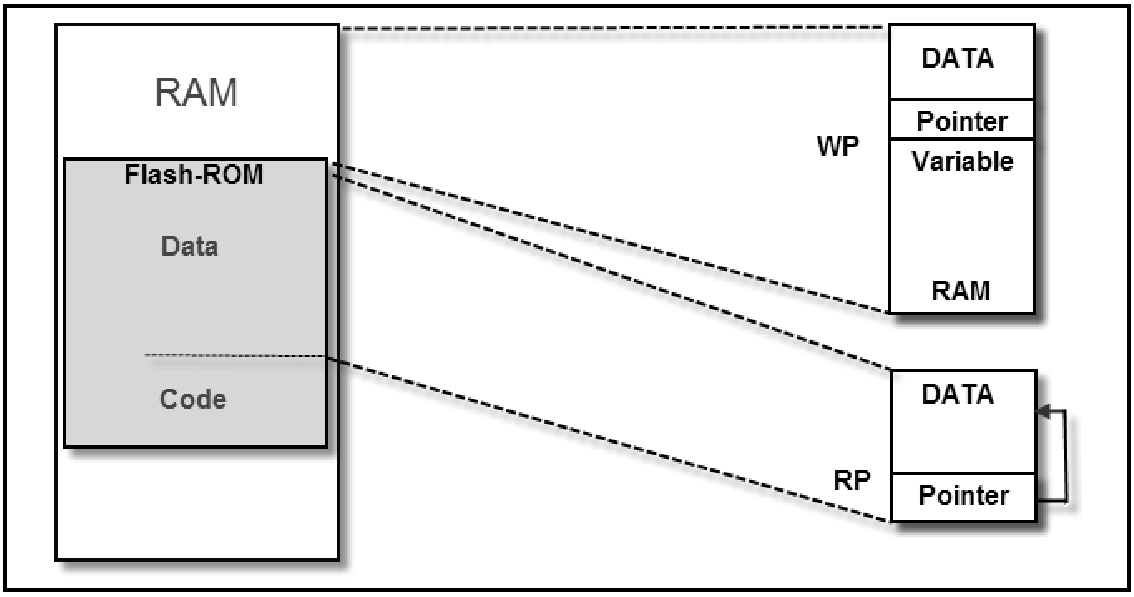


Figure 3.3-12 D-SERAP
Font: Bosch Group. Electronic control unit concept, Esslingen: Steinbeis Transferzentrum, 2008 adapted

During the start of the calibration session (HW-Init), the pointer table is copied from the Flash into RAM. The pointer table in the RAM therefore is a copy of the pointer table in the Flash and thus it points to the same calibration values at the start of the calibration. When a switchover between the working page and the reference page occurs during calibration, only the pointer table is exchanged. The calibration system can only access the page which is active in the control unit.

If a calibration value is written over by calibrating, the control first checks whether this data set is already present in the RAM. If this is not the case, the group is copied into the RAM and the pointer in the RAM (working page) is modified in such a way that it points to the copy of the group.

The EDC17 uses the overlay technique, for which there is 8 KBytes (LEDA-Light) or 16KBytes (METIS) of space available for calibration. These are 16 or 32 x 512- byte blocks. This is not sufficient for all parameters.

As a result, only a limited number of the parameters can be simultaneously adjusted on the working page in the control unit.

As soon as the first parameter has been adjusted, the first block is reserved. After this, all parameters in the superimposed 512-KByte area can be adjusted. As soon as a parameter is adjusted, which is not in this superimposed area, the next 512- Byte block is used. This continues until all 16 blocks have been used. Afterwards, only those parameters which lie within the 16 blocks can be adjusted.

As soon as another parameter is adjusted, an error occurs and the adjustment access on the working page is blocked.

To re-establish adjustment access, a checksum calculation or a new initialization of the connection must be started. If further adjustments are desired, the control unit must be flashed with the working page data so that more space is made available.

3.3.4.2 Parallel Calibration

The calibration hardware (parallel ETKT) is connected to the address/data bus of the control unit. Since the data of a series control unit is stored in the Flash and is thus not freely configurable, a so-called emulator test probe (ETK) is utilized which takes over the function of the control unit ROM. It emulates the data area of the ROM and allows for modification of the data set. As the data exchange and measurement data exchange occur in parallel via the bus, this process is called parallel calibration. The control unit doesn't read the calibration data from the EPROM, but from the ETK.

First of all, to use an ETK is necessary an EDC17 with an ECU with sockets processor. An address-data-bus located in the control unit is required for the ETK. A memory emulator, the processor and an interface to the calibration interface are mounted on the ETK which represents the interface to the calibration system. The ETK (adapter) is mounted to processor socket and the memory area of the Flash containing the calibration data is replaced by RAM. This RAM is located on the ETK and is written over from the Flash of the ETK or from the calibration tool. Since it is a Dual-Port-RAM, independent access by the processor and the calibration tool is possible.

The ETK emulates the control unit data from the Flash using a RAM. Since a RAM memory can be written and read in the ETK, the data can be modified during the execution of the control unit program. Due to the special design of the RAM, read accesses by the control unit and read/write accesses by the calibration system are possible at the same time.

The ETK contains three memory ranges: working page (WP), reference page (RP) and Flash data. In the moment that the calibration starts, a proper data set is used. This data set is loaded into the ETK Flash and the RP. During the calibration, data is changed, however these modifications are only carried out on the working page, which means that there is always a correct data set available on the reference page. If something goes wrong during the calibration that leads to a critical state, it is possible, with the calibration tool, to switch over from WP to RP. Besides that, handling two pages also allows to perform comparative measurements.

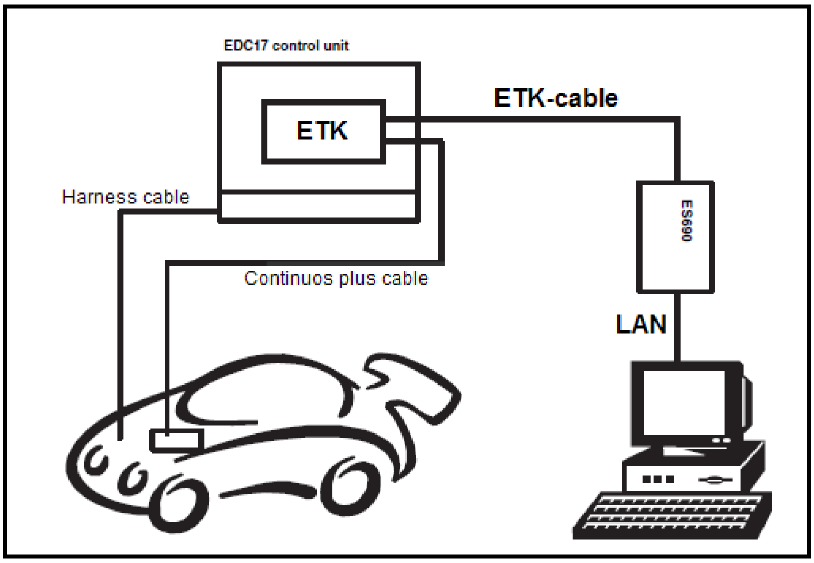


Figure 3.3-13 Linking the ETK-ECU in the calibration system (parallel)
Font: Bosch Group. Electronic control unit concept, Esslingen: Steinbeis Transferzentrum, 2008 adapted

Before starting to use the ETK, the control unit checks the compatibility:

1st: ETK is present or not,

2nd: The identification string (ID) in data area (ETK) is compared with the ID string in the code area.

3rd: If they match, the data from the ETK is used, otherwise the data from the ECU Flash.

4th: The address in the emulation area is written over to provide the calibration system with information on which data is active.

The advantages of a parallel calibration with ETK are: comfort, performance and data integrity at ignition OFF. The disadvantage is that an ECU conversion is necessary.

3.4 BASIC CONCEPTS OF CONTROL ENGINEERING

All kind of systems are exposed to changes. For a technical system, such as an engine which is supposed to keep a specific engine speed on a constant level, disturbances (like load changes or a change in the internal friction) can influence the engine speed. Without appropriate counter-measures the engine speed cannot be kept constant.

In the control engineering the engine speed is called manipulated variable, the load changes as external disturbance variable, friction as an internal disturbance variable, and the fuel quantity as correcting variable.

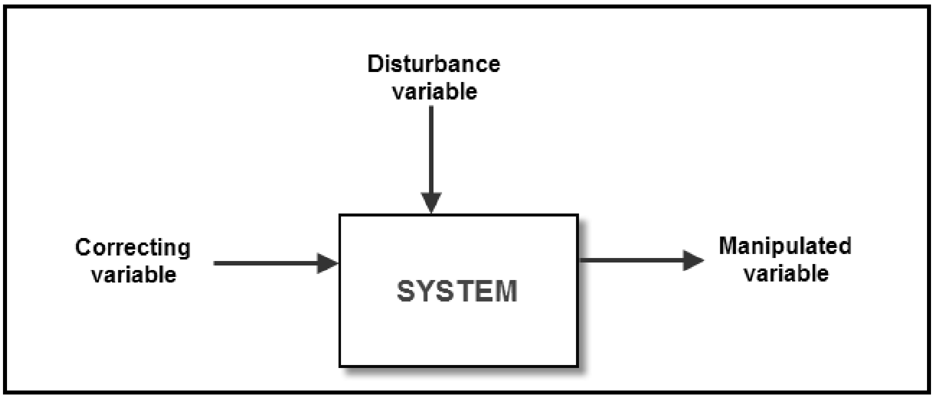


Figure 3.4-1 System (basic concepts)

There are two approaches to control a system: closed-loop and open-loop.

An open-loop control means a process that the manipulated variable is set through a predefinition of a correcting variable. The correcting variable is determined by environmental parameters which are not dependent on the manipulated variable, which means, no feedback takes place.

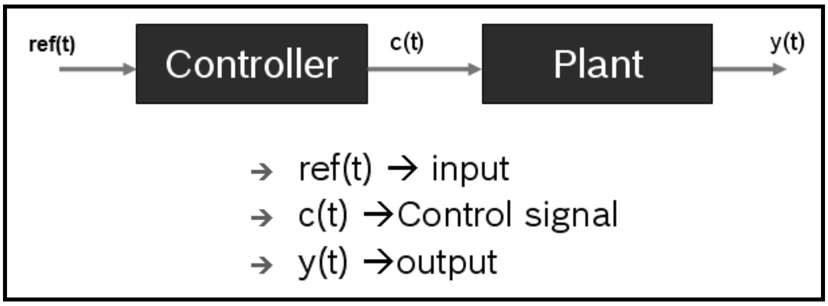


Figure 3.4-2 Open-loop control

A closed-loop control means a process which the manipulated variable is measured continually and compared to another variable called reference (setpoint value). The deviation between both values is used to influence the manipulated variable to adjust to the reference variable.

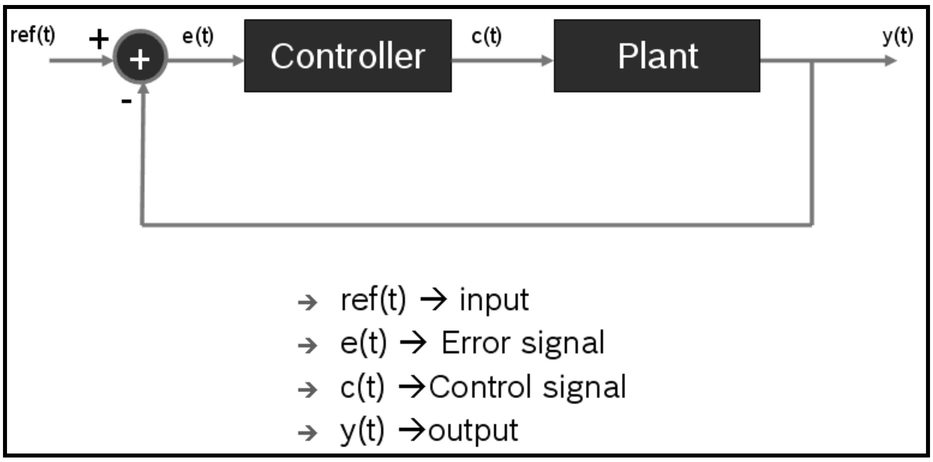


Figure 3.4-3 Closed-loop control

Figure 3.4-4 shows advantages and disadvantages of both approaches.

System	Advantages	Disadvantages
Open-loop control	Fast	Only controlled variables changes are acquired
Closed-loop control	All deviations from the setpoint value are acquired	Slow
General	Stability, dynamic and accuracy can be improved	Costs due to application effort and sensor system Risk of destabilization of stable systems

Figure 3.4-4 Comparison table

Three keywords for the evaluation of the system performance can be defined: stability, dynamic and accuracy. Figure 3.4-5, Figure 3.4-6 and Figure 3.4-7 show qualitatively these three characteristics.

The dynamics describes how fast the system reacts to a change of the setpoint or the occurrence of a disturbance.

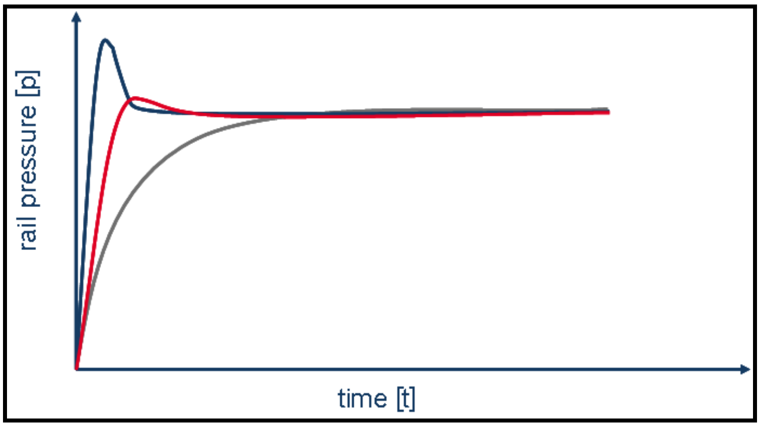


Figure 3.4-5 Dynamics

The stability specifies whether the system reaches a steady state, executes a permanent oscillation or takes on an unstable state.

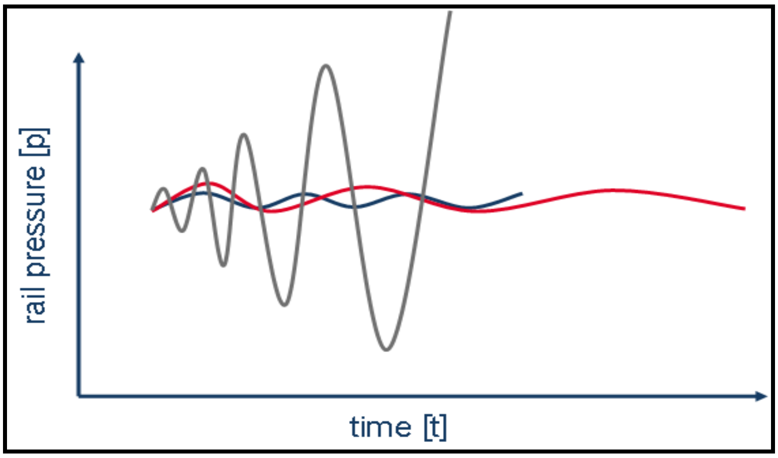


Figure 3.4-6 Stability

The accuracy specifies how accurate the set point of the controlled variable is reached in the steady state.

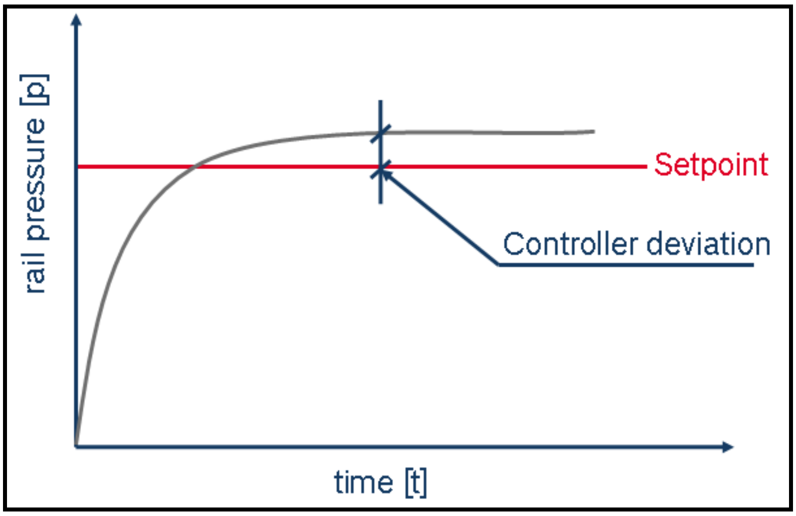


Figure 3.4-7 Accuracy

Since open-loop and closed-loop controls have their advantages, as showed on Figure 3.4-4, both methods are used in technical systems and thus also in EDC.

As an example of systems using open-loop, there is the Boost-pressure control; the speed governor and the rail pressure governor uses a closed-loop control.

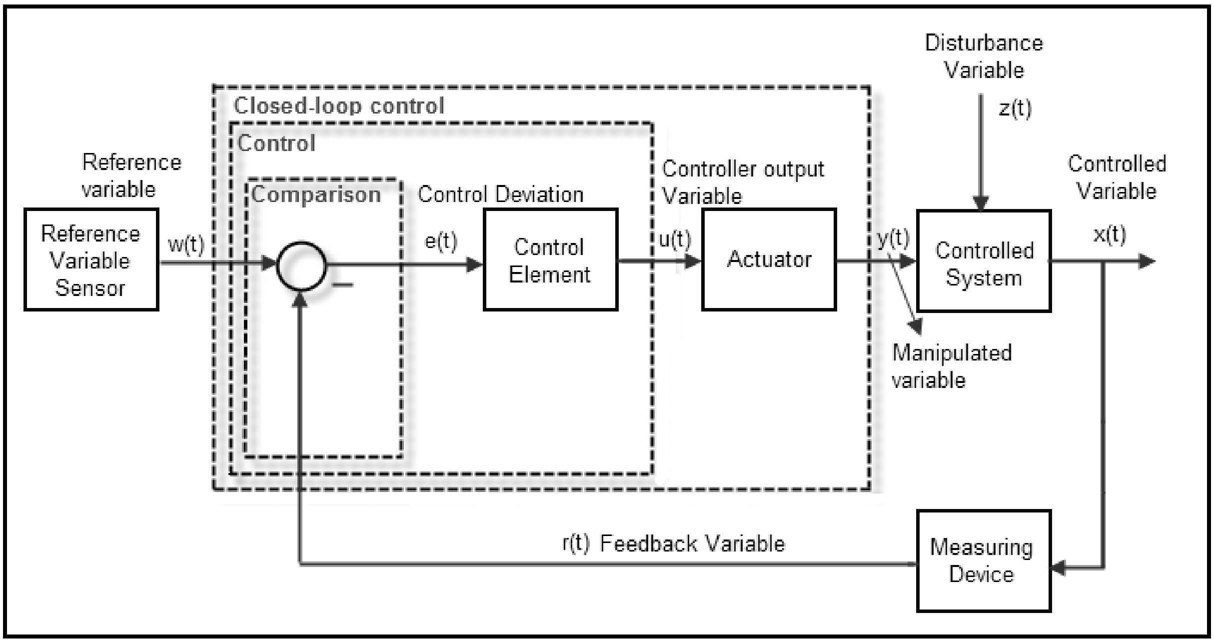


Figure 3.4-8 Control engineering's terms
Font: Bosch Group. Governor technique, basics and applications, Esslingen: Steinbeis Transferzentrum, 2008 adapted

Figure 3.4-8 shows a closed-loop system disassembled into standard blocks. The signal w is the reference variable and in the EDC Rail governor can be found with the label Rail_pSetPoint. The feedback variable is called RailP_pFlt; and the control deviation is the Rail_pDvt.

In control engineering, two different types of control can be identified: set-value and follow-up control. On set-value (also known as disturbance value) the controlled variable has to be kept constant. On follow up control (also known as set point response) the reference variable (set-point value) changes.

Sometimes is necessary to compare systems response quantitatively. For that, is defined some evaluation criteria (Figure 3.4-9). These criteria can be divided between dynamic and static behavior. The dynamics of the transient response is given by the rise time and is a measure for the speed. The static behavior describes the steady state of the system and is defined via the steady-state control deviation (accuracy).

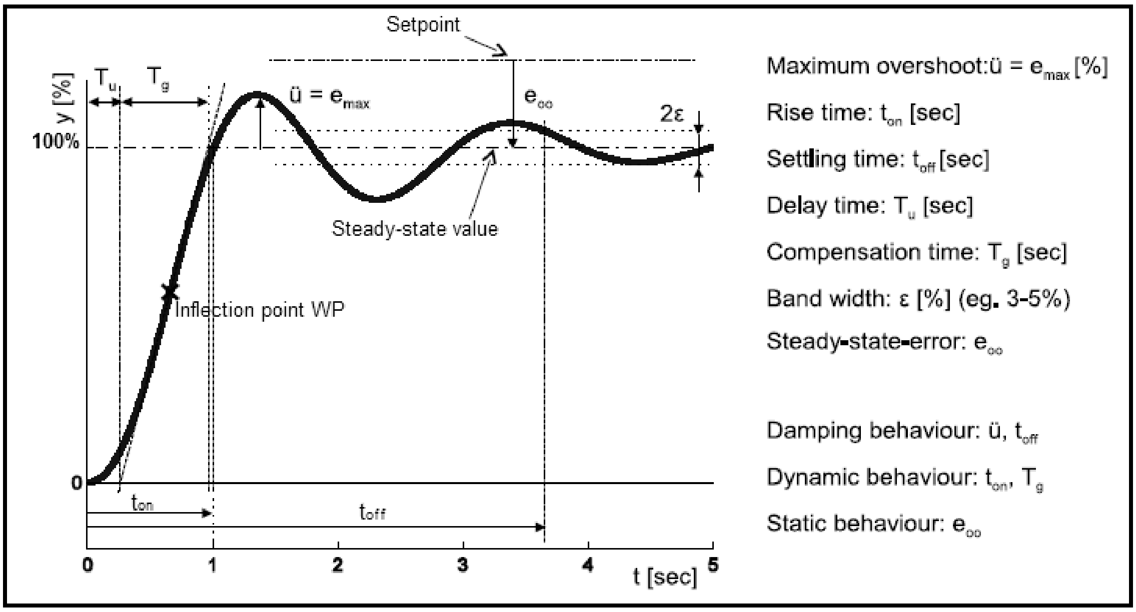


Figure 3.4-9 Accuracy in the time domain

Font: Bosch Group. Governor technique, basics and applications, Esslingen: Steinbeis Transferzentrum, 2008

Other way to analyze the response characteristic of a system is using a Bode diagram. Basically, a bode diagram is a graphic that describes the frequency response function F(jω). The complex frequency response function can be separated according to its absolute value |F(jω)| and the phase angle Φ(ω). Figure 3.4-10 shows an example.

The amplitude response describes how strong an input signal is amplified depending on the frequency. It is a ratio of the output to the input amplitude of a system. In the example on Figure 3.4-10, the system has higher gains in low frequency. For the gain response, the amplitude is logarithmically represented. Thus, the multiplication of individual transfer functions can take place as a graphic addition in the Bode diagram. Besides that, the amplitude is not represented as a physical ratio in the logarithmic scale, but as decibels (dB = 20.log A).

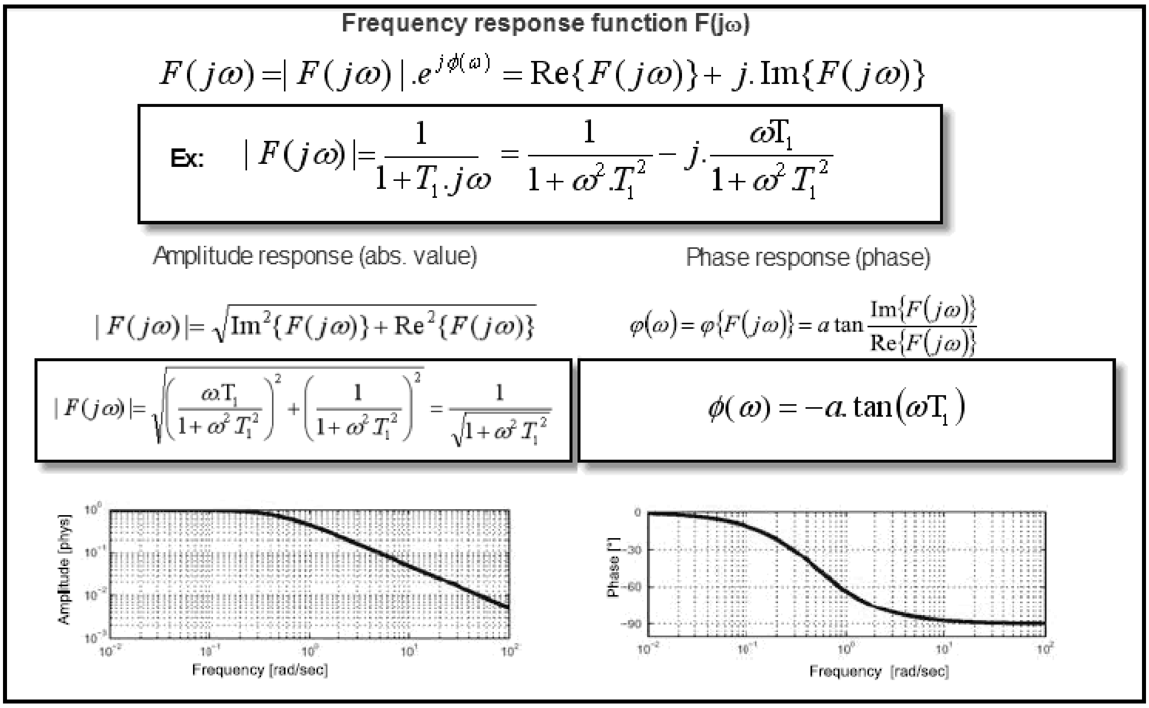


Figure 3.4-10 Amplitude and phase response with example
Font: Bosch Group. Governor technique, basics and applications, Esslingen: Steinbeis Transferzentrum, 2008 adapted

The phase response describes the shift of the phase between input and output signal. In the example on Figure 3.4-10, a phase of -90° is achieved, which is typical for a low-pass.

To understand the behavior of a certain system is necessary a knowledge of the basic elements in control engineering. Next three tables show the most important standard elements.

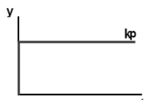
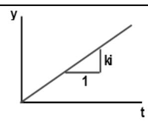
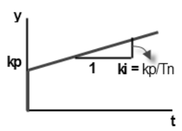
Element	Step Response	Transfer Function	Parameters
P		G(s) = k_p	K_p: proportionality constant
I		G(s) = k_i/s = 1/(T_i.s)	K_i: integration constant T_i: Time constant
D		G(s) = k_d.s	K_d: differentiation constant
PI		$G(s)=k_p+\frac{k_i}{s}$ $=k_p.\frac{T_n.s+1}{T_n.s}$	K_p: proportionality constant K_i: integration constant T_n: integral time

Figure 3.4-11 Standard Elements 1

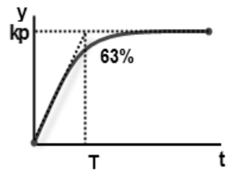
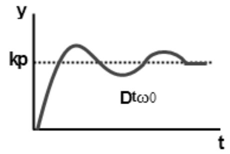
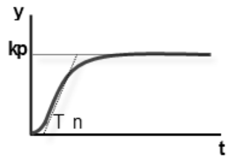
Element	Step Response	Transfer Function	Parameters
PT$_1$		$G(s)=\dfrac{k_p}{T.s+1}$	K_p: proportionality constant T_i: Time constant
PT$_2$		$G(s)=\dfrac{k_p}{\dfrac{1}{\omega_0^2}.s^2+\dfrac{2D}{\omega_0}.s+1}$	K_p: proportionality constant ω_0: angular frequency D: damping
PT$_n$		$G(s)=\dfrac{k_p}{(T.s+1)^n}$	K_p: proportionality constant T_i: Time constant n: degree of order

Figure 3.4-12 Standard Elements 2

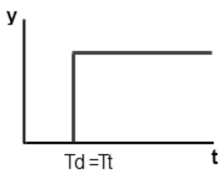
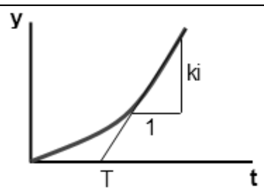
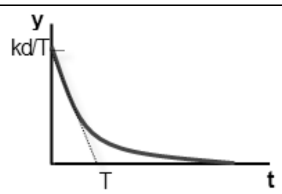
Element	Step Response	Transfer Function	Parameters
T$_t$		$G(s)=e^{-s.T_t}$	T_t: dead time
IT$_1$		$G(s)=\dfrac{k_i}{s.(T.s+1)}$	K_i: integration constant T: Time constant
DT$_1$		$G(s)=\dfrac{k_d.s}{T.s+1}$	K_d: differentiation constant T: Time constant

Figure 3.4-13 Standard Elements 3

3.5 PID CONTROLLER

The proportional, integral and derivative (PID) controller is the most used feedback controller. The input of a PID is the “error” variable which is desired setpoint minus measured variable.

The PID controller calculation (algorithm) involves three separate parameters; they can be interpreted as follows:

P: depends on the present error (fast reaction)

I: depends on the past errors (precise but slow reaction)

D: is a prediction of future errors (damping governor output depending from setpoint changes).

The PID’s transfer function is:

$$PID(s) = K_p + \frac{K_i}{s} + K_d \cdot s = \frac{K_d s^2 + K_p s + K_i}{s}$$

Looking to the block diagram on Figure 3.4-3 and considering the controller block as a PID controller, e(t) is the error variable. This signal will be the input variable of the PID controller, which will calculate the integral and the derivative. The output signal y(t) will be equal to the magnitude of the error multiplied by the proportional gain (k_p), plus the integral gain (k_i) multiplied by the error integral, plus derivative gain (k_d) multiplied by the error derivative.

$$y(t) = K_p e(t) + K_i \int e(t)dt + Kd\frac{de(t)}{dt}$$

Using as reference signal a step, Figure 3.5-1 shows the effect of increasing each one of the PID parameters.

Parameter	Rise time	Overshoot	Settling time	Steady-state Error
K_p	Decrease	Increase	Small Change	Decrease
K_i	Decrease	Increase	Increase	Eliminate
K_d	Small change	Decrease	Decrease	Small change

Figure 3.5-1 PID behavior

The proportional component is responsible to strengthen the input signal. It acts in the transitory response of the system decreasing the rise time and decreasing the steady state. As k_p increases, the system becomes more dynamic, as it is possible observe in Figure 3.5-2. The disadvantage is that if k_p is too high, can occurs in an instability, but the control deviation persist.

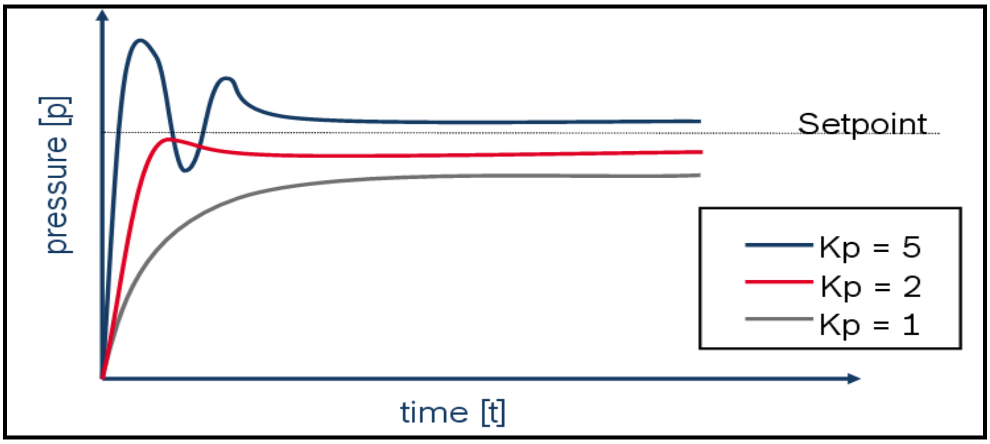


Figure 3.5-2 P controller behavior

The integral component reacts to deviations between set point and actual values. It eliminates the steady-state, but can deteriorate the transitory response (Figure 3.5-3). The advantage of this controller is the control deviations are adjusted automatically, but adjusts are more slowly than P-component.

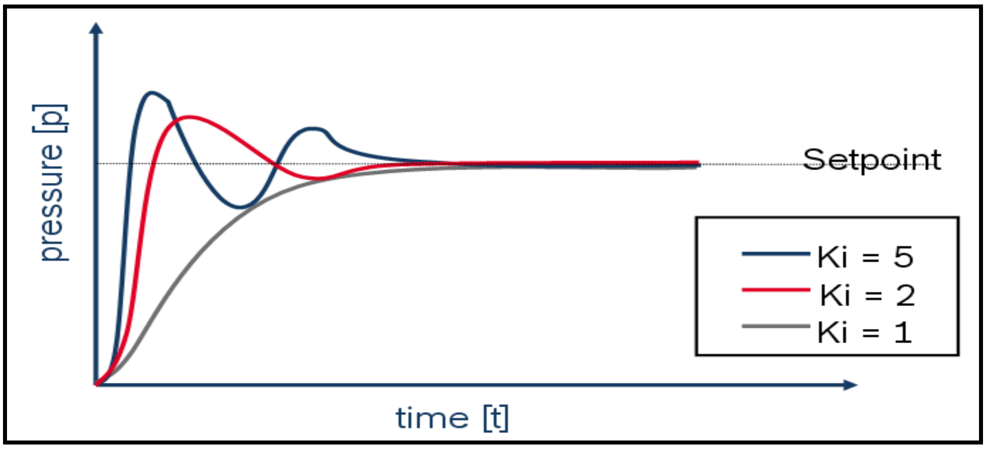


Figure 3.5-3 I controller behavior

The derivative component has a damping effect on the system when the set point changes (Figure 3.5-4). It increases the stability of the system, decreasing the overshoot and increasing the transitory response.

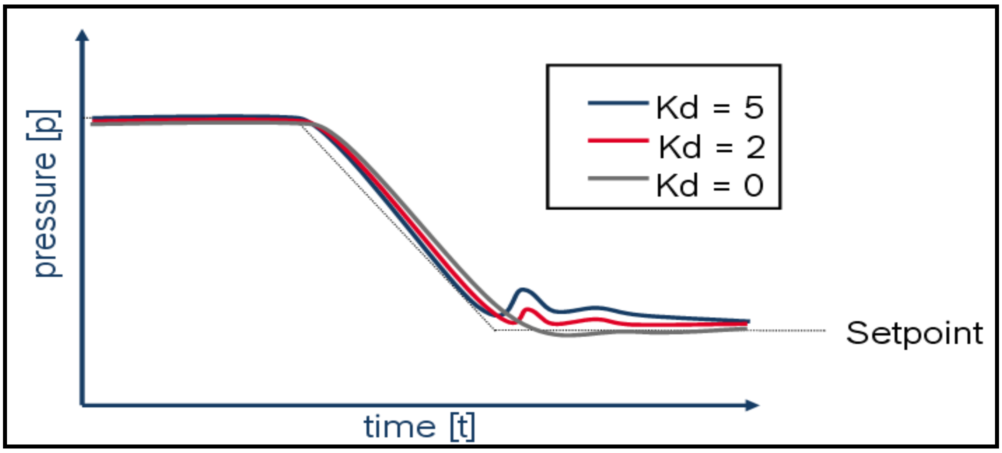


Figure 3.5-4 D controller behavior

In a PI-controller the I-component is decisive in the static behavior, since the I-element executes the automatic offset adjustment. The P-element does not contribute to this in the steady state (control deviation = 0).

With regard to speed and thus rise time the P-component is dominant in the PI-controller, as is possible observe in Figure 3.5-5.

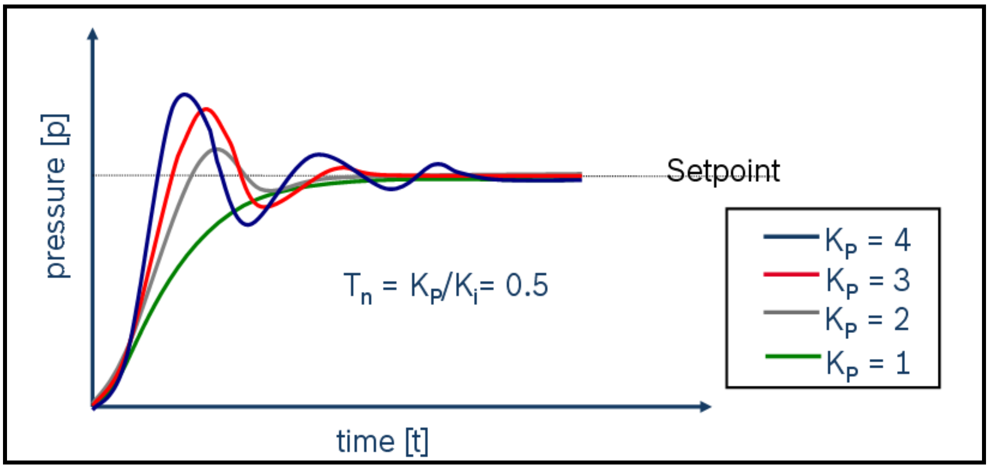


Figure 3.5-5 PI controller behavior 1

The I-component is responsible for reaching the steady state (see Figure 3.5-6). It determines "how" the steady-state value is reached. A large integration constant k_i (small T_n) causes fast reaction to a control deviation but also makes an "overreaction" possible. Small values of k_i lead to more sluggish settling.

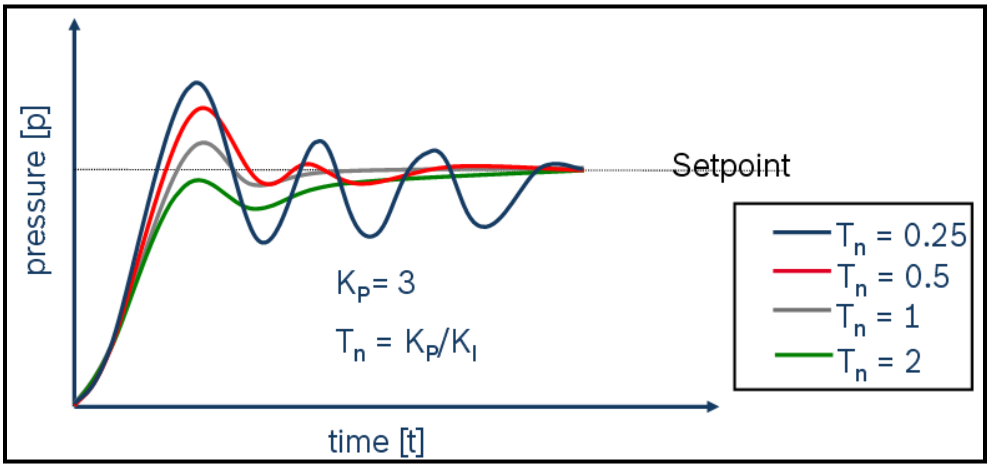


Figure 3.5-6 PI controller behavior 2

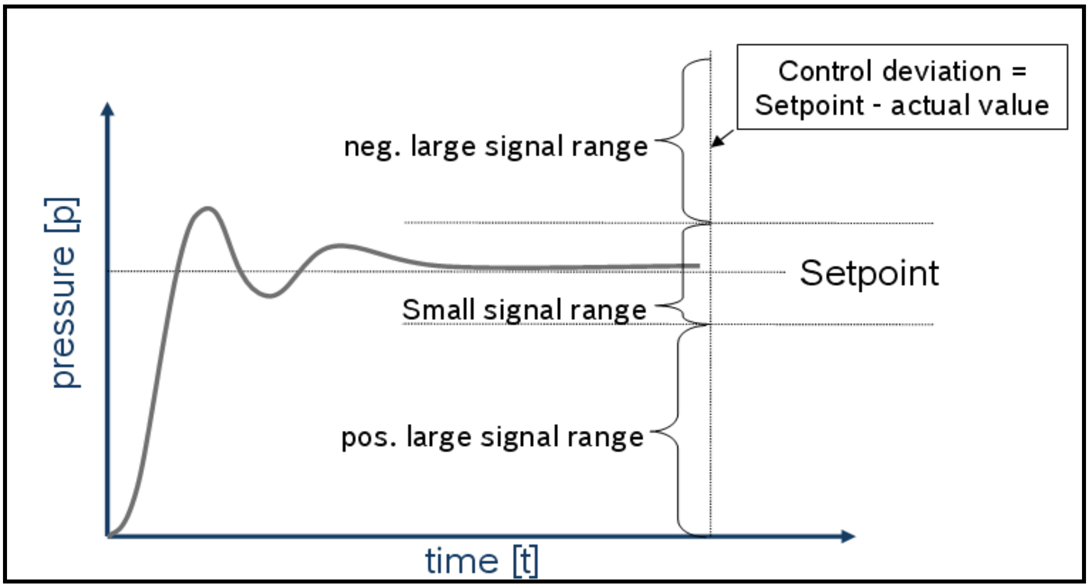


Figure 3.5-7 Window switchover

A trade-off between accuracy, stability and dynamic leads to a necessity of different parameters in each 'part' of the signal. Figure 3.5-7 shows the definition of window switchover. The idea is that with large P component the adjustment is fast, but the overshooting is large. That is why the signal is divided in small and large signal. When the control deviation is larger than a certain value defined in the project, a larger k_P is used, but if the deviation is smaller, it is necessary a smaller p-component. Depending on the sign of the control deviation, a distinction is made between positive and negative large signal values (see Figure 3.5-8).

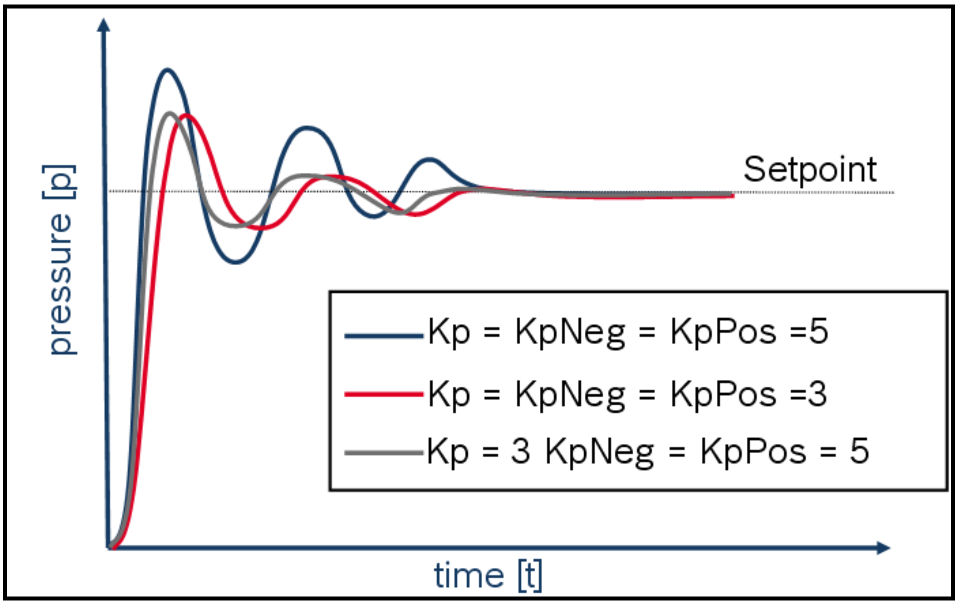


Figure 3.5-8 Window switchover behavior

At the limits between large and small signal range the change of P-component value must not be hard, otherwise steps would arise at the controller output. To solve this problem, the controller output of the P-controller is back-calculated accordingly which corresponds to a shift of the controller curve (see Figure 3.5-9). Such a discontinuity in the switching of the controller parameters does not occur with an I-controller. An alteration to the k_i only leads to an alteration of the integration behavior; it does not lead to a jump-shaped alteration at the output of the integrator.

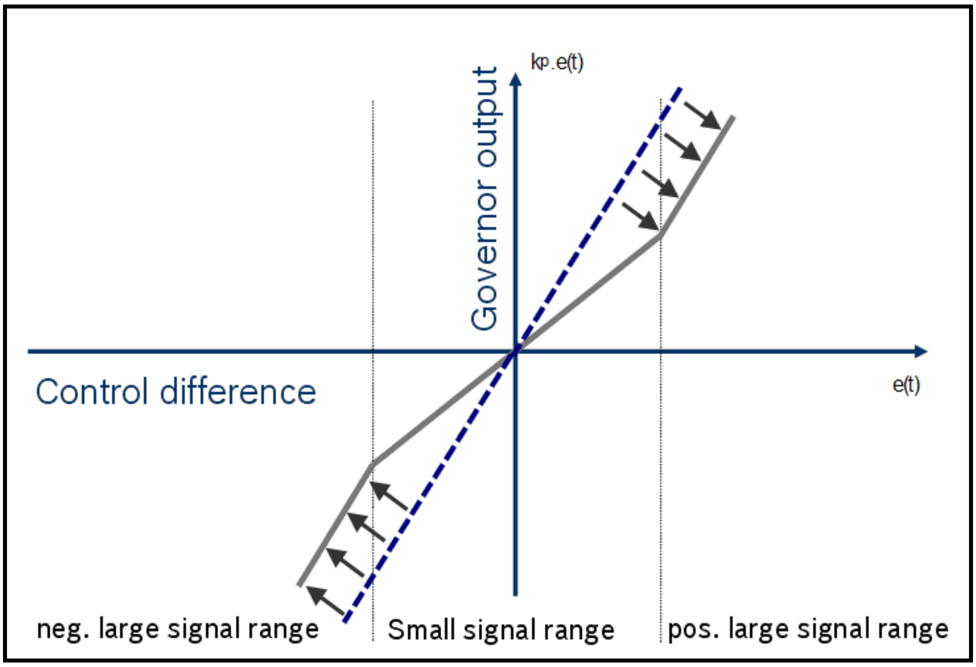


Figure 3.5-9 Continuity in P window switchover

Other important point to project a controller is the physical limitation conditions. Due to that, the values of the correcting variable may not take on every value. As a simple example, a valve cannot be more than completely closed or opened.

