Politecnico di Milano

DEPARTMENT OF MECHANICAL ENGINEERING
School of Industrial and Information Engineering

Big data analysis for a high perfomance vehicle

Candidate:
Davide Maria Mussi
ID code 852461

Thesis Advisor:
Prof. Gianluca D'Errico

Company Tutor:
Ing. Alessandro Paone

ACADEMIC YEAR 2017/2018

Acknowledgement

I want to thank professor Gianluca D'Errico, my thesis advisor, who supported and helped me with a lot of useful advices and suggestions. He allowed me to write the thesis without any type of imposition: he tried to ease my job as much as he could, and for this I can just thank him.

I want to thank the company, Maserati, for having given me the opportunity to develop my thesis work within it: it was a great experience, that gave me the opportunity to see the inner mechanism of a famous and renowned company.

I want to thank my company tutor, Ing. Alessandro Paone, for introducing me to my job and for having followed me during the entire stage period, always answering my questions, even the simple ones, with calm and absolute clarity.

I want to thank all the members of the Validation team, within Maserati: Andrea, Giacomo, Giuseppe, Domenico, Luke, Andrea and Josè. They were always available to answer my questions and to give me suggestions on how to better fulfil all the tasks I had to do.

A special thank goes also to my friend and colleague Marco, whose suggestions really helped me in shaping the thesis.

A particular mention goes also to Lorenzo, whose skills and competences in using any type of software were really useful in helping me when I didn't know how to solve some problems. This aspect balanced the continuous and often really time consuming requests that he did, even if he was out of Italy on a business trip.

I want to thank my parents: they allowed me to live for six months in Modena, so that I could carry out my thesis work without any concern, focusing totally my attention on the job.

I want to thank my sister, Erica, and also Alessandro: I really appreciated their visits in Modena.

Contents

List of Figures

Abstract - English

In the nowadays automotive industry, the vehicle's ECU has to process a high amount of data, ensuring that all the components of the vehicle are correctly working and respecting the regulations of all the different countries. In order to accomplish such regulations, the car makers must do, after the design phase, a calibration and validation phase, in order to test the reliability of the vehicle. The data that are acquired during this test phase must be studied, in order to understand if all the normed diagnosis are fulfilled, if the pollutant regulations are respected and so on.

As the amount of data to be monitored increase, it isn't any more possible to rely on standard data analysis tool. The speed at which the data are produced is always increasing, leading to the field of Big Data: data that are so numerous, so quickly produced, that cannot be analysed with common tools. In order to overcome this problem, a specific, fast and reliable tool is needed, that allows to rapidly analyse an enormous amount of data in acceptable times, and to obtain consequently informations, in the shape of statistics, on those data.

For this reasons, ETAS, a Deutsch company, developed the tool Moogle, that allows to study the Big Data in short times; Maserati bought the license, and so it uses the tool for analysing all the vehicle's data.

In this thesis, I have shown quickly how the data are acquired from the vehicles; I have given to the reader a simple explanation of the Big Data, and their difference with the standard data. Eventually, I focused the attention on the tool, Moogle: how it works and the procedure that must be followed in order to produce statistic analysis. Then I have shown the results that can be obtained by the use of this tool, using as example an analysis performed on the knock phenomenon, on the Gasoline Particulate Filter and on the validation of an Environmental Temperature Model, so that the it is easier for the reader to understand the potential of such a tool.

I have dedicated also a section of the thesis to the description of the development of the tool: I was responsible of the communications between Maserati and ETAS, in order to provide to the Deutsch company our feedback and feature development's

request, so that the tool could be more and more refined.

Abstract - Italian

Nell'odierna industria automobilistica, la ECU deve elaborare una grande quantità di dati, assicurando che tutti i componenti del veicolo funzionino correttamente e rispettino le normative di tutti i diversi paesi. Al fine di attuare tali regolamenti, i produttori di automobili devono fare, dopo la fase di progettazione, una fase di calibrazione e convalida, al fine di testare l'affidabilità del veicolo. I dati acquisiti durante questa fase di test devono essere studiati, al fine di capire se sono soddisfatte tutte le diagnosi normate, se le normative sugli inquinanti sono rispettate e così via.

Con l'aumento della quantità di dati da monitorare, non è più possibile fare affidamento su strumenti di analisi dei dati standard. La velocità con cui i dati vengono prodotti è sempre crescente, portando al campo dei Big Data: dati così numerosi, così rapidamente prodotti, che non possono essere analizzati con strumenti comuni. Per superare questo problema è necessario uno strumento specifico, veloce e affidabile, che permetta di analizzare rapidamente un'enorme quantità di dati in tempi accettabili e di ottenere di conseguenza informazioni, sotto forma di statistiche, su quei dati.

Per questo motivo, ETAS, una società tedesca, ha sviluppato lo strumento Moogle, che consente di studiare i Big Data in tempi brevi; Maserati ha acquistato la licenza e quindi utilizza lo strumento per analizzare tutti i dati provenienti dalle vetture di flotta.

In questa tesi, ho mostrato rapidamente come i dati vengono acquisiti dalle vetture; ho dato al lettore una semplice spiegazione dei Big Data e la loro differenza con i dati standard. Successivamente, ho focalizzato l'attenzione sul tool, Moogle: come funziona e sulla procedura da seguire per produrre analisi statistiche. Poi ho mostrato i risultati che possono essere ottenuti con l'uso di questo strumento, usando come esempio un'analisi eseguita sul fenomeno del knock, sul filtro anti particolato per motori a benzina e sulla validazione di un modello di temperatura ambiente, in modo che sia più facile per il lettore capire il potenziale di un tale strumento.

Ho dedicato anche una sezione della tesi alla descrizione dello sviluppo dello

strumento: ero responsabile delle comunicazioni tra Maserati e ETAS, al fine di fornire alla società tedesca i nostri feedback e le richieste di sviluppo di determinate funzionalità, in modo che lo strumento potesse essere sempre più raffinato.

Chapter 1

Introduction

Politecnico di Milano gives to the students of the second level degree of Mechanical Engineering the possibility to perform the thesis work within a company. The stage has a duration of 960 hours, and must be performed into a company that operates within the course of study; at the end of the stage the student can write its thesis, that summarises the topics faced during the stage.

1.1 Company presentation

The history of the Emilian brand began on December 1, 1914, when the three brothers Alfieri II, Ernesto and Ettore Maserati founded, in Bologna, the "Alfieri Maserati anonymous company", a specialist workshop in the development of Isotta Fraschini and Diatto engines.

During the First World War Alfieri patents an innovative spark plug and, at the end of the conflict, focuses on the racing world, mounting a Hispano-Suiza engine on an Isotta Fraschini; with this car he won, together with his brother Ernesto, the Susa-Moncenisio of 1921.

In 1922 the Turin brand Diatto offered the Maserati brothers the direction of the brand's sporting activity and Alfieri the role of official driver, but three years later the Piedmontese company renounced the races due to debts. Thanks to the financial support of the Marquis Diego de Sterlich, the Maserati are able to buy ten Diatto 30 Sport chassis.

The first ever Maserati was the Type 26 of 1926, an evolution of the Diatto GP 8C turbo; in the same year Mario, artist and brother Maserati devoid of passion for mechanics, realizes the legendary logo of the Trident, inspired by the one on the fountain of Neptune in Piazza Maggiore in Bologna.

In 1927, during a race in Sicily (Coppa Messina), Alfieri was the victim of

Figure 1.1: *Maserati's brothers in front of their headquarters*

a terrible accident in which he lost a kidney and, the following year, he obtained the first important victory, at the Coppa dell'Etna, thanks to Baconin Borzacchini. The Umbrian pilot obtained, in 1929 in Cremona, at the wheel of a V4 car equipped with a 16-cylinder engine, the world speed record on the 10 *km* launched and in 1930 won the first international success of the Trident at the Tripoli GP.

When, in 1932, Alfieri Maserati died as a result of surgery, Ernesto decides to abandon his career as a pilot to take over the company: he focuses on the technical side along with Ettore and appoints the fourth brother, Bindo Maserati. The following year the first victory came, in France with Giuseppe Campari, in one of the Grandes Épreuves (the most important races before the advent of F1) and Tazio Nuvolari (who left Ferrari after a fight with Enzo due to the refusal of this last to want him as a partner to 50% of the Cavallino) triumphs in the GP of Belgium after starting in last place on a modified 8CM.

In 1937 Maserati was bought by the Modena entrepreneur Adolfo Orsi, who focuses on the brand's coat of arms to promote his other activities. Bindo, Ernesto and Ettore, free from the management burden, divide the tasks within the company: the first deals with the commercial part, the second of the design and the third of the candle factory. In the same year, Giulio Severi won the Targa Florio with a 6CM: the Sicilian race was won for another three consecutive times by the

Trident with Giovanni Rocco and on two occasions with Luigi Villoresi.

In 1939 Orsi decides to transfer the Maserati to his hometown, Modena, in an area owned by him; in the same year and in the following the 8CTF, first vehicle designed under the new direction, won the Indianapolis 500 with the American driver Wilbur Shaw. To date, this remains the last victory of an Italian engine in Indianapolis.

Following the outbreak of the Second World War, the production of racing cars was temporarily arrested, to favor the making of candles and products for the army. At the end of the war Adolfo Orsi lost interest in racing, but the three Maserati brothers continue to keep alive their "racing" dream thanks to some 4Cs hidden in Milan during the conflict. On the other side of the Atlantic Ocean, in the meantime, Louis Unser won two editions of the legendary uphill race Pikes Peak (1946 and 1947) at the wheel of a Trident car.

The A6 GCS is the last Maserati designed by the Bindo brothers, Ernesto and Ettore before leaving the company. In 1947 the series production begins and the first Trident car available for the general public is the A6 1500; presented at the Geneva Motor Show of the same year, is designed by Pininfarina and features a 1.5 six-cylinder in line engine.

The sport successes, meanwhile, continue to arrive: in 1948 the Trident returned to win after 15 years one of the Grandes Épreuves (Monte Carlo) with Giuseppe Farina and the 4CLT and Silverstone with Villoresi and the 4CLT / 1948; moreover, the following year, the Swiss Toulo de Graffenried triumphed again on the British circuit.

In 1950 the Modenese House is able to take part in the first F1 World Championship in its history and gets the first podium in the second race of the season thanks to the Monegasque driver Louis Chiron, who finishes in third place the home GP.

In 1952 he was hired the best driver of those years, the Argentine Juan Manuel Fangio, who only debuts in 1953 because of an accident in F2, which forces him to skip the entire season.

The 250F, designed to participate in the F1 1954 World Championship and featuring a 240 HP 2.5 engine, is still today considered one of the best cars of all time. Fangio is under contract with Mercedes but, waiting for the German car to be ready, asks (and gets) to be able to race the first two GPs of the year (Argentina and Belgium) with the car of the Trident; he wins them both and at the end of the season he becomes world champion.

In 1956 the protagonist of the races of the Modenese House is the British Stirling Moss: close to the World Championship triumphing in Monte Carlo and Monza

and together with the Argentinean Carlos Menditéguy conquered in Buenos Aires with the 300S the first victory of the brand, in a race valid for the World Sports Prototypes.

1957 is the best year of Maserati history: Fangio becomes F1 world champion for the fifth and last time in his career with four wins (Argentina, Monte Carlo, France and Germany) and two second places in seven GPs. That same year saw the light of the first Trident car built on a large scale: the 3500 GT, a model that officially transforms the Modenese brand into a car manufacturer.

In 1958 Adolfo Orsi finds himself in crisis due to the failure to pay an important delivery of milling machines by the Argentine government. It sells various personal properties, sells the machine tool division to a foreign company and obtains a controlled administration regime for one year. He pays all the debtors but is forced to close the Reparto Corse: he manages to settle all his employees elsewhere, even among the bitter rivals of Ferrari.

The '60s opened with the latest success in the World Sports Prototypes of a Maserati: a Type 61 run by the US Camoradi team that climbs on the top step of the 1000 km Nürburgring of 1961 thanks to the duo composed of Masten Gregory and Lloyd Casner. In the same year the 3500 GTI was presented, the first ever Italian with an injection engine.

1963 is the year in which the Mistral (last car of the Trident equipped with the legendary six-cylinder in line derived from that of the 250F) and above all the Quattroporte: the one that at the time is the fastest sedan in the world is presented at the Turin Motor Show and is powered by a powerful 260 HP 4.1 V8 engine able to reach a maximum speed of 230 km/h.

The last triumph of a Maserati F1 engine dates back to 1967, when the Mexican Pedro Rodríguez won the South African GP at the wheel of a Cooper. In the same year the Ghibli was born, the first Modenese car designed by Giorgetto Giugiaro.

In 1968, 60% of Maserati shares went to Citroën (Orsi remains honorary president). The French company uses the engines of the Modenese brand to create models such as the SM, while the Trident brand acquires some technologies of the transalpine manufacturer such as hydropneumatic suspensions.

Three years later, concurrent with the launch of the Bora (a mid-engine coupé created to steal customers from the Lamborghini Miura), the Orsi family finally left Maserati but, in 1973, due to the oil crisis, the Michelin family, owner of Citroën, sells the French brand to Peugeot and the new owners decide to put into liquidation the legendary sports brand in Emilia.

Thanks to the public money coming from GEPI (Società per le Gestioni e Partecipazioni Industriali, translated as Company for Management and Industrial

Figure 1.2: *Maserati 3200GT*

Participations) the Argentinian businessman Alejandro de Tomaso acquires a large part of the Maserati shares: he heals the company's debts and invests on new models such as the third generation of the Quattroporte (1979, drawn da Giugiaro) and the Biturbo (1981), a two-door sedan / coupé characterized by an affordable price but also by a precarious reliability due to reduced design times. The car, assembled in Milan in the Innocenti plants, achieves an incredible success but negatively affects the image of the brand.

In 1993 Maserati moved to the Fiat Group, which four years later sold 50% of the company's shares to Ferrari (Ferrari itself was owned by Fiat). In 1999 Ferrari took full control, making Maserati its luxury division; a new factory was built, which replaces the existing structure of the 1940s. Ferrari is credited with having brought the Maserati back to business after it was on the verge of bankruptcy.

The first car of the Trident under the new management was the 3200 GT (designed by Giugiaro, Figure 1.2) that was unveiled at the 1998 Paris Motor Show, a year before the complete transfer of the Modenese brand to the Cavallino.

In 2001, Ferrari decided to throw away all the old tools and install high-tech devices in the Modena plant (Figure 1.3), making it one of the most advanced in the world.

In 2001 the Emilian House returns to the US market with the Spyder, shown at the Frankfurt Motor Show: the first ever Maserati to mount a gearshift with a steering wheel has a line similar to that of the 3200 GT but is distinguished by the rear lights more traditional, for the revised chassis (shorter wheelbase), for the Ferrari 4.2 V8 engine and for the transaxle transmission (at the rear axle in block with the differential).

Figure 1.3: *Maserati's headquarters in Modena, viale Ciro Menotti 322*

2003 is an important year for Maserati: the fifth generation of the Quattroporte (unveiled in Frankfurt) is the first Trident car designed by Pininfarina after half a century and obtains numerous customers. In 2004, instead, the official return in racing (after 47 years) with the MC12: the car, equipped with a 6.0 V12 engine and a monocoque carbon fiber frame, participates in the FIA GT championship and gets the first victory over German circuit of Oschersleben with Mika Salo and Andrea Bertolini.

Between 2005 and 2009 four Pilot titles arrive (three for the couple Bertolini-Bartels and one for Thomas Biagi), two Constructor titles (2005 and 2007, the year in which the GranTurismo designed by Pininfarina makes its debut) and five consecutive championships for the Vitaphone team (Figure 1.4). In 2010 the FIA GT series changes its name to the GT1 World Championship, but the results do not change: the triumph of the championship for Bartels and Bertolini and dominance among Vitaphone teams.

The Maserati and Alfa Romeo group, under Fiat Group, started in 2005, when Maserati was split off from Ferrari and partnered with Alfa Romeo. Although Maserati and Alfa Romeo are in a brand group, Alfa Romeo is structured under FCA Italy S.p.A., which itself is structured under FCA, whereas Maserati is structured solely under FCA.

In 2013, Maserati began its expansion with the sixth-generation Maserati Quattroporte, designed to compete better with the Mercedes-Benz S-Class. This was followed by the introduction of the Ghibli, which was designated to compete against Mercedes-Benz E-Class and BMW 5 series.

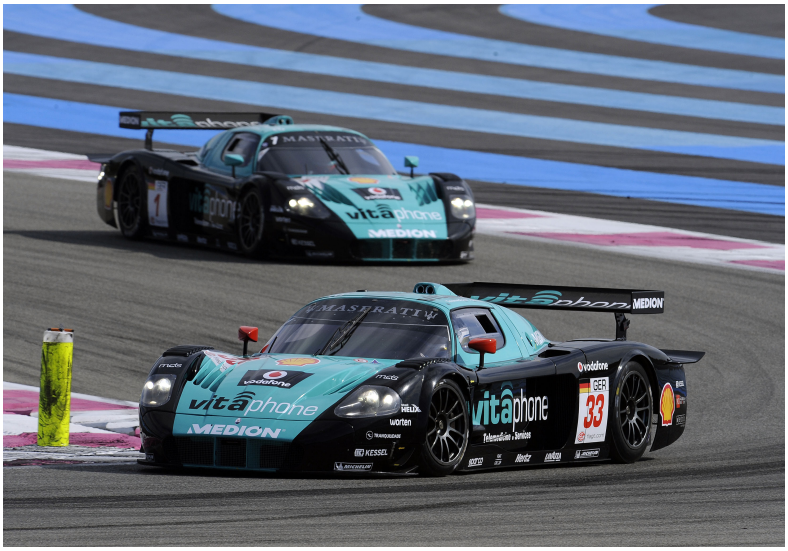


Figure 1.4: *Maserati MC12 by Vitaphone Team*

1.1.1 Actual Production

Maserati's actual production is composed by the following models:

- Quattroporte (Figure 1.5a): is a sporting luxury saloon. The sixth generation Maserati Quattroporte was introduced in 2013. The Quattroporte is currently available in S Q4, GTS and Diesel trim. The S Q4 has an advanced four wheel drive system, and a 430 HP twin-turbo V6. The GTS is rear wheel drive, and has a 530 HP V8. A Quattroporte Diesel model is offered on selected markets, making 275 HP (250 HP in Italy). The sixth-generation Quattroporte has grown in size in order to better compete with the roomier luxury saloons like the Mercedes-Benz S-Class. It is produced in the italian factory of Grugliasco, near Turin;

- Ghibli (Figure 1.5b): it is a sporting/luxury executive saloon that competes against the BMW 5 Series, Mercedes E-Class or Audi A6. The car, along with the new Quattroporte, is built in the Italian factory of Grugliasco, Turin. The base Ghibli comes with 350 HP, the Ghibli Diesel with 275 HP (also 250 in Italy only), and the Ghibli S Q4 with 430 HP;

- GranTurismo and GranCabrio (Figure 1.5c): the Maserati GranTurismo is a grand tourer introduced in 2007. The GranTurismo has a 4.7-litre V8, making 460 HP (339 kW; 460 PS) in Sport form and MC form. A convertible (GranCabrio) version is also available in standard, Sport, and MC form, equipped with the same engine of the GranTurismo;

(a) *Maserati Quattroporte*

(b) *Maserati Ghibli*

(c) *Maserati GranTurismo*

(d) *Maserati Levante*

Figure 1.5: *Actual Maserati's production vehicles*

- Levante (Figure 1.5d): it is a mid-size luxury crossover four wheel drive SUV based on the Kubang concept car that debuted at the 2011 Frankfurt Auto Show, and built by Maserati at the Mirafiori factory in Turin, Italy starting in 2016. It features Maserati's 3.0L V6 engine, in two states of tune(350/430 HP). The Levante diesel comes in two version: 250 or 275 HP. In 2018 Levante will feature a 3.8 V8 gasoline engine with 520 HP.

Moreover, Maserati is also producing, in Modena plant, Alfa Romeo 4C and Alfa Romeo 4C Spyder (Figure 1.6) . It is a mid-engined, lightweight, rear-wheel drive sports car and it uses a carbon fiber tub, front and rear crash box, and hybrid rear subframe mainly out of aluminium to keep weight at 895 kg and 1,050 kg in the United States. 4C Spyder shares its engine with the Coupé version, but it has different external parts such as the headlights, exhaust and engine hood, as well as a different roof section that features a removable roof panel. Alfa had to make some strengthening tweaks to the 4C to cope with the removal of the top, with

Figure 1.6: *Alfa Romeo 4C*

the net result being a 45 kg weight increase (dry weight 940 kg). In the US, the weight difference is listed at only 10 kg.

1.2 Validation fleet

The Validation fleet is a collection of cars, all equipped exclusively of petrol engines, with instrumentation on board the vehicle, thanks to the which the data from the sensors on the vehicle can be acquired, in particular all the sensors related to the motor and to the related organs (low and high pressure fuel pump, lambda sensors, etc..).

These measures serve, thanks to powertrain team studies, for the approval of the calibration performed on the engine, and move on to the next phase development, until it reaches the final car, which will be available to the buyer.

Modifications to both the software and the hardware of the listed vehicles are introduced each year to improve performances and durability and therefore each year such modifications need to be calibrated and validated. For this reason the Powertrain team acquires new vehicles every year, designated by the sign MY followed by the year of expected market production.

The fleet consists of vehicles in different stages of development, all however following the prototype phase. Validation fleet owned by the Powertrain Team in Modena consists on a limited amount of gasoline vehicles for each Maserati and Alfa Romeo model and motorization. These are distinguished by means of a alphanumerical code, that allows the workers to discriminate between vehicles. These are:

- Alfa Romeo (AR) V6 2.9 510 HP (Quadrifoglio)
    1. Stelvio
    2. Giulia
- Alfa Romeo (AR) 4 in-line cylinders 2.0 200/280 HP
    1. Stelvio
    2. Giulia
- Maserati (M) V6 3.0
    1. Quattroporte (430 HP)
    2. Ghibli (350/430 HP)
    3. Levante (350/430 HP)
- Maserati (M) V8 3.8
    1. Quattroporte (530 HP)
    2. Levante (520 HP)

The vehicles are differentiated depending on the stage of the production process they belong to. So there are the following categories:

1. PVP (Pre-Vehicle Production)
2. VP (Vehicle Production)
3. PS (Pre Series)
4. IP (Initiated Production)

The first two types (PVP and VP) are different from the market production ones in the hardware, having some less developed parts of the shell and frame for cost reduction, while the PS and the IP ones are totally identical to the ones sold to the public.

1.3 On board instrumentation and data acquisition

As it can be read in [1], in order to acquire the data, each vehicle is equipped with a particular instrumentation. Actually, Maserati's vehicles are equipped with two different acquisition systems: ControlTec and ETAS. Before describing these two arrangements, I need to briefly describe the bus systems adopted by the two acquisition systems.

Network for data communication, also known as bus systems or protocols, are widely used in today's motor vehicles. Various components such as sensors, actuators or ECUs - the "nodes" - are connected to each other via a single channel. A wealth of data, also known as messages, telegrams, packages or frames, is exchanged via this channel.

Comparing with conventional cabling, in which the transmitter and receiver of information are each connected by separate lines, bus systems offer significant advantages:

- The costs of materials for the cables are lower.
- The space required and the weight of the cabling are lower.
- The number of plugs susceptible to fault is lower.
- Data can be distributed to various receivers.
- All the systems in the vehicle which are connected via the bus can be reached from one access point. This provides for simpler diagnostic and the configuration of all the ECUs at the end of the line.
- The performance of the calculations can be distributed to different ECUs.
- Analog sensor signals must be digitalised for data processing. The sensor signal can be conditioned directly in the sensor, the information is then disseminated via the bus.

The performance of the bus must be clearly defined in a public standard in order to have a benchmark against which components can be verified. This ensures that verified components of different suppliers function together in a single network.

1.3.1 CAN Protocol

As it can be seen in [2], the CAN bus (Controller Area Network) has established itself as the standard since its first series introduction in motor vehicles in 1991. The major features are:

- Priority-controlled message transmission with non-destructive arbitration.
- Low costs through the use of a low-cost twisted two-wire cable and use of a simple protocol with low computing-power demand.
- A data transfer up to 1 $Mbit/s$ for the high-speed CAN (CAN-C) and up to 125 kb/s for the low speed CAN (CAN-B, lower expenditure for hardware).
- High reliability of data transfer through recognition and reporting of sporadic faults and permanent faults and through network-wide consistency via Acknowledge.
- The multi-master principle.
- High availability by locating failed-stations.
- Standardization in accordance with ISO 11898.

The maximum permissible transfer rate is dependent on the overall length of the bus. ISO specifies 1 $Mbit/s$ for 40 m. For longer lines, the possible transfer rate is roughly inversely proportional to the line length. Networks with a 1 km reach can be operated with 40 $kbit/s$.

A traditional CAN network is composed by a linear bus, realised with a twisted two-wire cable, and by a theoretically infinite number of nodes, linked together by means of the aforementioned cable. All the components, linked in the bus, have the same "rights", they listen in parallel all the messages, they are always ready to receive data and, when necessary, they start transmitting if any other component is sending messages on the bus.

1.3.2 ETK Protocol

As it can again read in [3], ETK is an Ethernet-based interface. The term Ethernet refers to a family of buses in which the addressing, the format of the messages and access control are identical (laid down in IEEE 802). Ethernet and the Internet Protocol were developed for data communication between computers or peripherals which are locally separated. The Ethernet buses are identified by the following important features:

- The transfer rate is in the range of 10 $Mbit/s$ up to 10 $Gbit/s$.
- Data transfer is possible via assorted media such as coaxial cable, twisted two-wire cable, radio or glass fiber.

