Politecnico di Torino

Department of Mechanical and Aerospace Engineering
Master of Science in Automotive Engineering

Big data analytics role in Maserati powertrain systems development. Case study: continuous improvement of IUMPR indexes.

Academic Rapporteur:
Ch.mo Prof.
GIANCARLO GENTA

Corporate supervisors:
Ing. ALESSANDRO PAONE
(Maserati S.p.A.)
Ing. JACOPO GABRIELLONI
(AKKA Technologies S.p.A.)

Candidate:
LEONARDO RATEGNI
Matr.: 251560

A.Y. 2019-2020

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Leonardo Rategni

Contents

List of Figures

List of Acronyms

AIS	*Air Intake System*
ASAM	*Association for Standardization of Automation and Measuring systems*
BD	*Big Data*
CAN	*Controller Area Network*
CARB	*California Air Resource Board*
CCP	*CAN Calibration Protocol*
CCV	*Canister Check Valve*
CN6B	*China's Stage 6 emission stadndard*
CPK	*Process Capability Index*
CPU	*Central Processing Unit*
CPV	*Canister Purge Valve*
CSMA	*Carrier Sense Multiple Access*
CV1	*Check Valve 1*
CV2	*Check Valve 2*
CV3	*Check Valve 3*
DMTL	*Diagnostic Module Tank Leak*
DTC	*Diagnostic Trouble Codes*
ECU	*Electronic Control Unit*
EGR	*Exhaust Gas Recirculation*
EMC	*Electromagnetic Compatibility*
EMEA	*Europe, the Middle East and Africa*
EOBD	*European On-Board Diagnostics*
ESIM	*Evaporative System Integrity Module*
EVAP	*Evaporative Emissions Control System*
FTPS	*Fuel Tank Pressure Sensor*
GEPI	*Società per le Gestioni e Partecipazioni Industriali*
GME	*Gasoline Medium Engine*
HC	*Unburnt HydroCarbons*
IEEE	*Institute of Electrical and Electronic Engineers*
INCA	*Integrated Application and Calibration Tool*

IUMPR	*In-Use Performance Ratio*
MDA	*Measure Data Analyzer*
MDF	*Measurement Data Format*
MIL	*Malfunction Indicator Lamp*
NA	*North America*
NEDC	*New European Driving Cycle*
OBD	*On-Board Diagnostics*
OPL	*Open Point List*
PWM	*Pulse-Width Modulation*
SHED	*Sealed Housing Evaporative Determination*
S&S	*Start&Stop*
TC	*Turbocharger*
TWC	*Three-Way Cathalist*
VDR	*Vehicle Data Recorder*
VOC	*Volatile Organic Compounds*
WLTC	*Worldwide harmonized Light-duty vehicles Test Cycles*
XCP	*Universal Measurement and Calibration Protocol*

Summary

The purpose of this document is to illustrate the internship experience I had at Maserati S.p.A. and to explain the results I obtained with my daily duties. As member of the Validation and Fleet Management team — a branch of the Powertrain Testing department of Maserati S.p.A. — I had access to the complete sets of data regarding every Validation fleet's vehicle accumulation schedule. Since those programmed activities imply the generation of huge amounts of data everyday, these ones can be gladly and fully-fledged addressed as Big Data. Pursuant of their extreme dimensions and thickness, these data are not manageable with traditional data analysis techniques, and require particular studies to exploit, give meaning and extrapolate usable indications to and from them. Big Data are surely a very precious and breaktroughing instrument, which — if properly used — can either definitely determine the success of a company with respect to competitors, or at least increase the quality and the profitability of its products in the market segments of interest.

In the following Chapters, I have explained how Big Data are retrieved and stored within Maserati, and how this experimental philosophy is orienting the calibration cycle towards finer and faster results settlement. Furthermore, I have also described in detail the functionality and the usage of the developing software ETAS Moogle, which has been the main tool I have experienced and which is used as interface to manage those data and extrapolating usable, structured and synthetic indications. Eventually, I have described a practical application of how Big Data analysis can practically affect the improvement of vehicle performances, in the form of IUMPR indexes enhancements. These ones (the acronym "IUMPR" stands for *In-Use Monitoring Performance Ratio*, i.e. quantitative performance indicators of every specific emission-related variable that can be measured in a vehicle) are continuously-monitored quantities, whose value is real-time evaluated and stored by the engine electronic control unit for reporting purposes. Thanks to the remarkably high aggregating and didactic potential of this data analysis tool, it has been possible to identify a software miscalculation that returned unrealistic values of IUMPR — which could gave reliability problems against Authorithies inquiries — and to discern a way for its correction. The results of my study demonstrated the outstanding improvement possibilities that the analysis of Big Data implies, which could be a breaktrough attribute in nowadays automotive industry.

Acknowledgements

The present document is aimed at illustrating the internship experience of the undersigned candidate Leonardo Rategni within the motor vehicle OEM Maserati S.p.A., in collaboration with the consulting agency AKKA Italia S.r.l. The internship has been exclusively carried out inside the Validation and Fleet Management team — which is in turn part of the PWT Testing Department of Maserati S.p.A., sited in viale Ciro Menotti, 322, 41121 Modena (MO) — on behalf of the abovementioned AKKA Italia S.r.l., which had supplied technological resources and software licenses to be used during the whole job assignment. In these few lines, I would like to express my deepest gratitude for all the people that have someway partnered with me in the drafting of this document, offering their frienship, their support and precious advices.

Alessandro Paone — leader of the Validation and Fleet Management team whom I have worked for — that has been subsequently my main and daily corporate reference, both in technical and logistic terms. Encouraging me to address him openly from the very beginning, as it is used to do with a good friend, he allowed me to rapidly integrate in the Company, so that both my experience and the collaborations established with other corporate entities could be as fruitful as possible. Furthermore, he closely followed the initial, cumbersome allowances assignment process, for both entering the Company and accessing the corporate network with a proprietary account, sustaining me in a emotionally tricky moment.

Andrea Palma — technical leader of the abovementioned Validation team — for whom there is no shortage of statements. Andrea has been my Master under all points of view, welcoming positively my idealist and speculative nature and answering to all my questions with huge involvement and didactic spirit. There have been frequent times in which we have literally flown with fantasy, trying to figure out the reasons behind unintelligible or well-established facts, without ever neglecting a necessary and cathartic trait of irony. Although he dedicated an absolutely not obvious quantity of time to me, which it really honours him and that I am really grateful for, he was

capable of being strict and absent everytime I needed to get out of the quagmire alone. If I managed to reach a level of detail that could satisfy both the requirements that are used to a Master of Science thesis, and my personal aspirations to learn a not discounted and easy job as well, it is only merit of the availability and to the enthusiasm that he has given me. I cared about every single word and explanation he gifted me with great respect and admiration, and it is with these feelings that I will remember our collaboration from now on.

Jacopo Gabrielloni — employee in Calibration OBD-II team of Maserati S.p.A., as external consultant on behalf of AKKA Italia S.r.l. — allowed me to get more insights about the characteristics of the evaporative emissions control system that I have treated in Chapters 4 and 5, upon which the *Case study* mentioned in the title of this dissertation is based. His contribute has been fundamental, not only for what concerns with a thorough comprehension of that system, but also for the development and the refinement of these months' analyses, both the ones actually included in this document and the others that had anyway been part of the job. Eventually, he has kindly given appreciated support during the delicate period for the assignment of the allowances from AKKA's side.

Giancarlo Genta, who has been my professor for the courses of *Motor Vehicle Design* and *Automotive Evolution* during my biennium as Automotive Engineering student at Politecnico di Torino. As for Andrea, also in his case there is no shortage of words. I am probably the last one of a very long list of admirers who had the pleasure to receive teachings and testimonies from him, hence I will not dwell on the enumeration of his overt expertise. The only characteristics that I want to thank, since they have been very important to me, are three: at first, the precision and the ease with which he can transform his knowledge into living stories and innovative perspectives, adding something recognizable and distinguishable to the students' understanding. Secondly, his old fashioned manners — both from the emotional and professional points of view — are certainly a precious and rare lesson of ethic and style in a modern world so permissive and lapse. Eventually, the care and the literary pledge with which his books are written have inspired me to do the same on my thesis. He has been a model for the engineer I want to become in the future, and that is the reason why I have chosen him as Academic Rapporteur.

All the people of the Validation and Fleet Management team in Maserati S.p.A., which has been my home for six months and which has been thoroughly included in the picture below.


MASERA

"True innovation is the fruit of a single mind".

Marcello Gandini, automotive designer.

Forewords

Whoever is born italian and motoring enthusiast as well, should feel very lucky. Hidden behind national-popular glows of entertainent, television, chauvinism and folklore, a stately and picturesque automotive tradition still looms, variegated and easily recognizable despite its age. The reason of such a persistent survival may be attributed to a simple fact: the one and only innovation and enhancement driver of the homegrown automotive industry has always been the *passion*. This was particularly evident especially in the first half of XX Century, when automobiles were still considered a luxury item, and anyway not necessary to the ordinary citizen life: the construction of a car, with all the technical and technological challenges it implied, was foremost a demonstration — to themself and to the rest of society, alike — of the demiurgic potential of Man, who found itself placed back at the center of progress (assisted by an increasingly involved and stimulating cultural context like the Futurist one) of which he could be the arbiter and advocate.

Today society has deeply changed, and new economic and social needs — which had manifested over the years — had partially clouded the creative vein upon which car production has always had its roots. In spite of this, the tales of some families, some brands, some factories and products — which have contributed to gift the italian automotive tradition a halo of charm, care, elegance and sophistication, almost on the border between myth and reality — still survive in the present.

To understand how significant, still today, is the link between cars and italians it is enough to head to Modena, Emilia Romagna, central Italy; in this middle land, far from the drumming and ceaseless echoes of the larger and faster Milan and Turin, as well as from the idealism and oestrum of Bologna, there is an equally industrious, proud and committed daily life. In the north of the city, in the nearby of the railways and at a short distance from the centre, a magnificent mirrored octagonal tower stands out on the horizon, holding a red trident on its top (Fig. 1). I guess very few people do not know the meaning of that symbol — which is visible from many different city areas, thanks to its favourable position — provided with a communicative power with few equals in the automotive world. The encumbrance, the minimalism of its architectures and the standing of that tower, in a relatively confined space in which the industrial pole of the company fits, are a metaphor for the enormous impact and meaning that Maserati has always had on Modena and — in a broad sense — that the automotive industry has always had on Italy.

Figure 1: Footage of viale Ciro Menotti 332 Maserati plant, the directional centre of Maserati S.p.A.

This ending dissertation, which closes my intense and stimulating experience at Politecnico di Torino, relates the internship experience of a guy who finds himself catapulted in a large-company dynamic reality for the first time, with proven habits and mechanisms and requiring fast and autonomous adaptation. I soon realized that the preparation that my course of study offered me was fundamental to achieve a rapid and effective alignment with the requests of the working team to which I was assigned. The language, the procedures, the documentation, the habits, the power relationships, the timing, the diligence and many other features that define the *status quo* of a company have given me the references within which adapting my knowledge, my personality, my skills and my need to learn. Such a step is not automatic and requires to leave the student mentality — too often linked to perfection in performance and protected by a rather invariable environment over time — to embrace a new perspective, in which the ability to take the initiative and accept schedule changes, as well as the ability to relate to a wide range of different personalities, take on central importance.

Chapter 1

Company presentation and history

If you think about the brand Maserati today, luxury, prestige and wealth are probably the first ideas you come up with, anyway it has not ever been likewise. There was a time in which Maserati was a synonym of *racing*, similarly to Ferrari today: it was not only a successful car maker — especially in the racing world, where it ventured from the very beginning to almost the ‘60s — but it was also a laboratory of craftmanship, passion and innovation.

Although it achieved many important trophies at both national and international level, Maserati faced several periods of financial instability, which led to as many changes of property. Because of those vicissitudes, the nature and the philosophy of the brand started gradually changing, mostly due to the impossibility of sustaining the economic efforts required by races, which were becoming more and more competitive and exclusive starting from the ‘50s. Thus, the Trident was forced to open up to large-scale market, with a significant reduction in innovation and exclusiveness of its technical and productive solutions. This conversion at least allowed the company to financially survive as a standalone entity, but inexorably spoiled the brand sporty and high-quality image.

It was only in the ‘90s that, thanks to the healing action of Ferrari’s administration, Maserati was elevated to the status of luxury division of the world’s most famous brand, recovering part of its lost reputation. Starting from 2000, the brand was renewed in style and received financial sustain from the creation of the newborn FCA[1], as a result of the fusion between the former Fiat and Chrysler groups. Today, the brand is waiting for some huge and important investments that will lead to the commercialization of a new

[1] Fiat-Chrysler Automobiles, ed.

full-electric sport car in 2021 and a new D-segment SUV in 2022. The best *seems to be* yet to come.

1.1 The dawn

The history of the Aemilian brand officially began on december 1st, 1914, in Pepoli st., Bologna. Here, after some years of training experiences in the former Milanese luxury car maker Isotta Fraschini, the three brothers Alfieri II, Ernesto and Ettore Maserati founded in Bologna the ***Società Anonima Officine Alfieri Maserati*** (Fig. 1.1), a specialized factory working on the development and adaptation of Isotta Fraschini and Diatto engines to urban and extra-urban road racing. This company — which was initially composed by just five workers, included the abovementioned Maserati brothers — could bank on the valuable experience gained in a huge and innovative company like Isotta Fraschini, which at that time was at the pinnacle of its legacy, and on the brilliance and inventiveness of Alfieri II, who had started working on the tuning of their racing cars as youngster. Unfortunately, the activity of the company was soon interrupted due to the outbreak of WWI, in which Alfieri II and Ettore had been enlisted; during this period, the company was led uniquely by Ernesto Maserati.

Figure 1.1: From left to right: Carlo, Ernesto, Bindo and Alfieri II Maserati in front of their *Società Anonima Officine Alfieri Maserati.* Photography of the ‘20s.

After some time, Alfieri II was prematurely demobbed and could grapple again on mechanics. It was in this period that he concepted (and successively patented) an innovative **spark plug**, which gave him the opportunity to initiate a complementary company in Milan, the newborn *Fabbrica Candele Maserati*, devoted to their production (Fig. 1.2). At the end of the conflict, this company was relocated in Bologna suburbs, near the elder *Officine Alfieri Maserati*, to rejoin forces with his brothers: the core business of the two companies was still the elaboration and development of Isotta Fraschini cars, but in this period they started to open up also to different car makers.

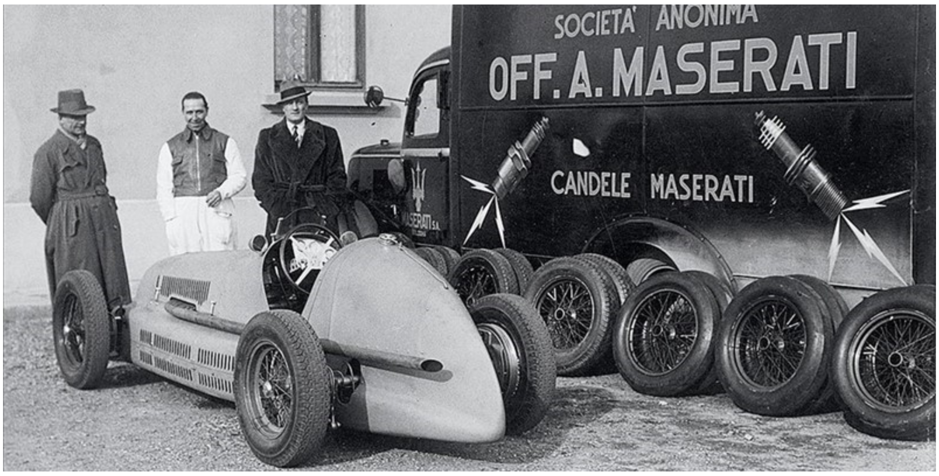


Figure 1.2: Carlo, Ernesto and Bindo Maserati pose next to their spark plugs factory commercial equipment, 1935.

These businesses were not enough for Alfieri II, who was craving for creating something of his own: following his inclinations and his outstanding talent in mechanics, he ventured on the construction of a prototype — which mounted a Hispano-Suiza engine of aeronautical derivation, but properly modified — based on a Isotta Fraschini chassis. With this frankly attempted and artisanal vehicle, he still managed to win the Susa-Moncenisio, Mugello and Aosta-Gran San Bernardo races in 1921, together with his brother Ernesto. This can be considered the start of the racing pilot career of Alfieri II Maserati, which would have been soon enriched by plenty of emotional successes.

His remarkable achievements did not go unnoticed, as the Turinese car maker Diatto offered Alfieri II the executive direction of the brand's sport activities, as well as the role of official driver in racing competitions. Alfieri II accepted this role as part-time job, until the 1924 early closing of the project due to financial debts. Exploiting the difficulties of Diatto, and thanks to the support of marquis Diego de Sterlich — which was vehemently fond of racing and were a tight friend of Maserati family — the *Anonymous Society* managed to acquire thirty Diatto chassis, upon which it became possible to start an autonomous and self-branded production.

In 1926 the very first Maserati was built: although it was just an evolution of the Diatto GP 8C Turbo, it already featured, for the very first time, the worldwide famous Trident logo (Fig. 1.4a), and was called **Maserati Tipo 26** (Fig. 1.3). According to tales, the symbol was concepted by Mario Maserati (Fig. 1.4b) — i.e. the only one, out of five brothers, to be scarcely interested in motors and to be working as an artist — upon suggestion of the abovementioned patron de Sterlich, who taught it could spread the idea of royalty and power. The logo design was inspired by the *Nettuno Fountain* (Fig. 1.5) in Piazza Maggiore, in the city centre of Bologna, as well as the official colours recalling the city's banner.

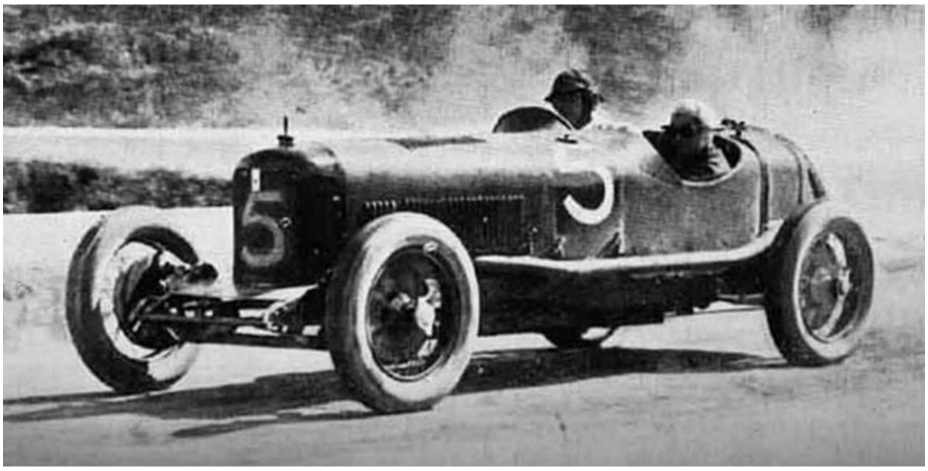


Figure 1.3: Footage of Maserati Tipo 26 competing in 1926 "Targa Florio" Sicilian race.

Other than being famous as the first ever Maserati-branded vehicle, this car succeeded in winning its Class in the well-known Targa Florio in 1926, with Alfieri II as driver. Unfortunately, things started getting worse for him, as in 1927 — when his racing pilot career was still emerging — during Coppa Messina race in Sicily he suffered a dramatic accident in which he lost the use of a kidney. Although he survived, he will never manage to completely recover from that injury. During this period of forced recovery, in which Alfieri II could not compete, Maserati succeeded in winning many different minor races thanks to *customer-drivers*[2] opuses, who helped in increasing the

[2] *customer-drivers* were basically passionate and wealthy customers who spontaneously

prestige of the company, offering at the same time technical and economic support.

(a) Maserati "Trident" Logo on its *Tipo 26*.

(b) Sketch by Mario Maserati (1926), today housed in *Maserati Showroom* in Ciro Menotti st., Modena (Italy).

Figure 1.4: The first documented *on-vehicle* usage and drafting of the world-famous Trident symbol.

Figure 1.5: *Nettuno Fountain* in Piazza Maggiore, Bologna (Italy).

One of the most remarkable ones was the 1929 Coppa dell'Etna victory, thanks to the driver Baconin Borzacchini. He also achieved the world speed

offered their proven and time-refined technical knowledge to improve the tuning and adaptaion of their vehicles' chasses to road racing. This unusual collaboration was beneficial for both parties: on one hand, the drivers could enhance their racing experience with superior and tailored development, directly concerted with the Company technical leaders (mainly Alfieri II Maserati himself). On the other hand, the Company could bank on free and aware development from skilled and involved enthusiasts, increasing the quality and reliability of their products.

record on 10 km launched (246,069 *km/h*) in the following year, with the new-born Maserati V4 (Fig. 1.6), equipped with a 16-cylinders engine: this had been basically derived by arranging in parallel two 8-cylinder engines — of the same type mounted on the abovementioned Tipo 26 — and could provide a rated power between 280-305 HP, at 5500 *rpm*. Due to a high power/weight ratio, although the car was pretty performing and competitive, the handling and stability of the vehicle was quite poor, and an enlargement of the wheelbase became necessary to avoid serious oversteering behaviors. The following season, in 1930, Borzacchini also won the brand's first international award in the Tripoli GP[3]: these achievements hugely contributed to renown the reputation of the brand, increasing the amount of investors and usable capitals that allowed to expand and diversify the company activity.

Figure 1.6: Maserati V4 16-cylinders, which achieved the launched world speed record in 1930 with Baconin Borzacchini as driver.

1.2 Alfieri II passing and first major racing successes

In 1931 two new cars started to be designed: Maserati 4 CTR and Maserati 8C 2500, which was actually the last car concepted by Alfieri II Maserati.

[3]At that time, Libia was still part of Italian territory, ed.

Unfortunately, in fact, he would have died on march 3rd, 1932 due to consequences of unsuccessful surgery, as his health conditions were anyway slowly decaying with time. After his passing, Ernesto Maserati decided to abandon the career as professional racing driver to take over the company, together with Ettore. Moreover, they also appointed the elder brother Bindo, who had continued working for Isotta Fraschini in the meantime.

In spite of this saddening and trumatic event, Maserati's racing activity continued to be very intense and successful: starting from 1933, for instance, the arrival of the famous driver **Tazio Nuvolari** gave a huge technical contribution, especially in the chassis improvement and in the engine adaptation. At that time, he could already be considered a major driver, since he had been 1932 European Champion, two times Speed Italian Champion and had won a lot of GPs. Actually, he managed to prevail, in the same year of his Trident debut, in the Belgian GP — starting from the last place — with a modified 8C 3000, as well as in Nizza and Montenero GPs. In parallel, Giuseppe Campari managed to win the first race of the brand in the *Grandes Épreuves*[4] in France. Then, the arrival of Mercedes-Benz and Auto Union to the top of the racing competitions determined a harder challenge for Maserati, which seldom managed to prevail. On the other hand, from minor races' side many awards continued to come: this fact convinced the brand administration to orient the company production towards this sector, rather than the *Grandes Épreuves*'s one.

In 1936, the company finds in Gino Rovere another patron: he hugely invested in Maserati competitiveness in the mid-engines racing sector, initiating the production of the newborn Maserati 6CM. The drawback of this passage was a change of leadership to the top of the company, which started bringing to light a progressive departure of Maserati brothers from the company: it was in this period — for instance — that Nino Farina, a Rovere's scion, became the company's President. In 1937, although they were not really in financial difficulties, Maserati brothers sold the property of the company to Modenese entrepreneur **Adolfo Orsi** and his family, who came from the steel industry, foundries, and metal manufacturing. From that deal on, the two distinct societies based in Bologna — *Officine Alfieri Maserati*, dedicated to racing cars production, and *Fabbrica Candele Maserati*, devoted to the production of spark plugs, batteries and electric components — were merged and relocated to the actual place in Modena.

[4]The term, which stands for *Great Trial*, began to be officially used in the second decade of the XX Century, to define the eight most important races of the year before the advent of Formula One Championship. Races in that period were heavily nationalistic affairs, with a few countries setting up races of their own, with no formal championship tying them together until 1931.

A period of major changes was on the way, in particular from the point of view of the company's management and policies: it can be said, in fact, that the process of transformation of the company *nature* — from a mere racing and performance-oriented mentality to a more organized, sophisticated and production-focused one — was engaged in those years. In 1951, for instance, the Trident logo was encolsed in an oval-shaped form for the first time, having two cusps at their top and bottom (Fig. 1.7a). Then in 1954, maybe for lack of policies on the matter, the oval shape appeared also with more rounded edges (Fig. 1.7b).

Thus Bindo, Ernesto and Ettore Maserati, free from management duties, decided to subdivide the operative divisions of the company into three distinct compartments — the commercial and financial part, the design and concept part and the spark plugs factory — equally distributed among the three brothers.

(a) Maserati "Trident" Logo, 1951. (b) Maserati "Trident" Logo, 1954.

Figure 1.7: Maserati logos in the '50s. Two versions were contemporarily used for lack of official directives.

In the meantime, the company managed to come back to the top of world class racing championships: the first entirely concepted and produced vehicle under the new directive board, i.e. the Maserati 8 CTF (Fig. 1.8), succeeded in winning at its first attempt the **1939 Indianapolis 500 race** with Wilbur Shaw, embellished by a second victory the following year (1940). Today, this is still the last Indy500 win achieved by an italian engine. Between 1937 and 1940, Maserati gained other four victories in the well-known Sicilian race

Targa Florio, at the Favorita Park: the first three with a 6CM (drivers were respectively Giulio Severi, Giovanni Rocco and Luigi Villoresi), while the fourth one occurred with a 4CL with Luigi Villoresi.

Figure 1.8: Wilbur Shaw, at the wheel of the "Boyle Special" Maserati 8 CTF, wins the Indianapolis 500 at an average speed of 185.131 km/h (115.035 mph, ed.), 1939.

1.3 Pinnacle and Maserati brothers departure

During WWII, Maserati temporarily arrested the production of cars to deal with the increasing demand of machine tools, spark plugs and electric vehicles, which were needed by the defence industry and had less reliability problems with respect to internal combustion engine-based vehicles. At the end of the war, the core business was slowly restored, despite the increasing concerns from the patron Adolfo Orsi about races. Maserati brothers, who still retained the technical and financial direction of the brand, continued igniting the racing dream with the production of a brand-new GT model

(Maserati A6 1500). In the meantime, on the other side of the Atlantic, the Trident managed to win two consecutive editions of the legendary Pikes Peak uphill race, with Louis Unser on Maserati 8 CTF 3000 (Fig. 1.9). Anyway, the conceptions of the company of the Maserati brothers and the Adolfo Orsi's one started to depart quickly, and this was one of the reasons of the exit of the formers from the company they had founded more than thirty years before. It was in this period, after the end of their 10-year contract with Adolfo Orsi, that Ernesto Maserati — as engineering manager — and the brothers Bindo and Ettore Maserati — as operations managers — founded **O.S.C.A.**[5], another successful activity by Maserati brothers, based on the production of competitive racing vehicles, which remained in business up to 1968. This is another story, though.

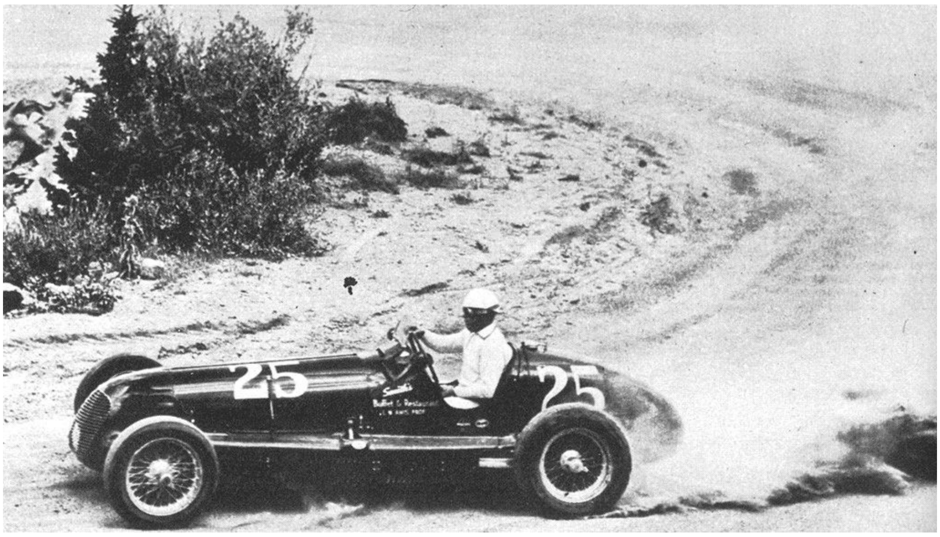


Figure 1.9: The legendary 6-time Pikes Peak champion Louis Unser wins the uphill race upon Maserati 8 CTF 3000.

Maserati A6 GCS, which Alberto Ascari won the Modena Grand prix with, was the last car designed by Maserati brothers before leaving the company. In 1947 the series production began, and the model was commercialized to the public with the name A6 1500 (Fig. 1.10) at the Geneva Motor Show. The car body was designed in collaboration with the designer Giovanni Battista "Pinin" Farina and featured a 1.5 liters, six-cylinder in-line engine. In the meantime, racing awards continued to snow down: in 1948 the Trident had a big comeback in the *Grandes Épreuves* series, winning firstly

[5] it is the acronym of *Officine Specializzate Costruzione Automobili — Fratelli Maserati S.p.A.*, ed.

the Monte Carlo GP after 15 years, and then Silverstone GP, both with Giuseppe Villoresi on Maserati 4CLT. Moreover, the victory in the British GP was repeated the following year by the swiss Toulo de Graffenried.