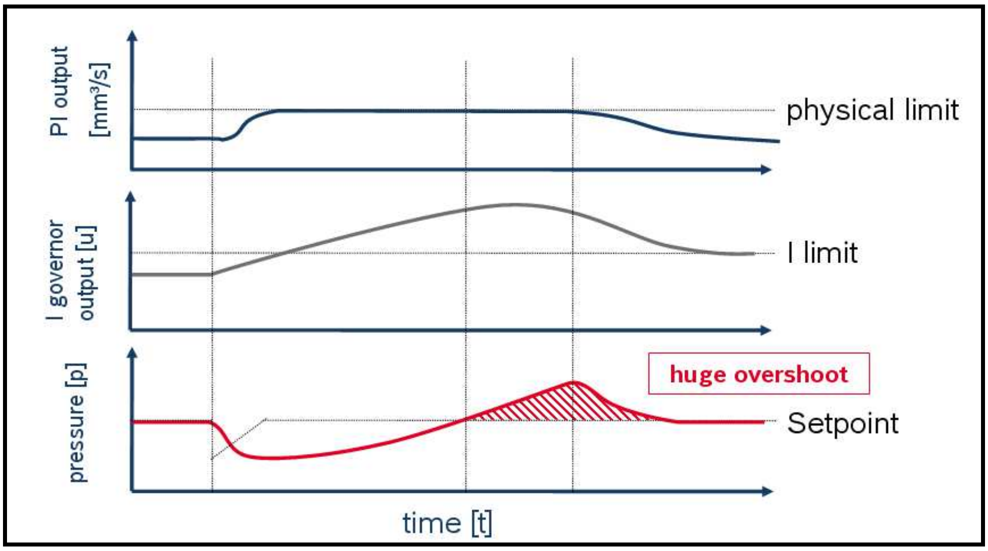


Figure 3.5-10 Wind-up effect

For the P-controller the limitation becomes visible through to worse dynamics. However, no further "dirt" effects occur.

The I-controller reacts to limitations with the so-called "Wind-up-effect" due to its integrating behavior (see Figure 3.5-10).

To avoid this effect, I component is frozen, limiting the I set value, as it is possible to see in Figure 3.5-11.

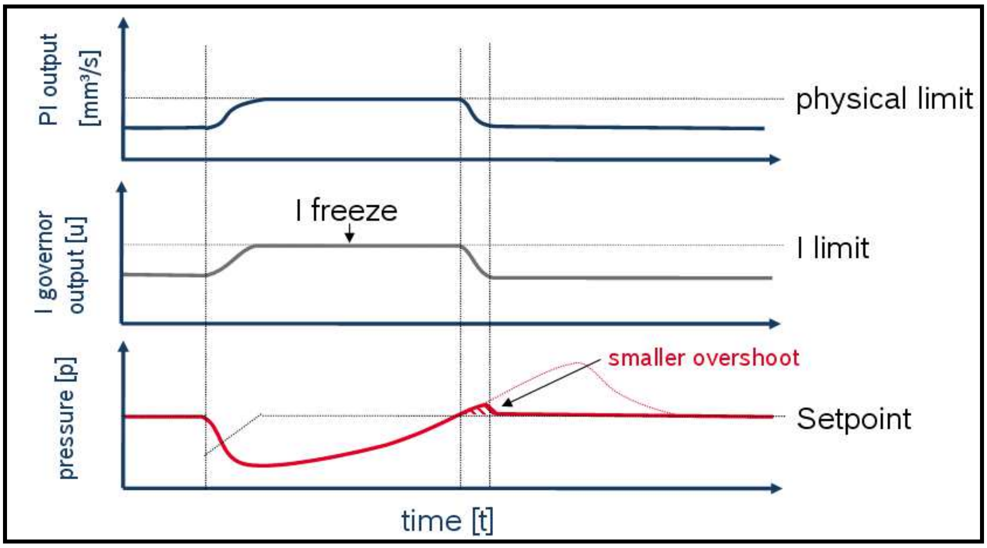


Figure 3.5-11 Avoiding windup effect

3.5.1 Tuning Rules

Different methods can be applied to find the parameters of a controller. To decide what method is the best to the project it is necessary analyze the characteristic of the controlled system and what measured value is available.

Ziegler and Nichols conducted numerous experiments and proposed rules for determining values of k_p, k_i and k_d based on the transient step response of a plant.

In order to be able to apply the classic Ziegler-Nichols (ZiNi) methods, the controlled system must have a behavior which is similar to the behavior of the series connection of a dead time element with a PT1-element (Figure 3.5-12).

There are two methods of ZiNi. To use the first one is necessary knowing the step response of the system.

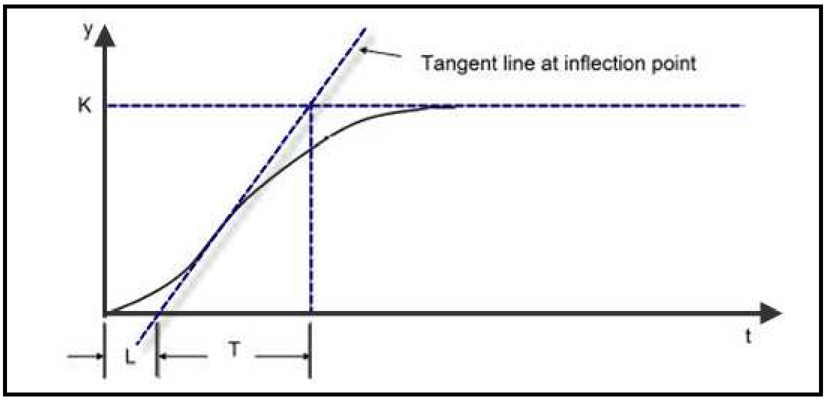


Figure 3.5-12 ZiNi Method 1

The reaction curve can be characterized by two constants, delay time L and time constant T, which are determined by drawing a tangent line at the inflection point of the curve and finding the intersections of the tangent line with the time axis and the steady-state level line. Using the parameters L and T, it is necessary set the values of k_p, k_i and k_d according to the formula shown in the table below.

Table 3.5-1 ZiNi – Method 1 [OGATA, 2003]

Controller	k_p	k_i	k_d
P	T/L	∞	0
PI	0.9*T/L	L/0.3	0
PID	1.2*T/L	2L	0.5*T

These parameters will typically give a response with an overshoot about 25% and good settling time.

The second method of ZiNi the stability limit of the closed control loop is sought.

Table 3.5-2 ZiNi - Method 2 [OGATA, 2003]

Controller	k_p	$T_n = k_p/k_i$	$T_v = k_d/k_p$
P	$0.5*K_{Pcrit}$	∞	0
PI	$0.45*K_{Pcrit}$	$0.83*T_{crit}$	0
PID	$0.60*K_{Pcrit}$	$0.5*T_{crit}$	$0.125*T_{crit}$

For that, the integral controller components are switched off and the control loop is only operated with a P-component. The proportionality constant is increased until the control loop begins to oscillate. If the control loop oscillates, the critical parameters of the controlled system, k_{Pcrit} and T_{crit}, can be determined (see Figure 3.5-13). The controller parameters are calculated according to the formula shown in Table 3.5-2.

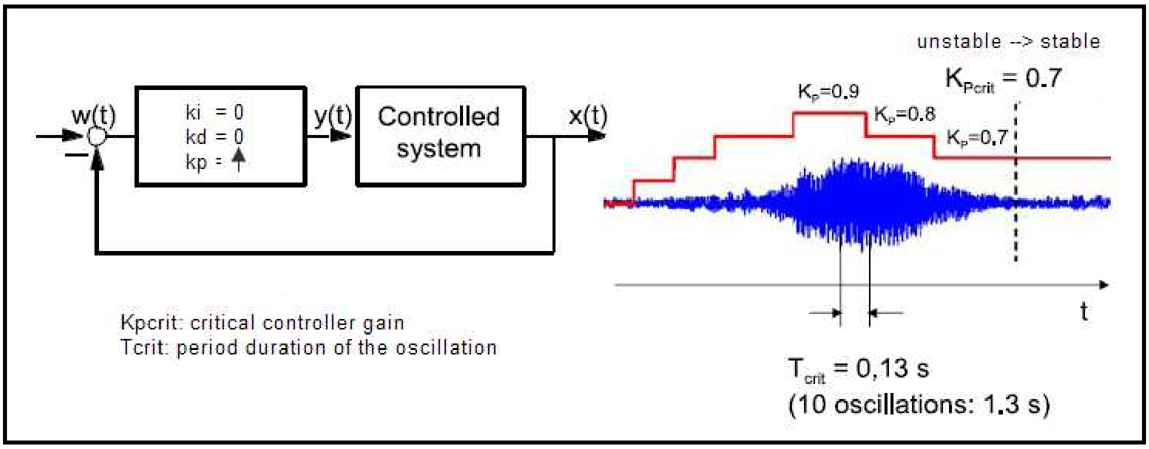


Figure 3.5-13 Determining K_{Pcrit} and T_{crit}

Font: Bosch Group. Governor technique, basics and applications, Esslingen: Steinbeis Transferzentrum, 2008

Other tuning rule that will be used in this project is the Bode method. The Bode diagram can be obtained directly from the measurement of the amplitude and phase response; or from the transfer function.

These methods use the desired stability of a system to find parameters. In the process, the stability can either be defined as phase margin or as gain margin.

For the phase margin method, a desired phase margin is predefined for the overall system (controller plus controlled system).

To find k_p and k_i the follow steps should be done:

1st: Phase crossover frequency (ω_{pc}) need to be found. This is the frequency that gives a phase of -180°.

2nd: A phase margin needs to be defined and found in the Bode diagram. The frequency in this point is called ω_k.

3rd: The gain at ω_k need to be found. In order to retain the set phase margin for the overall system, the amplitude may be a maximum of 1 at this point ω_k, $A(\omega_k)$ multiplied per k_P is equal to one. This is the maximum possible controller gain (for this phase margin).

4th: the phase response of the PI-element is superimposed in such a way that the set phase margin is not reduced. This is made possible by choosing the frequency of the inflection point of the PI-phase ω_n one decimal power smaller than the frequency ω_k. In this way, no substantial contribution is subtracted from the phase at the point ω_k (the phase response of a PI-element change from -90° to 0° within two decimal powers). Figure 3.5-14 summarize the method of phase margin using the bode diagram.

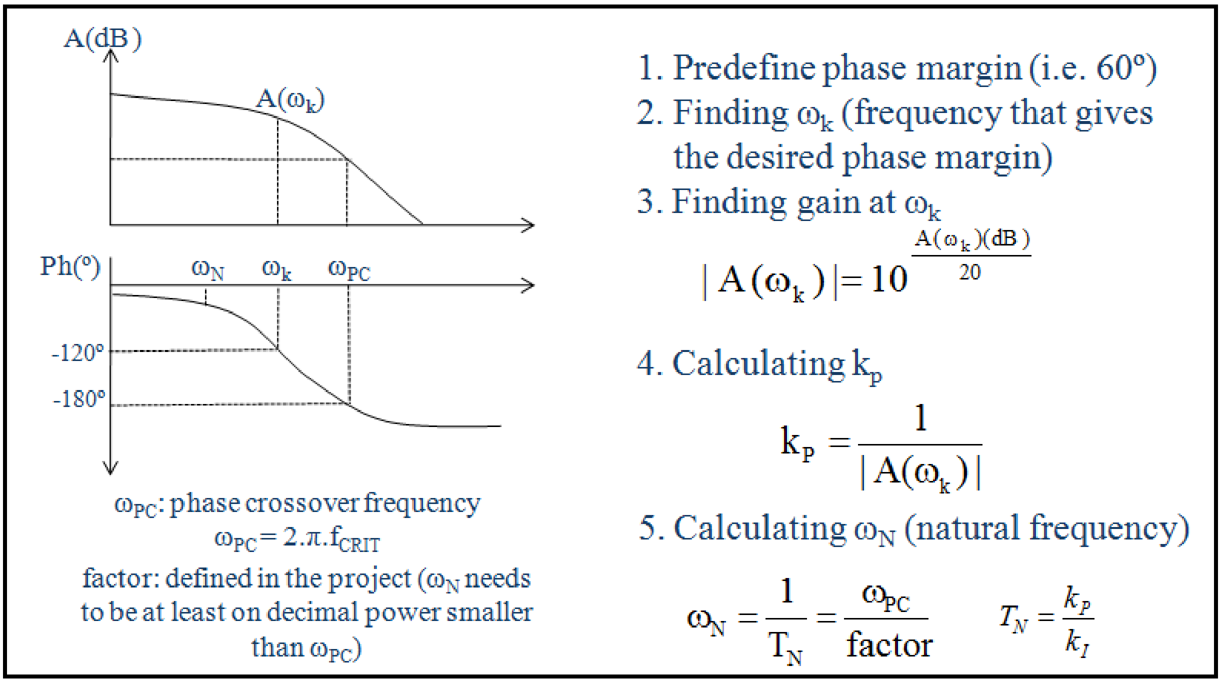


Figure 3.5-14 Bode diagram - phase margin method

For the gain margin method, a desired gain margin is predefined for the overall system (controller plus controlled system).

For a stable system, the gain of the open control loop must be below one at the phase crossover frequency (phase = -180°) (Nyquist criterion). If the overall system should have a gain margin, the result of the multiplication of the controller gain, gain of the controlled system and the gain margin must be equal to 1. Figure 3.5-15 shows step-by-step how to use the gain margin method.

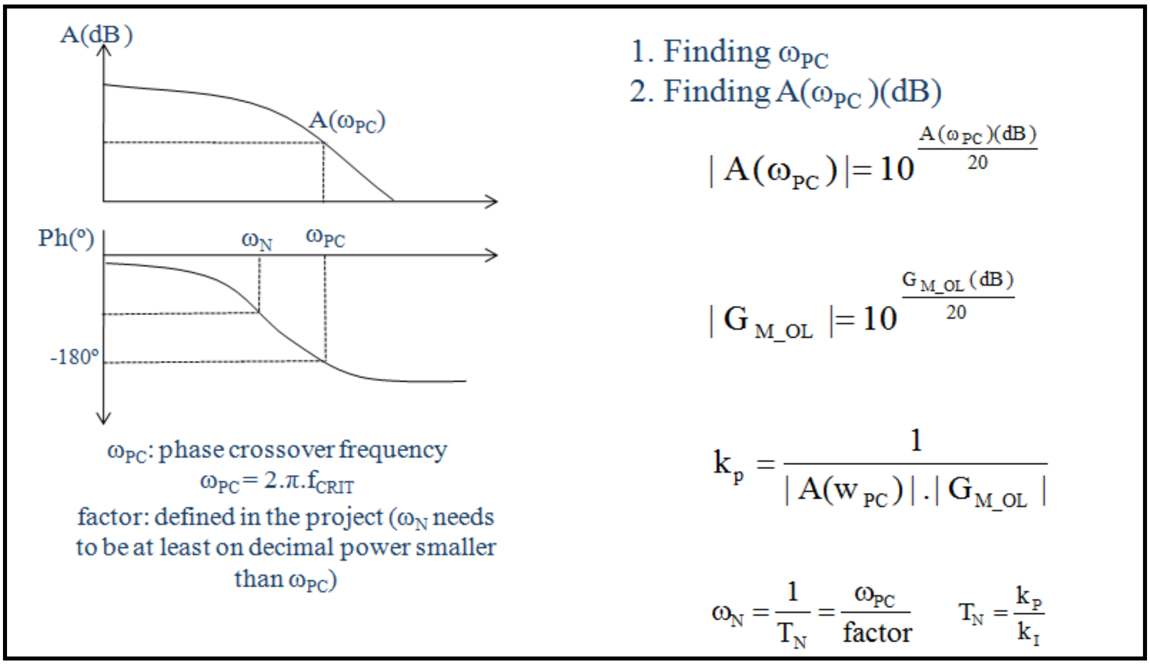


Figure 3.5-15 Bode diagram - gain margin method

As an example, Figure 3.5-16 shows a simulation of a bode diagram. The gain margin method will be applied to this bode diagram. The desired margin is 9,55dB and the ω_n will be twenty times smaller than the frequency ω_{pc}.

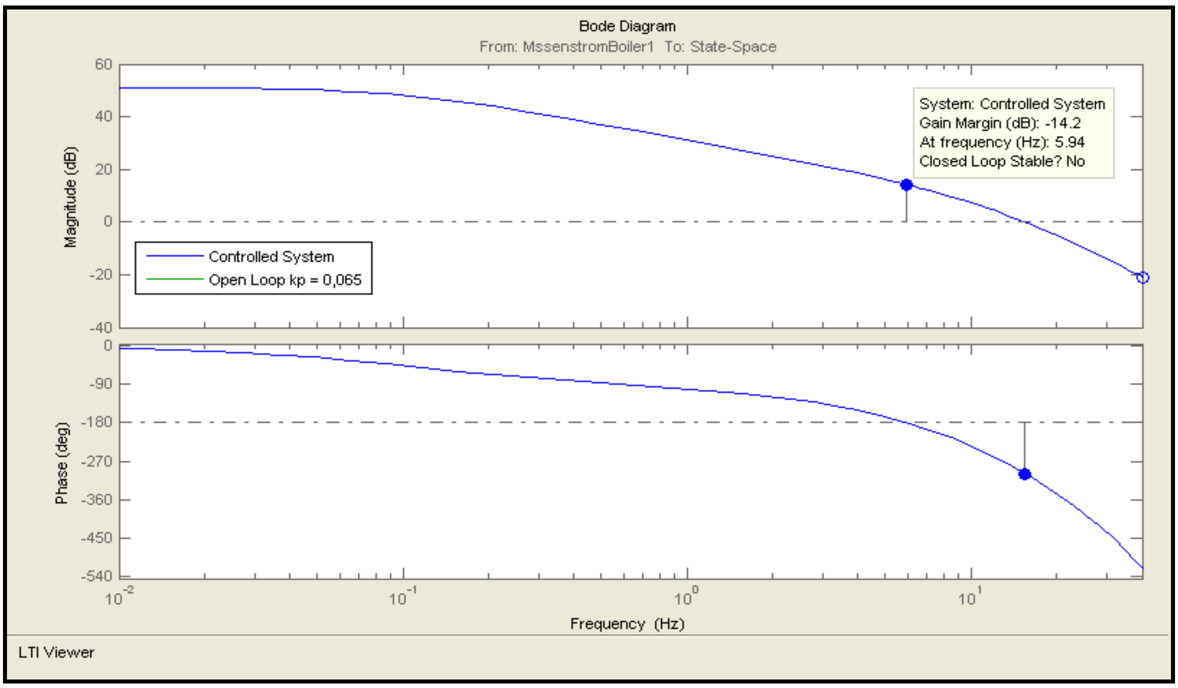


Figure 3.5-16 Simulated bode diagram 1

First it is necessary to know the crossover frequency. In Figure 3.5-16 it is possible to see that the crossover frequency is 5,94 Hz. The gain in this frequency is 14,2 dB – which means a gain margin of -14,2 dB.

$$f_{crit} = 5{,}94 Hz$$
$$A(w_{PC}) = 14{,}2 \text{dB}$$
$$|A(w_{PC})| = 10^{\frac{14{,}2}{20}} = 5{,}13$$

Second, defined the desired gain (9,55dB), k_p can be calculated. Figure 3.5-17 shows what happened in the bode diagram when a k_P is added.

$$G_{M\_OL}(\text{dB}) = 9{,}55$$
$$|G_{M\_OL}| = 10^{\frac{9{,}55}{20}} = 3{,}003$$
$$k_p = \frac{1}{5{,}129 \cdot 3{,}003} = 0{,}065$$

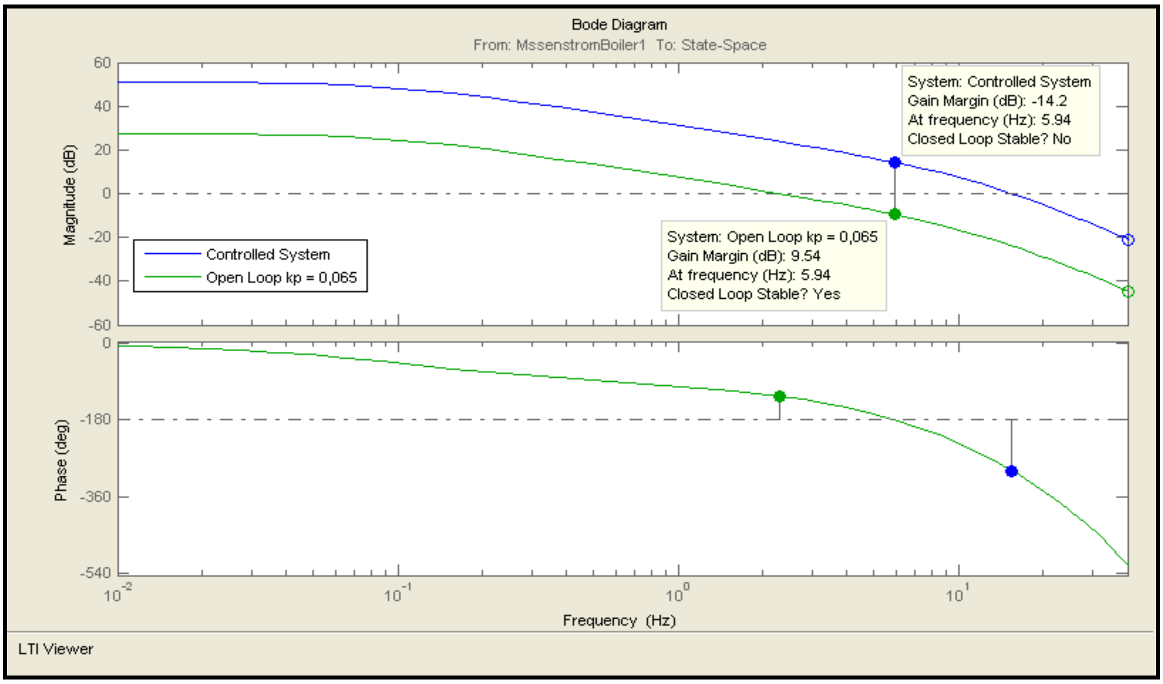


Figure 3.5-17 Simulated bode diagram 2

And, the last step is adding an I-gain to the system. As defined before, the natural frequency should be twenty times smaller than the crossover frequency. Figure 3.5-18 shows in red the bode diagram of the open loop system with k_i and k_p. The desired margin was reached.

$$w_N = \frac{1}{T_N} = \frac{w_{pc}}{20} = \frac{2\pi .5{,}94}{20} = 1{,}867$$

$$w_N = \frac{1}{T_N} = \frac{k_i}{k_p} \therefore k_i = 1{,}867 * 0{,}065 = 0{,}12$$

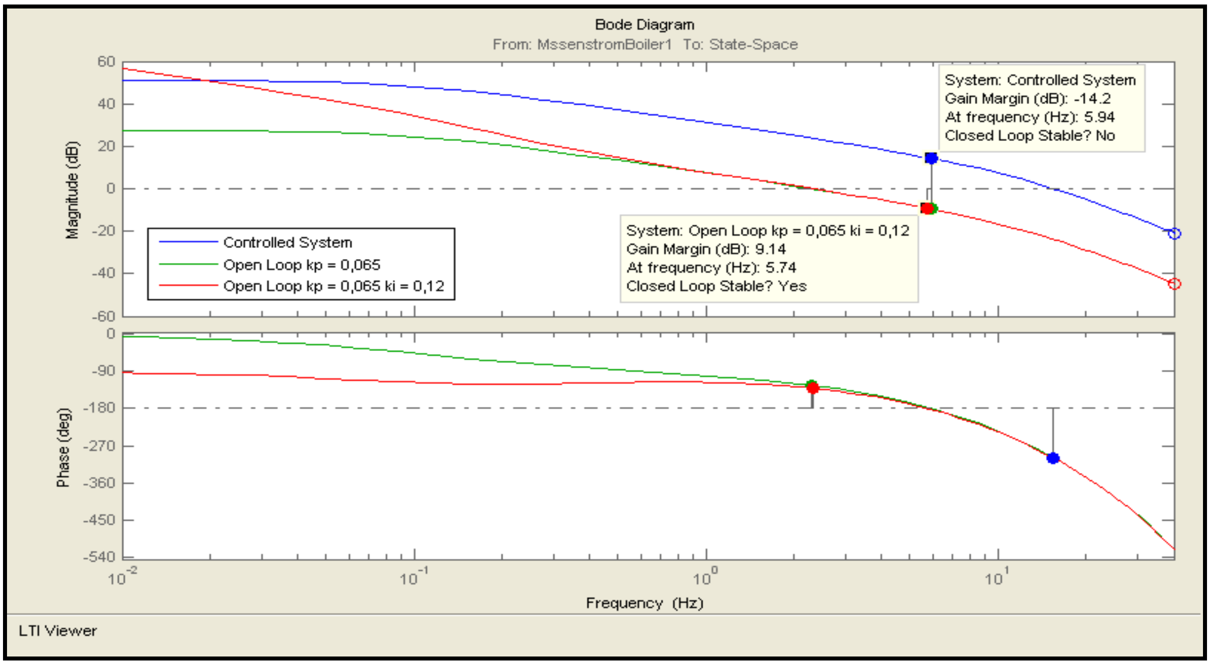


Figure 3.5-18 Simulated bode diagram 3

3.6 RAIL PRESSURE CONTROL

The rail pressure control uses the concept of two actuators: pressure control valve (PCV) and a metering unit (MeUn). At start and in a cold state the control takes place via the PCV. Otherwise, control takes place via MeUn. The combination of these two actuators brings the advantage of a fast pressure reduction via the valve with the advantage of the high efficiency of the pressure generation.

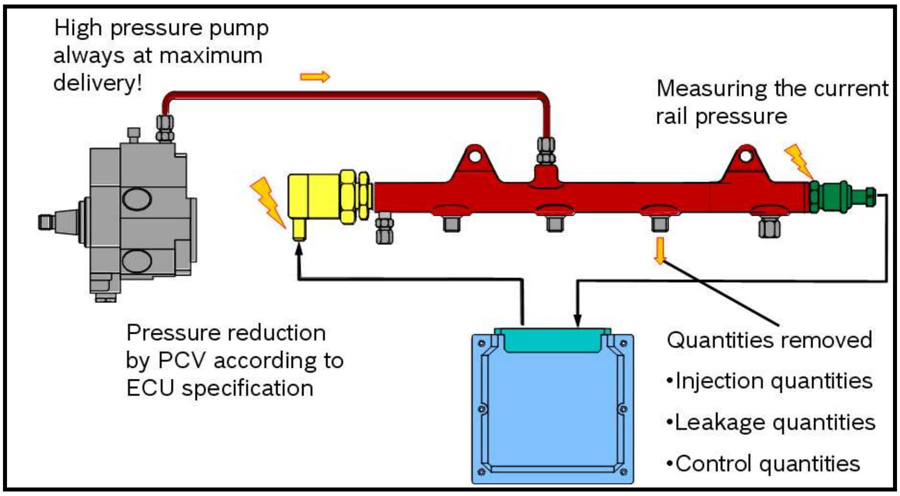


Figure 3.6-1 RP controlled by a PCV
Font: Bosch Group. Introduction to the CR system, Esslingen: Steinbeis Transferzentrum, 2008 adapted

A structure of a rail pressure control via the PCV or via the MeUn are very similar, both can be displayed with the same flowchart. Figure 3.6-2 shows a simplified structure of rail-pressure control.

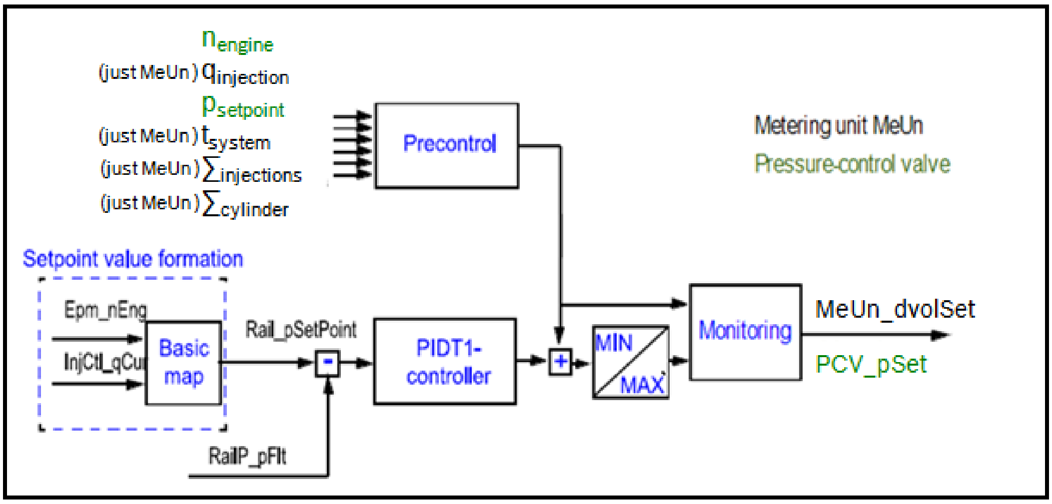


Figure 3.6-2 Simplified structure of rail-pressure control

Font: Bosch Group. Governor technique, basics and applications, Esslingen: Steinbeis Transferzentrum, 2008

The controlled system of pressure control valves behaves like a PT1-element with dead time. The controlled system of the metering unit can be interpreted as IT1-behaviour with dead time - because of continuous fuel injection there is a "displacement" to a PT1-behaviour.

Both controllers have the same task, both receive the same control deviation. However, their overall regulation systems are different because of the different actuators.

The software control architecture of the high pressure control is described in the figure below.

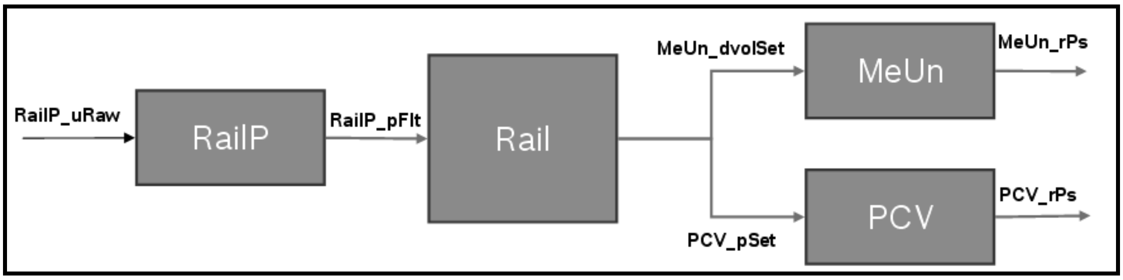


Figure 3.6-3 High pressure control

The RailP is responsible for rail pressure sensing, anti-aliasing function and ISP function (reduces systematically rail pressure measure errors to increase injection accuracy).

The Rail calculates PCV/MeUn parameters and PCV/MeUn pre-control. It is responsible for the calculation of the control modes (State machine), calculation of setpoint value (volume flow) of rail pressure governing for MeUn and PCV (labels called MeUn_dvolSet and PCV_pSet in case of EDC17 dataset).

The governor parameters are a function of engine speed, since the plant gains depend of the engine speed. It is necessary calibrated seven curves per actuator: k_p, k_i, k_d, k_{ppos}, k_{pneg},

k_{ipos} and k_{ineg}, which represents the value of P, I and D and the values of the window switchover of P and I.

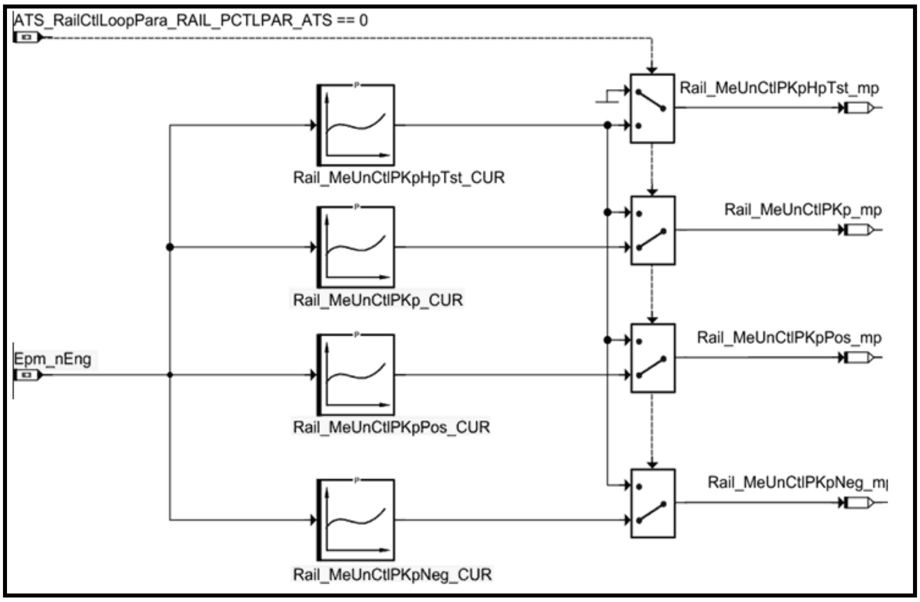


Figure 3.6-4 MeUn governor P Gain
Font: BOSCH Group, High pressure generation, Calibration hints (internal document), Stuttgart: Bosch, 2010

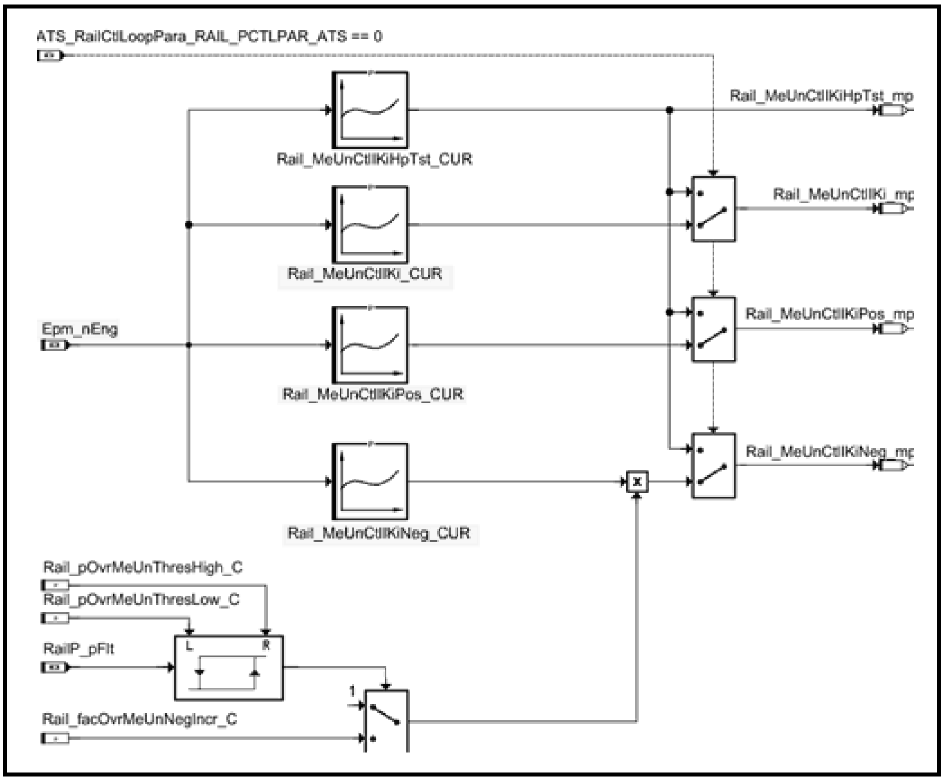


Figure 3.6-5 MeUn governor I Gain
Font: BOSCH Group, High pressure generation, Calibration hints (internal document), Stuttgart: Bosch, 2010

In the MeUn governor D-gain (see Figure 3.6-6), DT1 and PT1 values are standards available.

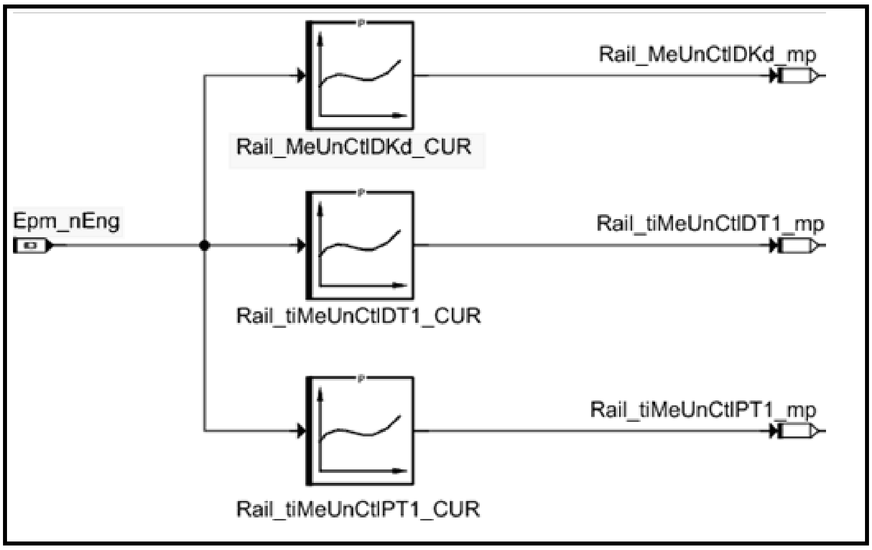


Figure 3.6-6 MeUn governor D Gain
Font: BOSCH Group, High pressure generation, Calibration hints (internal document), Stuttgart: Bosch, 2010

The same 7 curves need to be calibrated for the PCV actuator.

The MeUn software component is responsible for the MeUn current governor and actuator protection function. And the PCV software component calculates PCV current governor and APCV function.

The control mode depends of engine speed and injection mass. When there is a low fuel temperature and in the startup, the PCV is controlling. In overrun and at low injection quantities the Coupled Pressure Control (CPC) is active. In a high injection quantity and high rail pressure the MeUn control is actuating.

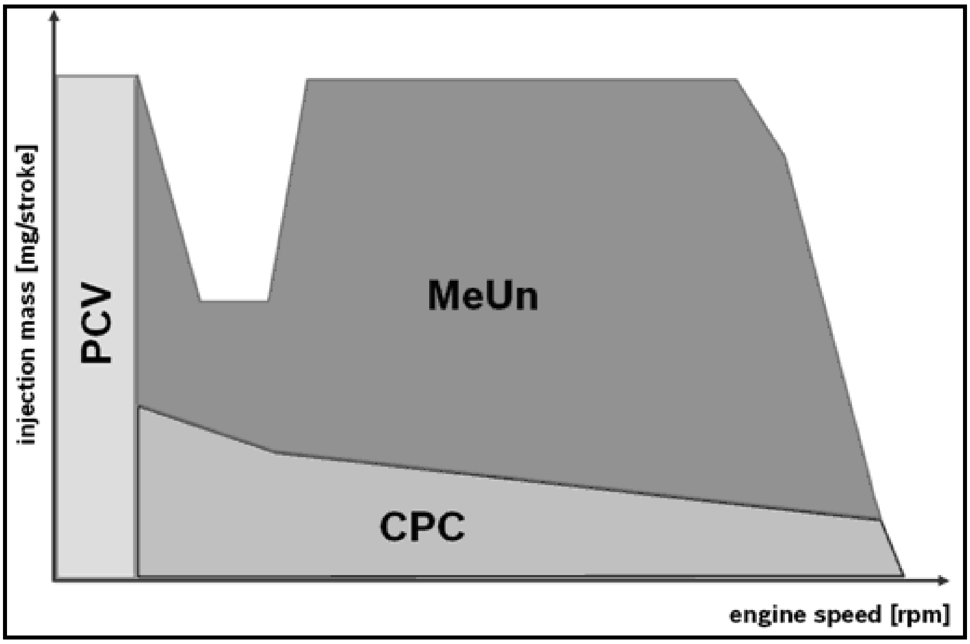


Figure 3.6-7 Control mode ranges
Font: BOSCH Group, High pressure generation, Calibration hints (internal document), Stuttgart: Bosch, 2010

3.7 ADVANCED SIMULATION FOR CALIBRATION

Advanced Simulation for Calibration (ASC) is a method for modeling the input/output behavior of unknown systems based on measuring data that are obtained using methods of the design of experiments (DoE). Advanced Simulation for Calibration, Modeling and Optimization (ASCMO) is the tool used for this kind of simulation.

If a precise description of the system is not possible, one solution is using this data-based modeling. With ASCMO is possible find high model quality to very complex systems as, for example, the global behavior of an internal combustion engine.

DoE is a method for data-based modeling of unknown system. The idea is in a first step to have a test plan to obtain the data, using the minimum measuring effort. This data will be used to train the model. The model is based on mathematical approximation methods and is capable of reproducing the behavior of the measured system.

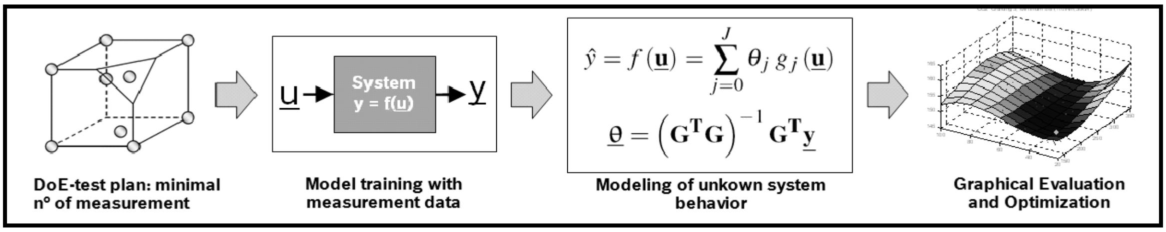


Figure 3.7-1 From experiment plan to model-based optimization
Font: ETAS GmbH: ASCMO 4.0 user's guide. Stuttgart, 2010

The classical experiment plan of the first generation describes a behavior of a system based on polynomial. The advantages are a comprehensible and established process since many tools are available. However, it is necessary a high parameterization and it's not a robustness process with respect to outliers.

Other process that can be used to describe a system is called neural networks. This one allows mapping complex relationships and the process is relatively illustrative, but, as the classical method, can be necessary a high parameterization effort and there is a risk of over fitting. Besides that, a high number of measuring data can be required.

ASCMO belongs to the DoE of the 2nd generation which is based on statistical learning processes. This process consists of an optimal relationship between measuring effort (low) and model accuracy (high). Besides that, no parameterization is required and it is robust with respect to outliers. The disadvantages are the low illustrative of the theory and a relatively large system memory of the PC.

The classic experiment plans are based on a grid-shaped (Figure 3.7-2) or star-shaped (Figure 3.7-3) measurement of the experimental space.

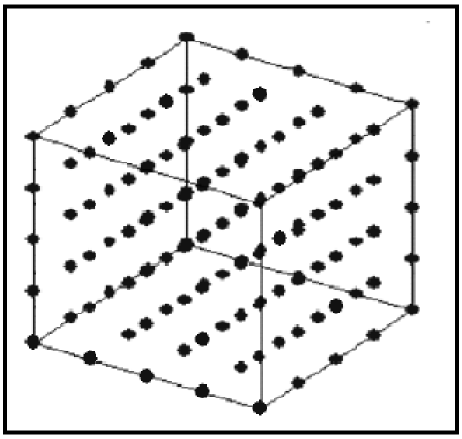


Figure 3.7-2 Grid: 3 parameters, 5 steps, 125 measurements
Font: ETAS GmbH: ASCMO 4.0 user's guide. Stuttgart, 2010

In principle the grid measurement can be used for all types of modeling, but the problem is that the number of measurements points when one more parameter is necessary needs to increase exponentially. And despite the high number of measurements, an optimal coverage of the parameter space is not achieved.

For the star-shaped measurement, just one parameter is varied at a time. That leads to few points to be measured if compared to the grid shaped. The disadvantage is the relationship between the parameters is neglected, so it is not possible find the actual optimal model frequently, which means that is unusable to model complex systems.

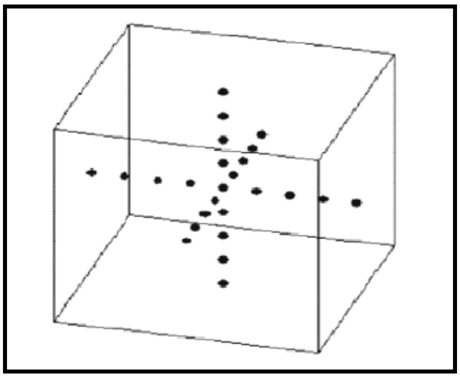


Figure 3.7-3 Star-shaped: 3 parameters
Font: ETAS GmbH: ASCMO 4.0 user's guide. Stuttgart, 2010

As alternative to the aforementioned processes are the experiment plans according to DoE of the latest generation, the so-called D (or V) optimal plans and space-filling experiment plans according to DoE.

The D (or V) optimal plans are specifically adapted to polynomial models whose precise specification is dependent upon the polynomial order. The problem is to create this experiment plan is necessary prior knowledge about the system behavior. That is the reason that this kind of experiment plan is not used with ASCMO.