- The technology involved is standardised and very widely used.
- Simple insertion and removal of nodes is possible.
- The time response in the case of real-time applications is not guaranteed.

Ethernet is used in series production vehicles.

From [4], it can be read that in order to validate the correct function of an ECU, data from different source need to be measured, analysed and recorded. Especially in the calibration phase, a high amount of ECU data has to be collected. Standard ECU interfaces like CAN are often not sufficient for these kind of analysing tasks. For this reason, the ETK interface was developed.

The ETK interface by ETAS provides direct access to the measurement variables and control parameters of an ECU via the parallel data and address bus, or via a serial micro controller testing or debugging interface. The ETK interface are real-time capable. Its dedicated power supplies enable the preparation and initiation of cold-start testing independently of the ECU. Due to its extremely compact design, ETK can be accommodated inside the housings of production ECUs. It is impervious to the temperature extremes and vibrations at the ECU's location in the vehicle.

The benefits of ETK interface are:

- Independent of the ECU vehicle-bus interfaces.
- Small form factor.
- Mechanically and electrically robust.
- Simultaneous access to the ECU for measurement, calibration and rapid prototyping tools.

Thanks to the ETK, measurement and calibration of ECU variables and control parameters, prototyping of ECU functions and ECU flash programming can be performed.

1.3.3 ControlTec instrumentation

ControlTec works with CAN bus. In order to acquire the data, ControlTec uses the CT-1000 acquisition system. Thank to this tool, the data are sent to ControlTec servers; there, they can be studied from Maserati technicians by means of the ControlTec Qualifier software.

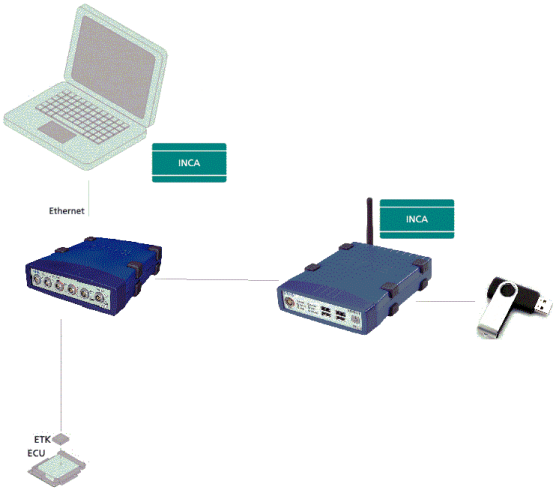


Figure 1.7: *ETAS instrumentation chain*

1.3.4 ETAS instrumentation

ETAS instrumentation is composed of different tools:

- ES592.1: Interface Module that provides one ETK connection plus two CAN interfaces. This component takes the data from the ECU, and sends them to a PC that has installed INCA, so that the data can be saved and stored.
- ES720: Replacing the INCA PC or laptop, the module captures and records signals from ECUs, buses, networks, sensors, and measuring instruments in the vehicle. Once configured, the module records all data from connected devices without user or driver interaction[5]. This tool has also mounted a USB-port: in fact, the data can be stored either in ES720 internal flash memory, or in an external USB device, such like classic Pen Drive.

The files extracted from ES720 are the measurement files that are then studied within Maserati. The ETAS instrumentation chain is schematized in Figure 1.7.

1.4 Available Software

As a member of the Powertrain team, I have access to a group of software that allows me to study and analyse the acquired data. The software I use are the following:

- ETAS MDA 7.1 (Measure Data Analyzer);
- ETAS Moogle;

- Delphi Automotive CONTROLTEC Qualifier;
- MS Office.

1.4.1 ETAS MDA

From [6], it can be read that during the ECU development process, measurement data that refers to different states of calibration data must be compared. The ETAS MDA (Measure Data Analyzer) measurement data analysis tool lets users visualize, further process, analyze, and document measurement data. The tool supports MDF (Measurement Data Format) and enables users to evaluate correlations between diagnosis data and measurement signals from the ECU and from sensors and vehicle buses.

Measurement data can be displayed in a uniform layout with the help of predefinable configurations. Using the same views and settings to display signals from different measurements makes it easy to compare measurements. Layouts for printing and documenting measurement results can easily be configured to comply with corporate design specifications. Evaluation results can be saved in MDA as MDF and XDA files for selected signals and time ranges.

The MDA V7.1 measurement data analysis tool lets users display and analyze measurement data using virtual oscilloscopes with YT and XY display modes, as well as tables and statistical analyses. In addition, this version offers a wide range of functions that support the display and analysis of measurement data.

MDA can be used for the following applications:

- Visualize and evaluate ECU and vehicle measurement data;
- Generate measurement files in MDF format.

The advantages of using such a software are:

- Synchronization of cursors and time intervals for different displays;
- Calculation of derived signals;
- Separation of display layouts and measurement data.

1.4.2 Delphi Automotive CONTROLTEC Qualifier

Software that allows a first level analysis of the acquired data. The complete and intuitive interface makes the use of the software very simple; on the other hand, it is not possible to work with a very large number of channels. The main features are:

- Fleet management with geolocation functionality;
- Managing problems with notifications and reports in real time;
- Automated analysis to evaluate the performance of a data subsystem;
- Time-based analysis of the data acquired, with links to all data acquired during each event / trip with maps, statistics, histograms, calculated signals, graphs and so on;
- Tracking the number of Hardware / Software components of all modules guarantees quality test miles;
- Data accessible worldwide, at any time;
- Database easy to consult for in-depth analyzes;
- Limit thresholds to compare vehicles, calibrations and so on.

1.4.3 ETAS MOOGLE

It is a post-processing data software. Once measures are available, with this program I can extract, from the data available on the Maserati servers, only the requested information. Moogle has some advantages and drawbacks. The advantages are:

1. High number of study channels.
2. Ability to perform post processing work on measurements.
3. Possibility to export data extracted from measurements to formats such as PDF, Excel or JPEG.
4. Possibility to isolate particular conditions in the vehicle use (e.g: double injection and half engine mode conditions for analysis on the combustion chamber knock phenomenon).

Its drawbacks are:

1. The duration of the crawling does not allow having results immediately after the creation of the trigger. Therefore, it means that between the request of a certain statistics and its final emission always passes a certain amount of time.

2. Moogle can only be used if the user has a simple knowledge of basic programming, because, in order to have it to extract the desired information, a short and simple C++ language program must be written, as will be explained in 2.

Chapter 2

Big Data

2.1 Definition

The amount of data, generated by human activities, is increasing over time. For this reason, a new kind of data definition was needed. Basically, Big Data (BD) is data that exceeds the processing capacity of conventional database systems. The data is too big, moves too fast, or doesn't fit the strictures of standard database architectures[7].

BD can be defined according to the three V classification, that describe its characteristics:

- Volume: it represents the quantity of generated and stored data. Usually, a standard minimum size for BD is in the order of Petabyte (PB). Of course, it is a definition that will change over time, due to the continuously increasing data production of the society. However, volume is not enough in order to describe BD.

- Variety: the type and nature of stored data must be different. So we can have BD characterized by having pictures, video, text files and so on.

- Velocity: It is the speed at which the data is generated. Often, BD are real-time data. This of course has a great impact on the ability of managing and studying this type of data.

Anyway, during time the definition of BD has been enriched with other characteristics:

- Veracity: most of BD comes from sources outside our control and therefore suffers from significant correctness or accuracy problems. Veracity represents

both the credibility of the data source as well as the suitability of the data for the target audience[7].

- Variability: Inconsistency of data can hamper the data processing procedure.
- Flexibility: it means holding the traits of extensionality (can add new fields easily) and scalability (can expand in size rapidly)[8].

From this definition, it can be easily understood that the common analysis instruments cannot be used when we are dealing with BD. For this reason, the growth in importance of BD led to the creation of dedicated analysis tools, such like ETAS Moogle, related to Automotive industry.

Traditionally, data analysis techniques have been designed to extract insights from scarce, static, clean and poorly relational data sets, scientifically sampled and adhering to strict assumptions, and generated and analysed with a specific question in mind. The challenge of analysing BD is coping with abundance, exhaustivity and variety, timeliness and dynamism, messiness and uncertainty, high relationality, and the fact that much of what is generated has no specific question in mind or is a by-product of another activity. Such a challenge was until recently too complex and difficult to implement, but has become possible due to high-powered computation and new analytical techniques[8].

Moreover, different BD analysis can be performed. According to [7], we can distinguished between two type of analysis:

- Basic analytic: it includes simple visualizations or simple statistics.
- Advanced analytic: it provides algorithms for complex analysis of either structured and unstructured data. It includes sophisticated statistical models, machine learning, neural networks, text analysis and so on. Among its many use cases, advanced analytic can be deployed to find patterns in data, prediction, forecasting, and complex event processing.

It is immediately obvious that the most interesting and convenient type of analytic is the advanced one. Especially in the Automotive field, where this type of analyses could really improve the situation, leading to a save of time and, consequently, money.

2.2 Big Data in Maserati

As vehicles becomes more and more complex for what concerns the electronic controls, the amount of data that the ECUs normally treat is increasing. Due to the fact that emission-control legislation are becoming more and more stringent, as it can be read in [9], lawmakers have now acknowledged on-board diagnostic as an aidto monitoring exhaust gas emissions, and have produced manufacturer-independent standardization. This system is termed OBD system (On-Board Diagnostic system).

2.2.1 OBD

Always in [9], it can be read that the engine system and components must be continuously monitored in driving mode so that compliance with the emission limits required by law can be achieved in everyday use. Therefore, starting in California, regulations were adopted to monitor exhaust-gas-related systems and components.

OBD I (CARB)

In 1998 the first stage of CARB legislation(California Air Resource Board) came into force in California with OBD I. This first OBD stage requires the monitoring of emission-related electrical components (short-circuits, line breaks) and storage of the faults in the control-unit fault memory as well as a malfunction indicator lamp(MIL) that alerts the driver to detected faults. On-board means(e.g flashing code on a diagnosis lamp) must also be in place to provide a readout of which component has malfunctioned.

OBD II (CARB)

In 1994 the second stage of diagnosis legislation was introduced in California with OBD II. In addition to the purposes of OBD I, system functionality was now monitored(e.g plausibility check of sensor signals).

OBD II stipulates that all emission-related systems and components must be monitored if they cause an increase in toxic exhaust-gas emissions(and thus the OBD limits to be exceeded) in the event of a malfunction. In addition, all the components used to monitor emission-related components or which affect the diagnosis result must be monitored.

OBD isn't a fixed legislation: it is continuously updated and revised in order to further improve the effectiveness of the diagnostic system, and also to increase the

number of phenomenon that are under investigation. For instance, since 2007 it is mandatory the diagnosis of cylinder-individual mixture trimming, extended requirements with regards to diagnosis of the cold-start strategy, and the permanent error/fault storage[9].

EOBD(European OBD)

On-board diagnostic attuned to European conditions is termed EOBD. EOBD has been valid since January 2000 for passenger cars equipped with gasoline engines. This version of OBD has to deal with the European laws in terms of exhaust gas emission, that are the EuroX requirements. So the EOBD is updated and revised every time that a new EuroX requirement is emitted. The last revision is the one concerning Euro6-2 requirements(September 2017), in which some OBD pollutant emission limits were decreased.

OBD functions overview

As can be found in [9], whereas EOBD only contains detailed monitoring specifications for individual components, the specific requirements in CARB OBD II are much more detailed. The following list shows the current of the CARB requirements for gasoline-engined vehicles. The requirements that are also described in detail in the EOBD legislation are marked by (E):

- Exhaust-gas recirculation system(E).
- Cold-starting emission-control system.
- Crankcase ventilation.
- Combustion misses/misfiring(E).
- Fuel system.
- Variable valve timing.
- Exhaust-gas sensors(λ oxygen sensor (E), NO_x sensors (E), particulate sensor).
- Engine cooling system.
- Other emission-related components and systems(E).
- In-use Monitor Performance Ratio (IUMPR) for checking the frequency of diagnostic functions in everyday operation(E).

- Secondary-air injection.
- Three-way catalytic converter(E), heated catalytic converter.
- Tank-leak diagnosis, with(E) at least electrical testing of the canister-purge valve.
- Air-conditioning system(in the event of influence on emission or on OBD).
- Direct ozone-reduction system.

From here, it is easily understandable that the necessity of monitoring such a high number of functions leads to a very high number of variables that must be analysed, and so they must necessarily be acquired.

2.2.2 Data recording and storage

Actually, Maserati plans a mileage accumulation strategy for its own vehicles. So the vehicles must perform a planned number of kilometres during their life; this mileage accumulation is needed to collect the data, thanks to the acquisition chain seen in Chapter 1, that will be consequently studied by Maserati's technicians.

Maserati doesn't use directly internal employees for the recording of data, but relies on external companies that have the duty to drive the vehicles as decided by Maserati's strategy. In fact, Maserati plans the accumulation in order to do a certain amount of driving time in urban, suburban and highway routes, and also high-speed routes, meaning on-track drive sessions. After each driving shift, the drivers extract the data from ETAS acquisition system, storing them on a USB pen drive, and load the measurement file on Maserati's server structure.

Obviously, as the number of vehicles increase, also the amount of stored data increase. And the data dimensions increase day by day, at quite a good rate. Just to give an idea, usually a driving shift means to have a measurement file of more than 1 Gigabyte. Each day is composed of two driving shift, that multiplied for the number of vehicles give to Maserati a significant amount of data per day. So we can effectively talk of Big Data for what concerns Maserati.

The stored data can be used for different purposes, but the main ones are:

- Calibration.
- Diagnosis.

Calibration

Internal combustion engines are organs that result particularly complex and their control is particularly important and delicate. The engine represents, in fact, the main system for the propulsion of any vehicle and it is therefore necessary for it to provide an appropriate response to the driver's requests, in order to guarantee a consistent driveability. For responding to these needs, the engine control is entrusted to the electronic control unit (ECU) dedicated. ECU has the task of monitoring the operation of the engine, processing in real time a large number of signals coming from instruments measurement and sensors, installed in strategic positions inside the engine block. Moreover, it has the task to process the aforementioned signals and to command the actuators, following certain strategies, to perform the functions that are required. The control unit therefore has a function essential for the correct functioning of the engine.

Within ECU is present a software that defines the algorithms that allow the correct process of calculation. Within the software a big number of function are present, each one characterised by its own algorithm and that must fulfil a determined task. Each function contains maps, that can be seen as tables of various type and size that, receiving a certain number of signal as input, provide the corresponding calibrated values as output.

The calibration consists in defining the maps, modifying so the engine software, in order to have the car to behave in the desired way.

It is then clear that calibrator's activities can be hugely enhanced by having at disposal the data recorded during the mileage accumulation. So they can immediately understand if their calibrations worked properly, or if there is the need to further improve them.

Diagnosis

As the accumulation goes on, it can happen that sometimes some errors are detected by the ECU, and displayed to the driver's instrument panel. In that case, the vehicle must immediately be stopped, and the measurement file sent to Maserati, so that it can be analysed in order to perform an accurate diagnosis. The diagnosis purpose is law mandatory, as previously explained in Section 2.2.1.

At this stage, the engineers must understand if there was effectively an error detection: in this case, the root cause must be found and eventually it must be fixed, usually by substituting the damaged or faulty component. Sometimes it can happen that the error detected by the ECU is simply a misdetection: it means that in reality there isn't any error, and the software wrongly detected one. This means

that there is a software bug, that requires a fix in the following software release.

This part of the process is of fundamental importance: any vehicle's producer cannot sell to the final customer a vehicle that doesn't work at best, especially when such high luxury brands are involved, like Maserati or Alfa Romeo.

2.2.3 Weakness of Standard Data Storage

Such type of use of data is very good in order to improve the development of engine's software and can save time and money if properly implemented within car manufacturer's companies. What I have described until now foresees that the engineers manually have to open and study the single measurement files, in order to find the exact moment where the critical situation took place.

The big drawback of such a procedure is that it is highly time demanding. The standard procedure used to study the files is the following one:

1. Location of the file: often more than one file per day, for every vehicle, is recorded. So the first thing to do is to find the correct file between the ones of the day.

2. File opening: this can take also some minutes, especially in the case of standard laptops.

3. Signals selection: in MDA the channels that will be displayed must be manually chosen. If they are a few, the requested time is not high. If they are a lot, this time increases. Moreover, as the displayed signals increase, the file opening time increases as well.

4. File study: at this point, the file can be finally studied. It means that engineer has to find the exact time window in which the problem arose. Normally, the measurement files are characterised by having a remarkable time length, in the order of different hours. So also this step can take a discrete amount of time.

This process is too much time-consuming, and cannot be adopted on nowadays Validation fleets. For this reason, new tools are developed in order to allow a fast and reliable way to study the stored data. Within Maserati the software that is in charge to do this process is in fact ETAS Moogle.

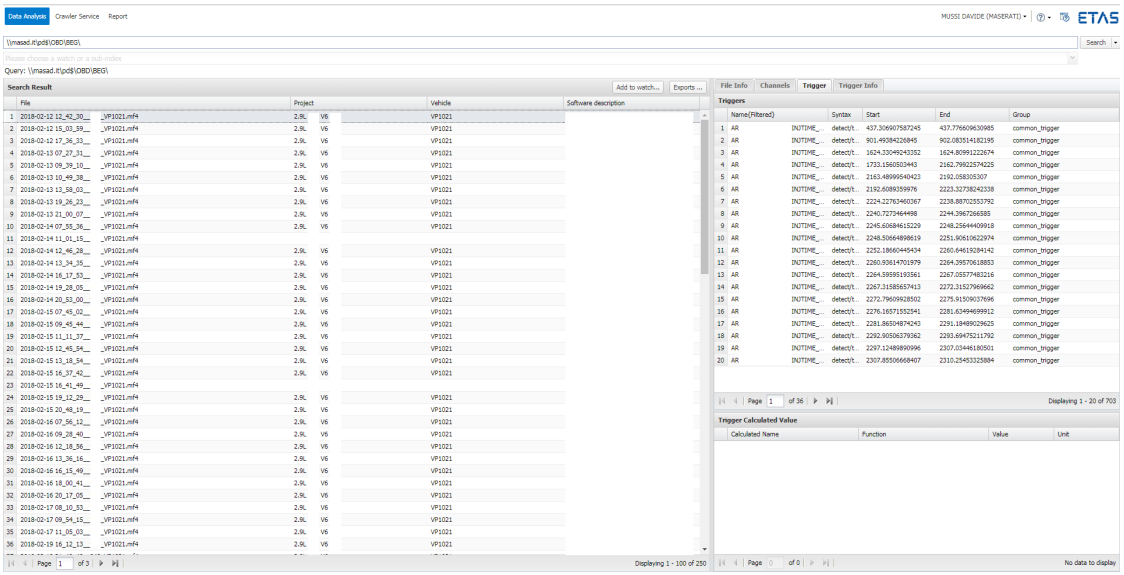


Figure 2.1: *Moogle starting window: Data Analysis. In this page it is possible to have a rapid overview of the measurement files and the triggers*

2.3 ETAS Moogle

As already mentioned in Chapter 1, Moogle is a post-processing software that analyses the measurement files present within Maserati. Moogle allows to extract synthetic informations, in the shape of charts, from all the measurement files stored in Maserati's servers, and consequently to create statistics. In order to do so, the user must follow a simple but mandatory procedure, explained in the following part:

- To write the so called *trigger*: it contains the instructions that Moogle receives, and that allows it to understand what informations extract from the measurement files.

- To test the trigger, on a small and accurately selected sample of measurement files, in order to ensure that the trigger effectively extract the requested informations. In fact, an error during trigger creation can lead to wrong results and, in the end, loss of time.

- Trigger conversion.

- Indexing of the measurement file: Moogle follows the instructions contained in the trigger and applies them to the measurement file. It is the most time-consuming part, because the user can only wait the software to complete the process. The time required by Moogle to complete this process depends

on the overall dimension of the measurement files, and the complexity and number of triggers.

- Creation of the statistic reports, using the Moogle graphic interface in order to select what type of chart must be created, what variables will be placed on the axis and other additional filters.

In Figure 2.1 it is possible to see the starting window of Moogle. In this page it is possible to select a specific vehicle or a specific model (Alfa Romeo Giulia/Stelvio, Maserati Quattroporte/Ghibli/Levante), and then to have a rapid and instantaneous overview of the triggers that have been applied to those vehicles, and also to extract the snippets of a measurement files related to a specific occurrence of a specific trigger. Thanks to this feature, it is possible to deliver to the colleagues directly the part of a measure in which a critical situation happened, without loosing time in order to find it in the whole database.

In the following sections I'm going to better show the aforementioned points, in order to make as clear as possible how Moogle works, and how I had to deal with it.

2.3.1 Trigger definition

The trigger is a small program that must be developed by the Moogle's user in a language very similar to a simplified version of C. The trigger is essentially a text file, written with Notepad++. In the trigger a series of things must be specified:

1. Trigger name: due to the fact that the user can write as many triggers as he desires, each trigger must be named in a way that Moogle can recognize it. If two triggers have the same name, Moogle will not consider one of the two.

2. Trigger conditions: these are the conditions in which I want to see some signals. For instance, I could want to study the behaviour of the engine load in a specific engine speed range: in this simple case, the trigger condition is the selected engine speed range. Moogle will then study each single measurement file in order to find every time that the engine speed is in the desired range. Of course, the advantage of writing a trigger is that the conditions can be more complex: there is no limit to the conditions that the user can specify, so that also very particular and rare engine conditions can be detected.

3. Trigger outputs: these are the outputs that Moogle gives to the user, extracted from each trigger window. Moogle allows to extract the maximum,

minimum and average of the desired channels in the trigger conditions. The important thing is that the outputs do not necessarily depend on the inputs: I can extract whatever signal in the output, even if that signal is not included in the trigger conditions.

Once the writing of the trigger is over, it can be saved in the *mgl* format. After this phase, the test phase must be carried out.

2.3.2 Trigger Test

The testing phase is of fundamental importance, because it allows to avoid to use a wrong trigger. ETAS provides a specific environment, called Moogle Test, that has characteristics very similar to the standard Moogle ones. The biggest difference, anyway, is the fact that Moogle Test can only study a limited amount of data, around 50 GB.

So the user must select very carefully the sample measurement files: they must contain the trigger conditions, so that he can ensure that the trigger recognizes them and extracts the selected informations. The test is indeed very similar to the standard Moogle also for what concerns the crawling: simply, the time required is obviously inferior in Moogle Test than in Moogle.

If the trigger successfully overcomes this phase, it can be converted.

2.3.3 Trigger conversion

As previously mentioned, the triggers are written in *mgl* format. The advantage of this format is mainly that it can be created by means of standard Windows text editors, such as Notepad. The drawback of this format is that Moogle is slower in computing it. So, instead of using the trigger in such a format, the trigger is converted in the fastest *dll* format. The computation speed increases without any extra cost, because the conversion tool has been developed within ETAS: Maserati can use it without any extra cost.

2.3.4 Indexing

Once the trigger has been converted, it is possible to start processing the data: this is possible thanks to the window, shown in Figure 2.2, where the different vehicles can be selected and then the processing can take place. As already said, the user is a passive element in this step, because the software must perform its computations before being able to deliver to user the final results. In order to let

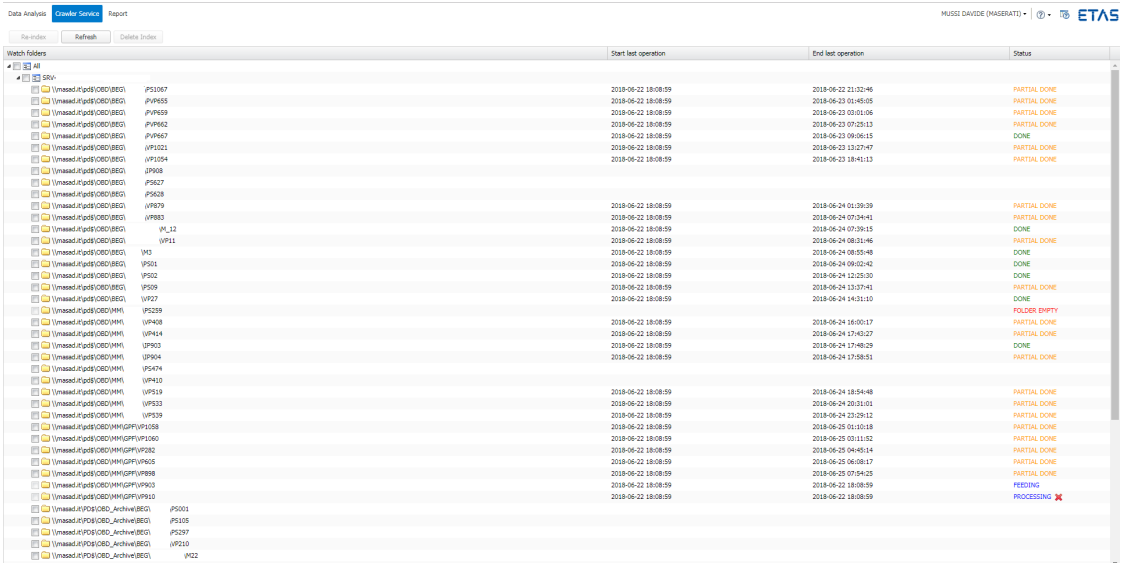


Figure 2.2: *Moogle crawler window: in this page it is possible to let the system start the indexing of the files*

the reader understand the time required for the indexing phase, actually Moogle needs several days and nights to process al the files stored on Maserati's servers, and as the time goes on, this time will gradually increases. Computation time depends in fact both on the amount and dimension of stored measurement files, and on trigger number and complexity.