Thanks to the reputation it gained over years of professional championships and the mastered craftmanship, Maserati was invited to participate in the first **Formula One World Championship** in 1950. The first podium came in the second race of the season thanks to the monegasque driver Louis Chiron, who finished the home GP in the third place (Fig. 1.11). However, the '50s started uphill for the Trident, which had to challenge emerging super-powers as Ferrari and Mercedes-Benz. After two years, in 1952 the best driver in the world at that time was hired: its name was **Juan Manuel Fangio**, from Argentina. Actually, he would only debut in 1953 because of an accident he had in Formula Two, which forced him to skip the whole season.

Figure 1.10: Maserati A6 1500, the first Trident car designed by Giovanni Battista "Pinin" Farina.

In 1953 engineer Gioacchino Colombo was hired in the Squadra Corse team, with the task of improving and mastering the A6 GCM, to allow Fangio to be competitive for the title. Maserati 250F (Fig. 1.12), designed to participate in the F1 1954 World Championship and featuring a 240 HP 2.5 liters engine, is still today considered one of the best cars of all time. Fangio is under contract with Mercedes-Benz but, waiting for the German car to be ready, asked for permission to race the first two GPs of the year (Argentina and Belgium) with the car of the Trident, which he actually won with. At the end of the season, also pursuant of the two victories he obtained with Maserati, he became Formula One world champion.

Figure 1.11: Louis Chiron races the Monte Carlo GP, 1950. It will be the first Formula One podium ever for Maserati.

Figure 1.12: Juan Manuel Fangio on his Maserati 250F, 1957. With this car he became fifth-times World Champion, winning four GPs out of seven, plus two second places.

In 1956, the Trident enrolled the legendary British driver Stirling Moss, which managed to win two GPs (Monte Carlo and Monza), allowing him to finish in third place in drivers' standings. Moreover, together with the Argentinean driver Carlos Menditéguy, he conquered in Buenos Aires the first ever World Sports Prototype race with a Maserati 300S. But the best is yet to come: 1957 is the most successful year of Maserati history. Fangio becomes F1 world champion with four wins (Argentina, Monte Carlo, France and Germany) and two second places out of seven GPs. It was the first Formula One world champion title for Maserati and the fifth of his career (a record which has been equaled just by Michael Schumacher in 2003 and by Lewis Hamilton in 2018).

1.4 Financial crisis and large-scale production conversion

In 1958 Adolfo Orsi found himself in financial crisis, due to the failure of a huge trade — which should have involved the selling of many milling machines, coming from other collateral activities of the Orsi family — with the Argentinean government. He managed to save the company from bankruptcy by selling both various personal properties and the machine tool division to a foreign company and obtaining the controlled administration regime for one year. Eventually, he succeeded in paying all his debts, but was forced to announce the shut down the *Squadra Corse*, settling all his employees elsewhere; some of them found new professional challenges in Ferrari, the most bitter rivals. Anyway, Maserati would still have continued challenging on racing cars, both by producing prototypes for private teams (e.g. the Maserati *Birdcage*), and by equipping other Formula One teams with Maserati engines (e.g. 1965 Cooper, with a V12, three valves Maserati engine).

In this moment of financial uncertainty, which led to a steep decrease of interest and economical sustain for the racing world, the production of ordinary cars gradually became more and more pronounced. In the same year, the company manufactured the first Maserati-branded vehicle built on purpose for large-scale volumes: the **3500 GT**, designed by body-maker Touring (Fig. 1.13). This model signed the transformation of the company into a full-title car manufacturer, with the racing activities left to the margin.

The '60s opened with the latest success in the World Sports Prototypes series: a Maserati Type 61 engineered by the US Camoradi team (Fig. 1.14), won the 1000 km Nürburgring of 1960 and 1961 with Stirling Moss and Lloyd Casner, respectively. This car had been especially studied for long-distance endurance races, like 24 Hours of Le Mans; it featured front, longitudinal engine with rear wheel drive transmission, mounted on an innovative chas-

Figure 1.13: Maserati 3500 GT can be considered the first Trident car specifically built for mass market, 1959.

sis made of tubular structures composed by chromium and molybdenum (Fig. 1.15). This structural solution allowed to strengthen the frame with a consistent reduction of weight due to lighter materials employed, and was one of the many reasons that made this car one of the most estimated front-engined racing cars of that period.

Figure 1.14: Maserati Type 61 "*Birdcage*", engineered by US Camoradi team, winner of the 1000 km Nürburgring in 1960 and 1961.

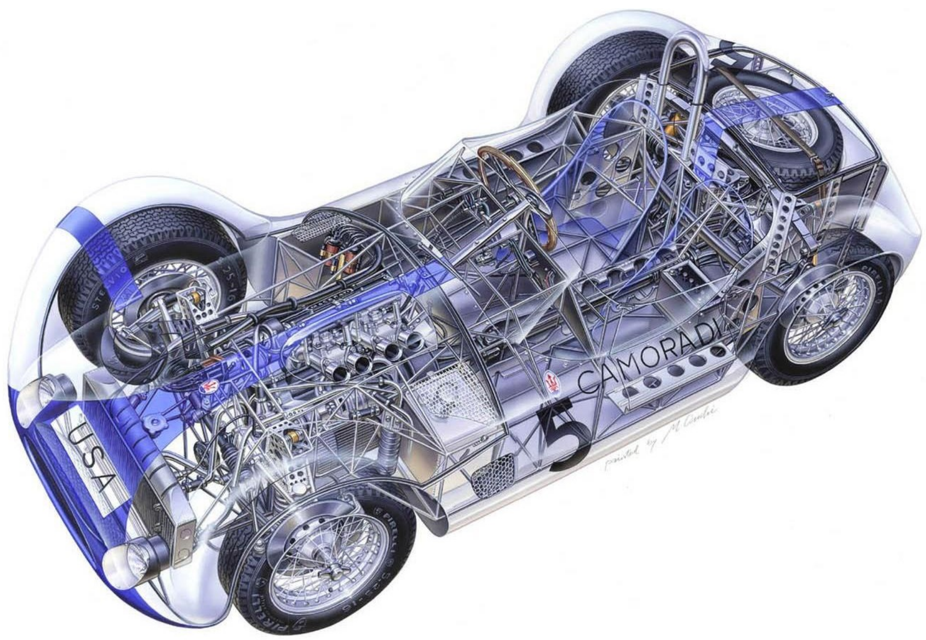


Figure 1.15: Maserati Type 61 "*Birdcage*" chassis, made by lightweight materials and innovative thin-tubular structures.

In the meantime, the car production for the large-scale market proceeded on a roll, and the 3500 GTI — i.e. the first italian car featuring an injection-based engine — was presented, followed by 1962 Maserati Sebring. In 1963 two masterpieces of the Trident born: Maserati Mistral (Fig. 1.16), which was the last Maserati equipped with the legendary six-cylinder, in-line engine derived from the 250F of the Fangio's years, and the first series of **Maserati Quattroporte**, presented at the Turin Motor Show as the fastest sedan in the world, powered with a 260 HP, 4.1 liters, V8 engine with 90° span, able to reach a maximum speed of 230 *km/h*. In 1967, the last racing win of a Maserati F1 engine occurred, thanks to the Cooper-Maserati victory in the South African GP, with the Mexican driver Pedro Rodríguez. In the same year the Ghibli was born, a famous model designed by the worldwide famous Turinese designer Giorgetto Giugiaro.

Although the production was constantly increasing, as well as the technical innovation presented in each model, in 1968 the company was acquired by **Citroën**, with Adolfo Orsi maintaining an honorary participation. The partnership allowed the French company to use the more sophisticated and reliable engines of the Modenese brand to enhance the quality and the appeal of the range, while the Trident could acquire some breakthough technologies of the transalpine manufacturer, such as hydropneumatic suspensions which had been successfully implemented in the contemporary and worldwide fa-

Figure 1.16: Maserati Mistral, the last six-cylinder, in-line engine of the Trident.

mous Citroën DS. In the meantime, Maserati occasionally prepared Citroën racing vehicles with proprietary engines: a famous example was the Citroën SM, which the French company won the Marocco GP with.

In 1971 Geneva Motor Show the first central longitudinal engine car of the Trident, the Maserati Bora (Fig. 1.17), was presented. It was a mid-engined coupé, specifically designed by Giugiaro with the aim of stealing customers from the market segment of the upcoming Lamborghini Miura and De Tomaso Mangusta. Furthermore, entirely independent suspensions were mounted on this car for the first time in Maserati's legacy. The advanced Citroën hydraulics was adapted to action the self-ventilating disc-brakes, the power steering, the front retractable lights, the steering column and last but not least the single-disc clutch. From the engine's side, it featured a twin shaft V8 engine, developing up to 310 HP at 6000 *rpm*. The balancing was fairly good, with auxiliary frames to suspend both the engine and the 5-speeds gearbox, connected to the main frame by means of elastic suspensions.

Contemporarily, the Orsi family definitively left the company, yielding the full control to Citroën. Successively, Maserati Merak and Khamsin were launched, gifting continuous satisfaction from the market point of view. Unfortunately, the 1973 oil crisis put in danger the whole automotive sector, with some car makers that suffered more than others: Maserati was awfully among them. However, difficulties were not initially insurmountable,

Figure 1.17: Maserati Bora, the first medium-engined supercar — in the modern meaning of the term — of the Trident.

as Merak SS and the new generation of Quattroporte (Quattroporte II, with Giugiaro-deisgned body) could be anyway launched. On the other hand, the Michelin family — who owned Citroën in those years and who had reached a deal for a fusion with the French car maker Peugeot — soon decided to put the legendary Aemilian brand into liquidation. It was may 23rd, 1973.

Thanks to the pressure exerted by trade unions and industrial associationism, the local and distrectual municipality convinced the central government to avoid the company shut down, providing a period of controlled administration by GEPI[6]. In august, 8th 1975, GEPI signed a deal with the local motorcycles maker Benelli, with which the latter acquired the majority of the shares of the company; they consequently appointed the argentinean Alejandro de Tomaso — that was already patron of the homonymous brand and former Maserati driver — as chief executive officer of the Trident. De Tomaso, among many difficulties, managed to heal the finances of the brand and already in 1976 launched the new model Maserati Kyalami, while in 1979 the third generation of Quattroporte — again designed by Giugiaro — and eventually the 1981 Maserati Biturbo, a two-door sport sedan characterized

[6]it is the acronym of *Società per le Gestioni e Partecipazioni Industriali*, i.e. a national society for the financing and sustainment to companies in difficulty, to discourage unemployment

by affordable prices, but poor reliability due to reduced design times. In this way production registered noticeable increasings, but there were some cons: although this car — which was assembled in Innocenti plants in Milan — gained a remarkable popularity among customers, which also determined the launch of up to thirty different versions of the model, it negatively affected the brand image both in national and international borders due to those poor quality issues.

1.5 FIAT and FCA eras

One of the greatest transformation of the brand occurred in 1993, when **FIAT Auto** acquired the property of the Aemilian brand. Four years later, 51% of the company shares were sold to Ferrari, which was also part of Fiat Auto group, before it gained the full control in 1999. Ferrari began the Trident's renaissance, making Maserati its luxury division: a new factory was built, replacing the existing structures of the '40s, and the brand was renewed, drifting apart bad reputations of poor quality established during '80s decadence period. For this reason, Ferrari is credited to have brought Maserati back to business, after many years on the verge of bankruptcy. Under this new property, a huge renewing opus was initiated, starting from the mounting lines. In 1998 the first Quattroporte Evoluzione was presented, then the 3200 GT (Fig. 1.18), designed by Giugiaro, in Paris Motor Show of that year. In 2001, Ferrari decided to replace the old equipment with new generation, high-tech machines and devices, making the production centre of Ciro Menotti st. one of the most advanced in Europe. Moreover, the directional centre of the brand was definitely established there.

In 2001 the Trident returns to the US market with Maserati Spyder, shown at the Frankfurt Motor Show. This was the first ever Maserati to mount a gearbox derived from the Formula One Ferraris (called "Cambio-Corsa"), and had lines and fashions similar to the 3200 GT ones, but is distinguished by some little details on the rear and by a shorter wheelbase. Moreover, the chassis had been made shorter to adapt a central longitudinal Ferrari 4.2 liters, V8 engine, with propeller shafts to the rear differential, making the car a rear wheel drive vehicle.

2003 is an important year for Maserati: Quattroporte V was unveiled in Frankfurt (Fig. 1.19), being the first Trident car designed by Pininfarina in 50 years, achieving remarkable popularity. In 2004, on the other hand, an ephemeral and fleeting return to races occurred with **Maserati MC12** (Fig. 1.20), after 47 years.

Figure 1.18: The Turinese worldwide famous car designer Giorgetto Giugiaro, with his Maserati 3200 GT (1998).

Figure 1.19: Maserati Quattroporte V, the first Maserati designed by Pininfarina in 50 years.

This car, equipped with a 6.0 liters, V12 engine mounted on a monocoque carbon fiber chassis, participated in FIA GT Championship in the same year, obtaining the first win in the German circuit of Oschersleben, with drivers Mika Salo and Andrea Bertolini. The experience of this car in the racing world was quite outstanding: between 2005 and 2009 it achieved four Pilot

World Champion titles — three with the duo Bertolini-Bartels and one with Thomas Biagi — and two Constructors' titles in 2005 and 2007. Moreover, five consecutive championships were achieved by Vitaphone team.

Figure 1.20: Maserati MC12, the last racing car of the Trident, also designed for ordinary road usage (2004).

In 2005 the company Alfa Romeo-Maserati was created and sited in Via Emilia, Modena, since it was disconnected from Ferrari's ownership and buckled to the former Milanese brand, on the verge of a technical and economical crisis. Although Maserati and Alfa Romeo are today in the same brand group, Alfa Romeo is structured under FCA Italy S.p.A., which itself is structured under FCA, whereas Maserati is structured uniquely under FCA.

In 2013, Maserati began its expansion with the sixth-generation of Maserati Quattroporte, designed to challenge the Mercedes-Benz S-Class. In the following, the new Maserati Ghibli was introduced to compete against Mercedes-Benz E-Class and BMW 5 series.

1.6 Actual production

The Trident's actual production is facing profound transformations and renovation, since old models are mature for a while, and ready to be end-of-series. In these months, a new growth path is catching in Modena plants, where it is deeply rooted, which will provide the launch of new models in next months and years, included a new sportcar forecast for 2021. Up to now, the newest product is the restyled high-end SUV Maserati Levante Trofeo (Fig. 1.21), featuring 3.8 liters, twin-turbo, V8 Ferrari-handcrafted engine, and developing up to 580 HP. Furthermore, it mounts a 8-speeds ZF gearbox, with four-wheel drive Q4 traction and a predictive system for the vehicle stability control, developed in collaboration with Robert Bosch GmbH.

Figure 1.21: Maserati Levante Trofeo, the latest model of the Trident (2016).

For what concerns with the brand's actual production, the complete portfolio is composed by:

- *Maserati Quattroporte VI* (Fig. 1.22, 2013): it is a sporty luxury saloon car, one of the most famous and long-lasting models of the Trident. It is currently available in SQ4, GTS and Diesel trims: the former is provided by the same perfectioned Q4 traction system that is available in the abovementioned Levante Trofeo, and features a 3.0 liters, twin-turbo, V6 engine. The GTS is a slightly higher performance vehicle, featuring the same engine of Levante Trofeo (3.8 liters, twin-turbo, V8 engine), while the Diesel model, which is offered only on selected markets, develops up to 250 HP. The Quattroporte VI, which is produced in the piedmontese Grugliasco factory (Turin), has grown in size — with respect to its ancestors — to challenge roomier luxury saloon cars like Mercedes-Benz S-Class.

Figure 1.22: Maserati Quattroporte VI (2013).

- *Maserati Ghibli IV* (Fig. 1.23, 2013): it is an executive (E Segment) luxury/sporty sedan. This model is named after the hot dry south-westerly wind of the Libyan desert, and its first generation was originally designed by a youngster Giorgetto Giugiaro, which worked at Carrozzeria Ghia at that time. Successively, it was presented in 1966 as a grand tourer car. Today, it is currently available in the same fashions of Maserati Quattroporte, but mid-engined. Just like the relative, it is assembled in Grugliasco factory, and challenges some well established cars like BMW 5 Series and Mercededes-Benz E-Class.

- *Maserati GranTurismo & GranCabrio* (Fig. 1.24, 2007): it is a luxury *gran tourer* (S Segment), 2+2 coupé. It represents the top of the line of the Trident, featuring a 4.7 liters, twin-turbo, V8 engine, developing a rated power of 460 HP. It is installed centrally and longitudinally on the front, with propeller shafts and rear gearbox-differential group, to achieve good weight balancing. The roominess in the engine compartment allows to install double wishbone suspensions, gifting optimized elasto-kinematic performance, coupled to multi-link suspensions on the rear. Although this model has come to end-of-line status, Maserati decided to celebrate the retirement of this iconic and beloved car by realizing a last model called **GranTurismo Zéda** (Fig. 1.25): the name recalls the modenese pronounciation of the letter "Z", probably to symbolize the closure of this car's cycle in the same way the letter Z closes the alphabet. This car was presented to employees in Ciro Menotti

Figure 1.23: Maserati Ghibli IV (2013).

plant in nomvember, 12th, 2019 in a private party, and will be probably sold in auctions in the future. It features a particular varnishing: in the rear of the car a white satined painting leaves gradually space to a metallized dark grey on the sides. The front of the car, on the other hand, is painted with a shiny and translucent cornflower blue, with a pearly effect. The transition of the three paintings, although questionable from the style point of view, symbolizes the passages of the whole handcrafted production.

- *Maserati Levante* (2016): as said for the *Trofeo* version, it is a mid-sized luxury crossover, based on the Kubang concept car that debuted in 2011 Frankfurt Motor Show, and built by Maserati in Mirafiori factory (Turin, Italy). This car initially featured Ferrari's handcrafted engine, in two versions: 3.0 liters, V6 engine and 3.8 liters, V8 engines, both twin-turbo-based. Additionally, a Diesel fashion is also offered, reserved to right-hand-drive markets only. Curiously, it achieves the same drag coefficient of the relative Quattroporte VI. Although it surely benefits from the huge success of SUVs in late '10s, it can be considered the largest-selling model of the actual production.

After the creation of Alfa Romeo-Maserati company, the production and engineering of the Alfa Romeo vehicles have been permanently transferred in Modena, when they are produced today. Even though there are rumours

Figure 1.24: Maserati GranTurismo (2013).

that the production of Alfa Romeo vehicles will be rearranged elsewhere, as a consequence of important corporate reorganization, there is one last model that is still produced in that plant, the Alfa Romeo 4C (Fig. 1.26). Despite this model is rapidly heading towards end-of series, it will be probably used as base for the new sporty model of Maserati, which will be presented in the event "MMXX" in may 2020, at least for what concerns with the chassis.

Figure 1.25: Footage of me and two of my colleagues next to Maserati GranTurismo Zéda, in occasion of the november 12th, 2019 end-of-series celebrations in Ciro Menotti plant, Modena.

Figure 1.26: Alfa Romeo 4C, 2019 preparation.

Chapter 2

The engine calibration cycle and data acquisition

The concept of *vehicle* today has considerably mutated from the sense it had several decades ago, especially with the advent of electronics and computer science, which have speedily raised the bar of objectives and performances related to the vehicle status monitoring. At that time — when the automotive industry was entirely reliant on mechanics — vehicles' behavior was characterized by a certain degree of **uncertainty**, related to the functionality of various subsystems that were not thoroughly understood and subsequently were not satisfactorily engineerable.

It can be said that the degree with which a subsystem's response manages to satisfy the driver's requests, in fact, is a function of the number of variables and parameters that it is possible to monitor and regulate: without the aid of electronics, the only sensors that could be exploited to regulate a subsystem's action were purely mechanical ones. One example could be the surface carburetor, which was needed as mixture preparation device and was fairly common before the introduction of UEGO sensors: the air-fuel mixing was granted by air suction through a stationary fuel layer, thanks to the pressure differential between the intake manifold and the external environment. To nowadays observer, it is immediately clear that the accuracy of such mixture preparation was quite poor — since its title could be only very roughly controlled — with negative cons from the emissions, fuel consumption and performance points of view. To give a second example related to the actuation side, the accelerator mechanism was realized with a mechanical connection (usually a *bowden cable*) between the accelerator pedal and the throttle valve, which made the system very reliable but not adjustable to driver's needs, with frequent drawbacks on drivability.

Both these two examples picture an outdated state of the art, which was more concerned about making the whole vehicle system *working*, rather than optimizing its behavior. Additionally, a sophomoric comprehension of several dynamic and thermodynamic phenomena, which could have been discovered only with an aware usage of the compiler, made the situation even worse. Furthermore, the abovementioned concept of *satisfactorily engineerable* is questionable: at the time I am referring to, a certain degree of unreliability was basically accepted, because it was not imaginable that a deep and profound vehicle control — just like the one we have achieved nowadays — was even feasible. One of the aspects that probably made the racing world so interesting and emotional was the uncertainty about the vehicles' performance and reliability over the whole race, which was furtherly exaggerated by the limit-condition stresses to which those vehicles were actually pushed. This, together with the high speeds, rebounded dramatically on pilots and professionals' safety, who have gradually assumed — in the epic of posterior reimagining — the role of modern heroes, to whom different contemporary and later cultural instances inspired.

Nowadays, major changes introduced by electronics and computer science on society and economy have finally made the idea of **optimization** accepted almost everywhere. As it often happens, everything that for a certain period grants a competitive advantage in a market — as it was the case of the idea of continuous improvement, starting from the '90s — soon becomes a need. At the same time, the reached feasibility of a deeper and more conscious comprehension of the vehicle subsystems, which led the exploration of new design possibilities, soon became a necessity, with every car maker slowly forced to orient their businesses towards this direction. In this context of enhanced technology, internal combustion engines — which are the main object and protagonist of my stage experience — needed an improved and a more proper control, to return — with as fine as possible approximation — the dynamic behavior requested by the driver, so that a satisfactorily drivability is granted and the onset of unsafe conditions for the occupants is avoided.

The task of monitoring that the engine actually works as expected is a duty that is today accomplished by the electronic control unit (Fig. 2.1), also known as ECU, or ECM – electronic control module). As it can be noted from the figure, its external metallic structure is temperature and vibrations insensitive to cope with the hard environment developed inside the engine compartment during vehicle runs. Fins are realized to have better heat rejection in case of overheating events, but also as vibrational dampers. Inside this protective structure, there is a sophisticated circuitry including a microprocessor — which can elaborate the inputs from the engine sensors in real-time — and a microcontroller chip, onto which the engine control software is loaded. ECUs are typically provided with a flash memory, so that the microprocessor's CPU can be re-programmed by simply uploading the updated code: this operation is usually called ECU *flashing*, because it is named after the kind of memory it features. Eventually, the electrical interface with the external environment is realized in form of specific connectors that can be appreciated in the figure below.

Actually, there are many ECUs in a vehicle, each one controlling a different subsystem and/or a different function: sometimes, for packaging reasons but especially due to huger computational effort required, two ECUs (one per cylinder bank) may be required, for example, to control a V-engine. The ECU control over vehicle functions occurs cyclically according to the following logic scheme:

- A very large number of signals — coming from sensing devices, strewn about the vehicle in strategic places — is continuously monitored by the engine ECU(s) as a result of the continuous exchange of data occurring through the established network. The higher the sampling rate[1] with which this information is measured and stored, the higher the computational effort required to the ECU to elaborate the signals, the finer the approximation it gets of the real phenomena occurring.

- Incoming signals are then *processed*. Usually, this operation consists in calculating input values and/or making comparisons between the measured value and the setpoints. The latter are defined at the level of engine calibration, which basically consists in giving the ECU the instructions on how and when to behave about the actuators' actioning, to implement the required control strategy. This procedure is also known as *engine mapping*.

- At the end of the ECU's elaboration phase, actuators are commanded following the specific instructions contained in the engine calibration. The whole process occurs in loop, according to a *negative feedback principle*, in which the system response is continuously monitored in terms of the difference with the expected values, also considering the effects

[1]The sampling rate can be defined as the *speed* with which an analog signal (i.e. a continuous-time signal) is measured in time, in order to get its corresponding reduced digital signal (i.e. a discrete-time signal). Thus, a constant sampling rate corresponds to measure the analog signal every constant and specified amounts of time. It is clear that this process — which is intrinsic to the analog-to-digital conversion — brings along information losses about what happens in-between two consecutive samples: to reduce as much as possible this effect, it is necessary to increase the sampling rate in order to increase the *resolution* of the digital representation.

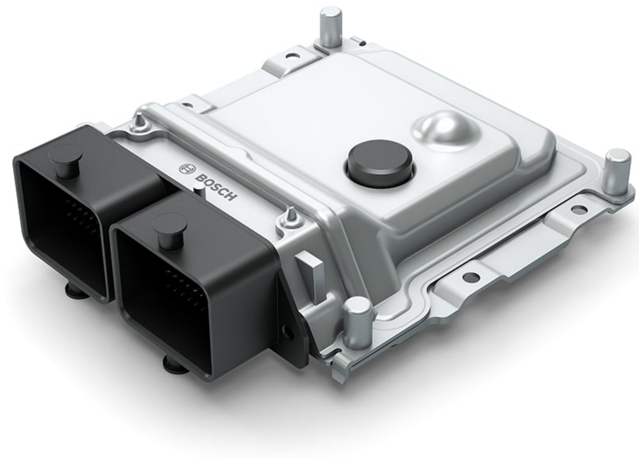


Figure 2.1: Example of Electronic Control Unit (ECU) made by Bosch.

of the exogenous inputs or disturbances on it. The communication between sensors, ECUs and actuators occurs on settled infrastructures called *networks*, each with a specific and peculiar communication protocol granting not only the correct reception of the messages, but also the right priority handling.

During the data processing phase, the engine control software — loaded on the ECU — grants the proper control strategy: it features a large number of *functions*, which consist in a series of specific *algorithms* allowing to execute precise *operations*. Within each algorithm there are some tables — also called look-up tables, or simply maps — which consent to return a calculated output in the face of a certain number of incoming signals. Every map contains a series of reference values — or thresholds — that are drafted at the level of the engine calibration, which is vice versa devoted to the definition of those thresholds. To make the ECU control precise and efficient, every map needs to be continuously updated and optimized: this iterative and continuous process lasts until the quality of the engine control is compliant with the regulations and company standards. To decide if the engine control has actually reached the quality targets, a periodic **validation activity** — providing a series of dedicated test performed on real vehicles on the road — is needed. The importance of this validation activity is that the engine control must be not only compliant, but also resilient to disturbances and ensure a satisfactory robustness.

In Maserati, this activity is performed by triggering precise troubleshooting inquiries on a series of test vehicles — loaded with the engine control softwares that are updated from time to time — with the aim of reproducing the behavior that it is wanted to be analyzed. During these tests, the evolution in time of the variables of interest is measured, and the signals so obtained are recorded. Then, since the volume of data to be analyzed is very huge, statistics are performed and — on the base of them — decisions about the compliancy of the new softwares are drafted.

2.1 The validation fleet

The *validation fleet* is a collection of probationary vehicles that are used to test and validate the calibration progresses made with the updated engine control softwares. These vehicles are specifically built by car makers to perform mileage accumulation programs, in order to complete the ending stages of calibration process leading to a marketable vehicle. Since they are nothing but test vehicles, they cannot be sold and must be scrapped at the end of their accumulation period.

Every car of the validation fleet is equipped with on-board diagnosis instrumentation, which furnishes an access gateway to every ECU present in the vehicle. Connecting a portable computer to those gateways, is possible to establish a direct communication between a specific ECU and the laptop, acquiring and storing data from any sensor communicating with that ECU. For our purpose, since I have worked in the Powertrain Department, the only data of interest are retrieved from the engine and/or the driveline. From time to time, when modifications in hardware and software of fleet vehicles occurs, it becomes necessary that the Department could acquire new vehicles, with updated specifications.

Maserati's fleet contains vehicles at various development stages, each one successive to the prototype gimmick. The development status of a fleet vehicle is defined by the evaluation of a quantitative index, called the *significance*, measuring how much a fleet vehicle differs — from the hardware point of view — from the one that will be marketed. According to their significance, fleet vehicles are named by assignment to various classes of equivalence, which are synthetically identified with a serial alphanumeric code (e.g. VP874). For instance, the largest categories are:

- VP: stands for "*Verifica Processo*" (trans. process verification);
- PS: stands for "*Pre-serie*" (trans. pre-series);
- IP: stands for "*Inizio Produzione*" (trans. initiated production);

In each project's phase, the evaluation of vehicle's signifcance has to be made according to a standardized set of rules: in this way, it is possible to know and distinguish every fleet vehicle and knowing exactly its development status. Usually, from the point of view of a global vehicle maturity target[25], the percentage quantifying the resemblance between a fleet vehicle and the marketed one are 55% for VP, 80% for PS and 100% for IP, respectively. Actually, in everyday company life, there are plenty of in-between situations, in which the exact definition of a vehicle's significance is labile.

Maserati's validation fleet consists of a limited number of gasoline vehicles, one for every Maserati and Alfa Romeo model and motorization:

- Alfa Romeo (AR):
  - Giulia Quadrifoglio, 2.9 liters, V6 engine, 510 HP;
  - Stelvio Quadrifoglio, 2.9 liters, V6 engine, 510 HP;
  - Giulia, 2.0 liters, 4 cylinders, in-line engine, 200-280 HP;
  - Stelvio, 2.0 liters, 4 cylinders, in-line engine, 200-280 HP;
- Maserati (M):
  - Quattroporte, 3.0 liters, V6 engine, 430 HP;
  - Ghibli, 3.0 liters, V6 engine, 350-430 HP;
  - Levante, 3.0 liters, V6 engine, 350-430 HP;
  - Quattroporte, 3.8 liters, V8 engine, 530 HP;
  - Levante, 3.8 liters, V8 engine, 520 HP;

2.2 Principles of communication networks and intra-vehicular communication

As explained in previous Sections of this Chapter, today's motor vehicles need to be equipped with an electric and electronic infrastructure to allow the establishment of a communication network. In the automotive sector, those networks are usually addressed as *on board intra-vehicular networks*, and are required to exchange data in real time, to optimize the vehicle performance and to regulate every subsystem's action. In networks of this kind, data transmission occurs, as arguably worldwide-accepted rule, in form of *packets*[2] (also called frames, windows, messages or telegrams) whose format is standardized and known by every network user. The existence of regulated formats allows network agents to discriminate between correctly-delivered and empty, broken or non-informative packets, plus other vital applications such as internal clock synchronization.