Space-fillings plans (Sobol, Latin Hypercube, etc.) are the best option to use with ASCMO. They are characterized by an even distribution of the measuring points in the parameter space and an optimal coverage of all parameter levels. No previous knowledge about the system to be measured is required, and the data gained are perfectly suited for model training in ASCMO.

The tool so-called EDOR is used to generate this space-fillings experiment. In page 66 there is a brief explanation of EDOR tool.

Some points need to be analyzed before start to use this experiment plan. First, the result needs to be reproducible. In particular, the drift of measured variables must be ruled out or at least recognized and responded to accordingly.

It is necessary to measure several repetition points to determine the experiment repeatability and to estimate the time drift. Besides that, measurements must be performed at a random order of the individual parameters.

The advantages of the DoE methodology are significant reduction of the measuring effort, if the optimization criteria need to be changed is not necessary a new measurement and the models allow a description, interpretation and documentation of the system behavior, such as interactions between parameters.

A series of options are available for evaluating the models created with ASCMO, they include:

- "Display of measured values in comparison with the model prediction"
- "Error depending on the size of the training data"
- "Display of measurements as "Adjusted Data""
- "Display of Sigma in intersection plots"
- "Overview of model statistics"

The first option, "Display of measured values in comparison with the model prediction", it is possible plot some graphics to see the model error.

In the true prediction plot (Figure 3.7-4) the measuring points are displayed on the y-axis and the model prediction on the x-axis. A perfect match between the two would result in a "pearl necklace" (y = x) on the dotted line drawn.

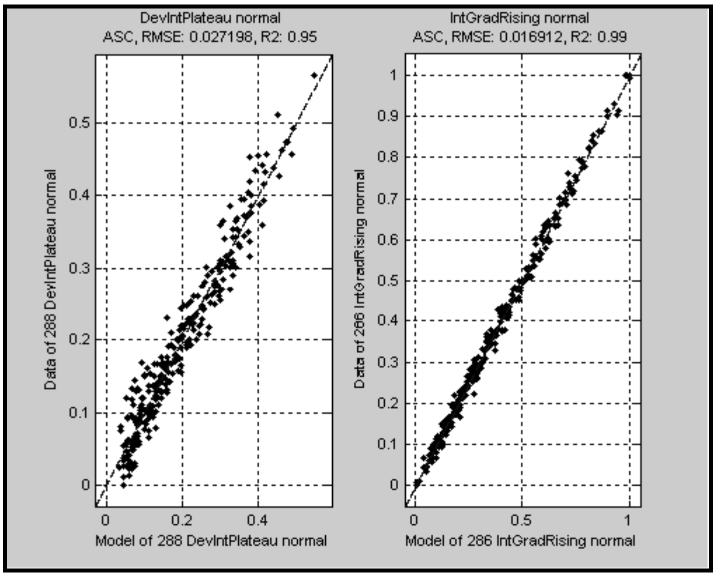


Figure 3.7-4 Example of a true prediction plot

Error depending on the size of the training data is other option available for evaluating the models created with ASCMO. This function helps in evaluating the degree to which the size of the training data used affects the model quality.

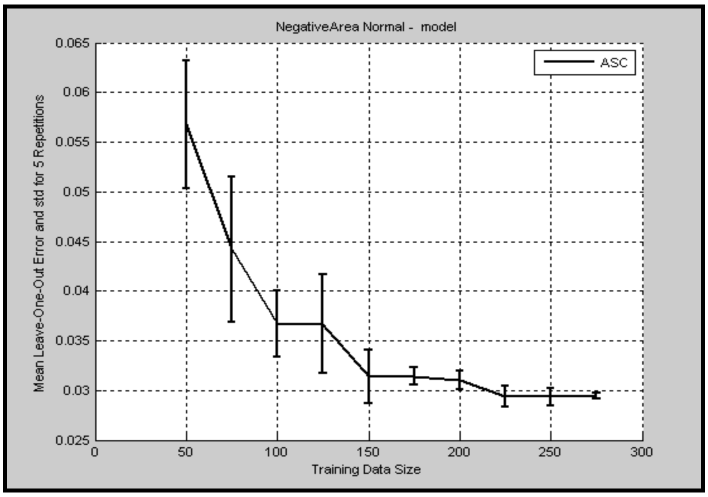


Figure 3.7-5 Analysis Model LOO error for output

It is necessary define the start training size for the investigation and the interval to the next data size. So, each subset of the training data are selected for the analysis and in each case the Leave-One-Out error is determined. The bar shows the variance of these five results, the solid line shows the mean value of the results.

This allows identifying whether the model improves if more training data are used or if the size of the training data can even be reduced since no appreciable model improvement can be achieved starting at a certain size.

The Leave-one-out is a method that n models are formed, each one with n-1 training data. Afterwards, the model error of the one data point that was not involved in the model training is being determined. The advantage of this method is that it enables a realistic model evaluation without incurring an additional measuring effort.

Some variables are used to quantify the model quality. Root Mean Square Error and Coefficient of determination are the mainly ones.

The Root Mean Square Error (RMSE) is defined as:

$$RMSE = \sqrt{\frac{SSR}{n}}$$

Where n is the number of measuring data and SSR is the Sum of Squared Residuals.

$$SSR = \sum_{i=1}^{n} (X_{i,pred} - X_{i,meas})^2$$

The RMSE describes the variance to be expected (standard deviation) about the model.

The coefficient of determination (R^2) is derived from the comparison of the variance that means after the model training (SSR) with the variance concerning the mean value of all measuring data.

$$R^2 = 1 - \frac{SSR}{SST}$$

Where SST is the Squares Sum Total:

$$SST = \sum_{i=1}^{n} (X_{i,meas} - \overline{X}_{meas})^2$$

R^2 is a relative measure for evaluating the model error – it indicates which portion of the total variance of the measuring data is described by the model. This variable is a number between zero and one, and can be analyzed in the following evaluations:

$0<R^2<0.6$: The model is not suitable for reliable predictions.

$0.6<R^2<0.9$: The model is suitable for qualitative predictions.

$0.9<R^2<1.0$: The model is very good and therefore suitable for quantitative predictions.

Other improvement that can be applied to the model is a transformation of the output variables. In this case, functions as square root, inversion and logarithm can be used. To determine the optimal transformation, all transformations are applied to the output and the one resulting in the lowest RMSE is selected.

The Model can be improvement through recognition and deletion of outliers too. Outlier is a measuring point whose model error (i.e. the deviation of the measured value from the model prediction) is high.

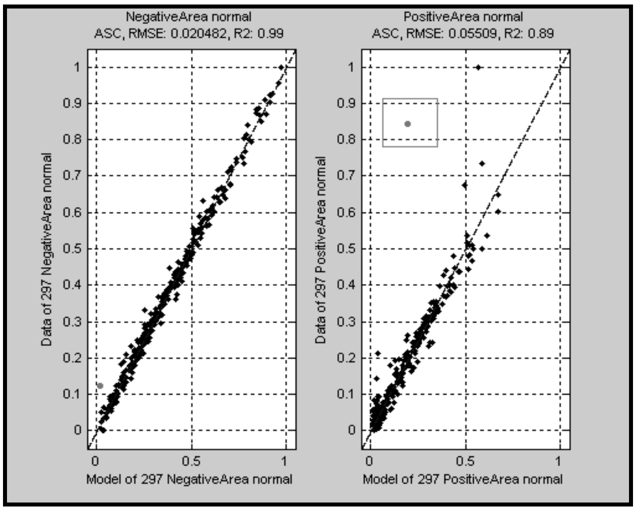


Figure 3.7-6 Outliers in the "True Prediction Plot" display

In Figure 3.7-6 it’s possible see how to define an outlier just looking to the true prediction plot. The measurement inside the square can be considered an outlier. The term can be also interpreted quantitatively: an outlier exists if the residual is bigger that 3-4 RMSE.

This kind of deviation can be caused by an error in the measurement or the measurement took place in the limit range of the engine, and hence, could not be mapped by the model.

It can easily be seen that such measuring points have a negative effect on the model. While the green graph in Figure 3.7-7 results from the modeling based on the blue points, the model training including the red outlier results in a graph (red) that features significant deviations to the measured data.

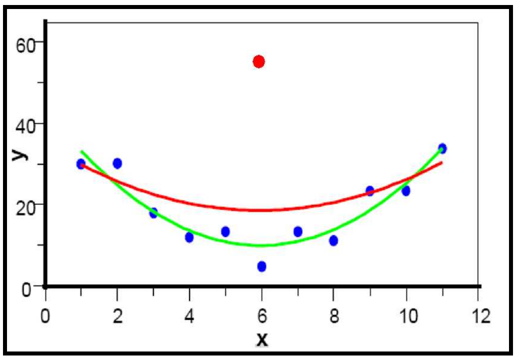


Figure 3.7-7 Modeling with (red) and without (green) outlier

Finally, after all preparation of the data, the optimization can be done.

The optimization depends of the criterion chose. A criterion here is a standard characteristic of the system that can be measured and used to compare different TipIn’s or

TipOut's. The input of ASCMO is an excel table where each row has one TipIn with the values of the parameters used and the value of the criterion (criteria). It is necessary to choose the goal of the optimization, which means, how should be the criterion result of the optimum behavior. For example, if the system to be optimized has as a criterion the overshoot, probably the goal will be to find a combination of parameters that gives a behavior with an overshoot of zero.

Some options to optimize criteria are: min/max (it will look for parameters that give the minimum or maximum value) and target (a specific values can be choose).

There are two optimization methods: single and multiple criteria.

The single criterion is the optimization of a variable or weighted total of several variables according to a gradient descent. The result is a set of setting parameters (input values) for the desired optimization target.

The Multiple Criteria is a true multiple-target optimization that leads to a set of Pareto-optimal solutions. At that point, the selection of the solution can also be performed by means of other criteria (e.g. the values of other inputs or outputs).

A Pareto-optimal (Figure 3.7-8) is a graphic with a set of solutions that can optimize the results, but the target function can be achieved only through the deterioration of another one. That because sometimes mainly independent objectives are defined for an optimization task, but it is frequently not possible to achieve all optimization targets at the same time, which means a trade-off between the criteria.

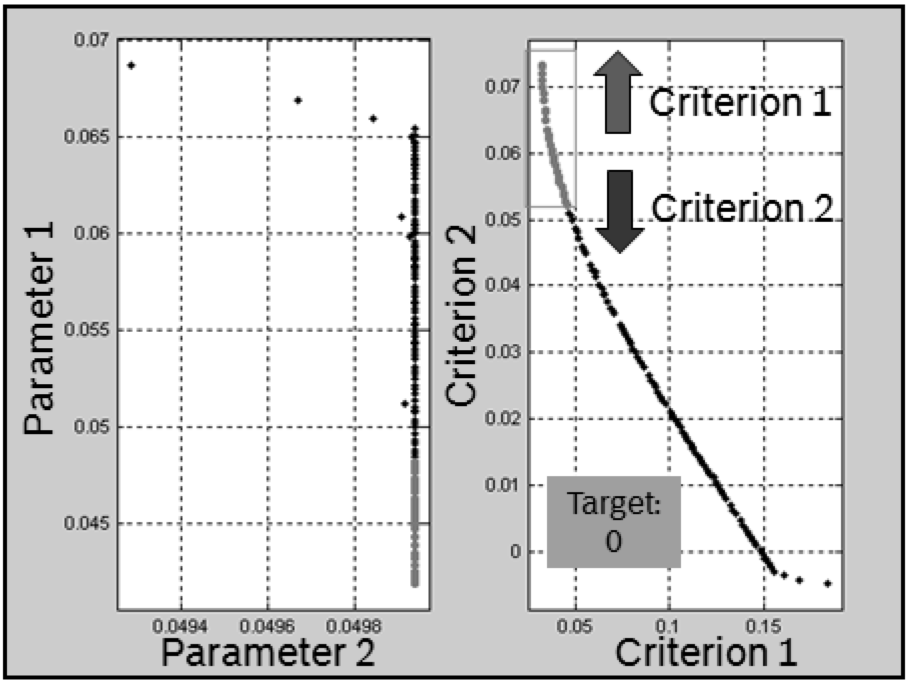


Figure 3.7-8 Pareto-optimal

3.8 TOOLS

During the development of this project, some tools were used as interface between ECU and computer and to simplify the measurement process and the evaluation of the data. Next pages will give a brief explanation of each one of the tools.

3.8.1 Inca 6.2

Inca is a tool developed by ETAS group. It is used for ECU development and test as well as for validation and calibration of electronically controlled systems in the vehicle, on the test plan, or in a virtual environment on the PC.

It is possible to visualize in the main window of Inca (Figure 3.8-1) the database structure and access all items stored in the data base.

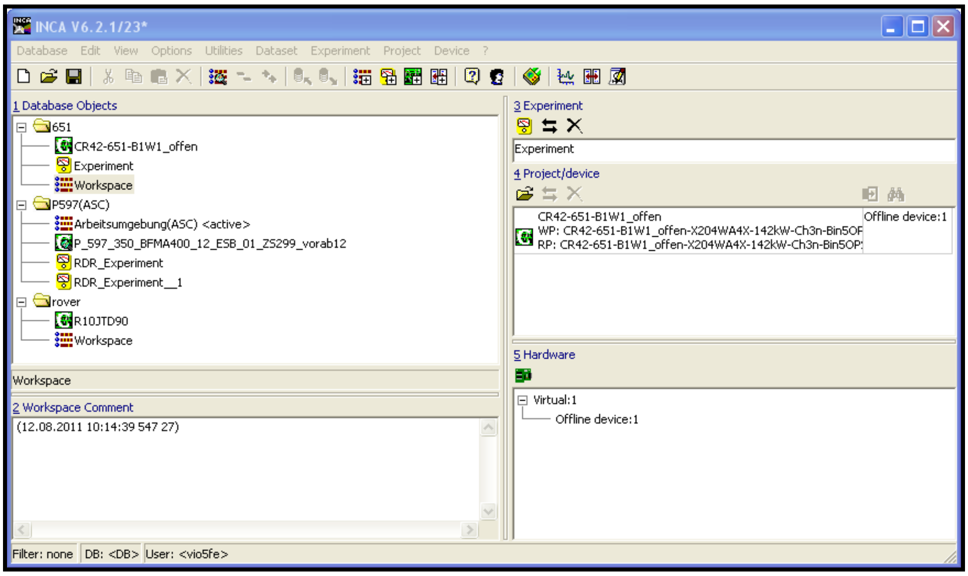


Figure 3.8-1 Inca main window

To initialize a project is necessary to create a workspace. In this workspace is added an experiment, a project with a dataset and the hardware is configured. The Workspace combines all of the elements needed for a calibration or measuring session.

The Hardware Configuration Editor comprises a software-based reproduction of the hardware that will be used for a specific measuring and calibration task.

It is possible see in Figure 3.8-2 the window to the hardware configuration editor. In the left side a list of all devices connected will appear. A tree structure visualizes the interconnections between various modules, devices, and protocols.

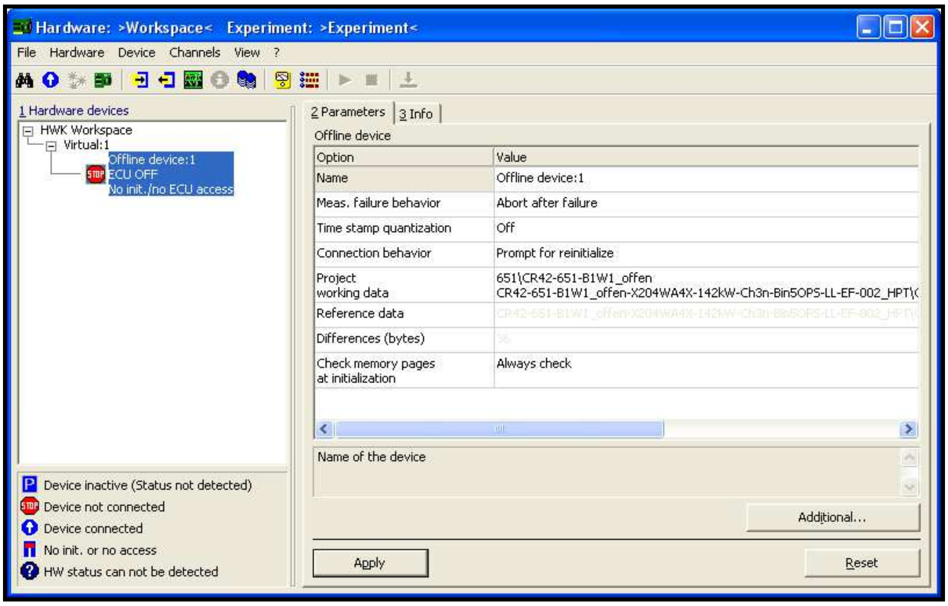


Figure 3.8-2 Hardware Configuration Editor

The Experiment Environment provides the user interface for measuring and calibration activities. The Experiment Environment user interface can be suitably configured to accommodate individual tasks. It facilitates the simultaneous handling of measuring and calibration tasks, and also supports concurrent access to multiple devices.

The following display elements are provided: oscilloscope, numeric display, numeric editor, graphical editor and calibration scenario editor. Figure 3.8-4 shows the experiment used to this project. All variables that it is necessary to measure are added to the experiment. It is possible visualize an oscilloscope with the important variables and register everything in a ‘dat.’ file.

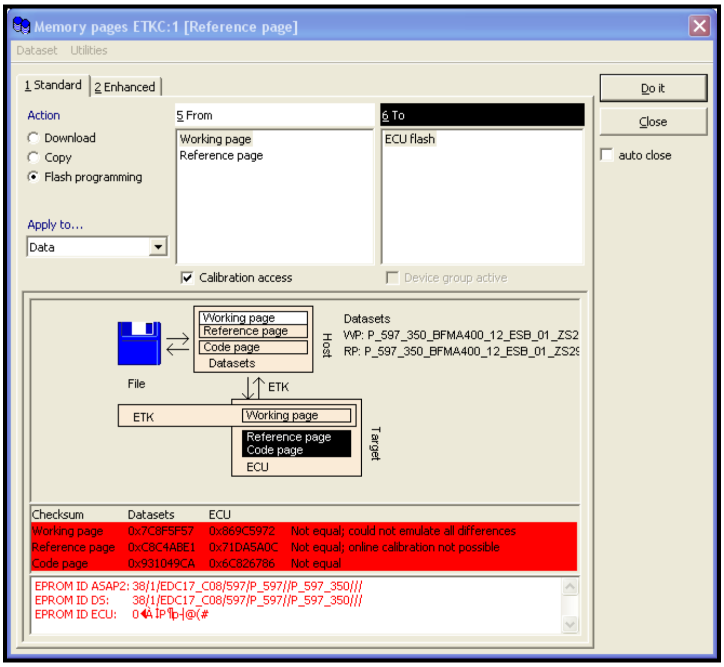


Figure 3.8-3 Memory Page Management Dialog

Calibration data and program code are stored onboard the ECU and in INCA as memory pages. The data transfer between these pages, e.g. from PC to ECU, is handled by the Memory Page Management Dialog (Figure 3.8-3).

The pages are represented by a small graphic, and the following actions may be chosen: download to ECU, upload from ECU, copy from working page to reference page and programming flash memory onboard ECU or in ETK memory emulator.

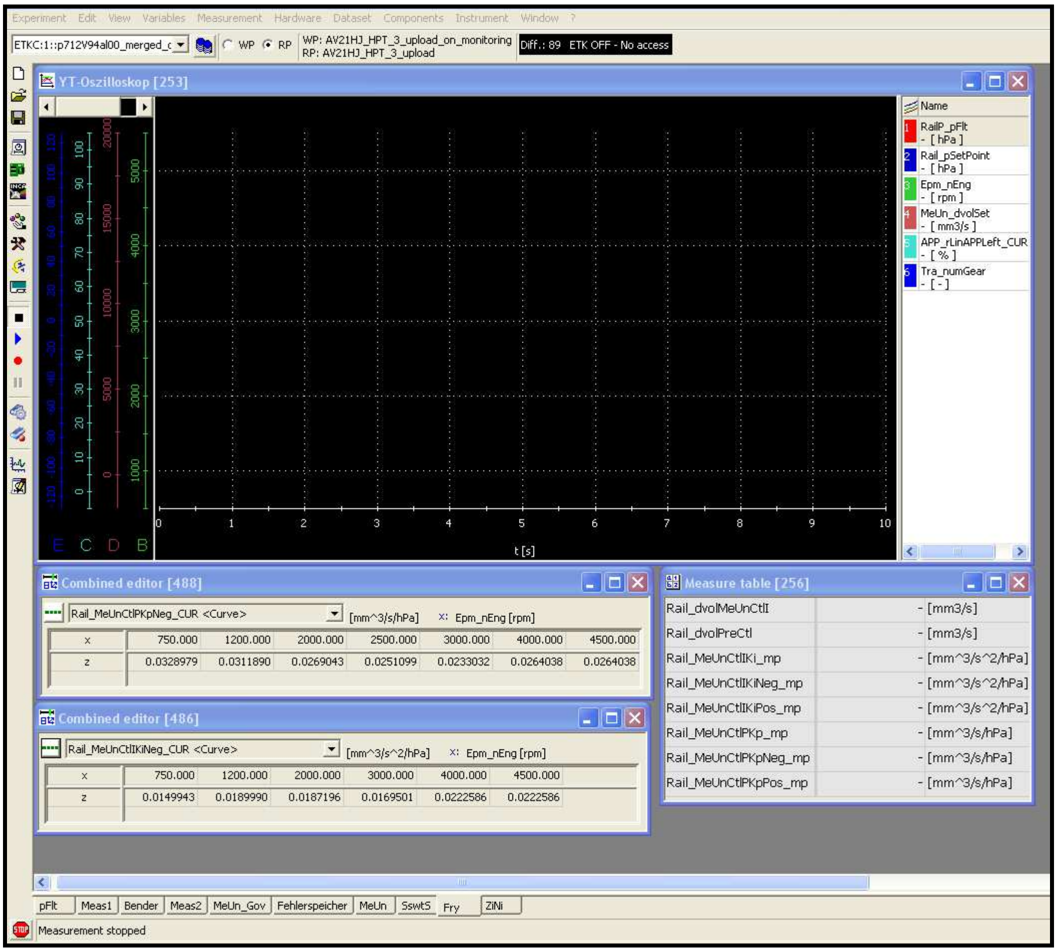


Figure 3.8-4 Inca Experiment

3.8.2 Edor

The program EDOR (Experimental Design for Operating Range) is a tool for creating space-filling experiment plans according to DoE. It is developed to generate global experimental designs for an operating range of a gasoline or diesel engine. That means it will design a space filling design across multiple operating points. It is possible use the tool to a local approach, where an individual experimental design is made for each operation point.

The program can be used via a graphical user interface or from the command line. The process of generating a design is split into five steps:

Step 1: choosing number of factors, names.

Step 2: designing operating points.

Step 3: creating space filling design, rounded to operating points.

Step 4: scaling design to physical units.

Step 5: writing the design to an xls file.

3.8.3 Bender

Bender is a tool developed in MatLab by BOSCH Group whose goal is to connect to INCA to evaluate the Test Plan with the vehicle changing the parameters in the ECU to changing the parameters of the rail pressure governor (in the case of this project, k_P and k_I from the large signal).

The idea of the tool is: the user informs what channel should be the stimulus and what parameters should be calibrated and Bender will adjust Inca configuration and calibrate the values using the test plan that it will give as input to Bender.

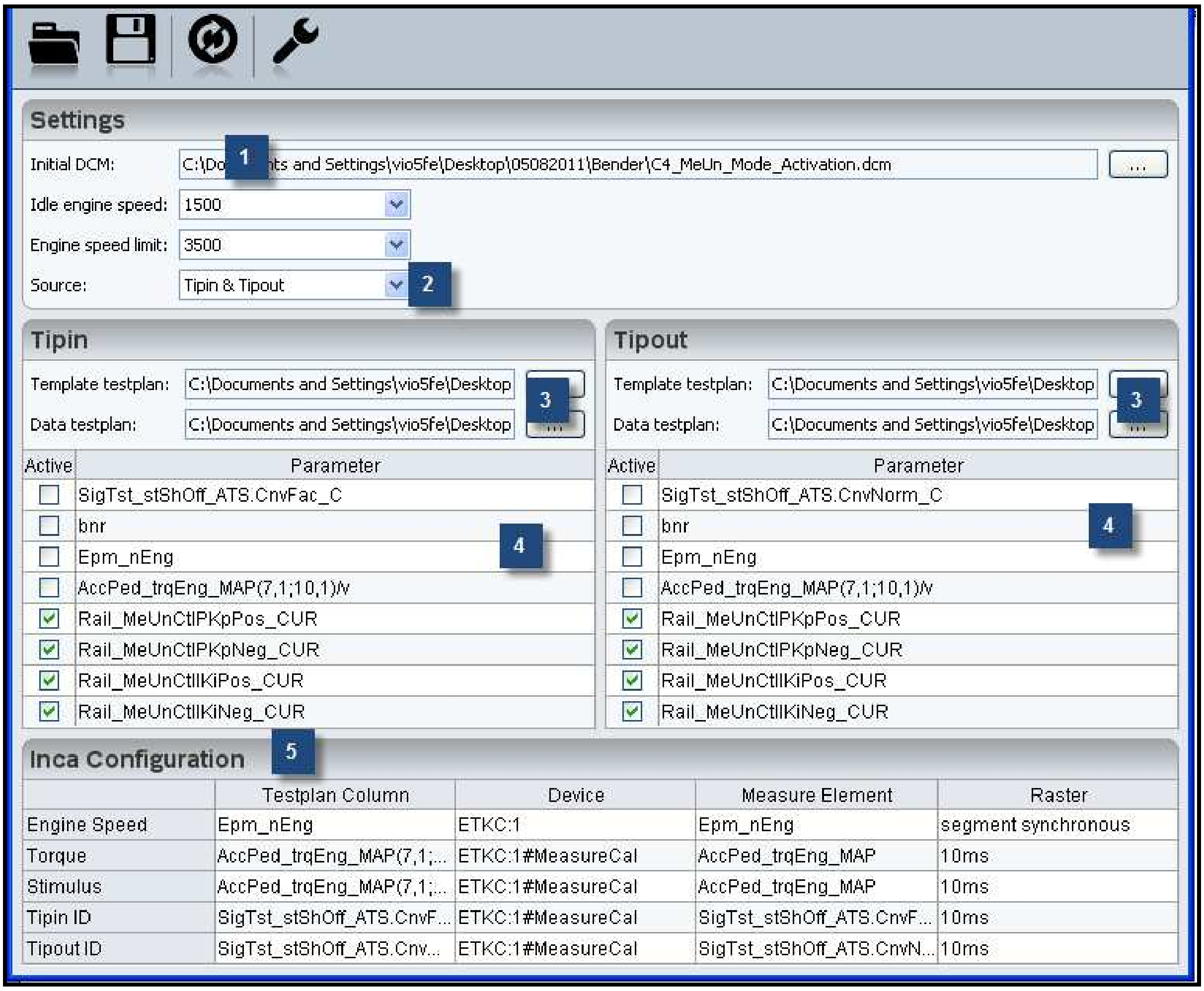


Figure 3.8-5 Bender package editor

Figure 3.8-5 shows the window used to edit a Bender package. First step is adding the initial DCM file with all calibration that need to be initialized at Inca. Second is defining the limits of the engine speed that Bender should work. In the next step (number 3), the test plan (output of Edor) should be add and (number 4) the parameters that should be calibrated need to be checked. The last step (number 5) is define Inca configuration and the relation between all test plan columns and the measured element.

After the creation of a Bender package the measurement can start. Figure 3.8-6 shows the main screen of Bender. To start the measurement is necessary connect Inca and press play.

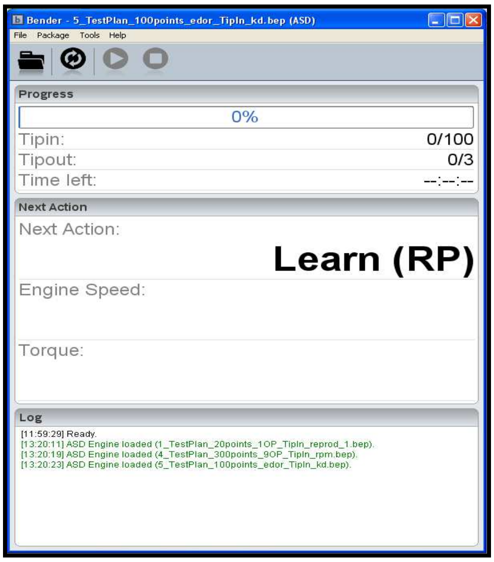


Figure 3.8-6 Bender GUI

After the measurement start, it is necessary executing the follow steps:

1) Moving to min. Engine Speed.

2) Stimulating with maximum Stimulus.

3) Waiting until maximum Engine Speed.

4) Stimulating with 0.

5) Waiting until minimum Engine Speed.

6) Learning Latency.

7) Executing Strategy.

The data will be saved in a '.dat' file that can be read for Bender 4 or for Measure Data Analyzer (MDA).

3.8.4 Bender 4

Bender 4 is a tool developed inside Bender in MatLab by BOSCH Group to evaluate criteria in the measurement signal. As input, Bender uses a .dat file. It is necessary to define the project's properties, the measurement allocation, select criteria that want to be calculated and press the 'Evaluate' button to start the evaluation.

The important point of using Bender 4 is developing the correct criterion for the project. There is a template that should be used in Matlab to programming the criteria. It is necessary to implement all interfaces for a successful integration into Bender 4. In the appendix there is a Matlab code that was implemented to calculate the overshoot and undershoot criterion. Figure 3.8-7 shows how the data is evaluated at Bender 4 and the black point is the output of the Overshoot criterion.

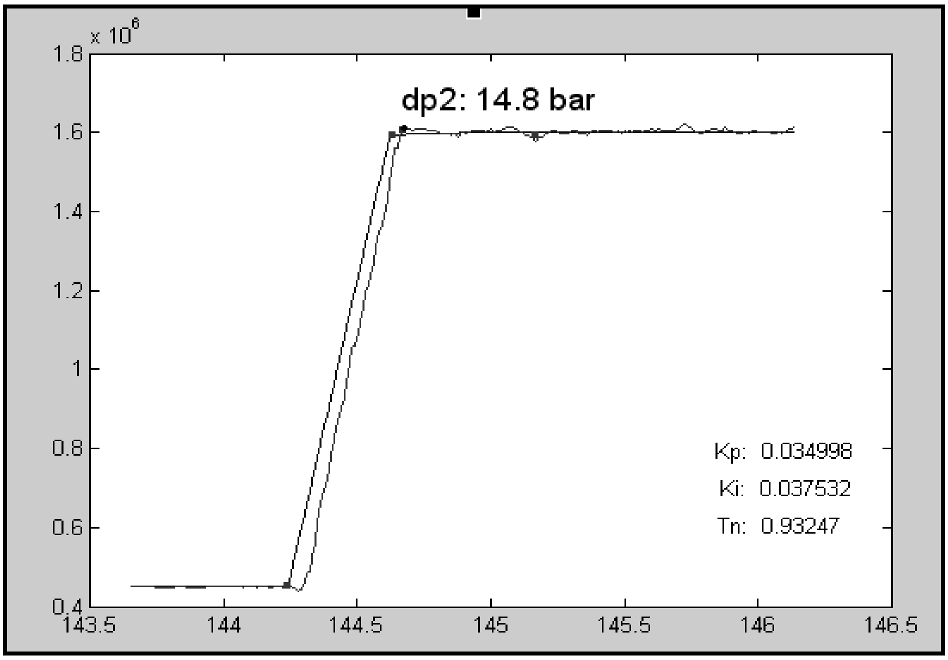


Figure 3.8-7 Bender 4's output

3.8.5 Fry

Fry is a tool developed in MatLab used to change parameters in the ECU using sliders. The user has to give as input the excel table with a Pareto front and an 'a2l' file, that is a memory-description file with all parameters name of the software that is used in the ECU that will be changed. With these sliders, is possible change the parameters online (modify the parameters "on the fly" directly in the ECU memory) according to the desired criteria. The Pareto front gives as input here, is the output generated by ASCMO.

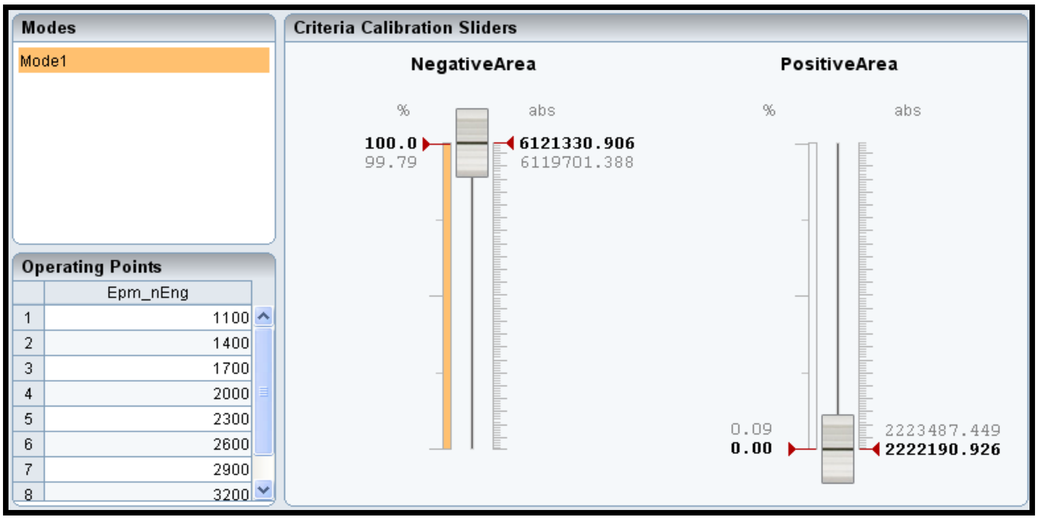


Figure 3.8-8 Fry

3.9 CONSIDERATIONS

This chapter showed the basic concepts to figure out how this project was developed. Now it will be easier to comprehend where the whole project is inside and how the complete system works. To develop this project was necessary a deeply knowledge about all the tools, control engineering and how the common rail system works. At this moment, you are prepared to understand step by step how the method was completed.

4 DEVELOPMENT

The first study of this project was developed in a Citroen C4, platform demonstrator, connected using an Emulator Test Probe. The ECU is an EDC17, the common rail system is two actuators (CPC system), the injectors are from the second generation (CRS2.2); and it is using a CP4.1. For this work, just the MeUn actuator will be calibrated. For that, the PCV will be closed, so the car will have just one actuator active.

The second part of this project will consist in validate this project in other C4 and another car (Ford Fiesta).

Nowadays the common rail calibration with a MeUn controller system is divided in 12 steps:

1. Rough calibration.
2. Rough calibration of current controller MeUn.
3. Calibration of current controller MeUn.
4. Calibration of rail pressure sensor signal.
5. Calibration of characteristic curve of MeUn.
6. First calibration of closed loop control for MeUn.
7. Measurements necessary for pre controller for MeUn.
8. Fine tuning MeUn control.
9. Validation MeUn control.
10. Calibration rail pressure at engine start.
11. Final validation of complete rail pressure control.
12. Troubleshooting.

All the steps will not be described here; however it is important to know where this project is inside in all process calibration. To understand that, just steps 6, 8 and 9 need to be clear.

Step 6 is the determination of k_{Pcrit} and f_{crit} with ZiNi method (described in chapter 3.5). With these values is possible find k_p, k_i and k_d curve. This step will not change if the new method proposed in this work applies, but it will be used as start values.

Step 8 is not completely clear how to proceed. The goal is adapting $k_{ipos/neg}$ and $k_{ppos/neg}$ (large window switchover) to improve the step response with nominal systems. There are some tips on how to find these values, but the engineer basically needs to drive the car and try to find the best parameter observing the step response. And step 9 is the validation of the last step. If something is wrong, step 8 needs to restart.

This project has the proposal of creating a method to calibrate the parameters $k_{ipos/neg}$ and $k_{pppos/neg}$. That means it will be a method to substitute step 8 and 9. As defined in the goal of the project, the idea is that the method will reduce time, bring economy and quality, since will not be necessary an engineer in the complete process and will not be a recursive method, which means, will not require to repeat the complete procedure if some calibrated point was not optimum.

Figure 3.9-1 describes some basic labels from EDC17 used in this project.

Label	Description	Unit
APP_r	Standardized accelerator pedal angle in percentage	%
Epm_nEng	Engine speed	Rpm
AccPed_trqEng_MAP	This is the Pedal Map 1. Size is 9 x 9. It depends of APP_r and Epm_nEng	Nm
InjCtl_qSetUnBal	Current injection quantity	mg/hub
Rail_pSetPoint	Rail pressure set point	hPa
RailP_pFlt	Maximum rail pressure during last 10ms.	hPa
Rail_pDvt	Governor deviation of the rail pressure governor	hPa
Rail_MeUnCtlPKp_CUR	P amplification for small signals for pressure control by metering unit	$mm^3/s/hPa$
Rail_MeUnCtlIKi_CUR	I amplification for small signals for pressure control by metering unit	$mm^3/s/hPa$
Rail_MeUnCtlDKd_CUR	D amplification for pressure control by metering unit	$mm^3/s/hPa$
Rail_MeUnCtlPKpPos_CUR	P amplification for positive large signals for pressure control by metering unit	$mm^3/s/hPa$
Rail_MeUnCtlPKpNeg_CUR	P amplification for negative large signals for pressure control by metering unit	$mm^3/s/hPa$
Rail_MeUnCtlIKiPos_CUR	I amplification for positive large signals for pressure control by metering unit	$mm^3/s/hPa$
Rail_MeUnCtlIKiNeg_CUR	I amplification for negative large signals for pressure control by metering unit	$mm^3/s/hPa$
Rail_pMeUnCtlIWinNeg_C	Lower small signal limit for I component of pressure control by metering unit	hPa
Rail_pMeUnCtlIWinPos_C	Upper small signal limit for I component of pressure control by metering unit	hPa
Rail_pMeUnCtlPWinNeg_C	Lower small signal limit for P component of pressure control by metering unit	hPa

Rail_pMeUnCtlPWinPos_C	Upper small signal limit for P component of pressure control by metering unit	hPa
MeUn_dvolSet	Setpoint value (volume flow) of rail pressure governing	mm^3/s
Rail_dvolMeUnCtlI	I physical component of pressure governor by metering unit	mm^3/s
Rail_dvolMeUnCtlP	P physical component of pressure governor by metering unit	mm^3/s
Rail_dvolMeUnCtlDT1	DT1 physical component of pressure governor by metering unit	mm^3/s
Rail_dvolPreCtl	Pre-control physical value for pressure governing by metering unit	mm^3/s

Figure 3.9-1 Basic labels

In this table is possible to see that some labels have in the end of the name C, CUR or MAP. When the name finish with “C”, means that is a constant. When the label has the name “CUR” in the end, means that this label corresponds to a vector. For example, Rail_MeUnCtlIKiNeg_CUR is changing according to the engine speed. When the name has “MAP” in the end, means that the label correspond to a matrix. For example, the output of AccPed_trqEng_MAP (Figure 3.9-2) depends of engine speed (x-axis) and APP_r (y-axis).

AccPed_trqEng_MAP <Map> [Nm] x: Epm_nEng [rpm] y: APP_r [%]

y \ x	400.00	650.00	750.00	1000.00	1250.00	1500.00	1750.00	2500.00	2750.00
1.0986	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000
15.0024	127.500	118.500	114.400	82.000	44.800	32.700	30.100	27.500	27.500
27.0020	218.200	216.300	214.800	186.300	119.000	98.100	89.500	67.300	57.800
34.7778	238.100	235.900	235.400	204.600	173.800	155.900	126.000	90.100	96.700
39.0015	249.400	246.700	245.500	221.900	198.300	182.700	161.600	145.500	139.500
47.9980	268.600	265.900	262.300	256.400	250.400	239.900	237.300	211.200	209.200
57.9956	287.300	287.400	284.000	284.000	283.700	280.700	279.600	264.300	253.800
69.9951	324.300	324.300	324.300	324.300	324.300	324.300	324.300	314.300	302.300
80.0049	324.300	324.300	324.300	324.300	324.300	324.300	324.300	314.300	302.300
100.0000	324.300	324.300	324.300	324.300	324.300	324.300	324.300	314.300	302.500

Figure 3.9-2 MAP

The MeUn_dvolSet, as explained in Figure 3.9-1, is the setpoint volume flow value of the rail pressure governor, which means, it is the physical output of the PID controller and the pre-controller of MeUn. This value is compound for the output of the P-component (Rail_dvolMeUnCtlP), I-component (Rail_dvolMeUnCtlI), D-component (Rail_dvolMeUnCtlDT1) and pre-controller (Rail_dvolPreCtl). Figure 3.9-3 is an example measurement that shows how it is the general behavior of all this components in a step response. In this example the D-component is zero.

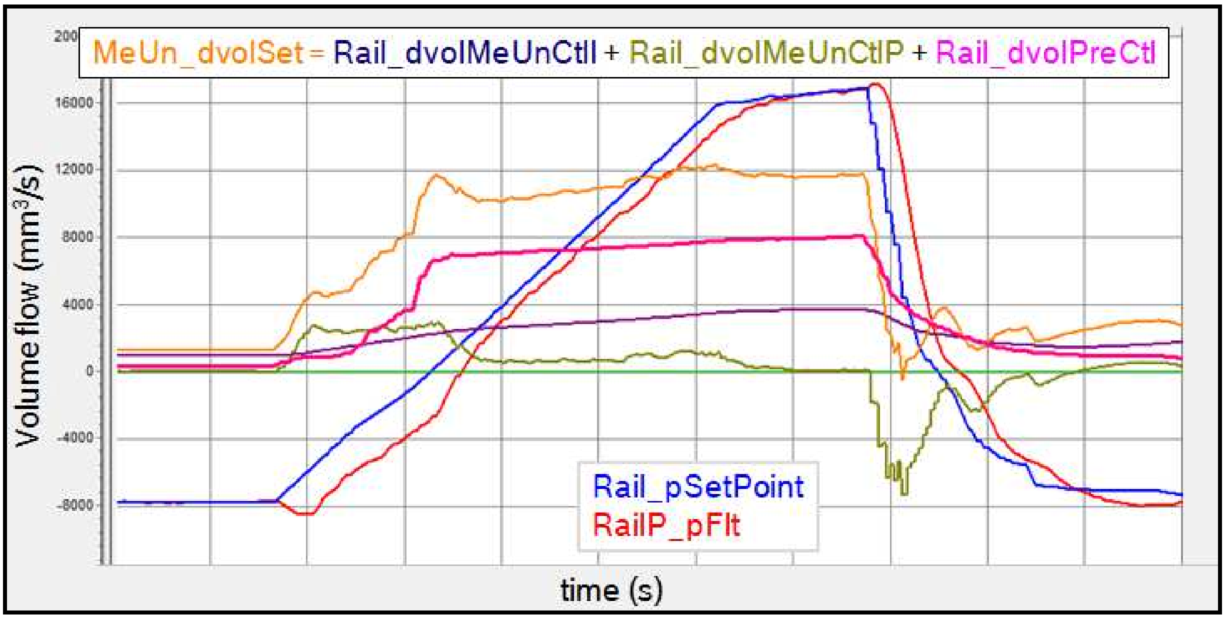


Figure 3.9-3 MeUn_dvolSet

Before start the development of the method, it is necessary to be sure that the system is reproducible. For that, the same point was measured twenty times and the behavior of each step was observed. Figure 3.9-4 shows the behavior in full load of four points. It is possible to observe some variation from one point to another.

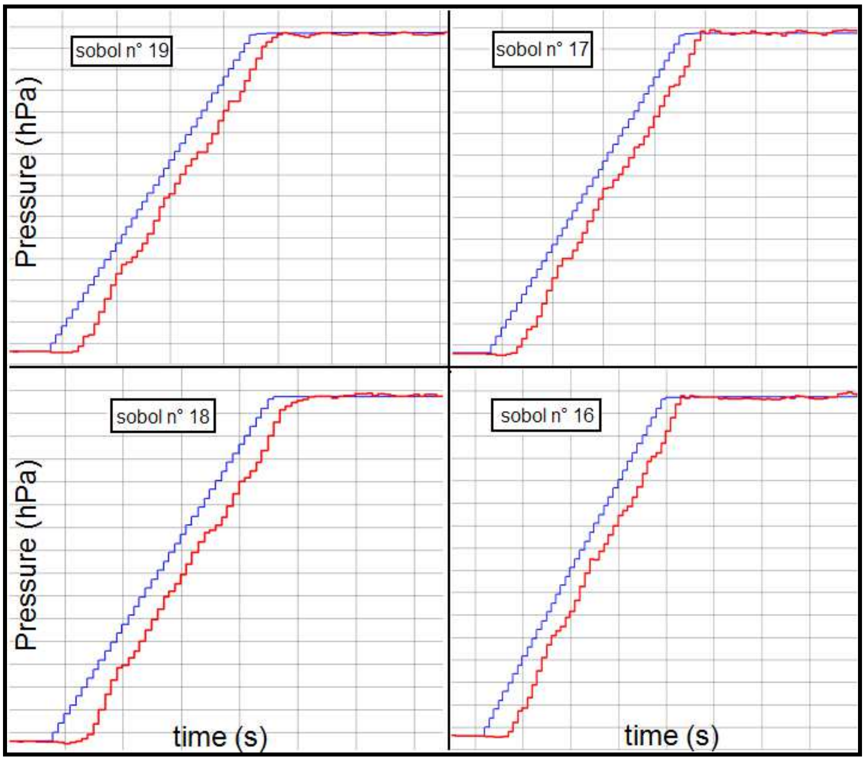


Figure 3.9-4 Reproducibility

However, this variance is completely acceptable. This can be explained for the expected spread of the system. As it is possible see in Table 3.9-1, the standard deviation of the delay is nearly 1ms and of the overshoot is nearly 6 bars.