In order to avoid this possible problem, a correct management of the indexing must be performed: it isn't necessary to process all the data contained within Maserati's servers, but only the ones needed for the requested statistic. Moreover, a correct management of the triggers can heavily improve this phase duration: eliminating the oldest and unused triggers can allow to save time, especially if those triggers were applied to a wide range of measurement files. Obviously, a backup version of those triggers must be kept, in order to avoid to loose the information contained in that trigger.

2.3.5 Creation of statistic reports

Once the indexing is finished, the user can finally create the report, containing the results of the processing, that will be eventually shared with the colleagues. The reports can be created in a specific section of Moogle, that can be seen in Figure 2.3. Once all the charts have been created, it is possible to save the report, so that they must not be created from zero every time; moreover, the reports automatically update the charts after every indexing phase. The saved reports can be seen in the left part of Figure 2.3.

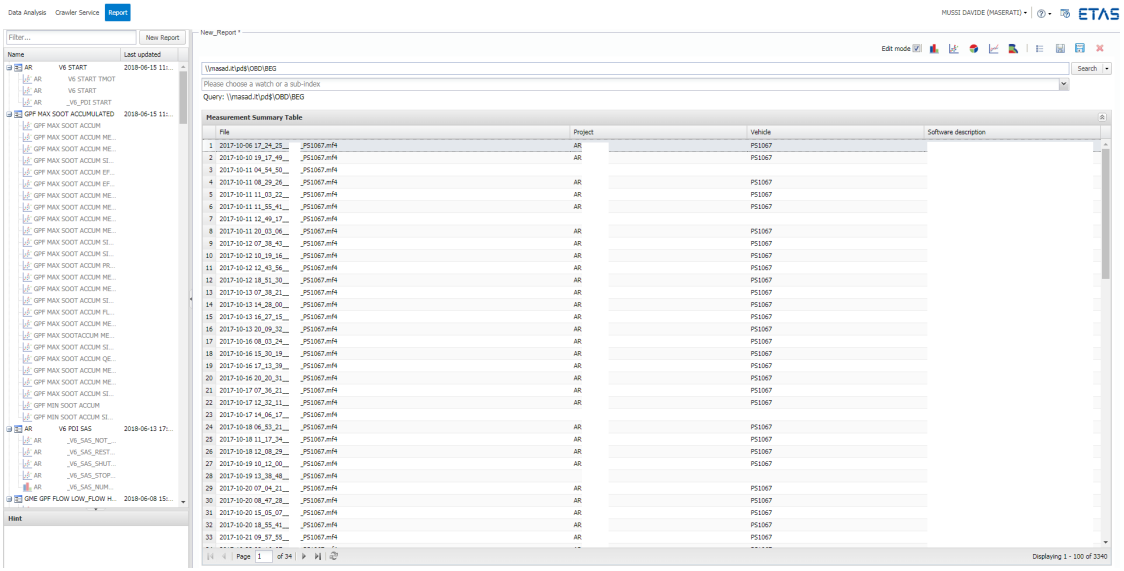


Figure 2.3: *Moogle report window: in this page it is possible to select the different vehicles and then to create the different charts*

Once the report is ready, meaning that all the charts are ready, Moogle allows the user to export these charts, so that they can be shared to the components of Powertrain department without having them to necessarily use Moogle; the software relies on an external site, High Charts, in order to deliver high level charts. The charts can be exported in different formats:

- Picture files (PNG or JPEG format).
- Excel files (CSV format).
- PDF files.

Within the company, usually the reports are delivered by means of a Excel file. This allows the colleagues to study the charts in a very effective way, since they can rapidly have access to all the points present in the chart. In fact, the bigger weakness of exporting the charts in PNG or JPEG formats is that from those pictures the values of the points in the charts aren't clear. Instead, in an Excel file, the colleagues can inspect immediately the values assumed by every point in the chart.

The Report page in Moogle allows the user to select what type of chart to create. Actually, these are the possible solutions:

- Scatter chart.
- Bar chart.

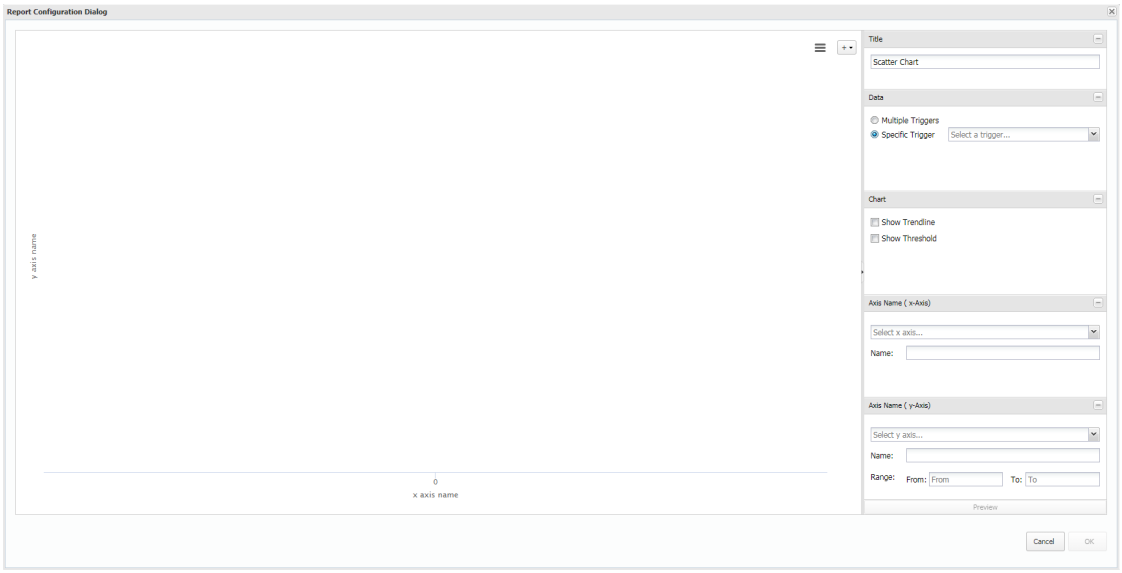


Figure 2.4: *Moogle scatter chart window: in this page it is possible to select the parameters to be plotted in a scatter chart*

- Pie chart.
- Heat map.
- Line chart.

After having chosen what type of chart he is going to create, the user can also choose what parameters to place on the y and x axis of the chart. The important thing is that those parameters must be set as output during the trigger's writing, otherwise it won't be possible to plot them: a modification to the trigger will be needed, and then a new crawling. The result is a big loss of time.

Every type of chart has some unique features that can be very useful, depending on what the user needs to show. For instance, in the scatter chart the user can apply a trend line; in the bar chart, a event counter can be applied, so that, for example, the chart will tell to the user how many times, or for how much time, the vehicles were in a certain specific conditions, determined in the trigger. These features allows the user to create more elastic statistics, and to ease the readability of the report for the colleagues.

Scatter chart

As can be seen in Figure 2.4, in this page it is possible to create the scatter chart. As first thing, the trigger must be selected: it is fundamental in order to be able to extract the data that are searched by a specific trigger. After the trigger's

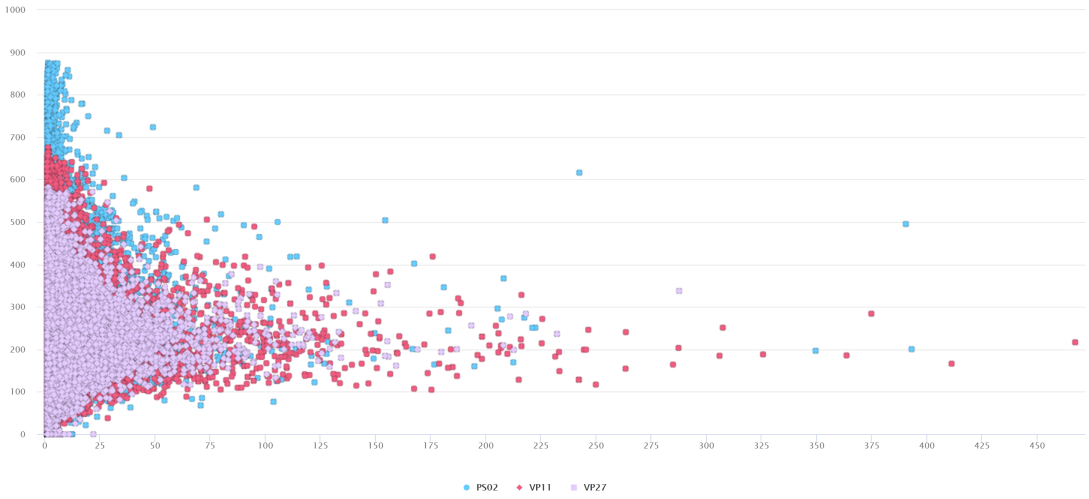


Figure 2.5: *An example of a finished scatter chart as it is visible on Moogle itself*

choice, it is finally possible to choose what parameters must be plotted on the chart.

It is possible to select the x axis, the y axis and also the z one: in the z axis it is possible to place a sort of vehicle's filters, that allows the user to distinguish, in the final chart, between the different vehicles, or even the software filter, and so on. So it is possible to create a highly customizable chart.

In Figure 2.5 it is possible to see how, once all the parameters are set, a scatter chart will be displayed in Moogle. It is then possible to see the filter based on the vehicles: every colour means a different vehicle, so that it is easier to understand which one has an abnormal behaviour, and consequently to investigate it. The chart here shown is purely an example: in fact the scale axis and parameters are not shown. I want only to let the reader understand how works Moogle and its chart's creation procedure.

Bar chart

For what concerns the bar charts, the window in which it is possible to select the parameters that are going to be displayed on the chart is essentially similar to the ones of Figure 2.4, so I won't replace it neither here or in the following part of this chapter.

In Figure 2.6 it is possible to see how a finished bar chart is displayed on Moogle. Also in this case it is possible to apply a sort of filter on the z-axis, just like for the scatter chart. Also in this example the applied filter was the one related to the vehicles.

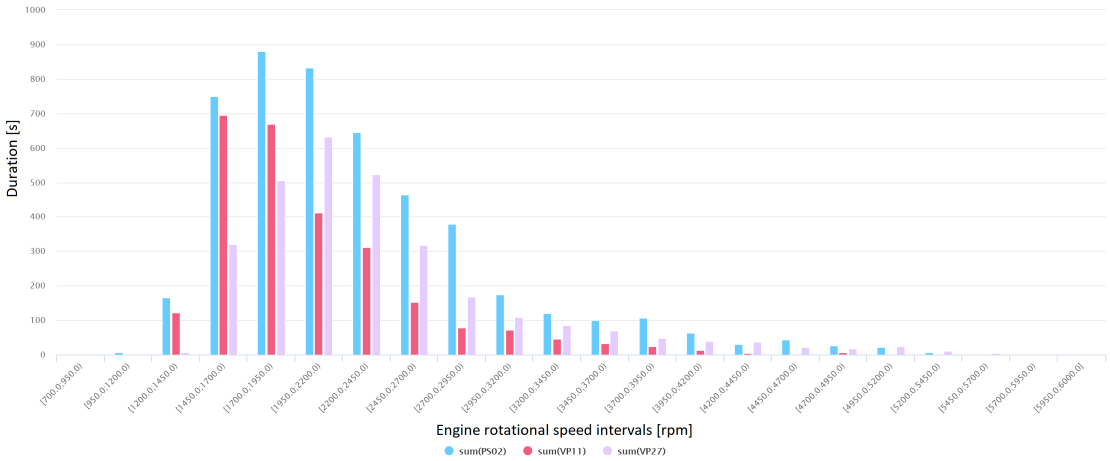


Figure 2.6: *An example of a finished bar chart as it is visible on Moogle itself*

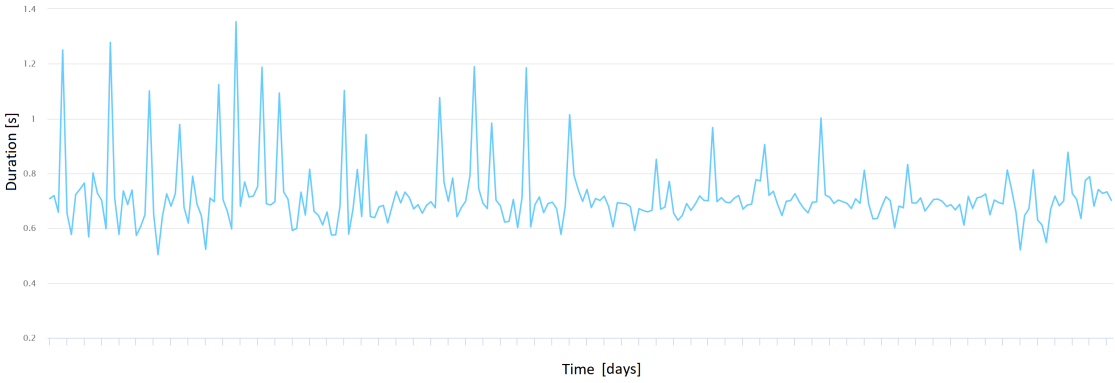


Figure 2.7: *An example of a finished line chart as it is visible on Moogle itself*

Line chart

The same speech that was done for scatter and bar charts is of course valid also for the line chart. The line chart can be basically be seen as a different way to plot the same information of a bar chart. For this reason, I want only to show in Figure 2.7 how this chart is reproduced on Moogle.

Pie chart

This type of chart is a type that usually is less requested within Maserati, essentially because it is easily creatable on Microsoft Excel. Anyway, Moogle gives the possibility to create directly within it a pie chart. The major difference is that this type of chart requires a specific trigger, that hardly can be used for any other type of chart. In the end, it is really due to this reason that is preferable, for me, to avoid doing pie charts within Moogle, while it is better to create the other types

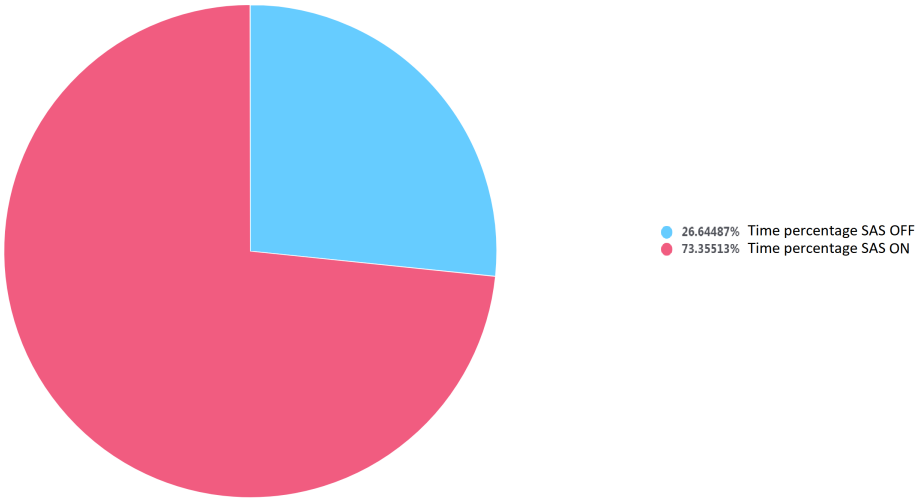


Figure 2.8: *An example of a finished pie chart as it is visible on Moogle itself*

of chart in Moogle, extract them as CSV files and then use those data to create a Pie chart in Excel.

In Figure 2.8 can be seen how this type of chart is reproduced within Moogle.

Heat map chart

The final chart that can be created is the heat map. The window in which the axis's parameters are selected is a bit different, but essentially is still similar to the one of Figure 2.4. The greatest difference is, beyond the fact that is mandatory to select a z axis (while in the scatter chart it was optional, I used it always only to have a way to track the different vehicles, softwares, and so on). Moreover, after the choice of the axis's parameters, it is also mandatory to choose the intervals for each axis, and also the step.

In Figure 2.9 it is possible to see how Moogle represents a heat map chart. It is also possible to see the intervals that have been specified by the user, and also the steps of each interval. It is of course a powerful tool, but as the pie chart it needs a specific trigger in order to create such a chart. In this case, by the way, it is a totally reasonable effort, because it is impossible to create a heat map internally in Excel.

2.3.6 Moogle application to calibration

Moogle can be used, as previously explained, in order to save a huge amount of time in the analysis of the measurement files. But it can also be used, with the same purpose, in a support activity to the calibration.

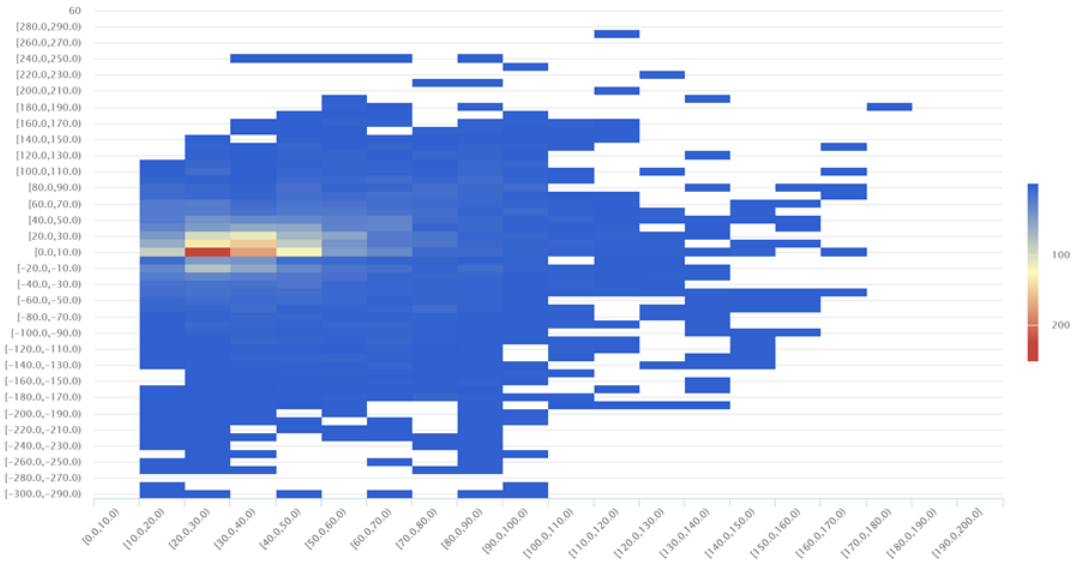


Figure 2.9: *An example of a finished heat map chart as it is visible on Moogle itself*

In fact, the standard calibration method requires the calibrators to develop a particular ECU map or function, and then to test it on some vehicles for some time, usually some days. After this phase, they have to study the measurement files in order to understand if their function worked properly, or if there is the need of a correction. And this phase goes on until the function isn't perfectly calibrated.

Thanks to Moogle, this approach changes totally. The calibrator can directly ask the Moogle user to create a trigger that reproduces the function or map he was working on. This is possible only thanks to his know-how on the topic, as he knows what are the parameters that must be studied. At the end of the indexing phase, a statistic can be created and delivered to the calibrator: he can then analyse the results and understand if everything worked correctly.

The use of Moogle as support in the calibration phase has the following advantages:

- The amount of data that are processed are enormous. Maserati's database contains years of measurement files of many different vehicles, so that the final statistic will have a huge sample base, while the calibrator could record data only for a few days and on a small number of vehicles. This leads to a statistic based result: if the calibration is validated thanks to a statistic based approach on a so wide amount of data, it means that the robustness of that solution is very high. This point is particularly important, because having a more robust solution means that its reliability, efficiency and effectiveness are more and more high, leading to a better vehicle, especially to the final

customer's eyes.

- There is a big time saving: the calibrator can study, in a preliminary phase, the results of his calibration on older vehicles, understanding then if the direction of his work is correct, allowing him to focus on that particular direction, without wasting time in a trial and error approach. Then he can test his final calibration on new vehicles, in order to have the final confirm of the goodness of his job.

The company is then trying to continuously increase the use of Moogle in this activities, especially keeping in mind that such an approach could eventually lead to the total disposal of the vehicle's fleet in the future. In fact, theoretically, it will be possible to calibrate the vehicles relying only on the data of the older ones, thanks to Moogle and its statistic approach. This will lead to a big economical saving for the company, that should use that saved money in order to improve many other fields, like for instance the IT infrastructure, improving then the Moogle indexing time, or on the development of new vehicles or technological solutions that could elevate the brand in the market.

2.4 Moogle management and development

During my stage period, I didn't have only to use Moogle in order to create statistics for my colleagues, but I was also responsible for the relationships between ETAS and Maserati. In fact, Moogle isn't still a complete software, so ETAS and Maserati are everyday in contact in order to improve the software, meaning the introduction of new features within Moogle or the bug-fixing of the software. The final purpose of this relationship between the two companies is to continuously improve the software, in order to make it more efficient, robust and user friendly. Moreover, I was responsible not only for the test of the new versions of the software released during my job period, but also for the suggestions of new features and bug fixing.

The development of the software follows the classic feedback system: I take note of some critical aspects of the software, bugs, or possible improvements, and report them to ETAS. The Deutsch company can then evaluate the requests that I propose, and eventually decide, beside the mandatory bug-fixing, what features could effectively be added to the software, and which ones instead cannot be implemented, due to too high economical or engineering effort.

I have different instruments in order to communicate with ETAS:

- Mail: Maserati provided me an official mail account, in order to manage the correspondence with my colleagues and ETAS technicians.
- Phone: Sometimes happened that I had to call ETAS in order to clarify, in short time, some points, or have some urgent informations.
- Bi-weekly WebEx Meeting together with two German ETAS technicians and two ETAS Italian coordinators dedicated to the Maserati and Alfa Romeo project. Usually it is during these meeting that we propose our desired improvements to the software, and listen to their feedback about the feasibility of the requests, and the work progresses.
- Face to face meeting: very rare occasion, but it can happen, especially when closer to a new Moogle release, that a face to face meeting becomes mandatory, in order to easily discuss the future of the software.

More than the creation of trigger and statistic, the development of the software is the biggest legacy that I left to the company at the end of my stage. As it will be explained in a following chapter, since my beginning of stage to the end, the software has been improved thanks to my requests and suggestions to ETAS. Of course, I have also created statistics that were used from my colleagues. For this reason, in the next sections of the thesis I will show a few detailed examples of statistics I created, and also the development of the software.

Chapter 3

Knock Event

3.1 Physical phenomenon

As it can be read in [10], knock is a form of abnormal combustion, that may occur in the last part of the process, when some part of the combustible mixture auto-ignites ahead of advancing flame front.

As shown in Figure 3.1a, usually the charge is regularly ignited by the spark plug with the correct timing. The first part of the combustion process develops in a normal way.

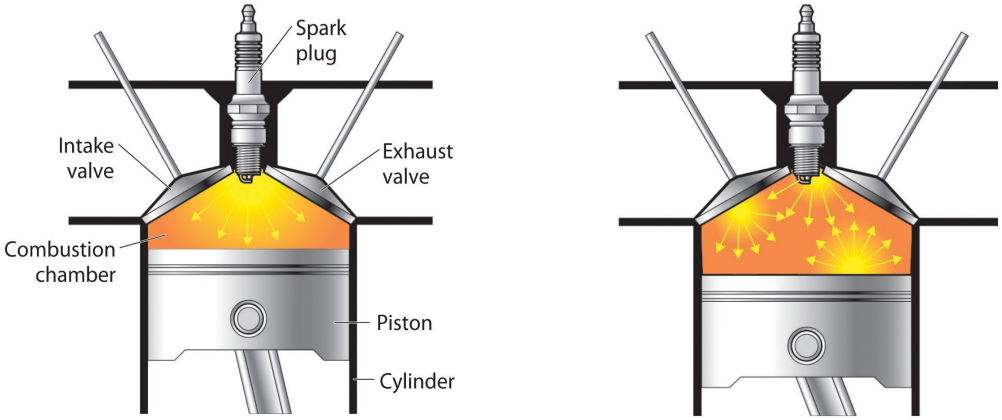
The volume of the mixture, ignited by the spark plug, is gradually transformed into a turbulent flame front spreading through the combustion chamber. The fuel-air mixture, ahead of the advancing front, is therefore progressively compressed and heated. Usually this mixture portion is also called *end-gas*, to point out that it is the part of the fresh charge that burns at the end of the process.

In sever engine conditions, it may happen that the end-gas is brought to such a high level of temperature and pressure, that one or more volumes of it suddenly auto-ignites before they are reached by the propagating front flame, as can be seen in Figure 3.1b.

This explanation of the knock phenomenon is confirmed by many experimental evidences, based on high speed films of knocking combustion.

The high release of energy due to the sudden combustion of a large mass of end-gas, produces a significant local increase of the gas pressure and temperature, causing a shock wave to propagate from the auto-ignition point through the combustion chamber.Then the pressure waves are repeatedly reflected by the walls and they create an oscillatory pressure trend versus time.