However, the existence of a transmission medium, which is also named (in telecommunication jargon) *channel* — constituting the physical span onto which the information actually travels — is not enough to allow a proper communication: a set of *rules* — not dissimilar to the language syntax rules we everyday use in oral speeches — has to be defined and known by every communication agent. This set of rules is commonly called *protocol*, and its knowledge is required to allow the correct transmission, reception and understanding of messages. For instance, if there was not any protocol specifying the timing with which every station of the network is allowed to transmit, collisions between packets could occur: if this happens the information is simply lost, and retransmissions are needed. Since this event could impair the efficiency of the communication (that is related to the rate at which packets are successfully delivered) and subsequently the network functionality — especially if a high degree of real-timeness is required — the definition of a resilient and efficient protocol is a must-have. Moreover, when data rates[3] increase, the probability to have collisions increases too: as a consequence, for automotive purposes — where the amount of data to be processed is very huge — it is crucial to exploit efficient protocols limiting as much as possible the number of collisions (corresponding to communication errors) and the time employed to successfully end a communication.

The reason for this preamble is to introduce the basic concepts behind communication in vehicular networks, which are the base to understand the architecture of automotive data logger and acquisition systems. Maserati's fleet vehicles, as it will be seen in next Sections, are all equipped with two different acquisition systems: **Control-Tec CT-VDR 1000** (Fig. 2.4) and

[2]Packets are nothing but units containing portions of the information to be transmitted. They not only contain the net information to be exploited by the receivers, but also additional details — which must be readble and understandable by every other network agent — in which they are specified, for example, the *identity* of who is transmitting, the one of the required receiver, the *priority* of the message, and so on. A frequent metaphor used for didactic purposes juxtaposes packets to *envelopes* containing written messages: to correctly deliver a message to a specific person — in fact — it is necessary not only that the envelope actually contains a message readable and understandable by the receiver, but also that the envelope reports some additional details (which similarly have to be completely transparent to the Postal Service) like the identity of sender, the one of the addressee, the postage stamp, and so on.

[3]The amount of data transmitted per unit time, ed.

ETAS ES720 (Fig. 2.8). They exploit two different communication protocols – CAN and XCP, respectively – that need to be discussed before proceeding with the explanation of how data acquisition practically occurs.

2.2.1 CAN Bus

A Controller Area Network (CAN) is a robust vehicle bus standard designed to allow a reliable and fast communication between ECUs, sensors and actuators, in applications without a host computer[9]. It was originally designed in 1986 by Bosch — which was probably thinking about a smarter way to handle the increasing impact of electronics in cars world — with the aim of multiplexing[4] electrical wiring within automobiles, thus avoiding the installation of complex dedicated wiring in between. Furthermore, the standard allows a direct communication between ECUs, therefore each station can use the information stored in the others, eliminating the need to install the same sensors in multiple places.

A traditional CAN network is composed by a *linear bus*, realized with a couple of twisted cables, and by a theoretically infinite number of *nodes*, also called stations, whose role is played by the ECUs. All nodes have in principle the same right to transmit, since there is no central host, and information is sent exploiting a packet-based protocol. CAN bus most remarkable features are:

- *Priority-controlled message transmission with non-destructive arbitration.* In real-time processing, the urgency of messages to be exchanged over the network can differ greatly: a *low memory dimension* (i.e. rapidly changing variable, e.g. engine speed) has to be transmitted more frequently[5] and therefore with fewer delays than other dimensions (e.g. engine temperature, which has larger memory) changing relatively slowly. In every packet-based communication protocol, the priority at which a message is transmitted — compared with another less urgent message — is specified by the frame identifier (Fig. 2.2) of the packet, which is a predefined and recognizable sequence of bits (usually 29). In this way, the priorities are laid down during system design in the form of corresponding binary values and cannot be changed dynamically. The identifier with the lowest binary number has the highest

[4]In telecommunications and computer networks, multiplexing (sometimes contracted to *muxing*) is a method by which multiple analog or digital signals are combined into one signal over a shared medium, with the aim of sharing a scarce resource.

[5]The reason is that a rapidly-changing variable needs to be sampled more frequently than a slowly-changing one, to sufficiently keep bounded the intrinsic loss of information.

priority: this is a generally adopted *error-resilient* criterion, since in case of malfunction of a specific subsystem, the lack of transmission from that node (i.e. a sequence of 0s) is considered priority.

On the other hand, bus access conflicts (i.e. possibile packets collisions) are solved with the implementation of a carrier-sense multiple access (CSMA) protocol with collision detection (CD) paradigm, based on *arbitration*. To be synthetic, and neglecting out-of-interest details, each network station starts observing the frame identifiers of the respective transmitted and received packet, bit for bit (this is why the procedure is also addressed as *bitwise arbitration*). In accordance with the general criterion, by which the dominant state (logical 0) overwrites the recessive state (logical 1), the competition for bus allocation (i.e. for the allowance to transmit) is lost by all those stations with recessive transmission and dominant observation. All losing stations automatically become *receivers* of the message with the highest priority and do not reattempt transmission until the bus is available again. This paradigm allows to minimize the collisions, hence increasing the amount of information delivered in the unit time.

- *Low costs*. The reduced number of dedicated wired mediums decreases weight, wiring harshness, power consumption and costs, increasing at the same time the network reliability. Since control functions are becoming more and more complex, as the amount of devices involved in a single control strategy, this aspect is crucial because the number of connections cannot be increased much further.

- *Medium performance data transfer*, between 200 *kbit/s* — characteristic of the low speed CAN (CAN-B, which involves lower hardware cost) — and 1 *Mbit/s*, typical of high speed CAN (CAN-C).

- *High reliability* due to minimized losses (short spans and reduced lossy collisions) and due to recognition and signaling of sporadic faults and permanent faults. The mechanism is furtherly perfectioned by the implementation of the *Acks* (which stands for "Acknowledged"), which are dummy bits used by any station to verify the bus availability before starting a transmission.

- *Multi-master principle*: no central host is needed, and the hierarchy is based on the messages timing and priority. This aspect has been already described in the first point of this list.

- *Standardization* in accordance with ISO 11898.

Figure 2.2: The 8 fields of a CAN bus frame (also called frame format). The frame identifier is here called CAN-ID, and is specified by a set of 29 bits.

2.2.2 XCP protocol and ETK, XETK interfaces

In parallel with CAN bus, a complementary interface is used — in Maserati fleet vehicles — to increase the network capability and measurements storage. This interface is called ETK, and is an Ethernet-based interface developed by the German software developer company ETAS. The term "Ethernet" refers to a predefined standard (IEEE 802), identifying the addressing, the format of the messages and medium access control. The communication protocol it exploits is XCP, also known as *Universal Measurement and Calibration Protocol*: according to the definition given by ASAM[6], the primary purpose of XCP is to adjust internal parameters and acquire the current values of internal variables of an ECU [12]. The first letter "X" in XCP expresses the fact that the protocol is designed for a variety of bus systems. In other words, the protocol enables read and write access to variables and memory contents of microcontroller systems at runtime, and can be considered an evolution of the traditional CAN Calibration Protocol (CCP), which allowed lower data rates and read-only access. Entire datasets can be acquired or stimulated synchronous to events triggered by timers or operating conditions. In addition, XCP also supports programming of flash memory, characteristic of development ECUs as explained at the beginning of this Chapter.

As said, the main advantage of this enhanced interface — with respect to traditional CAN bus — is the remarkably higher data rate: in a continuously improving environment, as the automotive sector is, the development of acquisition systems that are able to increase the amount of successfully recorded data is a breakthrough attribute. In this way, finer and more significant analyses on the vehicle's performances can be led, with the aim of enhancing the indicators that does not satisfy the corporate quality requirements. This fea-

[6]The acronym ASAM stands for Association for Standardization of Automation and Measuring systems

ture can be the difference between a successful and well-positioned car maker and an unsuccessful one.

ETK interface is specifically designed to work with a lot of ETK hardware — i.e. a series or small, robust and high-performance components to be mounted directly inside development ECUs — equipping validation fleet vehicles. The presence of this hardware with the relative interface gives software and calibration engineers direct and real-time access to control measurement variables and parameters during ECU runtime. The objective of this interface is to reduce engineering effort and computational overheads for tasks such as measurements, calibrations and flashing of ECU software. Furthermore, their dedicated power supplies enable the preparation and initiation of cold-start testing independently of the ECU: this is a particularly interesting attribute, since sometimes it is required to study the response of a certain system before cranking occurs. Due to their extremely compact design, this hardware can be accommodated inside the housings of development and production ECUs, as well: they are impervious to temperature extremes and vibrations insensitive, which is a breakthrough attribute considering the engine ECUs locations in the vehicle. The most remarkable features of ETK interfaces are:

- Very high accuracy of the measures and very low losses due to the high sampling rate (up to 10 μs raster);
- Higher maximum data troughputs[7] which can reach 30 $Mbyte/s$, remarkably higher than the one guaranteed by CAN bus;
- Time and/or crank angle synchronization between a larger number of channels in the same measure;
- Possibility to make cold-start recordings and to keep the measure running in spite of ECU restarts.

On the other hand, XETK (Fig. 2.3) represents an enhanced ETK interface, to cope with remarkably higher data troughputs that will be faced with increasing electrification. It enables cost-efficient ECU software developing during all major development phases. Similarly to their elder relative, it offers:

- Embedded ECU hardware with sufficient performance for all jobs to be done during calibration work;
- System cost optimization, without hardware arbitration for debuggers;

[7] i.e. the amount of data that are simultaneously travelling on the network

- Direct access to ECU with laptop, via Standard Ethernet and open protocol XCP or Ethernet.

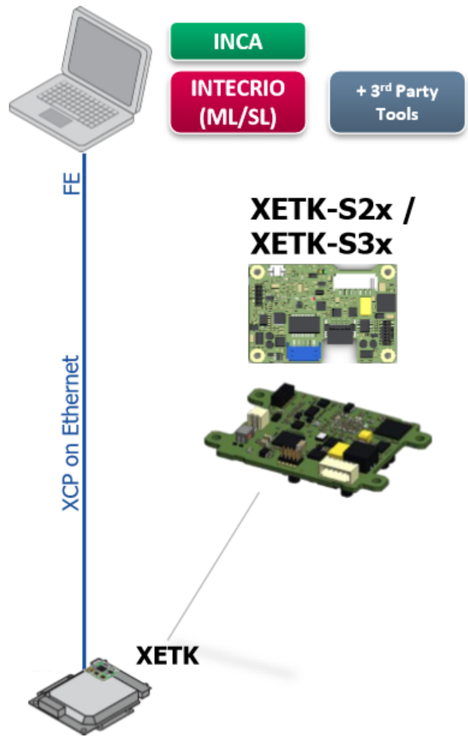


Figure 2.3: The enhanced XETK configuration.

2.3 On board instrumentation and data acquisition softwares

In Sections 2.2.1, 2.2.2 the principles of the communication protocols used by the two acquisition systems used by Maserati have been illustrated. On one hand, Control-Tec CT-VDR 1000 runs on a CAN bus communication infrastructure and is able to communicate with both the CAN vehicular network — allowing to retrieve information from the chassis compartment — and the private CAN network, which is based on CCP protocol and allows to acquire just the engine parameters stored and processed by the engine ECU. Due to the modest performances supplied by the traditional CAN infrastructure, this system is able to handle just a reduced number of channels (few hundreds), but has the advantage to convey in real time all the measured data to a remote server, in which they can be stored and processed. In this way, all the recorded data are worldwide accessible on Control-Tec Qualifier®[8] portal, in which they can be protagonist of a very detailed analysis, albeit a first level one.

On the other hand, ETAS instrumentation needs the existence of a developing ECU[9] to work properly, as said. The connection is realized with an Ethernet cable and/or with the proprietary ES592 cable via XCP communication protocol. The interface between the recording devices and the communication network is a specific ETAS property, which is sold to every development ECU suppliers (in case of Maserati, the two suppliers are Bosch and Magneti Marelli) and allows to communicate with a larger bandwidth, corresponding to higher data rates. Thanks to this characteristic, the amount of data acquired through ETK is much larger (thousands of channels) with respect to the ones acquired with Control-Tec. For this reason, the usage of this instrumentation is particularly useful in engine calibration phases.

Differently from Control-Tec, ETAS stores the measured data on internal memory installed in the data logger, or in specific USB pen drives which can be connected through the dedicated ports. These data are daily downloaded in form of *.dat*/*.mf4* measures, and can be remarkably long (up to 1-2 Gb, even more in case of acquisitions coming from US). Then, these downloaded data are stored in a corporate server, containing several folders that are specific for any vehicle of the fleet, and shared between the whole powertrain department, in which every engineer can visualize and analyze the behavior of all the experimented channels. Those subdivisions are necessary because the channel names vary not only with the development ECU provider, but also between vehicles at different development stages (see Section 2.1). In Maserati's fleet, as a general criterion, V-engines are controlled by Bosch-produced developing ECUs, while GME (which stands for Gasoline Medium Engine, a 4-cylinders, in-line engine) are controlled by MM ECUs.

After having discussed general concepts about these two acquisition systems, next Sections will deal with more technical specifications, arrangements and softwares used to handle the recorded data.

[8]The applicative has recently changed its name in Aptiv Qualifier®for commercial reasons.

[9]The term refers to engine control units that are not concepted for being mounted in production vehicles, but that are specifically thought to be continuously and rapidly flashed, to facilitate and speed-up the calibration loop process.

2.3.1 Delphi Automotive Control-Tec

Delphi Automotive Control-Tec CT-VDR 1000 (Fig. 2.4) is basically a recorder, connected to the vehicle OBD port[10] by means of a Control-Tec C-CAN-based cable, acquiring data from the CAN Bus vehicular network and storing the measured data upon a remote server. It can also retrieve calibrated powertrain channels directly from the Private CAN bus — through CCP communication protocol — to avoid wasting bandwidth in measuring non-necessary variables. As explained, Private CAN bus is a secured and dedicated network addressing the engine control unit: it is the only interface that provides the possibility to visualize and change engine calibrated parameters, and it needs to be secured for many reasons, concerning safety, law, corporate security, and so on.

Figure 2.4: Delphi Automotive Control-Tec CT-1000 data logger.

As said, this tool allows to make a first level analysis of the acquired data, pursuant of the low data-rates and number of channels offered by CAN bus interface. Its main features are:

- Fleet measurement with geolocation functionality;
- Real-time notifications and reports on possible issues;
- Automatized analysis to evaluate performance indicators of a specific system;

[10]The OBD port is a standardized access gateway to the vehicle embedded self-diagnostic system. It will be discussed in Section 3.1.

- Comprehensive data storage, in which for any time instant it is possible to retrieve collateral information as events, trips, maps, statistics, calculated channels, reports, and so on;

- Worldwide accessibility, since data are stored on servers.

Control-Tec leverages real-time data and the cloud to identify and solve issues. Moreover, the tool CT-Edge computing allows the data to be analyzed locally, onboard the vehicle, rather than in the cloud. It reduces the data volume, the distance the data must travel, reduces transmission costs, shrinks latency[11], improves quality and improves cyber security by discovering viruses, compromised data and active hackers early in the process. Meanwhile, Delphi Automotive is developing software, data loggers and a special chip that intelligently sifts through data and picks a sample size big enough to act on but small enough to manage.

2.3.2 Control-Tec Qualifier®

Control-Tec Qualifier®platform provides a comprehensive data management and analytics solution for vehicles testing. It can provide many interesting features:

- *Asset management and geolocation features.* The testing activity efficiency is maximized by knowing the status of the validation fleet at all times: Qualifier®can track and record vehicle location and VDR (which stands for *Vehicle Data Recorder*, a sort of archive of what is going on inside and outside the vehicle, in terms of speed, accelerations, maneuvers, trips, and so on) status, providing live data subdivided by trip, to ensure test compliance and understand customer usage.

- *Issue management* with real time notifications and report subscriptions. Each registered issue can be thoroughly scoped thanks to deep data consolidation, knowing what system is affected, where and when the issue has occurred. Furthermore, the intuitive and easy-to-use interface provide quick access to determine *whys* and *hows* of any issue. This preliminary assignment is led by analyzing freezed snippets of the whole measure, to adequately confine the relevant event. Furthermore, user comments can be inserted to promote the comprehensibility and

[11]Latency is a sort of *delay* introduced in the transmission of information. More precisely, it identifies the time that a station has to wait before it is allowed to transmit something. This value depends on the network congestion, but also on the protocol structure.

accessibility from multiple users. Eventually, there is the possibility to send e-mails and SMS notifications to be real-time updated on the status of the diagnoses.

- *Root cause analysis for any issue event.* It is possible to retrieve the whole CAN bus network log, which can be captured synchronously to the time of the issue, to identify its root cause.

- *Intuitive reporting.* Qualifier®contains an embedded reporting platform which allows to draft quick and easy analysis to determine the performance of any subsystem within the validation fleet. The features of this reporting platform include:

  - Generation of scorecards including several useful metrics, such as channels average, standard deviation, counters based on customized filters, and Cpk[12]. The latter is a quantitative product quality indicator to measure not only how close the performance of a certain set of measured variables are to the predetermined targets, but also the degree of consistency[14] with time;
  - Reports on the vehicle status, vehicle hardware and software history, channel performance metrics, including Pareto charts — either of the overall system, or by single subsystems;
  - Reports can be conveyed telematically as e-mail content, at specified time intervals, and shared as PDF files;
  - Easy plotting with drag and drop charting and intuitive user interface;
  - Integration of all data sources in one single file;
  - Templates for automatic plotting are provided.

- *Hardware/software part number tracking* to have traceability of each trip, ensuring the proper quality and compliance of mileage accumulation cycles.

- *Test thresholds* are available to profitably compare different vehicles, calibrations, and much more.

2.3.3 ETAS ES592, ES720

The ETAS instrumentation used in Maserati's validation fleet is composed by two different tools:

[12]Process Capability Index, ed.

- ES592 (Fig. 2.5): it is a physical interface module that provides one ETK connection plus two CAN interfaces. This component is provided with an upstream Ethernet interface that guarantees data exchange with the host PC with INCA software downloaded on it and allows connections with ECUs having ETK/XETK interface (Fig. 2.7). The Ethernet switch guarantees the time-synchronous sampling of all measurement channels, either in test rigs or directly on board. Other characteristics of the module are:
  - Each channel is galvanically isolated from each other, from the device ground and from the supply voltage, to ensure electromagnetic compatibility, avoiding that the measures are corrupted;
  - High mechanical stability and durability;
  - Not sensitive to environmental conditions, for instance hostile temperatures or electromagnetic interferences (EMC compatible).

Figure 2.5: ETAS ES592 module, front view.

With respect to Figg. 2.5, 2.6, it is possible to see all the interfaces provided by the module:

- *HOST port*: it allows the ETAS software tools to access the connected modules;
- *ETK port*: proprietary interface of ETAS with direct connection with ECU, allowing higher data rates;

– *Ethernet ports (ETH1, ETH2)*: they allow to connect the device with other measurement/calibration devices, like ES720. ECUs having a proper XETK interface or their own Ethernet interface can be connected directly with a ES592 module and communicate with the calibration software (INCA) via XCP on Ethernet cable (see Fig. 2.9).

– *CAN/LIN ports*: classical CAN interface ports. CAN1 and CAN2 are completely independent channels, with separated connections and controllers. For this reason, they can be configured and used simultaneously and independently one from the other.

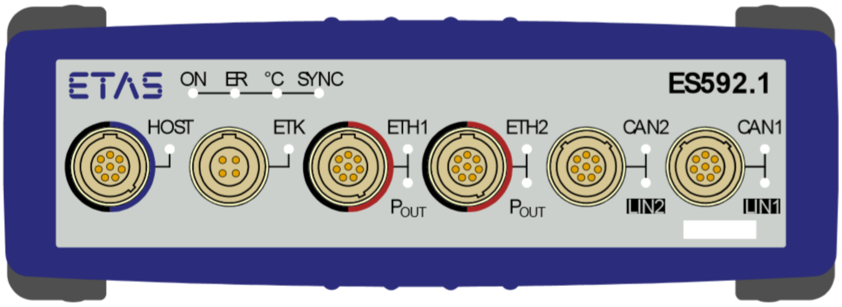


Figure 2.6: Detail of ETAS ES592 module, in which the layout of every port is highlighted.

Figure 2.7: Possible connections layout between the ES592 module, host PC and ECUs.

- ES720 (Fig. 2.8): Complying with the same role of INCA PC or laptop, this 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. This tool has also mounted a USB port: in fact, data can be stored either in ES720 internal ash memory, or in an external USB device, such like classic pen drive. Files extracted from ES720 are the measurement files that are then studied. Obviously, the Drive Recorder ES720 is fully compatible with INCA, the standard tool for measurement, calibration and diagnostics of ECUs. In everyday practice, it is used in combination with ES592 module and a INCA-provided laptop (Fig. 2.9), to continuously adjust the communication exchange between the recorder and the ECU, on the base of the instructions given by the calibrators through the experiments loaded on INCA. For example, to save on time and on computational effort, it may be required to record only a few of the channels provided by the experiment, and this can be made only with the network layout displayed in Fig. 2.9.

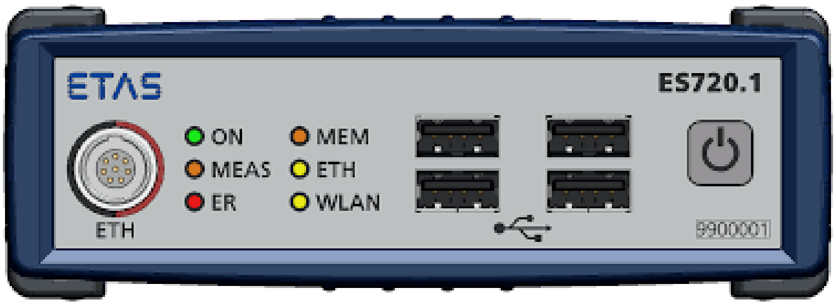


Figure 2.8: ETAS ES720 module, front view.

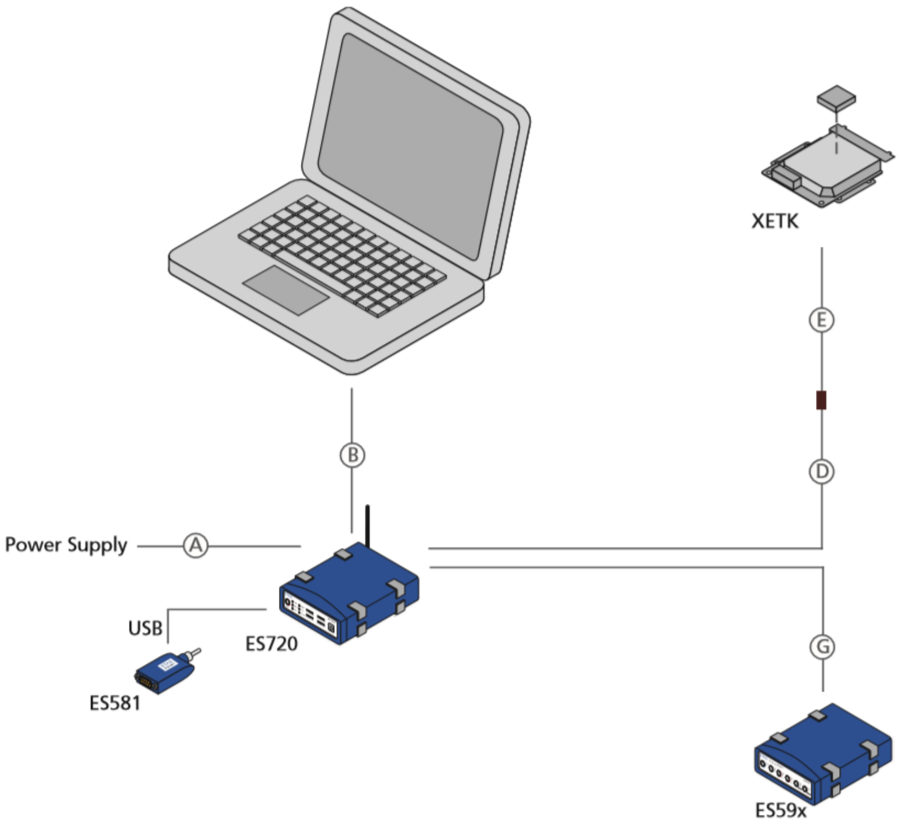


Figure 2.9: Possible connections layout between a laptop, a USB drive, the ES592, ES720 modules and the developing ECU provided with XETK hardware.

2.3.4 ETAS Integrated Calibration and Application Tool

ETAS Integrated Calibration and Apllication Tool (synthetically named INCA) is a measurement, calibration and diagnostic software published by ETAS. With its large installation base in the auto industry, it can be deployed during all phases of the development of the ECUs, as well as their software programs for measuring, calibration, diagnostics and programming. Using simpler words, INCA can be considered the main interface between the development ECU(s), the recording devices and the calibrators. Vice versa those ones, to flash or to retrieve information from the ECU(s), need to prepare the adequate communication environment by means of INCA software. This communication environment is commonly named *experiment*, and is basically a huge set of variables that calibrators prepare in order to real-time monitoring and recording those data from the ECU. Unfortunately, due to the ECUs finite memory, it is not possible to include every variable in the experiments, hence they should be carefully designed.

Calibrating an ECU software with the aid of INCA enables engineers to adapt the behavior of control and diagnostic functions to a variety of vehicle models and/or model variants, without requiring the modification of calculation routines. As part of this process, characteristic values of function algorithms are entered while simultaneously acquiring signals from ECUs, vehicle data buses and measuring instruments. During the calibration process in INCA, the ECU signals are visualized, which means that any change occurring inside the ECU can be followed up by a detailed examination and analysis of system behavior. This type of characteristics calibration may take place on board the vehicle, in the lab, on test benches or in combination with simulation environments, such as Simulink.

2.3.5 ETAS Measure Data Analyzer

ETAS Measure Data Analyzer (synthetically called MDA) tool allows to visualize, further process, analyze, and document the measured data[15]. The process undergoing from data recordings and data analysis develops as follows: once the data from the engine ECU have been acquired — for example with a INCA-provided laptop connected to the control unit through a ES592 module, as Fig. 2.7 suggested — they are stored in form of single time-varying channels. The number of channels to be simultaneously saved depends firstly on the number of the inquiried variables — which are optioned, as said, during the experiment drafting — and secondly on the network bandwidth, which is always far broader than needed, especially in ETK case. Each channel is saved in a unique $.dat/.mf4$ file, which is commonly addressed as *measurement file*, whose size is strongly dependent on the time window it lasts and secondly on the number of channels stored in it.

These files are processed by default by MDA, which is a very simple and intuitive interface that can display an arbitrary number of channels on the base of the time (Fig. 2.10). They can be displayed in their whole time frame, or zoomed in specific points of interest to study a specific behavior (Fig. 2.11). Furthermore, adjustments of the y-axis scale of each signal are always allowed, as well as enlargements of the signals' splines. It is also possible to display *cursors* (which are represented in red in the figures below) which help both in understanding the values that each channel assumes in a specific time instant, and in evaluating a custom time window. Obviously, for the digital nature of the data and of the record, each signal is discrete, having specific raster (or sampling rate): it is not so rare to investigate the signals behavior up to this zooming, since sometimes the increments (or decrements) experienced between a sample and the following (or the preceding), may be the difference between a pathogenic or pleasant behavior.

With reference to Fig. 2.10, three channels have been simultaneously reported during the whole measure span. The scale of the channels, as well as the name of the measure and the name of the vehicle have been hidden due to confidentiality reasons. The displayed channels represent the engine external temperature (violet), the engine rpm (green) and the *start&stop* status (light blue); each of them, for the reasons explained above, is a discrete signal having more or less the same raster, but their memory is noticeably different: for instance, the engine temperature changes slower with respect to engine *rpms*. If there was the need to identfy a specific behavior from this measurement file, it could be necessary to zoom in up to the channels' rasters (Fig. 2.11): here, it is possible to notice the relation between the start&stop status and the engine rpm, especially when this goes to zero. The red cursors, with respect to the start&stop channel, are pointed to "in operation" and "start" statuses, respectively. The engine temperature has weak correlation with these two parameters, and anyway changes far slower.

The number of channels to be contemporarily displayed depends on the data analyst needs: in some cases the precise framing of a specific behavior needs just a bounded number of channels, while in others — for instance if the engineering or physical nature of the behavior is not completely known, or if this is affected by lots of variables — a huger number of channels may be required. The existence of a tool of this kind, allowing to visualize the behavior of every single variable of the experiment in every time frame, is a breakthrough advantage for engineers, since it helps in understanding the problem up to its roots. This is particularly important, especially when the results of the current calibration have to be validated: in fact, data analysis phase is probably the most important feedback to be given to calibrators, for at least two reasons: to understand if and at what extent the purpose of the calibration has given the desired effects, and to orient the direction of the calibration if results are not completely compliant. Furthermore, correlations between variables can be esteemed, and the global knowledge of the problem, at the corporate level, can be enhanced. Other interesting features offered by this software are:

- Possibility to save snippets of the measurement file, when a certain condition of interest has been recognized. The snippet is a far shorter segment of the measure, which can be shared everywhere throughout the corporate network, pursuant of its low weight. They are written in the same format of the whole measurement file (MDF), and saved as $.dat/.mf4$ files.