Table 3.9-1 Reproducibility

Sobol n°	Engine speed (rpm)	Delay (ms)	Overshoot (bar)
12	2491,5	0,052	12,8
13	2502	0,053	9,3
14	2494,5	0,053	7,9
15	2484,5	0,052	21,5
16	2482	0,055	6,2
17	2493,5	0,054	8,3
18	2470	0,052	1,2
19	2491	0,051	1,4
Average	2488,63	0,053	8,575
Standard Deviation	9,7	0,001	6,52

Through these measures, it is possible conclude that the system is reproducible, that means the method can be used.

The validation of this method is developed in a nominal system, but physical variations of the MeUn unit are expected (the drilling where the pressure is going inside can change) and when the system has this variation, it is called minimum or maximum system. Because of that, the pressure inside the common rail will behave different from one vehicle to another. That means, if the response has no overshoot in nominal system, probably in a max system a small overshoot can be observed. And overshoots can damage the system, reducing the life time (material stress). The method here developed needs to be robust enough to work in 'min', nominal and 'max' systems in the same time. For that reason, the most important characteristic of a good calibration is find parameters that leads to a minimum overshoot (and reacting as fast as possible). Figure 3.9-5 shows an example of a good calibration.

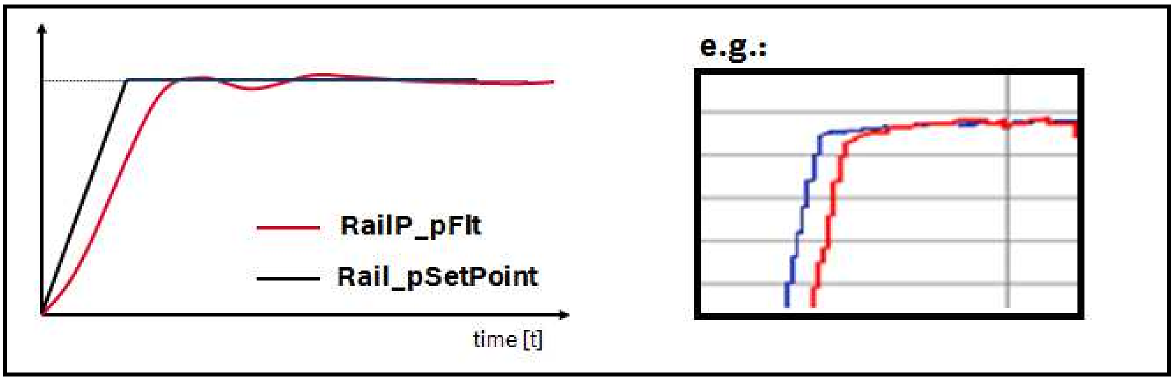


Figure 3.9-5 Example of a good calibration

In the following chapters the development of this project will be describes step by step.

4.1 DESIGN OF EXPERIMENT TEST PLAN

The first step to start the development is creating a Test Plan. For that, it is necessary first of all to decide the borders of the parameters that need to be calibrated. As explained before, this project has the proposal of calibrating k_{ppos}, k_{pneg}, k_{ipos} and k_{ineg}. The standard value of the small window limits are used, that means within -40 and 40 bars, the small window is active, outside large window is active.

To define the borders is necessary to know how the behavior of the system is. What value of k_p would lead to an unstable system. For that, the ZiNi method number two (described in chapter 3.5 - PID Controller) will be applied, but just the concept of how to find k_{Pcrit} and f_{crit}.

To apply the ZiNi method, first it is necessary to change to MeUn mode (since C4 is a two actuators system). Besides that, to observe the behavior of k_p in the complete signal, the large signal window needs to be expanded, I and D components need to be set to zero; and engine speed needs to be set to the desired one.

After that, the measurement starts. The idea is increasing k_p until the instability is reached, keep it during three seconds in instability and come back to stability. This procedure needs to be repeated at least three times. Figure 4.1-1 shows how the measurement looks like. With this measurement is possible to define k_{Pcrit} and f_{crit}. The k_{Pcrit}, as is possible see in Figure 4.1-1, is the k_p found when the system has a significant reduction of the oscillation. To find f_{crit} it is necessary to go to the measurement file and find the frequency that appears when the system is oscillating. The normal procedure is to take ten periods and divide per ten, so, the inverse of this value (T_{crit}) is the f_{crit}.

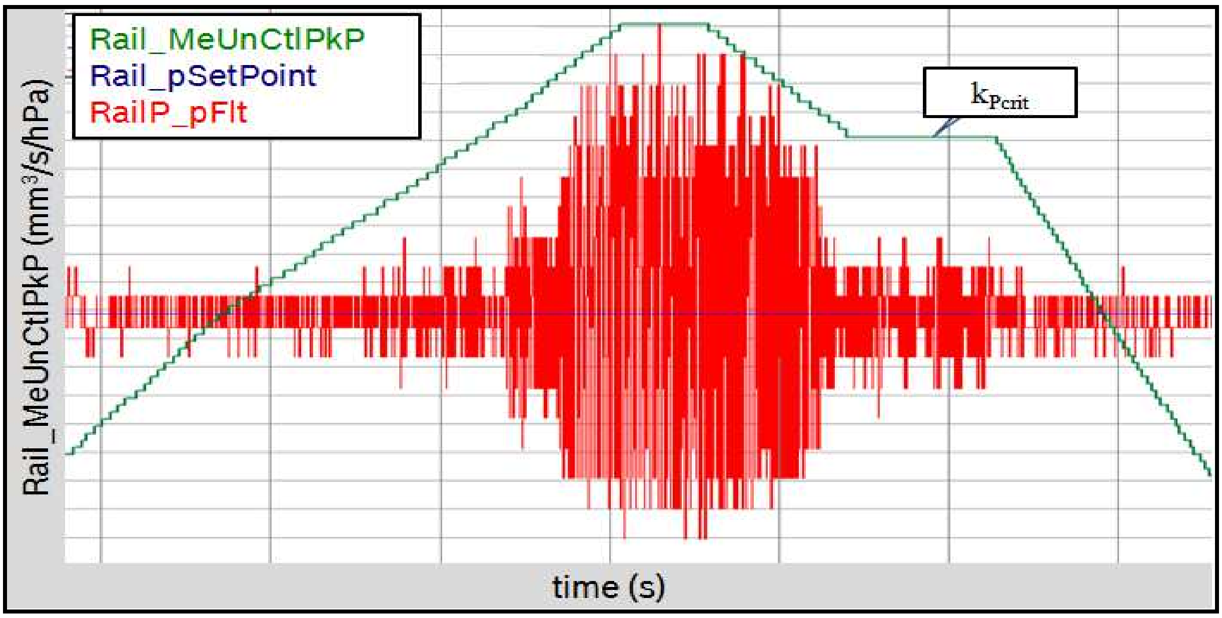


Figure 4.1-1 ZiNi method

This procedure will repeat for more than one engine speed, so it is possible to define how k_{Pcrit} and f_{crit} vary in all range of engine speeds. The results are described in the table below.

Table 4.1-1 P-gain critic for C4

EngSpeed (rpm)	1100	1400	1700	2000	2300	2600	2900	3200	3500
k_{Pcrit} (mm^3/s/hPa)	0,0725	0,0625	0,06	0,06	0,06	0,06	0,06	0,06	0,06
f_{crit} (Hz)	5,86	6,67	7,18	7,69	7,88	8,28	8,71	9,37	9,80

With k_{Pcrit} and f_{crit} is possible to define k_p and k_i to the small window and k_d. They were defined using the gain margin method. Below, the calculation of k_p and k_i for 1100 rpm, as an example. In this project, k_d is zero.

As explained in chapter 3.5 (PID Controller), to use the gain margin method is necessary to know the crossover frequency (ω_{pc}), which can be defined as:

$\omega_{pc} = 2\pi * f_{crit}$

Engine speed of 1100 rpm: $\omega_{pc} = 2\pi * 5{,}86 = 36{,}82$ rad/s

Remember that the crossover frequency is the frequency that gives a shift in the phase of -180º, so, it’s the frequency that makes the system go to instability.

It is necessary to know the magnitude in this frequency $A(\omega_{pc})$ too. Since k_{Pcrit} is the gain margin of the system in the crossover frequency, $A(\omega_{pc})$ can be defined as:

$|A(\omega_{pc})| = 1/k_{Pcrit}$

Engine speed of 1100 rpm: $|A(\omega_{pc})| = 1/0{,}0725 = 13{,}79$

And, finally, it is possible to find the value of k_p which gives the desired gain margin. In this project, the value used is 9.55 dB:

$k_p = 1/(|A(\omega_{pc})| * |G_{M\_OL}|)$

Engine speed of 1100 rpm: $|G_{M\_OL}| = 10\text{^}(9{,}55/20) = 3{,}003$

$k_p = 1/(13{,}79 * 3{,}003) = 0{,}024$

Now, to find the value of k_i, it is necessary to define how far the natural frequency should stay from the crossover frequency. In this project, the natural frequency will be twenty times smaller than the crossover frequency.

$\omega_n = 1/T_N = \omega_{pc}/factor = k_i/k_p$

Engine speed of 1100 rpm: $\omega_n = 1/T_N = 36{,}82/20 = k_i/0{,}024 \rightarrow k_i = 0{,}044$

However, all this procedure takes time and it is necessary to know how the gain margin method works. To make easier for the user, standard values will be used with the same meaning:

Factor	Value
a	0.333
b	0.05

The factor ‘a’ represents the desired gain. Factor ‘b’ represents how far the natural frequency will be from the crossover frequency.

Figure 4.1-2 explains the relation between this factor and the gain margin method. It is exactly the same that was done in the example above with an engine speed of 1100 rpm, but it leads to these two simple equations:

$k_p = a^*.k_{Pcrit}$

$k_i = b^*k_p\, 2\pi^*f_{crit}$

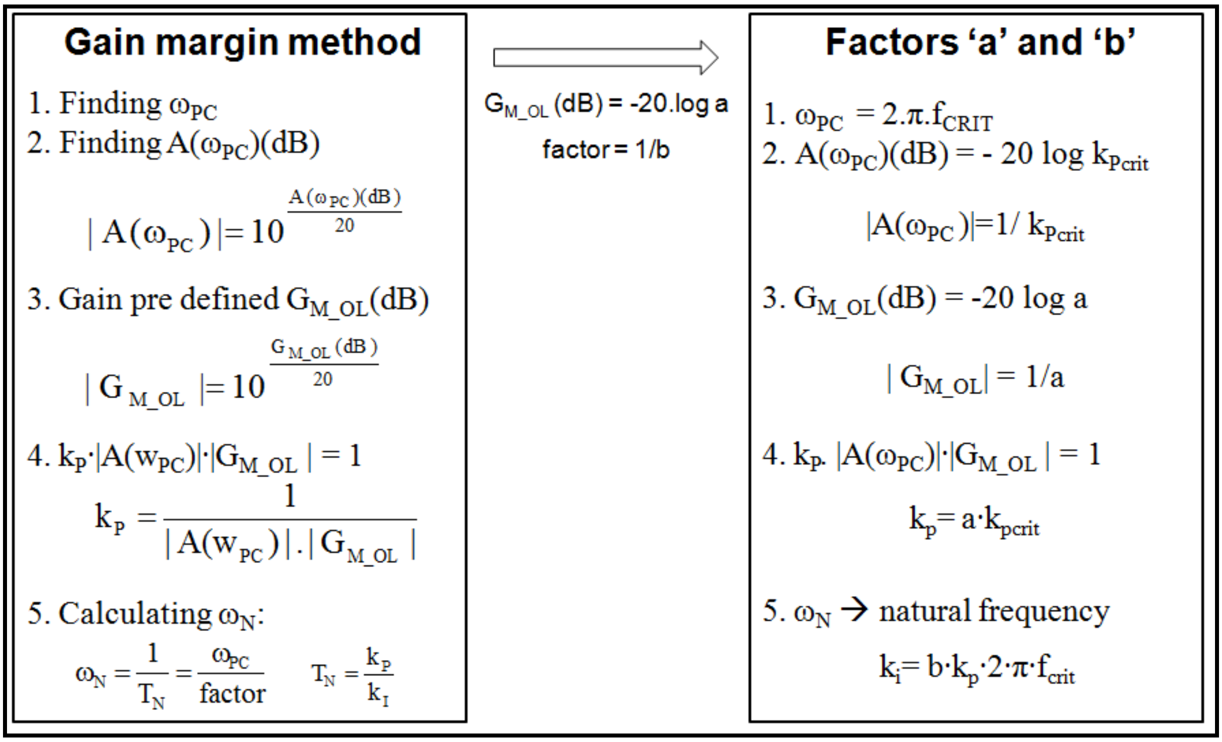


Figure 4.1-2 Standard values definition

Now, there is all information about how to define the small window. After that, it is possible to define the borders of the large signal to create the test plan. The same border is used for negative and positive signal.

For k_p in the large signal, it is necessary a higher value to have a fast behavior. The factors that will be used are described below.

Lower border:

$$k_{plow} = 1{,}5 \cdot a \cdot k_{Pcrit}$$

Upper border:

$$k_{pupp} = 2{,}2 \cdot a \cdot k_{Pcrit}$$

That means in the large signal a gain margin between 2.7 and 6dB is acceptable.

a	G_{M_OL}(dB)
0,333	9,55
0,333*1,5	6,03
0,333*2,2	2,70

The k_i in the large signal will be defined as a function of T_N.

$$T_{Nmax} = \frac{k_P}{k_I} = \frac{k_P}{b \cdot k_P \cdot 2\pi \cdot f_{crit}} = \frac{1}{0{,}3 \cdot b \cdot 2\pi \cdot f_{crit}}$$

$$T_{Nmin} = \frac{k_P}{k_I} = \frac{k_P}{b \cdot k_P \cdot 2\pi \cdot f_{crit}} = \frac{1}{0{,}8 \cdot b \cdot 2\pi \cdot f_{crit}}$$

That means, the ω_n maximum (T_N minimum) will be twenty five times smaller than the crossover frequency and ω_n minimum (T_N maximum) will be almost seventy times smaller than the crossover frequency.

b	factor
0,05	20
0,05*0,3	66,7
0,05*0,8	25

$$k_{iupp} = \frac{k_{plow}}{T_{Nmin}}$$

$$k_{ilow} = \frac{k_{pupp}}{T_{Nmax}}$$

The table below shows the C4 borders. It’s possible to see the upper and lower border of k_i, k_p and T_N and the small signals.

EngSpeed		1100	1400	1700	2000	2300	2600	2900	3200	3500
kpcrit		0,073	0,063	0,06	0,06	0,06	0,06	0,06	0,06	0,06
fcrit		5,86	6,67	7,18	7,69	7,88	8,28	8,71	9,37	9,8
large signal	kpupp	0,053	0,046	0,044	0,044	0,044	0,044	0,044	0,044	0,044
	kplow	0,036	0,031	0,030	0,030	0,030	0,030	0,030	0,030	0,030
	Tnmax	1,811	1,591	1,478	1,380	1,346	1,281	1,218	1,132	1,083
	Tnmin	0,679	0,597	0,554	0,517	0,505	0,481	0,457	0,425	0,406
	Kiupp	0,053	0,052	0,054	0,058	0,059	0,062	0,066	0,071	0,074
	Kilow	0,029	0,029	0,030	0,032	0,033	0,034	0,036	0,039	0,041
small signal	kp	0,024	0,021	0,020	0,020	0,020	0,020	0,020	0,020	0,020
	ki	0,044	0,044	0,045	0,048	0,049	0,052	0,055	0,059	0,062
	Tn	0,543	0,477	0,443	0,414	0,404	0,384	0,365	0,340	0,325

The Test Plan will be created to the worst case, which means the torque will be the maximum (in the case of C4, a step from 0 to 250 Nm). The first study was with just one operation point (engine speed of 2500 rpm and torque of 250 Nm) and 300 points were measured. With this first measurement, the PI-behavior was observed and all the tools (Bender 4 and Fry) were tested. Figure 4.1-3 shows a data plot, where x-axis is k_{ppos} and y-axis is k_{ipos}. The graphic is divided in five areas and each one the behavior was analyzed.

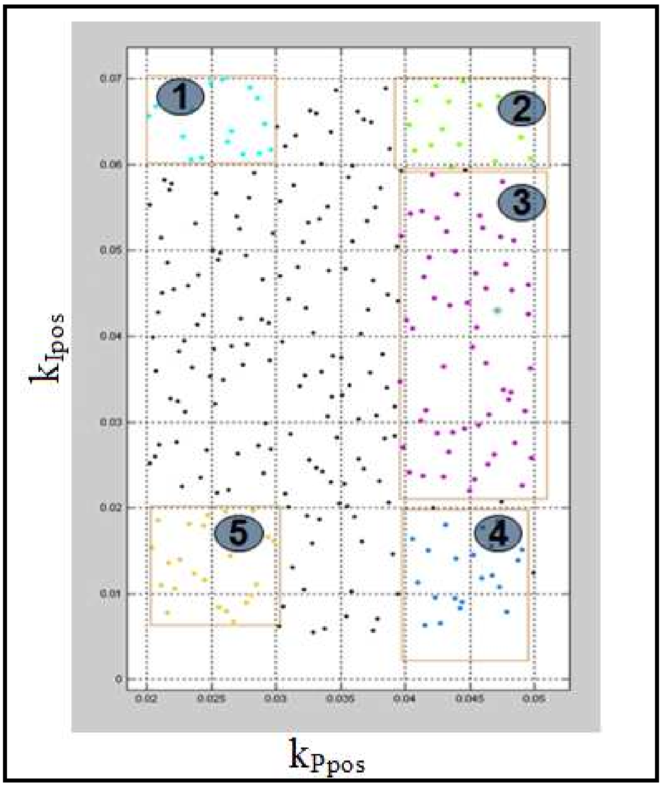


Figure 4.1-3 PI-behavior in the measurements

The measurements in the first area are points with high k_i and small k_p, which means a very small T_N. As studied before (chapter 3.4), smaller k_p means a slower reaction and higher T_N leads to higher overshoots with more settling time. Figure 4.1-4 shows a measured point in this area.

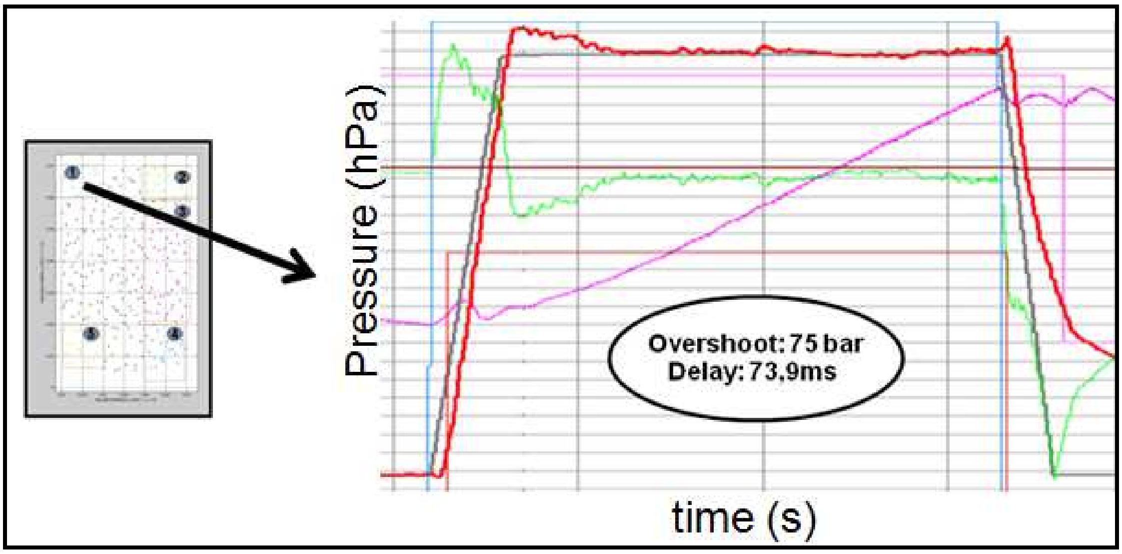


Figure 4.1-4 Measurement 1: PI-behavior

The second area has still high k_i, but now a higher k_p, which means a faster system than the area before and a decrease of the settling time, as it is possible observe in Figure 4.1-5.

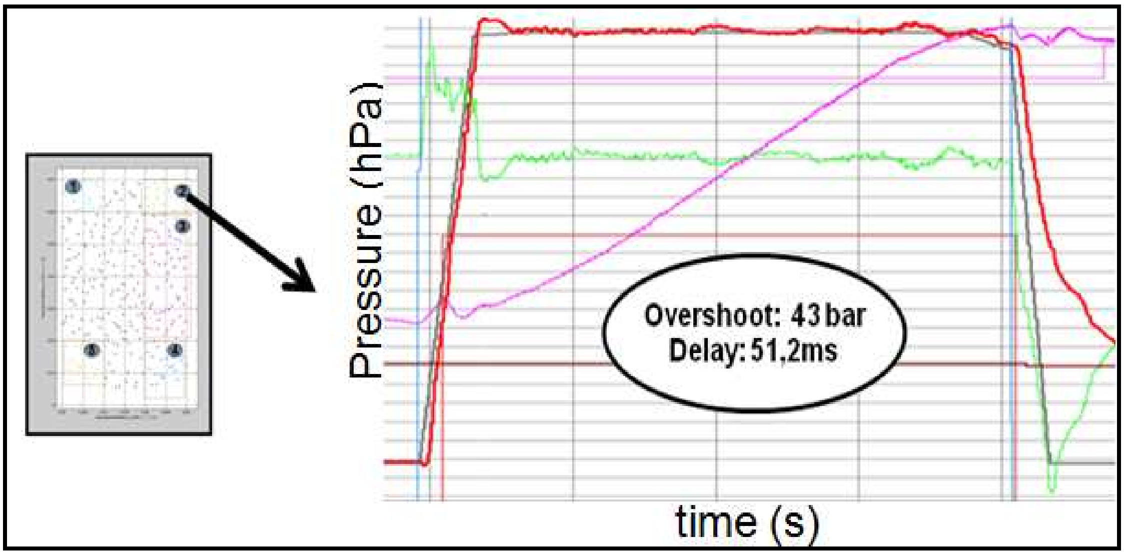


Figure 4.1-5 Measurement 2: PI-behavior

The third area has the same k_p as the second, however a k_i smaller, which means a system a little bit faster than before and with less overshoot. Figure 4.1-6 proves this behavior.

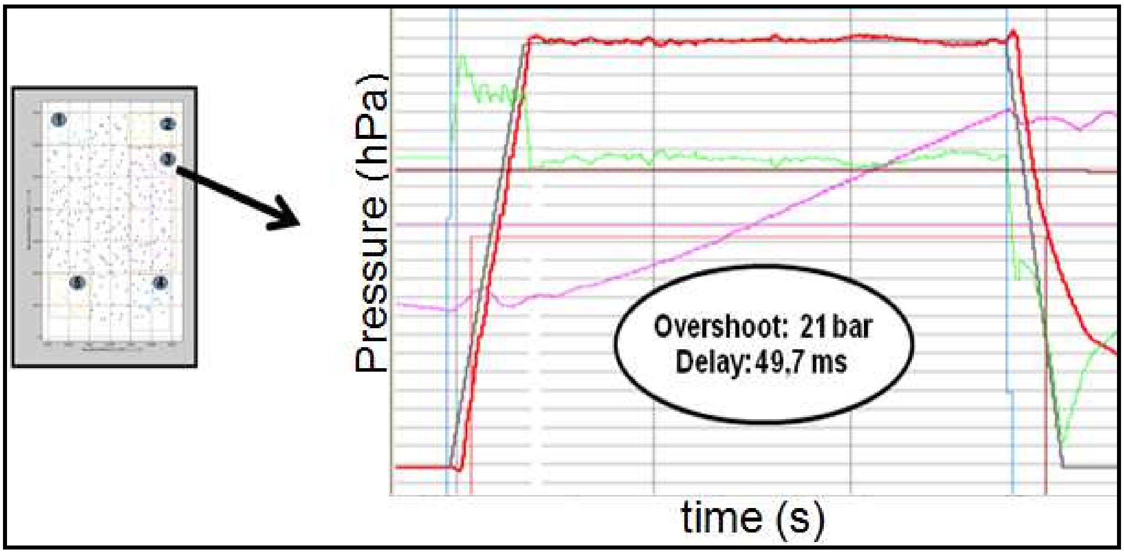


Figure 4.1-6 Measurement 3: PI-behavior

The forth area has even smaller k_i than before, which means that the behavior is still fast, but it takes more time to reach the set point and, since k_p is still high, there is more oscillation. This can be observed in Figure 4.1-7.

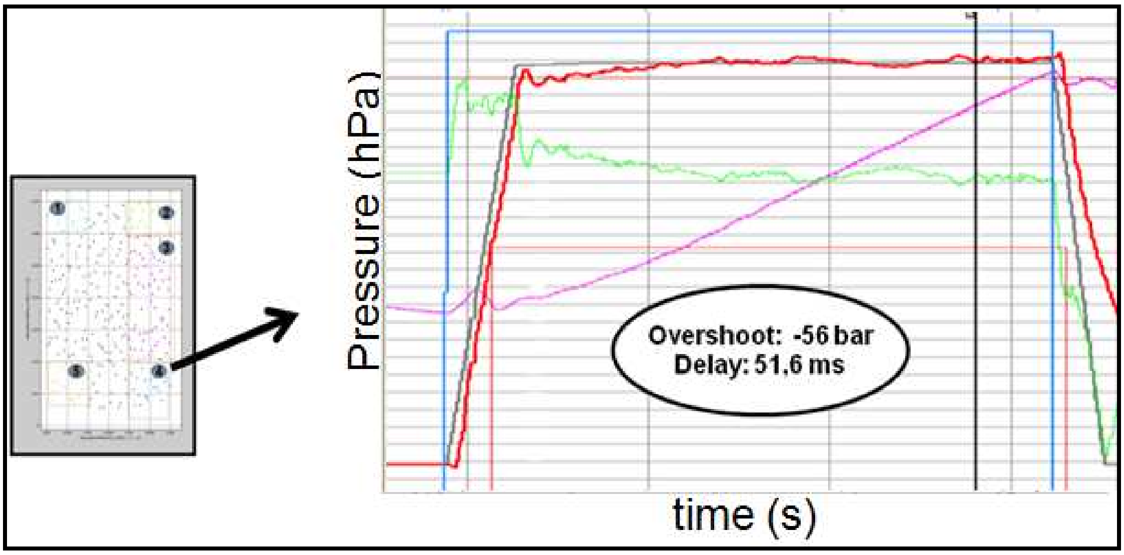


Figure 4.1-7 Measurement 4: PI-behavior

And the fifth and last area has a smaller k_p than the fourth area, which means a very slow behavior. Figure 4.1-8 shows a measurement in that area.

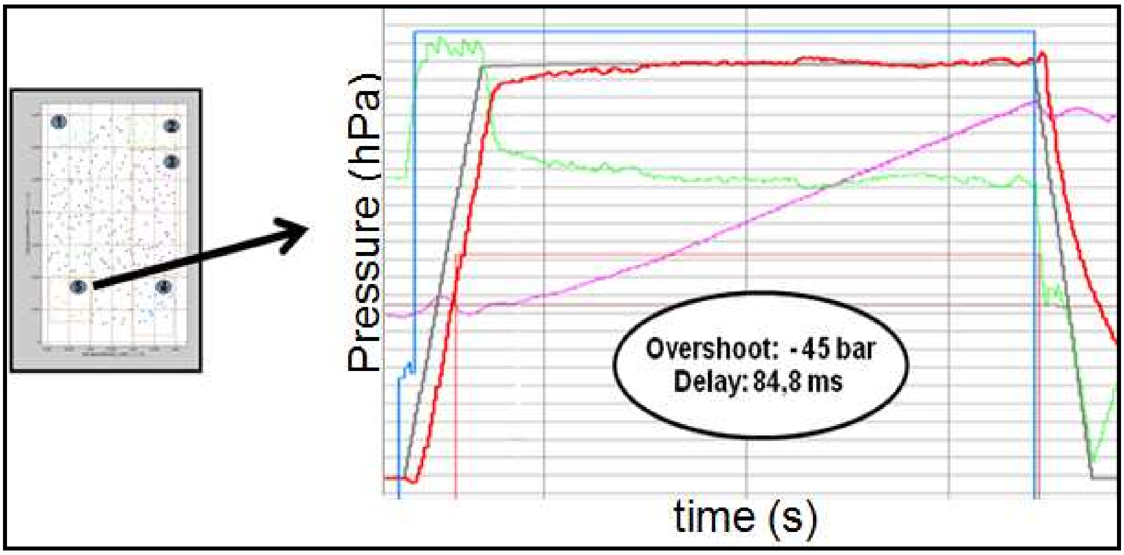


Figure 4.1-8 Measurement 5: PI-behavior

After analyzing the results, a new test plan was created with nine operation points (9 different engine speeds and still one torque) using EDOR (chapter 3.8.2).

As explained in chapter 3.8.2, EDOR tool is divided in five steps. First step (Figure 4.1-9) the input names are defined. In this project, the operation point is defined as engine speed (Epm_nEng) and torque (AccPed_trqEng_MAP). The inputs are the four large signal parameters.

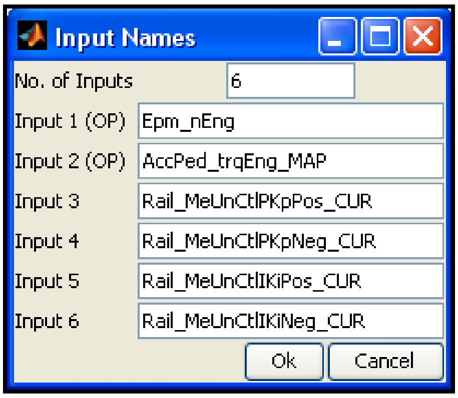


Figure 4.1-9 Edor Step 1

Second step (Figure 4.1-10) is the design of operating points. Nine engine speeds were defined and one torque.

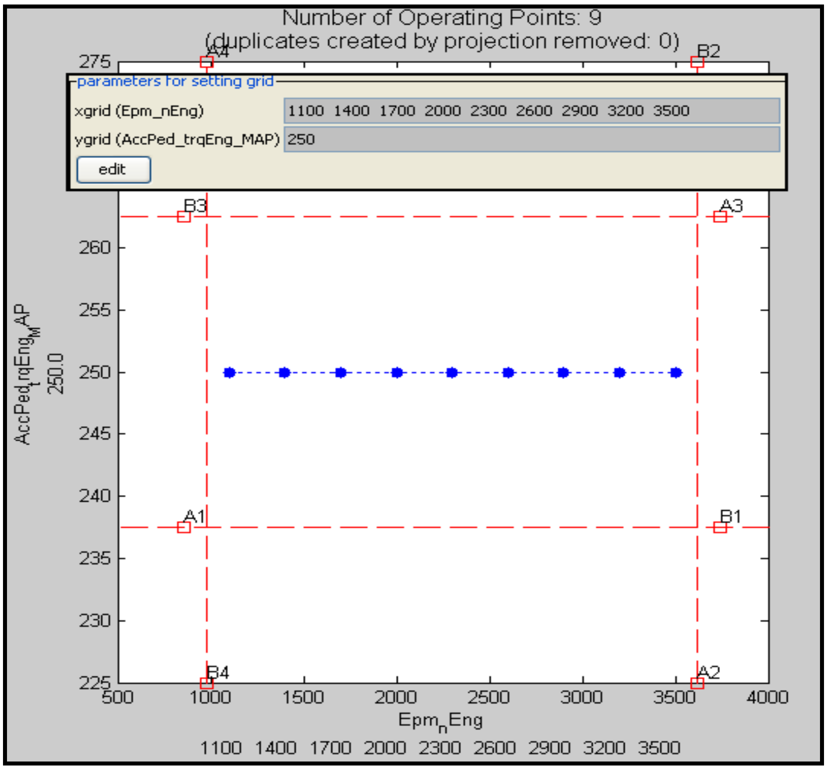


Figure 4.1-10 Edor Step 2

Third step (Figure 4.1-11) is the space filling design. It is possible to observe the distribution of the point in each operating point. The operating points in the borders have more points than the others (histogram in the bottom of the figure), since they are closer to the limits.

Fourth step is where the borders are defined. It is possible observe in Figure 4.1-12 how the k_p lower border behavior with the variation of the engine speed.

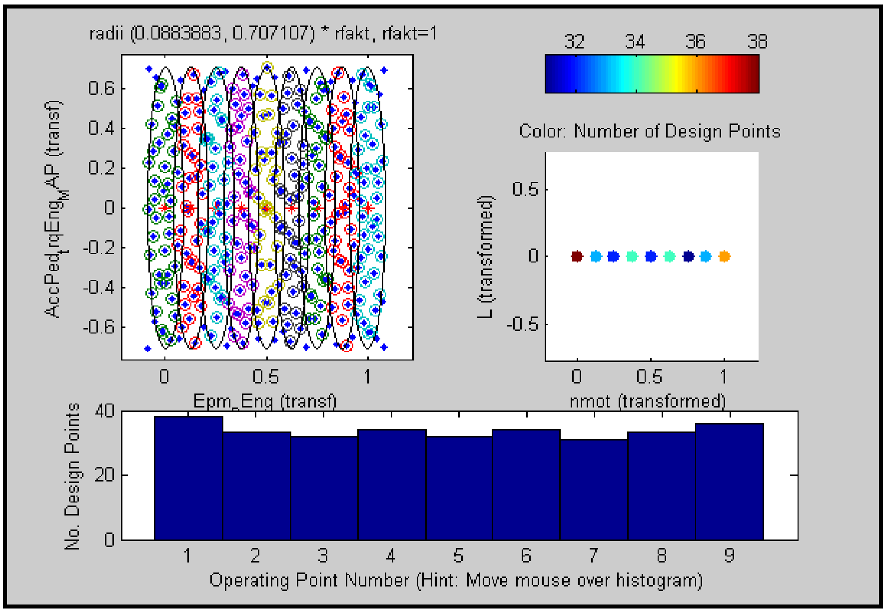


Figure 4.1-11 Edor Step 3

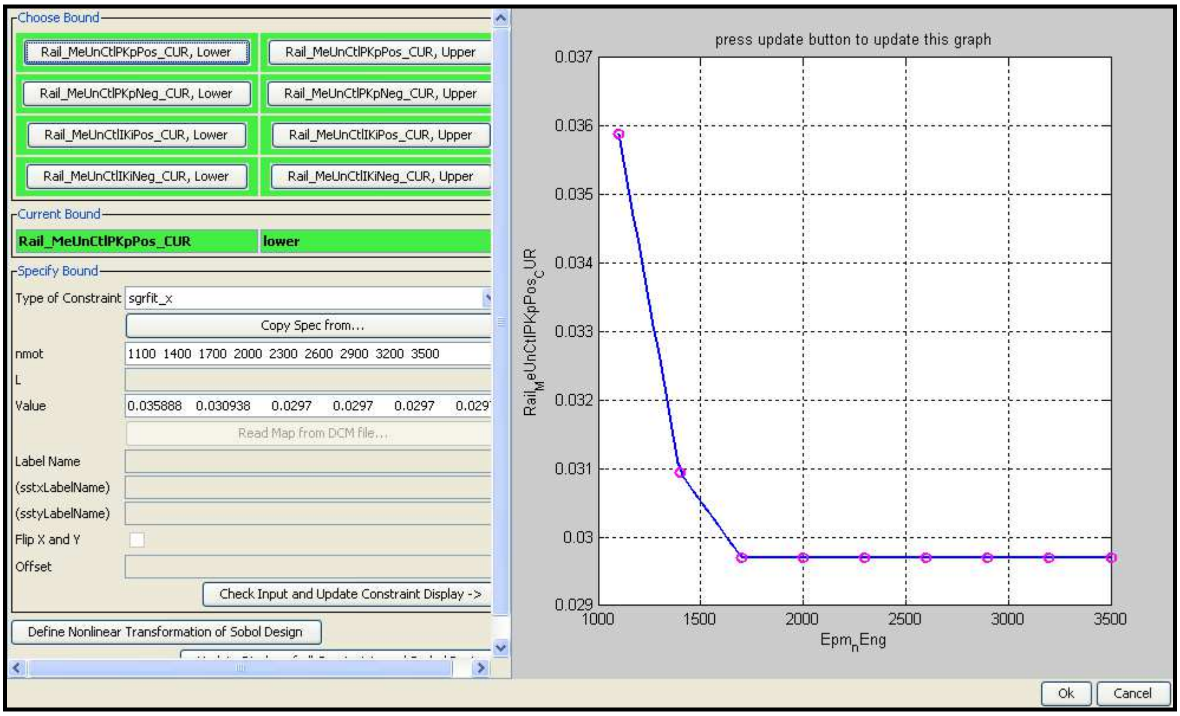


Figure 4.1-12 Edor Step 4

Fifth and last step is saving the result in a excel file. Figure 4.1-13 shows the exported table.

Sobol-No.	bnr	Epm_nEng	AccPed_trqEng_MAP	Rail_MeUnCtlPKpPos_CUR	Rail_MeUnCtlPKpNeg_CUR	Rail_MeUnCtlIKiPos_CUR	Rail_MeUnCtlIKiNeg_CUR
6	1	1100	250	0,050541625	0,050541625	0,0396625	0,0312125
14	1	1100	250	0,047401562	0,041121437	0,03226875	0,03015625
15	1	1100	250	0,050018281	0,042691469	0,033853125	0,034909375
29	1	1100	250	0,042691469	0,039551406	0,037021875	0,033853125
30	1	1100	250	0,046093203	0,045569859	0,040982812	0,039926562
42	1	1100	250	0,038766391	0,048709922	0,037814062	0,040982812
43	1	1100	250	0,040336422	0,052373328	0,045735937	0,045735937
57	1	1100	250	0,051849984	0,047139891	0,042567187	0,044679687
58	1	1100	250	0,043345648	0,049364102	0,032664844	0,043755469
72	1	1100	250	0,048579086	0,046224039	0,029496094	0,042699219
73	1	1100	250	0,047009055	0,046747383	0,037417969	0,037946094

Figure 4.1-13 Edor Step 5

4.2 AUTOMATED MEASUREMENT

The automated measurement is the step of the project where the vehicle measurement starts. With the help of INCA (chapter 3.8.1) and the tool Bender (chapter 3.8.3), all the parameters necessary in the EDC17 will be changed to measure each point with the correct configuration.

First of all, it is necessary to be clear all labels that we need to add to the Experiment in Inca (Figure 3.9-1 shows the basic ones) and take care about the rate of each signal. The rate used in the rail pressure is 10 ms.

After the creation of a good experiment, the bender package must be created. For the C4 is necessary an initial calibration to switch to MeUn control and switch k_d curve to zero. For that a so-called DCM file is created with all information necessary to execute that change before the measurement starts.

Besides that, it is necessary to know how to control the torque, which it will be the stimulus to start the TipIn (see figure below). A TipIn is how it is called a step response in a vehicle. A step response in a vehicle is basically when the accelerator pedal is pressed completely and, as a reaction, the engine speed increases and, of course, the pressure increases too with a high gradient. In the case of C4, this control will be done using the label AccPed_trqEng_MAP.

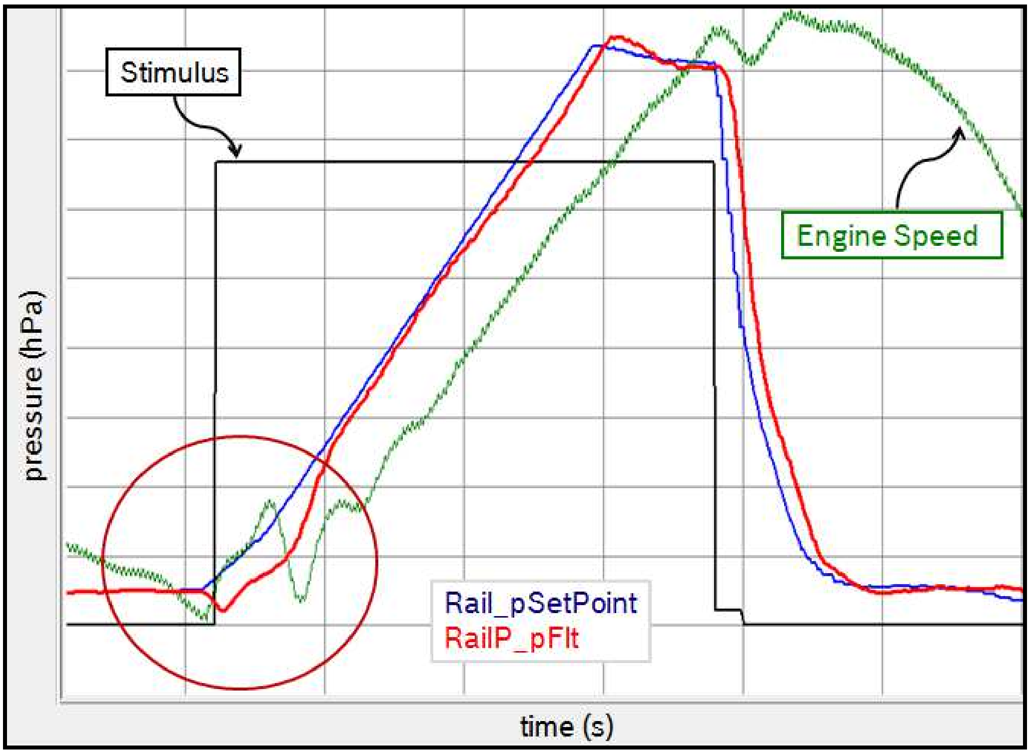


Figure 4.2-1 Stimulus

The Bender package can be created as explained in chapter 3.8.3 and the measurements can start. The measurements must be done on a properly track. The procedure is: the driver will start the vehicle, go to the second gear and press play in Bender, after that he will keep the accelerator pedal completely pressed and Bender will make the TipIn's with the strategy defined in the test plan automatically. As it was explained before, the idea is validate the method to the worst case, so the second gear was chose because in this gear the gradients of the pressure are higher and, besides that, it is the more safety gear to do TipIn's, since the vehicle speed is not so high.

You can find some photos of the two tracks used in this project and of the measurement in the C4 attached to this paper.

4.3 CRITERIA CALCULATION

In the rail pressure governor calibration some criteria are defined already, with standards names, as it is possible see on Figure 4.3-1.

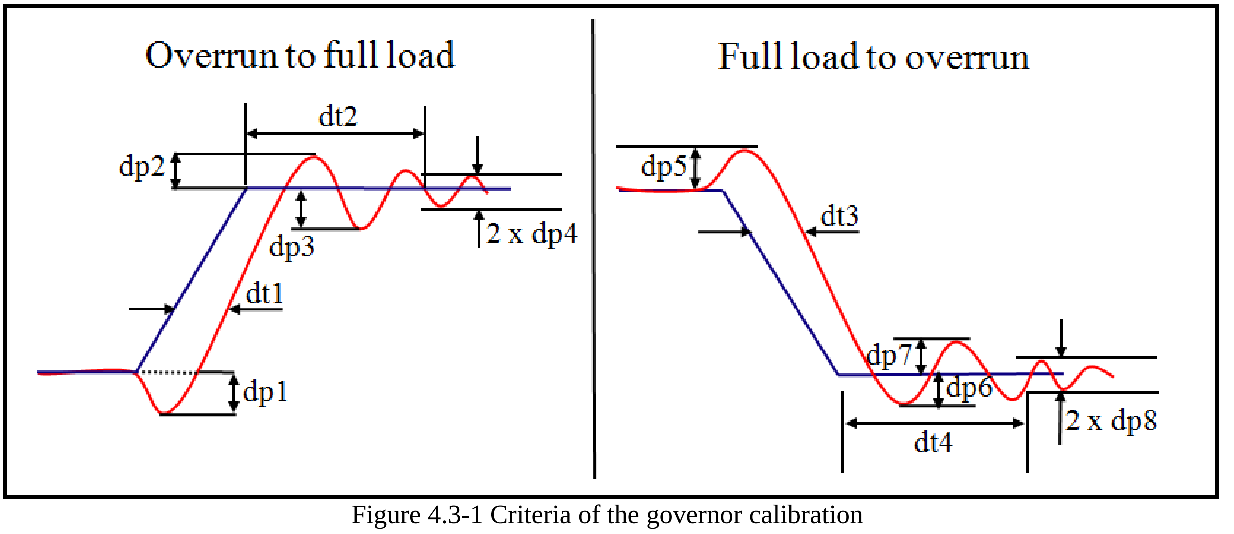


Figure 4.3-1 Criteria of the governor calibration

In a TipIn (from overrun to full load) there are six criteria: dp1, dt1, dp2/dp3, dt2 and dp4. A dp1 is the undershoot when the accelerator pedal is pressed. The delay between rail pressure set point and measured rail pressure set point during the rise time is called dt1. The overshoot or undershoot when rail pressure set point reach a plateau is called dp2, if it is an over and dp3, if it is an undershoot. Dp4 is rail pressure oscillation in full load. Dt2 is the duration till measured rail pressure is stabilized (rail pressure deviation below dp4).

In a TipOut (from full load to overrun) there are six criteria defined too: dp5, dt3, dp7/dp6, dt4 and dp8. The overshoot when the gas pedal is released is called dp5. Dt3 is the delay between rail pressure set point and measured rail pressure set point. The overshoot (undershoot) when rail pressure set point reach a plateau is called dp7 (dp6). Dp8 is the rail pressure oscillation in overrun. And dt4 is the duration till measured rail pressure is stabilized (rail pressure below dp8).