Indeed *knock*, in the proper sense of the word, is the name given to the metallic noise (a sort of hammering) produced by the vibrations of the engine components,



(a) *Normal Combustion Process* (b) *Knock Event*

Figure 3.1: *Normal combustion process vs knock event*

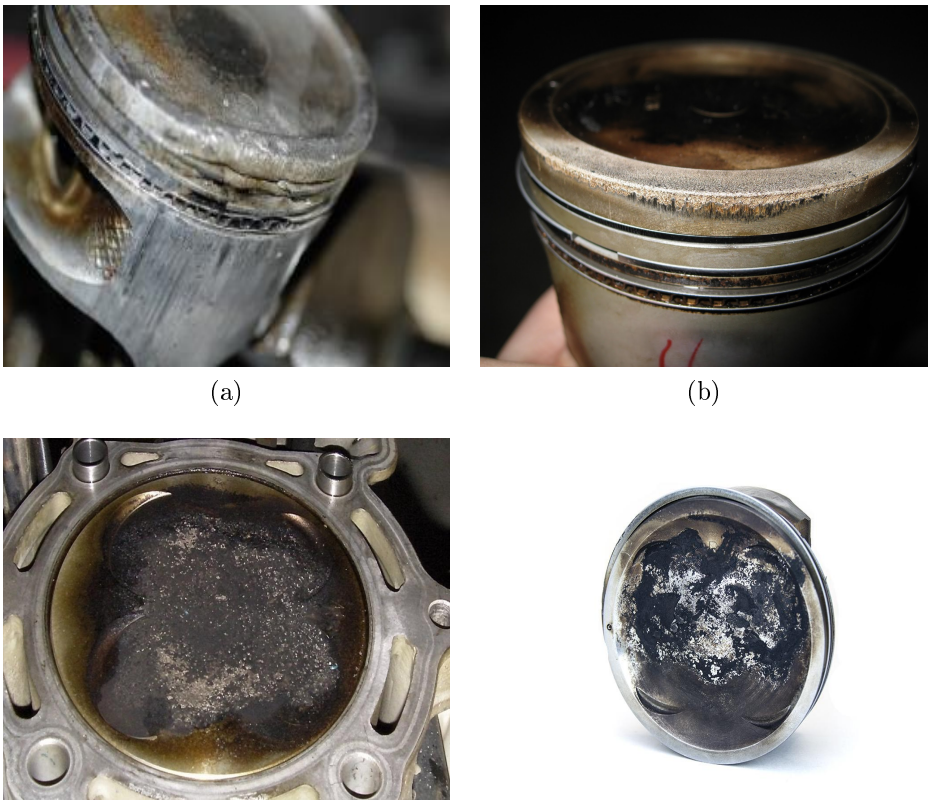
excited by the pressure oscillations inside the combustion chamber.

The frequency of these pressure oscillations depends on the velocity of the shock waves (about 1000 m/s) and on the distance covered (i.e the cylinder bore D) between consecutive reflections from the walls. Considering the bore values of current automotive engines ($D = 70 - 100mm$), the usual frequency is about 4-8 kHz.

Spark Ignition (SI) engines should never operate for a long time with knocking combustion, since pressure oscillations remove the boundary layer from the surfaces of the combustion chamber, increasing the outwards heat fluxes. Therefore, the increased thermal loads, together with the time variable mechanical load, may produce fatigue failure of some parts of engine components (piston rings and lands, cylinder head gaskets, piston crown, etc...), as it can be seen in Figure 3.2

Some help in avoiding knock is provided by all those factors that increase the auto-ignition resistance of the end-gas (high octane number of the fuel, lower compression ratio, reduced spark advance, decrease load, richer mixture, cooling of the fresh charge, more effective cooling of the end-gas, etc...) and accelerate the combustion process (higher engine speed, geometry producing more turbulence, smaller distance to travel by the flame front, etc...), so that the end-gas is ignited by the flame front before its auto-ignition.

Usually the knock risk limits the engine performances, since it limits the compression ratio and therefore the efficiency, and imposes the use of an expensive fuel of high octane number. Because of its importance, today this abnormal form of combustion in many advanced car engines is automatically controlled, using knock detectors based on the measure of engine structure vibrations, caused by knocking



(a) (b)

(c) (d)

Figure 3.2: *Damage examples due to knock event*

combustion. When the typical knock frequency is detected, the control system automatically changes some engine operating parameters (for example, reducing the spark advance and/or increasing the fuel air ratio), so that the combustion process become regular again.

3.1.1 Suppression methods for engine knock

As written in [11], conventional knock is a race between the flame in the engine and the thermal auto-ignition of the unburned "end-gas". The principle for avoiding engine knock is that the time of flame propagation to the end gas (τ_1) is less than the time of the end-gas auto-ignition (τ_2). Possible control strategies present in SI engines are summarized next.

Retarding spark timing, improving octane number and enriching mixture

Retarding spark timing, improving octane number and enriching mixture are commonly applied approaches for knock suppression in production engines without

component modifications. Retarding spark timing is the most effective method to simultaneously reduce the end-gas temperature and pressure. Lower end-gas temperature and pressure prolong the ignition delay (τ_2). However, late spark timing usually leads to an un-optimized combustion phase with lower thermal efficiency. It also may deteriorate engine performance due to the decrease in mean combustion chamber temperature and pressure.

Increasing the fuel octane number can be achieved by introducing antiknock additives to the fuel, to increase the time for the end-gas auto-ignition (τ_2) and then suppress knock.

Injecting excessive fuel could increase the effect of the charge cooling and decrease the temperature of the mixture, which could prolong the time to end-gas auto-ignition and finally suppress knock. However, under the condition of fuel enrichment, HC and CO emissions increase, and since the Three Way Catalyst (TWC) can only work under stoichiometric conditions, this causes problems of fuel economy and emissions.

Exhaust gas recirculation

Exhaust gas recirculation (EGR) is regarded as an effective method for suppressing knock in advanced SI engines due to the resulting increased τ_2. Furthermore, cooled EGR has more potential in knock mitigation without a loss of output power.

In order to quantify the effect that EGR has on knock intensity, a series of experiments have been performed. In [11] it is written that the improvements in effective AKI (anti-knock index) of the fuel from using EGR is about 0.5 RON per % of EGR for commercial grade gasoline fuels. Higher levels of EGR (EGR $>$ 15%) have been shown to have a significant effect on engine efficiency by reducing knock, leading to improved combustion phasing and the ability to operate at higher loads at compression ratios above 12. High temperature internal EGR has a disadvantageous effect on knock suppression because of its heating effect, which should be avoided for gasoline engines.

Stratified mixture

Use of a stratified mixture is a flexible approach to suppress knock and is usually achieved using direct injection. Fuel directly injected in the cylinder can lower the combustion temperatures due to the cooling associated with fuel vaporization. Therefore, the cooling can lower knock sensitivity and the compression ratio can be increased.

Use of stratified stoichiometric mixture has been demonstrated. Using a two-stage injection strategy in gasoline direct injection engine, , Stratified Stoichiometric Mixture (SSM) could suppress knocking effectively compared with a Homogeneous Stoichiometric Mixture (HSM) case. Investigations of a two-stage strategy indicated that locally slightly rich mixtures due to a late injection helps to suppress auto-ignition by providing a local cooling effect and by enhancing the local flame propagation speed. The assessment concludes that the two-stage injection is an effective method for suppressing knock under high load operating conditions. However, the injection strategies and combustion systems need to be carefully designed and optimized to avoid soot emissions.

Turbulence

Increasing turbulence intensity usually suppresses engine knock. Once combustion has started, increased turbulence leads to fast combustion and decreases the tendency to knock. However, at the start of combustion, increased turbulence also increases heat transfer from the spark electrode. To solve this problem, modern engines adopt high energy ignition to support strong turbulence. Introduction of high tumble-induced-high-turbulence is accompanied by low tendency to knock.

Cooling

Cooling of the combustion chamber walls is also an effective approach to reduce end-gas temperatures. An effective method to suppress knock is changing the coolant flow patterns to improve the distribution of wall temperatures. However, increased heat transfer lowers the thermal efficiency. To suppress knock without a considerable loss of thermal efficiency, only the upper portion of the exhaust side bore should be cooled, and foam rubber can be utilized to provide heat insulation to maintain the temperature of the center and lower portions of the bore.

Cooling the intake charge is also useful for knock suppression. The initial condition of the end-gas during a cycle is the intake air temperature. In addition, cooling the exhaust manifold is also important. A water-cooled exhaust manifold (WCEM) with high cooling capacity could be adopted to reduce the temperature of the exhaust valve surface and the EGR so as to suppress knock.

Lowering effective compression ratio

Variable valve timing (VVT) is a practical way to charge effective compression ratio at relatively low cost for the different engine operating regions. Late intake

valve closure (LIVC) is commonly used at high load to achieve a lower effective compression ratio to avoid knock.

Using a variable compression ratio (VCR) is a more ideal method, but the complex structure makes the cost increase significantly.

3.1.2 Knock control system

As it is discussed in [12], electronic control of the moment of ignition offers the possibility of accurate control of the ignition angles as a function of rotation speed, load, temperature, etc. Nevertheless, if there is no knock control, there must still be some means to define a clear safety margin to the knock limit.

This margin is necessary to ensure that, even in the most knock-sensitive case with regard to engine tolerances, engine ageing, environmental conditions and fuel quality, no cylinder can reach or exceed the knock limit. The resulting engine design leads to lower compression, retarded moment of ignition, and thus worsening fuel consumption and torque.

These disadvantages can be avoided through the use of knock control. Experience shows that knock control increases engine compression. This results in lower fuel consumption and higher torque. Now, however, the pilot control ignition angle no longer has to be determined for the conditions most sensitive to knocking but rather for the conditions least sensitive to knocking (e.g compression of the engine at the lowest tolerance limit, best possible fuel quality, cylinder less sensitive to knocking). Each individual engine cylinder can now be operated throughout its service life in virtually all operating ranges at its knock limit, and thus at optimum efficiency. For this type of ignition angle adjustment, a reliable method of knock detection is essential. It should detect knock for each cylinder throughout the engine's operating range starting from a specified knock intensity.

A knock control system consists of:

- Knock sensor.
- Signal evaluation.
- Knock detection.
- Ignition angle control system with adaptation facility.

Knock sensor

A typical symptom of combustion knock is high-frequency vibrations which are superimposed on the high pressure curve in the combustion chamber. These vibra-

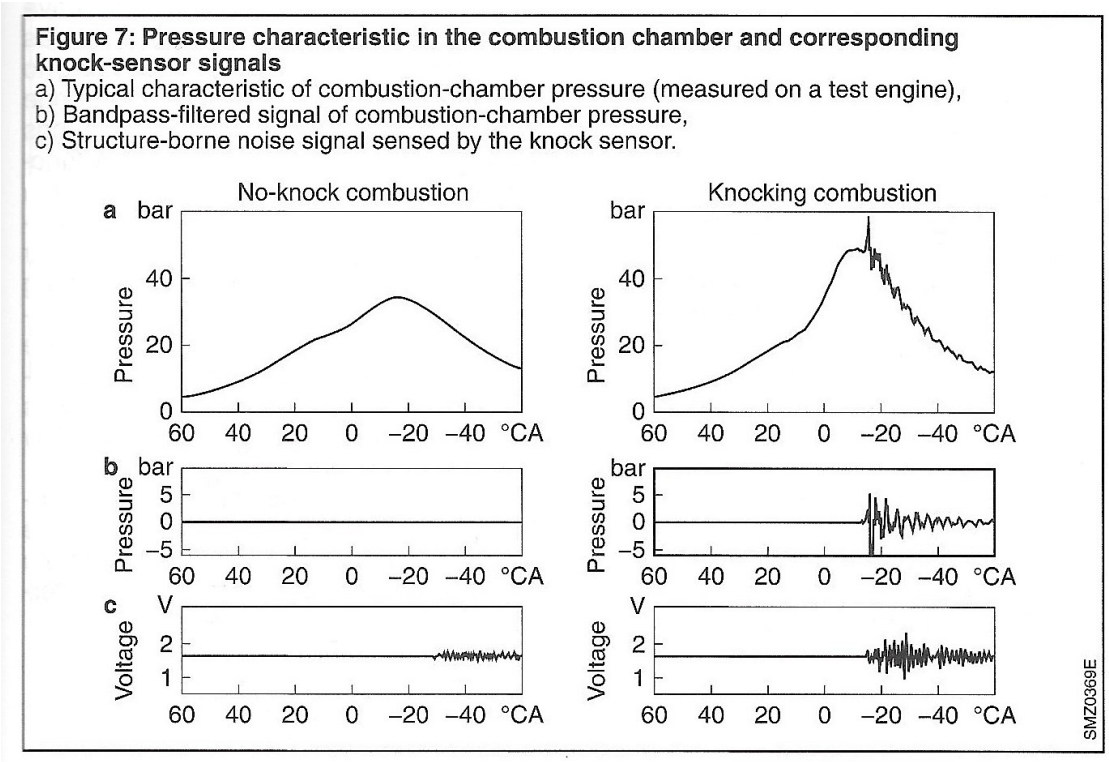


Figure 3.3: *Pressure curves for normal and knocking combustion[13]*

tions are best detected directly in the combustion chamber by means of pressure sensors. As fitting these pressure sensors in the cylinder head for each cylinder is still relatively costly, these vibrations are usually picked up using knock sensors fitted to the exterior of the engine. These piezo-electric acceleration sensors pick up the characteristic vibrations of knocking combustion and convert them into electric signal.

There are two types of knock sensor. A wide-band sensor, with a typical frequency band of 5 to 20 kHz, and a resonance sensor, which preferably transmits only one knock-signal resonance frequency. When combined with the flexible signal evaluation system in the control unit, it is possible to evaluate different or several resonant frequencies from one wide-band knock sensor. This improves knock detection performance, which is why the wide-band knock sensor is increasingly replacing the resonance sensor.

To ensure sufficient knock detection in all cylinders and across all operating ranges, the number and location of the required knock sensors must be carefully determined for each engine type. Four-cylinder in-line engines are usually fitted with one or two knock sensors, while 5- and 6-cylinder engines are fitted with two, and 8- and 12-cylinder engines with four knock sensors.

As it can be read in [13], the knock sensor is in principle a vibration sensor.

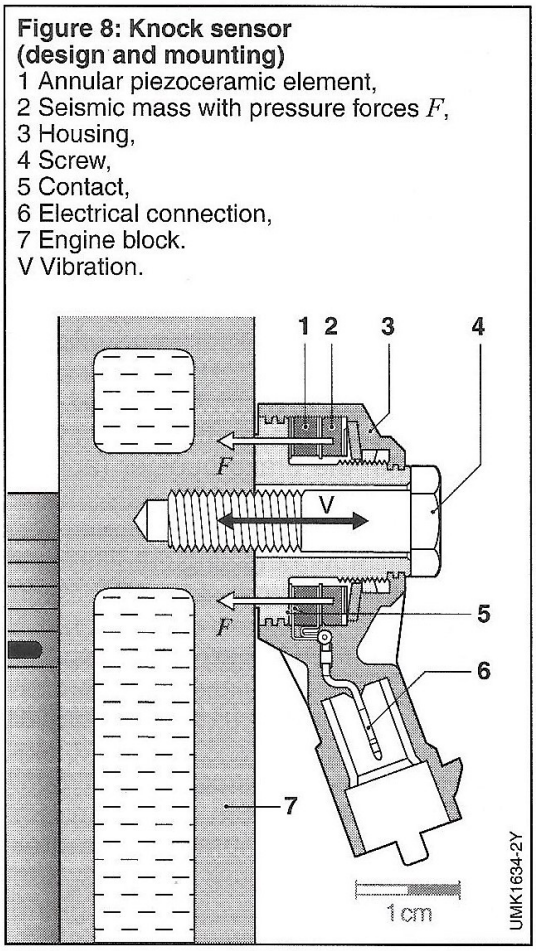


Figure 3.4: *Piezoelectric knock sensor[13]*

It senses structure-borne sound vibrations that occur as "knocking" in a gasoline engine during uncontrolled combustions, as can be seen in Figure 3.3. The structure-borne sound vibrations must be able to be introduced from the measuring location on the engine block undamped and without resonance into knock sensor. A suitable measuring point and a fixed screw connection are required for this purpose.

The knock sensor is screwed to the engine block, as it can be seen in Figure 3.4. A seismic mass exerts as a result of its inertia pressure forces in time with the exciting vibrations on an annular piezoceramic element. These forces induce a charge transfer within the ceramic element. This generates between the upper and lower sides of the ceramic element a voltage that is tapped via contact disk. The knock signal generated in the sensor is forwarded to the engine ECU, where it is conditioned and evaluated. When knocking noises are detected the engine ECU retards the ignition timing and thus counteracts any further knocking.

Signal evaluation

Always in [12] it is written that for the duration of a timing range in which knock can occur, a special signal evaluation circuit in the control unit evaluates from the wide-band signal the frequency band(s) with the best knock information and generates a representative variable for each combustion process. This very flexible signal evaluation of the wide-band sensor enables high detection quality. When using a resonance knock sensor that transmits just one resonant frequency for analysis of all cylinders assigned to it across the entire engine map, knock detection is normally no longer possible at higher engine speeds.

Knock detection

The variable produced by the signal evaluation circuit is classified in a knock-detection algorithm as "knock" or "no knock" for each cylinder and for each combustion process. This is done by comparing the variable for the current combustion process with a variable which represents combustion without knock.

Ignition-angle control system with adaptation facility

If combustion knock is detected in a cylinder, the moment of ignition for that cylinder is retarded. When knock stops, the moment of ignition is advanced again in stages up to to pre-control value. The knock-detection and knock-control algorithms are matched in such a way as to eliminate any knock that is audible and damaging to the engine, even though each cylinder is operated at knock limit within the optimum efficiency range.

Real engine operation produces different knock limits, and thus different moment of ignition for individual cylinders. In order to adapt pre-control values for the moment of ignition to a particular knock limit, the ignition retard values are stored for each cylinder dependent on the operating point. They are stored in non-volatile program maps in the permanently powered RAM for load and engine speed. In this way, the engine can be operated at optimum efficiency for each operating point and without audible combustion knocks, even if there are rapid load and engine-speed changes.

This adaptation even enables the use of fuels with lower anti-knock properties (e.g regular instead of premium grade petrol).

3.2 Knock event monitoring

Within Maserati, Alfa Romeo and other car makers, the knock phenomenon is always monitored, in order to avoid the disastrous consequences of a heavy knock on the engine components. Due to the fact that the easiest way to prevent knock is to retard the spark timing, because it can be done easily by means of the electronic control, the main parameter used in order to study and monitor the knock is in fact the spark advance. Moreover, as support to this analysis, also a statistic related to the ongoing of the noise within each cylinder was performed: in fact, in presence of knock phenomenon, within the engine can be measured a higher noise than the one occurring during standard combustion processes.

3.2.1 Retard of the spark advance

I had to create a statistic that showed the behaviour of this parameter on the 200/280 HP Alfa Romeo engines. This parameters is acquired and then monitored on all the four cylinders of the engine. As previously explained, the first part of the process is the creation of the trigger. In this case the trigger is pretty easy and simple, because I just wanted to see, in every measurement file, what is maximum value of the retard of the spark timing. The trigger I created is the following one:

```
1 KNOCK_MONITORING := detect/time{RPM >= 0};
  KNOCK_MONITORING += max RETARD_SPARK;
```

As it can be seen, it is a very straightforward trigger. The trigger condition is the one enclosed in the brackets, while the output of the trigger is the maximum value of the retard of the spark timing. This trigger will give back to the user, for every measurement file, a single value of retard. Actually there isn't a command in Moogle that allows to automatically take the maximum, or minimum, values of some parameter over the whole measurement file. For this reason a small workaround is necessary: asking Moogle to consider when the rotational speed of the engine is greater and equal than zero means, obviously, to take the whole measurement file.

The results of this analysis can be seen in Figure 3.5. The different colours of the points indicate the different vehicles of Validation Fleet. The first thing that must be noticed about this chart is that there isn't effectively, on the ordinate axis, the values, in angular degrees, of the retard of the spark timing. In fact, it is a sensible data for the company, so I had to do a slight translation into another reference system. So I transformed the retard of the spark timing into the quality of spark advance: it is a dimensionless parameter, ranging from zero to ten, that

Figure 3.5: *Quality of Spark Timing within Cylinder 1*

indicates the quality of the spark advance. Quality equal to ten means that the cylinder is in the ideal spark advance situation, leading to a very good combustion. As this value decreases, the spark advance situation is worsened.

The knock phenomenon can occur when this quality tends to zero. Usually, values below two lead to often have knocking combustion processes. As it can be seen in Figure 3.5, Fleet vehicles never goes below this threshold value. This situation was not completely satisfactory for the company, because, even if the quality was always over the threshold, initially it was also close to it. So a safety factor, in order to avoid such unpleasant situations, was requested.

Due to this reason, the vehicles were stopped around the beginning of March, in order to install on the ECU a new version of the engine software, as underlined by the orange vertical line. This intervention effectively led to an improvement of the Spark Timing situation: as it can be seen in Figure 3.5, after the update the Quality of Spark Timing increased, reaching values higher than four.

After having achieved this result, the company worked in order to further improve the behaviour of the engine, in order to improve more and more the safety factor. So the vehicles were again stopped toward the end of March, as the red line clearly shows, to install a second revision to the engine's software. The results are clear: the quality of the spark advance increased again, and no vehicle went below the quality value of six.

In this analysis I monitored the behaviour of all the four cylinders but, due to

the fact that the four charts were very similar, I have shown only the one related to the cylinder 1. The same considerations are valid also for the other three cylinders.

3.2.2 Engine Noise

As previously told, another parameter that can be used as a support tool to the analysis of knock is the noise within the engine's cylinders. Due to the fact that knock is an abnormal combustion that happens when some part of the mixture auto-ignites ahead of the flame front, the noise produced within the cylinder will be greater than the one of a normal combustion process. Of course, this parameter can't be used as a primary source of information about the knock in the engine, because there are many factors that contribute to the total final noise. But, after having studied the retard in the spark advance, the noise can give us some useful hint and, possibly, confirmations about knock's behaviour.

The original output of the noise sensor within each cylinder is in mV, so that it must be translated to the more common dB. But, as for the retard in the spark advance, also in this case I had to transform the original data in order to be able to show it outside the company. So, from the original values of noise expressed in mV, I passed to a noise level expressed in a dimensionless scale, ranging from zero to one hundred. The most important thing is that in this case the value one hundred of this scale represents the translated full scale of the instrument. So a value of one hundred means that a very bad and abnormal combustion is happening within the cylinder, and for this reason it must be absolutely avoided.

The results of this analysis can be seen in Figure 3.6. Also in this case the same consideration done for the quality of spark timing is valid: it is reported only the chart related to the cylinder one, and not all the four charts of the cylinders. This because the behaviour of the four cylinders is very similar, so, in order to not place too many almost identical charts, only one has been placed.

Also in this chart three different zones can be noticed. The first one, that ends toward the beginning of March, is characterised by values of noise far from the full scale, but too high. In this zone we can find a sort of correlation with Figure 3.5: as the quality of spark timing was sufficient but the safety factor wasn't high enough, so the noise level within the cylinder is good, but still too high in terms of a safety factor.

The second zone of the chart can be found after the first intervention on the engine's software, happened at the beginning of March. Also in this case, Figure 3.5 and Figure 3.6 lead almost to the same consideration: the software's update effectively was able to improve the combustion process within the chamber, and so also

Figure 3.6: *Linearised noise level within Cylinder 1*

the noise, as the quality of the spark timing, were improved.

The third zone, finally, still confirms us the goodness of the second software's update, scheduled at the end of March. For what concerns Figure 3.6, the noise level was decreased below the value of sixty: according to my colleagues and company directives, it is totally an acceptable value.

3.2.3 Analysis considerations

This analysis was performed over the whole life of the fleet vehicles. This because knock is a very dangerous phenomenon, and requires a constant monitoring. Standard monitoring techniques imply to manually open one by one the measurement files: this leads to a consistent loss of a time.

Thank to the use of Moogle, it was possible to carry out the analysis in short time and, moreover, to have it continuously updated. As the new measurement files were uploaded, Moogle immediately processed them, so that the chart was automatically updated with the last data.

Moogle's use led also, for the calibrators, to a fast action against the problem: in fact, knock was initially monitored only in order to avoid big and disastrous damages to the engine. Then, toward March, a more heavy and meticulous approach was applied. Every day I was requested to provide to my colleagues the updated statistic, so that they could evaluate the behaviour of the vehicle and then think to a new improvement. For this reason, only from March it can be seen an

effective improvement of the vehicle's parameters, while the fleet was indeed active since January.

The use of Moogle was fundamental in providing to the colleagues a day by day detailed statistic, and also in helping to shorten the development times. As final conclusion, this led to a save of time and, of course, money for the company. Beside obviously the fact that the vehicle are now more reliable and safe for what concerns the knock phenomenon:this is highly important for final customer satisfaction.

Chapter 4

Gasoline Particulate Filter

4.1 GPF General description

Gasoline particulate filters (GPF) are an emission aftertreatment technology based on diesel particulate filters (DPF), developed to control particulate emissions from gasoline direct injection (GDI) engines[14].

As it can be read in [15], over the last years, the Gasoline Direct Injection (GDI) technology has been boosted as a result of EU climate policy and regulatory drivers towards reducing CO2 emissions from passenger cars. 40% of new non-diesel passenger car registrations in the EU were GDIs in 2015 as indicated by the International Council on Clean Transportation (ICCT).