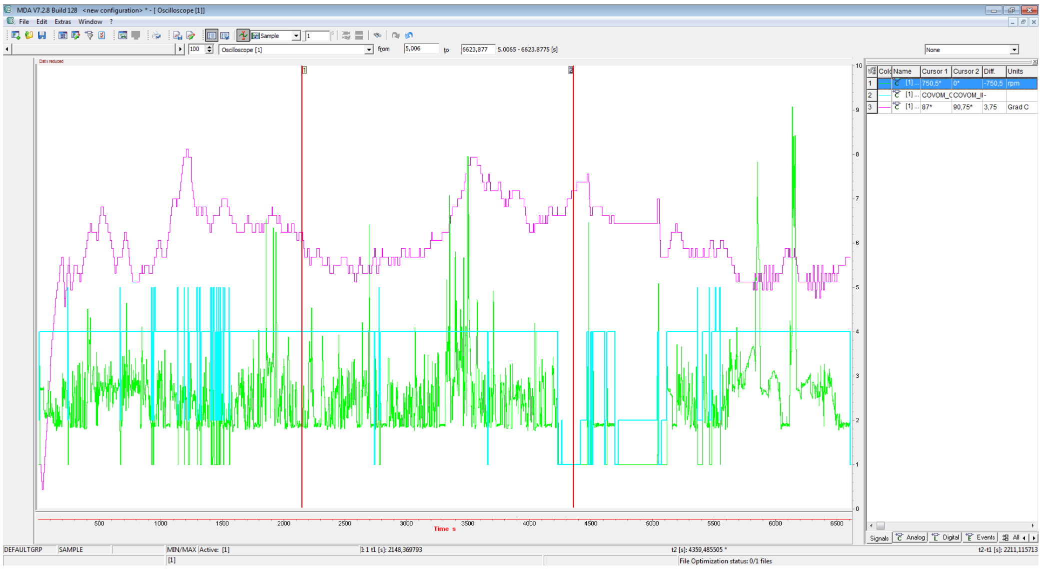


Figure 2.10: Footage of a possible MDA screen. The red cursors can be used as reference to point out every desired time window.

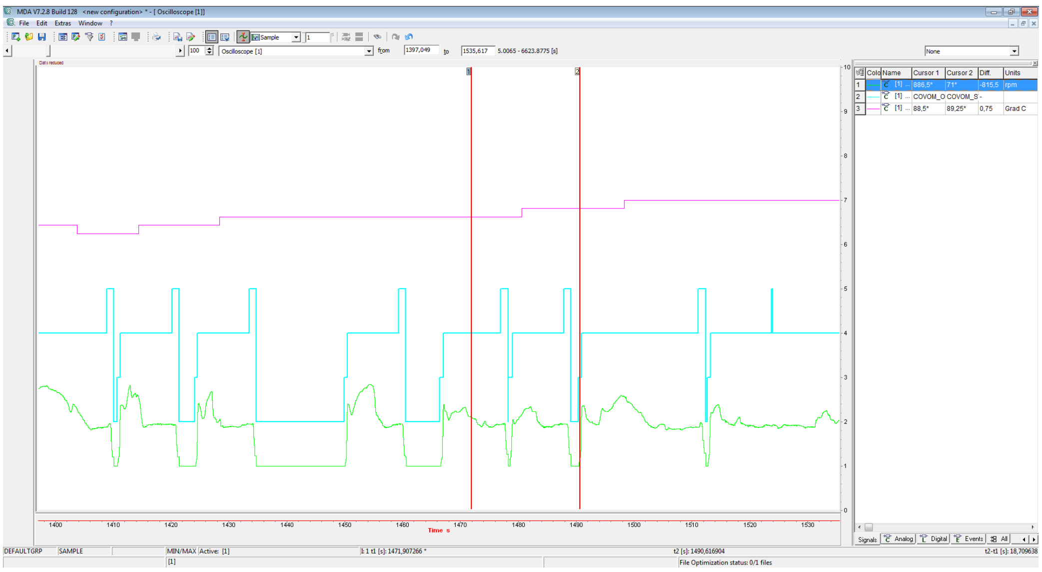


Figure 2.11: Zoomed footage of the same measurement file of Fig. 2.10. It can be seen the raster of each discrete signal, which differs on the base of their memory.

- Possibility to calculate derived signals, to analyze with larger level of detail the correlations between anyone of the experiment variables.

Chapter 3

Big data in the automotive industry. The Maserati example

Imagine a world *without data storage*; a place where every detail about customers, product performance or specification, every activity performed, or every aspect which can be documented is lost directly after use. Car makers would thus lose the ability to extract valuable information and knowledge, perform detailed analyses, as well as provide new opportunities and advantages, in every aspect of the vehicle development and the company management. Anything ranging from customer names and addresses, to subsystems reliability, to international markets performances, to employees hired, etc. has become essential for day-to-day continuity. Data is the **building block** upon which any organization thrives in nowadays world.

Now think about the extent of details and the surge of data and information provided nowadays through the advancements in technologies and the internet. With the increase in storage capabilities (e.g. online clouds) and methods of data collection, huge amounts of data have become easily available. Every second, more and more data are being created and needs to be stored and analyzed in order to extract value. Furthermore, data have become cheaper to store, so organizations need to get as much value as possible from those huge stored amounts. This environment has favored the onset of a buoyantly increasing tendency to generate traffic, since nearly every human activity has become partly or completely in touch with the Internet.

In Chapter Two, it has been explained how easy can be — in the everyday testing activity of a car maker — to generate hundreds of Gb of measurement files in just some hours, with the modern acquisition systems and communication protocols. Within a small-medium scale production automaker as

Maserati, there are many different fleet vehicles, not only bounded to the validation activity: this means that the volume of data to be stored every day is far beyond from the one that characterized my everyday activity in the validation team, which was anyway still remarkable. Furthermore, the *cost* of these data is very high, since it involves not only the production of fully-resembling vehicles that are destined to scrapping after use, but also thousands of work hours by drivers, engineers, project leaders, the cost of fuels, the acquisition systems, the cost of the energy to be provided to the plant, and much more. For this reason, data must be compulsorily made valuable, through detailed and significant analyses and through the measure of a number of channels that is hopefully as large as possible.

A new kind of data definition, capable of framing the exact mole of data that is shaping out our modern societies, was therefore needed. Big Data (BD) are data that exceed the processing capacity of conventional database systems, since they may be too big, may be transmitted too fast, or simply may do not fit the traditional database architectures [18]. Thus, to make them valuable it is necessary to define an alternative way for the elaboration.

BD can be defined according to the well-known "three V" classification, that contains the description of their characteristics:

- *Volume*: it represents the quantity of generated and stored data. Usually, a standard minimum size for BD is in the order of petabytes. Of course, it is a definition that will change over time, due to the continuously increasing data production of the society, and anyway changes in relation to the use of the specific organization. To be more specific, the big data processed by a National Authority for Taxes will be much huger than the ones processed by an automaker.

- *Variety*: it refers to the different types of data that are available. Traditional data types are structured and fit neatly in a relational database, while big data are usually known to be quite unstructured[1] and of low-density kind. Unstructured and semi-structured data types, such as text, audio, and video, require additional preprocessing to derive meaning and support metadata.

[1]Structured data are typically highly-organized and formatted to be easily searchable in relational databases. On the contrary, unstructured data have no pre-defined format or organization, making them much more difficult to be collected, processed, and analyzed. The former could be addressed as *quantitative data*, for instance people names, dates, addresses, credit card numbers, stock information, geolocation, and more, while the latter are mostly categorized as *qualitative data*, e.g. text, video, audio, mobile activity, social media activity, satellite imagery, surveillance imagery – the list goes on and on [19].

- *Velocity*: it is the speed — in terms of data rates — at which data are generated or received. Normally, the highest velocity of data streams directly into memory versus being written to disk. Usually, big data in the automotive industry have also *real-time* nature, hence they require real-time evaluation and action. This of course has a great impact on the ability of managing and studying them.

However, the definition of BD has evolved in time, and has been enriched by additional 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.
- *Variability*: the measure of how much data can vary in a certain spectrum is a parameter of consistency of data, which could hamper the processing procedure.
- *Flexibility*: it measures the capability of big data types to allow extensionality — i.e. the possibility to easily add new fields — and scalability, i.e. the possibility to enlarge their sizes rapidly.

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 overall generated and analyzed with a **specific question in mind**. The challenge of analyzing BD is coping with abundance, completeness 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 has always been too complex and difficult to win, but the chance became possible due to high-quality computation and new analytical techniques, comprising the usage of innovative tools. One of them, which is an important co-protagonist of my thesis work, is ETAS Moogle, a developing software for data analysis and statistics drafting, which will be introduced along this Chapter.

In the automotive industry, the usage of advanced analytics has become a necessary tool to deal with the complexity of the issues and the demanding requirements from the product quality point of view. The advanced analytics approach, for completeness, provides algorithms for complex analyses of both structured and unstructured data, and it may include sophisticated statistical models, machine learning, text analysis and so on. Among its many usages, advanced analytic can be deployed to find patterns in data, prediction, forecasting, and complex event processing.

3.1 On-Board Diagnostic system (OBD)

As vehicles became more and more complex for what concerns the electronic controls, for the reasons explained at the beginning of Chapter Two, the amount of data that the ECUs normally treat has increased up to the extent of requiring a suitable and enhanced database architecture to handle it. In addition, stricter and stricter regulations in terms of pollutant emissions led to the introduction of systems to monitor exhaust gas emissions externally to the official emissions driving cycle tests. All those systems are grouped in a unique standard named OBD (On-Board Diagnostic system).

OBD is a standardized vehicle's *self-diagnostic* and *reporting* system, giving the vehicle owner or repair technician access to the status of the various vehicle subsystems [23]. The amount of diagnostic information available via OBD has varied widely since its introduction in early '80s versions of on-board vehicle computers. Early versions of OBD could simply illuminate a malfunction indicator light (or *idiot light*[2]) if a problem was detected but did not provide any information on the nature of the problem, as the name suggests. On the other hand, modern OBD implementations use a standardized digital communications port (Fig. 3.1) to provide real-time data in addition to a standardized series of *diagnostic trouble codes* (DTCs), which rapidly allow to identify and recover malfunctions within the vehicle. Since it is strictly bounded to the vehicle's exhaust emissions monitoring, this diagnostic tool is fully operative both while the vehicle is running and when it is at rest. Hence, the engine system and components must be continuously monitored in driving mode so that compliancy with the emission limits required by law can be achieved in everyday use.

3.1.1 OBD standards

OBD has a short history, albeit full of updates. It officially started in 1988 in California, a place which has always been sensitive to environmental topics. In the following, a short overview of the actual standards that are recognized in most of the world countries is presented, whose differentiation has survived up to now.

[2]It was a single indicator lamp, usually labeled "*trouble*" or "*engine*", indicating serious and macroscopic troubles with the engine (e.g. low oil pressure, overheating, or charging system problems) and an imminent breakdown. This usage of the "engine" light was then discontinued in the mid-'80s, to prevent confusion with the MIL and due to the lack of specific information it could provide

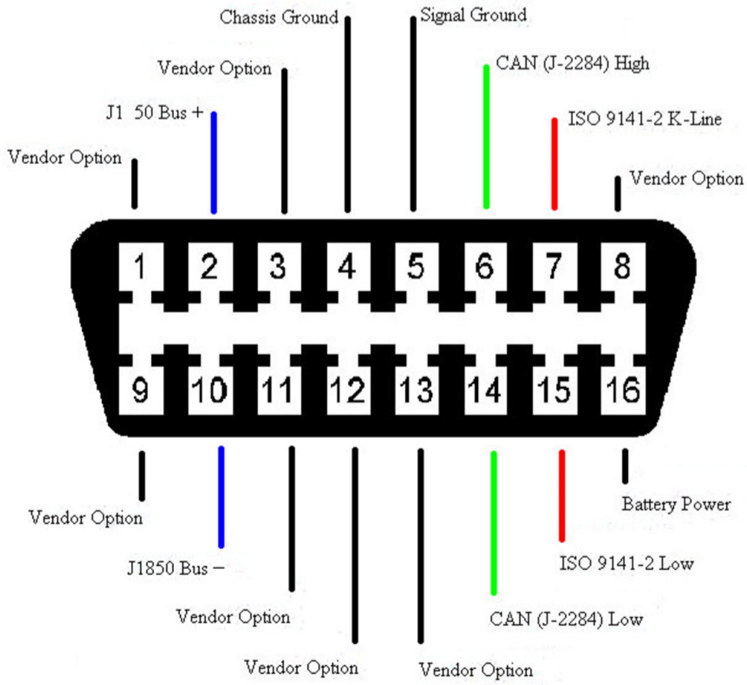


Figure 3.1: Representation of an OBD port with the description of every pin.

- **OBD-I (CARB)**: in 1991 the first stage of CARB[3] legislation was published in California with the name OBD-I. This first stage standard required 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 the installation of a malfunction indicator lamp (MIL, Fig. 3.2), alerting the driver about detected faults. On-board indications should have also be provided as a readout of which component had malfunctioned.
- **OBD-II (CARB)**: in 1994 the second stage of diagnosis legislation was published in California with the name OBD-II. In addition to the purposes of OBD-I, *system functionality* became monitored, too (e.g. plausibility check of sensor signals were now provided). In the case at hand, OBD-II stipulates that all emission-related systems and components must be monitored if they cause an increase in toxic exhaust gas emissions (hence the OBD limits to be exceeded) in the event of a malfunction. In addition, all the components used to monitor emission-related components or affecting the diagnosis result must be monitored.

[3] *California Air Resource Board*; it is the governmental agency for healthy air maintenance.

- **EOBD (European OBD)**: on-board diagnostics attuned to European conditions is termed EOBD, and it has been published since 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: therefore, EOBD standard is updated and revised every time that a new EuroX requirement is emitted. As a matter of fact, the last revision is the one concerning Euro6d requirements (September 2017), in which some OBD pollutant emission limits were decreased.

Figure 3.2: The Malfunction Indicator Lamp (MIL), present in the control panel of each vehicle.

As it can be imagined, OBD is not at all a fixed legislation: in fact, it is continuously updated and revised in order to further improve the effectiveness of the diagnostic system, and also to increase the number of phenomena 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.

3.1.2 OBD functions overview

Whereas EOBD only contains detailed monitoring specifications for individual components, the specific requirements in OBD-II standard are much more detailed. In this short paragraph, an overview of all the CARB-required parameters — according to the current standard — is listed for gasoline-engined vehicles. The requirements that are also described in detail in the EOBD legislation are marked by $^{\mathrm{E}}$:

1. Exhaust-gas recirculation system (EGR)$^{\mathrm{E}}$;
2. Cold-starting emission-control system;

3. Crankcase ventilation;
4. Combustion misses/misfire ring[E];
5. Fuel system;
6. Variable valve timing;
7. Exhaust-gas sensors (lambda sensor[E], NOx sensors[E], particulate sensor);
8. Engine cooling system;
9. Other emission-related components and systems[E];
10. In-use Monitor Performance Ratio (IUMPR) for checking the frequency of diagnostic functions in everyday operation[E];
11. Secondary-air injection;
12. Three-way catalytic converter[E], heated catalytic converter;
13. Tank-leak diagnosis, with at least electrical testing of the canister-purge valve[E];
14. Air-conditioning system (in the event of influence on emissions);
15. 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.

3.2 Data recording and storage

As said in Section 2.1, each automaker — included Maserati — should plan a mileage accumulation strategy for validating the results of calibration on its own vehicles. The higher the amount of data to be recorded, the higher the number of measured channels, the more complex and thorough the analyses. Since today the processing possibilities are quite unbounded, practically any data could be in principle interesting to be retrieved, especially considering multiple correlations sometimes existing between them. To widen as much as possible the spectrum of the driving conditions under analysis, the mileage accumulations have to comprise some kilometers in urban, extra-urban and highway routes, as well as mountain driving and on-track sessions. Typically, extreme situations — e.g. from the climatic point of view — are always of interest: specific winter tests in northern countries, as well as many summer tests in the most arid and hot places (e.g. the Death Valley, United States) can be led to verify the behavior of specific subsystems under extreme environmental conditions. Two examples could be, respectively, fuel consumption and the fan load.

Anyone of these tests is led by professional drivers, capable of replicating the required behavior. After each driving session, drivers extract data from the already cited data loggers, storing them on a USB pen drive/cloud, and loading the measurement files on Maserati's servers. Clearly, as the number of vehicles increase, the amount of stored data increases too, moreover data dimensions increase day by day, at quite a buoyant rate. Making first level calculations to give an idea the sizes of data involved, normally a driving shift means to have measurement files between 10-20 Gb; each day is composed of three driving shift, and considering the number of vehicles available to Maserati, hundreds — not to say thousands — of Gb have to be everyday stored. This is why Big Data is the name that can be fully-fledged assigned to the kind of data analysis handled by car makers. The stored data can be used for different purposes, but the main ones are:

- *Calibration*: it is necessary for imposing a determined behavior to the powertrain subsystems, in any driving conditions, with the aim of providing an adequate response to the driver's requests. For the practical point of view, the mere act of calibrating consists in the definition of the engine maps, which will be exploited by the control software for returning the desired outputs. Big data analysis is an outstanding aiding instrument for calibrators, which can immediately understand if their calibrations have worked properly, or if there is the need to furtherly 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, where it can be analyzed in order to perform an accurate diagnosis.

The diagnosis purpose is law-mandatory, as previously explained. At this stage, engineers must understand if there has been actually an error, or if a misdetection has occurred: in the former case, the root cause must be found and eventually the issue must be fixed, usually by substituting the damaged or faulty component, while the latter occurrence is usually a software fault, which has to be carefully analyzed and corrected in the next software release. For sake of completeness, *misdetection* means either that the software wrongly detected an error that in reality has not occurred, or that it has failed in deteting an error that actually occurred. This part of the process is of fundamental importance, as it involves the satisfaction of both market and brand's quality standards.

3.3 Weaknesses of standard data elaboration

The *status quo* of data elaboration process is too weak to withstand the increase of data available for analysis — together with the intrinsic increase of costs it implies — and overall to keep the pace of an increasingly competitive market. The standard procedure used to study measurement files to aid calibration and diagnosis articulates as follows:

1. *Find the right occurrence*: once one or more diagnoses — out of the total that are continuously performed in a vehicle — have detected an anomaly from data retrieved in the engine control module, an *error flag* is displayed on first level data analysis systems, such as Control-Tec. When calibrators, or members of the validation team, explore the reports about those diagnosis, they have to identify the correct instant of the correct measurement file — out of the many that are daily stored — to graphically evaluate the behavior of all the channels that are involved in the definition of the specific analysis in object, in that specific time window. Sometimes, for instance when anomalies occur after a long time from the accumulation period start, this operation can be hard because it may require additional efforts to retrieve the information about which driving cycle the vehicle was actually performing at the moment of the triggered event. This operation may involve the consultation of objected vehicles' *reliability histories*, which are simply excel tables in which every step of the accumulation period is recorded, included the mileage, the path followed, the location, the driver executing the driving cycle, the mileage and the general vehicle conditions.

2. *File opening*: this is a problem of informatic resources, that are intrinsically limited from the financial resources of a company, which are almost completely reliant on the sales volumes. Since data analysis deals with files which are easily larger than 1 Gb — that may be intensively processed, such as zoomed, enlarged, changed in scale, color, dimensions, raster and more — and since the analysis is not necessarily

limited to a single measurement file, the whole opening time of those sets of data can be surprisingly and unacceptably high.

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, while if it is not the case, this time increases. Moreover, as the displayed signals increase, the computational effort required to represent the signals increases as well.

4. *File study*: engineers have to find the reason for the anomalies by comparing what actually occurs in the interested channels of the measurement file, in the time windows that produced the errors, and what is the response of the control software in the face of those operating conditions. In the past and partly nowadays, this procedure was led by manually recording the values that the channels of interest assumed in specific time instants identified by calibrators, and then displaying them into Excel tables: in this way, it was easier to make graphical representation or manual interpolations, to highlight tendencies and unexpected correlations between variables. The main problem is that it is impossible, in a reasonable amount of time, to record enough points to realize valuable statistics, hence there is not the possibility to keep track of what is really going on over time from the physical point of view, and to sustain hypothesis with consistent data. Eventually, there are not always easy or clearly established reasons for any anomaly, but it is necessary to retrieve as much information as possible to adequately orientate the software or calibration modification in the next release.

It is easy to understand that such data analysis process is too highly time-consuming to handle such daily amount of data, and at the same time — for the reasons explained above — it is mandatory to make those data valuable to create enough value for the company. The big drawback of such a procedure is that it is highly time demanding. This is why a new and innovative tool for data analysis has been experimented by Maserati's Powertrain Validation team in the past two years: its name is Moogle and it is a data analysis online software powered by ETAS. The most interesting and challenging point, also with respect to the thesis experience point of view, is that it is currently under development and not yet commercialized.

3.4 ETAS Moogle

Moogle is a developing, post-processing software used within Maserati to analyze the big amount of data that are daily coming from accumulation programmes of the validation fleets. Its most valuable characteristic is that it can extract synthetic information about the status of the vehicle development at the engine calibration level, organizing it in simpler and intuitive forms as charts — comprehensive of all the measurement files stored in corporate servers — with the aim of producing statistics. Statistics are then needed to *validate* the progresses made with calibration cycle, which can be either confirmed or rejected. The procedure to get valuable statistics from the measurement files is subdivided in many steps, which will be described in the following:

1. *Information collection.* When calibrators or other members of the Validation team need to keep track of the progresses or the regresses that the updated engine control software determined in terms of desired vehicle behavior, they ask Moogle users for statistics. To execute a precise job, which hopefully does not require to be repeated, it is necessary to tune in with the figure that is communicating the request: this means to establish a prior dialogue in which the physics of the system that manifested the problem must be deeply understood, as well as what conditions caused the issue, and what variables need to be monitored to substantiate and circumscribe the phenomenology of the problem. This is a crucial passage, because understanding the physics of the problem allows to isolate just the most impacting variables.

2. *Trigger writing.* The trigger is a short C++ script (Fig. 3.3) — that will be better explained in following Sections — which is used by the software to properly isolate the conditions of interest. This procedure can be also said to *trigger the problem*, because with the programming ability of the Moogle user it is possible to reproduce in troubleshooting a pathologic behavior, with the subsequent aim of correction. Furthermore, a second function of the trigger is to indicate to the software the set of outputs that will be made available for graphical statistics: any variable that is needed to be displayed in charts need to be specifically indicated within the trigger script, otherwise it would not be at dispose.

3. *Trigger testing.* The drafting and preparation of a trigger is a process full of errors, which can be of many different fashions. The most common ones are **syntax errors**, dealing with the correctness of the C++ instructions to be given to the software: however, they are also

```
pre_RPM := timeshift/time [1] (RPM\ETK);

GME_CRANKING_OIL := detect/time {pre_RPM = 0 AND RPM\ETK != 0
                    AND OIL_REQ\ETK = 1; this.duration >= 30.0};
GME_CRANKING_OIL += max PRESSURE\ETK;
GME_CRANKING_OIL += {
    MAX_PRES := max PRESSURE\ETK;
    GME_CRANKING_OIL_MAXPR := detect/time {PRESSURE\ETK =
                              MAX_PRES};
    GME_CRANKING_OIL_MAXPR += start_value TOIL\ETK, ODOMETER\ETK;
    };
```

Figure 3.3: An example of short C++ trigger.

the simplest ones to be avoided because with experience they typically diminish. Then there could be **alias errors**: since measurement files are a product of two different acquisition systems and since there are two different developing ECUs providers, which have been Introduced in Chapter 3, channel names can vary significantly. For this reason, since triggers should be universal and cannot be changed once they are processed with measurement files, it is frequent to write a channel name that is present with a different name in the measurement files of interest. In that case, statistics are not performed and a lot of time is wasted. For these and other reasons, it is necessary to test the trigger with a limited number of measurement files, to see if everything works. This, thanks to significantly lighter data processed, allows to save on time and to do the thing right once.

4. *Trigger conversion.* C++ trigger scripts need to be converted from the *.mgl* format to *.dll* format. For this activity, a dedicated tool – named *Compling* – is available within Maserati. It is simply a drag&drop script, which has the aspect of a standard Windows executable (Fig. 3.4).

5. *Indexing of the measurement file.* To make the indexing of a Moogle folder means to put in crawling[4] a certain number of measurement files with a certain number of triggers. During this procedure, Moogle reads the instructions contained in the trigger and applies them to extract

[4]Although we use it as a synonym of processing, *to crawl* is a very nice and interesting verb having multiple meanings. The most adherent one, with respect to our discussion, is *to move slowly or with difficulty* (i.e. in the same way Moogle meticulously and patiently extracts data), as you were stretching out your body along the ground on hands and knees.

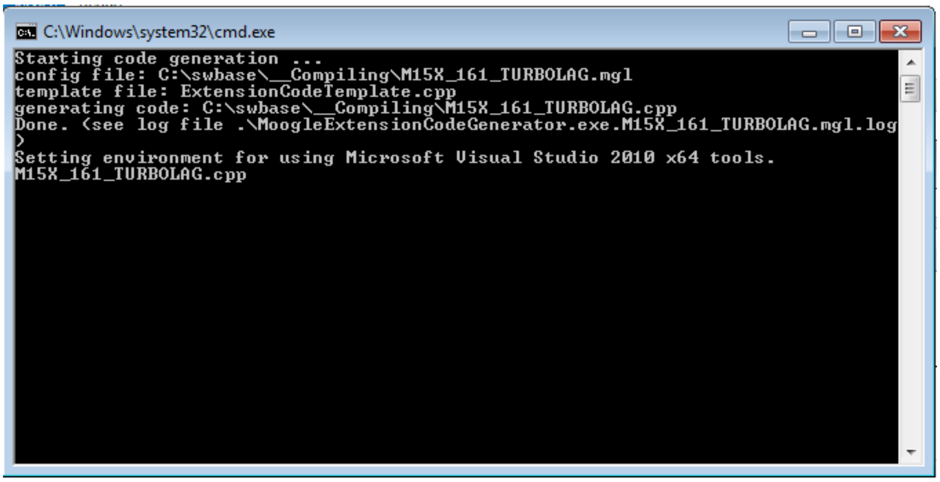


Figure 3.4: The interface of Moogle's *Compiling* tool.

data from the measurement file. This process is probably the most time-consuming part of the whole statistics generation, since it deals with the processing of huge amounts of data, however its duration may depend on many parameters. Some of them are the processing capability of the corporate servers, the quantity of triggers simultaneously crawled, the quantity of measurement files present in the folders as well as their dimension, the quantity of folders simultaneously crawled, and so on.

6. *Creation of the statistic reports.* Exploiting Moogle's graphic interface, a great variety of charts can be realized, displaying all the data extracted during the crawling.

3.4.1 Trigger definition

As said in Section 3.4 introduction, a trigger is a small C++ program directly written by Moogle user(s). The definition occurs always as plain text, which can be drafted in any text editor: inside Maserati, the text editor of choice is **Notepad++**, since it allows to validate the syntax, once specified the language. In a trigger, a series of elements must be specified:

1. *Trigger name.* Each trigger must be named with different labels, to facilitate the recognization from the user. In the eyes of Moogle, the name is nothing but a differentiating element, and there is no need to write any specific code or address, apart from a different string for each trigger. The only criteria that are followed to name the trigger

are of a corporate kind: aiding standardization and the conservation of technical documentation, triggers are usually named as follows:

`Motorization/Vehicle_Market_Title`

Where "`Title`" stands for a brief description of the physical problem and/or the channels involved, or which behavior we want to see. Two examples of names may be:

`GME_NAFTA_BLOWBY_DETECTION`;

`M15X_161_RPM_UNDERSHOOT`;

2. *Trigger conditions*. They actually determine the *width*[5] of the triggered time windows, which depend on the specific calibrators' requests. Trigger conditions are the most important part of a trigger, since they allow to shape up the problem and to retrieve just the interesting data, amid a jungle of not interesting ones. The **conditions definition** is an operation which is concerted between many people of the team, because a deep understanding of the problem is necessary to isolate only the desired behaviors. One example of trigger condition could be:

   `M15X_M161_RPM_AT_IDLE := detect/time {VEHSPEED\ETK = 0 AND RPM\ETK > 100};`

   The symbol "`:=`" represents the *defining indicator*: this means that everything that lies on the right of this symbol is interpreted by Moogle as the conditions to be pursued. At its left there is the trigger name, whose format has been specified at Point 1, while at its right there is specified the trigger condition, with its right syntax. The logic behind this condition is to trigger every time window (`detect/time`), amid all the measurement files of the folder, in which the vehicle is at standstill (vehicle speed equal to zero) and the engine is on (*rpm* larger than one hundred rounds per minute). The delimiter \ETK is the name of the acquisition system which have measured the channel: in the example case, this one has been measured with ETK protocol.

   With this simple instruction, Moogle will be allowed to scan each single measurement file in order to find every time that the specified conditions are met. If this happens, an *occurrence* is found out, and this can be plotted in many forms that will be described hereafter. The more comprehensive the conditions, the larger the number of detected occurrences; on the other hand, if the conditions are too strict,

[5]Because, isolating the onset of certain conditions, their duration is automatically retrieved, too.

non realistic (in the sense that they do not respect the physics of the phenomenon) or badly written from the point of view of the logic, no occurrences could be found. Of course, the advantage of writing a trigger is that there is theoretically no limit to the number and the type of conditions that can be specified, thus expanding the analysis capabilities to unknown boundaries. In this way, also very particular and rare conditions could be in principle detected.

3. *Trigger outputs*. They represent the variables that will be made available to display in the report page of the software. If the user does not insert any output, even if the software correctly processes the data and individuates the desired windows, no data could be displayed at all. An example of trigger output could be:

   `M15X_M161_RPM_AT_IDLE += max ODOMETER\ETK, RPM\ETK, ENG_TEMP\ETK;`

   Which can extract the maximum value (`max`) of the channels that are written in the following: in the present case the odometer indication, the engine *rpm* and the engine temperature. The symbol "`+=`" is necessary to tell Moogle to consider everything that follows as an output. Clearly, it is not necessary to have included a channel in the trigger conditions to make it disposable as an output.