Table 4.3-1 Desired quality[6]

CRS	dp1	dt1	dp2	dp3	dt2	dp4	dp5	dt3	dp6	dp7	dt4	dp8
	bar	**ms**	**bar**	**bar**	**sec**	**bar**	**bar**	**ms**	**bar**	**bar**	**sec**	**Bar**
Nominal	60	180	20	30	0,3	25	70	350[7]	30	50	1,4	10
Min/Max	70	220	40	50	0,8	25	80	4507	60	60	2	10

Above is a table with the desired values of defined criteria to a MeUn control mode in case of one controller calibration. It is just a tip, in some vehicles this values cannot be reached.

[6] Font: BOSCH Group. Calibration hint (internal document). Stuttgart, 2010
[7] This value depends strongly on injector leakage and cannot be influenced by rail pressure controller.

Since this project tries to optimize just the large signal, there is almost no influence on dp1, dt2 and dp4, because the rail pressure deviation is inside the small window.

Figure 4.3-2 shows an example of the C4 behavior in a TipIn. In this chapter, just some of the ideas about which criteria to use will be exposed. However, during the progress, a total of seven ideas were developed just for TipIn. For each idea models were created and validations in the car were measured. Studies were done to decide what criterion would reflect the system behavior better and, after two months, the final criteria for TipIn were defined.

The first idea for developing a code to calculate the TipIn criteria was using criteria already defined. The code for dt1, dp2/dp3 and dp5 was developed in MatLab.

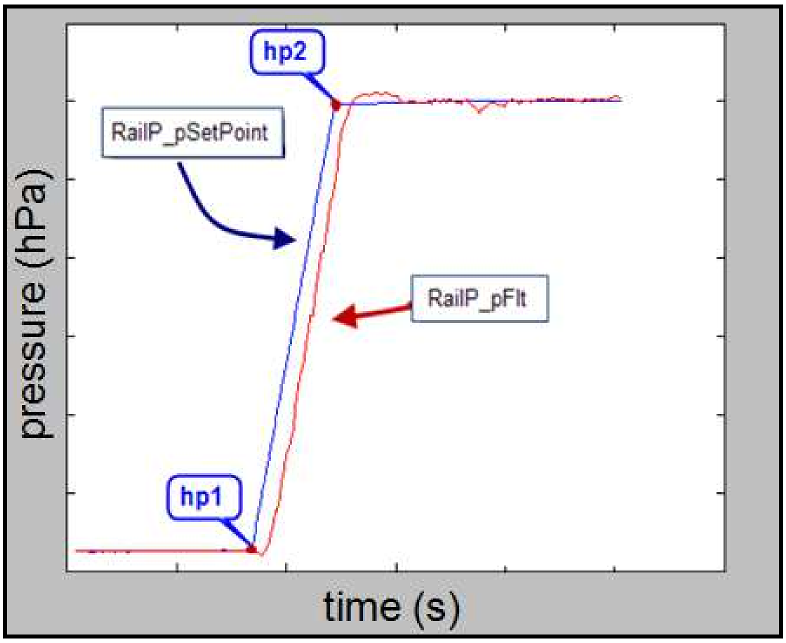


Figure 4.3-2 TipIn

To find this criteria is necessary to find help point 1 and 2 (hp1 and hp2 in Figure 4.3-2). You can see the code and the UML model to find dp2/dp3 and the help points in appendix B. One of the problems of choosing these criteria are that is not a simple code to find the precise point.

For example, let’s analyze Figure 4.3-3. How to find the delay? How to decide if the rise time in the area ‘1’ should be included or not in the calculation? If included, the delay will be ‘a’, that is not correct. If excluded, how should be the size of area ‘1’? Similar problems appeared to find dp2/dp3.

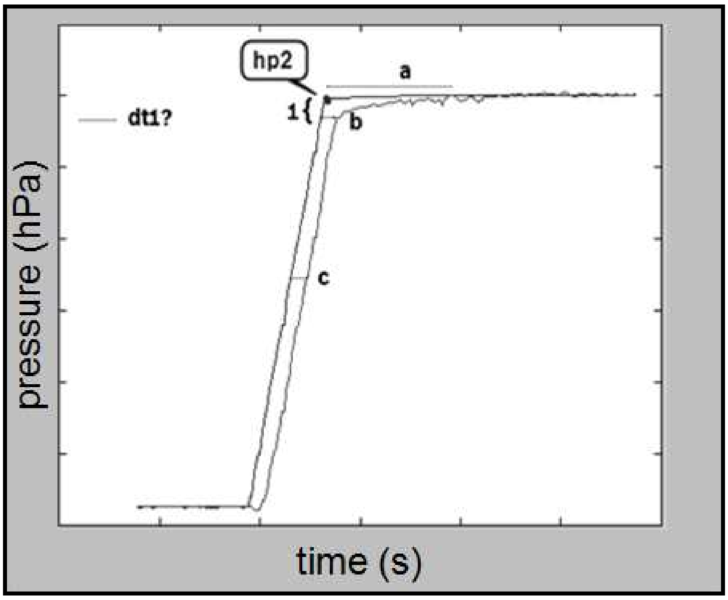


Figure 4.3-3 Delay

To solve this problem, instead of using one specific point (e.g. dt1 and dp2/dp3), the integral deviation will be used. The second idea developed is the deviation integral in Plateau (reflect dp2/dp3) and deviation integral in rise time (reflect the delay) as Figure 4.3-4 shows.

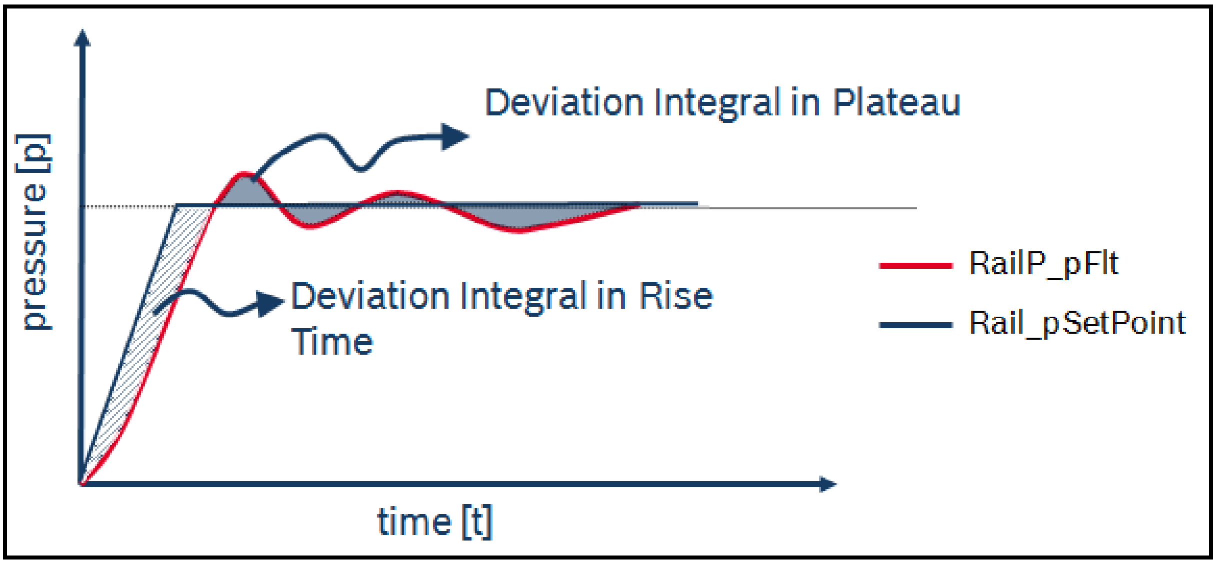


Figure 4.3-4 Deviation Integral

That way, there is no problem to decide a specific point. However, a new problem appears. It is still necessary to find help points 1 and 2. For the C4 behavior, it is not so difficult to find these help points. However if the shape of Rail_pSetPoint has no hard edge, finding these points can be problematic (remember that this method should be applied for other vehicles too – See Figure 4.3-5). That means the code is not robust enough to be applied in other shapes of the rail pressure set point.

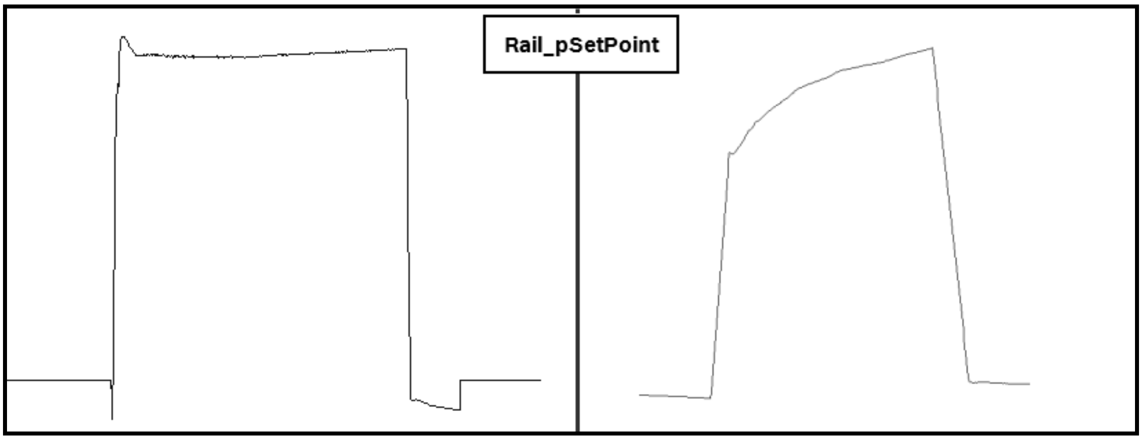


Figure 4.3-5 Rail_pSetPoint' shapes

To solve this problem, simple criteria were chosen to find a model: so-called 'Positive' and 'Negative area' (Figure 4.3-6). Using these, it is not necessary to find the help point 2 and it is a deviation integral, not a specific point in a curve. The negative area criterion reflects the delay and the undershoot; the positive area criterion is a reflex of the overshoot.

$$Rail\_pDvt(t) = error(t) = Rail\_pSetPo\text{int}(t) - Rail\_Flt(t)$$

$$IE^* = \int_0^\infty error(t)dt$$

$$Rail\_pDvt(t) \geq 0 \rightarrow NegativeArea$$

$$Rail\_pDvt(t) \lhd 0 \rightarrow PositiveArea$$

$$*\,Integral\_of\_Error$$

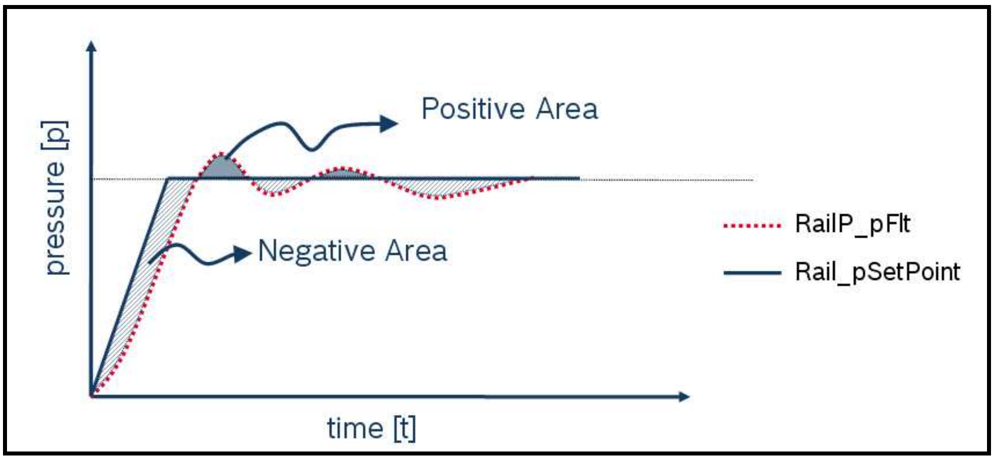


Figure 4.3-6 Positive and Negative area

Now, criteria to find a model for a TipOut need to be developed. A TipOut is when the accelerator pedal is completely pressed and, suddenly is completely released, going to overrun. When this happened, the injections go from a high value to close to zero very fast. This means that the MeUn device is completely closed to reduce the pressure as fast as the setpoint is reduced and, since the injections were reduced too, the only way to keep reducing the pressure is the leakage. However, the PID controller has no influence concerns the leakage. So, it is expected that the influence of the PID in the TipOut is not as strong as in the TipIn.

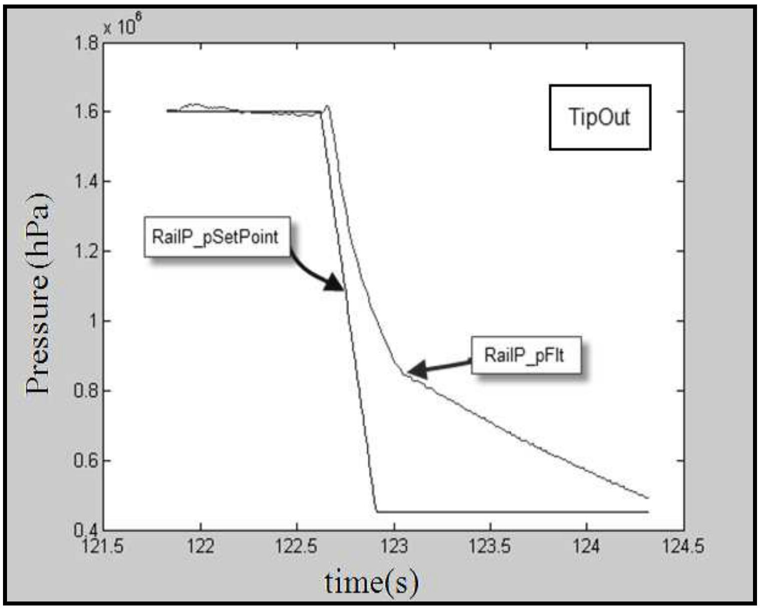


Figure 4.3-7 TipOut – C4

Figure 4.3-7 shows a TipOut behavior in C4. In the measurements it is possible to analyze that when a TipOut occurs, the MeUn is completely closed, but the pressure cannot be reduced as fast as the setpoint decreases. A Matlab code to find dp5 was developed, but it was not possible to find a model with the results. To confirm this hypothesis, T90 was measured too, that means the time that the system takes to go from 90% to 0% (see Figure 4.3-8). And, as expected, this time was always the same for all parameters.

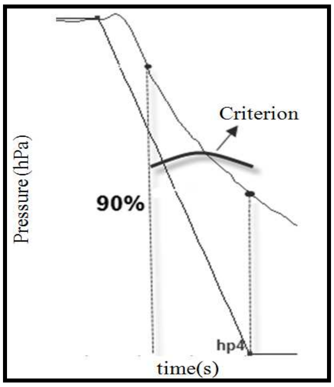


Figure 4.3-8 T90

It is important to remember, as explained in chapter 3.5, that k_{pneg} and k_{ineg} are active when the deviation (setpoint minus actual value) is negative. That means that these parameters are active just in TipOut and when there is a big overshoot in TipIn. In the calibration hints, the suggestion is to use the same values to k_{ppos} (k_{ipos}) and k_{pneg} (k_{ineg}) and analyze the behavior. The same procedure was done here for C4. The values of k_{ppos} and k_{ipos}

were optimized using the TipIn measurements and used to k_{pneg} and k_{ineg} too. This suggestion will be used to all other systems that have the same behavior.

However, the behavior observed in Figure 4.3-7 is specific to some systems. Another car was used to make new tests about TipOut and to validate the method. It is a Ford Fiesta with just a MeUn actuator. Figure 4.3-9 shows how the TipOut of this vehicle is.

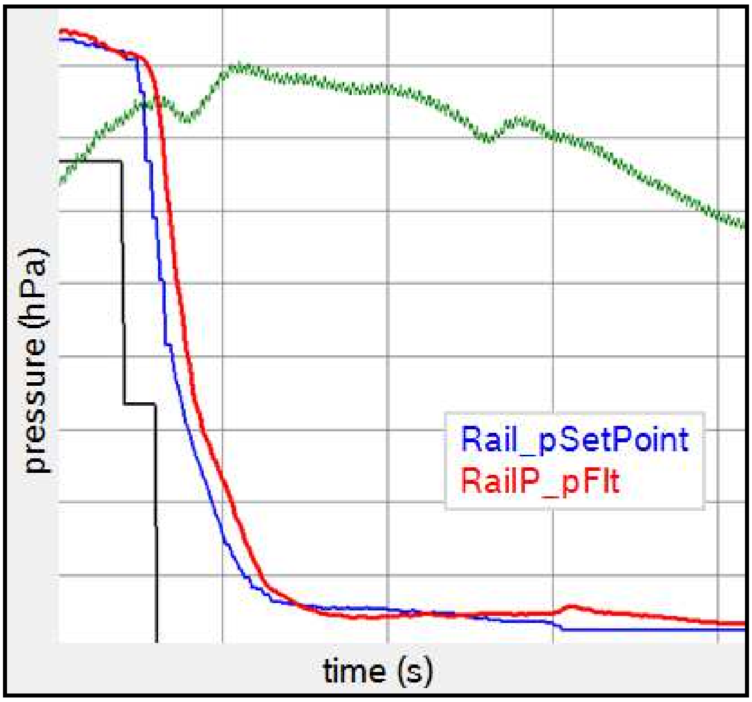


Figure 4.3-9 TipOut - Ford Fiesta

Four criteria were developed for TipOut:

- dp5 (overshoot when gas pedal is released) – already used in C4
- T90 – already used in C4
- Positive integral deviation
- Negative integral deviation

The model quality from the first and second criteria were less than 80% and no relation were find between the large signal and these criteria. This result was expected for two mainly reasons:

- As studied during the development of criteria to TipIn's, find a specific point can be problematic, because of the big diversity of setpoint shapes.
- And, as explained before, the influence of the PID in a TipOut is not as strong as in a TipIn. The duration time to reduce the pressure cannot be decreased just with a PID, since the leakage is the mainly way to reduce the pressure during a TipOut. That explains why the model using T90 was not good. In respect of dp5, beside these reasons, the small window has more

influence than the large window in this case, since the deviation is still inside the small window.

With the last two criteria (Positive and negative integral deviation) combined in the same way as in the TipIn it was possible to understand what is the influence of the PID controller in the TipOut.

Figure 4.3-10 and Figure 4.3-11 shows two TipOut's, the first with a high T_N (and small k_{ineg}) and the second with a small T_N (and high k_{ineg}).

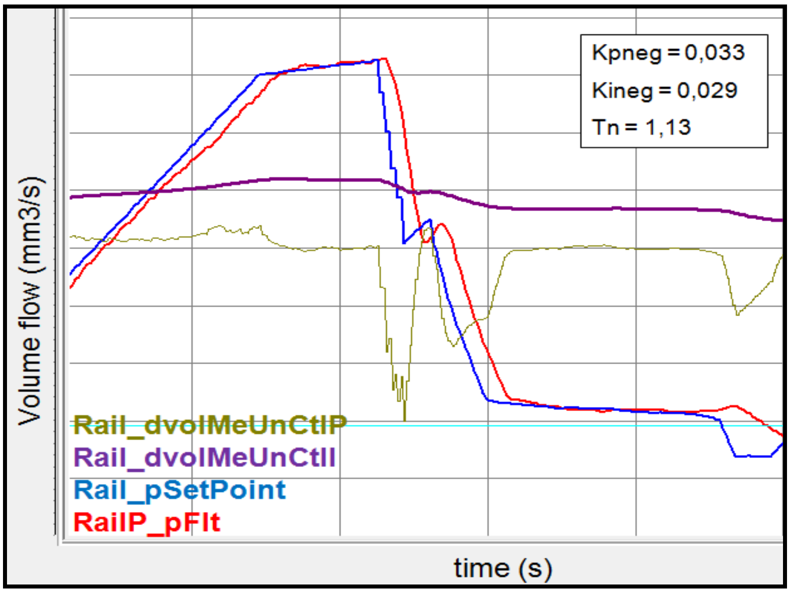


Figure 4.3-10 TipOut - analyses f behavior 1

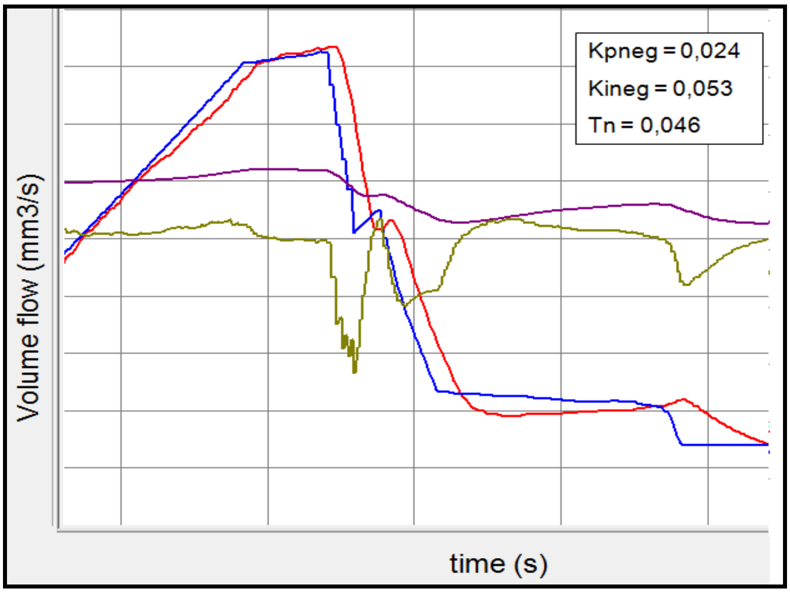


Figure 4.3-11 TipOut - analyses f behavior 2

The influence of the controller appears in the undershoot in overrun. With the two pictures above it is possible to see the reason and how to avoid this problem. The signal in purple, called Rail_dvolMeUnCtlI, is the physical output of the I-gain, which means, it is a volume flow. In the case of a small T_N (Figure 4.3-10), the value of Rail_dvolMeUnCtlI is reduced slowly and stay almost constant when the RailP_pFlt reaches the setpoint. However, when T_N is too small (or k_{ineg} is too high), as in Figure 4.3-11, the output of I-gain reduces too fast and in the moment that RailP_pFlt reaches the setpoint, the I-gain output is already too small and it needs some time to reach the correct value again. So, it is clear that, to avoid this behavior the value of k_{ineg} cannot be too high.

After this analyses it is clear that the negative integral deviation is related direct to this undershoot, so it is expected that the model with this criterion will optimize the only characteristic that the PID can really control in a TipOut. To make sure that the model will keep a reasonable value in $k_{p,}$ the positive integral deviation will be used as in the TipIn. In this way, the time to reduce the pressure will stay as faster as possible (remember that the positive integral deviation reflects the delay). So, the criteria defined for a TipOut are positive and negative integral deviation, as in a TipIn (Figure 4.3-12).

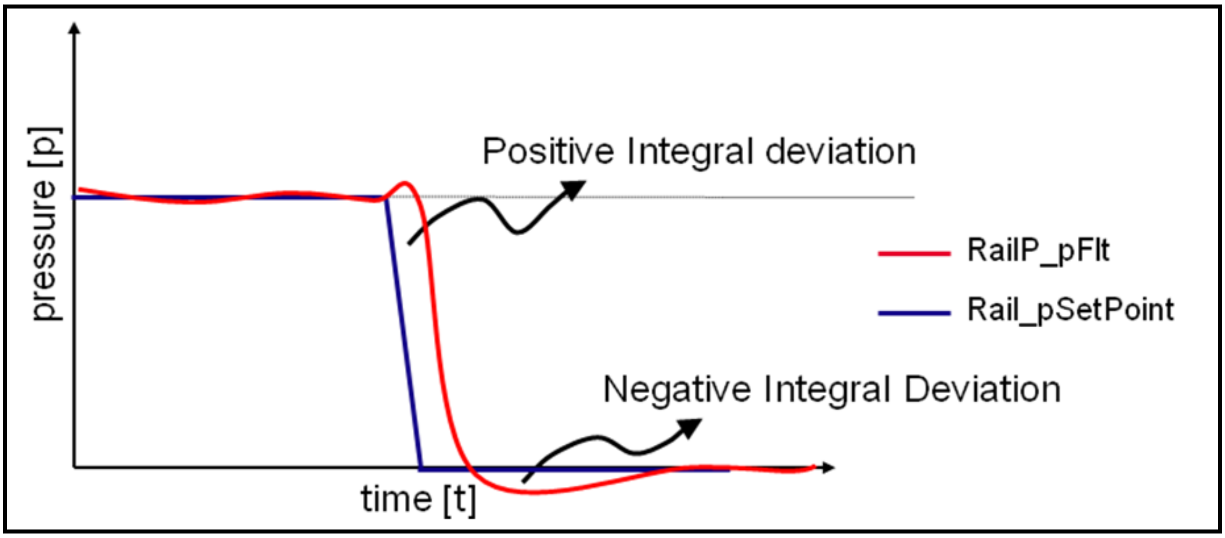


Figure 4.3-12 TipOut criteria

To finalize the step of criteria calculation, the codes developed are integrated to Bender 4 (chapter 3.8.4) and the measurements are evaluated using these criteria. As output of this step, an excel table with all points measured (parameters value, engine speed and torque) and the respective criteria values (negative and positive area) is created (one for TipIn and one for TipOut).

4.4 MODEL OPTIMIZATION

After doing the measurements and evaluating them in Bender 4, it is necessary to find the model optimization using ASCMO (chapter 3.7).

First just TipIn will be analyzed. As said before, the criteria chosen were negative area and positive area. Figure 4.4-1 shows how the model of this system looks like. In the left part of the figure is the negative area and positive area and in the bottom is the engine speed (left), k_{ppos} (middle) and k_{ipos} (right). It is possible to see the influence of k_p and k_i in criteria. For example, if k_i is too high, the tendency is having a faster system, but with more overshoot. This means more positive area and less negative, as it is possible observe below.

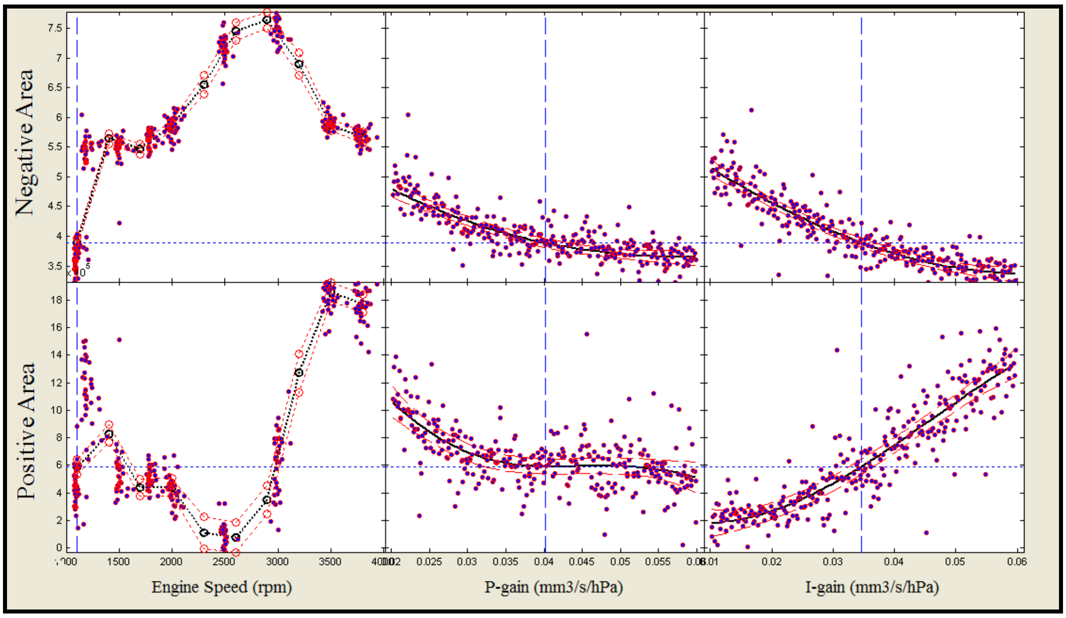


Figure 4.4-1 Model design - TipIn

Figure 4.4-2 shows the model error. The R^2 of both criteria are almost perfect (96% for negative and 95% for positive). Fourteen points were considered outliers in negative area and eighteen in positive area.

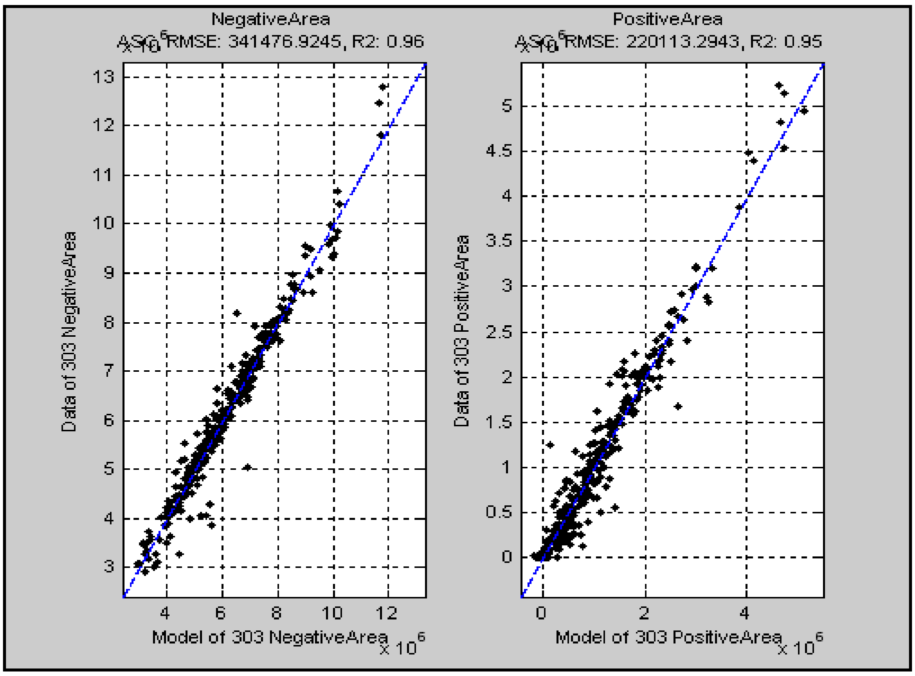


Figure 4.4-2 Model Error

Figure 4.4-3 shows the Pareto for this optimization for an engine speed of 2300 rpm. This model is looking for the smallest negative and a positive area closest to zero. The optimum points are located in the bottom right of the figure, where positive area is closer to zero and negative area is the smallest value possible for that. For each of the engine speeds (nine, in this case) a Pareto front will be designed. Attached in this document, it is possible to find the graphic for other engine speeds.

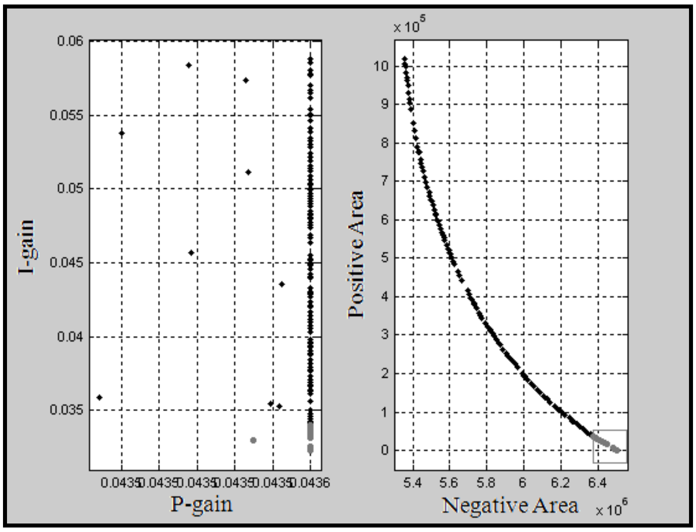


Figure 4.4-3 Pareto Optimal 2300 rpm

Figure 4.4-4 shows one point in the measurement that k_p and k_i from the large window are very close to the point in the Pareto. It is a good calibration, with a very small overshoot (6,9 bar) and a delay of 48,7 ms (have a look again in Table 4.3-1).

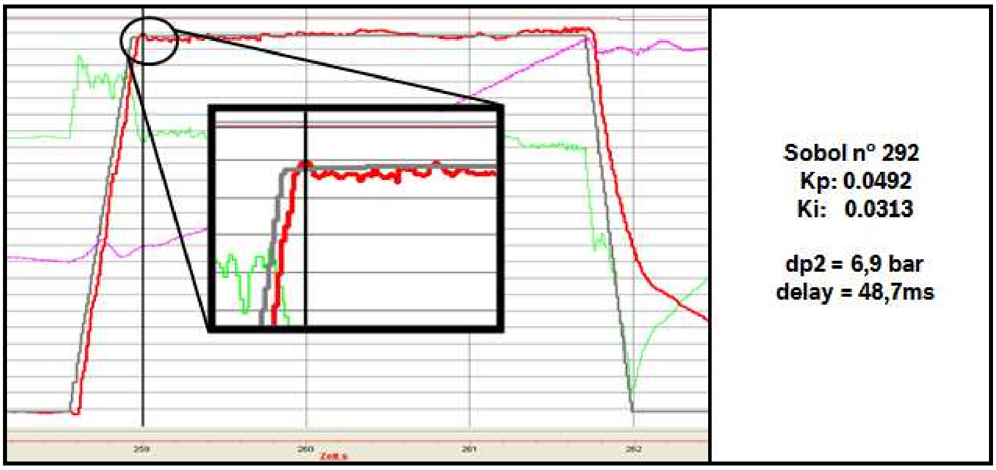


Figure 4.4-4 Measurement

Using the error over training data size (explained in page 60), it is possible to conclude that at least 250 points are necessary to have a good model quality. Figure 4.4-5 and Figure 4.4-6 shows the result for all nine engine speeds.

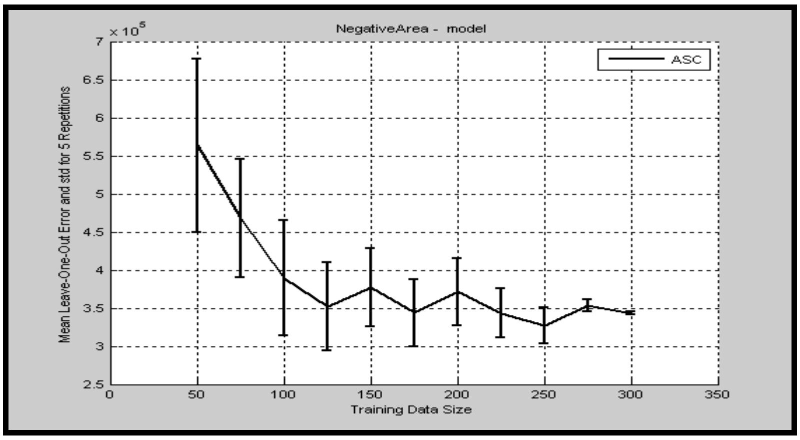


Figure 4.4-5 Error over training data - Negative Area

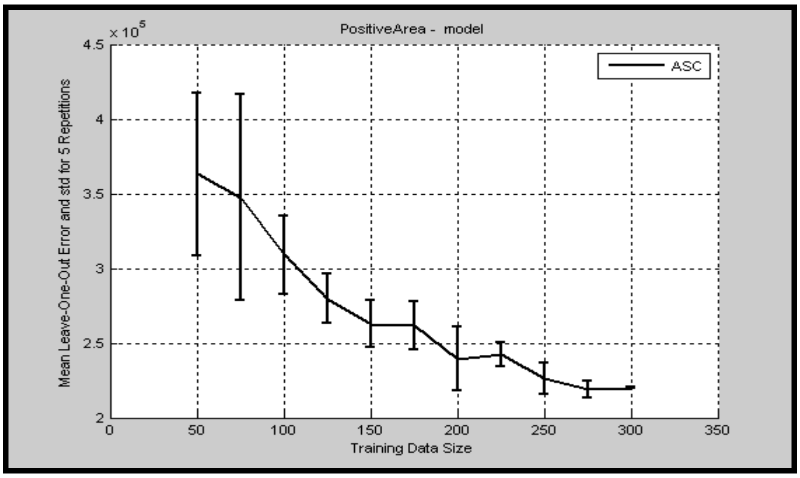
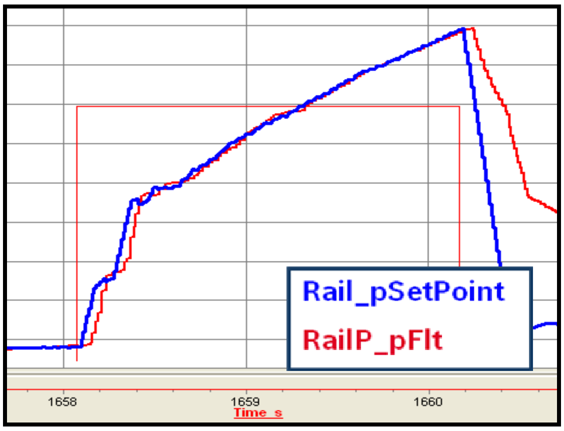
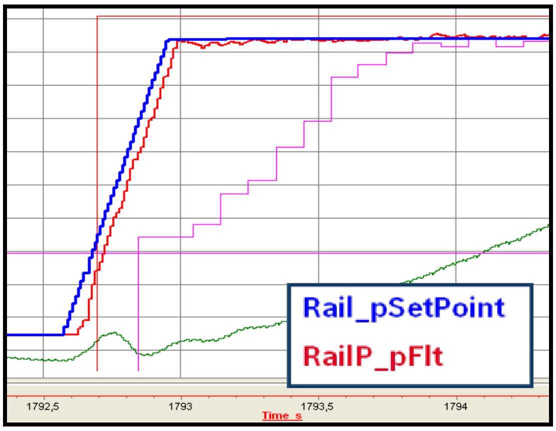


Figure 4.4-6 Error over training data - Positive Area

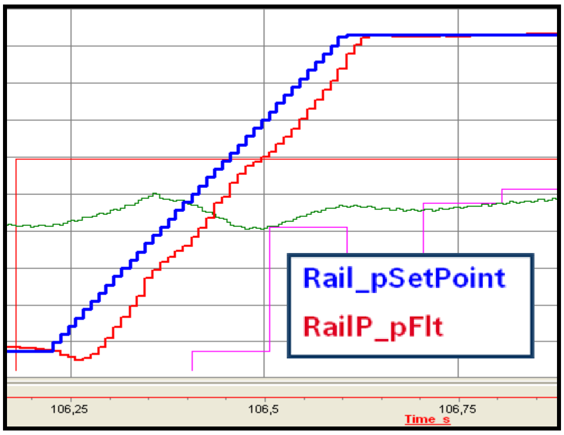
To validate these results, one more time is necessary going to the test track and with the Fry tool confirms if the results are good or changes are necessary. As explained in chapter 3.8.5, the only thing necessary to use fry is an a2l file and excels tables with the Pareto front of each engine speed that is necessary to optimize. The next three figures show the result of this validation in some of the engine speeds chosen.



TipIn: 1100 rpm	
Torque	250 Nm
kp	0,0387
ki	0,029
Overshoot	6 bar
Delay	50 ms



TipIn: 2600 rpm	
Torque	250 Nm
kp	0,044
ki	0,036
Overshoot	-1,5 bar
Delay	50ms



TipIn: 3200 rpm	
Torque	250 Nm
kp	0,043
ki	0,038
Overshoot	0 bar
Delay	51 ms

With this validation, it is possible to conclude that the calibration goal can be reached for a TipIn. With this method, the system has a fast behavior and a minimum overshoot.

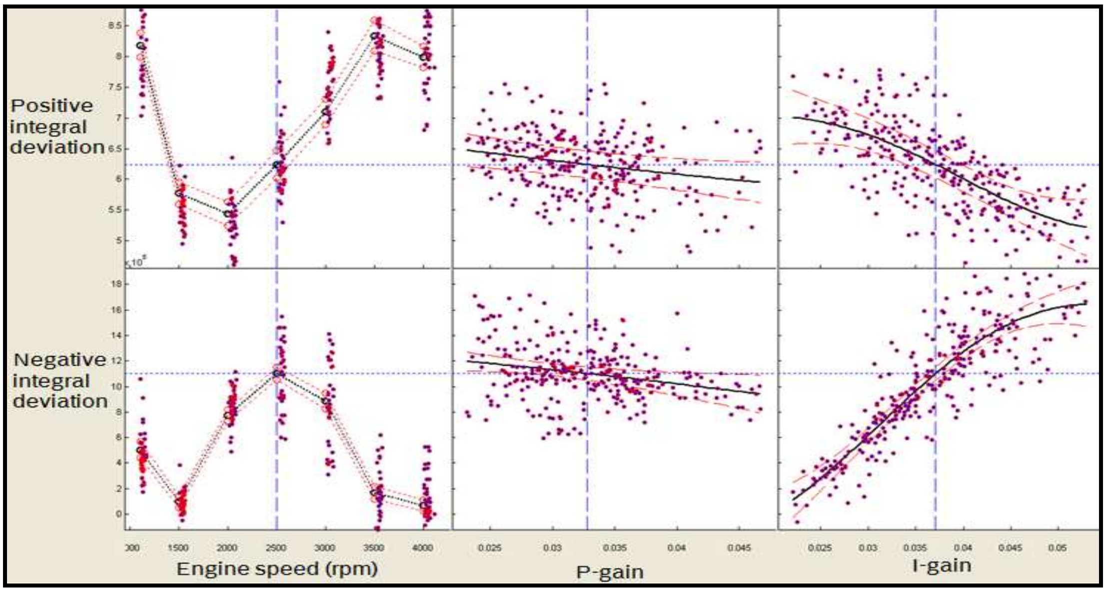


Figure 4.4-7 Model design - TipOut

The same procedure is repeated to a TipOut. As said before, a Ford Fiesta was used to make the TipOut study, since with C4 was not possible to find a model. Figure 4.4-7 shows the model design. As expected, to reduce the negative area it is necessary to decrease the value of I-gain. Besides that, it is possible to see that, the variation of P-gain has no big influence in the output.

All the others steps were repeated to a TipOut. It was observed that is necessary 250 points to have a model and the quality was not as good as in a TipIn. The negative area has an R^2 of 82% and the positive area 84%.

The Pareto front from each engine speed was generated and the optimum values were found. The results have an interest behavior: the values of k_{ppos} and k_{pneg} were almost the same, the biggest variation appears in the k_{ineg} value, it is always smaller than k_{ipos}.

The next three pictures show the validation in three engine speeds: 1500, 3000 and 4000 rpm.

Figure 4.4-8 Validation TipOut - 1500 rpm

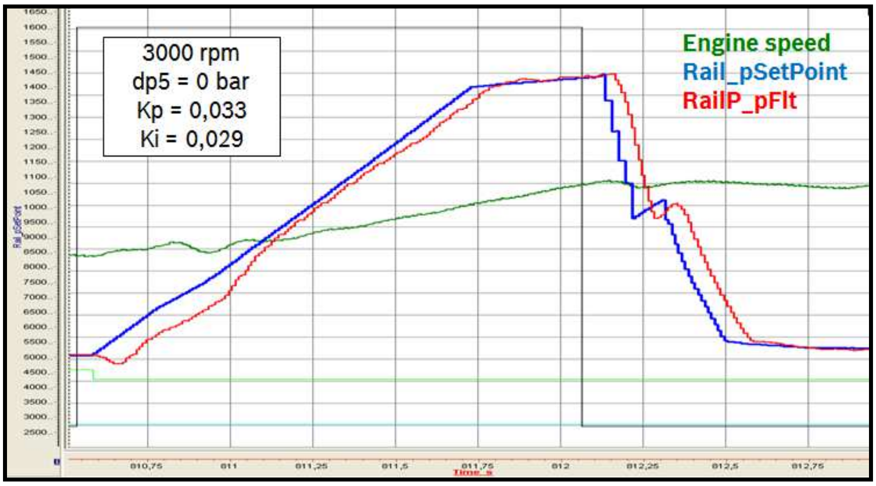


Figure 4.4-9 Validation TipOut - 3000 rpm

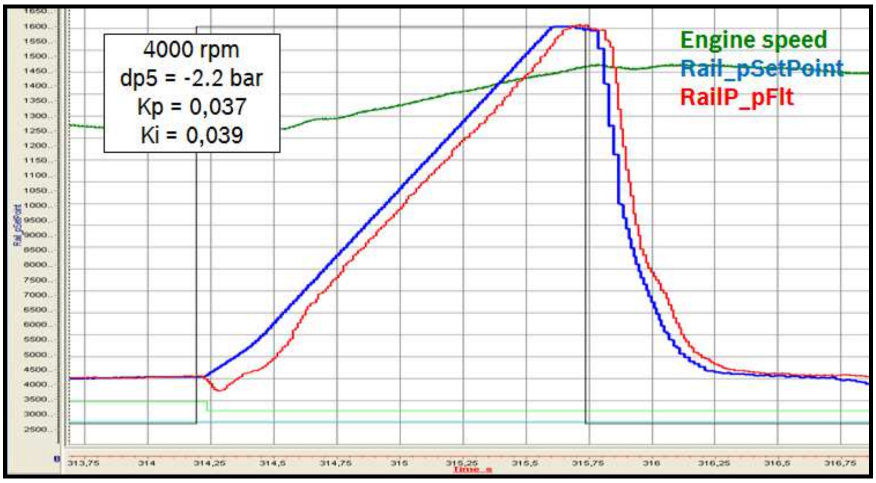


Figure 4.4-10 Validation TipOut - 4000 rpm

4.5 VALIDATION WITH A SECOND CAR

To confirm the applicability of this method, all the procedure was executed again in a second C4 with the same characteristics as the first. The same test plan with 300 point, nine engine speeds and a torque of 250 Nm were measured again in this second car and the results will be presented here. The TipOut was not analyzed, since the system behavior is the same as the other C4.