Always in [14] it is written that the European PN standards, both effective for new types of GDI cars from September 2017, are:

- A PN limit of 6×10^{11} #/km over the NEDC / WLTC test cycle (Euro 6c).
- RDE (Real Driving Emission) testing for PN emissions with a conformity factor of 1.5, i.e., PNRDE = 9×10^{11} #/km (Euro 6d-TEMP).

The above standards could also be met, at least in certain types of vehicles, via in-cylinder controls such as fuel injection strategies, without particulate filters. However, the GPF has several advantages compared to in-cylinder controls:

- Effectiveness under all operating conditions: while in-cylinder strategies tend to be more effective under certain modes of operation, the GPF provides PN emission control under all engine operating conditions. An advantage that is especially important in RDE testing.

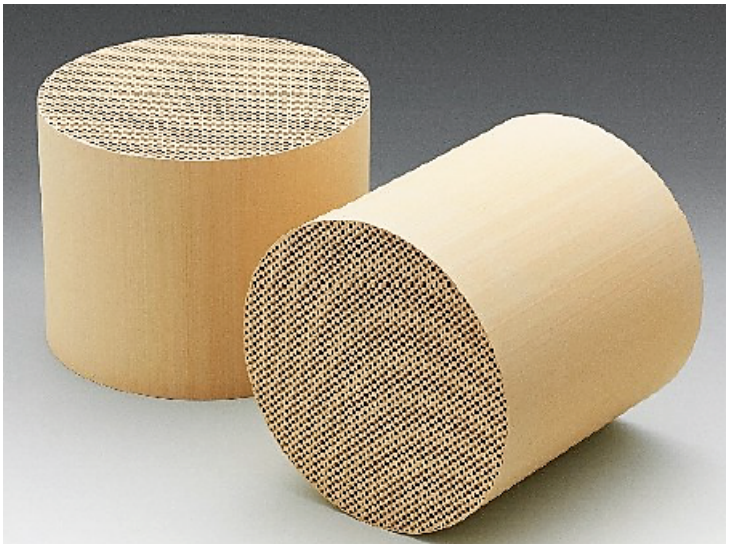


Figure 4.1: *Gasoline Particulate Filter (GPF) [15]*

- Control of emissions from engine faults: increased PN emissions can occur as a result of engine faults and malfunction, such as increased lube oil consumption. These emissions can be effectively controlled by particulate filters.

- Control of unregulated emissions: the GPF can control certain unregulated emissions, including polynuclear aromatic hydrocarbons (PAH). GDI engines, even though equipped with three-way catalysts (TWC), may produce significant levels of toxic PAH emissions.

For these reasons GPF have been developed in order to filter out the ultra-fine particles from the exhaust gases.

4.1.1 GPF structure

On [16] it is written that GPFs look very similar to DPFs and, generally speaking, work in a similar way. The filter has a honeycomb structure, usually made from cordierite, a synthetic ceramic, with alternately sealed inlet and outlet channels. The exhaust gas is forced to flow through the porous filter substrate, which traps the soot. At about 200 to 350 channels per square inch, the canal density of the GPF is nearly the same as a DPF. The major difference between the two types of filter is that the porosity of the GPF is higher because the substrate is lighter. Although this allows the gas to move more easily across the substrate, it also means the GPF is more fragile than a DPF.

An example of a GPF can be seen in Figure 4.1

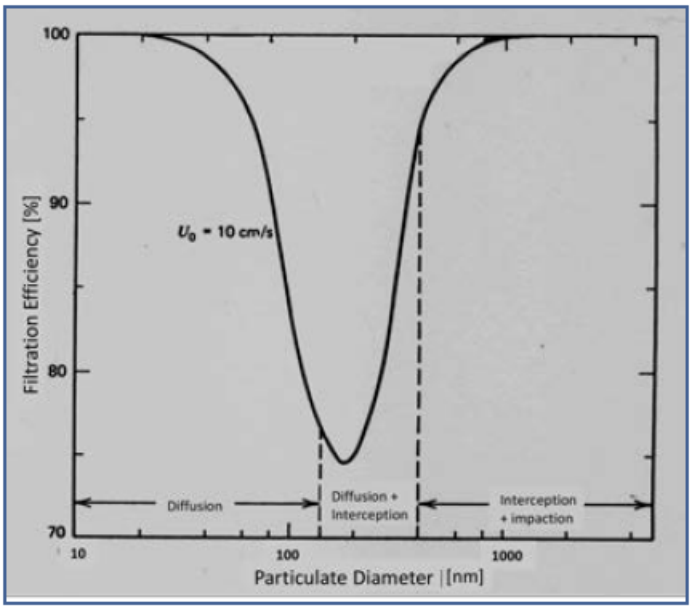


Figure 4.2: *Trapping mechanisms as a function of particle size, Aerosol Technology, William C. Hinds*

4.1.2 GPF trapping mechanism

Always in [15], it is said that for GPFs, like for wall-flow Diesel Particulate Filters, there are three particulate trapping mechanisms: interception, impaction, and diffusion. The trapping mechanisms depend on the particle size. The smaller particles are trapped by diffusion, the larger particles are trapped by interception and impaction. As a consequence, the initial filtration efficiency of the new GPF varies for different particle sizes. The smaller and bigger particles are all trapped; the lower filtration efficiency is observed for particles of around 200 nm in diameter, as it is shown in Figure 4.2.

4.1.3 Regeneration

These type of filters are very efficient, but over the time the soot is trapped inside the filter, and it can act like a plug, increasing so the backpressure. The best way to remove the soot from the filter is to burn it off: this can be done in presence of oxygen and at a temperature higher than $600°C$.

From [16] it can be read that unlike diesel engines, where oxygen is in excess, gasoline engines generally run at stoichiometric mixture, which means there is no oxygen in the exhaust to burn off the soot when the engine is under high load.

Consequently, for gasoline engines, regeneration can only be effective for non-power conditions and so is performed under deceleration, when the engine is being motored, which results in oxygen being pumped though the engine. Another major

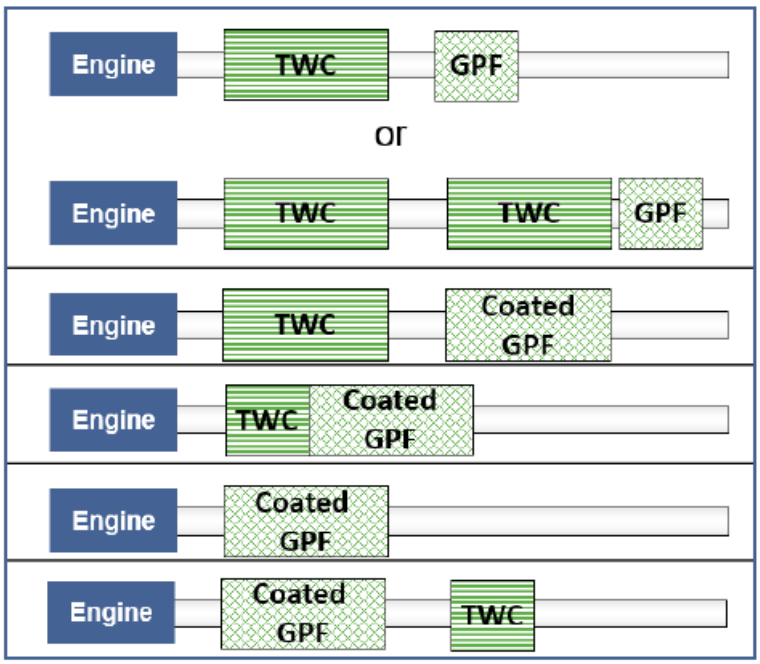


Figure 4.3: *Possible system architecture*

difference in gasoline engines is that the regeneration is passive, i.e. there is no need to purposely increase the exhaust temperature.

To initiate regeneration, the catalyst converter is fed with air for short periods. This oxygen, combined with high exhaust temperatures (400°-700°C), leads to soot ignition. Where engines operate for long periods without deceleration, for example driving on a traffic-free motorway without any downhill slope, engine control is required to initiate regeneration. In this case, the exhaust temperature is increased by delaying the spark timing and oxygen is made avaliable by creating a lean fuel/air mixture.

In [17] it is written that a continuously or near-continuously regenerating PM filter never builds up a soot layer which has the advantage of keeping back pressure to a minimum but the disadvantage of a slightly lower filtration efficiency because the presence of a soot layer actually increases the filter efficiency.

4.1.4 GPF positioning within the vehicle

In [16] it is said that generally the GPF is integrated in the three-way catalyst (TWC) downstream of the normal substrate, which then becomes a four-way catalyst(FWC): coating the GPF with the three-way catalyst (TWC) allows some substitution of the TWC volume, so that the packaging space and cost are reduced. An alternative is to keep a TWC close to the cylinder head and introduce a 2nd four-way catalyst under the flooring.

Every solution has of course its advantages and drawbacks. Integrating the GPF in the TWC leads into having a continuous soot clean out, at the cost of needing a smaller TWC system in order to avoid the GPF to be constrained by packaging. Keeping the GPF apart from the TWC means that there is no need of modifying the TWC system, but the drawback is represented by a inconstant soot clean out.

These two solution aren't the only one achievable: there is a wide variety of possible solution, as shown in Figure 4.3. It is up to the car manufacturer the choice of what one to implement on the vehicle.

4.2 GPF main parameters monitoring

Actually Maserati is testing on part of the 200/280 HP Alfa Romeo fleet the behaviour of these type of filters, in order to place them on the next model year vehicles. The company has never used these filters so, beyond assessing the behaviour of these components, it's also monitoring different parameters, in order to increase the internal know-how about this topic.

The most important parameters that are kept under monitoring are the following ones:

- Filtering efficiency.
- Delta pressure across the filter.
- Amount of soot accumulated within the filter.
- Flow resistance through GPF.

In order to study these parameters, I had to write a trigger. Thanks to that trigger, Moogle was able to process al the data available within Maserati's database and, per each measurement file, it extracted both the maximum and the minimum values of the soot accumulated within the GPF. Moreover, Moogle was also able to extract the values of the aforementioned parameters in the correspondence of the maximum and minimum values of the soot, in order to create a direct relationship between the all the parameters.

The charts shown in the next part of the chapter are obtained thanks to the following trigger:

```
GPF_MONITORING := detect/time {RPM >= 0};
GPF_MONITORING += max SOOT;
GPF_MONITORING += min SOOT;
```

```
4
GPF_MONITORING += {
6       GPF_MONITORING_MAX_SOOT := detect/time {SOOT = max(SOOT)
            };
        GPF_MONITORING_MAX_SOOT += start_value GPF_EFFICIENCY;
8       GPF_MONITORING_MAX_SOOT += start_value DELTA_PRESSURE;
        GPF_MONITORING_MAX_SOOT += start_value EXHAUST_GAS_FLOW;
10      GPF_MONITORING_MAX_SOOT += start_value RESISTANCE_GPF;
        GPF_MONITORING_MAX_SOOT += start_value SOOT;
12                  };

14 GPF_MONITORING += {
        GPF_MONITORING_MIN_SOOT := detect/time {SOOT = min(SOOT)
            };
16      GPF_MONITORING_MIN_SOOT += start_value GPF_EFFICIENCY;
        GPF_MONITORING_MIN_SOOT += start_value DELTA_PRESSURE;
18      GPF_MONITORING_MIN_SOOT += start_value EXHAUST_GAS_FLOW;
        GPF_MONITORING_MIN_SOOT += start_value RESISTANCE_GPF;
20      GPF_MONITORING_MIN_SOOT += start_value SOOT;
                    };
```

The first information I extracted from the database with Moogle was the ongoing of the soot accumulated within the filter during the life of the vehicle. In Figure 4.4 and 4.5 the behaviour respectively of the maximum and minimum of soot can be seen. As already told in Chapter 3, due to the restrictions that the company applies on the data regarding its vehicles, I couldn't directly place inside the thesis the original data, but I had to modify them in order to not show the original values of each parameter. So, in order to accomplish this company's request, the soot accumulated within the GPF is expressed in percentage, instead of *mg*. This percentage has been obtained by dividing the real value of the soot by a critical value, that must never be reached.

Of course, for what concerns Figure 4.4, it can be noticed that the maximum percentage values are always below the 50%: it means that the passive regeneration of the filter is able to keep it far from the critical value, ensuring that the GPF is a robust solution for Alfa Romeo Giulia and Stelvio: in fact, it was never applied an active regeneration to the filter.

In order to say this, I wrote a trigger that checks a channel that has the role to say if there is an active regeneration: every time that there is an active regeneration the value of this channel increases by one unit. The trigger never found any

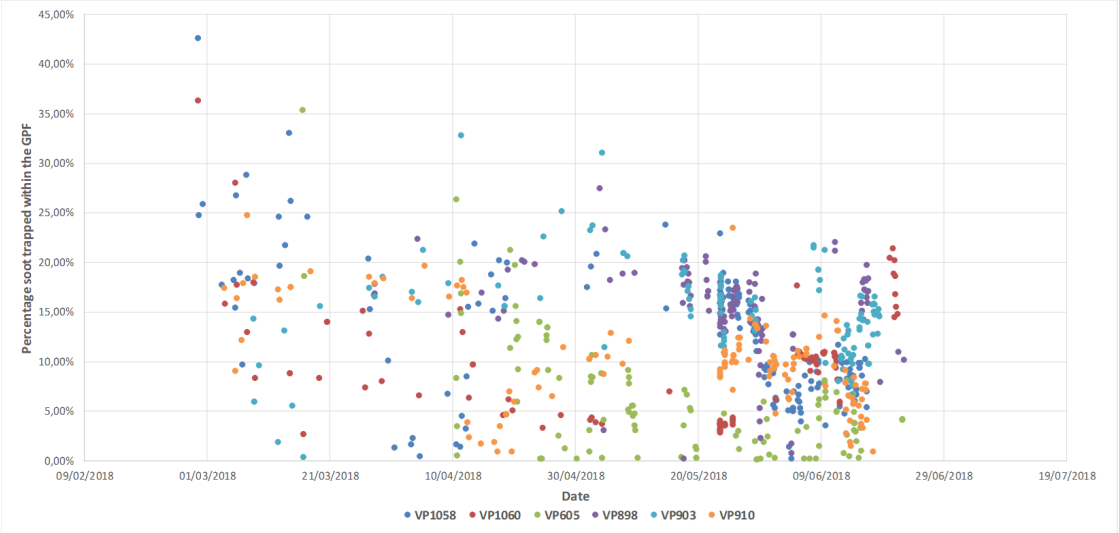


Figure 4.4: *Maximum amount of soot trapped within the filter, expressed in percentage*

occurrence, so it means that an active regeneration was never applied. Here the trigger logic can be found:

```
1       GPF_ACTIVE_REG := detect/time{ACT_REG > 0};
        GPF_ACTIVE_REG += max SOOT;
```

On the other hand, in Figure 4.5 it can be noticed that there is always a minimum amount of soot left within the filter. This makes perfectly sense because, as it will be shown in the following part, the filtering efficiency directly depends on the amount of soot trapped in the filter. In order to have a high efficiency, there must always be a certain amount of soot within the filter.

The charts that will be shown in the next sections of the thesis have been obtained considering the maximum values of soot trapped within the GPF, and not the minimum ones, as they are higher, of course, than the minimum, and consequently some filter's behaviours are easier to be seen.

4.2.1 Delta pressure across the filter

Before starting to show the results obtained with Moogle, it's necessary to introduce a couple of concepts. As it is written in [18], the *back pressure* is the pressure of the exhaust gases generated by the engine in order to overcome the hydrodynamical resistance of the exhaust system, so that the gases can be expelled into the atmosphere. In the case of a turbocharged engine the *back pressure* is the pressure in the exhaust system at the turbine's outlet, while in the case of naturally aspirated engines it is the pressure at the exhaust manifold.

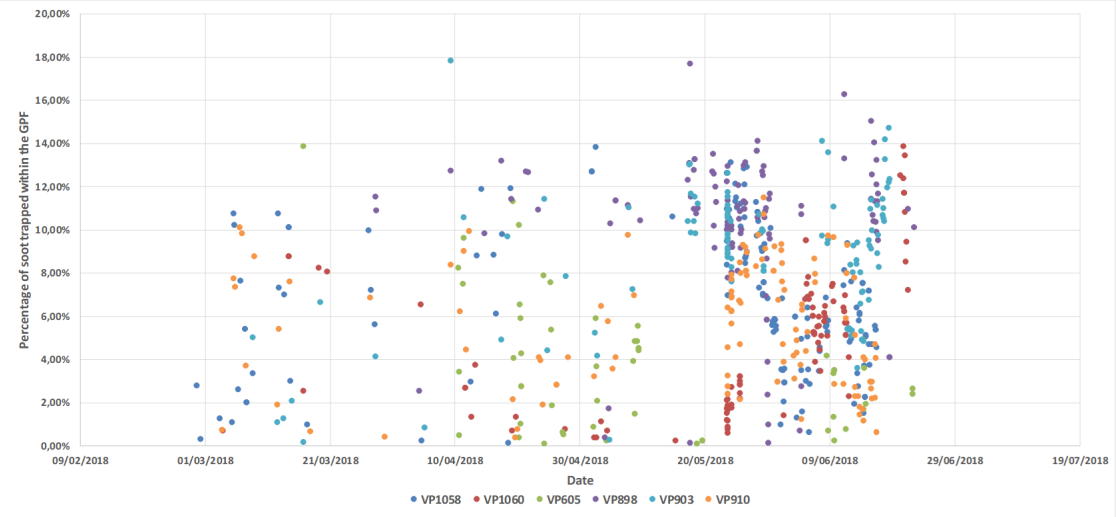


Figure 4.5: *Minimum amount of soot trapped within the filter, expressed in percentage*

Placing an anti particulate filter means an increase of the back pressure proportionally to the amount of soot trapped within the filter. As the back pressure increases, the engine must exert a higher mechanical work in order to expel the exhaust gases. This situation leads to a minor energy level extracted from the turbine, meaning a lower pressure of the mixture entering the combustion chamber; this leads to a power loss and to a consumption, PM and CO emissions and exhaust temperature increase.

For these reasons, the particulate filter must carefully be designed in order to avoid such malfunctions of the exhaust system. What I had to do was then to monitor the most important parameters of the filter in order to verify the correct working of the whole system.

The results of my analysis can be seen in Figure 4.6. As the variation of pressure across the filter increases, also the soot accumulated increases. I wanted to better underline this behaviour, so I created a trend line in the chart, so that the relationship between these two parameters is clearer. Of course, also the variation of pressure has been modified in order to place the chart in the thesis: it is shown in percentage, obtained by dividing the real value by a critical one. It can be seen that also in this case, the GPF guarantees a robust solution, being always far away from the critical situations.

Another interesting consideration can be done starting from Figure 4.7. In this chart it is shown the relationship between the exhaust gas mass flow rate and the variation of pressure across the filter. Again, also the values of the exhaust gas mass flow rate has been transformed in a percentage value, obtained dividing the

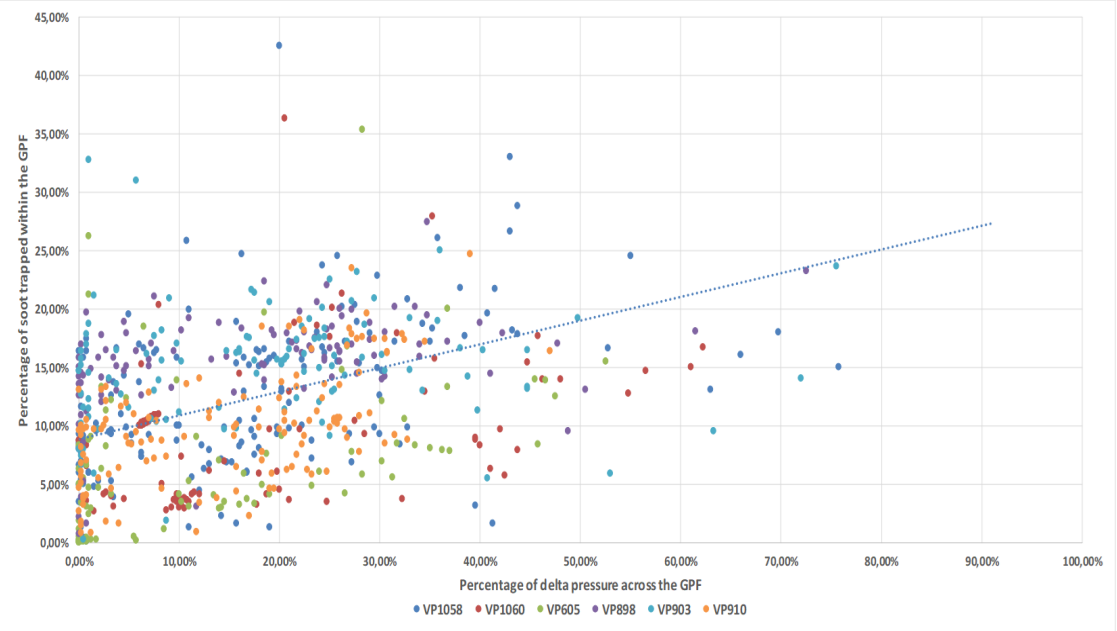


Figure 4.6: *Soot accumulated in the filter as a function of the variation of pressure across the filter*

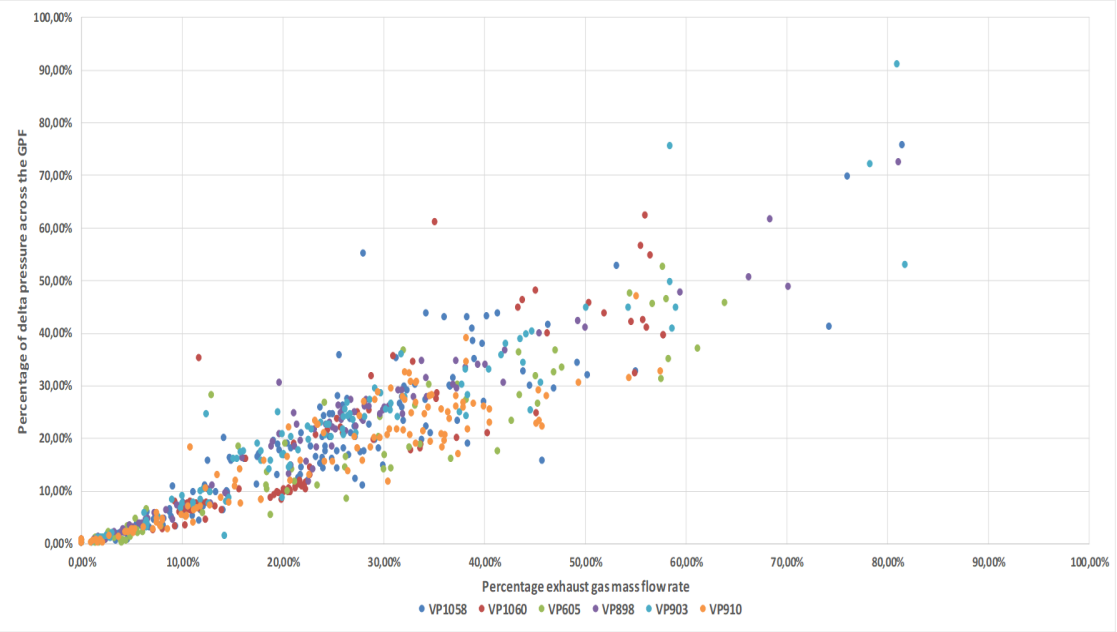


Figure 4.7: *Variation of pressure across the filter as a function of the exhaust gas mass flow rate*

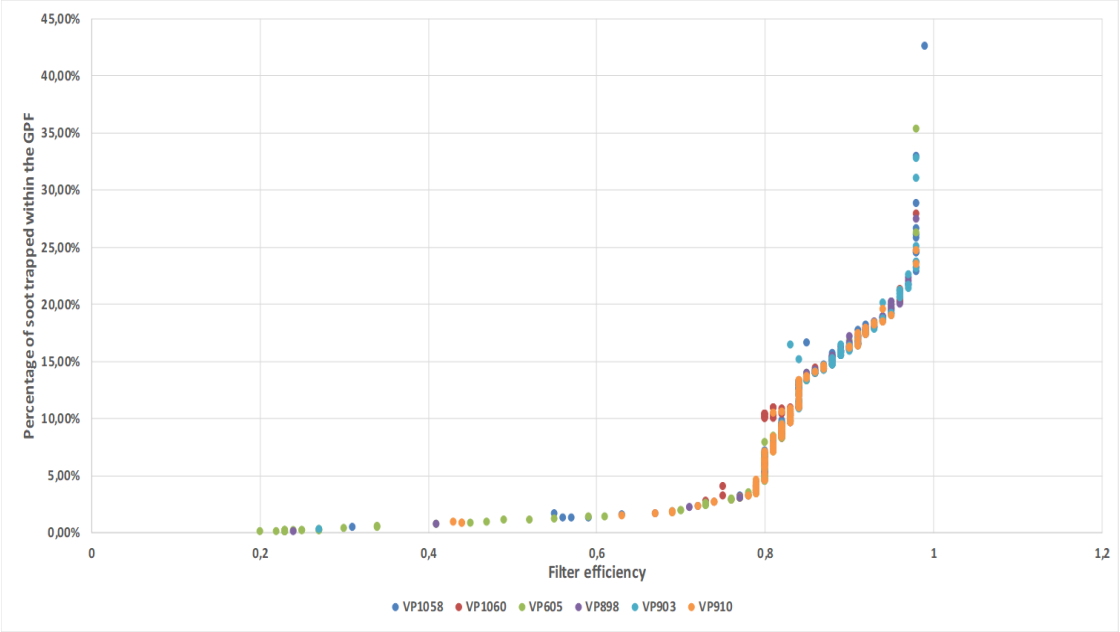


Figure 4.8: *Soot accumulated within the filter as a function of the filtering efficiency*

real value by the critical one. What this chart shows is quite intuitive: as the mass flow rate of the exhaust gases increases, also the variation of pressure across the filter increases. In fact, when the exhaust flow rate increases, more soot is trapped within the filter; more soot trapped in the filter means that the filter acts more as a plug, increasing in this way the variation of pressure across the filter. As before, I have underlined this relationship thanks to a trend line.