Once the trigger has been properly written, it is possible to save it with `.mgl` format. If everything is correct, the aspect it should have resembled is the following (Fig. 3.5):

```
M15X_M161_RPM_AT_IDLE := detect/time {VEHSPEED\ETK = 0 AND RPM\-
                         ETK > 500};
M15X_M161_RPM_AT_IDLE += start_value ODOMETER\ETK, RPM\ETK,
                         ENG_TEMP\ETK;
M15X_M161_RPM_AT_IDLE += end_value ODOMETER\ETK, RPM\ETK, ENG_TE-
                         MP\ETK;
M15X_M161_RPM_AT_IDLE += max ODOMETER\ETK, RPM\ETK, ENG_TEMP\ETK;
M15X_M161_RPM_AT_IDLE += min ODOMETER\ETK, RPM\ETK, ENG_TEMP\ETK;
```

Figure 3.5: Sample trigger provided with outputs to be displayed in charts.

The trigger logic displayed above, in the face of the same window described at point 2, retrieves the values of odometer, *rpm* and engine temperature at the initial and final instant of the time window, plus the absolute maximum and minimum in that window. It is not necessary that the channel to be displayed in the outputs are present in the trigger logic: it is enough that they are present in the measurement file, which in turn depends on the chosen experiment.

The example made above is showing the nucleic concepts on how a trigger has to be defined, but the logic structure can be also much more complex, for instance introducing *nested triggers*[6] to isolate furtherly different conditions with respect to the ones isolated by the external window.

To introduce the potentialities of nested triggers, a third example (see Fig. 3.6) is provided in the following. This trigger does the following things:

- As header, a calculated channel (`pre_STARTSTOP`) is defined. It is called likewise because it is actually a newborn variable, which is the result of a certain mathematical operation made on a pre-existing channel. In this case, the operator `timeshift` shifts all the values of the channel `STARTSTOP\ETK` (identifying the status of the start&stop system) one sample before. In this way it is possible to insert in the trigger logic a way to trigger the instant in which the channel `STARTSTOP\ETK` commutates from the value 2 (i.e. engine stopped) to the value 3 (i.e. cranking).

- In the trigger logic's curly brackets, the operator "`;`" has been inserted to indicate both the *closure* of the windows opening conditions, and the definition of a closing condition for the trigger, specified to the right of this operator. In absence of a closing condition, in fact, Moogle verifies instant-by-instant that the conditions for the trigger activation are met, with the consequence that it is difficult to extract, for instance, the duration of establishment of certain conditions. By using a closing condition, on the other hand, it is possible to force Moogle to keep opened the triggered time window until a certain different condition verifies. The window that is triggered in the example opens when the vehicle cranks after a start&stop and closes only when the command of the water pump reaches the calibrated threshold of 50. In this window, the duration (`delta(time)`) is retrieved, as well as the values of the *rpm* at the start of the window (`start_value RPM\ETK`). When a trigger logic contains the parser "`;`", it is said to be a *schmitt trigger* type (the origin of the name, which has not the capital letter at the beginning, is uncertain).

[6]Nested triggers are nothing but subsets of further indications, whose evaluation take place *only if* the hierarchically higher condtions have been already verified.

```
pre_STARTSTOP := timeshift [1] (STARTSTOP\ETK);

TRIG_1 := detect/time {pre_STARTSTOP = 2 AND STARTSTOP\ETK = 3;
          WCAC_PUMP_CMD\ETK = 50};
TRIG_1 += delta(time);
TRIG_1 += start_value RPM\ETK;
TRIG_1 += {
    NESTED_TRIG := detect/time {PRES_MEAS < 90};
    NESTED_TRIG += delta(time);
    NESTED_TRIG += {
        SUBNESTED := detect/time := {ENG_TEMP > 60};
        SUBNESTED += delta(time);
        SUBNESTED += start_value ODOMETER\ETK;
        };
    };
```

Figure 3.6: Sample trigger in which many possible logical solutions are listed out.

- `NESTED_TRIG` is nothing but a nested trigger, which means that — other than restricting the time window with additional conditions — it inherits the conditions already specified in the external trigger. In this specific example, `NESTED_TRIG` retrieves the time in which the pressure measured in the intake manifold (`PRES_MEAS`) is less than 90, within the time window specified in `TRIG_1`. The same happens for the sub-nested trigger `SUBNESTED`, which adds the further condition of having the engine temperature (`ENG_TEMP`) above 60 °C.

The examples showed in this Section are useful to understand the potentialities of the trigger as mean to implement a very wide range of logics, which can subsequently lead to isolate practically every condition that has been recorded in the measurement files. The higher the knowledge of the software language, in terms of syntax and operators that can be used, the higher the complexity of the analyses that is possible to lead out, the smaller the time required, regardless of the complexity of the problem. In the same way, the more complex a trigger, the higher the computational effort required to the software, and priorily to the corporate servers, the higher the time required to get coherent and presentable results.

3.4.2 Trigger testing

Once the trigger definition has been completed, it can be saved in `.mgl` format. Before running the crawling of the folders containing all the measures of a vehicle, it is necessary to test the trigger with a smaller subset of measures, to verify if it works as expected. It can happen, for instance, that the designed logic works for very particular cases, ignoring the majority of the ones that would be interesting to trigger. As it has been said, since the crawling is a very time-consuming process, it is better to be sure about the occurrences that the trigger is able to find out before actually running the crawler. To test triggers, ETAS provides a specific environment — called **Moogle Test**, whose interface can be scoped in Fig. 3.7 — that is functionally very similar to the standard Moogle version, also called Moogle Productive.

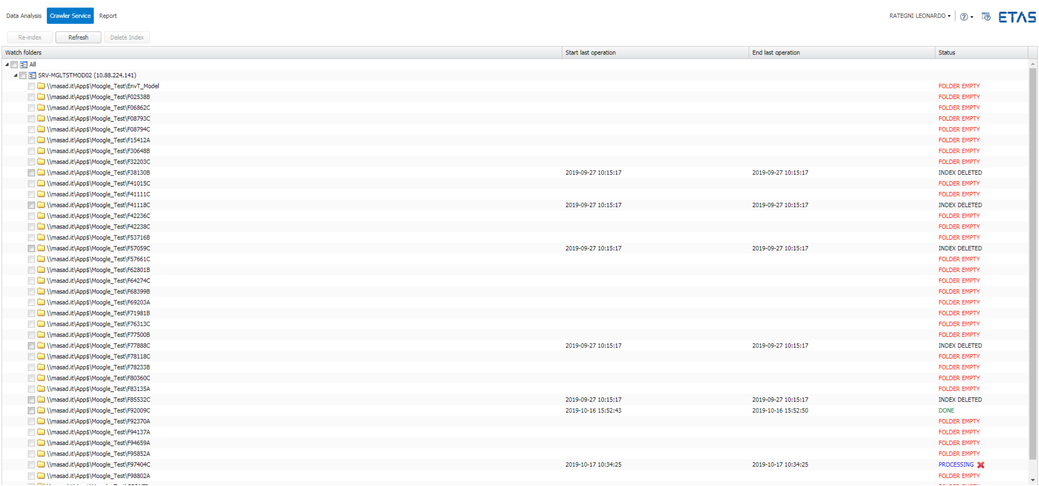


Figure 3.7: Moogle Test main interface, the *Crawler Service* page.

In the figure above, it is possible to see all the list of Moogle users that have been assigned to a specific Test folder, in corporate servers. In the complete list, there are only two folders that are actually active, the ones belonging to me and my colleague. The test of a trigger occurs in the same way of Moogle Productive crawling: once the selected subset of measurement files has been loaded in the Test folder, together with the testing trigger, it is enough to proceed to the *indexing* of the folder. After this operation, the icon “PROCESSING” will appear in the “*Status*” column, whereas when the indexing finishes, the icon “DONE” (or “PARTIAL DONE”, if the indexing of some measurement files has failed) will appear, instead (Fig. 3.8).

Figure 3.8: Icons “PROCESSING” and “DONE” in the “*Status*” column.

The biggest difference between Moogle Productive and Moogle Test, however, is that the latter can process just a limited amount of data, up to 40 Gb at a time. For this reason, the user must select very carefully the measurement files to be tested, preferably respecting some guideline criteria:

- Include selected measurement files coming from the whole accumulation period, in order to scope the differences implied by the various engine control softwares that have been equipped with.
- Choose both very particular measures, for instance relative to a very particular driving cycle, and more uniform and foreseeable driving conditions, in order to verify that the trigger logic is robust with respect to a wider range of operating conditions.
- Choose both long and short measures, belonging to different vehicles. This is done to comprehend all the differences that could be present between different vehicles and/or between vehicles of different markets.

Most important, the selected measurement files must contain occurrences related to the trigger conditions, so that the user can be sure that the trigger recognizes them and extracts the selected information. Once the test indexing has been successfully completed, if those criteria have been applied correctly and the trigger conditions are not too strict, it would be possible to display charts of the so obtained information. The Test environment is not only useful to test the trigger before the crawling phase, but it can be also exploited to make quicker analysis when the complete set of measurement files is not numerous: this could be the case for specific calibration procedures, or the emission cycles (WLTC), which generates short and a few measurement files.

3.4.3 Trigger conversion

As previously mentioned in Section 3.4, triggers are firstly written in `.mgl` format, since it allows the definition of a trigger in standard text editors, such as Notepad++. This editor has many advantages, such as the high portability, its lightweightness, the possibility to validate the syntax and to recognize a great range of programming languages. On the other hand, the drawback is that Moogle is slower in computing `.mgl` files: hence, when bigger data have to be processed, it is necessary to convert the `.mgl` trigger file to the `.dll` format. Since the compiling tool is a simple drag&drop script, with this expedient the computation speed increases without any extra cost.

3.4.4 Trigger indexing

Once the trigger has been converted, it is possible to start the real data processing. As mentioned, this is done by crawling all the data with the triggers of interest from the *Crawler Service* page, which can be seen in the interface of Fig. 3.9.

As it can be seen, this interface is very similar to the Moogle Test's one, previously displayed in Fig. 3.7. Instead of including the Test folders, specific for any Moogle user, it directly includes *vehicle folders*, containing all the measurement files of their respective accumulation period.

Each vehicle folder can be selected and indexed, once the users are sure that all the triggers needed for the analysis are included in the folder. Clearly, if a trigger is not included in the right folder, no analysis would be possible, and a considerable amount of time would be wasted. Multiple indexing at a time are possible, since the software autonomously require partialized processing power on the base of the number of indexings it has to complete.

The indexing process has a very variable duration, which depends both on the number of triggers and measures simultaneously present in the crawled folder, but the order of magnitude is between 5 and 10 hours. For this reason, it is necessary to organize the daily work to house the crawling in the last hours of the day, mainly for two reasons: to have the required results available for the following day and to not clogging the software if some other data, already processed, need to be extracted and revised again during the day. Furthermore, a common sense rule suggests to crawl only the interesting data, to save on power from the server. Additionally, it is necessary to periodically archive the unused triggers to save on computational effort and time: to this aim, a dedicated archive for any `.mgl` and `.dll` file is available for consultations, modifications or simply certification.

3.4.5 Creation of statistical reports

Once the indexing is finished, the user can create the statistical reports containing the results of the processing, which will be eventually shared with the colleagues. The reports can be realized in a specific Moogle page (called *Report page*, Fig. 3.10), in form of graphs of different fashions. Each variable that has been included in the trigger outputs can be used as chart axis, so that the analysis is as much as possible targeted.

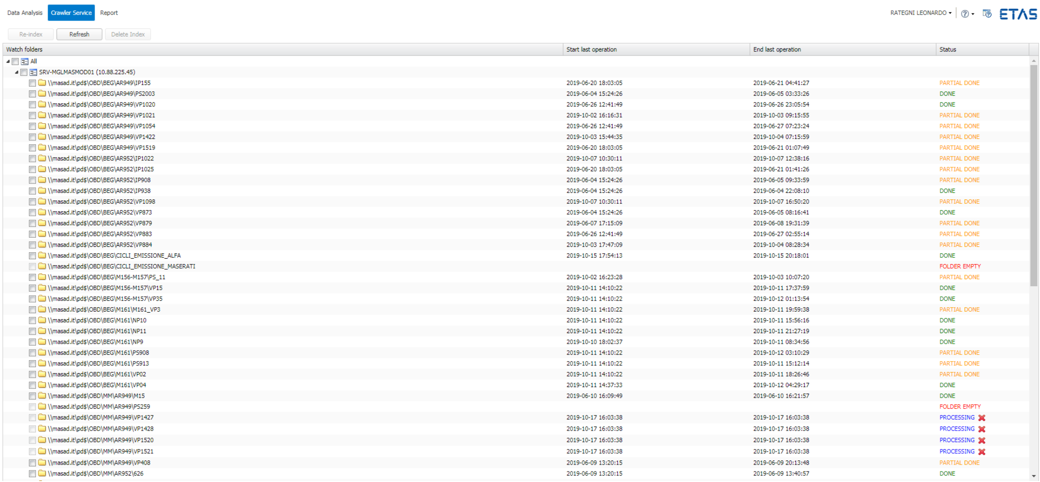


Figure 3.9: Moogle Productive's *Crawler Service* page.

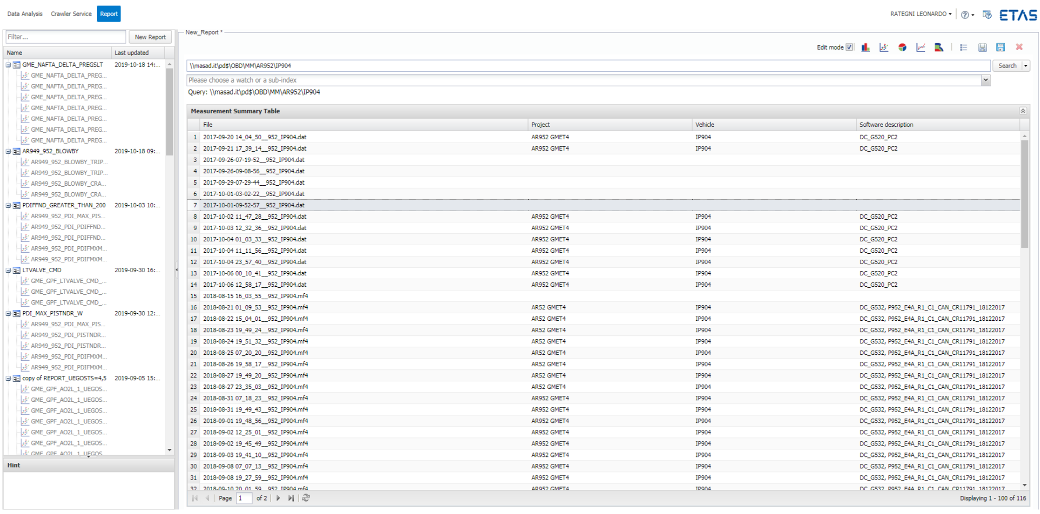


Figure 3.10: Moogle Productive's *Report* page.

At the left side of the page, the column of saved reports can be accessed. In this space, the old reports can be saved: this is a particularly useful feature — especially for reports concerning a lot of charts — to have an online record of any report that is produced. Furthermore, since Moogle automatically logs out after a short time (around half an hour), and since all the progresses made on the report page are cancelled when this happens, the presence of a section with saved reports can help in saving on time when an analysis needs to be repeated, for instance when data or trigger conditions have been updated. The right part of the page, on the other hand, displays all the measurement files that are present in the folder specified to the top, together with some insights.

It is possible to realize five types of charts:

Scatter chart

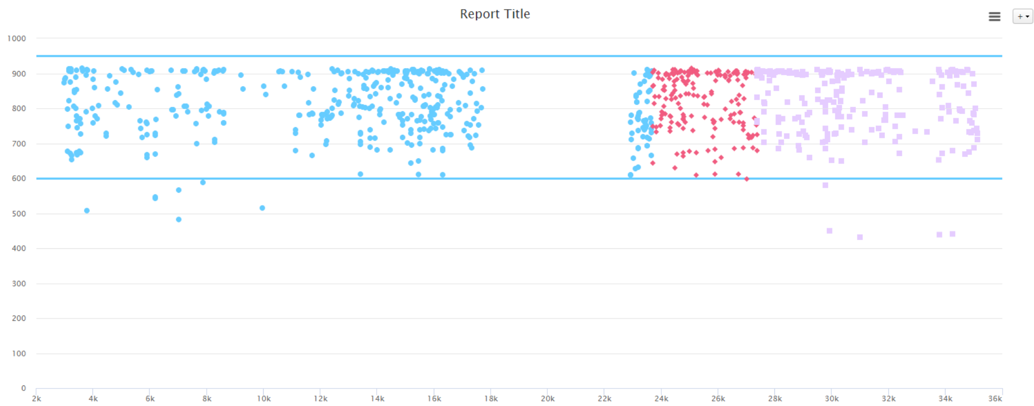


Figure 3.11: Example of *Scatter chart*.

The scatter chart is the most common kind of report that is used to present the results of a statistic throughout the company. The reason is that this kind of chart allows to highlight *any occurrence* that the trigger has given, without any sort of data aggregation. In other words, it can be considered the “cleanest” type of data, the least labored one. In Fig. 3.11, it is showed the scatter chart of a trigger which individuates the maximum values of a pressure differential in some particular driving conditions, on the base of the odometer indication. As it can be noted, it is also possible to insert thresholds, giving an idea on the compliancy of the specified variable from the point of view of the quality corporate requirements. Furthermore, each colour represents a different engine control software, therefore the compliancy progression can be compared. In this case, the very majority of the occurrences is compliant with the quality requirements of the company (represented by the blue horizontal thresholds), in the nearby of a certain confidence interval. Usually, for sake of differentiation, each chart is titled after the trigger and the variables displayed: Fig. 3.11 chart, for instance, could be called — in case of Maserati vehicles — `M15X_M161_max(PDIFF)_X_(ODO)`.

Variables name, as well as unit of measures and engine control software specifications have been hidden due to confidentiality reasons; however, the generality of the illustrated concepts is unaffected.

Bar chart (histograms)

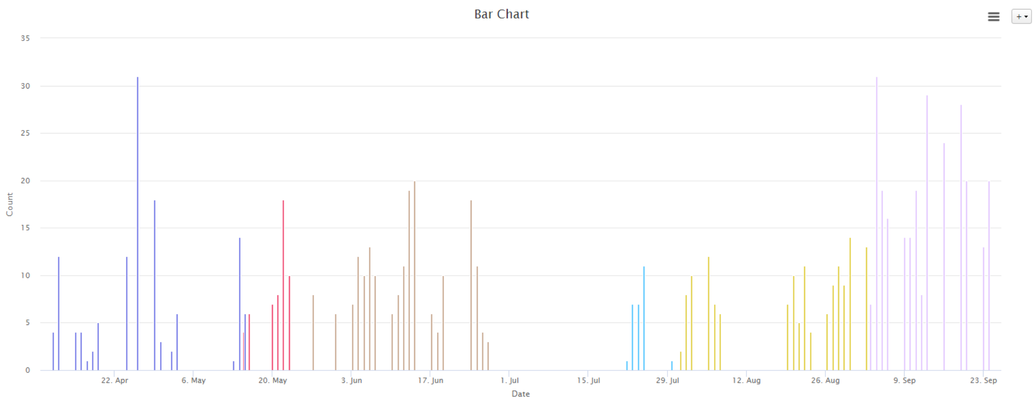


Figure 3.12: Example of *Bar chart*.

Data can also be arranged in form of histograms, for instance when it is more important to highlight the relative weights of each indicator, rather than a precise spectrum distribution. In the case presented above (Fig. 3.12), the y-axis variable "Count" specifies the number of times that a specific trigger has activated, on the base of the measurement file date. Each measurement file corresponds to a particular driving cycle, and each color of the histograms represents a different vehicle. This kind of chart could be particularly useful, for instance, when it is needed to know the distribution in time of how frequently a certain diagnosis has activated, for benchmarking and/or corporate monitoring reasons. Actually, data could be much less condensed than that, especially if the number of indicators to be monitored is smaller and there is the need to have highly aggregated data.

Line chart (trendline)

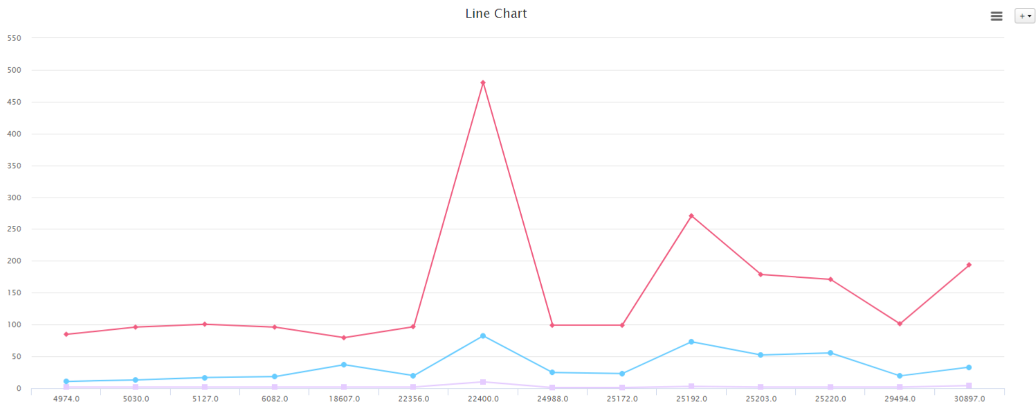


Figure 3.13: Example of *Line chart*.

Line charts can be useful to convey qualitative ideas about the *growth trend* of the selected variables. In the case presented in Fig. 3.13, the growth trend of three different variables is displayed on the base of the odometer indication. Each point of the line is an occurrence, therefore the higher the number of occurrences, the lower will be the memory of the output signal, the higher the accuracy of the statistical characterization.

This type of chart is less frequently used because in most cases the tendencies of the analyzed phenomena are sufficiently known, and can be equivalently stated starting from a scatter or a bar chart, according to the engineer knowledge.

Pie chart

Pie charts can be useful when it is necessary to convey ideas about the mutual proportions between data, neglecting the quantitative meaning or their distribution. In the example of Fig. 3.14, the pie accounts the number of times — in percentage — that a trigger has detected occurrences in the whole time spectrum, expressed in the form of dates. Each colour represents a day, while the percentage represent the concept explained above. This kind of graph is not particularly used, since we are usually interested in more analytic data, less aggregated and more interpretable.

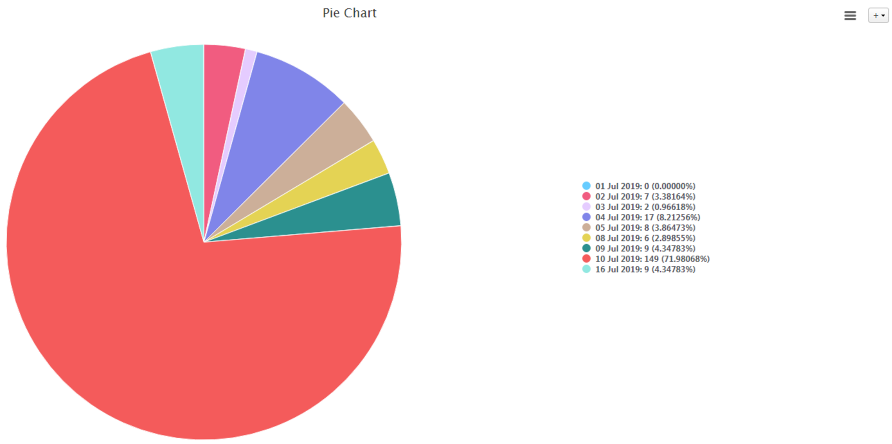


Figure 3.14: Example of *Pie chart*.

Heatmap chart

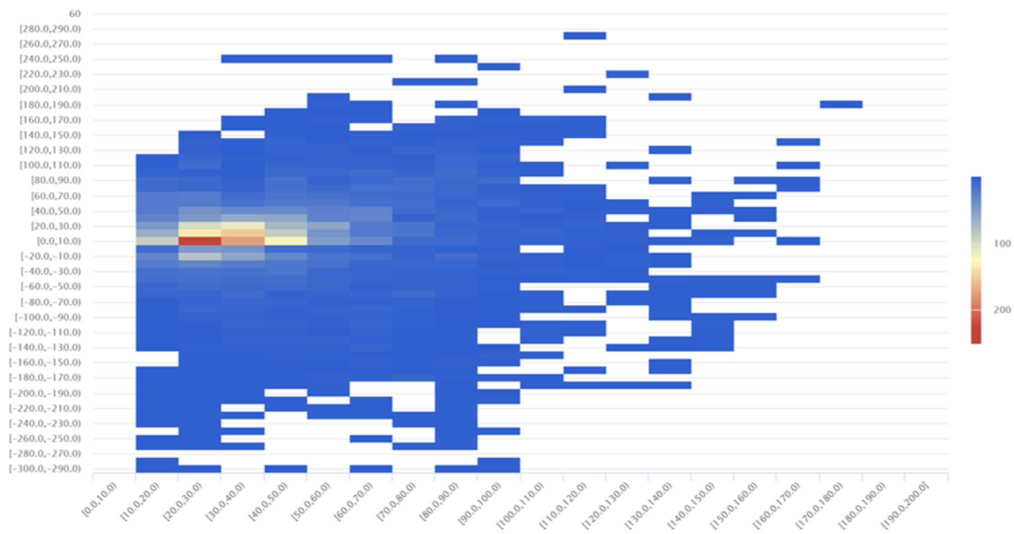


Figure 3.15: Example of *Heatmap chart*.

The main difference between a heatmap chart and the others is that the former requires the definition of a third dimension, increasing the comprehensiveness of the illustration. In this way, in addition to the two standard variables, it is possible to see the trend of a third variable in relation to the other two. Furthermore, the definition of the intervals for every axis, as well as the step between two classes of equivalence, is mandatory to build up the chart. As it is possible to see in Fig. 3.15, once the chart is defined it is possible to show on the x,y axes both the intervals and the steps defined before.

Every chart that has been described in the current Section can be exported and shared throughout the Powertrain department of the company, by means of the corporate network. Obviously, it is not sufficient to produce graphs and to send them to the interested people, but it is requested that the author of the statistics could adequately present the work he/she has done, by explaining — complete of every step and considerations about the reasonings made, as well as first level interpretation of the outcoming data — all the accomplished tasks by means of official and classified e-mails. The charts can be exported in the following formats:

- Simple pictures (`.png` or `.jpeg` formats);
- `PDF` report, containing all the charts that have been realized with the same set of measurement files;
- `.csv` excel files, which consists of columns of data that Moogle has used to calculate all the charts;

The latter are usually the preferred way to crown the graphic contribution, from the point of view of data sharing. In fact, by reading a `.csv` it is possible to spot out the occurrences that could be of particular interest, for example because they may be out of quality thresholds, or reporting unexpected values.

3.4.6 Data Analysis

Data Anaysis page is the third and last Moogle's macro-section. This environment is very useful to get deeper insights about what Moogle has processed, if the occurrences are coherent with what it would be auspicable to obtain and if a particular trigger has registered occurrences.

On the left side of the screen in Fig. 3.16, all the measurement files that have been processed are available. For any selected measurement file it is possible to see — in the right part of the screen — four important information to get preliminary insights about the available data:

- *File info*: in this subpage there is any information that is available to Moogle's eyes, and the amount of it depends on the metadata codified within the measurement file. Those ones, in turn, depend on the experiment that has been configured and the acquisition software that has been utilized by the calibrator. These data can include some important information like the vehicle name, the engine software mounted,

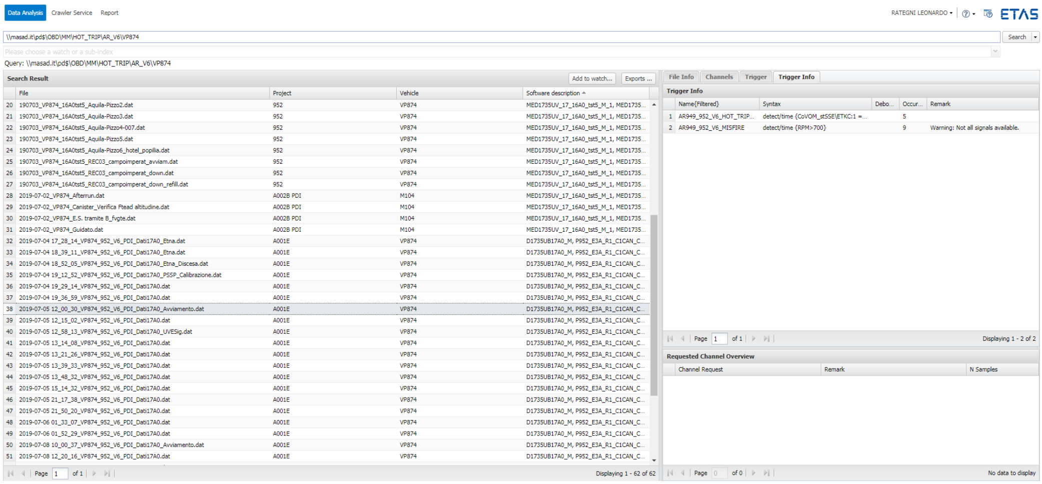


Figure 3.16: Moogle Productive's *Data Analysis* page.

the comments inserted by calibrators, the acquisition system, the date, the format of data, and many other information which can be useful for reporting's sake.

- *Channels*: here, all the channels that have been inserted as condition or as output by anyone of the triggers that have been processed are showed. This section can be usefull to understand if there are missing or non-recognized channels, and what values they assume in particular time instants.

- *Trigger*: this is maybe the most important sections of this page. Here, it is possible to spot out the exact duration of any triggered time window, for anyone of the triggers present in the indexed folder. This means that the user has the possbility to know if the triggered window opens and closes as desired, if the duration is realistic, if the number of seconds involved are compliant with what can be scoped through the measure in MDA, and so on.

- *Trigger info*: this subsection does the same of the previous one, but without giving the exact indication about the the windows duration. It simply gives a more synthetic information, like the counter for anytime an occurrence has been detected, instead of their analytic description. This subsection is needed because, if the occurrences are many, it could be discomfortable to roam the *Trigger* subsection, searching for the occurrence(s) of interest.