The model error has a very good quality (R^2 equal to 95% to negative area and 98% to positive area) as the first C4, as it is possible to observe in Figure 4.5-1.

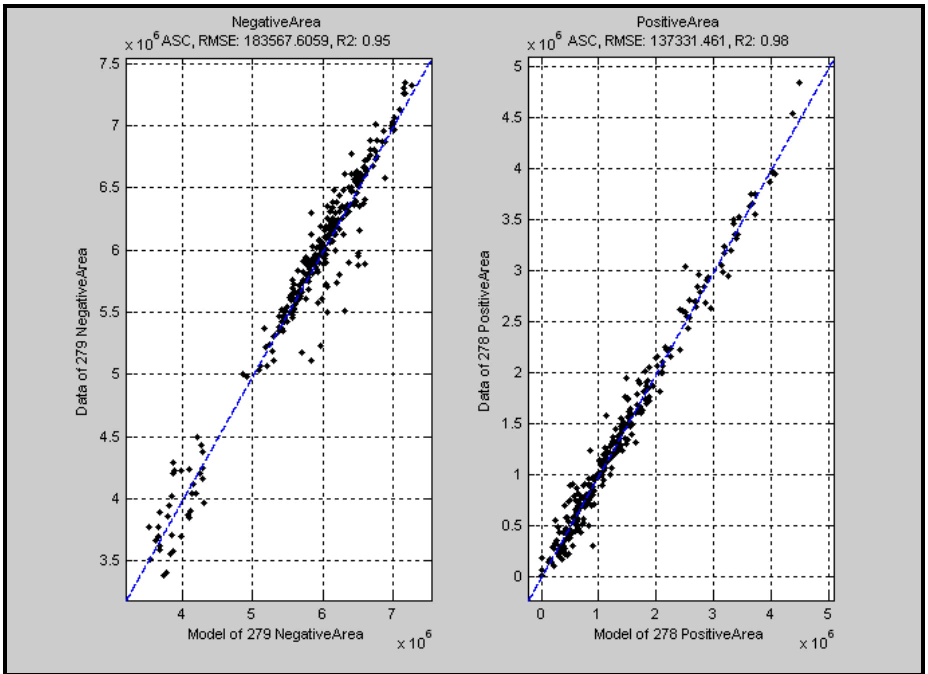


Figure 4.5-1 Model Error - 2nd C4

The number of points necessary to design a model is, as expected, the same minimum 200 points as before (Figure 4.5-2 and Figure 4.5-3).

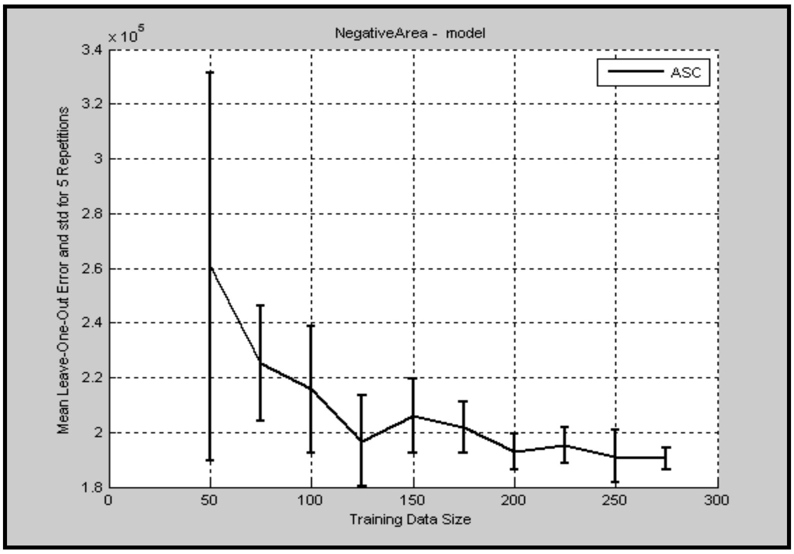


Figure 4.5-2 Error over training data size - 2nd C4

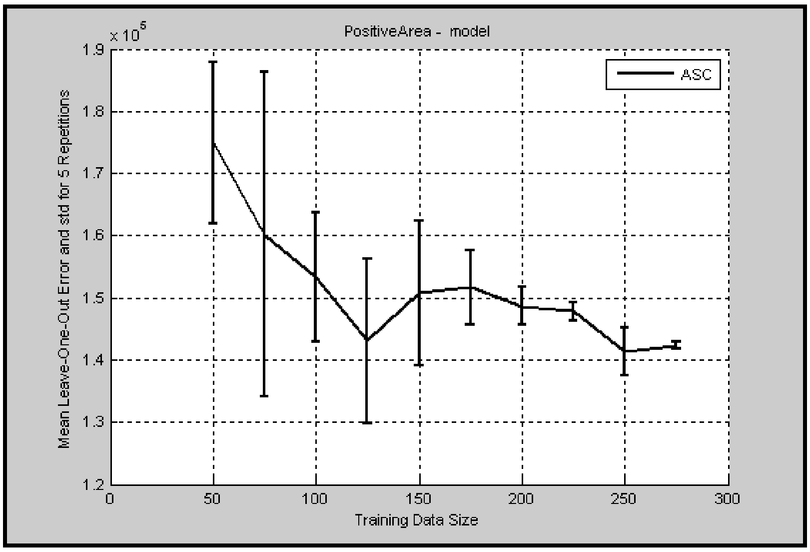


Figure 4.5-3 Error over training data size - 2nd C4

Comparing the model design from the first (Figure 4.4-1) to the second C4 (Figure 4.5-4), it is possible to observe that has the same shape and the same behavior with different engine speeds and a variation of k_{ppos} and k_{ipos}.

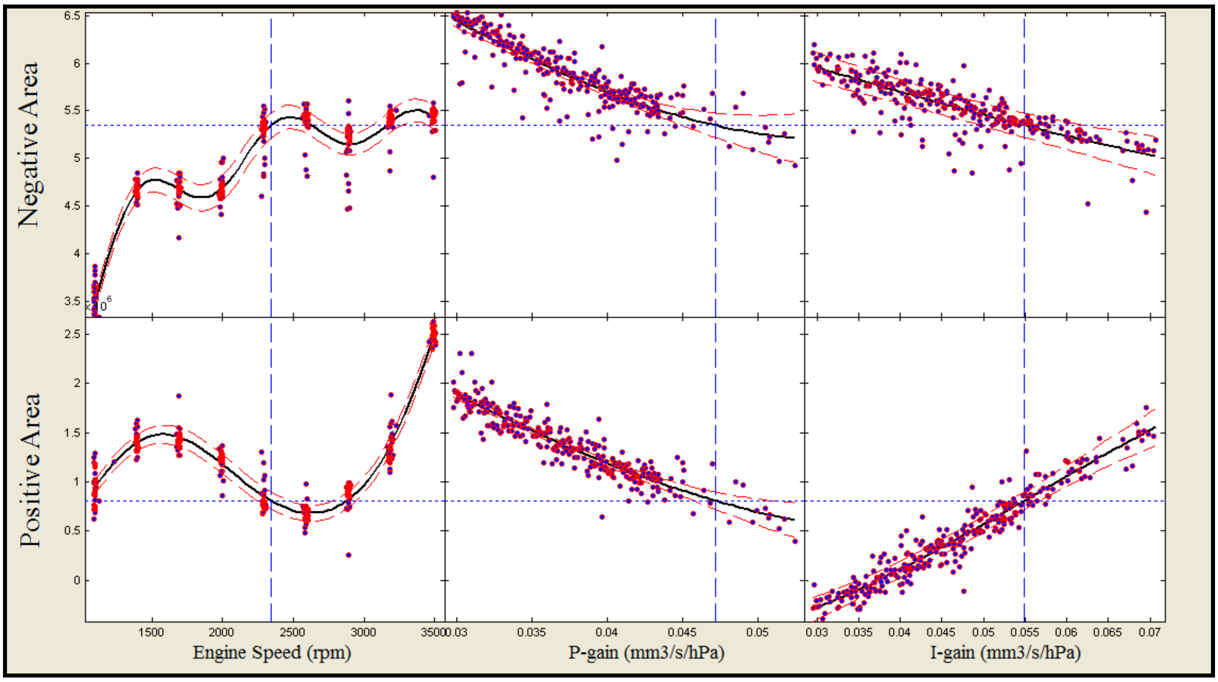


Figure 4.5-4 Model design - 2nd C4

Figure 4.5-5 shows the Pareto front for this optimization for an engine speed of 2300 rpm. As the first car, this model is looking for the smallest negative and a positive area closest to zero. The result is the same as with the first C4, which is really important to confirm the applicability of this method. For each of the engine speeds (nine, in this case) a Pareto front was designed.

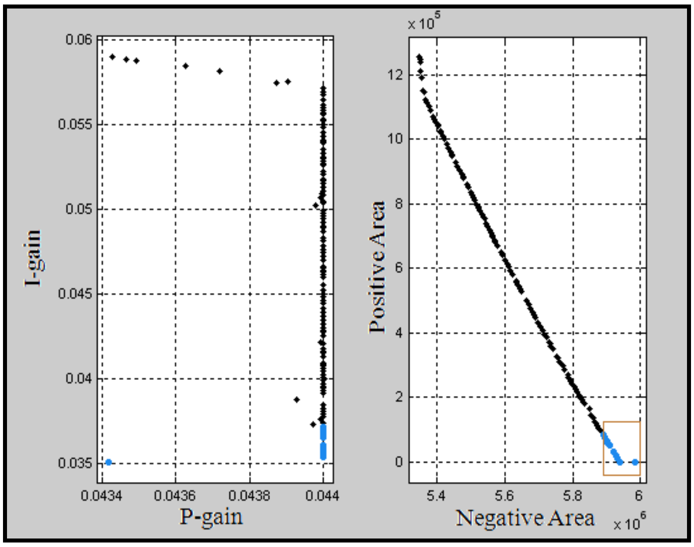


Figure 4.5-5 Pareto front 2300 rpm - 2nd C4

With the second C4 a complete test drive was realized to confirm the robustness of the results. All tests will be presented now.

The next five figures show the result of this validation in some of the engine speeds chosen (1100, 1400, 1700, 3200 and 3500 rpm). To confirm one of the advantages of this method - quality, it is possible to see in the next five pictures a comparison between the calibrations with this method and the calibration done with the traditional method.

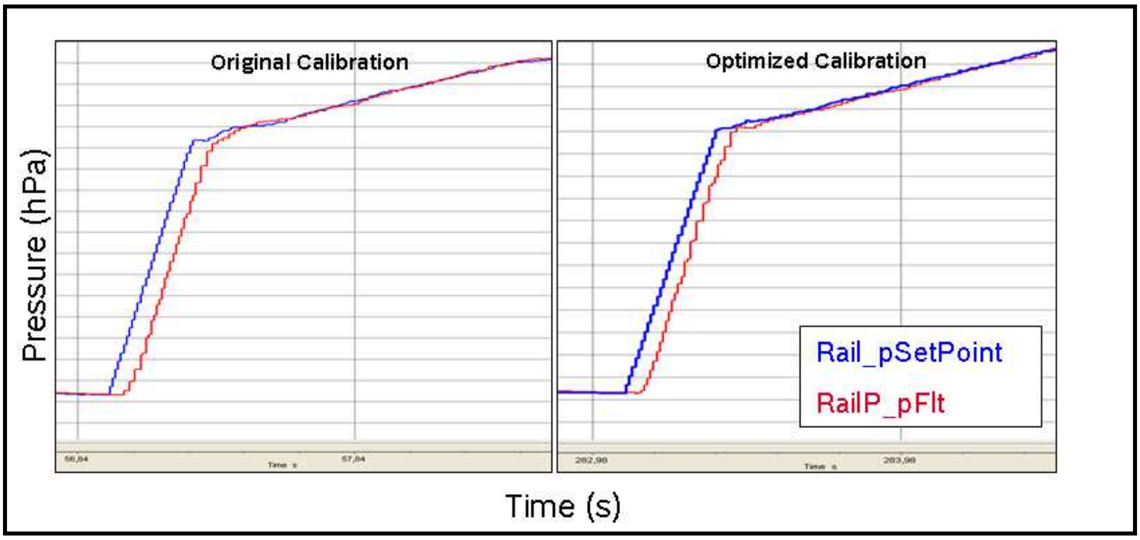


Figure 4.5-6 1400 rpm

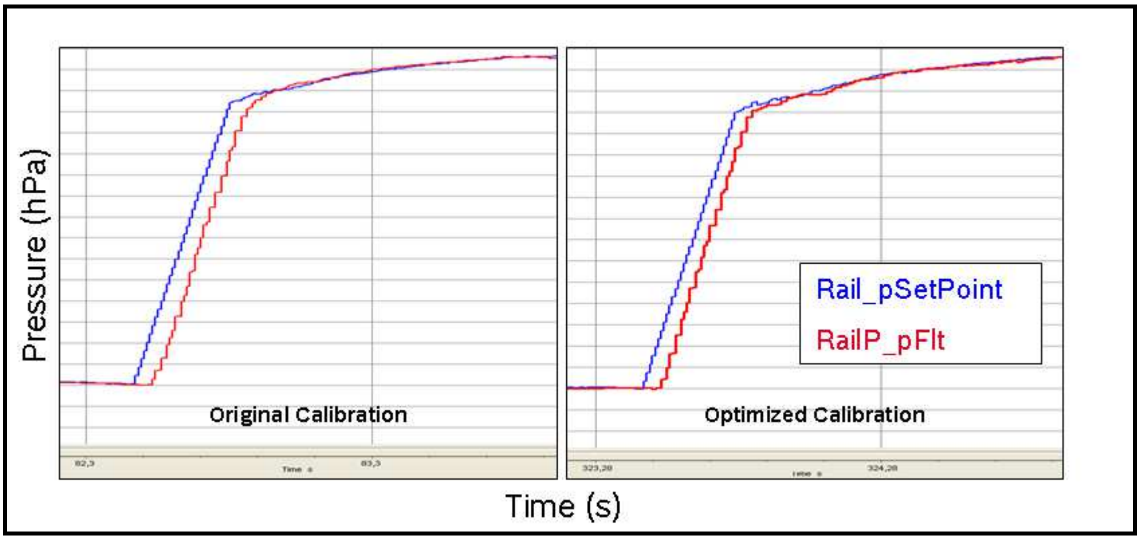


Figure 4.5-7 1700 rpm

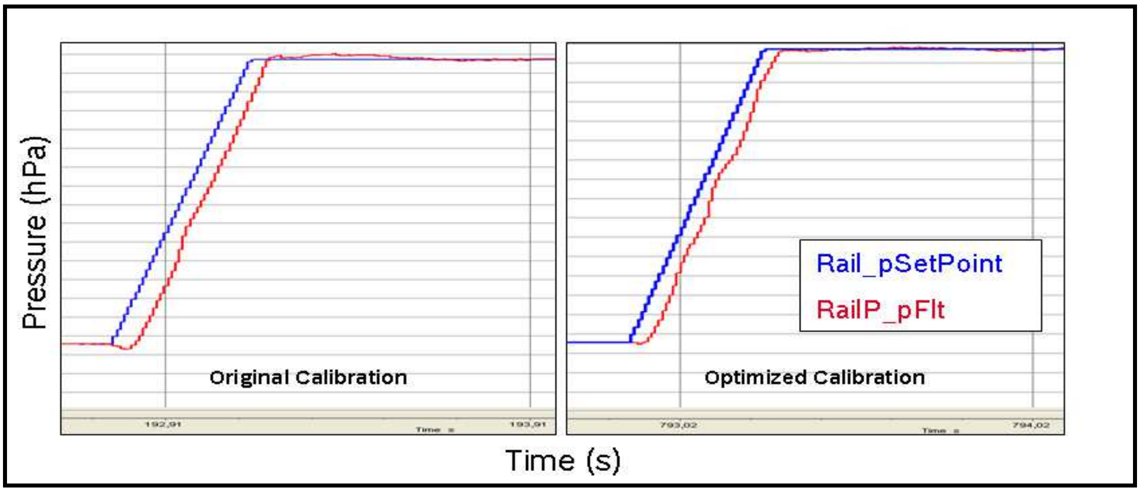


Figure 4.5-8 3200 rpm

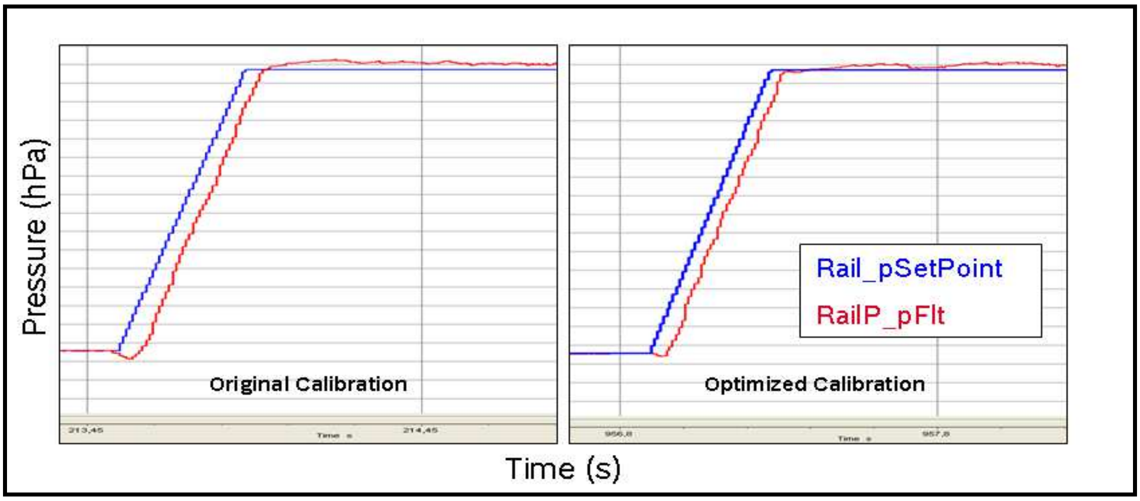


Figure 4.5-9 3500 rpm

Next table shows a comparison about overshoot and delay between the results with original and optimized calibration. It is possible to see that in all engine speeds the optimized calibration is faster (smaller delay) and has smaller overshoots.

Table 4.5-1Original vs. Optimized calibration

Torque: 250 Nm	Calibration					
	Original		Optimized		% of Reduction	
Engine Speed	Delay (ms)	Overshoot (bar)	Delay (ms)	Overshoot (bar)	Delay (ms)	Overshoot (bar)
1100	90	-12	60	-4	33,3%	66,7%
1400	71	-8	60	-6	15,5%	25,0%
1700	80	-20	61	-10	23,8%	50,0%
2000	60	-10	59	6	1,7%	40,0%
2300	70	20	50	2	28,6%	90,0%
2600	70	12	60	-10	14,3%	16,7%
2900	60	10	50	4	16,7%	60,0%
3200	60	15	51	-7	15,0%	53,3%
3500	70	36	51	2	27,1%	94,4%

Figure 4.5-10 show how the value of k_{ppos} and k_{ipos} are changing in function of engine speed. It is possible to observe the values of the first and the second C4.

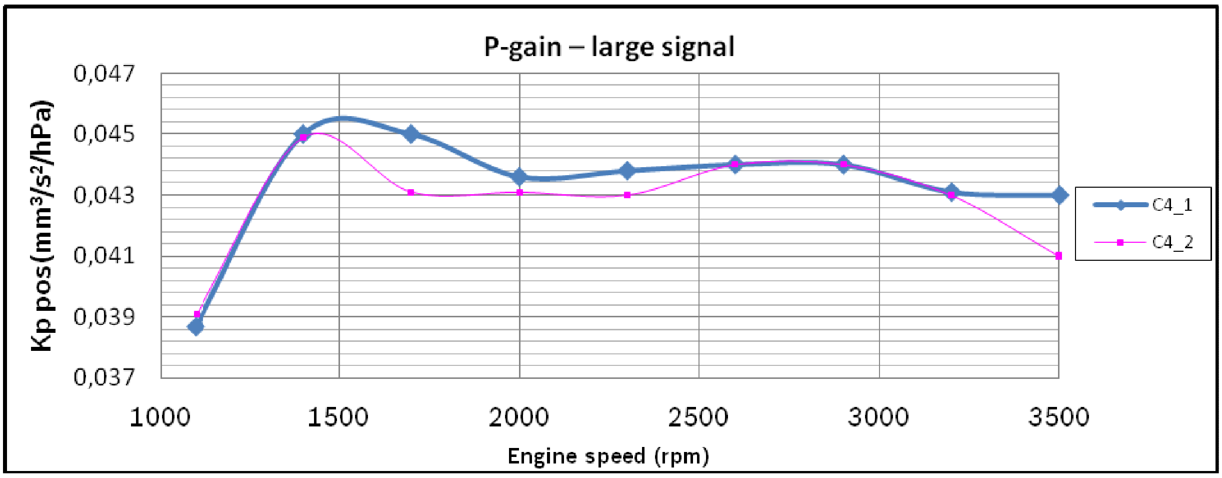


Figure 4.5-10 P-gain optimized

As should be expected, the variations between the two graphics are really small. In the graphic of P-gain, the biggest dispersion is in 1700 rpm with 5% of difference. In the graphic of I-gain, the biggest dispersion is in 1700 rpm too with 3% of difference.

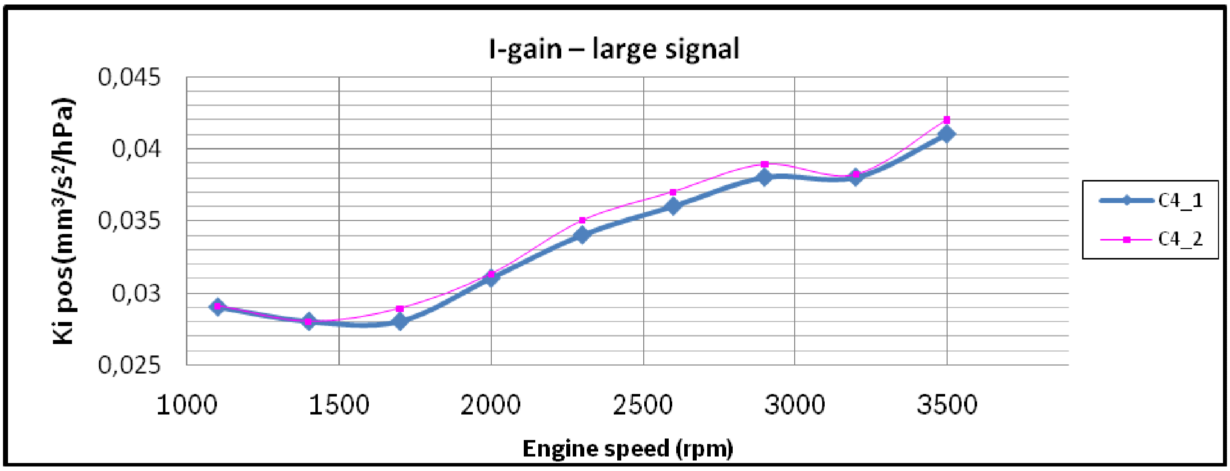


Figure 4.5-11 I-gain optimized

T_N is always within 0,9 and 1,6 and the biggest dispersion is in 1700 rpm, with 7%.

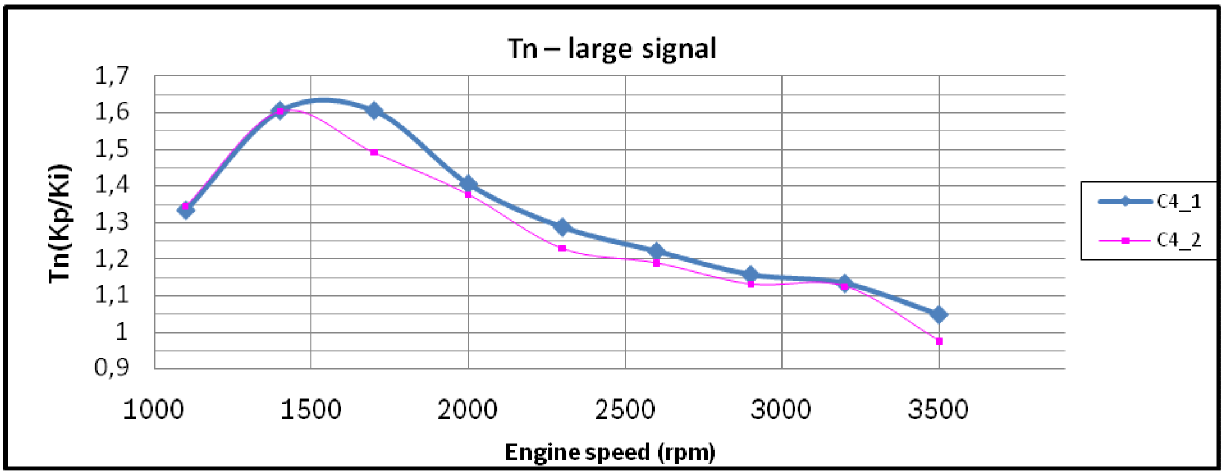


Figure 4.5-12 T_N-large signal

After a calibration, some test drives must be done to check rail pressure stability. It is necessary to check the behavior with nominal and min/max systems too. For that it is necessary to physically change MeUn component to a min and max or simulating this system. The procedure chosen here was simulation. The idea is simple: shift MeUn curve/map that converts volume flow to current about ± 150 mA. In this way, the quantity delivery for the MeUn will change, so it will look likes that the system is using min/max components.

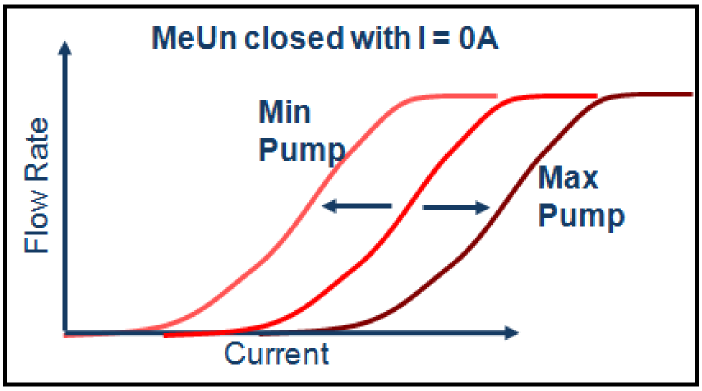


Figure 4.5-13 Min/Max system

The first test drive is basically full load acceleration by shifting gears 1^{st} to 5^{th} gear at 3000 rpm up to approximately 100km/h. This test drive was executed for the original calibration with nominal system, the optimized calibration with nominal system and the optimized calibration with min/max system.

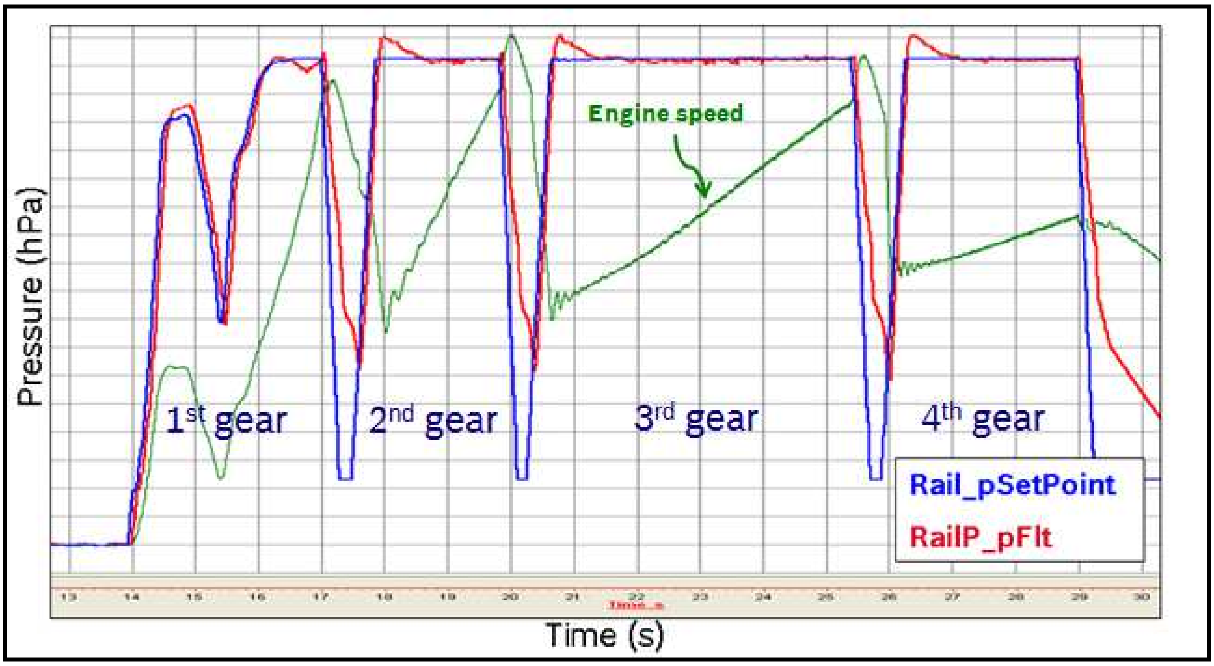


Figure 4.5-14 Test drive 1 - Original values

Figure 4.5-14 shows how this test drive looks like. It is possible observe that the engine speed in green is increasing and when the gear is changing, the value is coming back to a lower value and increasing again until the next gear. In this figure is possible to see a very high overshoot, this happened because the system doesn't have time enough to reach the plateau when is going to overrun (in the moment that is changing from one gear to other). But the important point in this test drive is a guarantee that the system is not going to the monitoring state.

Figure 4.5-15 shows in the same plot original and optimized measurement. The dispersion between both is 10 bars, which means that both have the same quality and both are not leading to a switch to rail pressure monitoring.

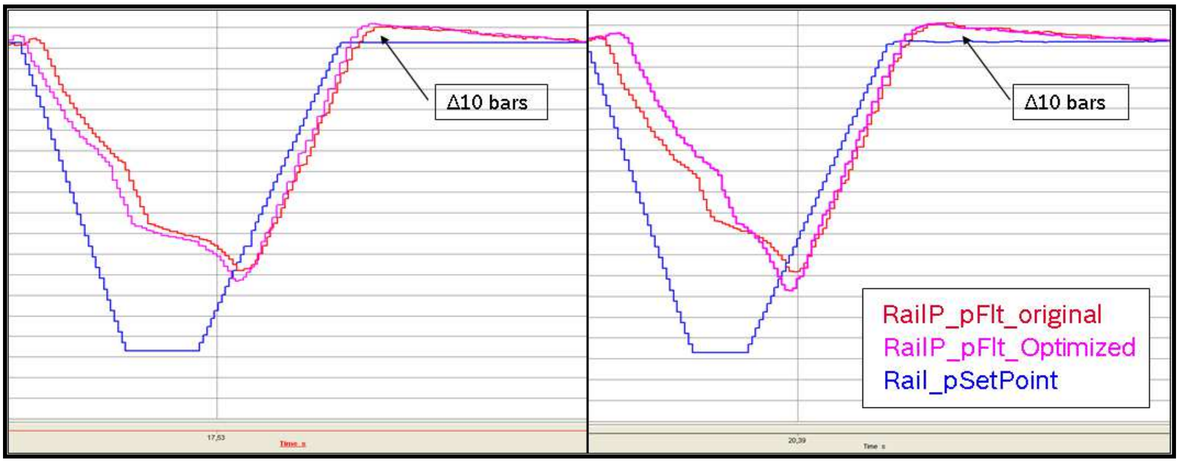


Figure 4.5-15 Test drive 1 - Original vs. Optimized

Below is possible to see a plot comparing the behavior of min/max system with the optimized calibration. The difference between max and min is less than 5 bars, which means the calibration is robust enough for this system variation.

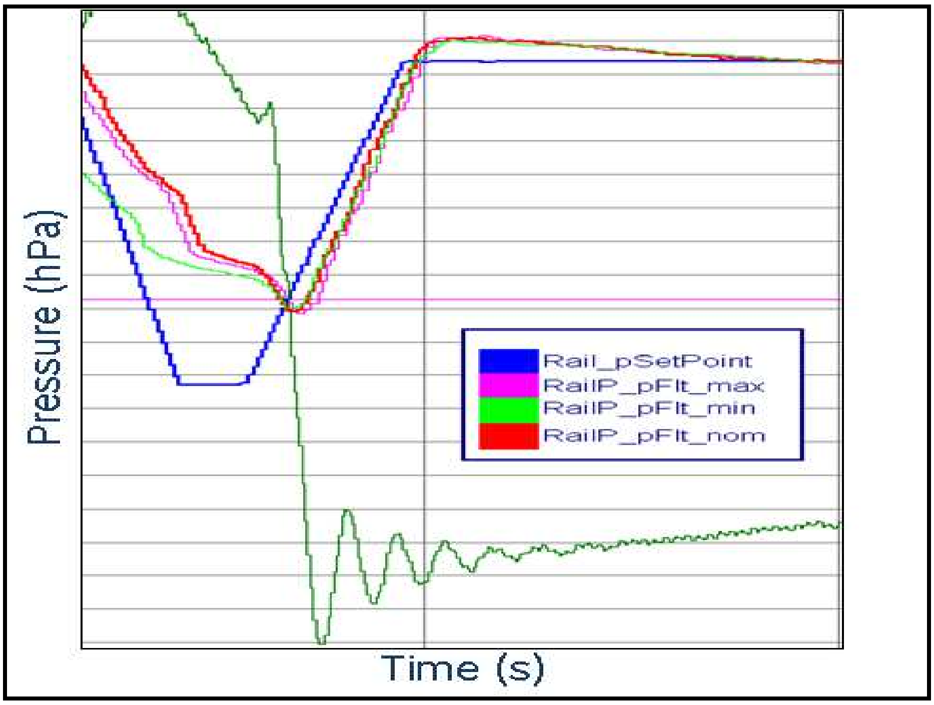


Figure 4.5-16 Test drive 1 - Min/Max system

The second test drive to be realized is the so-called "saw test full load – overrun". The procedure is driving full load/overrun in sequence, e.g.2nd gear (accelerator pedal should be pressed and released suddenly). Table 4.5-2 shows how was the sequence used to do this test drive.

Table 4.5-2 Saw test full load-overrun

Full load	1000…2000	1500…2500	2000…3000	2500…3500	3000…4000	3500…4000
Overrun	2000…1500	2500…2000	3000…2500	3500…3000	4000…3500	rpm

Figure 4.5-17 shows a complete test drive with the optimized values, it is easy to understand why it is called saw test; the engine speed is increasing like a saw.

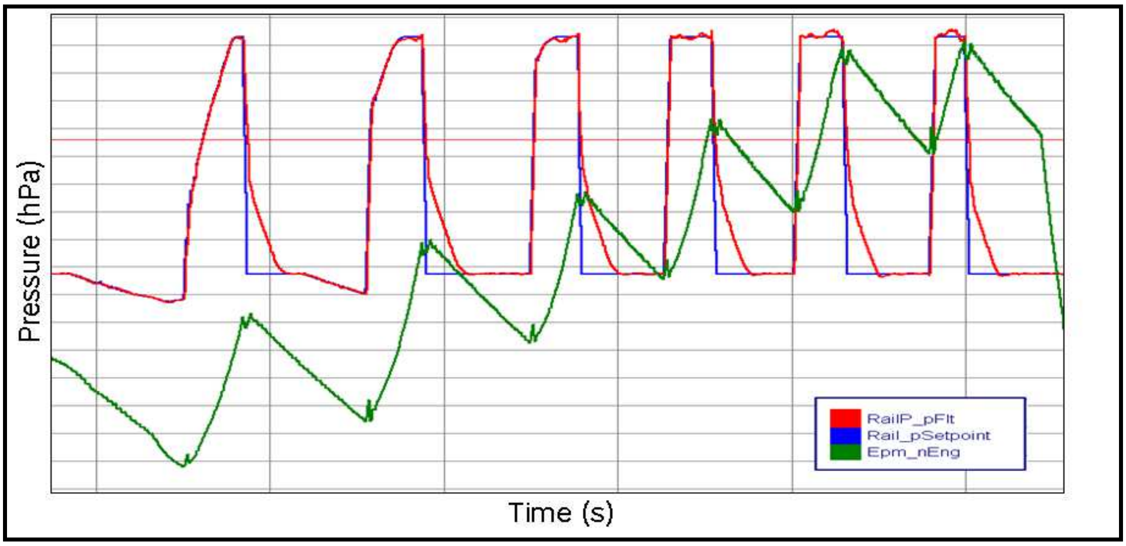


Figure 4.5-17 Saw test full load-overrun

Figure 4.5-18 shows a more detailed plot when the engine speed is going from 2000 to 3000 rpm (the others engine speed have similar behavior). In this measurement is clear why it is important to have the minimum overshoot as possible in nominal system: in a max system more overshoot can be observed, but is still an acceptable behavior (an overshoot of 15 bars); and in a min system some undershoot appears (-12 bars). The plot is using optimized calibration.

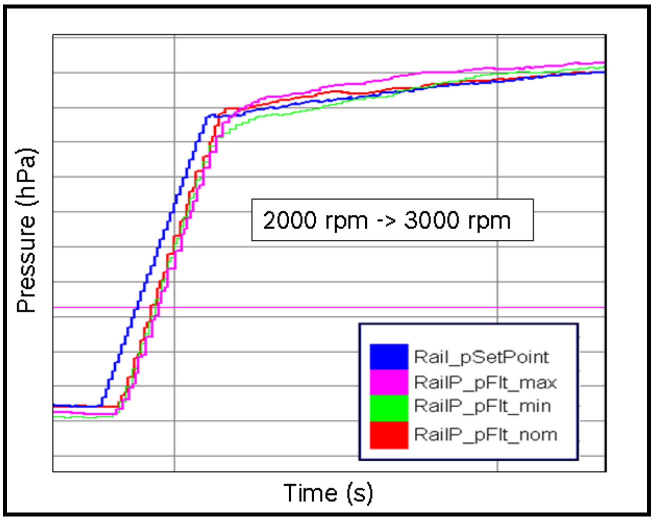


Figure 4.5-18 Saw test drive - 2000 to 3000 rpm

Figure 4.5-19 is comparing original and optimized system. In the left side, a step from 2000 to 3000 rpm to both systems; you can see that the optimized system has a fast behavior with no overshoot. But, the biggest advantage appears in a higher engine speed: in the right side a plot of one step from 3500 to 4500 rpm; with the original system an overshoot of 20 bars is observed, whereas in the optimized, a small overshoot of 9 bars appears.

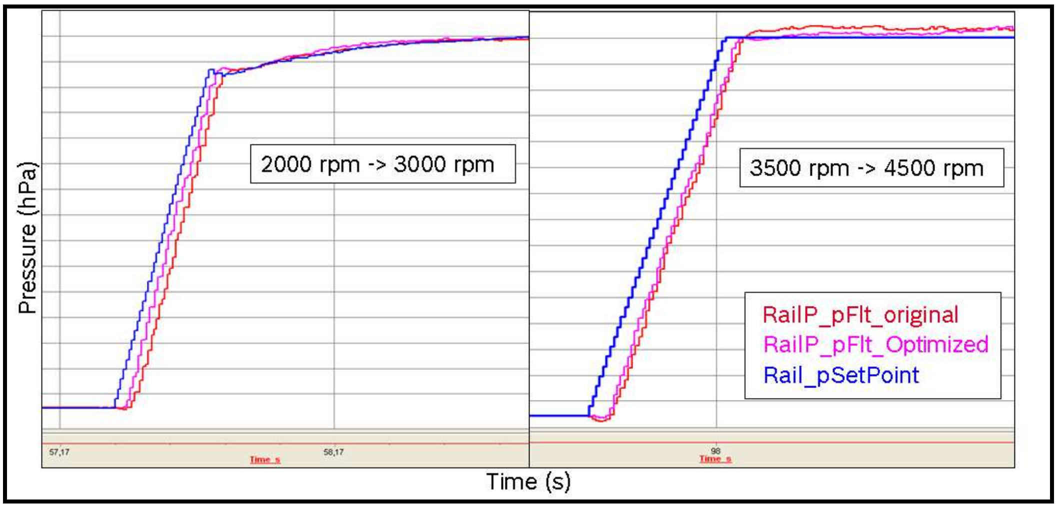


Figure 4.5-19 Saw test drive – original vs. optimized

The same test drive (0 to 100 Km/h and saw test full load-overrun) was executed one last time with the optimized values, but now with everything ON (air conditioning, wipers, warning light flasher, Taillights and so on) to confirm that the behavior will not be more critical than before. Figure 4.5-20 and Figure 4.5-21 shows the test drives' results.

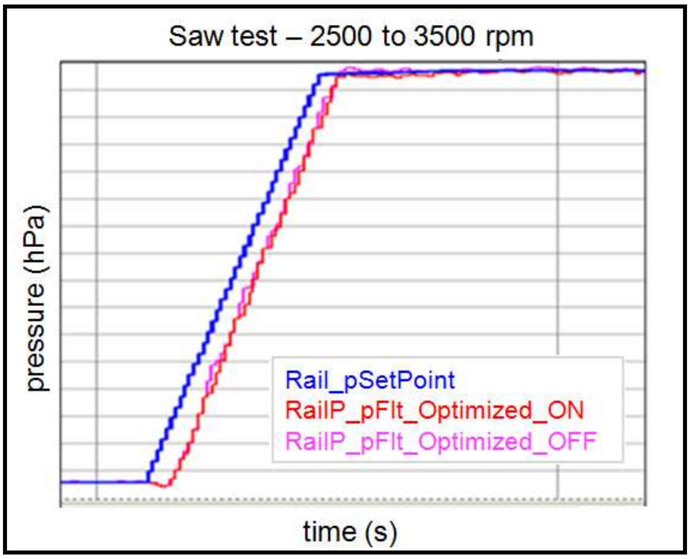


Figure 4.5-20 Saw test drive - equipments ON

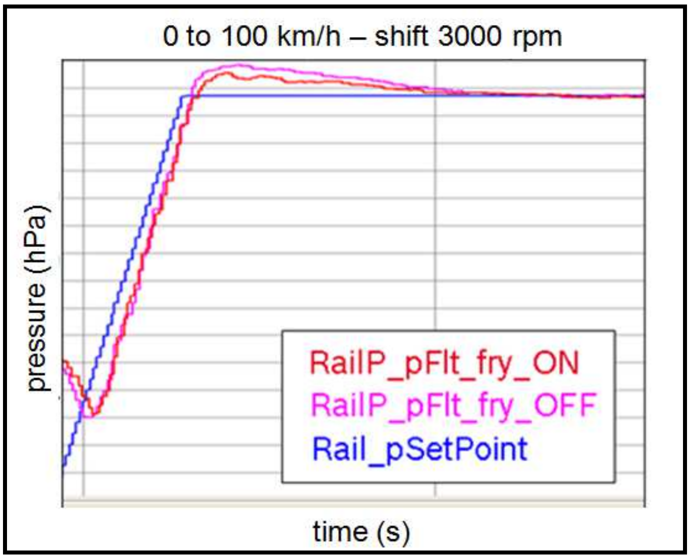


Figure 4.5-21 Shift gear - equipments ON

As expected, with all equipments ON, more injection will be necessary to have energy enough to supply everything, which means more volume overflow; therefore, less pressure will be inside the common rail. This can be clearly observed in the right side of the figure above: the RailP_pFlt with everything ON has 15 bars less in the overshoot than the normal measurement.

To complete all validation of a rail pressure calibration, a list with a lot more test drives needs to be done, however for the goal of this project, the test drives here presented are enough to confirm the quality and the robustness of the large signal calibration. All the other test drives not presented here concerns the quality of small signal, pre-controller and so on.

4.6 RESULTS AND TESTS

As defined in the beginning of this work, the mainly goal of this project is to have a robust method that leads to the best parameters (in the large signal), that can reduce the currently required time and costs. The robustness and the quality achieved were already demonstrated in the chapter before.

The required time in the old method is a completely day of working (8 hours) and needs to be executed completely by an engineer. With the new method, the total required time is four hours and half, which means a reduction of 44% in the time. Besides that, one hour can be executed by a technician, so the engineer time is reduced by 56% and, almost all work can be realized in the office, just the last step, which is Fry optimization, requires going to the test track again.

	Time	Responsible	Where
1) DoE Experiment:	**1 hour**	**Engineer**	**Office**
- Defining correct labels	(0,5 hour)	Engineer	Office
- Creating test plan	(0,5 hour)	Engineer	Office
2) Automated Measurement:	**1,5 hour**		
- Creating Bender Package	(0,5 hour)	Engineer	Office
- Measuring	(1,0 hour)	Technician	Test track
3) Criteria Calculation:	**-**	**Engineer**	**Office**
4) Data Model:	**0,5 hour**	**Engineer**	**Office**
5) Optimization:	**0,5 hour**	**Engineer**	**Office**
6) Validation (Fry and test	**1 hour**	**Engineer**	**Test Track**
Total	**3,5 hours**	**Engineer**	
	1 hour	**Technician**	

Figure 4.6-1 Time results

To test this tool, the method was applied in a new customer with a different system: Ford. The car used was a Ford Fiesta, with MeUn-Actuator, CRS 2.2 with CP4.1 and a DV6D engine.

All method was applied to this vehicle, since creation of a test plan until the entire test drives required. The TipOut study was described already in the chapter 4.4. Here, the validation of the method is just about TipIn. The number of points of the test plan is 250 and number of operation points is just 7 engine speeds. Figure 4.6-2 shows the model quality for the optimization: the negative area has a R^2 of 99% and positive area has a quality of 97%, which means the model is very good and therefore suitable for quantitative predictions.