In addition to this, it can be seen that the filters never reach critical situations: a further confirm of the robustness of such type of solution on Alfa Romeo's vehicles.

4.2.2 Filtering efficiency

As previously explained in Section 4.1.3, the regeneration in GPF is passive, and has the advantage of keeping always under control the value of soot in the filter, but reducing so the filtering efficiency.

In fact, the filtering efficiency is highly dependant on the amount of soot trapped within the filter. In every scientific publication it is written that as the filter becomes empty, also the filtering efficiency decreases. So I wanted to verify this assumption, thanks to the potential of Moogle in analysing such a huge database.

The results of this chart is very significant: it is based on the everyday use of a vehicle, not on a laboratory test. It can be clearly seen that there is effectively a tight and important relationship between the two parameters, confirming then, also in the common use of a car, the experimental results shown in scientific papers.

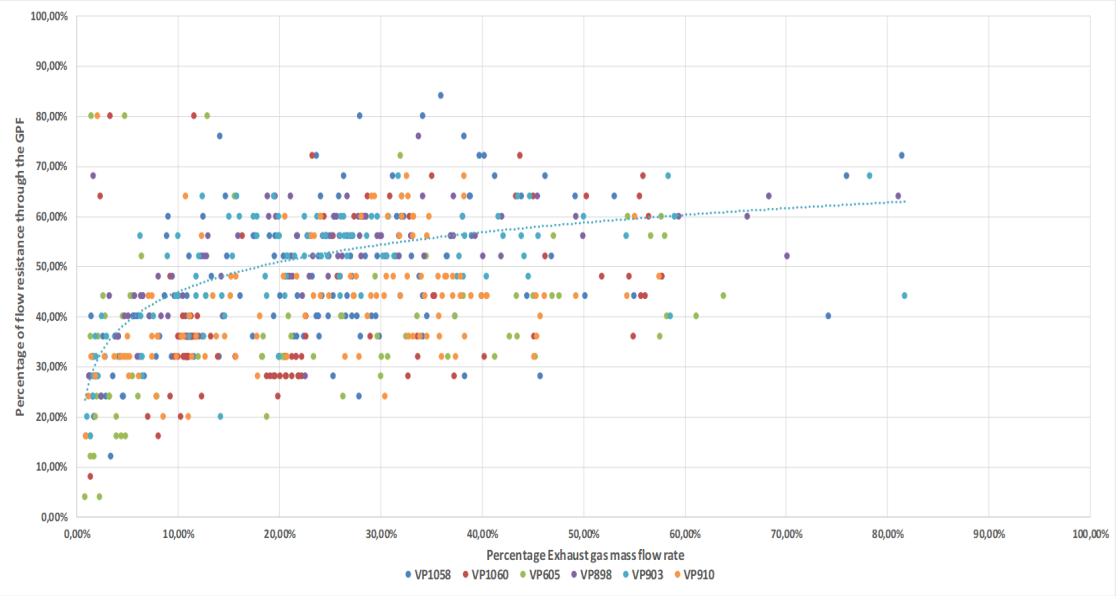


Figure 4.9: *Flow resistance through GPF as a function of the exhaust gas mass flow rate*

4.2.3 Flow resistance through the GPF

The last parameter I monitored is the flow resistance through the filter. Also in this case I had to transform the real value of this parameter, and just like all the other ones, the percentage shown in the chart has been obtained by dividing the real value by a critical one.

From Figure 4.9 it can be seen that, also in this case, there is a clear relationship between the two parameters. As the exhaust gas flow increases, also the flow resistance through the filter increases. Also in this case, this is an intuitive relationship. In fact, increasing the exhaust gas flow leads to an increase of the soot accumulated within the filter, and, as the filter is filled it acts as a plug, leading then to an increase of the resistance that the flow of gases meets within the filter.

From the percentage values, moreover, it can be seen that also for what concerns these parameters the GPF represents a robust solution. And, also in this case, I wanted to underline the direct relationship between these two parameters with a trend line.

4.3 Other parameters

After having analysed the main GPF parameters, I tried to find some possible correlations between the amount of soot trapped within the filter, and other parameters. I did this in order to increase the know how of the company about this

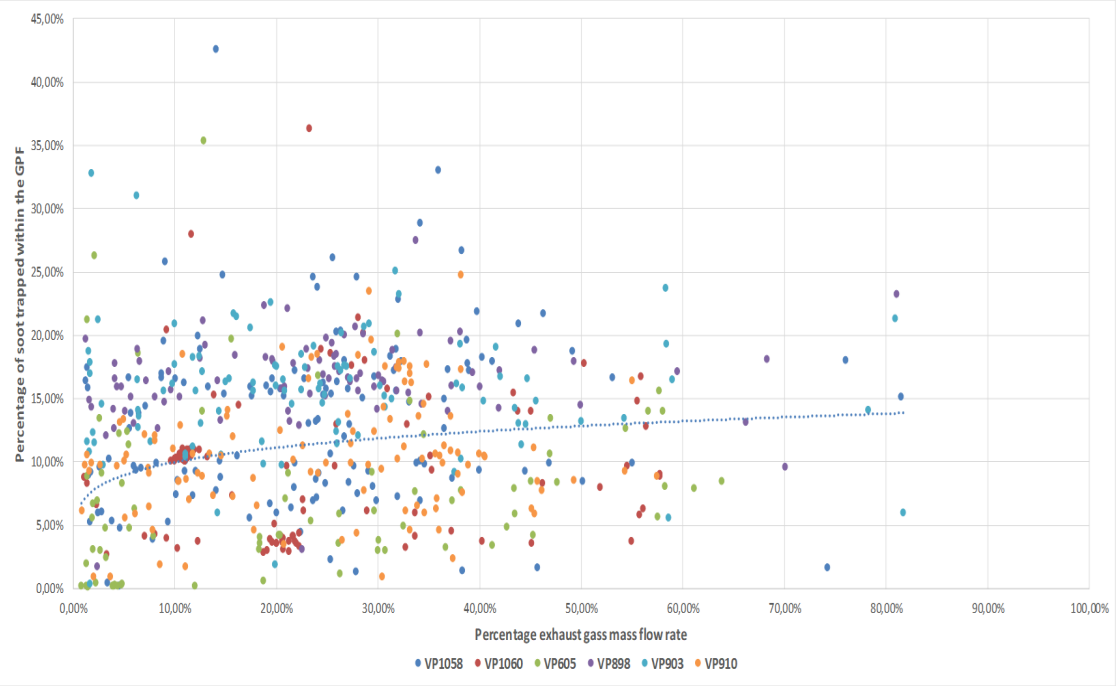


Figure 4.10: *Amount of soot trapped within the GPF as a function of the exhaust gas mass flow rate*

topic: Maserati is, in fact, in a pure experimentation for what regards the GPF. The results of this analysis can be seen in the following charts.

In Figure 4.10 it is shown the amount of soot trapped within the filter as a function of the exhaust gas mass flow rate. The trend line underlines a very slight relationship between these two parameters. This is due to the fact that, as the exhaust gas mass flow rate increases, the soot trapped within the filter increases. Anyway, this chart isn't able to underline a strong correlation.

In Figure 4.11 it is shown the amount of soot trapped within the filter as a function of the filter's mean temperature. Also in this case it is not possible to find a direct and strong correlation between these two parameters. Of course, after the warm up phase, the filter's temperatures will be always high, due to the temperatures of the hot exhaust gases, and this explains the higher density of points in that part of the chart.

In Figure 4.12 it can be seen the relationship between the amount of soot trapped within the filter and the boost pressure, expressed in percentage. This percentage has been obtained by dividing the real boost pressure, expressed in *mbar*, by a critical limit boost pressure; after this critical value the turbine could be seriously damaged, leading to a strong decrease in the engine's performances. As the chart highlights, this critical value is never reached, meaning that the turbine always does its job without any accidental damage. Also in this case I wasn't able

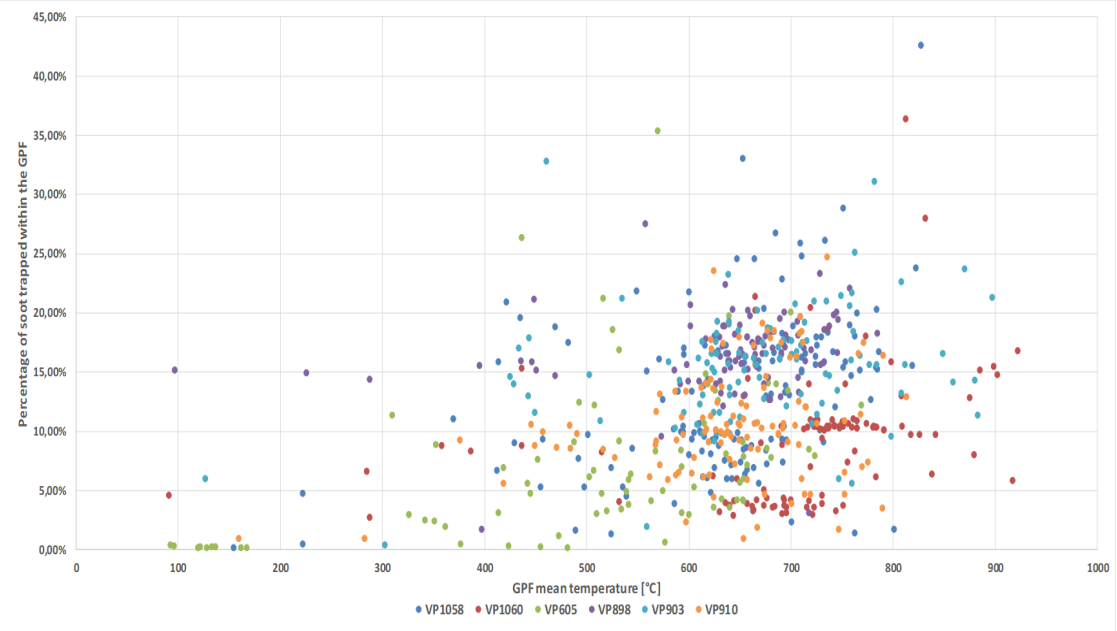


Figure 4.11: *Amount of soot trapped within the GPF as a function of the filter's mean temperature*

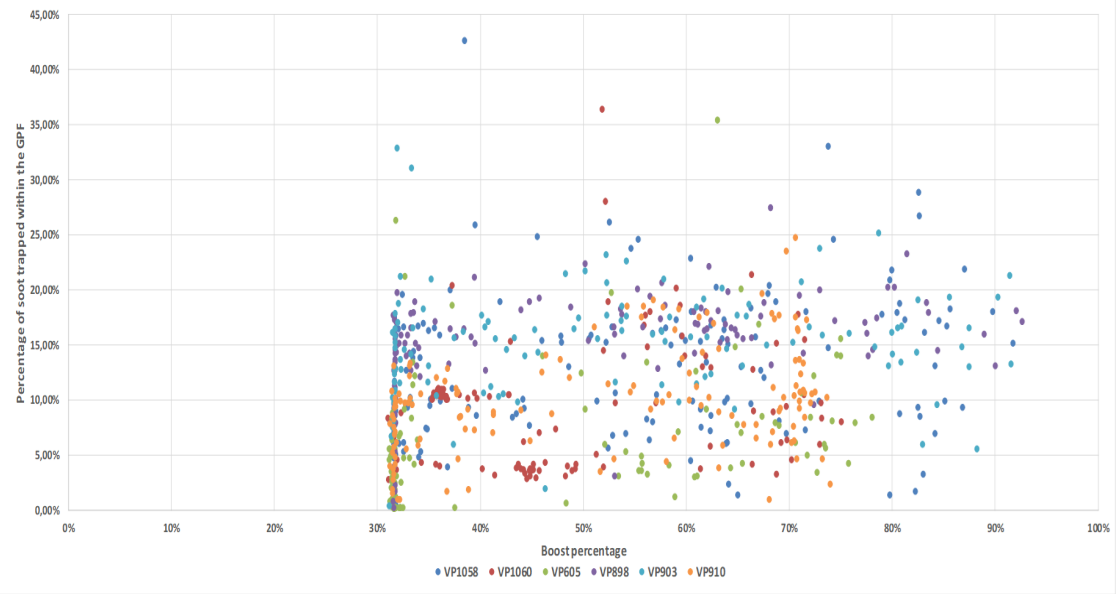


Figure 4.12: *Amount of soot trapped within the GPF as a function of the boost pressure percentage*

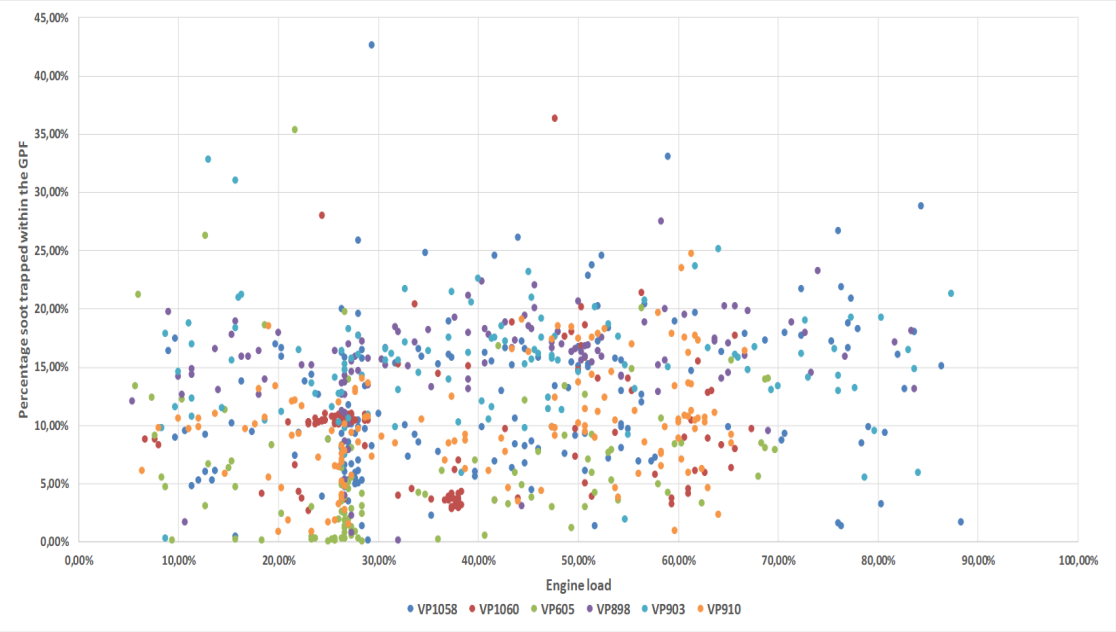


Figure 4.13: *Amount of soot trapped within the GPF as a function of the engine load percentage*

to find a direct correlation between the two parameters.

In Figure 4.13 it is shown the amount of soot trapped within the filter as a function of the engine load percentage. In this chart it is clearly shown that the amount of soot trapped in the filter doesn't depend at all by the engine load. It is an interesting conclusion, especially knowing that the pollutant emissions depend highly on the engine load and rotational speed.

In Figure 4.14 it is shown the amount of soot trapped within the filter and the engine's water temperature. It is valid a speech similar to the one done for the GPF temperature: after the initial warm up, the engine gets more or less stable around a certain engine temperature (between the 80°C and the 100°C). So the great majority of the points will lay mainly in that temperature's range, with some point, obviously, out of the range. So also these parameters do not show any direct and clear correlation.

The charts shown until here are related to engine or filter's parameters. The following ones will be related to environmental parameter: I wanted to understand if, for instance, the environmental temperature and pressure have a direct impact on the amoun of soot trapped within the filter.

In Figure 4.15 it is possible to see the amount of soot trapped within the filter as a function of the environmental pressure, expressed in *mbar*. Due to the fact that the vast majority of the kilometric accumulation was done near Turin and Modena, the minimum environmental temperature cannot be low. From the chart

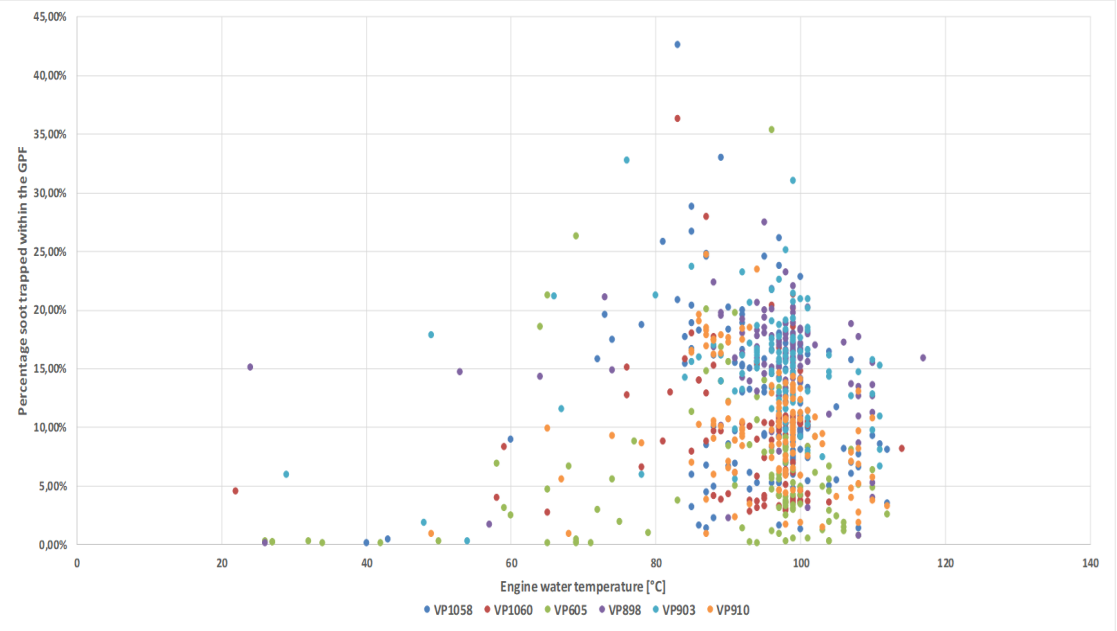


Figure 4.14: *Amount of soot trapped within the GPF as a function of the engine water temperature*

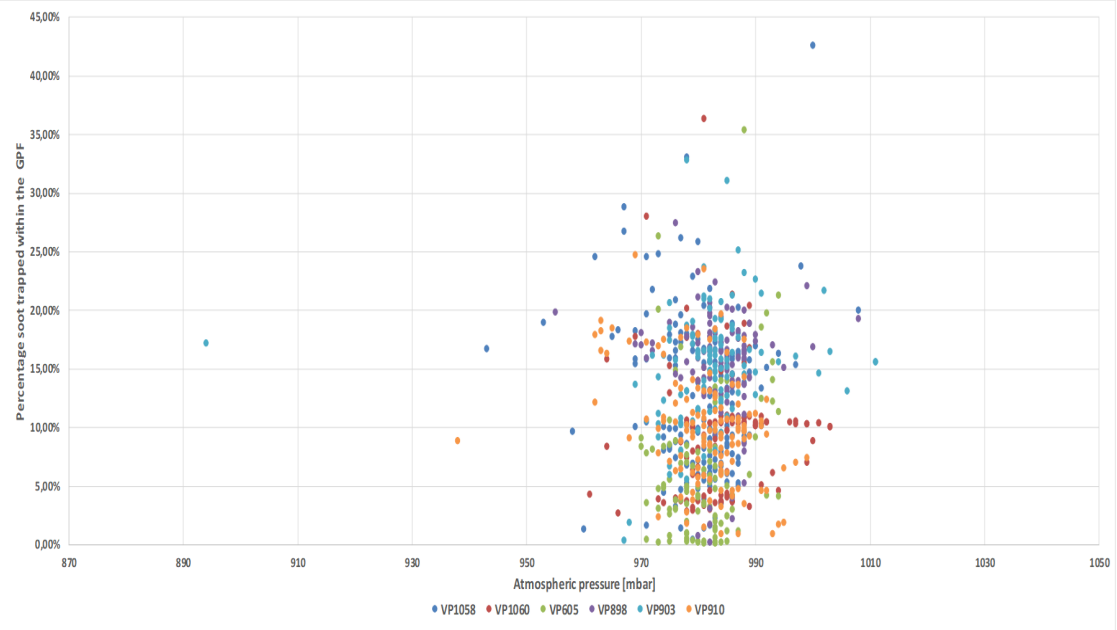


Figure 4.15: *Amount of soot trapped within the GPF as a function of the environmental pressure*

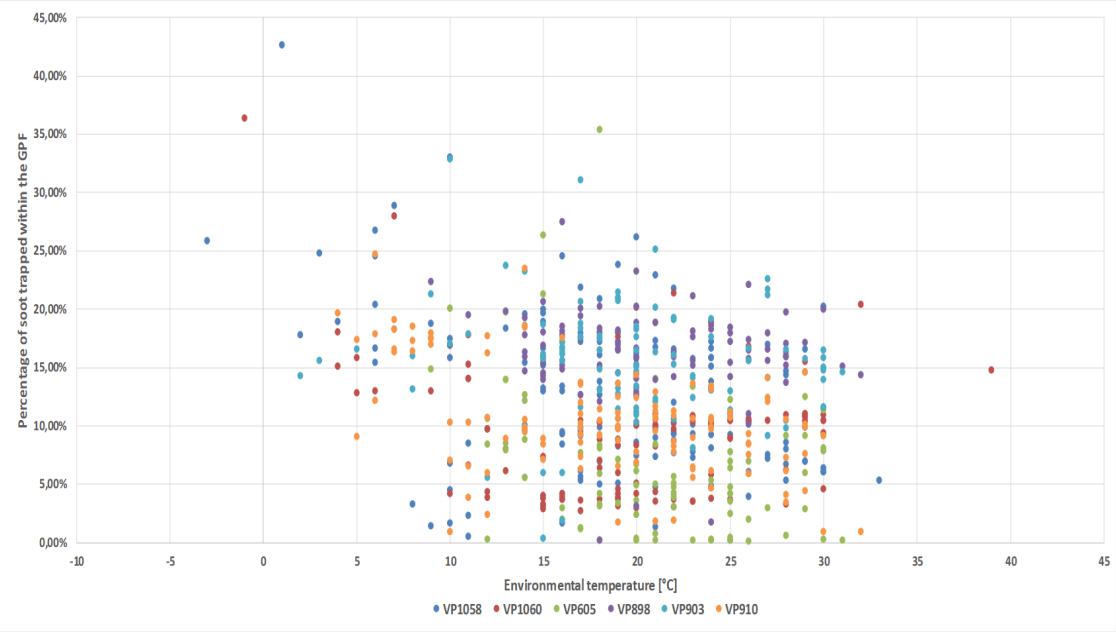


Figure 4.16: *Amount of soot trapped within the GPF as a function of the environmental temperature*

it is clear that the environmental pressure isn't a parameter that has an effect on the amount of soot trapped in the filter. It would be interesting, after a mountain trip, to see how the low pressure points are positioned within the chart. Until such type of test is done by Maserati, this chart is the current reference point for the relationship between these two parameters.

The last chart I created in Moogle is visible in Figure 4.16. This chart seems to be a bit more interesting than the one in Figure 4.15. In fact, it seems that, as the environmental temperature decreases, there are situations in which the soot accumulated within the filter increases. Due to the fact that these data takes in consideration only measurement files from the late February, it is not possible to have a high amount of low environmental temperatures points. So it isn't still possible to assume, on a statistical base, this relationship between environmental temperature and soot accumulated within the filter. As a cold trip will be done by Maserati, it will be possible to fill the low temperatures part of the chart, and see if my assumption is correct, or totally wrong.

4.4 Conclusions

This analysis was needed within the company in order to clearly understand if the GPF solution was a good one for the Model Year 19 vehicles, so that the new emissions regulations were fulfilled, or if there was the need of a different solution.

The relationships between the main GPF's parameters, that are expressed in all the scientific papers, have been confirmed on statistic basis, starting from data regarding the everyday use of a vehicle, and not laboratory test. I also tried to find a correlation between the amount of soot trapped within the filter and other engine's parameters, without particular success.

All the charts previously shown clearly prove that the Gasoline Particulate Filter is a good and reliable solutions for high performance vehicles, like Alfa Romeo Giulia and Stelvio 200/280 HP. The amount of soot trapped within the filter is always far below the critical values, avoiding then the plug effect that could arise when this type of filters are implemented. Moreover, also the values of filter's backpressure and flow resistance are far below critical values.

The GPF has so passed every possible test within the company, and so this component can be used without problems on each future Alfa Romeo vehicle.

Chapter 5

Model Validation: Environmental Temperature

5.1 Introduction

In Chapter 3 and Chapter 4 I have shown how Moogle can be used for a diagnosis and validation purpose. But the potential of the software goes far beyond these possibilities: it can also be used as a support tool to the calibration phase.