All those subsections are precious, especially for an expert user, to evaluate if the analysis has been correctly done before actually generating the charts, i.e. saving some more time.

3.4.7 Moogle's aid to engine calibration

Moogle can be useful not only as a mere post-processing software, giving validations of theses and theories that were already consolidated within the company, but also as aiding and supporting element to the calibration cycle. The standard calibration method, in fact — as explained in Chapter 2 — consists in developing a certain engine map, based on previous studies, and then testing it on some vehicles for some time, usually days. Afterwards, calibrators have to study carefully the measurement files in order to understand if their function worked properly, or if there is the need of corrections. This cycle lasts until the engine function has not been perfectly calibrated, to be compliant with the quality gates.

With the use of Moogle, this approach changes totally: the calibrator can now directly ask Moogle users to create dedicated triggers that isolate the conditions in which any function or map should act on. This could in principle eliminate the problem of multiple on-track validations, thus saving on many useless driving cycles: simply, the measurement files that had been recorded once could be **processed several times**, with several engine maps, to see if the results could be compliant or not. It should be remarked that this process, as explained, cannot in any case avoid completely the need of an on-track validation, since the answer of the real world is always priority with respect to the software one. Moreover, calibrators could only use this tool as a support for their everyday work, but it could never replace the need of studying fresh measures when the engine control softwares are updated.

To have a clearer view of Moogle's potential, as aiding tool to calibration, the most pleasant features will be here illustrated:

- The amount of data that can be processed are enormous. Maserati's database contains years of measurement files of many different vehicles: while Moogle processing can retrieve statistics upon this huge sample base, calibrators could only rely on few days of recorded data, and on a smaller number of vehicles. This leads to a more structured and statistic-oriented approach, which can enlarge to unknown boundaries the quality of improvements on present and future vehicles, since the *finesse* and the robustness of the conclusions are remarkably higher. In continuous improvement optics, this key feature reveals to be vital.

- There could be a huger time saving: in fact, calibrators can preliminarly study the results of his calibration on older vehicles, understanding if the orientation of their work could be correct, and allowing them to focus on that particular direction, without wasting time with trial-and-error approaches.

The company is then trying to continuously increase the exploitation of Moogle in its everyday activities, obtaining the double advantage of speeding up the calibration cycle and forming new people to work within the company. Actually, Moogle is currently under development, not only with Maserati but also with other market players outside Italy, therefore it is not so stable and cannot be considered — in strict sense — a company asset. The important point is that the advance with which Maserati has started this journey is remarkably higher than other car makers', since we are the first in Italy to implement a solution of this kind. Furthermore, the direction undertaken requires a finer and finer software development, which should be done by each master's candidate cycle together with ETAS technicians.

3.4.8 Software development

Along with the task of analyzing data and issues, producing reliable statistics, interfacing with many different teams and colleagues, and providing technical documentation to be preserved and used for archive purposes, one of my duties was to establish and maintain a solid collaboration with ETAS technicians for the development of the software. The development basically consisted in noticing issues while working, signaling them in the most transferable and reproducible form to ETAS, and sometimes proposing the introduction of new features, on the base of the everyday life deficiencies of the software. Each one of these reported issues or development purposes were labelled and archived into a shared **Open Point List** (also known as OPL) Excel file (Fig. 3.17): this file was available to both Maserati and ETAS technicians, to keep track of all the progresses made and the issues already solved. Generally, the steps for discussing new points to be added to the OPL are:

1. The problem is noticed while using the software: a signaling communication has to be written to ETAS in form of e-mail, describing all the steps that have led to the problem, the trigger logic, the size of the indexed folders, some images providing a support to the explanation, and generally everything that could be useful for the comprehension of the problem.

2. ETAS answers: if the problem can be solved with ease with simple implementations from one side or the other, the signaling can be archived without further efforts. If the problem is more complex or not reproducible in ETAS servers, telematic meetings have to be organized between ETAS and Maserati, in presence of two ETAS developers, the two Moogle users, and a third ETAS employee — based in Italy — who remotely act as a bridge to consolidate and refine communication between the parts. This person has also the duty of moderating and reporting meeting progresses to both parts, and to organize meetings upon request.

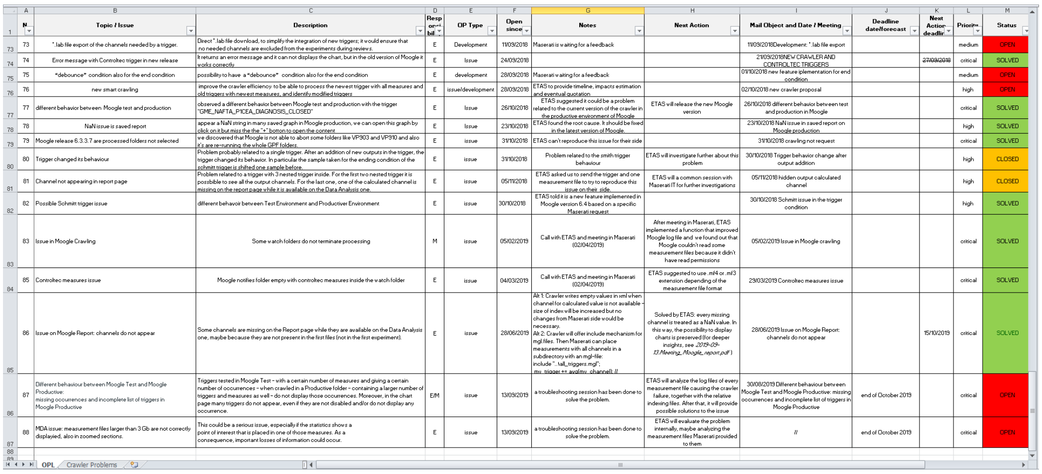
3. Meeting ETAS + Maserati: when Skype-based connections are established, communications starts and everyone tries to give support in the explanation and solution of a problem. Some particularly complicated problems could require two or more troubleshooting sessions, so that ETAS technicians can correctly address the problem and every possible cause can be excluded.

4. When the problem has been targeted, ETAS tries to provide a solution in the following Moogle releases, and users have to test new modifications. If they work, the OPL point is archived as *solved*, otherwise it is marked as *on hold* or still *open*.

These steps take place cyclically for anyone of the problems of the OPL. It is required to have particularly pronounced communication skills, to prevail in arguments and also to convince the other party to furtherly progress in the problem solution. Moreover, it is necessary to select carefully the order with which presenting the issues, because ETAS could decide to reject development purposes or to propose easier workaround that anyway do not resolve the problem. At the end of each meeting, the reporting `.pdf` of the meeting has to be drafted, describing carefully everything that has been said and the progresses that have been made, to keep trace of any progress. In this way, it is possible to bug-fix and improving the software, also improving the quality of the statistics and reducing the efforts needed to draft them. Furthermore, the two companies can cultivate their relationship, making it profitable and priority with respect to other players.

The OPL file (Fig. 3.17) is the container of any single improvement that has been officially recognized by both parties, and is needed as archive documentation of Moogle development. It is divided into eight subsections:

- *Topic/issue*: it contains a synthetic description of the issue or development purpose. It should be as much as possible comprehensive of all



N	Topic / Issue	Description	Responsibility	OP Type	Open since	Notes	Next Action	Mail Object and Date / Meeting	Deadline date/forecast	Next Action deadline	Priority	Status
73	*.lab file export of the channels needed by a trigger.	Direct *.lab file download, to simplify the integration of new triggers; it would ensure that no needed channels are excluded from the experiments during reviews.	E	Development	11/09/2018	Maserati is waiting for a feedback		11/09/2018Development: *.lab file export			medium	OPEN
74	Error message with Controltec trigger in new release	It returns an error message and it can not displays the chart, but in the old version of Moogle it works correctly	E	Issue	24/09/2018			21/09/2018NEW CRAWLER AND CONTROLTEC TRIGGERS		~~27/09/2018~~	critical	SOLVED
75	"debounce" condition also for the end condition	possibility to have a "debounce" condition also for the end condition	E	development	28/09/2018	Maserati waiting for a feedback		01/10/2018 new feature iplementation for end condition			medium	OPEN
76	new smart crawling	improve the crawler efficiency to be able to process the newest trigger with all measures and old triggers with newest measures, and identify modified triggers	E	issue/development	28/09/2018	ETAS to provide timeline, impacts estimation and eventual quotation		02/10/2018 new crawler proposal			high	OPEN
77	different behavior between Moogle test and production	observed a different behavior between Moogle test and production with the trigger "GME_NAFTA_P1CEA_DIAGNOSIS_CLOSED"	E	Issue	26/10/2018	ETAS suggested it could be a problem related to the current version of the crawler in the productive environment of Moogle	ETAS will release the new Moogle version	26/10/2018 different behavior between test and production in Moogle			critical	SOLVED
78	NaN issue is saved report	appear a NaN string in many saved graph in Moogle production, we can open this graph by click on it but miss the the "+" botton to open the content	E	Issue	23/10/2018	ETAS found the root cause. It should be fixed in the latest version of Moogle.		23/10/2018 NaN issue in saved report on Moogle production			high	SOLVED
79	Moogle release 6.3.3.7 are processed folders not selected	we discovered that Moogle is not able to abort some folders like VP903 and VP910 and also it's are re-running the whole GPF folders.	E	issue	31/10/2018	ETAS can't reproduce this issue for their side		31/10/2018 crawling not request			critical	SOLVED
80	Trigger changed its behaviour	Problem probably related to a single trigger. After an addition of new outputs in the trigger, the trigger changed its behavior. In particular the sample taken for the ending condition of the schmitt trigger is shifted one sample before.	E	issue	31/10/2018	Problem related to the smith trigger behaviour	ETAS will investigate further about this problem	30/10/2018 Trigger behavior change after output addition			high	CLOSED
81	Channel not appearing in report page	Problem related to a trigger with 3 nested trigger inside. For the first two nested trigger it is possbible to see all the output channels. For the last one, one of the calculated channel is missing on the report page while it is available on the Data Analysis one.	E	issue	05/11/2018	ETAS asked us to send the trigger and one measurement file to try to reproduce this issue on their side.	ETAS will a common session with Maserati IT for further investigations	05/11/2018 hidden output calculated channel			high	CLOSED
82	Possible Schmitt trigger issue	different behavoir between Test Environment and Productiver Environment	E	issue	30/10/2018	ETAS told it is a new feature implemented in Moogle version 6.4 based on a specific Maserati request		30/10/2018 Schmitt issue in the trigger condition			high	SOLVED
83	Issue in Moogle Crawling	Some watch folders do not terminate processing	M	issue	05/02/2019	Call with ETAS and meeting in Maserati (02/04/2019)	After meeting in Maserati, ETAS implemented a function that improved Moogle log file and we found out that Moogle couldn't read some measurement files because it didn't have read permissions	05/02/2019 Issue in Moogle crawling			critical	SOLVED
85	Controltec measures issue	Moogle notifies folder empty with controltec measures inside the watch folder	E	issue	04/03/2019	Call with ETAS and meeting in Maserati (02/04/2019)	ETAS suggested to use .mf4 or .mf3 extension depending of the measurement file format	29/03/2019 Controltec measures issue			critical	SOLVED
86	Issue on Moogle Report: channels do not appear	Some channels are missing on the Report page while they are available on the Data Analysis one, maybe because they are not present in the first files (not in the first experiment).	E	issue	28/06/2019	Alt 1: Crawler writes empty values in xml when channel for calculated value is not available - size of index will be increased but no changes from Maserati side would be necessary. Alt 2: Crawler will offer include mechanism for mgl.files. Then Maserati can place measurements with all channels in a subdirectory with an mgl-file: include "..\all_triggers.mgl"; my_trigger += avg(my_channel); //	Solved by ETAS: every missing channel is treated as a NaN value. In this way, the possibility to display charts is preserved (for deeper insights, see *2019-09-13.Meeting_Moogle_report.pdf*)	28/06/2019 Issue on Moogle Report: channels do not appear		15/10/2019	critical	SOLVED
87	Different behaviour between Moogle Test and Moogle Productive: missing occurrences and incomplete list of triggers in Moogle Productive	Triggers tested in Moogle Test - with a certain number of measures and giving a certain number of occurrences - when crawled in a Productive folder - containing a larger number of triggers and measures as well - do not display those occurrences. Moreover, in the chart page many triggers do not appear, even if they are not disabled and/or do not display any occurrence.	E/M	issue	13/09/2019	a troubleshooting session has been done to solve the problem.	ETAS will analyze the log files of every measurement file causing the crawler failure, together with the relative indexing files. After that, it will provide possible solutions to the issue	30/08/2019 Different behaviour between Moogle Test and Moogle Productive: missing occurrences and incomplete list of triggers in Moogle Productive	end of October 2019		critical	OPEN
88	MDA issue: measurement files larger than 3 Gb are not correctly displayied, also in zoomed sections.	This could be a serious issue, especially if the statistics shows a point of interest that is placed in one of those measures. As a consequence, important losses of information could occur.	E	issue	13/09/2019	a troubleshooting session has been done to solve the problem.	ETAS will evaluate the problem internally, maybe analyzing the measurement files Maserati provided to them	//	end of October 2019		critical	OPEN

Figure 3.17: Extract of the OPL file registering all the steps of Moogle development.

the aspects of the problematic, and should resemble in the best way the object of the e-mail blast. This is done to favour the retrieving of the problematic among e-mail conversations and internal documents.

- *Description*: contains a medium-length description of the problem.
- *Responsibility*: it contains the responsibility of the issue or development request, i.e. the subject — out of the two parties — that has or could have the major influence on the problem, or could be its source. The values can be "E" (if the responsibility is ETAS's), "M" (if it is Maserati's) and "E/M" if it is shared.
- *OP type*: it defines the open point classification. It could be "*issue*", "*development*", or also "*issue/development*" if the development request is particularly important for the correct utilization of the software, but anyway smaller than a real problem.
- *Open since*: it contains the date in which the point has been opened.
- *Notes*: this section contains more insights about the current status of the problem, usually defined during a meeting or an official mail exchange. This information is useful — especially when more than one meeting is needed to solve a point, or when the points to be discussed in a single meeting are many — to keep track of the advancement status of a problem coping.

- *Next action*: it contains information about the steps that are set to be taken towards the problem solution. It usually refers to the conclusions of the previous meeting or e-mail exchange.
- *Mail object & meeting date*;
- *Deadline date/forecast*: to have an idea on the time required to have a point solved. It usually depends — according to the responsibility — on the business or the complexity level encountered by the parties involved.
- *Next action deadline*: it refers to the deadline or forecast of the previous subsection.
- *Priority*: it defines the priority with which a point should be addressed. The values can be *low, medium, high, critical* according to the severity of the problem.
- *Status*: it defines the current status of an issue. The values can be:
  - *Solved*, if the request has been accepted, processed and solved.
  - *Partially implemented*, if the request has been accepted, but the solution of the problem has been quite different from the initial request. It could happen, for instance, when the needs of the software developers are somehow colliding with Maserati ones, or when the requests are not precise or not perfectly implementable.
  - *Open*, if the point is still to be processed or solved.
  - *On hold*, if the point is particularly labourious and require a longer processing time.
  - *Not accepted*, if the request cannot be implemented or goes against ETAS corporate standards.
  - *Closed*, if the point has been closed without a solution, or if the problem does not need a solution anymore.

Chapter 4

Evaporative Emissions Control System

After having discussed Moogle's outstanding potential in the analysis of Big Data coming from fleet vehicles' accumulation periods, in Chapters 4 and 5 a practical application, which I delved during my stage experience, will be illustrated. This *case study* is aimed at verifying the possibility to enhance a quantitative performance index (called IUMPR[1]) related to the Evaporative Emissions Control system (also known as EVAP system) of a vehicle, exploiting BD analysis. Before proceeding with the details of this insight, a deep description of the functionalities and the layouts of the EVAP system should be given.

Evaporative emission control systems were firstly introduced on passenger vehicles to decrease the amount of highly volatile, unburnt hydrocarbons (HC) released in the atmosphere. These HCs, pursuant of their high volatility, naturally evaporate from fuel system's components and the intake air system to the external environment, contributing to air pollution. Actually, the problem of HC release is more transversal and not just limited to the fuel system, since the origin of those particles can also be associated to other sources: from the vehicle's external plastic and rubber components to tires, brakes, plastic interior trims, carpeting, etc. For this reason, they are usually classified under the more general and comprehensive name of *volatile organic compounds* (VOCs). Although technologies for reduction and recuperation of these plastic-based substances are currently under research and development studies, devices for the collection of fuel system's evaporative emissions already exists, also because the percentage of VOCs that can be associated

[1]The acronym, which is visible at the corresponding Section of this document, indicates a measure of the *diagnosticability* of a variable in ordinary and standardized operating conditions.

to the evaporation from the fuel system is arguably the largest.

It is widely accepted in scientific world — already from the early '50s — that the combined action of sunlight on reactive hydrocarbons and nitrogen oxides (NOx) in the atmosphere is responsible for the formation of **low-altitude ozone layers**, which can cause a reduction in lungs functionality and an increase in respiratory symptoms [32]. In addition to the negative health effects of ground level ozone on health, it is also a greenhouse gas. For those reasons, the emission of unburnt HC must compulsorily be limited to very low or extremely low levels, to avoid as much as possible the urban smog formation.

Evaporative emissions can be classified according to their primary sources, which can be:

- *Diurnal emissions*, resulting from the evaporation of gasoline due to overnight temperature fluctuations;
- *Running loss emissions*, which represent gasoline that is vaporized from the engine and fuel system while in operation. Most of these HCs end up being captured by EVAP system and routed back through the engine intake to be consumed during combustion;
- *Hot Soak emissions*, occurring during the first hour of vehicle parking, after having experienced normal operation cycles. The introduction of fuel injection greatly reduced hot soak and running loss evaporative emissions, as they isolated the fuel from the atmosphere at all time except while refueling;
- *Permeation emissions* occur continuously once the polymer components of the fuel system become saturated with fuel. In this case, the impermeability of the tank is not 100% anymore;
- *Refueling emissions* occur as gasoline is pumped into the tank displacing the gasoline rich vapor. In modern passenger vehicles, these vapors are stored in the EVAP canister and purged into the intake air of the engine to be combusted.

The magnitude of these relative components depends greatly on the engine design, fuel delivery and application [21]. Permeation emissions are the most peculiar ones, since they require the adoption of fuel tanks, hoses, seals and gaskets completely realized with low-permeable and expensive polymers, that have to show very high stability over time and low aging subjection. The other contributions can be tackled with the usage of additional technologies, that nowadays equip almost all EVAP systems of every marketed vehicle.

4.1 EVAP system overview

In this Section, an overview of the evaporative emissions control system I dealt with is presented. As previously outlined, Maserati S.p.A. fleet vehicles include not only the whole Maserati vehicles portfolio, but also the current Alfa Romeo's one, and the EVAP systems' hardwares differ with the two brands. Alfa Romeo V6-powered vehicles, as well as Maserati vehicles — currently powered by V6 and V8 engines and controlled by Bosch ECUs — feature a semi-automatized EVAP system, provided with a self-diagnostic and self-calibrating module called DMTL[2]. On the other hand, GME-powered Alfa Romeo vehicles feature an articulated and diversified EVAP system, having a larger specific complextity than the previously-mentioned ones. Although V6 and V8 engines have two banks, i.e. more hardware to be housed, GME EVAP system contains all the nucleic concepts and hardware that is deployed in those systems, too.

The EVAP system that will be introduced in this Chapter and that is the reference for the explanation of the case study I have deepened is the GME's one (Fig. 4.1). In spot of the DMTL module, it includes several other components, like the Evap System Integrity Module (ESIM), the Canister Check Valve (CCV), and the Fuel Tank Pressure (FTP) sensor, which will be explained in the following Sections. As suggested, evaporative emission control systems have been introduced to be compliant with the current laws on pollutant emissions. This system is very versatile, since it allows not only the recuperation and usage of fuel vapors in naturally aspirated conditions, but also in **boost conditions** (i.e. turbo-compressor group working). As explained in the introduction to Chapter 4, the increase of fuel evaporation occurs majorly in the following circumstances:

- The fuel within the tank overheats due to a working temperature increase, either due to increasing environment temperature, or to increasing heat flux coming from hot components in the nearby (such as the exhausts pipes);
- The environment pressure decreases, for instance during mountain trips.

The vapor formation is not depending on the gear engaged and/or the vehicle speed, or the engine working point, since with direct injection systems there is not direct connection between the engine and the fuel supply system anymore. With reference to Fig. 4.1, fuel vapors are conveyed — through a venting tube — from the fuel tank to an **activated charcoal canister**, trapping vapors

[2]Diagnostic Module Tank Leak, ed.

within its porous pattern. The latter is usually housed above the right rear wheel arch, near the fuel tank and on the other side with respect to the refueling tube.

Within the fuel tank there are usually a certain number of rollover valves — placed in the highest point of the tank with respect to the ground — which prevent the liquid fuel from entering the canister through the venting tube. If this occurred, the canister functionality would be irreversibly impaired, releasing abrupt quantities of HC in the external environment. In addition, its immediate replacement would be compulsory, especially to avoid the ignition of accidental wildfires.

The carbon canister is directly connected to the intake manifold by means of a purge duct, whose flowrate is commanded by a solenoidal purge valve placed in between. Furthermore, the canister is also connected with fresh external air through a small air filter to facilitate the canister purging without allowing dirt and particles to enter. When the engine is running and its operating conditions are conducive, the ECU opens the purge valve through PWM[3] commands, putting in communication the partially-smeared canister with the intake manifold: since in the latter a depressure is experienced — due to the normal functionality of the engine — fuel vapors start flowing from the canister directly to the engine, being burned up as part of the aggregate air-fuel charge. The ECU manages this phase by adapting the duty cycle of the purge valve's solenoid on the base of the real-time canister's *recycle rate*: this variable is estimated by monitoring the variations of the lambda sensors indications, upstream the TWC. In this way, the right quantity of fuel to be drawned from the canister can be determined, hence the duty cycle of the purge valve. When fuel vapors enter in the cylinders, the quantity of injected fuel decrease accordingly, in order to maintain the air/fuel ratio around the stoichiometric value. Again, the quantity of not injected fuel is again determined on the base of the information coming from the pre-catalyst lambda sensors.

It is clear that, despite a small electric energy contribution needed to operate the purge valve, the EVAP functionality guarantees a significant decrease in fuel consumption, since it is completely reliant on favorable pressure gradients and since the injected fuel is periodically smaller than what it would be necessary in absence of this system. The quantity of fuel saved depends basically on:

[3]Pulse-Width Modulation (PWM) commands are square-wave input signals that are sometimes used to operate electronic devices. The amplitude and the frequency of each square wave's peak are decided on the base of a feedback monitoring of the information coming from other sensors and subsystems.

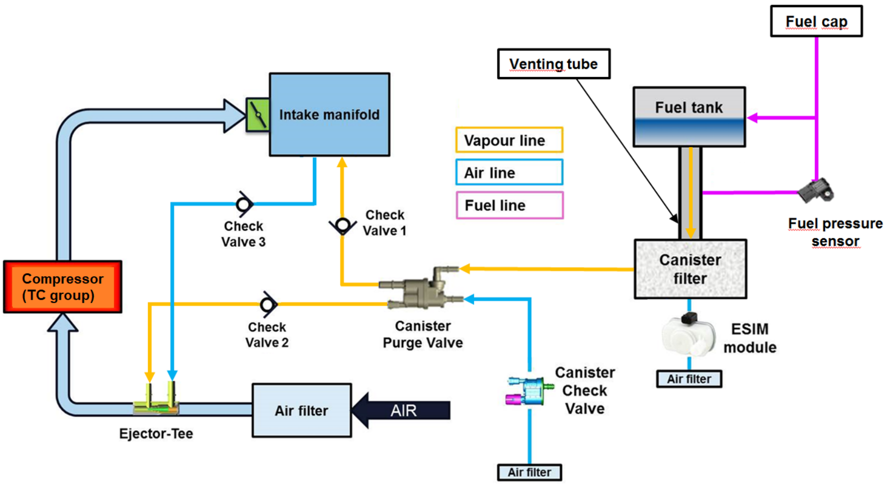


Figure 4.1: Representation of a standard EVAP system, comprising the *turbo* branch.

- The canister's *smearing rate*, which depends in turn on the operating temperatures of the fuel system, as well as the working capacity of the canister. Ultimately, it depends on the accuracy of the estimation model that exploits the indications of lambda sensors.

- The rate at which the boost conditions are met with normal driving cycles. This parameter depends also on the capability of the engine control software to recognize that the enable conditions have been reached.

The canister purging is subdued to **fault confinement** logic: in case of failures of the mixture preparation control systems, purging is disabled and no boost can occur. In case of particularly prolonged failures, canister purging would be interdicted accordingly, and canister smearing would continue without stops. If this uninterrupted smearing reaches a certain threshold, excess vapors start blowing from the ESIM module, hence determining the lighting of the MIL lamp.

As it can be seen from the above presented layout and discussion, the EVAP system is not only a passive storage system for collecting the fuel vapors that are escaping the fuel system, but can also be designed to send them through the engine's intake manifold. Demands on vehicle manufacturers to achieve higher fuel efficiency through the use of downsized engines and hybrid electric powertrains are creating challenging operating conditions for evaporative emission control technologies: the lower purge volumes resulting from smaller displacement engines or hybrid systems under partial or full electric drive require the development of special carbon adsorbents and advanced canister designs to achieve the lowest evaporative emissions demanded by current and future regulations. Today, EVAP system monitoring is mandatory in just NA and CN6B markets (i.e. North America and China, respectively), while in Europe the situation is not yet settled. With high probability, equivalent prescriptions will also be introduced there in the near future.

In the following subsections, the system behavior in boosted and naturally-aspirated conditions will be analyzed.

4.1.1 Naturally aspirated condition

When the power request is not so high, the kinetics of the exhaust gases is not enough to activate the turbo-compressor (TC) group. In these conditions, the engine works as if it was naturally-aspirated, and within the intake manifold a depressure is naturally experienced. Each time the purge valve is commanded, this depressure naturally purges the canister, redirecting fuel vapors from the canister to the intake manifold (yellow line, Fig. 4.2). Check valves 2 and 3 are closed, pursuant of the negative pressure gradient produced inside the intake manifold, hence the circuit involving the TC group is cut off.

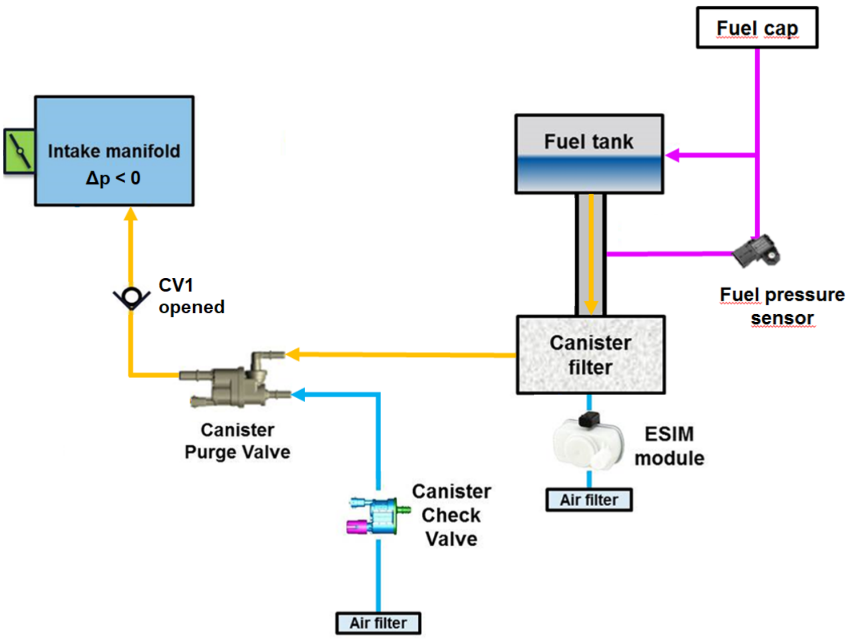


Figure 4.2: Simplified EVAP system layout with engine operating as it was naturally aspirated, only.

The reduction in the quantity of injected fuel is operated according to the monitoring of the pre-catalyst lambda sensors: when the purge valve is PWM-operated, the ECU is not instantaneously aware of the further fuel supply magnitude that is being conveyed to the intake manifold, therefore the engine starts burning a richer charge. As soon as this richer charge is burned and exhausted, the reduction in the exhaust gas oxygen upstream the catalyst will be notified by the local lambda sensor: this indication will be acknowledged by the ECU, which will command a reduction of the injectors' duty cycle to keep the stoichiometry of the charge. This control occurs with *negative-feedback* principle.

This operating conditions occur when the driving cycle does not provide a particularly high power request and the engine rpm are under a certain thresholds, for instance during urban driving. In these conditions, the canister purging is worse, and the risk of canister filling is higher.

4.1.2 Boost condition

When the power request is high enough, the turbo-compressor group can be activated with beneficial effects on engine's volumetric efficiency. With reference to Fig. 4.3, check valve 1 (CV1) is closed, pursuant of the positive pressure gradient created by the action of the TC group on the intake charge, while check valve 3 (CV3) is opened due to the overpressure it senses. As a consequence, the intake manifold is placed in communication with the ejector tee, which receives an overpressure upstream and returns a depressure at the outlet of the tube. In this way, a favorable pressure gradient is created with respect to the canister circuit side, and the latter can be purged. Since the ejector tee is placed in between the external air intake and the TC group, fuel vapors so collected are conveyed to the compressor and eventually forced into the combustion chamber. The duty cycle of the purge valve is determined as in the previous case, but canister purging is better because pressure gradients experienced are anyway larger than the ones experienced with the depressure in the bare naturally aspirated mode. Canister purging can be so efficient that the rate of evaporation and collection of fuel vapors inside the canister could be smaller than the effective purging rate: in this case it could happen that no fuel vapors — or at least a negligible amount — is induced to the intake manifold. What is purged, in this case, is simply fresh air coming from the canister check valve (CCV), which can communicate with the external environment.