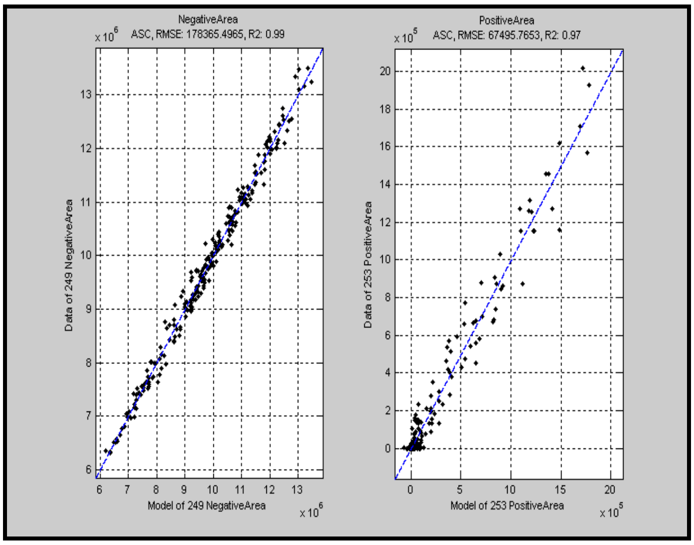


Figure 4.6-2 Model Error - Ford Fiesta

Figure 4.6-3 shows the optimized values for P-gain and I-gain to Ford Fiesta.

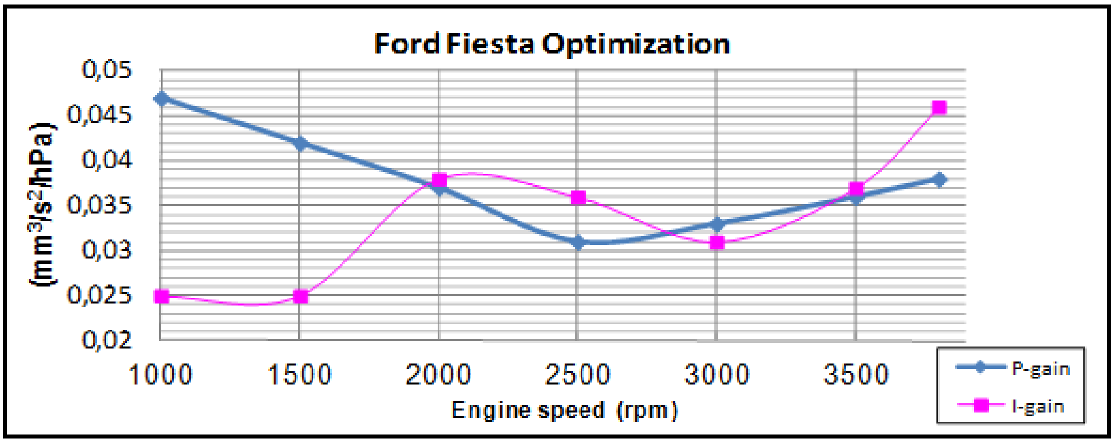


Figure 4.6-3 Optimized values - Fiesta

The validation of these optimization values are presented now. In the left side is the original calibration and in the right side is the optimized one. The next four figures are from a measurement with 1000, 2000, 3000 and 3800 rpm.

With an engine speed of 1000 rpm the new and the old calibration has the same undershoot (-10 bars), but the optimized calibration has a faster behavior.

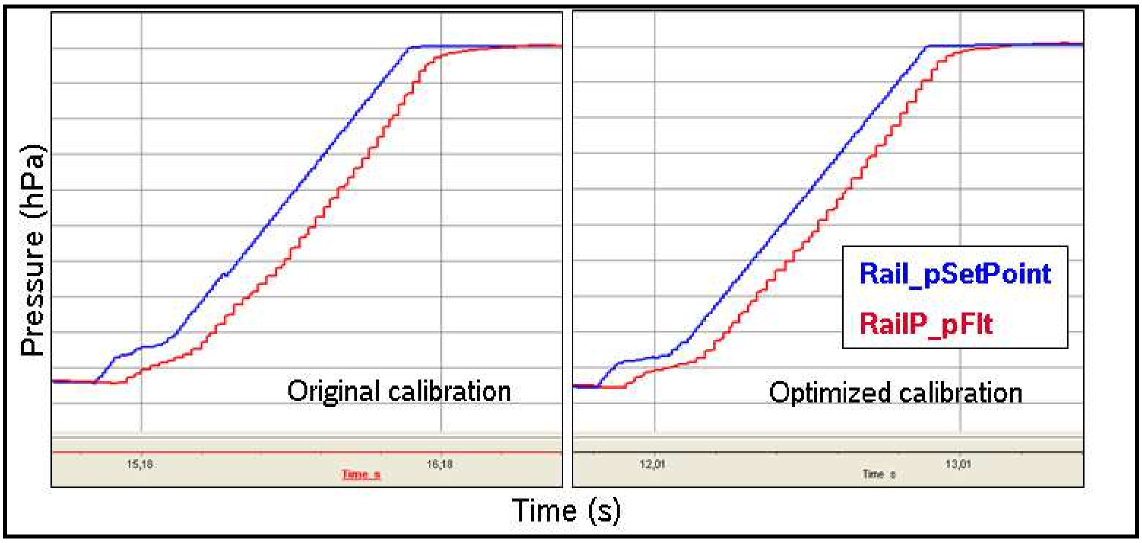


Figure 4.6-4 Ford Fiesta - 1000 rpm

In higher engine speeds the differences of quality between original and optimized calibration start to be bigger. With an engine speed of 2000 rpm, in the original calibration there is a delay of 190 ms and an undershoot of almost 30 bars. In the optimized calibration the delay is reduced to 150 ms and an undershoot of 9 bars, which means a reduction of 20% in the delay and 70% in the undershoot.

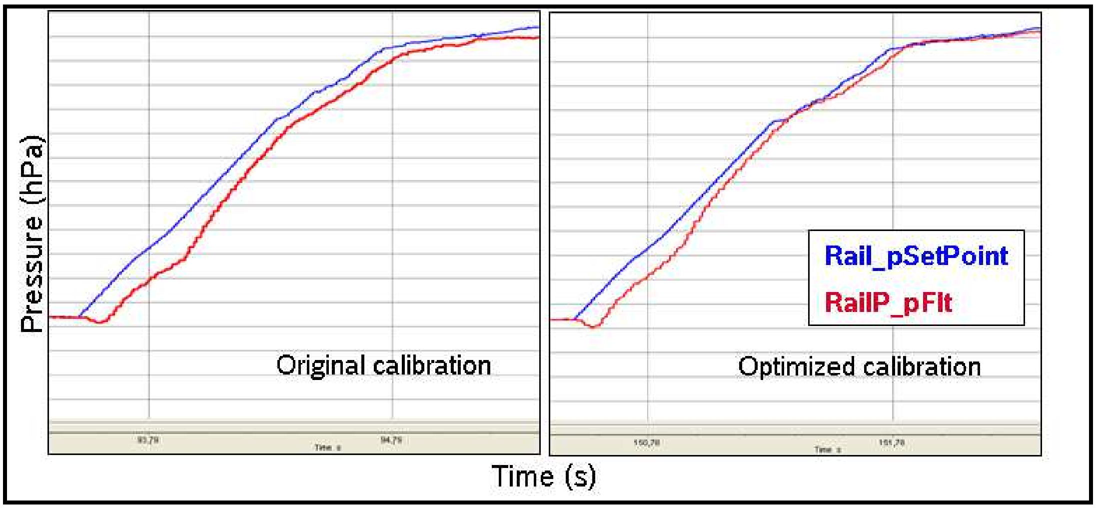


Figure 4.6-5 Ford Fiesta - 2000 rpm

In an engine speed of 3000 rpm the difference is even more, with original calibration the delay is 210 ms and the undershoot is almost 25 bars. With the optimized calibration the behavior is almost perfect: just 3 bars (reduction of 88%) of undershoot and 150 ms (reduction of 30%) of delay.

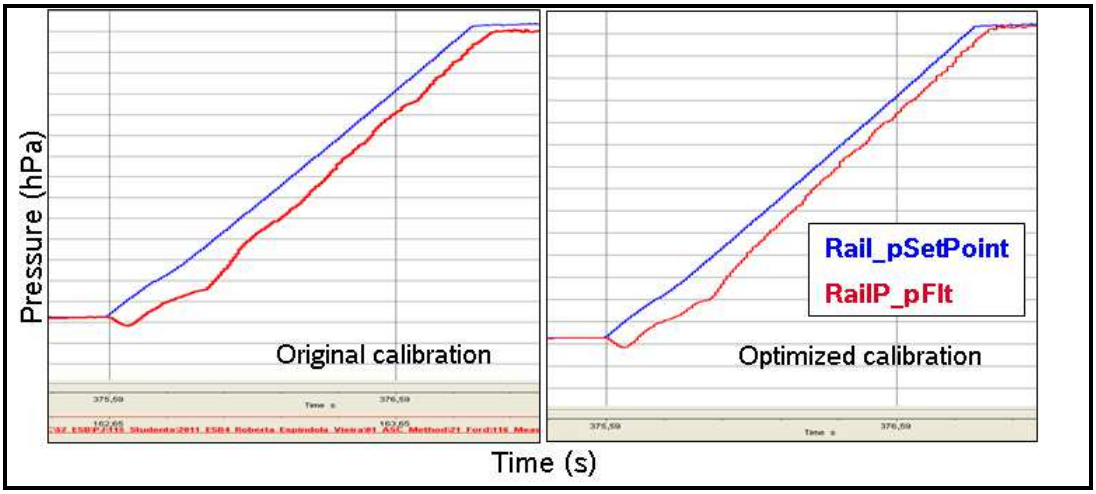


Figure 4.6-6 Ford Fiesta - 3000 rpm

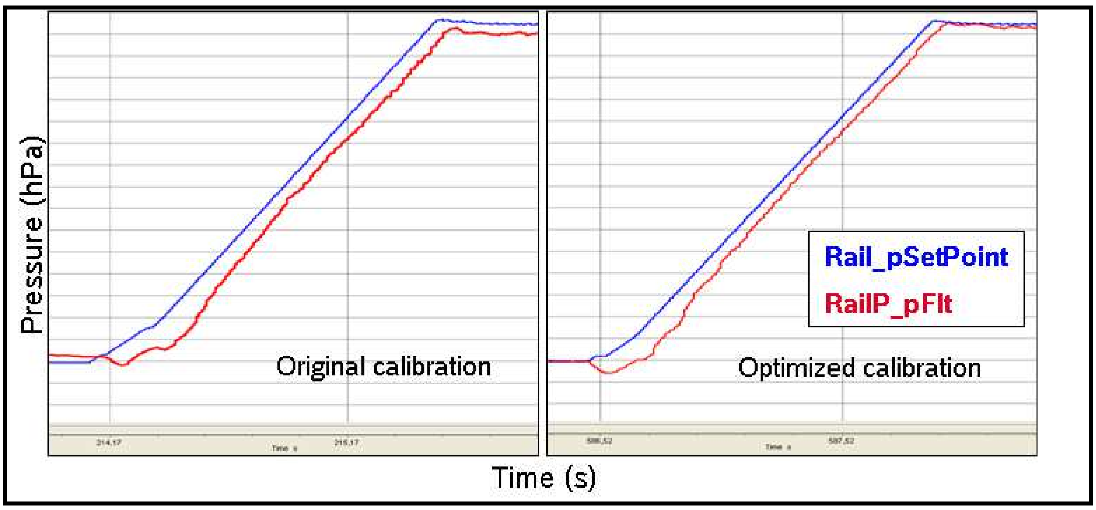


Figure 4.6-7 Ford Fiesta -3800 rpm

It is possible to observe with the original calibration that in higher engine speeds is not possible to reach de set point in the plateau (full load). This behavior probably is due from some mistake in the prior steps when the pre-controller was calibrated. The pre-controller has to guarantee that is possible to reach the plateau.

This brings one more good point of this method: even if in the prior steps some mistake occurs, the model optimization can reduce the problems. This can be observed looking in the curve in Figure 4.6-3. For higher engine speeds is necessary higher values of k_p, otherwise the system would not be able to reach the plateau, however this problem doesn't appear in lower engine speeds, so smaller k_i can be used.

Table 4.6-1 summarizes the behavior of the original and optimized calibration. In comparison to the C4, the delay in this vehicle is always at least two times more. This shows how much the system can change the behavior from one project to another and why the Table 4.3-1 is just a tip, cannot be used in all situations.

Table 4.6-1 Behavior: Original vs. Optimized (Fiesta)

Engine Speed	Calibration					
	Original		Optimized		% of reduction	
	Delay (ms)	Overshoot (bar)	Delay (ms)	Overshoot (bar)	Delay	Overshoot
1000	150	-9.5	127	-10	15.3%	5.3%
1500	149	-12	119	-9	20.1%	25.0%
2000	190	-23.7	150	12	21.1%	49.4%
2500	179	-54.8	140	0	21.8%	100.0%
3000	210	-22.8	150	-3	28.6%	86.8%
3500	155	-23.8	120	4	22.6%	83.2%
3800	179	-22	120	-5	33.0%	77.3%

Figure 4.6-8 shows the results of the test drive from 0 to 100 km/h. Min/max systems are in the same plot and it is possible to see that no big oscillation in this test drive because of this. The calibration used is the optimized result found using Fry.

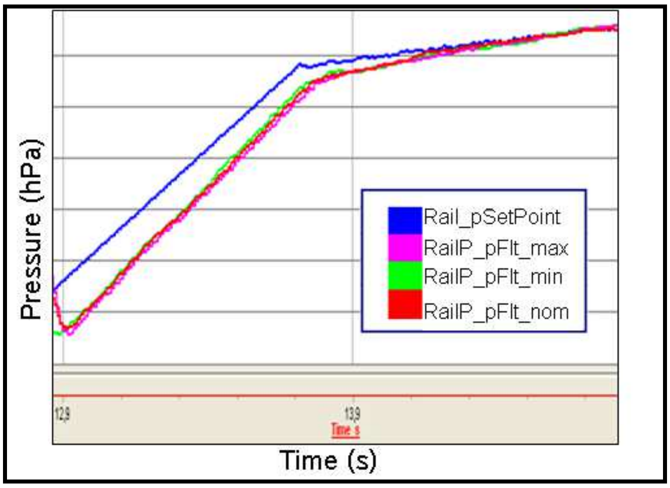


Figure 4.6-8 Test drive 0 to 100 km/h (Fiesta)

Figure 4.6-9 shows a short result of the saw test drive. It is possible to see a variation of 12 bars from min system to max system, which is expected and not critical.

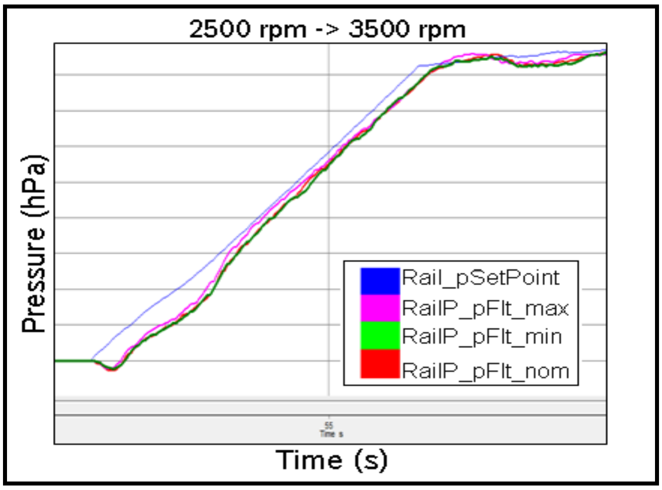


Figure 4.6-9 Saw test full load (Fiesta)

The results with Fiesta confirmed that the method is robust and can find the best values of P and I gain.

Besides that, one of the goals of this project is to define a strategy. After all development, the conclusion is:

- It is necessary at least seven operation points.
- The test plan will have 250 points.
- Each point will be measure just one time.

4.7 NEW STUDIES

After the end of the method development, some new questions were opened. First, until now, the maximum torque was used for giving a step response to the system. When accelerating the vehicle with maximum torque, this will lead to strongest gradients of engine speed, injection mass and setpoint pressure (the worst case of step response). The question is, if a test plan with different torques could bring any advantage. The problem of this point is that the values of PI-gain can only be calibrated in function of the engine speed.

The second question is if the optimization results are also valid for all gears (measurement was done in 2nd gear, check also the other ones).

To answer these questions, a lot of measurements with C4 and Fiesta were made and the conclusion will be shortly described here.

The next two pictures show a TipIn with different torques in an engine speed of 2200 rpm using the C4 and with an engine speed of 3000 rpm using Fiesta.

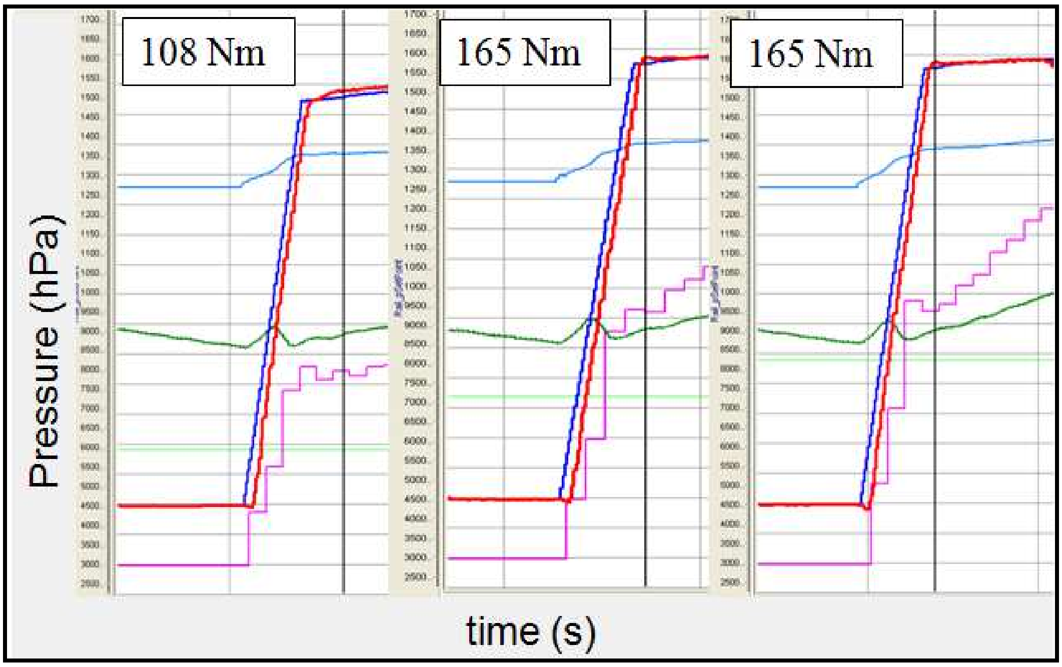


Figure 4.7-1 Different torques - C4

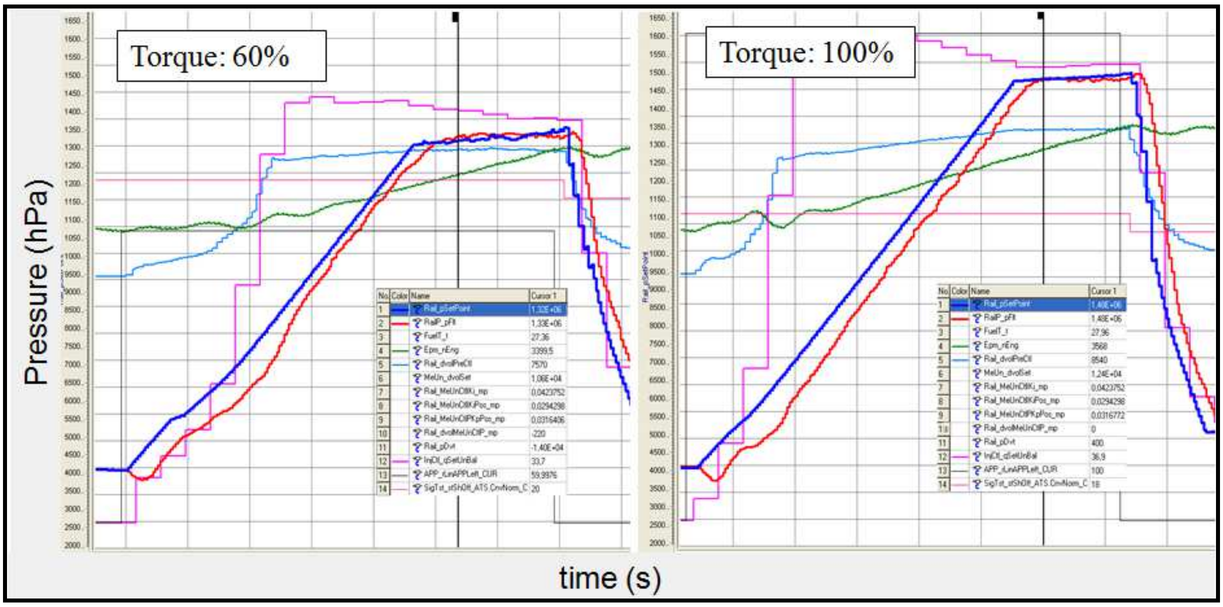


Figure 4.7-2 Different torques – Fiesta

As expected, the variation was not significant. The biggest one was found in Fiesta with 2500 rpm. In this operation point, the maximum torque (100%) has an overshoot of 8 bars and an absolute maximum value of 1430 bars. With a torque of 60%, the overshoot increased to 20 bars and the absolute value is just 1300 bars.

It is possible to see that when the torque is smaller, the overshoot can increase (the I-gain is increasing too fast, leading to an overpressure). However, the step response of the setpoint is smaller with 60% of torque than the step response with 100% of torque. This means that the control behavior with 60% of torque is worst, but not critical (absolute value is more far away from the maximum pressure).

The conclusion for this question is that use a torque with the maximum value gives the guarantee that the critical cases will always have more importance to define the parameters. If in a specific project the customer wants to optimize the behavior in different torques, two solutions can be used:

- If it is known since the beginning that a smaller torque is more important, the test plan should use this torque.
- If in the validation the customer realizes that a different torque has more problems than the maximum torque, the last step can be repeated for this change. That means the Fry optimization can be changed for this different torque. So, the customer will go to the test track and will change the sliders to find the best optimization for this torque.

The second solution is the recommendation for solving bad rail pressure behavior with different torques.

About different gears, the measurements show that the variation is not significant. The second gear leads to the best results also for other gears. If a specific project has problems with different gears, the step using Fry can be used to optimize this specific point. Figure 4.7-3 shows an example of TipIn's with different gears (engine speed of 1100 rpm) using Fiesta.

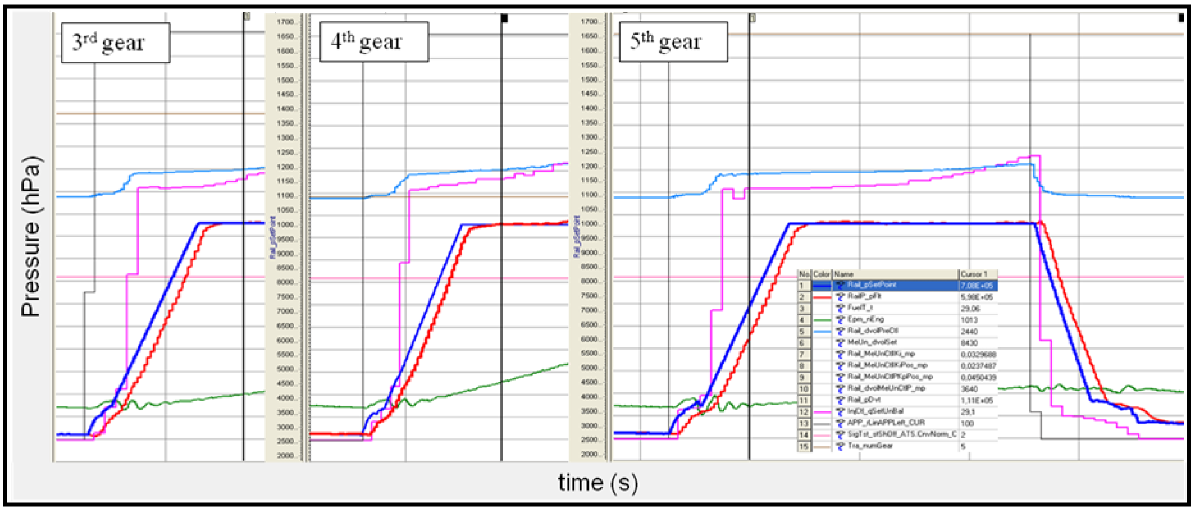


Figure 4.7-3 Different gears - Fiesta - 1100 rpm

4.8 CONSIDERATIONS

As all engeneering projects, some difficulties were observed during the method development. The first one was to understand the automotive mechanical and electronic systems and how to apply the theoretical knowledge on that. During the university, a good background about controlling engeneering was acquired. However, it was missing to have practical applications. The second problem observed was understand how to connect and to use all the tools. And, the hardest step, it was to learn how to make the measurements properly and to analyze all the signals. Each step developed, new problems appeared and new solutions had to be found.

As a student, to develop a project inside a company was a big challenge and a chance to learn things that it is not possible to learn when you are just inside the university. In this project was possible connecting all the theoretical background acquired until today, understanding how the management of project works, leading with deadlines and presenting results to possible customers. Step-by-step all goals defined in the beginning of this project were reached.

A completely method was developed to optimize the parameters of the rail pressure governor - definition how to create a test plan, how to define the borders, which criteria should be used, how to create a model, how to analyze the model and, finally, how to optimize and to validate the results. Attached to this work it is a final description of the method step-by-step.

The first specific goal was developing the best criterion to optimize the controller parameters of the rail pressure control. It was defined, after a lot of measurements analyses and a lot of studies about the system behavior that, the best criteria for a TipIn and a TipOut is positive and negative integral deviation, simple and robust criteria.

The second specific goal was developing an optimal final strategy. The number of operation points necessary to have an acceptable model is seven (seven engine speeds with the maximum torque), as it was possible to see with Fiesta. The optimum number of operation points for a TipIn and TipOut test plan is 250 points. The number of times that each point has to be measured is one time. This conclusion was made based on statistical analysis. If each point was measured more than one time and the mean value of this was given as input to the tool ASCMO, the information about the system spread would be lost.

And, the last specific goal was to integrate the method to another vehicle to see if it's usable. The integration was realized with a second vehicle with the same characteristics as the first one and, with a third vehicle with different system. The results were as expected for all them, which means the optimum calibration was found with the method.

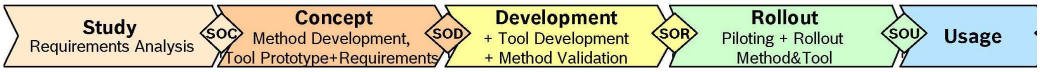
The method is applicable and robust. The time necessary to execute it is fifty percent less than the traditional method, which means a reduction of cost. And the results are even better: faster and less overshoot than traditional calibrations were found.

Until now all the theoretical studies are done, the development of a robust code to calculate criteria is finished, required verifications to confirm the results were realized, application of method with a first customer project was executed and calibration hints were developed.

However, a long way can still be investigated. The functionality of Advanced Simulation of Calibration in the Rail Pressure governor is confirmed and can be used. Now, methods used to calibrate other systems can be checked if cannot be executed using the same methodology. As a new idea, a study about how to include the calibration of the pre-control in this method can be realized.

5 MANAGEMENT

The figure above shows the steps for a completely development of a new project. This work had the goal of finishing the first two steps: study and concept.



A well-planned project is the key to achieve good results. Set milestones, risks, costs and availability of resources are the main factors studied to obtain a good performance. This chapter shows the analysis of these factors.

5.1 SCHEDULE

Set milestones are an important point to finish a project as expected. A detailed schedule with all milestones helps to not lose time and always have a general view how the whole project is developing. Table 5.1-1 shows when each step was developed and how many hours were necessary.

Table 5.1-1 Project schedule

Index	Tasks	Hours	Beginning	End
1	**Study of the problem**			
1.1	*Analyzing market customers*	17	2. Mai. 11	4. Mai. 11
1.2	*Patents verification*	20	5. Mai. 11	10. Mai. 11
1.3	*Researching services' cost*	17	11. Mai. 11	13. Mai. 11
1.4	*Governor Theory*	35	16. Mai. 11	20. Mai. 11
1.5	*Bosch Method*	35	23. Mai. 11	27. Mai. 11
	Total of hours	**124**		
2	**Feasibility' project study**			
2.1	*Analysis of the measurements reproducibility*	40	6. Jun. 11	14. Jun. 11
	Total of hours	**40**		
3	**Design of Experiment Plan**			
3.1	*Definition of border parameters*	35	15. Jun. 11	21. Jun. 11
3.2	*Test plan*	70	22. Jun. 11	11. Jul. 11
	Total of hours	**105**		
4	**Automated Measurement**			
4.1	*Study of EDC17*	35	12. Jul. 11	18. Jul. 11
4.2	*Study of ETK*	17	19. Jul. 11	21. Jul. 11
4.3	*Bender tool*	17	22. Jul. 11	27. Jul. 11
4.4	*Measurements*	35	28. Jul. 11	3. Aug. 11
	Total of hours	**104**		

5	**Criteria Calculation**			
5.1	*Defining which criteria will be programmed*	35	4. Aug. 11	10. Aug. 11
5.2	*Programming criteria*	90	11. Aug. 11	30. Aug. 11
5.3	*Defining model*	35	31. Aug. 11	7. Sep. 11
	Total of hours	**160**		
6	**ASCMO**			
6.1	*Data-model*	35	8. Sep. 11	14. Sep. 11
6.2	*Optimization*	40	15. Sep. 11	22. Sep. 11
6.3	*Visualization*	20	23. Sep. 11	27. Sep. 11
	Total of hours	**95**		
7	**Test Program**			
7.1	*Testing System*	20	28. Sep. 11	30. Sep. 11
7.2	*Validation of results*	35	3. Oct. 11	7. Oct. 11
7.3	*Analyzing results*	35	10. Oct. 11	14. Oct. 11
7.4	*Defining next steps*	35	17. Oct. 11	21. Oct. 11
	Total of hours	**125**		
8	**Seminar presentation**			
8.1	*Developing Thesis*	90	24. Oct. 11	11. Nov. 11
	Total of hours	**90**		
Total of hours in the Project		**843**		

5.2 COST ANALYSIS

The budget total expected for developing this project was 55 000 €. This total includes: price of booking test tracks, personal capacity and licenses. All vehicles used in this project were yielded from Bosch's customers to general studies (not just for this project), so the costs are not included here.

Description/ Year	**2011**
Budget Total	**55 000 €**
Tests	10 000 €
Personal Capacity	44 000 €
Other costs	1 000 €

The tests included booking of test tracks. The test track planned to be used during all project was the Boxberg track (see attached to this project), however, after some measurements, it was possible to conclude that a small test track and cheaper could be used to the basic tests, saving some money. One hour in Boxberg is 120 Euros, since this test track was used five days, each day five hours, just 3000 Euros were spend in test track, instead of 9600 Euros planned in the beginning.

The personal capacity is included one engineer. Since the project was developed with a student and a supervisor, this value was reduced too.

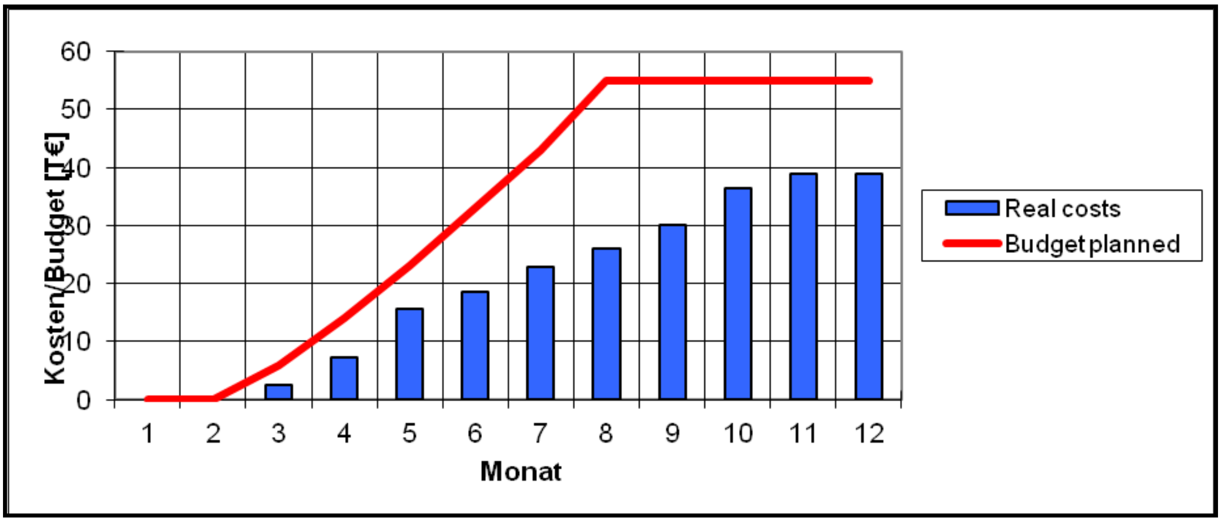


Figure 5.2-1 Real costs

The graphic above shows the budget expected and the real expenditures. The total expenditures were around forty thousand Euros, which means that ten thousand euro was saved in this project.

5.3 RISK ANALYSIS

A risk analysis was done with possible problems that could happen during the development of this project and what should be the action if this really happened. Once determined, these were sorted according to their chance of occurring and impact in the implementation (Table 5.3-1).

Table 5.3-1 Risk analysis

Severity	Effect/Description	Probability	Impact	Action
HIGH	Necessity of a new prerequisite as a request of the company: add a derivate parameter in the controller	60%	0.5	Mitigate: Checking possible influences and study how to define correct borders to this parameter.
MEDIUM	External influence that could lead to mistakes in the measurement (e.g. temperature, problems in the track)	70%	0.3	Living: doing measurement more than one time to observe influences

MEDIUM	Delay in the schedule	20%	0.8	Eliminate: Reorganizing the schedule to finish the project on time
MEDIUM	Necessity of extra time to study the technology and methods to develop the project	20%	0.8	Mitigate: Solving doubts with the supervisor and look for new solutions.
MEDIUM	Difficulty to have the car available in the moment that it is necessary	60%	0.2	Mitigate: Perform other activities (e.g. Report)

Label:

- **Impact:** Value between 0 and 1.
- **Severity:** Severity of the analyzed risk (2 x *Probability* **+** *Impact) / 3*
- **High –** 0,75 to 1
- **Medium –** 0,25 to 0,74
- **Low –** 0 0,24

The problem with external influences was observed during the project and difficulty to have a car available too. Both problems were solved using the actions described in the Table 5.3-1 and no effect in the end was felt.

6 FINAL CONSIDERATIONS

The usage of the Advanced Simulation for Calibration for the rail pressure governor is a new method developed at Bosch during the last year as a study. The idea was trying to develop a new method using ASC to find the best controller parameters that could bring cost and time reduction in relation of the traditional method.

After this year, a lot of research and measurements were done and a new method was developed with all validations necessary to confirm its functionality, everything based on control theory and all theory concepts required to the automotive engineering.

Using the proposed new method, the time reduction was almost 50% of the original time. The behavior of the system was even better than when the traditional method was applied. Besides that, with this method the engineer doesn't need to have a deep knowledge about controller engineering, doesn't need to be present all the time and, even that, it is possible to make adjustments easily if necessary.

A very important conclusion observed was about how the method behaves when something in the prior steps was not done as recommended. In the validation with a Ford Fiesta was observed that the values of the pre-controller were different from the suggested values in the calibration hints and the behavior of the system was really bad in higher engine speeds. However, with this new method, it is possible to reduce the expected problems without giving this information to any tool. This reflects the robustness of the method.

Other important result concerns the reliability of the method. The theoretical background was studied deeply and the results were always connected to the concepts. The traditional method had no clear theoretical background.

The results reached the goal defined in the beginning and it was possible to observe, that the Advanced Simulation for Calibration is a tool with a lot of applicability. New studies in other vehicle systems and new ways to use this technology are already being developed inside Bosch. About the future of this study, some steps still can be done to include some other calibration parameters inside the method. The first idea to be analyzed in the future is the possibility of including the pre-control calculation inside the method.

REFERENCES

BOSCH Group. **Basic of diesel technology**, Esslingen: Steinbeis Transferzentrum, 2008.

BOSCH Group, **Introduction to the CR system**, Esslingen: Steinbeis Transferzentrum, 2008.

BOSCH Group (Pub), **Automotive Handbook**, Germany: Vieweg-Publishing, 2000.

BOSCH Group, **Governor Technique, Basics and Applications**, Esslingen: Steinbeis Transferzentrum, 2008.

Holger Lutz; Wolfgang Wendt. **Handbook of control engineering 2nd revised and extended edition**, Germany: Harri Deutsch Publishing, 1998.

BOSCH Group, **Electronic Control Unit Concept with Components, Calibration Technique**, Esslingen: Steinbeis Transferzentrum, 2008.

BOSCH Group, **High pressure generation, Calibration hints (internal document)**, Stuttgart: Bosch, 2010.

ETAS GmbH: **ASCMO 4.0 user's guide**, Stuttgart: 2010.

BOSCH Group, **Structure of the common rail-software and dynamic fuel quantity corrections**, Esslingen: Steinbeis Transferzentrum, 2008.

BOSCH Group, **Hydraulics, high pressure system**, Esslingen: Steinbeis Transferzentrum, 2008.

BOSCH Group, **Injectors, injection curve**, Esslingen: Steinbeis Transferzentrum, 2008.

How cars work. Available: <http://www.howcarswork.co.uk/modules/articles/article.php?id=7> 20th July 2011 accessed.

How stuff works. Available: < http://www.howstuffworks.com/diesel1.htm> 20th July 2011 accessed.

Universidade Tecnológica Federal do Paraná, Comissão de Normalização de Trabalhos Acadêmicos, **Normas para elaboração de trabalhos acadêmicos**, Curitiba: UTFPR, 2008.

Felipini, Dailton. **Plano de Negócios para empresas na Internet,** Sao Paulo: ABC-COMMERCE, 2004

OGATA, K. **Engenharia de controle moderno** tradução Paulo Álvaro Maya; 4ªed. Prentice Hall, 2003.

APPENDIX

APPENDIX A – METHOD DESCRIPTION

ASC for RPG – step-by-step

Software's necessary:

- INCA V6.2
- ASCMO 4.1 (with plug-in EDOR),
- Bender 4.3.2
- Fry

Required files:

- ASCforRPG_edor.exde
- Borders_draft.xls

Information necessary:

- TipOut optimization included or not*.
- k_{Pcrit} and f_{crit} for all engine speed range.
- Variables name to be used as engine speed (e.g. Epm_nEng), torque (e.g. APP_rLinAPP_CUR), the four parameters of P-gain and I-gain to the large signal and two variables that can be use as ID (e.g.: SigTst_stShOff_ATS.CnvFac_C or SigTst_stShOff_ATS.CnvOfs_C)

Time necessary:

- Desk: 2h05
- Test track: 2h00 (without TipOut) or 2h30 (with TipOut)

Team necessary:

- One engineer: 3h05
- One technician: 1h30

***TipOut:**

If the project has no problem with TipOut it is enough:

1. Use the same value to k_{ppos} and k_{pneg}; and
2. Use the lower border of k_{ineg} ('Borders_draft.xls');

Otherwise, same procedure as TipIn has to be done.

Tip: the increase of the required time for the measurements is half an hour. Even if the first option is chose, a good idea is to make the TipOut measurement anyway. So, if you have any problem later with TipOut you already have the data necessary to analyze.

1st step – ZiNi Method

Required software's: INCA (Car is necessary)

Output: k_{Pcrit} and f_{crit} for the whole engine speed range.

a) Change large window to a higher value, in a way that just small window is active:

Rail_pMeUnCtlIWinNeg_C	-60000	[hPa]
Rail_pMeUnCtlIWinPos_C	60000	[hPa]
Rail_pMeUnCtlPWinNeg_C	-40000	[hPa]
Rail_pMeUnCtlPWinPos_C	40000	[hPa]

→

Rail_pMeUnCtlIWinNeg_C	-3276800	[hPa]
Rail_pMeUnCtlIWinPos_C	3276700	[hPa]
Rail_pMeUnCtlPWinNeg_C	-3276800	[hPa]
Rail_pMeUnCtlPWinPos_C	3276700	[hPa]

b) k_i and k_d must be zero.

c) Change the engine speed to the one desired. The label HLSDem_nSetPLoWrm_C defines the engine speed, the label Rail_pSetPoint_C stabilizes the value of the setpoint (otherwise, too much oscillation is observed).

HLSDem_nSetPLoWrm_C	1100.000	[rpm]
Rail_pSetPoint_C	390000	[hPa]
Rail_swtPSetPoint_C	1.000	[-]

d) Increment k_p until stability limit (F7), wait 3 seconds, and then use F6 to bring the system slowly back to a steady state. Wait 10 seconds and repeat again (three times). The measurement will look like Figure 1. With this measurement, it is possible calculate k_{Pcrit} and T_{crit} (Figure 2). Counting ten periods and divide by ten to calculate T_{crit}.

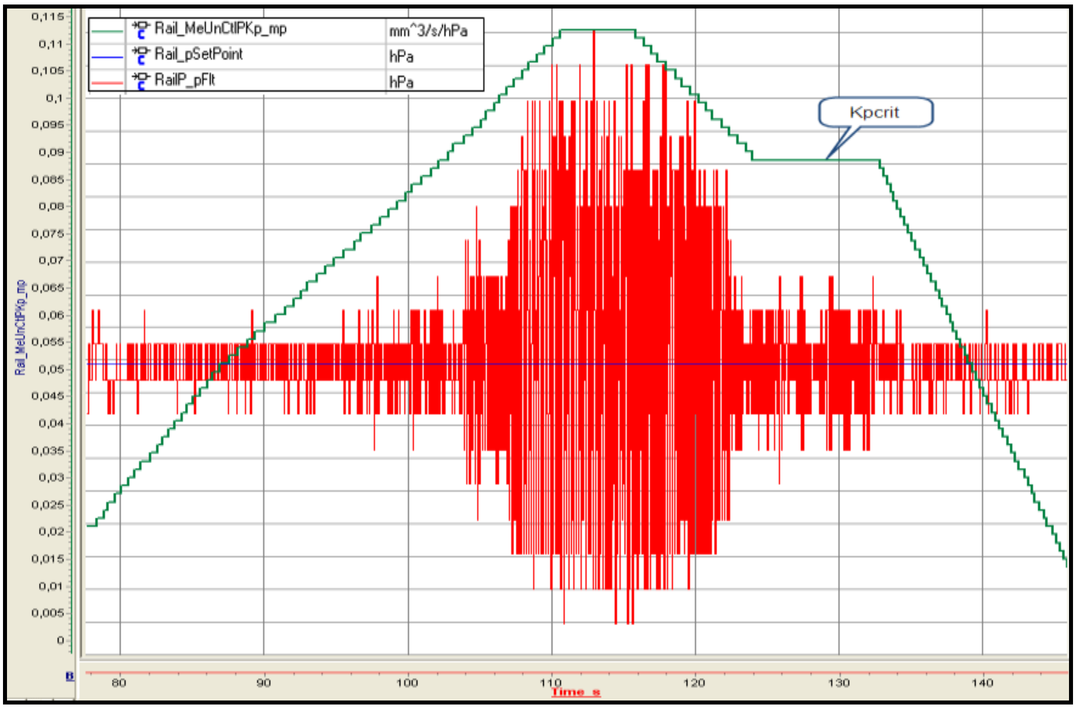


Figure 1 - k_{Pcrit}

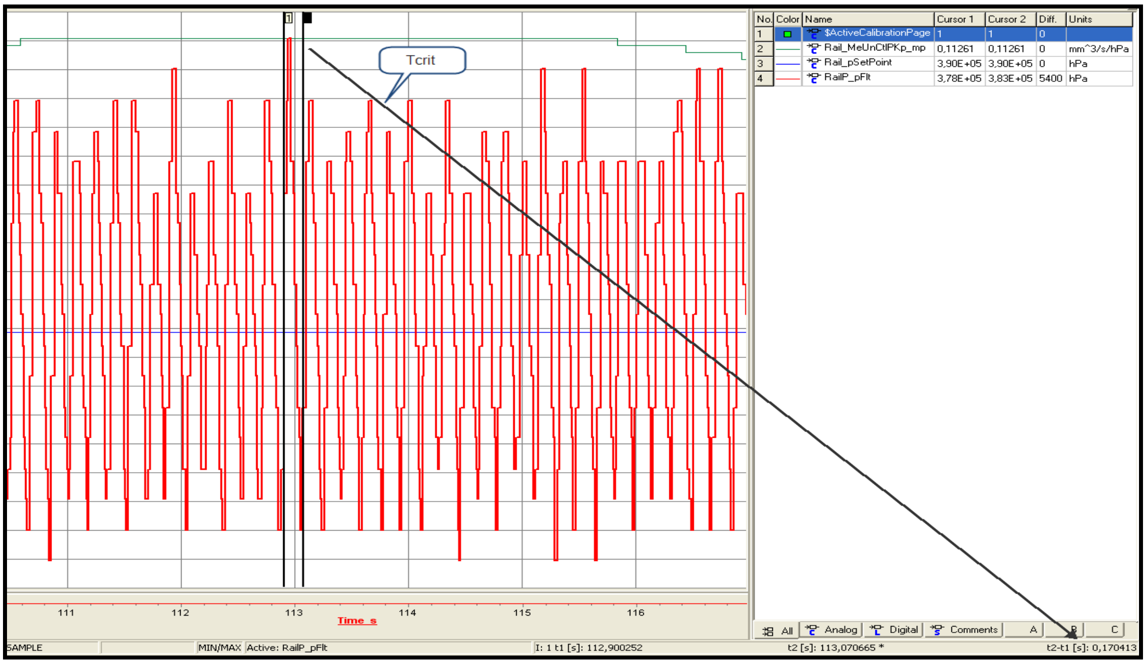


Figure 2 T_{crit}

2[nd] step – Design of Experiment (30 minutes)

Required software's: ASCMO 4.1 – plug-in EDOR

Required files: ASCforRPG_edor.exde and Borders_draft.xls

Input: k_{Pcrit} and f_{crit}

Output: Test Plan

a) Adding k_{Pcrit}, f_{crit} and engine speed to **Borders_draft.xls** and generating DCM file.