I had the possibility to work on the validation of a new model regarding the computation of the environmental temperature in steady state conditions. Currently on Maserati's vehicles is present a sensor that is responsible of measuring the value of the external environmental temperature. The sensor works very well, but, being part of the hardware of the vehicle, it involves a cost in the final price tag, both for the company and for the final customer. Being able to remove this sensor means that the final cost for the company decreases: this cost isn't that high for the single sensor, but when each car fits it, it becomes clear that its impact is relevant.

In automotive field it is quite common to directly derive or integrate some parameters from others in order to solve this problem. When the direct operation is not possible, the use of a model is needed: it must compute, with an acceptable error, the environmental temperature starting from other parameters. This model must be carefully calibrated, in order to ensure the requested precision: in fact, the environmental temperature is a parameter that is used in many diagnosis and other functions within the ECU, and so it must be as most precise as possible.

Actually, the experimentation of this model is running, and the efforts of the OBD calibrators are directed to the calibration of the steady state version of the model. Once this version will be considered reliable, it will be possible to ex-

tend the model also to the transient phase, making then possible to remove the environmental temperature sensor.

5.2 Physical Model

5.2.1 Steady state conditions definition

This model works only if the ECU recognises a steady state condition for the vehicle. Due to the fact that Internal Combustion Engines hardly work at perfectly steady state conditions, the model is flexible for the definition of such condition. In order to work, there must be the following conditions:

- Engine temperature: the engine temperatures must be high enough, meaning that all the cold engine situations must be avoided. It means that the steady state model can't work as soon as the engine is on, but it must wait it to become hot. Moreover, one the warm temperature is reached, the engine temperature must be in a specific range.
- Intake air temperature: it must be in a specific range to allow the environmental temperature model to work.
- Vehicle speed: there is a minimum vehicle speed to allow the environmental temperature model to work.
- Air mass flow rate: the model works only if this parameter is in a specific range, avoiding so too high or low flow rates.
- Engine load derivative: there is a parameter that describes the variation of the engine load over the time. This is a top priority parameter: if this variation is too high, the ECU bypass all the others conditions and exits from the model.

All the above conditions, with the exception of the engine load derivative, are debounced: they must be valid for a certain amount of calibrated time in order to let the model work.

5.2.2 Model definition

Once all the conditions are valid, the model can finally be used for the computation of the environmental temperature. Its block scheme can be seen in Figure 5.1. In the following part, I'm going to describe each part of the scheme:

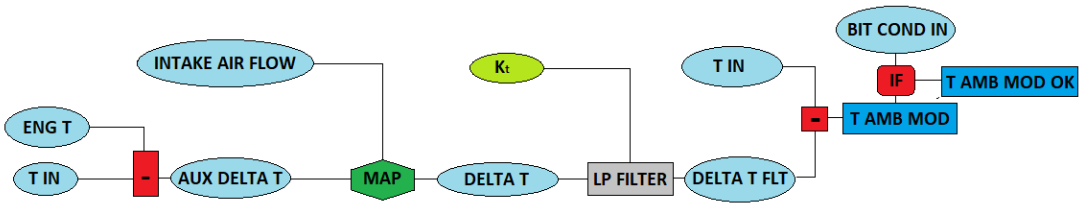


Figure 5.1: *Scheme of the Environmental temperature computation model*

- *Eng T*: Engine water temperature.
- *T In*: Intake air temperature.
- *Aux Delta T*: It is the difference between *Eng T* and *T In*. It is not in a usable form, because this parameter doesn't keep track of the intake air mass flow rate.
- *Intake Air Flow*: Once the steady state conditions are fulfilled, this parameter, representing the intake air mass flow rate, is then calculated.
- *Map*: This map is used to correct the *Aux Delta T* parameter by considering also the role of the intake air mass flow rate.
- *Delta T*: It is the true difference between the intake air temperature and the engine temperature.
- *LP Filter*: Low pass filter.
- K_t: Time constant of the low pass filter. Set in the calibration phase.
- *Low pass filter*: The output signal from the map is filtered in order to smoothen the output of the map.
- *Delta T Flt*: It is the filtered value of the difference between the intake air temperature and the engine temperature. This value represents an estimation of how much the ambient air is heated by the engine.
- *T Amb Mod*: Final output value of the model, that should simulate the real environmental temperature. It is obtained by subtracting, from the intake air temperature, the variation of temperature computed in the model, *Delta T Flt*.
- *Bit Cond In*: It is a bit that assumes value zero if the steady state conditions are not respected, one otherwise. If it is zero, the system discharge the computed value; if it is zero, the system keeps the computed values, assigning it the name *T Amb Mod Ok*.

In this phase of the test of the model, it is considered valid if the difference between the environmental temperature computed by the model and the environmental temperature measured by the sensor is below $40K$. Of course it is an enormous difference, that cannot be accepted in the final model, but is used only to validate this initial test of the model.

5.3 Trigger definition and working range

In order to analyse the measurement files, I had to write a trigger that was able to take all those points in which the difference between the temperature computed by the model and the measured ones were maximum or minimum, so we had the worst cases for the model. The trigger logic was indeed very easy, and it can be seen here:

```
DELTA_TEMPERATURE := ENVT_MODEL - ENVT_SENS;
ENVT_MODEL := detect/time {COMPUTING_MODEL = 1 AND
   SENSOR_STATUS = 1};
ENVT_MODEL += max DELTA_TEMPERATURE, ENVT_SENS, ENVT_MODEL;
ENVT_MODEL += min DELTA_TEMPERATURE, ENVT_SENS, ENVT_MODEL;
ENVT_MODEL += avg ENVT_SENS;
ENVT_MODEL += {
      ENVT_MODEL_MAX := detect/time {DELTA_TEMPERATURE = max (
         DELTA_TEMPERATURE)};
      ENVT_MODEL_MAX := start_value DELTA_TEMPERATURE,
         ENVT_SENS, ENVT_MODEL, INT_MASS_FLOW_RATE;
                  };

ENVT_MODEL += {
      ENVT_MODEL_MIN := detect/time {DELTA_TEMPERATURE = min (
         DELTA_TEMPERATURE)};
      ENVT_MODEL_MIN := start_value DELTA_TEMPERATURE,
         ENVT_SENS, ENVT_MODEL, INT_MASS_FLOW_RATE;
                  };
```

The channel `COMPUTING_MODEL` is used to extract only those situations in which the model is effectively computing the data, while the channel `SENSOR_STATUS` is used to take the data coming from the temperature sensor only when it is giving reliable informations. By imposing these conditions I'm sure that Moogle extracts data only when the model and the sensor are both correctly working.

The trigger extracts the maximum and minimum values of the temperatures

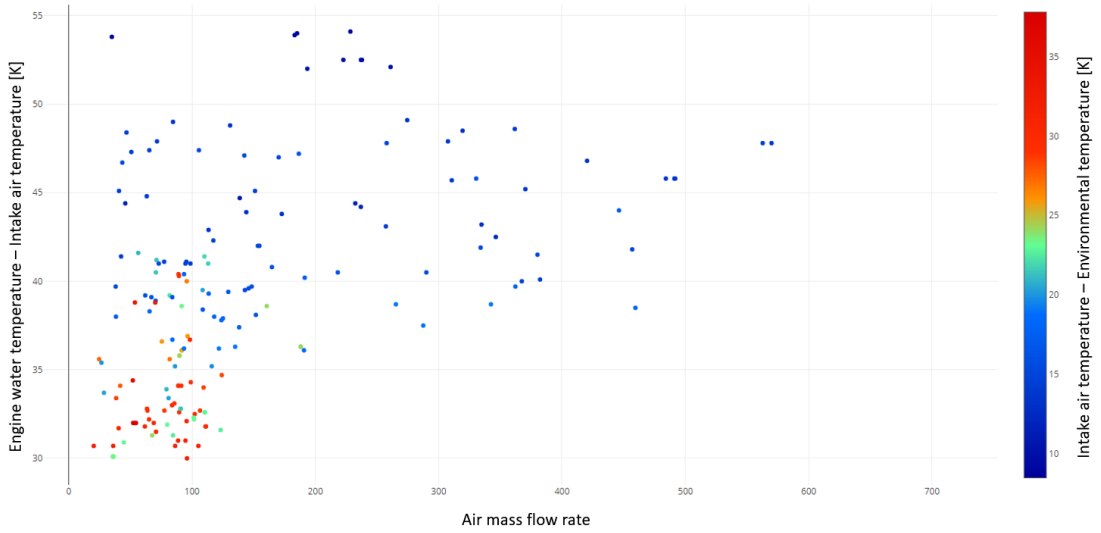


Figure 5.2: *Trigger working zone: it is a sort of rebuilding of the initial model map*

measured by the sensor, the one computed by the model and also the difference between these two temperatures. Moreover, in the nested triggers, I allow Moogle to take the values of the other channels corresponding at the instant in which there is the maximum, or minimum, difference between the computed environmental temperature and the measured one.

Moogle is so able to extract, beyond the values of the difference between the two temperatures (`DELTA_TEMPERATURE`), also the values of the temperature measured by the sensor (`ENVT_SENS`), the one computed by the model (`ENVT_MODEL`) and the intake air mass flow rate (`INT_MASS_FLOW_RATE`). Taking the maximum and minimum values of `DELTA_TEMPERATURE` and placing them on a chart let us understand if the model has worked well, or if it needs further refinement.

The trigger extracts data that come from a map, as shown inFigure 5.1. Due to the fact that I can't directly show this map in the thesis, I was able to recreate it thanks to Moogle. In fact, thanks to the nested trigger, I was able to extract the data and use them to create a simplified and partial version of the original map, as can be seen in Figure 5.2. The parameters and the axis are here explained:

- X-axis: Intake air mass flow rate. Is is voluntarily dimensionless, because it was the only way I had to show this chart without violating any company's rule. It gives and idea of the regions in which the model mainly works.
- Y-axis: Difference between engine water temperature and intake air temperature. The model mainly works when this difference is low(hot environmental temperature), because the biggest part of the tests were performed in June.

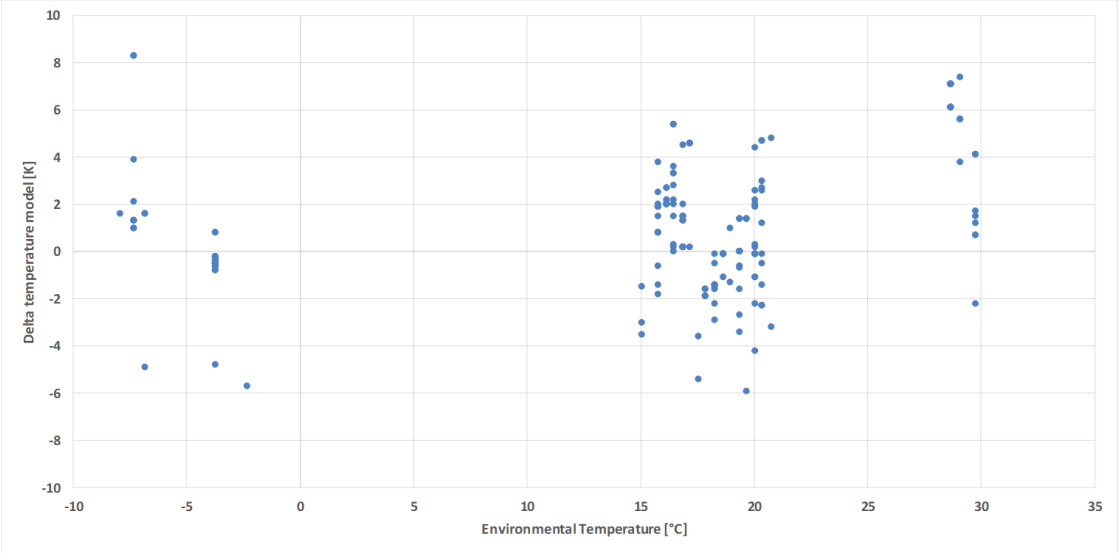


Figure 5.3: *Difference between environmental temperature computed by the model and the one measured by the sensor, as a function of environmental temperature. Data belonging to a Maserati Levante V8*

Higher points are due to tests performed at low temperatures thank to an engine test bench.

- Z-axis (Colour): Difference between intake air temperature and environmental temperature.

5.4 Experimental results

The first chart I created can be seen in Figure 5.3, where the ongoing of the difference between environmental temperature computed by the model and the one measured by the sensor, as a function of environmental temperature, is shown. This data have been obtained by the test outdone on a Maserati Levante.

From this first chart we can see that, actually, the model works quite well: the difference often lies between $-5K$ and $5K$, with some values with a higher delta, especially when the external environmental temperature becomes higher. It is a good first step, even if only a partial range of ambient temperatures have been tested, due to the current season. The model is good in the range $15K$ and $20K$, while it becomes less reliable at higher temperatures. Remembering that the first accepted range was $40K$, it is a very good result, that leads to the next steps of the test, in order to guarantee a robust and reliable model on a wider range of temperatures.

Due to the fact that actually the environmental temperatures are quite high,

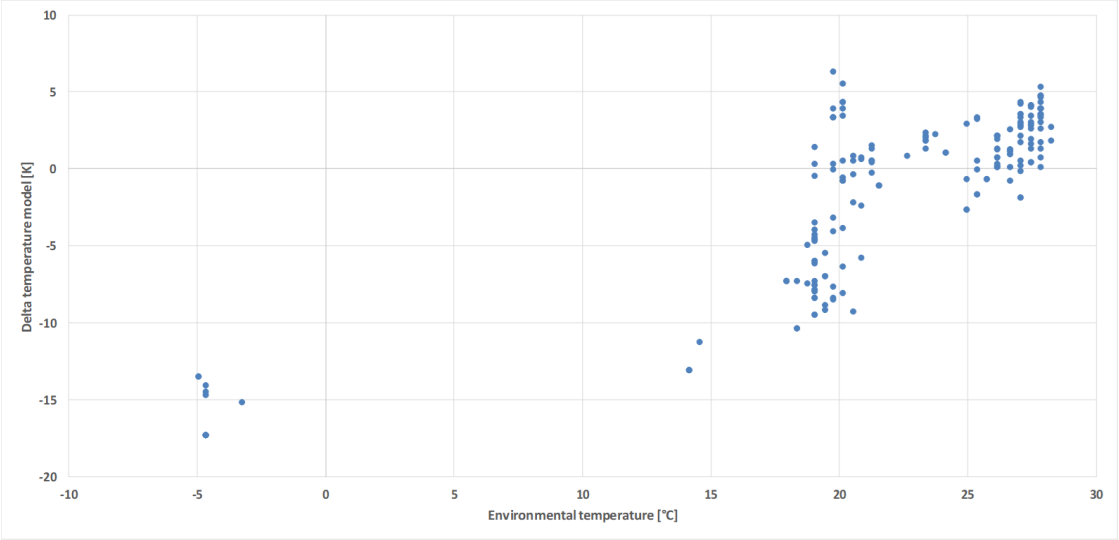


Figure 5.4: *Difference between environmental temperature computed by the model and the one measured by the sensor, as a function of environmental temperature. Data belonging to a Alfa Romeo Giulia V6 Quadrifoglio*

in order to have a result also at low, typically winter, temperatures, a series of tests were performed in a engine test bench, where it was also possible to control the environmental parameters, such as, in fact, environmental temperature and pressure: it was so possible to modify the temperatures of the environment.

As can be seen always in Figure 5.3, when the model has to work at lower temperatures it becomes a little less precise, but still acceptable, considering that this is a first stage of the validation of the model.

This type of model was also tested on other vehicles: in Figure 5.4 the results obtained on a Alfa Romeo Giulia V6 Quadrifoglio can be seen. Also in this case, the model works quite well at high temperatures, but it becomes very inaccurate at low temperatures. Of course, the target of having the difference between the temperature measured by the sensor and the one computed by the model below $40K$ has been reached, but it's clear that the model must be polished. The first thing that must be done, according to the calibrators, is to work on the map of the model in Figure 5.1: it must be modified the part of the map that works at low temperatures.

The calibrator who is working on this model hadn't any experimental data that he could use in order to set the map the first time; so he set the map according to a inductive method: he set it in a way that he thought the map could have been worked well. These experimental data show the goodness of his job at high temperature, while his imprecision at low ones. Now he can modify the map, and perform a second test phase: if the precision of the model will be improved also

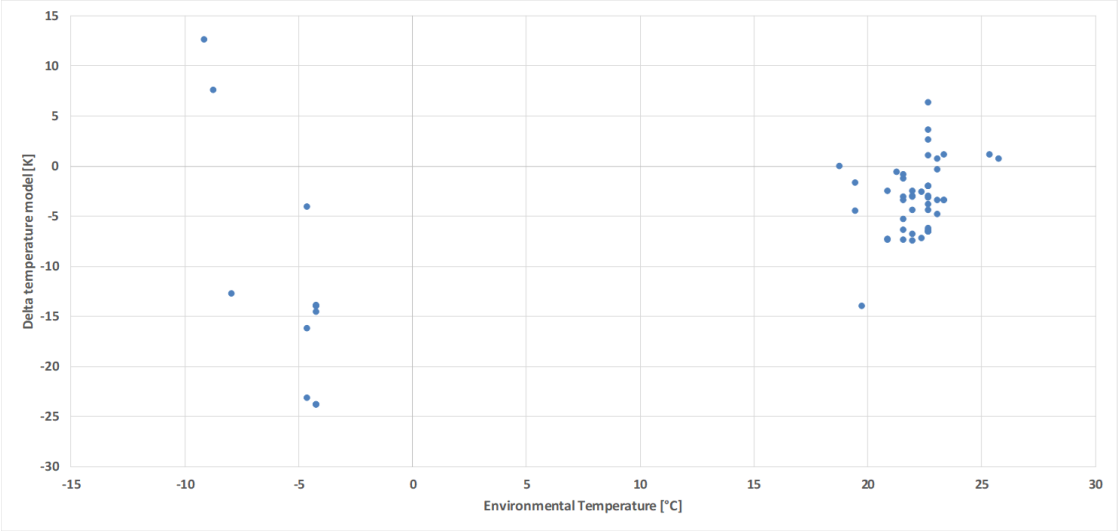


Figure 5.5: *Difference between environmental temperature computed by the model and the one measured by the sensor, as a function of environmental temperature. Data belonging to a Alfa Romeo Stelvio V6 Quadrifoglio*

at low temperatures, then the validation process will go on, otherwise another adjustment to the map will be needed.

The test was performed also on a Alfa Romeo Stelvio V6 Quadrifoglio, as can be seen in Figure 5.5: in this case the same considerations done for the Alfa Romeo Giulia are valid, even better if possible. In fact, it can be seen that, also here, the model works very well at high temperatures, but it becomes enormously imprecise at low temperatures: the results obtained are worse than the ones belonging to the Giulia. It is the definitive evidence that the map must be modified in order to be as precise as possible also at low temperatures.

5.5 Conclusions

This type of analysis couldn't have been possible, in the same short times, without the use of Moogle. In fact, the model could have been set anyway, but only thank to the processing abilities of the software we had available the results in a few minutes, while a manual verification on the measurement file would have needed several days, probably also weeks.

In a couple of days we were so able to test the model in dedicated driving and bench tests, and then to study then thanks to the power of Moogle. This first stage analysis underlined the fact that the model works very well at high environmental temperatures, while it must be recalibrated at low ones. Anyway, due to the fact that the model is considered valid if the difference between the environmental

temperature computed by the model and the environmental temperature measured by the sensor is below $40K$, it is clear that the model passed this first stage of test.

The next step is, of course, guaranteeing that the model works in every range of temperatures; after that, the threshold of $40K$ must be decreased. Every time that the model will be modified, it must be tested and approved by using Moogle.

The final step will consist in extending this model also in the non-steady state conditions: it is clear that passenger vehicles are hardly used in steady state conditions, especially if we think to the urban environment. It will require time, but the results will be analysed in a fast, efficient and effective way by Moogle. If this model will eventually pass all the tests and then be approved by the company, it will lead to an enormous save of money for the company. It will be the definitive proof, also for the high level management, that Moogle is a powerful and strong tool that can be very useful for the company.

Chapter 6

Moogle development

As written in Chapter 2, I have a double role within the company. The first one is related to the creation of statistics thank to the use of Moogle; the second one is the management of the contacts between the company and ETAS in order to continuously improve the software over the time.

Even if this second activity seems of course secondary with respect to the creation of statistics for the colleagues, in reality it has the same importance from the company's perspective. In fact, only a every-day user of the software can effectively suggest to ETAS new and efficient way to improve the tool and to add new features. And by improving the software, it is possible also to improve the quality and the precision of the statistics: this shows clearly how these two parts of my job are in reality very tightly linked.

The standard routine has already been discussed in Chapter 2. Thanks to the mails, the WebEx and the face to face meetings and also the phone calls, I could always be in touch with ETAS technicians. In this chapter I'm going to show the typical way that allowed me to think to a new possible feature and all the route that led to the final software addition or improvement. Moreover I want also to show the impact of these software's updates from the company's perspective.

As the main user of Moogle, I was able to be in contact with a wide variety of requests from my colleagues, and every request needed a unique approach to the problem in order to deliver a readable and useful statistic. This part of the job gave me the instruments to understand what could be the useful addition to Moogle, and what instead were absolutely not needed. The main tool that the two company uses in order to keep track of Maserati's request is an Excel file, the Open point list (OPL).

The standard communication process between the two company starts with an email sent by me to ETAS, containing the feature or improvement that I liked to

have implemented. Then I add this request to the OPL file, so that it is available for the discussion in the following WebEx meeting, keeping in mind that the answer to my mail is the basis for the upcoming discussion. During the meeting, the OPL file is shown to the ETAS technicians, and so the discussion starts from that file. For this reason it was very important to have a clear and full detailed OPL file.

6.1 OPL development and management

In order to have an effective and efficient relationship with ETAS, I modified heavily the way that the two company used in order to discuss the improvements. Before my arrival, the basis of this collaboration was a gaunt Excel file, containing the main points that Maserati asked ETAS to develop, without further information. In that way it was hard to keep trace of the date on which a request was placed, the ETAS feedbacks, the mail exchanges and so on.

This file was totally unsatisfactory: it was very hard to understand when a request was effectively made, and when instead there were an update from ETAS side. Moreover, it was not present a priority scale: every request was initially placed on the same level, dispersing then the developers efforts on the low priority requests, instead than focusing mainly on the critical ones.

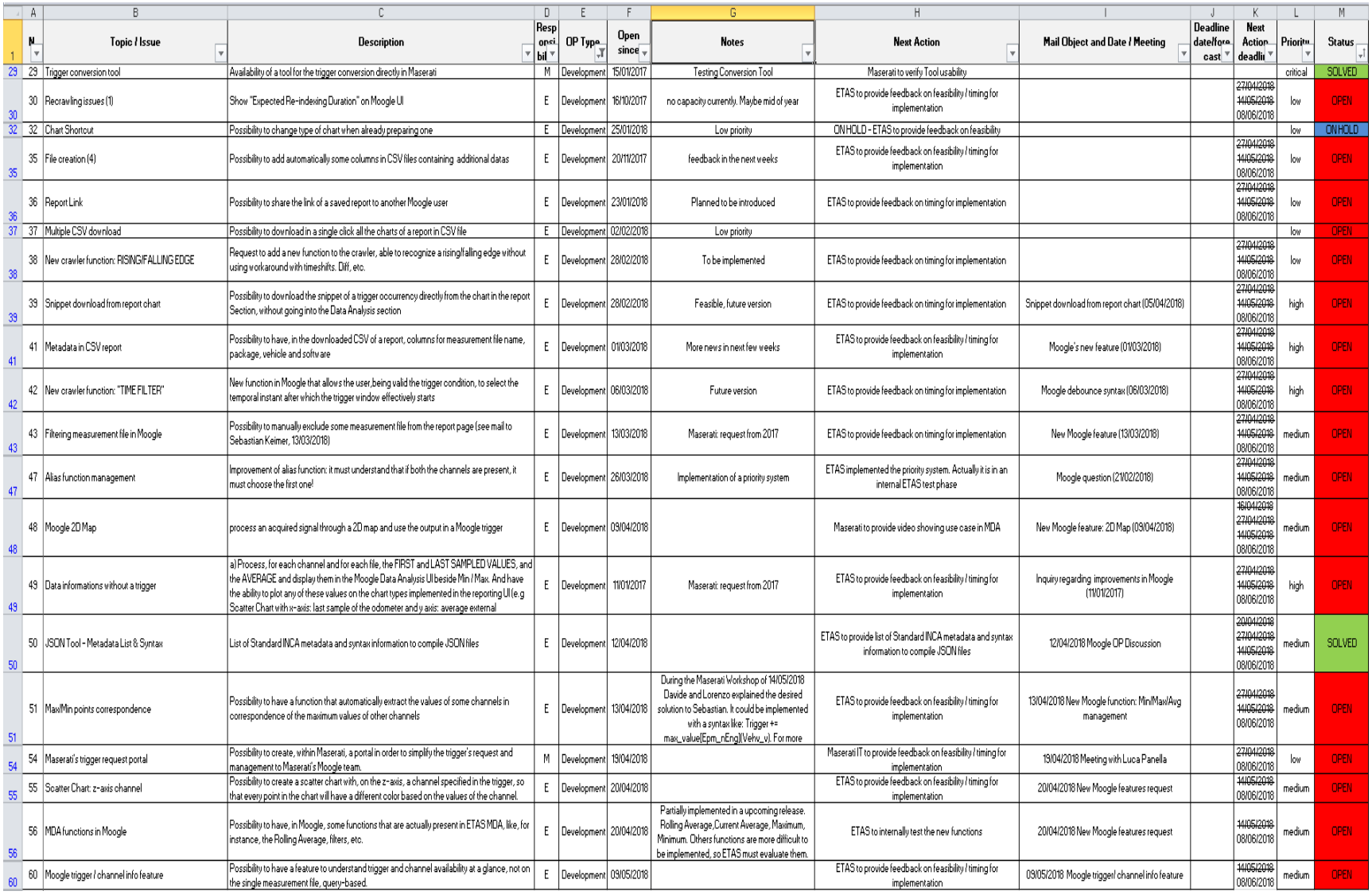
For these reasons the OPL was modified, and the final version can be seen in Figure 6.1. This result was achieved by adding or improving the following fields to the original OPL Excel file:

- Exhaustive explanation of the request: initially, every request was placed in a row of the Excel file, and characterised by a synthetic description. In order to make it easier to understand to ETAS, I added a column in which an exhaustive and detailed explanation of the request could be inserted.