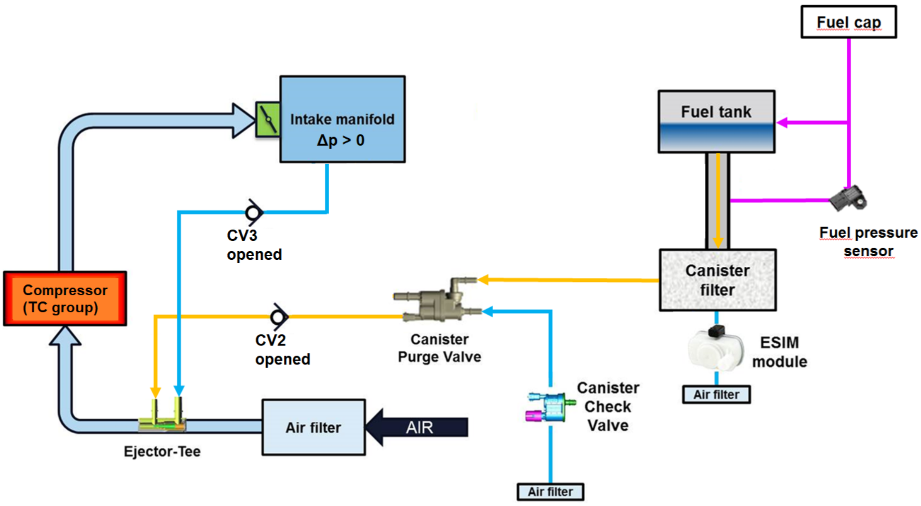


Figure 4.3: Simplified EVAP system layout with engine operating in Boost conditions.

4.2 EVAP system components

In this Section, a brief description of every component characterizing the EVAP system will be presented. The objective of this explanation is to better understand how the functionality and the physics of each components contribute to the functionality of the overall system.

4.2.1 Active charcoal canister

One of the essential components of the evaporative emission control system is the *active charcoal*, or *carbon* canister. The canisters employed on automobiles, and other gasoline-powered vehicles and equipment, are similar and consist of a plastic housing containing porous, high-surface area carbon adsorbent material. Thus, hydrocarbon molecules are attracted to the non-polar surfaces of the activated carbon and stored within the pores by **physical adsorption**. Carbon canisters can be realized in many shapes and sizes, which are proportional to the volume of vapor generated in the fuel tank: advanced canisters employ multiple chambers and specifically-designed carbon adsorbents to achieve very low or zero evaporative emissions, depending on the level of evaporative emission that must be achieved.

With reference to Fig. 4.4, HC vapors are forced out of the tank during heating or refueling, entering the first chamber of the canister. The second chamber communicates with the first in the downward region, hence the vapor collection surface can be increased. This phase is called *Adsorption* (orange arrow), and occurs continuously according to the vapor formation rate within the fuel tank.

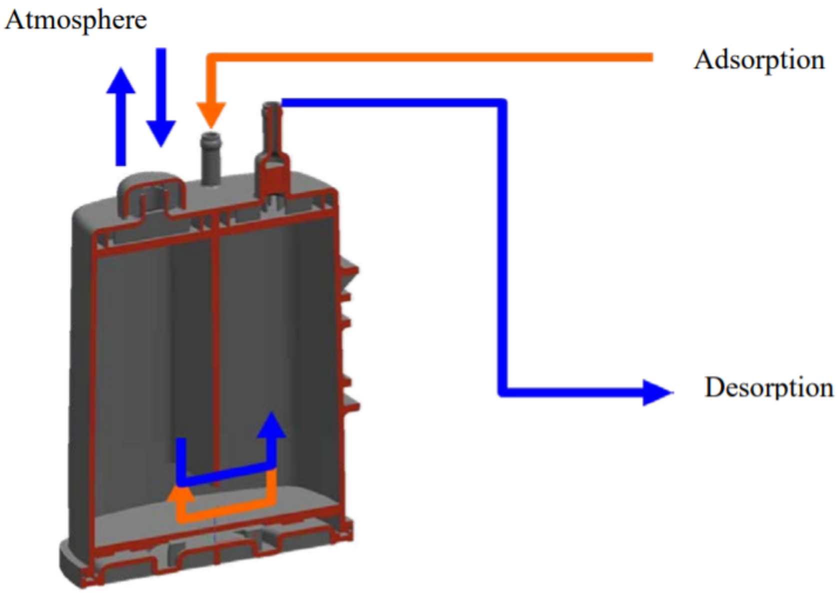


Figure 4.4: Internal surface of an EVAP canister without carbon adsorbent to show chambers's shape and adsorption/desorption process.

The second chamber can also communicate with the external environment through a calibrated, passive breather — i.e. the ESIM module, which will be described in Section 4.2.3 — which is designed to open when the canister pressure reaches a certain threshold. It is clear that canister breathing to the external air is an undesired condition, since it means to deliberately vent a certain amount of unburnt hydrocarbons in the atmosphere, therefore canister purging should be commanded in such a way that the breathing is avoided for as much comprehensive as possible ranges of driving conditions. Normally, before reaching the breather activation — i.e. before the complete canister filling — the purge valve intervenes and starts the Desorption process (blue arrow, Fig. 4.4): since a pressure differential downstream the canister has been created — regardless of the engine's current operating conditions — fuel vapors nestled in the charcoal matrix can be purged and conveyed to the engine cylinders. The most important parameters that characterize the canister design and can control its performance are:

- Cross-sectional area of the active charcoal body;
- Pressure drop determined by the canister presence;
- Length-to-diameter (L/D) ratio, which can be optimized to increase diffusion path length while minimizing pressure drops [33] or backpressure waves;
- Working capacity, i.e. the amount of hydrocarbons per unit mass that can be maintained within the activated carbon charge of the canister under standardized conditions test procedure. This parameter is also affected by the L/D ratio: increasing the latter, the canister purging is better, since a more effective removal of the adsorbed HCs is experienced.

Carbon canisters are very effective and extremely durable control technologies with little or no deterioration of performance over the full useful life of the vehicle.

Activated carbon

The core of any canister functionality is the activated carbon layer, which is housed inside the chambers of Fig. 4.4. This layer is available in different particle sizes and working capacities: the particle (or granule) size controls the backpressure events, whereas the working capacity — which describes, as explained before, the activation level of the carbon layer — is a function of surface area and porosity. Carbon porosity is described as a function of pore sizes (Fig. 4.5):

- *Micropores* (average diameter < 20 Å);
- *Mesopores* (average diameter in between 20-50 Å);
- *Macropores* (average diameter > 500 Å).

Vapor migration into the carbon matrix occurs via gas phase and surface diffusion of the hydrocarbon molecules, as it can be noticed from Fig. 4.5. Hydrocarbon molecules are driven to migrate and redistribute within the pores of the carbon by a combination of concentration gradient and surface energy. Another important property of the activated carbon is the *heel*, or the residual hydrocarbons remaining on the carbon after purging: the pore size distribution of the carbon directly affects both the working capacity and the heel of the carbon matrix. High working capacity is achieved by increasing the pore volume within a critical size range depending on the size of the hydrocarbon molecules being adsorbed, while a smaller pore size range is associated with the heel as pores of this size range trap the hydrocarbons and prevent them from being purged. Higher-activity carbons that have a high working capacity also have a more pronounced tendency for stronger adsorption or heel during purging: this can lead to higher diurnal emissions. Advanced carbon designs, which release HCs easily with a small volume of purge air, are best suited for downsized engines and hybrid powertrains.

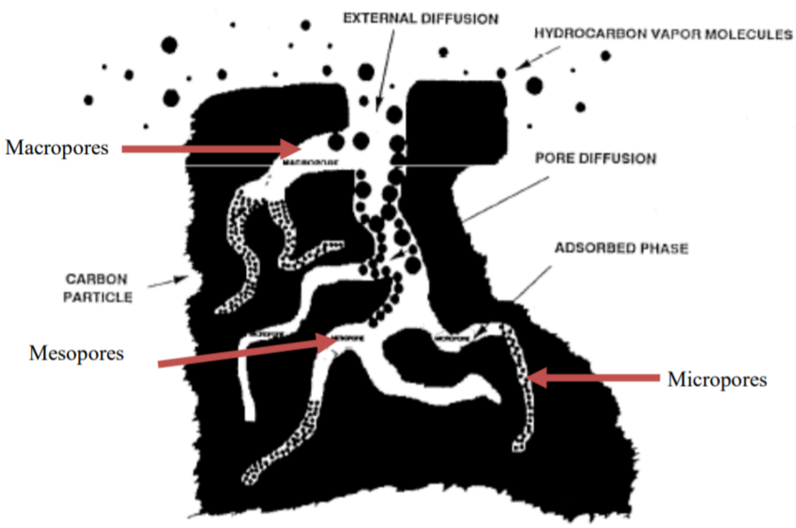


Figure 4.5: Pore types and diffusion mechanisms within an activated carbon particle.

4.2.2 Air intake system (AIS)

When the engine is shut off, the concentration of hydrocarbons in the cylinders and intake manifold is higher than the concentration upstream of the throttle body and air intake. In the absence of intake airflow, fuel vapors will migrate past the air induction system and into the atmosphere: these emissions are on the order of 0.1 g per day with some additional losses during hot-soak cycles [33]. Since very low levels of evaporative emissions are tolerated, these emissions would be detected in the SHED test[4] and must be

[4] *Sealed Housing Evaporative Determination* tests are performed to determine the quantity of evaporative emissions produced by a vehicle. As the name suggests, this testing method involves placing a vehicle or component in an enclosure and determining the level of emissions coming from it.

controlled. These slow-moving bleed emissions can be captured by incorporating a small hydrocarbon trap into the air induction system of the vehicle. The earliest designs utilized activated carbon within the air cleaner element, while in some applications, zeolite-based coatings have been applied to metal honeycomb substrates to control air intake evaporative emissions. Since these traps are not intended to capture a large amount of hydrocarbons, their working capacity is extremely low and they show very low pressure drops, too. Anyway, they are extremely effective in reducing the HC emissions, with efficiencies in capturing emissions from the air intake system about 90%.

4.2.3 Evaporative system integrity module (ESIM)

It is an on/off passive switch that opens when the pressure in the evaporative system is greater than the calibrated threshold of -2 $mbar$, to avoid the fuel tank collapse. It houses canting weights of different sizes (Fig. 4.6) to deal with both the overpressure and the depressure (or vacuum) events within the canister: the largest seals for pressure, while the smallest seals for vacuum (i.e. has better sensitivity for vacuum with respect to overpressure events). For this reason, it must be mounted vertically on top of the canister to avoid that the gravity action could impair the switch functionality.

This component has the following functionality:

- During **refueling** — or in general any other event that could determine a pressure increase, such as particular driving and/or environmental conditions aiding larger evaporations — vapor pressure from the canister pushes the heavier weight (i.e. the yellow component in Fig. 4.7a) off its seal and gas vapor can flow towards the external environment, through a remote air filter. In this way the canister is kept safe from overpressure events that could impair the structural compliancy of the component, as well as the chemical absorbency of the charcoal matrix.
- During **vacuum events** — for example while purging, or when the fuel cools during engine afterrun periods, or even when the external temperature decreases — the lighter weight is lifted off (Fig. 4.7b), allowing the entrance of fresh air through the remote filter from the external environment. In this way — also in case of heavy purging, where the air drawn from the canister may be larger than the evaporation rate of the tanked fuel — vapor drawing can occur at any rate, preventing canister crumpling.

In the former case, the pressure also pushes the diaphragm away from the electrical contacts upon which it normally lies. Therefore, since those electrical contacts receive a constant reference voltage from the ECU, in case of

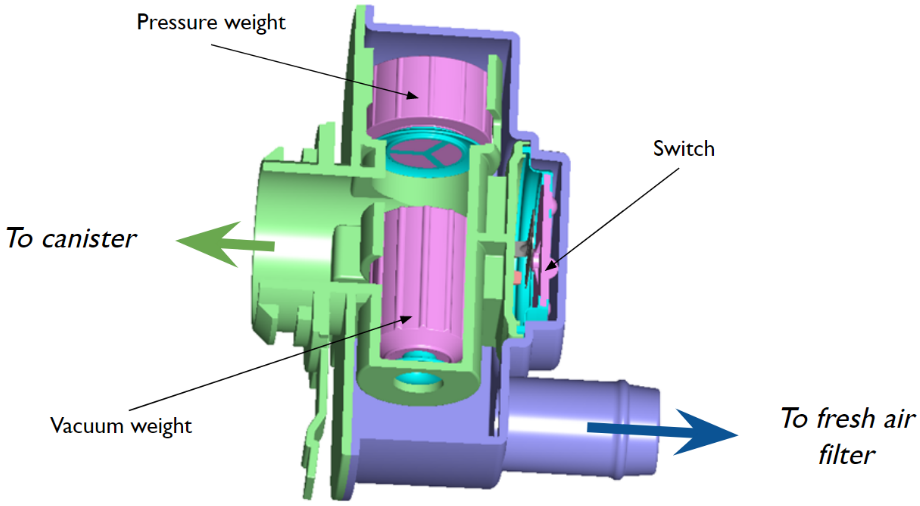
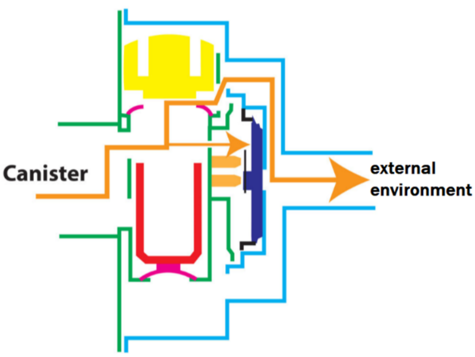
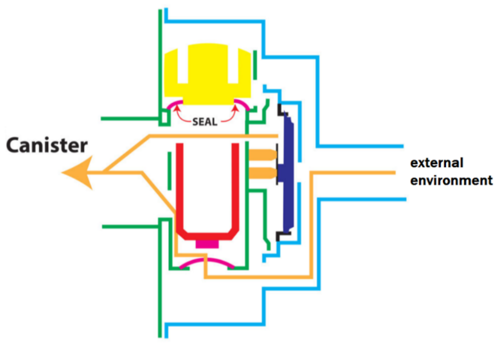


Figure 4.6: Illustration of a 3D ESIM switch.

overpressure of the system no voltage drops are evidenced, and no information can be subsequently retrieved. On the other hand, in the latter case the vacuum forces the diaphragm to close the electrical circuit, hence determining a current flow and a consequent voltage drop. This voltage drop can be sensed by the ECU to understand that the ESIM has opened, whose information can be exploited to acknowledge about the system's integrity.



(a) *Overpressure.*



(b) *Underpressure.*

Figure 4.7: ESIM switch functionality in overpressure and underpressure events.

4.2.4 Canister purge valve (CPV)

As explained previously, canister purging occurs during engine operations: in early automotive EVAP systems designs, it occurred passively thanks to the vacuum created in the fuel tank during cooling of the tank and fuel. In this way, the tank vacuum could pull clean ambient air through the canister causing desorption of the hydrocarbon molecules from the carbon surfaces to the tank. On the other hand, the vacuum created during engine operations could pull air through the carbon bed, resulting in desorption of HCs accumulated during fuel tank's evaporative emissions venting. Eventually, those vapor desorption was drawn into the intake of the engine, to have a further boost. Unfortunately, a sudden surge in HCs into the combustion chamber resulted in excessive hydrocarbons in the tailpipe, increasing the exhaust emissions. As a consequence, controlling the vapor concentration of the purging air stream has become necessary to meet both exhaust and evaporative emission standards. For this reason, the adoption of a purge valve (Fig. 4.8 – 2, Fig. 4.9), which could meter the purge flow, soon became a need: this valve, in fact, can restrict vapor flow from the fuel tank through flow-management orifices, to prevent large HC vapor transients from entering the intake of the engine, however allowing higher flowrates during refueling events.

In Fig. 4.9 it is possible to see a three-way purge valve: in the right part of the body there are the electric connection to the power plant and the access to the canister purge duct side. On the other side, there are the two access to the ejector tee and to the intake manifold, respectively.

The management of the canister purge valve is actuated through a solenoidal electric valve. The duty-cycle of this valve is a crucial parameter to determine the correct functionality of the EVAP system and depends, according to a very complex set of correlations, on many parameters:

- *Engine operating conditions*, i.e. intake and exhaust manifold pressures and temperatures, atmospheric pressure, environmental temperature, mass flowrate at the intake, angular speed working region, absolute position and phase of the cranks;

- *Engine working point*, for instance the engine mode (idle, running, cut-off, cranking, and so on) and/or precise values for macroscopic performance parameters. This information may impact on the natural or imposed pressure differential that comes downstream the purge valve, and may be useful to decide what could be a convenient duty cycle to be adopted to exploit thoroughly the onset conditions;

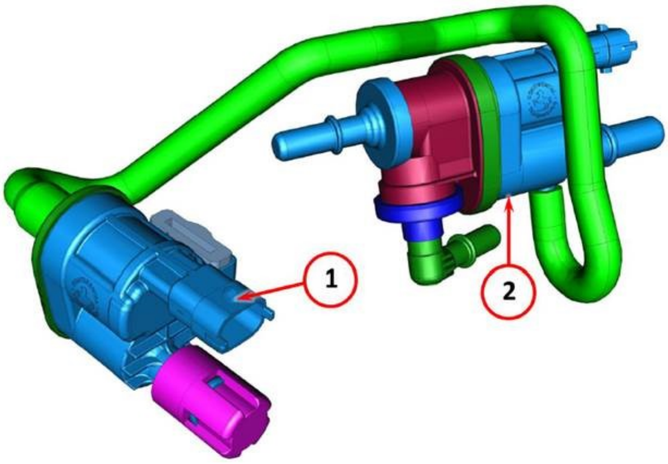


Figure 4.8: 3D Representation of a canister purge valve (2), with evidenced connection to the CCV check valve (1).

Figure 4.9: Footage of a real purge valve.

- *Lambda sensor information*, upstream and downstream the catalyst. This information is used to have a feedback on the intake charge title, considering both the contributions of the fuel injection and the fuel vapors coming from the canister purge. According to the results, the quantity of fuel to be injected is subsequently tailored;

- *Instantaneous volumetric efficiency*, to estimate how good is the engine replenishment at each instant. The better these values, the larger can be the duty cycle, regardless of all the other impacting parameters;

- *Canister instantaneous replenishment*: this value stem from a mathematical model, on the base of many instantaneous auto-diagnostic controls; it is one of the most important indicators to establish the average flowrate that should be delivered downstream the purge in each engine cycle. Clearly, the latter should be equal or at least comparable to the rate of canister replenishment, that in turn depends on the fuel system physics in relation to the contouring operating conditions;

- *Canister fuel contribution*, in terms of the quantity of fuel to be subtracted at the injecting fuel to keep the stoichiometry of operations;

- *Fuel rail pressures*;

- *Pedal position*, in terms of instantaneous power request;

- *ECU working state*;

- *Water circuit temperature.*

This list contains just a fraction of all the parameters that are actually accounted by the engine control software to calculate the proper purge valve duty cycle. Moreover, some of the cited parameters are measured by dedicated sensors, while some others are estimated on the base of experimental mathematical models. The grade of accuracy of each indicator is questionable, but has to be weighted in relation to its importance on the final output. Furthermore, some parameters of the list may be not completely independent one from the others, and some of them may have dependencies on other preliminary diagnoses that may have a bunch of different priorities. This reasoning has the aim of remarking that the control of the duty cycle variable is very far from being linear, but the fact that it involves a huge number of auxiliary parameters is an index of the accuracy with which the valve control is realized. To furtherly deepen the mechanism of evaluation of the purge valve duty cycle, please refer to [35].

Fig. 4.10 shows the duty cycle signal in while the engine is running, with no further special operating conditions are specified. It can be noticed that when the current flow is suddenly stopped, a remarkable voltage peak is induced across the ignition coil. The duty cycle here represented is around 30%.

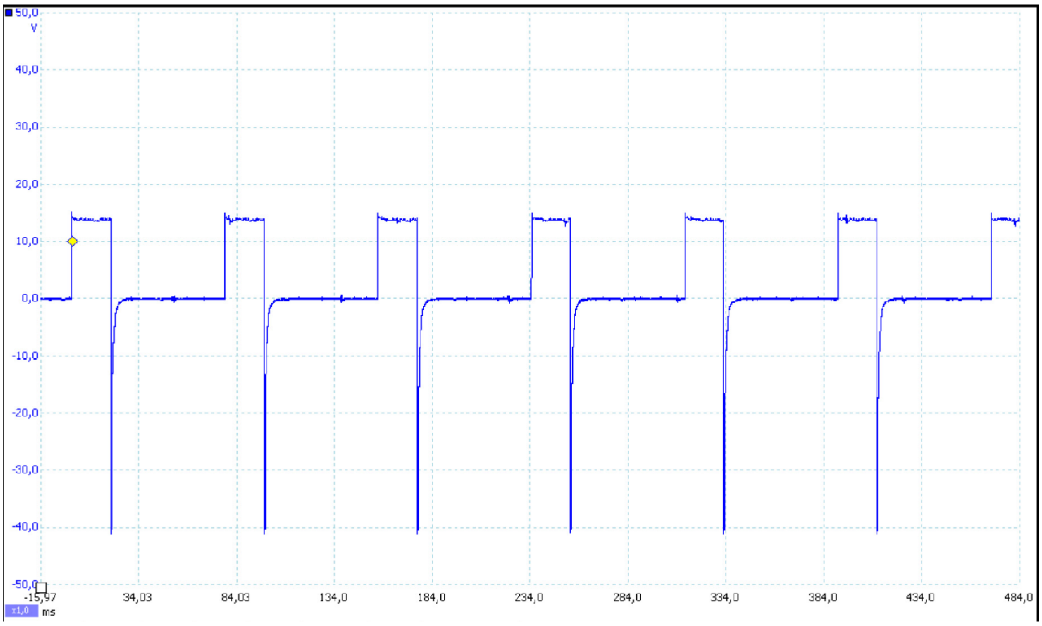


Figure 4.10: Voltage accross the purge valve versus the crank angle spectrum.

4.2.5 Canister check valve (CCV)

The canister check valve (Fig. 4.8 – 1, 4.11) — also known as OBD vent valve — is an active valve directly controlled by the ECU, sharing the same fuel vapors coming from canister's outlet with the evaporative emissions purge solenoid valve (Fig. 4.9). Due to this constructional layout, the CCV senses the same pressure (Fig. 4.3) that acts within the canister: in case of overpressure backflows from the purge — as it may occur in turbo-charged engines in boost conditions, when the purge valve is opened — it can vent the additional airflow to the external environment, avoiding dangerous flow recirculation into the canister. In this way, vacuum upstream the purge valve can be relieved in boost conditions; actually, this occurrence is extremely rare, because of the check valves (CV2, CV3) imposing a mono-directionality to the fuel vapor flow. A secondary function of the CCV is to permit the entrance of additional airflow in conditions where the purging duty cycle is larger than the capability of the fuel system to produce enough fuel vapors in the unit time. In those conditions, the fuel saving is close to zero, since the purge valve can only draw fresh air from the CCV and negligible amount of fuel vapors from the canister.

Figure 4.11: Continental's Canister check valve (CCV).

4.2.6 Ejector tee

In a turbo-charged powertrain system an alternative purge route exists in order to guarantee the canister purging even when the pressure inside the manifold is greater than the environmental one. To allow the correct functionality of this secondary circuit branch, a device aiding the realization of a favorable pressure gradient to the engine inlet must be employed. The ejector tee (Fig. 4.12) has exactly this task: it basically works thanks to the pressure differential between the inlet and the outlet of the compressor when it is spinning. The green cap is needed to avoid overpressure events, hence undesired recirculation, in the purge duct.

Thanks to the **Venturi duct** inserted within one of the ejector tee branches, in front of an overpressure coming from the engine inlet in boost conditions, a depressure is created downstream the Venturi, hence the backwash of the purge duct can be realized. The outlet of the ejector tee, which is installed directly upon the AIS, is the air intake duct itself, so that fuel vapors can be easily conveyed (Fig. 4.13).

The figure above shows the simplicity and at the same time the geniality of this component, which exploits the overpressure downstream the compressor to realize a favorable pressure gradient in the air intake system, downstream the air filter.

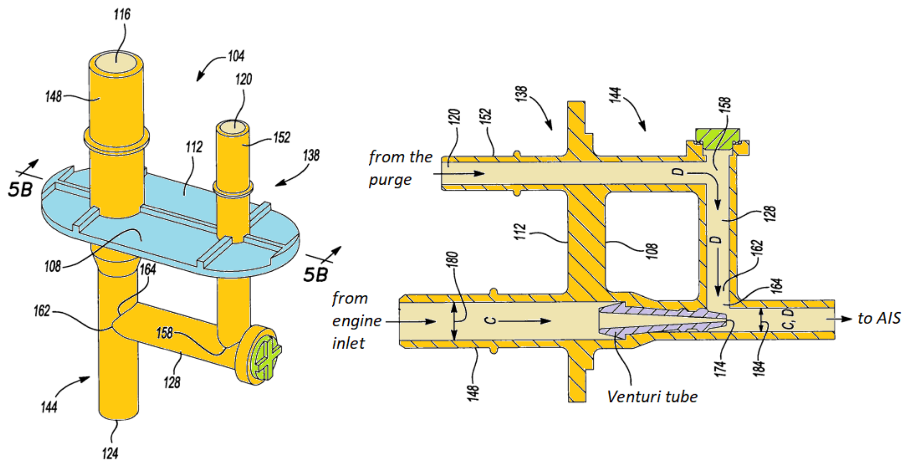


Figure 4.12: Technical drawing of an ejector tee.

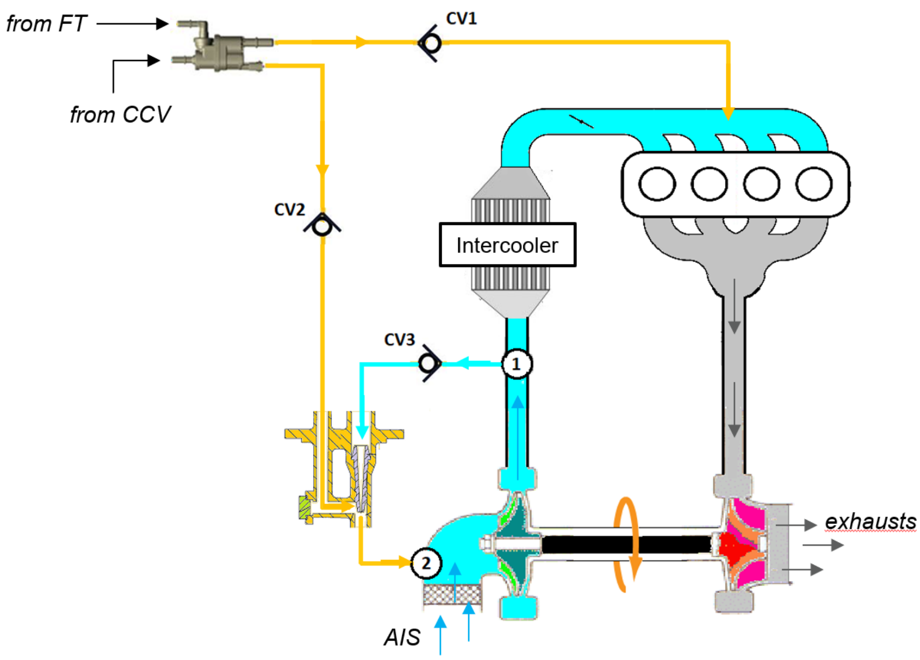


Figure 4.13: Layout scheme of the ejector tee and its role in the evaporative system functionality.

4.2.7 Fuel tank pressure sensor (FTPS)

This sensor is located directly upon or in the nearby of the fuel tank, and detects the pressure inside the EVAP system components to which it is connected. This sensor is obviously electrified, and it plays a fundamental role in the major diagnostic indicators of the EVAP system. Since it senses the pressure of the entire system, in fact, the information it provides can allow to determine if:

- The engine is in boost conditions: a steep depressure with respect to the ambient pressure is experienced;
- There are blocks in any circuit branches: overpressures of different intensities — according to the exact place in which the block occurs — are detected;
- There are leaks determining a poor efficiency in holding fuel vapors within the EVAP system: constant or abrupt loss of vacuum are detected. This former indication is one of the most important, because it allows to establish the compliancy of the system at the CARB level.

Figure 4.14: Fuel tank pressure sensor (FTPS).

Chapter 5

In-Use Monitoring Performance Ratio

It is already known that the study of vehicle emissions during a certified and standardized driving cycle (e.g. the older NEDC or the newer WLTC) is certainly a way to quantify the environmental impact of the overall technologies installed on the vehicle, but it does not say anything about its potential emissions during ordinary *in-use* maneuvers. In fact, the same attribute that makes those driving cycles and test procedures so powerful — i.e. their intrinsic standardization, which allows to make scientific and reliable comparisons between completely different vehicle brands, powertrains, weights, etc. — it is also their biggest limiting factor, because they are easily characterizable and entice engineers to work around them. For these reasons, OBD-monitored variables need to run frequently in-use to ensure that emission-related malfunctions are continuously diagnosed at specified time intervals. This operation needs the definition — for any major monitor, rationality and functionality that are recognized and legislated by emission authorities — of an index to quantify the diagnoses completion frequency. This index actually exists, it has become mandatory with OBD-II legislation, and is called IUMPR (In-Use Monitoring Performance Ratio). It is simply a rational number defined as:

$$IUMPR = \frac{N_{monitoring}}{N_{standard}}[.] \tag{5.1}$$

Where:

$N_{monitoring}$ is the number of times that a vehicle has been operated in such a way that the enable conditions for the diagnosis of a variable have been met;

$N_{standard}$ is the number of times that the vehicle has been operated in standardized and easily reproducible driving conditions. In this way, a reference to evaluate the frequency of a specific diagnosis is provided.