EngSpeed	250	500	700	900	1000	1250	1400	1750	1900	2400
kpcrit	0.052	0.053	0.054	0.055	0.057	0.06	0.061	0.062	0.062	0.062
fcrit	5.6	5.6	5.6	5.6	5.6	6.2	6.55	7.15	7.4	7.9
Rail_MeUnCtlPKpPos_upper_CUR	0.0381	0.0388	0.0396	0.0403	0.0418	0.0440	0.0447	0.0454	0.0454	0.0454
Rail_MeUnCtlPKpPos_lower_CUR	0.0260	0.0265	0.0270	0.0275	0.0285	0.0300	0.0305	0.0310	0.0310	0.0310
Rail_MeUnCtlIKiPos_upper_CUR	0.0366	0.0373	0.0380	0.0387	0.0401	0.0467	0.0502	0.0557	0.0576	0.0615
Rail_MeUnCtlIKiPos_lower_CUR	0.0201	0.0205	0.0209	0.0213	0.0220	0.0257	0.0276	0.0306	0.0317	0.0338

Create DCM

Minimum value of Rail_MeUnCtlPKpPos:	0.0260
Minimum value of Rail_MeUnCtlPKpNeg:	0.0260
Maximum value of Rail_MeUnCtlPKpPos:	0.0454
Maximum value of Rail_MeUnCtlPKpNeg:	0.0454
Minimum value of Rail_MeUnCtlIKiPos:	0.0201
Minimum value of Rail_MeUnCtlIKiNeg:	0.0201
Maximum value of Rail_MeUnCtlIKiPos:	0.0716
Maximum value of Rail_MeUnCtlIKiNeg:	0.0716

Figure 3 Borders_draft

b) Opening ASCforRPG_edor.exde

1. General settings**:** change input names to the one defined in your project and change minimum and maximum values of variables (in the **Borders_draft.xls** you can see the min/max value of $k_{ppos/neg}$ and $k_{ipos/neg}$).

   Tip: it's not possible use a constant value in torque yet, it'll be necessary change this in the excel file in the end. Besides that, if your variable is a CUR or a MAP, take care about what region of this CUR or MAP you want to modify (See *5.4"How can I modify only a special region of a map or curve?"* in Bender Help)
2. **Constraints:** click in each constraint (4 in total) and import the DCM file to upper and lower border.
3. **Cluster points:** define engine speeds for the grid (at least seven values)
4. **Export:** export to an excel file (if TipOut include, export two tables).

c) Opening the excel file and change the column of torque to the constant value required and change the header Experiment Id to a variable from your project that can be used as ID (e.g.: *SigTst_stShOff_ATS.CnvFac_C* or *SigTst_stShOff_ATS.CnvOfs_C*). TipIn and TipOut test plan must have different labels ID.

3rd Step – Creating Bender package (30 min)

Required software's: Bender 4.3.2

Required files: Test plan and DCM file (if necessary to initialize INCA configuration)

Input: Test Plan and DCM file (if necessary)

Output: Bender package

a) Opening Bender and creating a new project (Engine: ASD; TipIn or TipIn and TipOut).
b) Adding a new task; adding TipIn test plan (and TipOut); and DCM file to initial configuration (if necessary).
c) Changing setting according to your project (idle engine speed and engine speed limit).

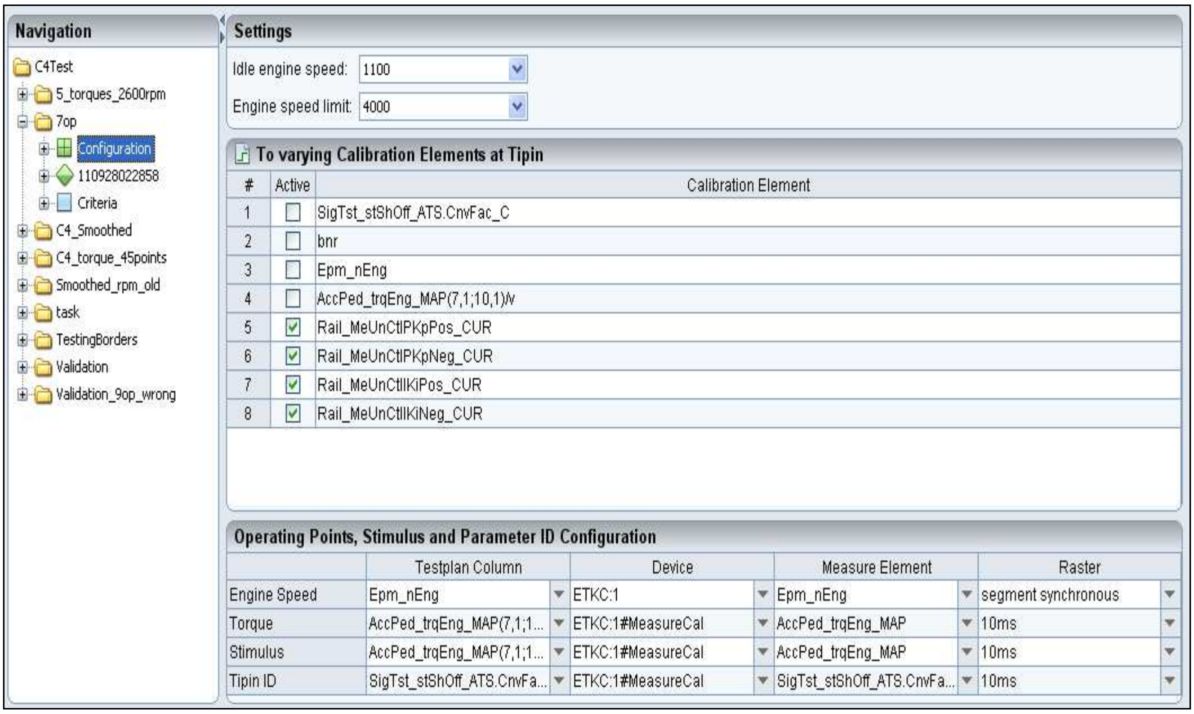


Figure 4 Bender Package

d) Selecting the four parameters to varying calibration elements at TipIn (and TipOut).
e) Connecting to INCA and do the 'Configuration of operation points, stimulus and parameters ID'.
f) Adding a measurement session.

4th Step – Measurement (1,5 hour)

Required software's: Bender 4.3.2

Required files: Bender package

Vehicle is necessary in this step and a properly test track!

Input: Bender package

Output: measurements (*.dat)

a) Opening bender package created in the last step.

b) Going to test track, pressing play and starting the measurement. **Tip:** The accelerator pedal must be completed pressed during all measurement.

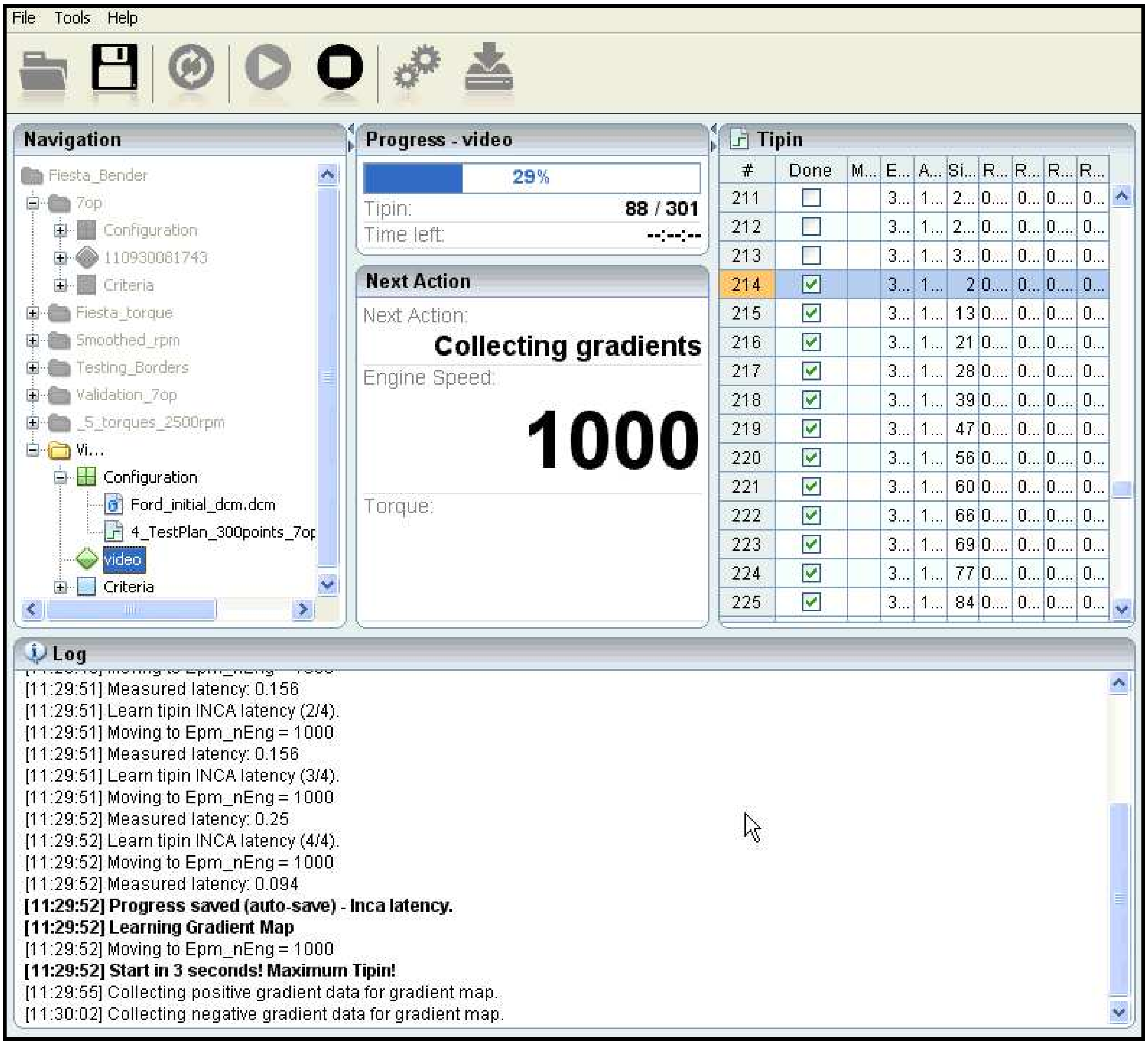


Figure 5 Automated Measurements

5th Step – Criteria calculation (5 min)

Required software's: Bender 4.3.2

Required files: Bender package

Input: Bender package

Output: excel file with criteria calculation

a) Opening bender package used in the last step.

b) In the navigation Menu, go to criteria; select the TipIn criteria (Negative and Positive area).

c) Select the correct measurement session and the correct data file in 'Criteria → Source'.

d) Selecting signal allocation and defining ECU signals in 'Criteria → Signal Allocation'.

e) Pressing the button 'Start criteria calculation' and saving the results in a excel file.

f) Repeat procedure to TipOut.

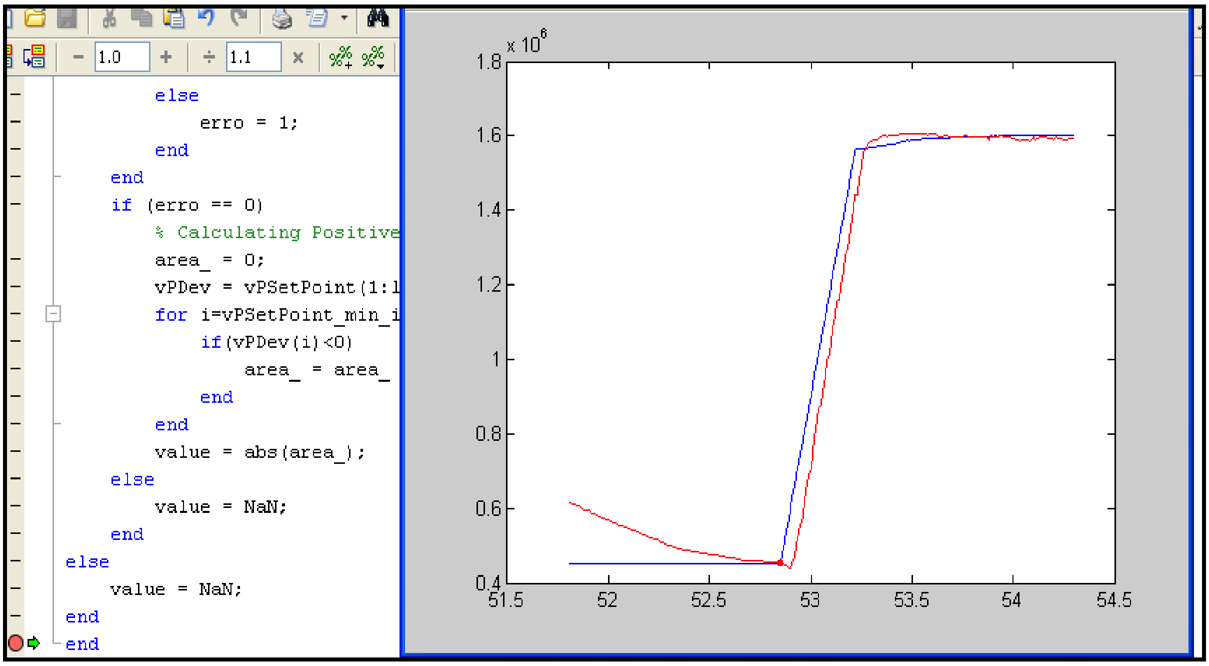


Figure 6 Criteria Calculation

6th Step: ASCMO (1 hour)

Required software's: ASCMO V4.1

Required files: Criteria calculation file.

Input: Criteria calculation file

Output: Pareto front (7 to 9 excel files)

All these steps need to be repeated to TipOut model.

a) Opening ASCMO and click in Start import. Select Criteria calculation excel file created in last step.

b) Selecting engine speed, kppos and kipos as input; positive and negative area as output.

c) Go to 'Model → Training → ASC (preferred)'.

Tip: go to View and select "Show model sigma" and "Show adjusted training data".

d) Go to 'Model → Error (Leave-one-out) → Measure vs. Predicted' and delete outliers if necessary.

Tip: R^2 (Coefficient of determination) should be higher than 90% to be suitable for quantitative predictions.

----

e) With the right button in the engine speed, set discrete values (type the operations points that you want, for example, [1100, 1400, 1700, 2000, 2300, 2600, 2900, 3200, and 3500]).

f) Move the cursor to the lower engine speed and deactivate the engine speed input.

g) Go to 'Optimization→ constraints → input bounds' and define the borders of kp and ki for this engine speed.

h) Go to 'Optimization → Multi-criteria'.

If TipIn, select as output

- Negative area and the criterion Minimize; and
- Positive Area and the criterion Target with a value of zero. Click in optimize.

If TipOut:

- Negative area and the criterion Target with a value of zero; and
- Positive Area and the criterion Minimize. Click in optimize.

i) In the plotted Pareto, go to Extra and export result.

Tip: named the file with the engine speed chooses. (e.g.: Pareto_2300rpm)

j) Adding column k_{pneg} and k_{ineg} to the excel files with the same values as k_{ppos} and k_{ipos}.

k) Repeating item ‘f’ to ‘j' for all engine speeds necessary.

7th step: Fry – validation (1 hour)

Required software’s: Fry 1.1.0

Required files: Pareto fronts (7 to 9 excel files) and an a2l file.

Vehicle is necessary in this step and a properly test track!

Input: Pareto fronts (7 to 9 excel files)

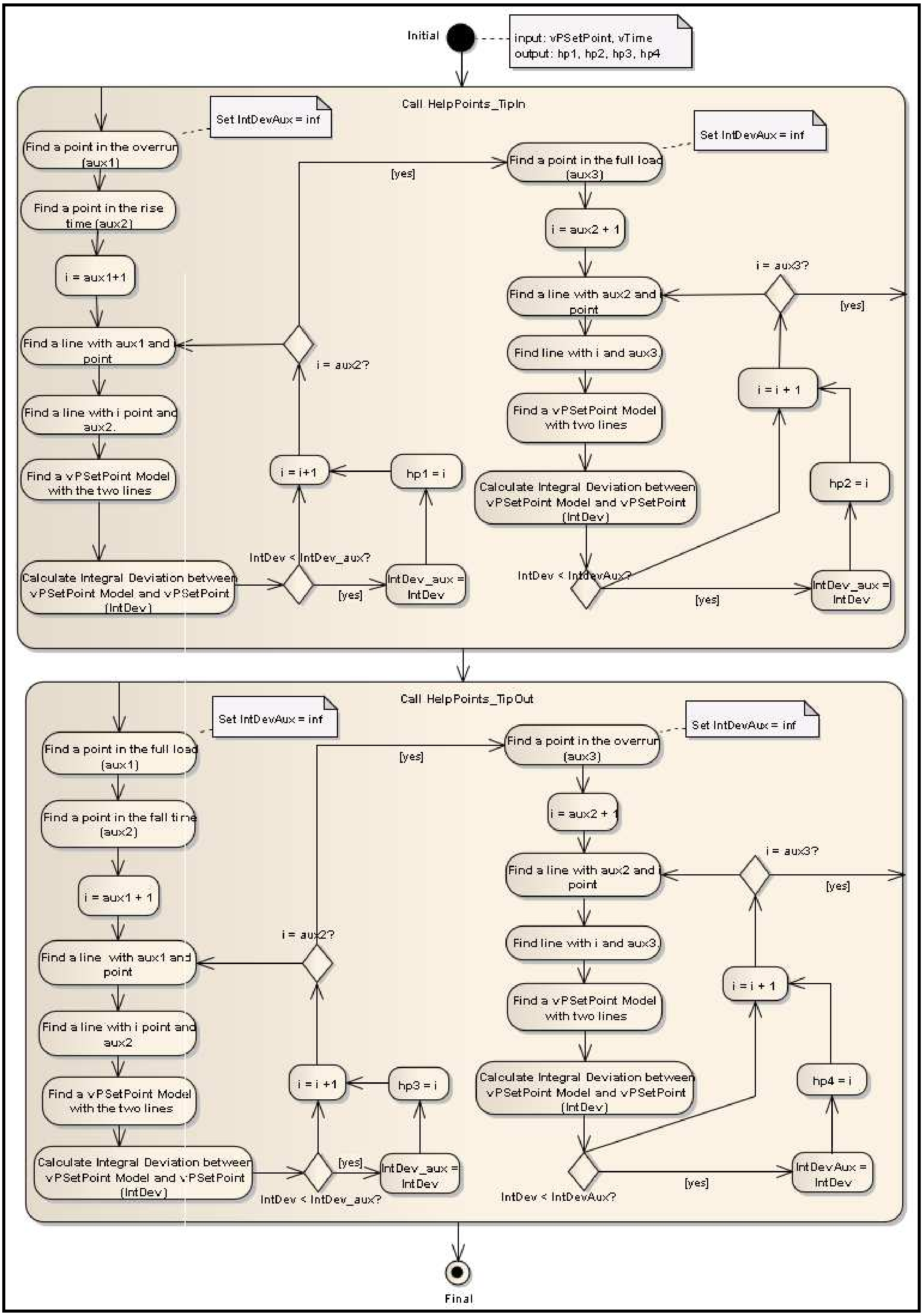
Output: Optimized values.

The same for TipIn and TipOut, if both model were created. Otherwise, change the values of kpneg and kineg according the results of this step before executing step ‘f’.

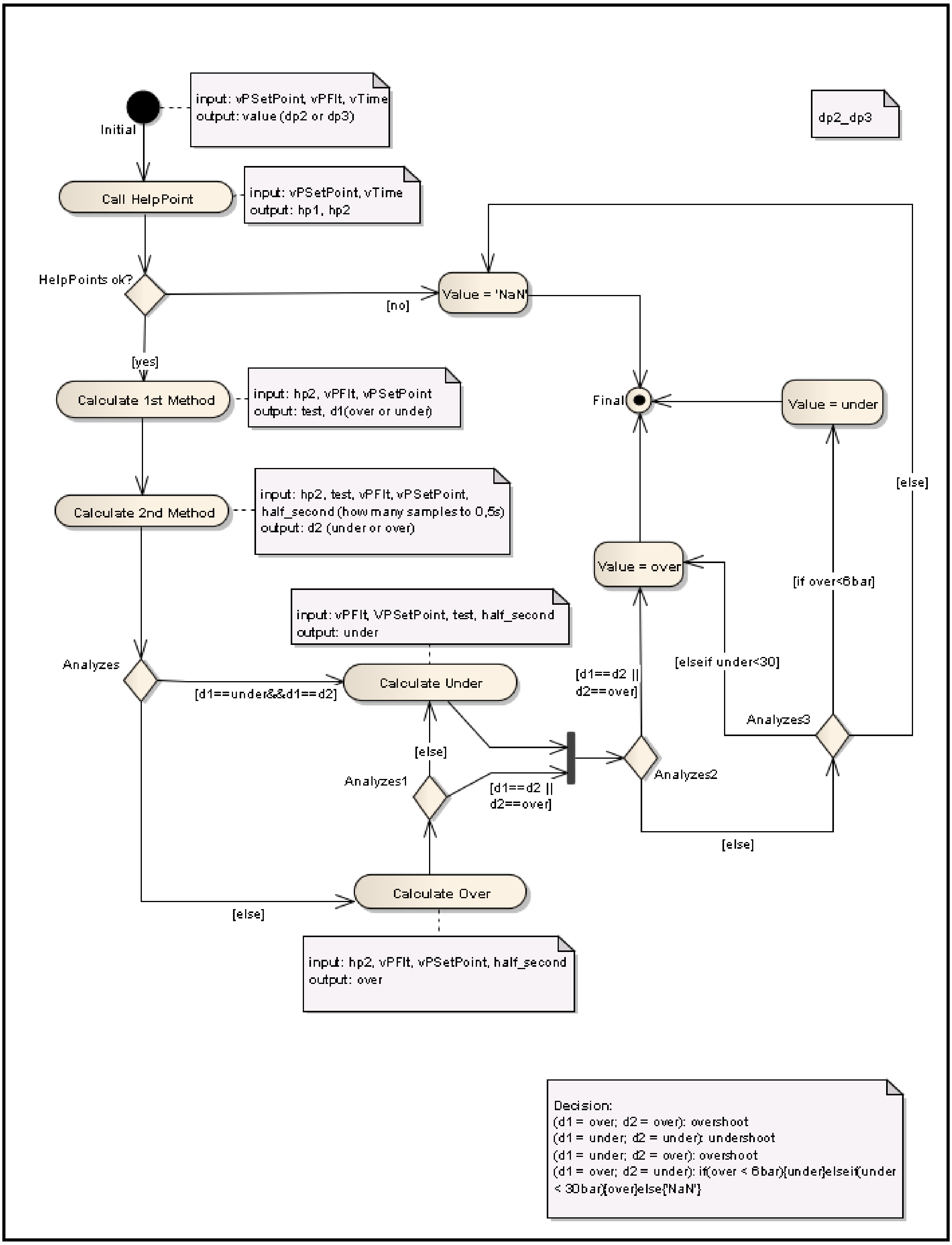
a) Go to ‘Show global settings’ and add the a2l file for your project.
b) Go to ‘Show settings of current mode’ and add all Pareto excel files.
c) Select column types and support point.
d) Go to ‘Show calibration slides’ and put all sliders in the minimum value of positive area.
e) Go to the test track, connect to INCA and press Play. For each engine speed execute TipIn’s and change the slider until you find the best results. *Tip: in the end of the calibration, save the results found in a DCM file in INCA.*
f) Execute Test Drives necessary

APPENDIX B - UML

HELP POINTS



OVERSHOOT CALCULATION

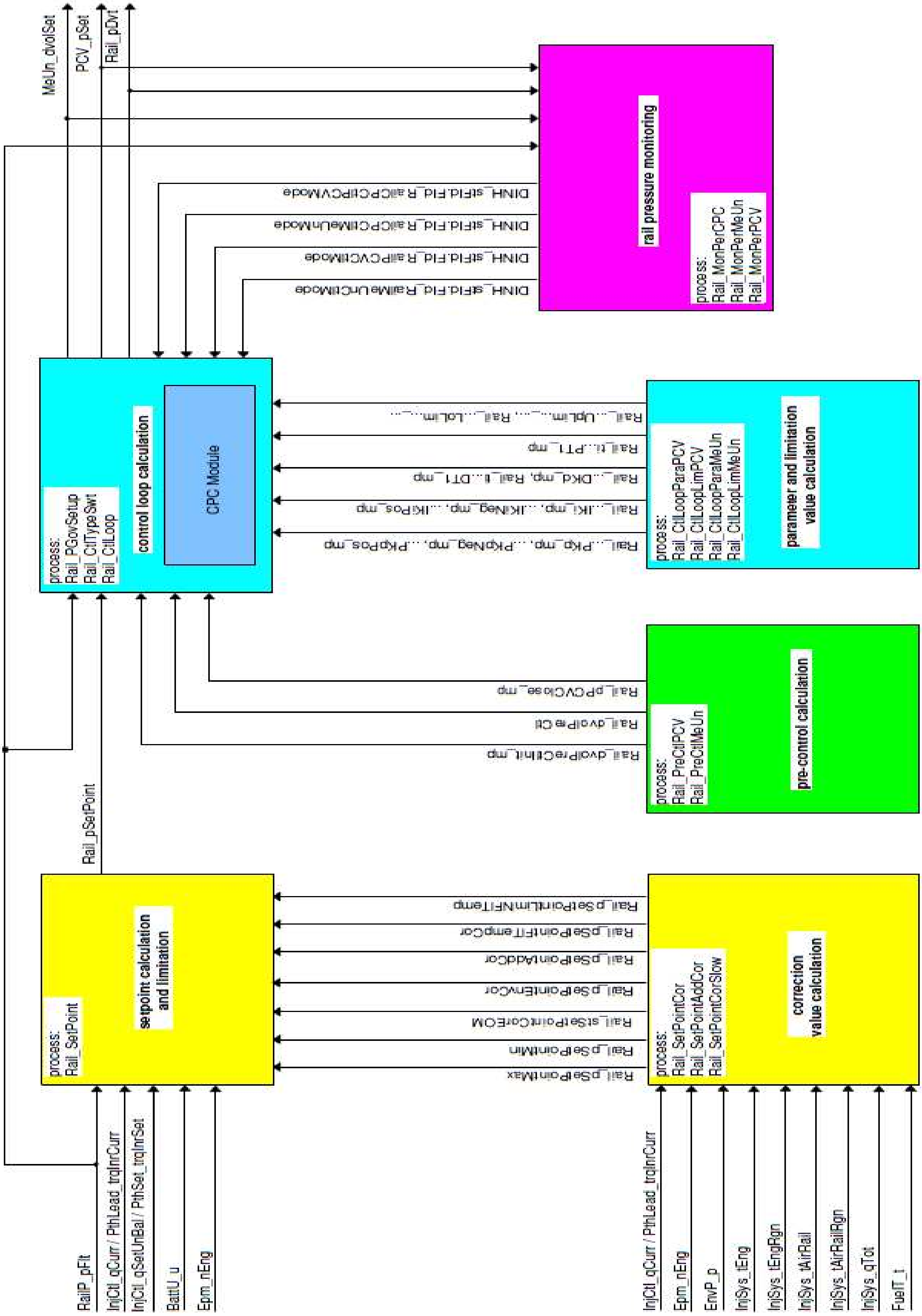


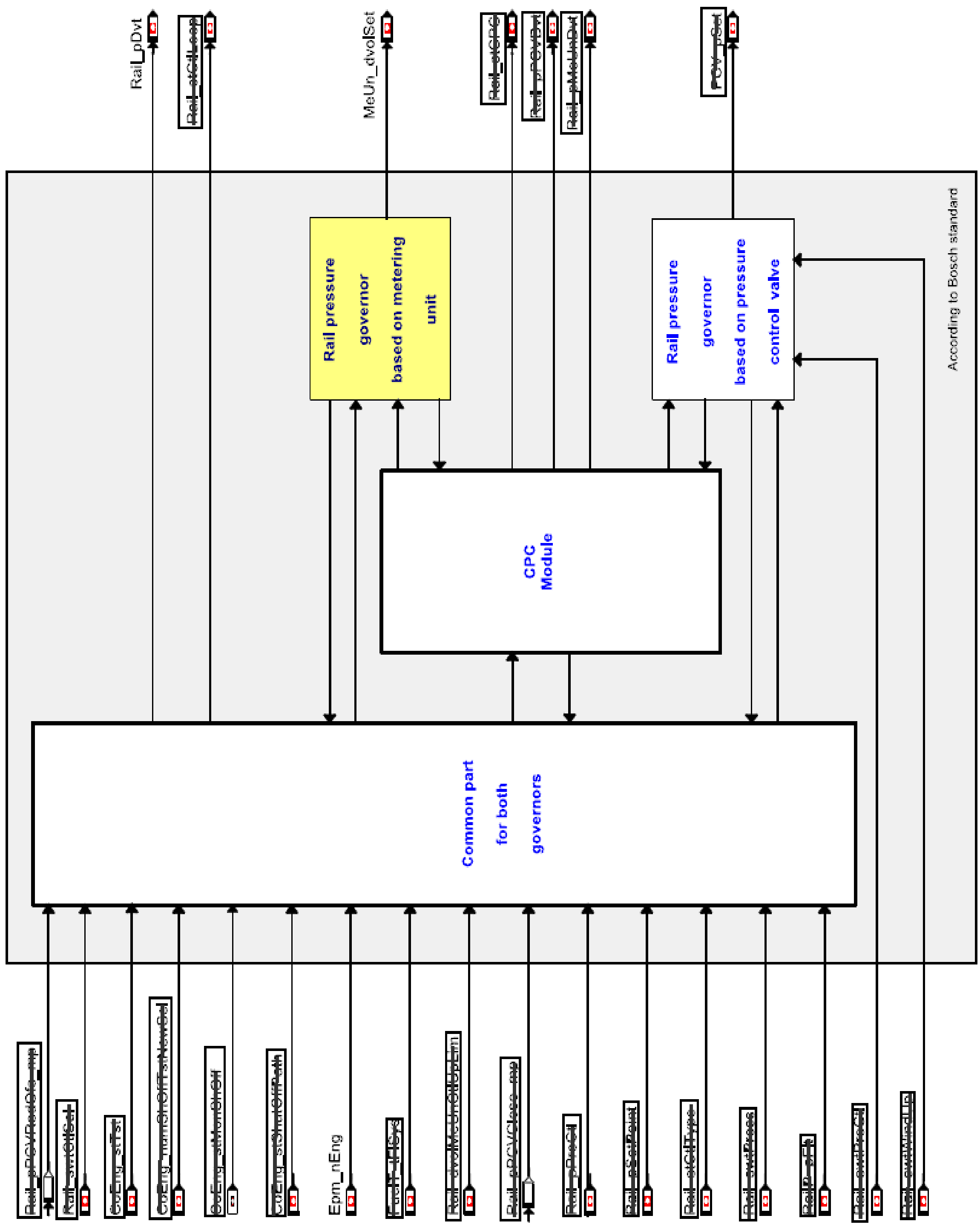
APPENDIX C– Matlab Codes

```
function value = calcPositiveAreaOvershoot(jBenderOptions, jEngineOptions,
jCriterion, MdfDataPre, MdfDataSegmentationSlice, axPlot)

% Data preparation
vTime1 = MdfDataPre.time;
vPFlt1 = MdfDataPre.RailP_pFlt.data;
vPSetPoint1 = MdfDataPre.Rail_pSetPoint.data;

vTime2 = MdfDataSegmentationSlice.time;
vPFlt2 = MdfDataSegmentationSlice.RailP_pFlt.data;
vPSetPoint2 = MdfDataSegmentationSlice.Rail_pSetPoint.data;

vTime = [vTime1; vTime2];
vPFlt = [vPFlt1; vPFlt2];
vPSetPoint = [vPSetPoint1; vPSetPoint2];
plot_ = 0;

% Finding help point 1
[vPSetPoint_min_index erro] = HelpPoints_1(vPSetPoint, vTime);

if (erro == 0)
    % Plottings
    if plot_ == 1
        figure
        plot(vTime,vPSetPoint,'b')
        hold on
        plot(vTime,vPFlt,'r')
    end
    % Calculating Positive Area
    area_ = 0;
    vPDev = vPSetPoint(1:length(vPSetPoint)) - vPFlt(1:length(vPSetPoint));
    for i=vPSetPoint_min_index:length(vPDev)
        if(vPDev(i)<0)
            area_ = area_ + vPDev(i);
        end
    end
    value = abs(area_);
else
    value = NaN;
end
end
---------------------------------------------------------------------------
function [HelpPoint1 erro] = HelpPoints_1(vPSetPoint, vTime)

%calculate just hp1!
index_helppoint1 = 0;
vPSetPoint_max_index = 0;
%first aux point
vPSetPoint_min_index = 5;
vPSetPoint_min = max(vPSetPoint)-vPSetPoint(vPSetPoint_min_index);
for i =2:length(vTime)
    % The second point will be the one that reach 40% of vPSetPoint_min
    if
(vPSetPoint(i)>(vPSetPoint_min*0.2+vPSetPoint(vPSetPoint_min_index)))
        vPSetPoint_max_index = i;
        break
    end
end
```

```
if vPSetPoint_max_index >0
    dev = 10000000000000;
    y1 = vPSetPoint(vPSetPoint_min_index);
    x1 = vTime(vPSetPoint_min_index);
    y2 = vPSetPoint(vPSetPoint_max_index);
    x2 = vTime(vPSetPoint_max_index);

    for i=(vPSetPoint_min_index+1):(vPSetPoint_max_index-2)

        a1= (y1-vPSetPoint(i))/(x1-vTime(i));
        a2= (y2-vPSetPoint(i))/(x2-vTime(i));
        b1= y1-a1*x1;
        b2= y2-a2*x2;

        vRegr1 = a1.*vTime(vPSetPoint_min_index:i) + b1;
        vRegr2 = a2.*vTime((i+1:vPSetPoint_max_index)) + b2;
        vPSetPoint_model = [vRegr1; vRegr2];
        aux_dev = sum(abs(vPSetPoint_model-
vPSetPoint(vPSetPoint_min_index:vPSetPoint_max_index)));
        if (aux_dev < dev)
            %first help point
            dev = aux_dev;
            index_helppoint1 = i;
        end
    end
else
    index_helppoint1 = 'NaN';
end
HelpPoint1 = index_helppoint1;
if strcmp(HelpPoint1,'NaN')
    erro = 1;
elseif (HelpPoint1>0)
    erro = 0;
else
    erro = 1;
end
end
----------------------------------------------------------------------------
```

APPENDIX D- Rail pressure: PHYSICAL OVERVIEW



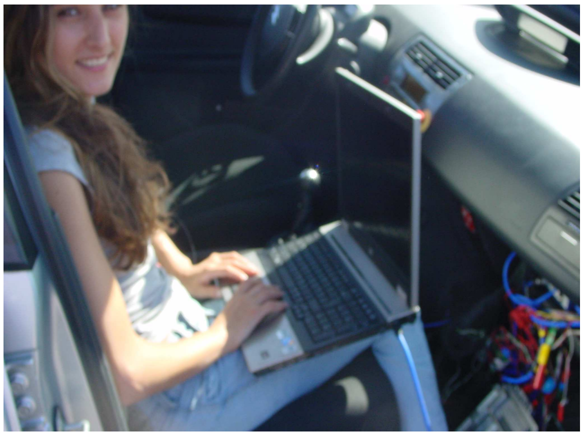


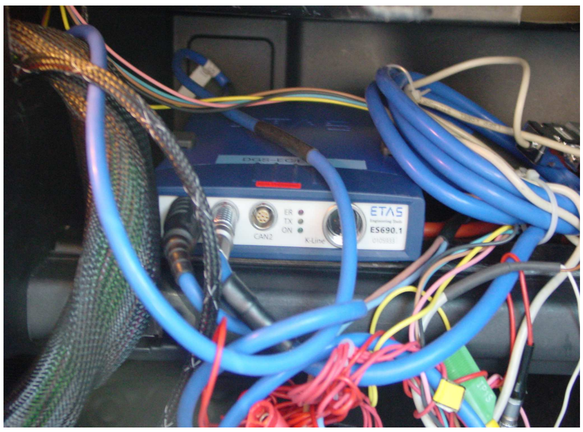
Rail_pDvt
MeUn_dvolSet
Rail pressure governor based on metering unit
CPC Module
Rail pressure governor based on pressure control valve
Common part for both governors
According to Bosch standard
Epm_nEng

APPENDIX E – PHOTOS

BOXBERG

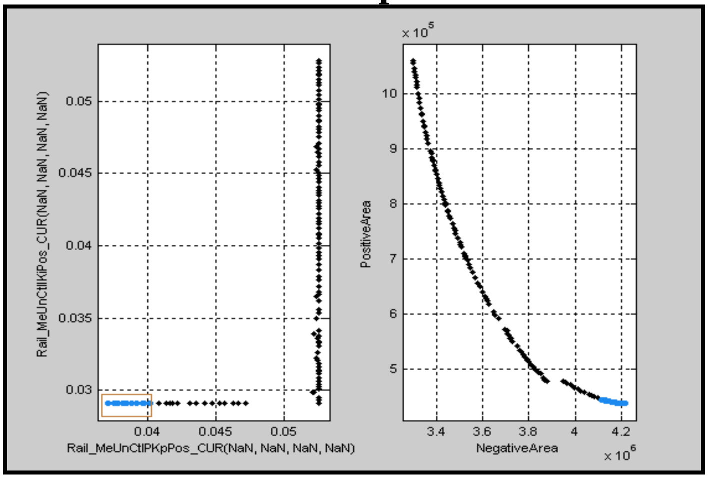
CITRÖEN C4



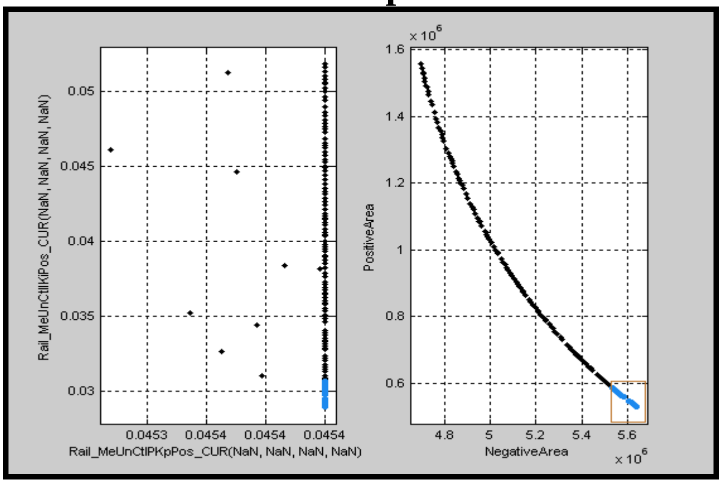


APPENDIX F – C4 Nº 1: OPTIMIZATION

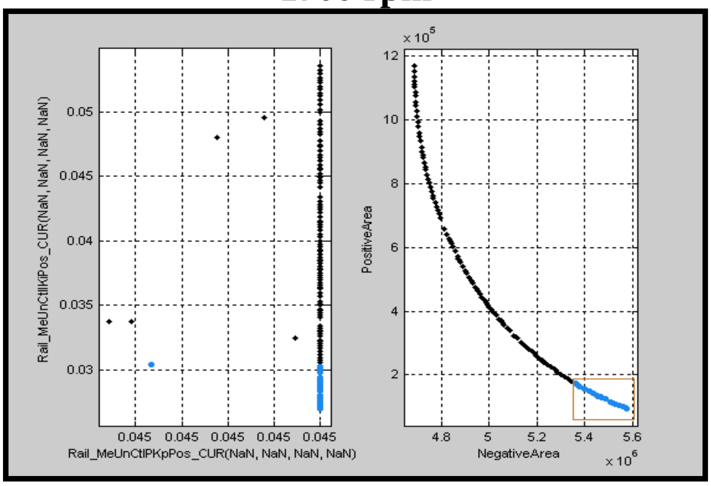
1100 rpm



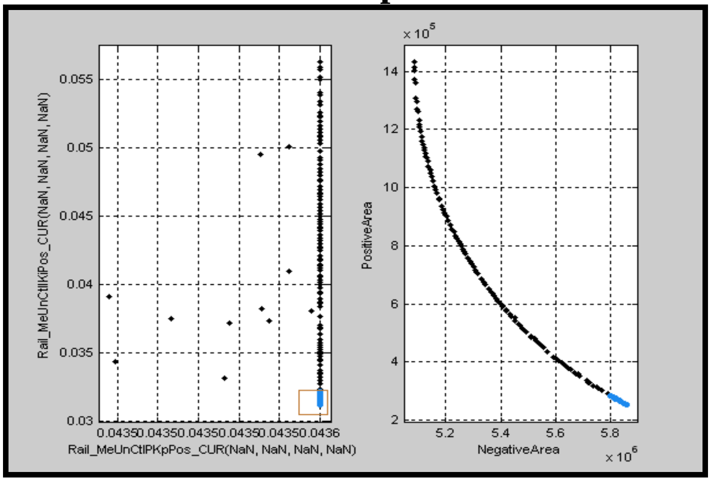
1300 rpm



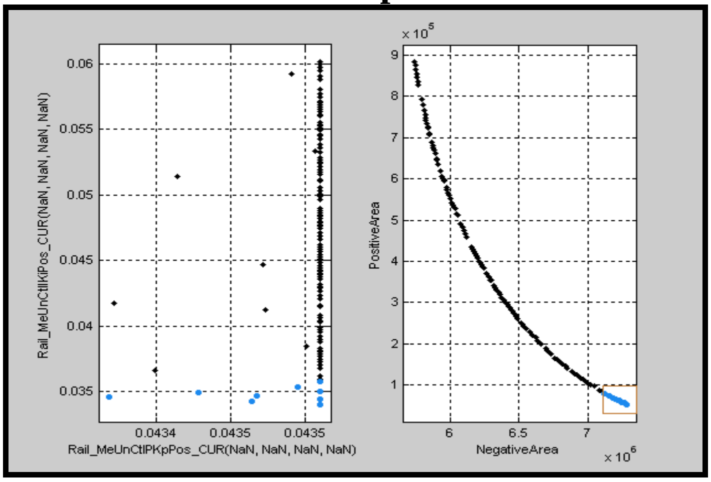
1700 rpm



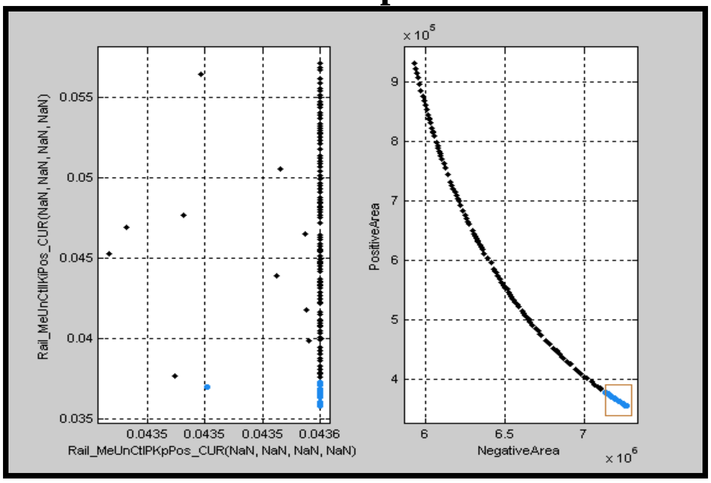
2000 rpm



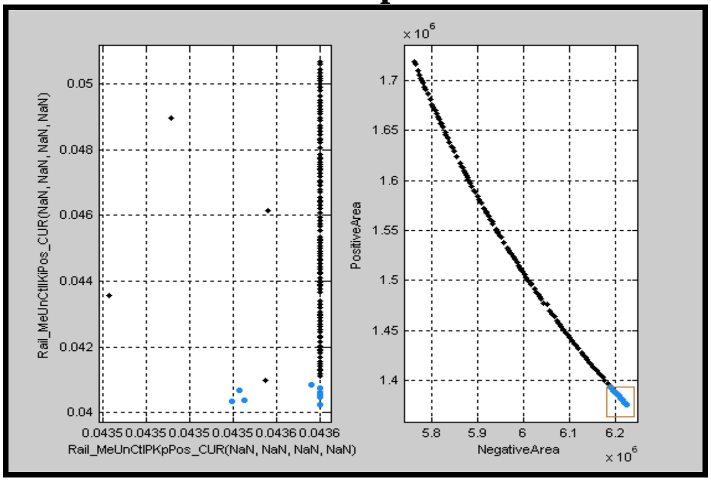
2600 rpm



2900 rpm



3200 rpm



3500 rpm