- Priority system: I added a column containing the priority of each request. In this way everything was classified according to the following sorting scale:

  1. Critical.
  2. High.
  3. Medium
  4. Low.

  So ETAS could be able to focus before everything else on the critical requests, and then to give importance to the high, medium and low levels. This classification systems improved heavily the disposal of the request's backlog.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	N	Topic / Issue	Description	Responsibil	OP Type	Open since	Notes	Next Action	Mail Object and Date / Meeting	Deadline datefore cast	Next Action deadlii	Priority	Status
29	29	Trigger conversion tool	Availability of a tool for the trigger conversion directly in Maserati	M	Development	15/01/2017	Testing Conversion Tool	Maserati to verify Tool usability				critical	SOLVED
30	30	Recrawling issues (1)	Show "Expected Re-indexing Duration" on Moogle UI	E	Development	16/10/2017	no capacity currently. Maybe mid of year	ETAS to provide feedback on feasibility / timing for implementation			~~27/04/2018~~ ~~14/05/2018~~ 08/06/2018	low	OPEN
32	32	Chart Shortcut	Possibility to change type of chart when already preparing one	E	Development	25/01/2018	Low priority	ON HOLD - ETAS to provide feedback on feasibility				low	ON HOLD
35	35	File creation (4)	Possibility to add automatically some columns in CSV files containing additional datas	E	Development	20/11/2017	feedback in the next weeks	ETAS to provide feedback on feasibility / timing for implementation			~~27/04/2018~~ ~~14/05/2018~~ 08/06/2018	low	OPEN
36	36	Report Link	Possibility to share the link of a saved report to another Moogle user	E	Development	23/01/2018	Planned to be introduced	ETAS to provide feedback on timing for implementation			~~27/04/2018~~ ~~14/05/2018~~ 08/06/2018	low	OPEN
37	37	Multiple CSV download	Possibility to download in a single click all the charts of a report in CSV file	E	Development	02/02/2018	Low priority					low	OPEN
38	38	New crawler function: RISING/FALLING EDGE	Request to add a new function to the crawler, able to recognize a rising/falling edge without using workaround with timeshifts. Diff, etc.	E	Development	28/02/2018	To be implemented	ETAS to provide feedback on timing for implementation			~~27/04/2018~~ ~~14/05/2018~~ 08/06/2018	low	OPEN
39	39	Snippet download from report chart	Possibility to download the snippet of a trigger occurrency directly from the chart in the report Section, without going into the Data Analysis section	E	Development	28/02/2018	Feasible, future version	ETAS to provide feedback on timing for implementation	Snippet download from report chart (05/04/2018)		~~27/04/2018~~ ~~14/05/2018~~ 08/06/2018	high	OPEN
41	41	Metadata in CSV report	Possibility to have, in the downloaded CSV of a report, columns for measurement file name, package, vehicle and software	E	Development	01/03/2018	More news in next few weeks	ETAS to provide feedback on feasibility / timing for implementation	Moogle's new feature (01/03/2018)		~~27/04/2018~~ ~~14/05/2018~~ 08/06/2018	high	OPEN
42	42	New crawler function: "TIME FILTER"	New function in Moogle that allows the user, being valid the trigger condition, to select the temporal instant after which the trigger window effectively starts	E	Development	06/03/2018	Future version	ETAS to provide feedback on timing for implementation	Moogle debounce syntax (06/03/2018)		~~27/04/2018~~ ~~14/05/2018~~ 08/06/2018	high	OPEN
43	43	Filtering measurement file in Moogle	Possibility to manually exclude some measurement file from the report page (see mail to Sebastian Keimer, 13/03/2018)	E	Development	13/03/2018	Maserati request from 2017	ETAS to provide feedback on timing for implementation	New Moogle feature (13/03/2018)		~~27/04/2018~~ ~~14/05/2018~~ 08/06/2018	medium	OPEN
47	47	Alias function management	Improvement of alias function: it must understand that if both the channels are present, it must choose the first one!	E	Development	26/03/2018	Implementation of a priority system	ETAS implemented the priority system. Actually it is in an internal ETAS test phase	Moogle question (21/02/2018)		~~27/04/2018~~ ~~14/05/2018~~ 08/06/2018	medium	OPEN
48	48	Moogle 2D Map	process an acquired signal through a 2D map and use the output in a Moogle trigger	E	Development	09/04/2018		Maserati to provide video showing use case in MDA	New Moogle feature: 2D Map (09/04/2018)		~~16/04/2018~~ ~~27/04/2018~~ ~~14/05/2018~~ 08/06/2018	medium	OPEN
49	49	Data informations without a trigger	a) Process, for each channel and for each file, the FIRST and LAST SAMPLED VALUES, and the AVERAGE and display them in the Moogle Data Analysis UI beside Min / Max. And have the ability to plot any of these values on the chart types implemented in the reporting UI (e.g Scatter Chart with x-axis: last sample of the odometer and y axis: average external	E	Development	11/01/2017	Maserati request from 2017	ETAS to provide feedback on feasibility / timing for implementation	Inquiry regarding improvements in Moogle (11/01/2017)		~~27/04/2018~~ ~~14/05/2018~~ 08/06/2018	high	OPEN
50	50	JSON Tool - Metadata List & Syntax	List of Standard INCA metadata and syntax information to compile JSON files	E	Development	12/04/2018		ETAS to provide list of Standard INCA metadata and syntax information to compile JSON files	12/04/2018 Moogle OP Discussion		~~20/04/2018~~ ~~27/04/2018~~ ~~14/05/2018~~ 08/06/2018	medium	SOLVED
51	51	Max/Min points correspondence	Possibility to have a function that automatically extract the values of some channels in correspondence of the maximum values of other channels	E	Development	13/04/2018	During the Maserati Workshop of 14/05/2018 Davide and Lorenzo explained the desired solution to Sebastian. It could be implemented with a syntax like: Trigger += max_value[Epm_nEng]\Vehv_v). For more	ETAS to provide feedback on feasibility / timing for implementation	13/04/2018 New Moogle function: Min/Max/Avg management		~~27/04/2018~~ ~~14/05/2018~~ 08/06/2018	medium	OPEN
54	54	Maserati's trigger request portal	Possibility to create, within Maserati, a portal in order to simplify the trigger's request and management to Maserati's Moogle team.	M	Development	19/04/2018		Maserati IT to provide feedback on feasibility / timing for implementation	19/04/2018 Meeting with Luca Panella		~~27/04/2018~~ 08/06/2018	low	OPEN
55	55	Scatter Chart: z-axis channel	Possibility to create a scatter chart with, on the z-axis, a channel specified in the trigger, so that every point in the chart will have a different color based on the values of the channel.	E	Development	20/04/2018		ETAS to provide feedback on feasibility / timing for implementation	20/04/2018 New Moogle features request		~~14/05/2018~~ 08/06/2018	medium	OPEN
56	56	MDA functions in Moogle	Possibility to have, in Moogle, some functions that are actually present in ETAS MDA, like, for instance, the Rolling Average, filters, etc.	E	Development	20/04/2018	Partially implemented in a upcoming release. Rolling Average,Current Average, Maximum, Minimum. Others functions are more difficult to be implemented, so ETAS must evaluate them.	ETAS to internally test the new functions	20/04/2018 New Moogle features request		~~14/05/2018~~ 08/06/2018	medium	OPEN
60	60	Moogle trigger / channel info feature	Possibility to have a feature to understand trigger and channel availability at a glance, not on the single measurement file, query-based.	E	Development	09/05/2018		ETAS to provide feedback on feasibility / timing for implementation	09/05/2018 Moogle trigger/ channel info feature		~~14/05/2018~~ 08/06/2018	medium	OPEN

Figure 6.1: *The new and more detailed OPL version*

- Mail object and date: due to the fact that usually every request started from a mail, it was very important to be able to keep track of these mails. Due to the high amount of daily mails, it was very easy to not be able to find a specific mail. In this field it was then contained the name of the mail's object and also the sending date: with these two informations, it was very easy to rapidly find the mails, and consequently the request's text.

- Responsibility: this field shows very quickly who, between Maserati and ETAS, is in charge of carrying out that specific request. Usually, after the first mail, the responsible was always ETAS, because it had to communicate the feasibility of the feature, the timing and other informations. But, for instance, in a second instant of the feature's development, the responsibility could come back to Maserati, for the test phase of the developed feature.

- Notes: this column was introduced in order to give, both to Maserati and ETAS, a field in which to write some notes, that could be related to the state of the development, to a particular feedback, and so on.

- Next action deadline: this column was added with the purpose to clearly show to ETAS the next deadline. Usually the deadlines were the WebEx or face to face meeting: ETAS had to be prepared in order to tell us the work progress of that specific feature. In fact, before this addition to the OPL, it was very easy to have ETAS to not discuss about some points until those features were ready. Thanks to this improvement, we were able to better follow each step of the development.

- Deadline date/forecast: this field is used to place the date in which the update or feature should be released to Maserati. It is usually an empty space until ETAS is sure to give the date. It is a reference point: if the update is not released within that date, ETAS must justify the retard and provide a new roadmap.

The results of this improved OPL file can be seen in Figure 6.1: this configuration includes a wide variety of informations, making easy for everyone to keep track of each request, and its state of development.

6.2 Software improvements

In the following part it will be shown the bigger improvements applied to the software thanks to my suggestions and help. It won't be shown any bug-fix, as it

isn't an interesting topic. Moreover, not all the improvements are strictly related to the software: for instance, I was able to improve the trigger conversion phase, without having ETAS to modify Moogle.

6.2.1 Trigger conversion tool

As written in Section 2.3.3, a part of the process for the use of Moogle is the conversion of the trigger from the *mgl* to the *dll* format. In reality it isn't a necessary step, as Moogle is able to process also the *mgl* files, but becomes mandatory when the data dimension to be analysed is so high, otherwise the computational time becomes unacceptable. This is due to the fact that Moogle is slower in processing *mgl* files than *dll* ones.

For this reason, ETAS created a conversion tool that was able to convert the triggers from the *mgl* format to the *dll* one. The problem arises when I arrived in the company, in January. In fact, at that time, Maserati hadn't direct access to the tool: so a sort of standard protocol between Maserati and ETAS was created in order to make up for this tool's lack.

As first thing, the trigger was written within Maserati in the *mgl* format using Notepad++, and eventually tested in the Moogle Test environment. When the trigger was ready and correctly working, it was sent via mail to ETAS that converted it in the *dll* format and sent it back to Maserati. ETAS wasn't able to bypass Maserati's network security system, so it had to load the converted triggers in a special sharezone with Maserati's IT. Only the IT had access to the sharezone, so I had to wait that my IT contact transferred those triggers in a path accessible to me. Only at that point I could effectively take and use the converted trigger.

This standard protocol was almost acceptable, but it had some big drawbacks:

- There were a big amount of steps, and every step means a possible waste of time. In fact, it could happen that ETAS, especially if we sent the trigger towards the end of the working day, was unable to let us have the converted trigger until the following day. This means the loss of a full processing night. Moreover, also the IT could have some other commitment, so it wasn't immediate to have the trigger transferred from the sharezone to an accessible path. Every step of those protocol could lead to a loss of time. Beyond that, ETAS responsibility wasn't to convert us the triggers, so if a big internal problem arised, the time could enormously increase.

- It could happen that we found a small mistake in the trigger after having already sent to ETAS the *mgl* file, even after the test phase, or that we had

to add some channel to the trigger. This meant that we had to send them the new version of the trigger, and consequently another waste of time-

So I asked ETAS if they were available to let Maserati have internally the conversion tool, in order to remove these time losses from both sides. It came out that ETAS was totally favourable to let Maserati own the conversion tool, but it needed, in order to correctly work, the installation on the workstation of Microsoft Visual Studio (MVS) 2010. This was a small problem, as Maserati owns the license of the 2015 version of that software, but after a relatively long discussion with the IT and the different members of PWT team, in which I acted as a mediator, it was possible to have MVS 2010 installed on my workstation; so ETAS was able to send us the tool and we were finally able to use it.

The ownership of the conversion tool, and also its incredible simplicity, improved sharply my working position. I wasn't constrained anymore to wait ETAS or the IT, but I could finally write, test and convert in a few minutes each trigger. This increased my productivity and, consequently, decreased the time needed for the creation and delivery to my colleagues of the requested statistics. This was really helpful especially in the high priority statistics, that now could really be delivered in a single day.

6.2.2 New Moogle major release

During my stage period, it was also released a new version of Moogle, containing many highly demanded features, beyond of course various optimizations and bug-fix. The most important features are the following ones:

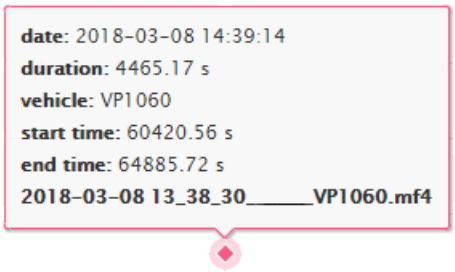


Figure 6.2: *All informations displayed on the chart when a specific point (red one in this figure) is selected*

- Additional informations directly on the chart: in the previous versions of Moogle, when the cursor was placed on a point of a chart, only the coordinates were shown, without any additional information. This was a problem in the specific case in which a specific point needed an additional study. In

fact, without any reference, it was really time consuming to find the exact measurement file in which that point was placed, so that the calibrator could study the measure. So, on Maserati's request, ETAS added more informations to the dialogue box. The informations contained in the box are here listed, as they can be seen in Figure 6.2:

1. Axis coordinates, just like the previous version.
2. Measurement file name, so that it is easier and faster to find the corresponding measure of a certain point of the chart.
3. Start and end instant. Due to the fact that usually the measurement files are hours long, knowing only the name of the file can be not enough. Having also the informations of the time instant in which that situation occurred is fundamental in order to extract quickly and efficiently that part of measure.
4. Duration of the phenomenon. As the triggers can extract very short phenomenon, but also long ones, to have directly the information about the duration of that phenomenon can be useful in order to have an information about the time duration. Long phenomenon means that the extracted part of measurement file will be bigger, and so a sort of estimation of the download weight can be done.

- PDF report: After having saved a series of charts in a report, it is now possible to directly download all them together in a single PDF file, that can then be quickly spread to all the necessary colleagues. Actually it is a feature that must be still refined, but it is an important step in order to have a complete tool.

- New status added to the crawler page of Moogle (Figure 2.2): Partial Done. Before this release, Moogle considered only two possible results: Done, if all the measurement files were correctly indexed; Failed, also if only one file wasn't correctly indexed. Of course, this system wasn't good, because it didn't allow the user to understand exactly what files failed the indexing phase, or what instead passed it, because the status was always failed. Of course it was possible to know the number of passed or failed, but not to know the name of the specific files. In this version, it is possible to download a text file that contains the name of the failed and the passed measurement files: it is then possible to look the failed ones, to understand if the files are broken, corrupted, or if simply the trigger didn't work on that specific file.

In this way the database's maintenance becomes easier and faster: knowing what measurement files must be opened is better than to open them blindly.

- Metadata management: the creation of the metadata of a measurement file is charged to ETAS INCA, where, in fact, each metadata, during the creation of the experiment, is chosen. The typical metadata are the name of the vehicle, the project, and other additional informations.

  From Moogle's perspective, having all the measurement files with the correct metadata is fundamental, because, in the filtering phase of the files on a chart, Moogle divides the points on the base of their metadata. So, for instance, if an Alfa Romeo Giulia has a wrong metadata typical o a Stelvio, in the chart it will be seen as a Stelvio. Before this version of Moogle, if an error was done during the creation of the experiment on INCA, it was impossible to avoid the aforementioned type of mistake. Now it is possible to correct them: it is possible to create, by using Notepad++, a so called JSON file, containing all the right metadata. And example of the informations contained in this type of file can be seen here:

```
{
    "vehicle" : "PVP1000",
    "project" : "Giulia_200_HP",
    "software" : "version_XYZ",
    "region" : "NAFTA",
    "driver" : "Mario_Rossi"
}
```

  On the left the specific metadata field name must be specified, while on the right it is specified the effective name or value of that metadata. It must be noticed that a JSON file must be created per each measurement file: in fact, this system works only if the JSON file has the same name of the measurement file. For this reason, the number of JSON and measurement files must be the same.

  Moogle reads the measurement files and the JSON ones, and overwrites the real metadata with the ones contained in the JSON file, but only in its internal index: the measurement file isn't modified during this operation, ensuring the integrity of the initial file.

  The strength of this solution relies in the fact that, by using the JSON file, it is possible not only to correct any wrong metadata, but also to add some

metadata that cannot be inserted in INCA, and then to use those metadata as filtering parameter in the chart creation.

- Partial reports loading: in the previous version of Moogle, when a specific report was opened in order to see the charts, the software opened all the charts of the report. If the number of charts was quite high, the loading time became significant; so in the current version it is possible to select both to load all the files, or to load only a specific one.

6.2.3 New crawler functions and other additions

The crawler is the part of Moogle responsible for the data processing and results indexing. So, when a trigger is written, the crawler takes it, processes it and extracts the data from the measurement files. All the functions that can be used within a trigger must be supported obviously by the crawler. The following functions have recently been added, or they are in an advance test phase:

- Moving average: this function was not present in the previous version of the crawler. This function creates series of averages of different subsets of the full data set: thanks to it, the possible statistic analysis that can be performed have been increased.

- Maximum and minimum values: the aim of these functions is to take, in the triggered window, the maximum, or the minimum, of a channel, and at the same time to take the values of other channels in the correspondence of that maximum or minimum point. Currently this is possible, but requires a nested trigger that simply detects the maximum or minimum: this workaround works, but the use of a nested trigger leads to a higher use of system resources.This function is already implemented in the crawler, while the user interface must be slightly modified in order to allow the user to select the results of this function.

- Automatic extraction of some values from any measurement file. With this function, it will be possible to have Moogle to extract automatically some values of a channel, without having the user to write a trigger. In fact, in the current release, if the maximum value of a channel must be studied, an apposite trigger must be written, leading to a high number of pretty useless triggers, and so to a trivial saturation of the system resources.

  The request of this feature was borne from the fact that the crawler always analyses any single measurement file, so it can automatically extract some

values, if appropriately programmed. Also for this function, it is already implemented in the crawler, but the user interface needs a slight modification in order to allow to use this function.

- Time filter: it is a function that allows the user,being valid the trigger condition, to select the temporal instant after which the trigger window effectively starts. In fact, in the current state of the tool, it is quite uncomfortable to move on the time axis within the triggered window: it is required the use of a two nested triggers, leading to an useless waste of computational resources. From the crawler perspective, according to ETAS, this function is pretty easy to be implemented, and it is in an ETAS internal test phase, before being released to Maserati.

- Rising/Falling Edge: these are functions that allow the software to understand if there is a step of a signal, both increasing (rising edge), or decreasing (falling edge). It was already possible to do something like this previously, but it was required the use of other crawler's functions in the writing of the trigger: the result was, just like the previous shown features, that the computational effort was artificially increased. Currently this is a low priority request, because, unlike the previous functions shown in this section, it is easily possible to detect rising or falling edges, and the extra computational effort isn't comparable with the ones that will be saved with the use of the others new functions.

- Metadata in the downloaded CSV: it is currently under test phase the possibility to have, directly in the CSV downloaded from Moogle, beyond the requested data, also a set od useful metadata, just like the vehicle name, the project, the installed software and so on. So it will be easier to create filtered charts directly on Excel.

6.3 Future developments

The development's direction for the future is clear: the stability, reliability and effectiveness of the software must be continuously improved, beyond obviously the bug fixing.

From the crawler's side, currently new functions are under development, in order to ease to the user the writing of triggers, and to give him more instruments in order to create better and more detailed statistics.

From the UI's side, it is under development a restyling of the interface, so that the creation of reports will be faster and easier, and also graphically a better result will be obtained.

Conclusions

As the amount of data that must be studied in order to guarantee that the modern vehicles are allowed to move in all the countries is increasing, standard data analysis instruments are not fast and reliable enough. New ways of analysis are then required, and one of the possible choice is the tool ETAS Moogle.

After the acquisition of the data from the fleet's vehicles, it is possible to exploit the potentialities of Moogle in order to analyse the data in acceptable times. Nowadays the management of any company, Maserati included, needs to take decisions on the future of the company itself, and in order to do it, they need a series of data that can guide their choice. These decisions must be taken as fast as possible, and only by using such a tool this request can be accomplished.

Moogle, during my working period, has become a fundamental tool within the company: now, every member of the Powertrain department knows about it and its capacities, and can use it in order to understand the correctness of a calibration, to validate every modification applied to the vehicles each year, and to help the faults diagnosis.

The road for the future is clear: to continue using the tool within Maserati, while actively working together with ETAS in order to understand how to improve the software, so that an efficient roadmap could be drawn. Probably, the role of this tool is going to become even bigger than now, especially if it will be joined by other instruments that will increase the report's productivity rate and clearness, so that it will be easier and easier also for the management to take difficult and important decisions, and for the others Powertrain members to understand if their job was correctly done.

The final aim of the use of such a tool is to eventually be able to decrease, year after year, the number of vehicles of the Validation fleet, using the data of the previous years as a basis for the calibration and validation of every possible news; moreover, this means also a huge economic saving for the company.

Bibliography

[1] Multiple authors. *Automotive Handbook, Ninth edition*. Wiley, 2014. Chap. Automotive Electronics, pp. 1252–1253 (cit. on p. 11).

[2] Multiple authors. *Automotive Handbook, Ninth edition*. Wiley, 2014. Chap. Automotive Electronics, pp. 1260–1264 (cit. on p. 11).

[3] Multiple authors. *Automotive Handbook, Ninth edition*. Wiley, 2014. Chap. Automotive Electronics, p. 1271 (cit. on p. 12).

[4] ETAS. *ETAS ETK*. 2018. URL: `https://www.etas.com/en/products/etk_fetk_xetk_ecu_interfaces.php` (cit. on p. 13).

[5] ETAS. *ETAS ES720*. 2018. URL: `www.etas.com/en/products/es820_drive-recorder-module.php` (cit. on p. 14).

[6] ETAS. *ETAS MDA*. 2018. URL: `https://www.etas.com/en/products/mda.php` (cit. on p. 15).

[7] Andrej Trnka. "Big data analysis". In: *European Journal of Science and Theology* 10 (2014), pp. 143–148 (cit. on pp. 19, 20).

[8] Rob Kitchin. "Big data, new epistemologies and paradigm shifts". In: *Big Data & Society* (2014) (cit. on p. 20).

[9] Multiple authors. *Automotive Handbook, Ninth edition*. Wiley, 2014. Chap. Emission control and diagnosis legislation, pp. 566–576 (cit. on pp. 21, 22).

[10] Giancarlo Ferrari. *Internal Combustion Engine*. Società Editrice Esculapio, 2014. Chap. Combustion Processes, pp. 198–201 (cit. on p. 39).

[11] Wang Zhi, Liu Hui, and Reitz Rolf D. "Knocking combustion in spark-ignition engines". In: *Progress in Energy and Combustion Science* 61 (2017), pp. 102–103 (cit. on pp. 41, 42).

[12] Multiple authors. *Automotive Handbook, Ninth edition*. Wiley, 2014. Chap. Management for spark-ignition engines, pp. 633–635 (cit. on pp. 44, 47).

[13] Multiple authors. *Automotive Handbook, Ninth edition*. Wiley, 2014. Chap. Automotive Electronics, p. 1327 (cit. on pp. 45, 46).

[14] W. Addy Majewski. *Dieselnet*. 2018. URL: www.dieselnet.com/tech/gasoline_particulate_filters.php (cit. on p. 53).

[15] Association for emission control by catalyst. "GASOLINE PARTICULATE FILTER (GPF)". In: *AECC article* (2017) (cit. on pp. 53–55).

[16] INFINEUM. *Insight GPF*. 2018. URL: www.infineuminsight.com/insight/jan-2018/gasoline-particulate-filters (cit. on pp. 54–56).

[17] California Environmental Protection Agency. *California's Advanced Clean Cars Midterm Review*. California Environmental Protection Agency, 2017. Chap. Appendix J: Vehicle PM Emission Control Technology Assessment, pp. 21–24 (cit. on p. 56).

[18] Federico Rulli. "Sviluppo di un modello predittivo del comportamento fluidodinamico di Gasoline Particulate Filter (GPF) e caratterizzazione dell'invecchiamento (Translation)". Experimental thesis. Università degli studi di Modena e Reggio Emilia, 2016 (cit. on p. 59).