With so defined quantities, IUMPR can be considered as an index of the **diagnosticability** of a certain indicator, with respect to ordinary driving cycles: if it is low, it means that it is difficult to run the diagnostic of a certain monitor in ordinary in-use conditions, while if it is high it means it is easier. As said, rate-based monitoring is required for all diagnostics with the exception of CAN signal checks and electrical circuit checks such as short-to-battery, short-to-ground and open circuit. To be compliant with CARB and OBD norms, a minimal monitoring frequency of all the required diagnostic indicators — i.e. a minimal IUMPR — has to be achieved and certified by the regulating entities. This index is continuously calculated and stored by the ECU in its internal memory, where it can be retrieved through OBD ports.

Therefore, IUMPR index can be considered a summary verification of the overall diagnostic performance of a vehicle's OBD-II system. Although the minimal IUMPR requirements are different for any OBD-monitored variable, some global limits exist: for instance, the newest OBD regulation currently requires the monitoring frequency of at least 0.100 for all monitors, which means that the OBD monitor must run and complete on 1 out of 10 CARB-defined driving cycles. These driving cycles are characterized by a set of specified enable conditions — that uniquely characterize an in-use driving cycle, and which are specified in the denominator definition — that must be met in order to start the evaluation of the IUMPR index. Today, some substantial correction have been proposed to furtherly reduce the environmental impact of vehicles, starting from 2022: for example, based on IUMPR data collected from heavy-duty engines — which are amongst the most-emitting machines — a new minimum global IUMPR of 0.300 has been proposed, i.e. the 30%.

5.1 IUMPR Numerator specifications

As previously said, the numerator is defined as a measure of the number of times a vehicle has been operated such that all monitoring conditions necessary for a specific monitor to detect a malfunction have been encountered [39]. The specifications for incrementing are:

1. The numerator, when incremented, shall be incremented by an integer of one. The numerator may not be incremented more than once per driving cycle.

2. The numerator for a specific monitor shall be incremented within 10 seconds if and only if every monitoring condition necessary for the monitor of the specific component and/or a functionality has been satisfied. Obviously, the enable conditions can be different for each parameter to be monitored, both in type and in number.

3. For monitors that run or complete during engine-off operation, the numerator shall be incremented within 10 seconds of engine start on the subsequent driving cycle if all conditions described above have been satisfied.

Therefore, the numerator can be considered an index of how frequently the diagnostic conditions are established, on the base of the ordinary usage of the vehicle. Since this number should be — as requested by authorities and for benchmarking reasons — as high as possible, the car maker is interested in designing its vehicles so that the monitoring conditions are more frequently established.

5.2 IUMPR Denominator specifications

The denominator is defined as a measure of the number of times a vehicle has been operated as defined below:

1. The denominator, when incremented, shall be incremented by an integer of one. The denominator may not be incremented more than once per driving cycle.

2. The denominator for each monitor shall be incremented within ten seconds if and only if the following criteria are satisfied on a single driving cycle:

   (a) The cumulative time since the engine or propulsion system has been started must be greater than or equal to 600 seconds, in conditions of elevation of less than 8,000 feet above sea level and at an ambient temperature of greater than or equal to 20 degrees Fahrenheit;

   (b) The vehicle must be operated at a speed equal or above 25 miles per hour for at least 30 seconds, with the same specified environmental conditions of point 2a;

   (c) The vehicle must perform an idle period (i.e. with the accelerator pedal completely released and at a speed less than or equal to one mile per hour) for at least 30 seconds, in the same environmental conditions specified at point 2a;

(d) For hybrid vehicles, the cumulative fueled-engine operation must occur for at least 10 seconds, in the same environmental conditions specified at point 2a;

It is easy to see how general and comprehensive are the conditions to increment the denominator. Since the denominator should provide a reference to discern what are in-use driving conditions, this characteristic is wanted because is sufficiently transversal and allows to assume that the vehicle is running whichever kind of trip.

5.3 Disablement of Numerators and Denominators

The disablement of Numerators and/or Denominators should occur whenever a malfunction — capable of disabling the monitor of a certain indicator — is detected. It shall occur within ten seconds since the malfunction detection (i.e. pending/confirmed fault code stored), and in these conditions the OBD-II system disable further increments of the corresponding numerator and denominator for each monitor that is disabled, within a driving cycle. When the malfunction is no longer detected (i.e. the pending code is erased through self-clearing or through a scan tool command), incrementing of all corresponding numerators and denominators shall resume within ten seconds. Furthermore, the OBD-II system shall disable further incrementing of all numerators and denominators within ten seconds if a malfunction of any component involved in the determination of the conditions for the incrementing of the denominator has been detected, and the corresponding pending fault code has been stored. In this way, the Standard Conditions for the incrementing of both the numerator and the denominator are no longer met because of a fault in the OBD-II system. Anyway, the incrementing of all numerators and denominators shall resume within ten seconds when the malfunction is no longer present (e.g., pending code erased through self-clearing or by a scan tool command).

5.4 High flow purge valve monitoring

The high flow purge valve monitoring — also addressed as *EVAP in Boost*, since it monitors the EVAP system branch working during in-boost engine conditions — checks the fuel vapor contribution coming from the canister purging while the turbocharger group is actually compressing the air upstream the intake manifold, as a result of a specific power request from the driver. In these conditions, the purge valve allows the passage of fuel vapors from the canister to the ejector tee — as illustrated in Fig. 4.3 — which realizes a favorable pressure gradient to draw anyway the vapors into the intake system. Since this functionality of the EVAP system can impact on the pollutant and greenhouse emissions, it is precisely regulated by the OBD-II standard and therefore undergoes to subsequent in-use requirements, collected in a dedicated IUMPR index. Following the definition of IUMPR given in Chapter 5 introduction, it can be considered a measure of how much frequently a vehicle with EVAP in boost feature is operated such that the *enable conditions* for the high flow purge valve monitoring are met, over standardized in-use driving conditions. To achieve perfected emission reduction, the higher this number, the better.

During accumulation periods, miscalculations of EVAP in boost IUMPR index were noted: unrealistic and exaggerated values were calculated by the ECU during urban driving cycles. Urban cycles are considered to be the worst case from the point of view of diagnosticability, since the medium experienced accelerations and speeds are quite low and do not involve a particularly pronounced power request. In fact, just particularly heavy accelerations could be prodromal of boost onset, or at most a remarkable speed achieved with the lowest gears engaged. This happens because the sensitivity of the whole system is very low: it is necessary that the engine develops continuous, stabilized and sufficiently large manifold overpressures to actually see some purging from the canister. This poor sensitivity is to be attributed to the physics of the system (e.g. ejector tee and FTPS position, low canister volumes, and so on). Moreover, the frequent Stop-and-Start (S&S) experienced in high traffic conditions make more difficult to operate the vehicle in the ways provided by the standardized in-use driving conditions, influencing denominator increments.

To be not only compliant with the authorities' requests, but also to be competitive during benchmarking, it is necessary to adapt the engine control software to recognize properly each of the operating conditions involved in the evaluation of the IUMPR index, with the aim of achieving as high as possible results.

5.4.1 Estimation of increments durations

The first request that I had to settle was to make a statistic about how many times the denominator incremented before the numerator, from the start of the driving cycle. Each driving cycle is considered to be starting when the acquisition system begins recording, usually before the engine is switched on, and ends when the driver stops recording, usually after at least 30 min. Normal customer driving cycles are simply delimited by the time interval between the engine starting and the engine shut down. Since the measured IUMPR index were too high with respect to normally outcoming values, the principal suspect was that the engine control software did not recognize that the settling conditions for the denominator increase had been actually achieved. The trigger that has been implemented to this aim has the following logic:

```
EXT_TRIGGER := detect/time {KEYON\ETK = 1 AND RPM\ETK > 0};
EXT_TRIGGER += delta(time);
EXT_TRIGGER += {
    NESTED_TRIGGER_1 := detect/time {KEYON\ETK = 1; diff(IUMPR_NU-
                        M\ETK) > 0};
    NESTED_TRIGGER_1 += delta(time);
    NESTED_TRIGGER_1 += delta IUMPR_DEN\ETK;
    NESTED_TRIGGER_1 += start_value ODOMETER\ETK, RPM\ETK;
    NESTED_TRIGGER_1 += end_value ODOMETER\ETK, RPM\ETK;
    };
```

- In the external trigger (`EXT_TRIGGER`), the condition for targeting each trip is simply key inserted (`KEYON\ETK = 1`) and engine rotational speed larger than zero (`RPM\ETK > 0`): in this way, we are sure to be in conditions of engine running. The output `delta(time)` is placed as a redundancy check to verify — during data analysis phase — if the engine running windows have been correctly targeted.

- The internal trigger (`NESTED_TRIGGER_1`) detects all the windows opening in the same instant of the mother trigger and closing when the numerator increments: in fact, the syntax `diff(IUMPR_NUM\ETK) > 0` means that the difference between the value presently assumed by `IUMPR_NUM\ETK` and its preceeding one must be larger than zero, which is a workaround to say that the numerator has incremented.

If the two conditions have been effective, the windows they have triggered will be as represented in Fig. 5.1:

While the external trigger would have detect the whole window of Fig. 5.1, pursuant of its comprehensive conditions, the internal window would have picked up windows starting from the origin of the trip (cursor 1 in red), to the instant in which the numerator (green signal) increments (cursor 2, red).

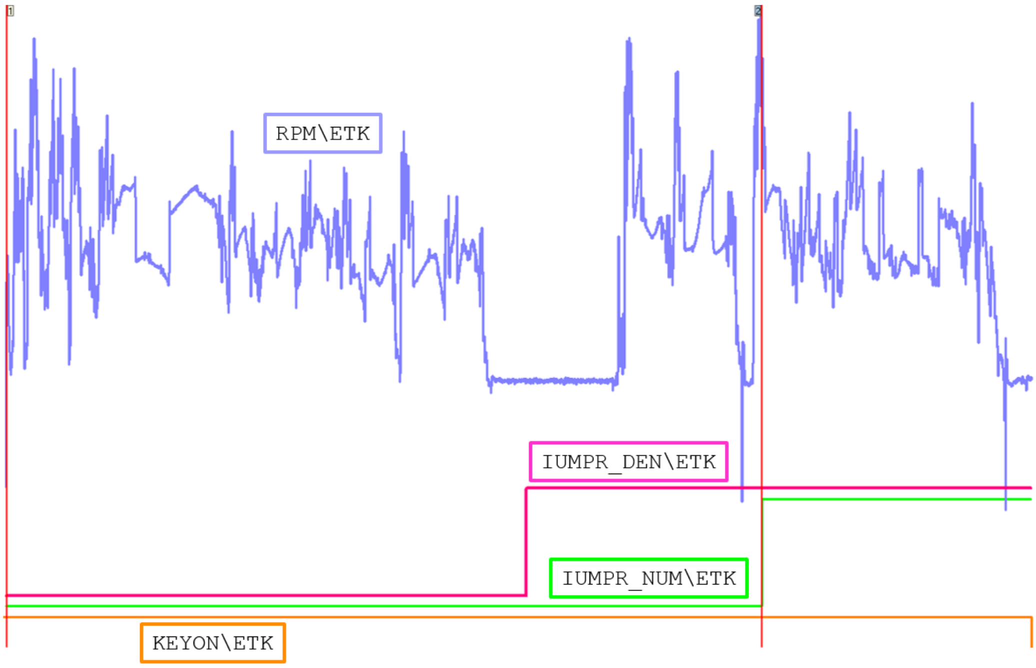


Figure 5.1: MDA representation of the window triggered by Section 5.4.1 trigger.

The most important output of the internal trigger is `delta IUMPR_DEN\ETK`, which measures the difference between the value the denominator assumes at the end of the triggered window (`end_value`) and at the start of the same window (`start_value`). Since — as specified in Sections 5.1, 5.2 — numerators and denominators can only increment of a unit in a driving cycle, `delta IUMPR_DEN\ETK` can only assume two values: zero and one. To be sure that the denominator has incremented before the numerator, `delta IUMPR_DEN\ETK` should be equal to one, otherwise it would be zero.

This trigger has been crawled with three different set of measures — corresponding to three different software updates, proposed and flashed on fleet vehicles in chronological order — of a single vehicle. The results of this prior statistic are displayed in Fig. 5.2: for confidentiality reasons, the names of the vehicle and of the engine control softwares have been hidden. From now on, the latter will be named software 1, software 2 and software 3, in chronological order from the oldest to the newest.

The three charts above show the value of `delta IUMPR_DEN\ETK` as a function of a temporal axis (in this case the odometer), although the stepping is not shown. As it can be noted — although the occurrences are too few to propose a rigorous statistical inference — the major indication is that

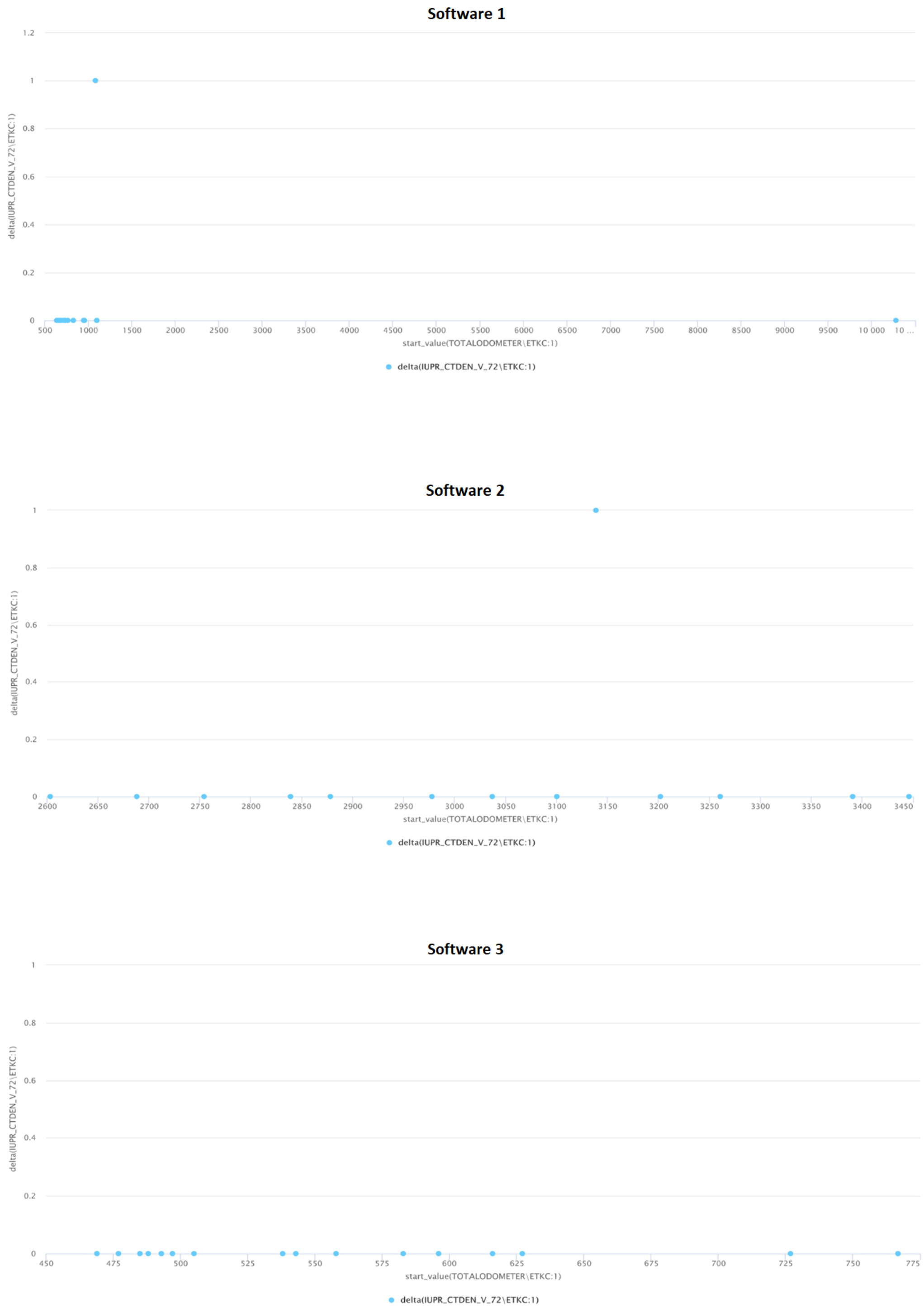


Figure 5.2: Results of Section 5.4.1 statistic, divided by software release.

the numerator increments almost always before the denominator: in fact, the statistic shows just two cases — over all softwares measures — in which `delta IUMPR_DEN\ETK = 1` (i.e. the denominator has incremented before the numerator, as shown in Fig. 5.1). This tendency induces the calculation of unbalanced values of IUMPR, which cannot cope with the authorities requirements.

5.4.2 Calculation of average durations

Once the main suspect had been confirmed, the second step was to calculate the average duration that the increments of numerator and denominator were involving and — secondly — to understand what could have been the causes for late increments of the denominator. To this aim, the conditions for the incrementation of the denominator are here exposed:

- The conditions for the incrementing of the **general denominator**[1] are the same exposed in Section 5.2. All those ones have to be satisfied, together with the following special conditions, related to EVAP in boost conditions:
  - The turbo-compressor group is operated at least twice — each time for at least 2 seconds — in a single driving cycle;
  - The turbo-compressor group is operated for at least 10 seconds in a single driving cycle.

It is necessary that at least one of the two special conditions verifies, to make the ECU considering that the special conditions for the denominator incrementing are satisfied as a whole. If both the special and the general denominator conditions are satisfied, the EVAP in Boost denominator can increment. Since the special conditions are very easy to be reached, both in urban and extra-urban driving cycles, they have not been considered responsible of the denominator delay. Hence, the focus of the analysis has been shifted to the monitoring of the times that the ECU spends in recognizing that the verification of the general denominator conditions have been actually occurred.

The information about the status of the general denominator is mirrored on a *bitmask* variable. In few words, a bitmask is a byte (i.e. a string of eight bits, from bit 0 to bit 7) in which every bit carries the information about

[1]The *general denominator* is the standardized set of driving conditions explained in Section 5.2. Together with fulfilling them, each monitor should fulfill another set of additional conditions (called *special conditions* or *enable conditions*) before it can increment.

the verification of a peculiar condition, specific of the single bit. In case the assigned condition is verified, the bit assumes the value 1 (i.e. TRUE), otherwise it assumes the value 0 (i.e. FALSE). Each time a bit becomes TRUE, the value it assumes (i.e. 1) is multiplied for the quantity $2n$, where n is the bit's relative position within the bitmask. As a consequence, each time a condition assigned to a bit of the bitmask verifies, the bitmask value is incremented of the result of the multiplication outlined above.

On the base of the bitmask logic, which is not here reported for confidentiality reasons, the matching between the channel `IUMPR_GENDEN` increments and its enable conditions are listed below:

- Increments of 1: the condition 2a (i.e. engine running time) reported in Section 5.2 is verified;
- Increments of 2: the condition 2c (i.e. vehicle speed) reported in Section 5.2 is verified;
- Increments of 4: the condition 2b (i.e. idle times) reported in Section 5.2 is verified;
- Temperature and altitude conditions are considered to be always verified, since they are very comprehensive.

To calculate both the incrementing times of numerator and denominator, and the ones of the general denominator conditions, the following trigger has been implemented:

```
EXT_TRIGGER := detect/time {KEYON\ETK = 1 AND RPM\ETK > 0};
EXT_TRIGGER += delta(time);
EXT_TRIGGER += {
    NESTED_TRIGGER_1 := detect/time {KEYON\ETK = 1; diff(x\ETK) >
                        0};
    NESTED_TRIGGER_1 += delta(time);
    NESTED_TRIGGER_1 += delta IUMPR_NUM\ETK;
    NESTED_TRIGGER_1 += start_value ODOMETER\ETK, RPM\ETK;
    NESTED_TRIGGER_1 += end_value ODOMETER\ETK, RPM\ETK;
    };
EXT_TRIGGER += {
    NESTED_TRIGGER_2 := detect/time {KEYON\ETK = 1; diff(IUMPR_GE-
                        NDEN) = y};
    NESTED_TRIGGER_2 += delta(time);
    NESTED_TRIGGER_2 += start_value ODOMETER\ETK, RPM\ETK;
    NESTED_TRIGGER_2 += end_value ODOMETER\ETK, RPM\ETK;
    };
```

Where:

- `EXT_TRIGGER` is exactly the same trigger exposed in Section 5.4.1, as well as `NESTED_TRIGGER_1`. The difference of the latter with its corresponding of Section 5.4.1 is that in spot of `IUMPR_NUM\ETK` the parameter **x** has been inserted. In place of this variable, the channel of the numerator and the denominator shall be inserted, respectively. In this way, with the same trigger, the time elapsed between the start of the trip and the instant in which they increment can be retrieved.

- `NESTED_TRIGGER_2` has the same logic of `NESTED_TRIGGER_1`, with the difference of having the increments of the general denominator bitmask (`IUMPR_GENDEN`) as window closure condition. In spot of the parameter **y**, the values of the increments (1, 2, or 4) described at the beginning of this Section shall be substituted. In this way, with a single trigger, three different incrementing times can be calculated.

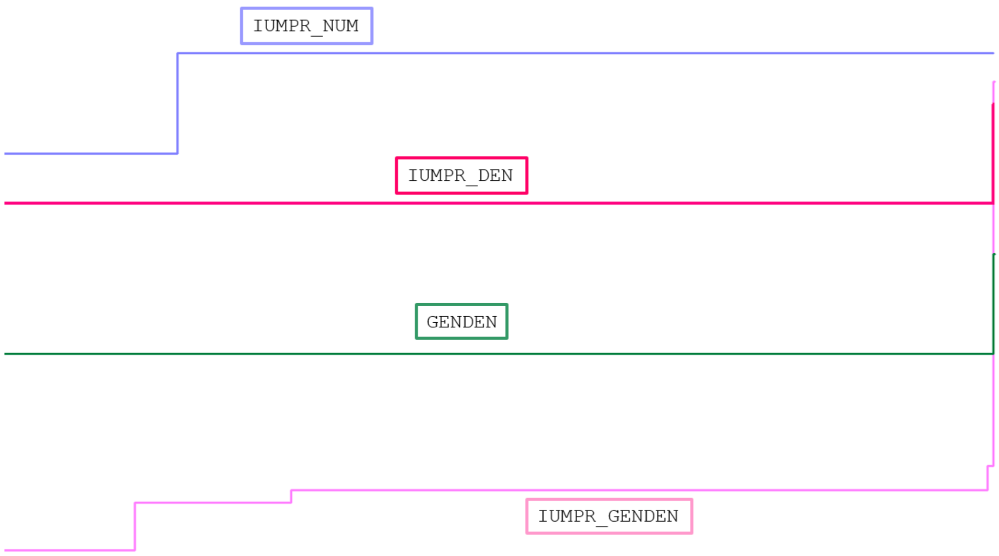
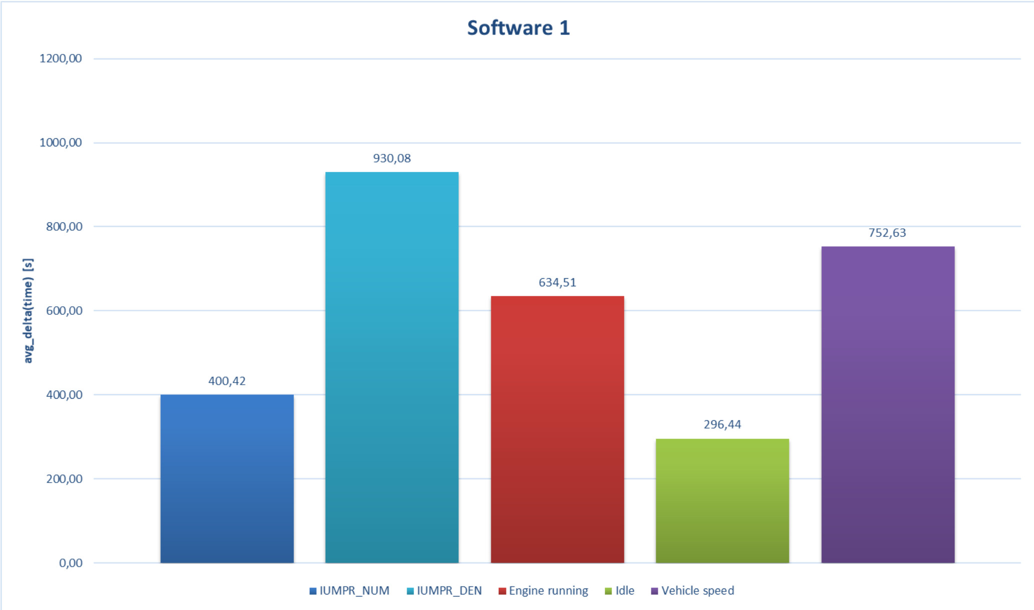
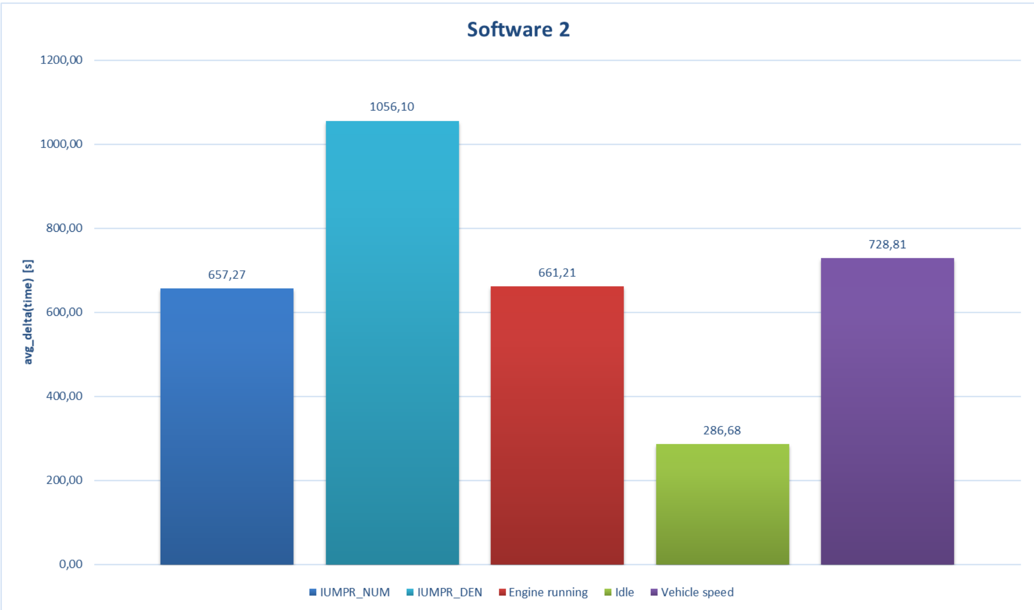


Figure 5.3: MDA representation of the window triggered by Section 5.4.2 trigger.

Fig. 5.3 shows a driving cycle in which anyone of the general denominator conditions has incremented, as well as the EVAP in Boost numerator and denominator. It is interesting to point out that the EVAP in Boost denominator increases *synchronously* with the general denominator: this means that the special conditions were already met, as imagined previously.

The so calculated five indicators increment with different delays with respect to others, moreover in each trip the situation may change. Therefore, to aggregate and give meaning to data, the average of those delays have been calculated for any specific index. The results are outlined in the figures above (Fig. 5.4):





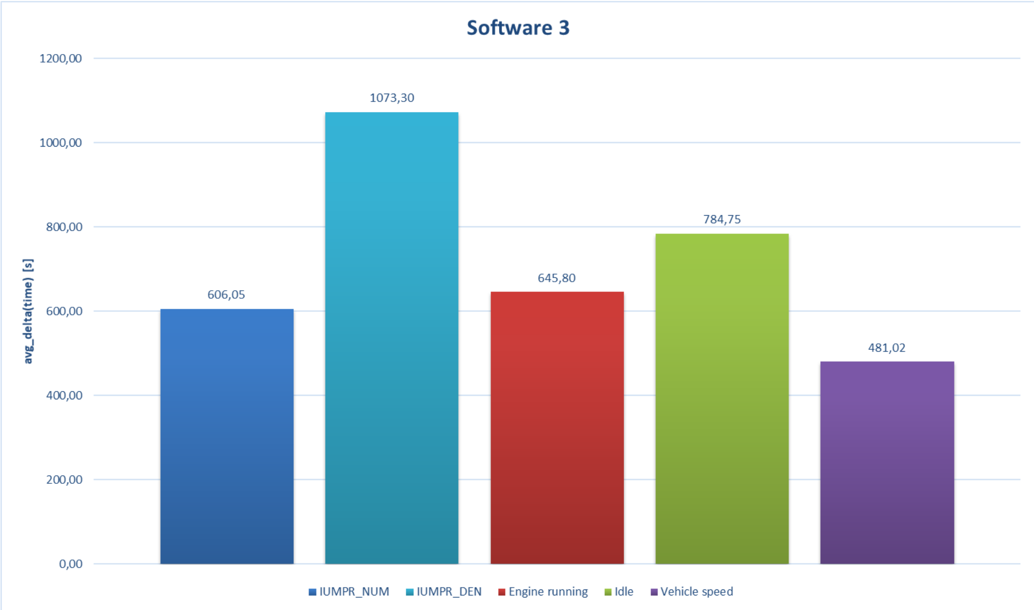


Figure 5.4: Bar charts representing the average time (in seconds) spent by each indicator for incrementing, divided by software.

As the results suggest, all indicators keep their delay more or less constant with each software development status, except the **idle** which visibly increments. It shall not surprise that such a pejorative behavior is experienced in the latest among those softwares: it is a common occurrence to find unexpected issues and detrimental collateral effects in the attempt to improve the behavior of another variable. This is why software updates should be continuously validated and calibrated, in order to obtain the desired set of behaviors without impairing any parameter. However, the charts contain two other interesting indications:

- The numerator is far faster than the denominator: this means that it is (on average) easier to reach the engine boost conditions than reaching the standardized conditions specified by the general denominator. This is fair because it suggests a high diagnosticability of the boost condition.
- The vehicle speed conditions, although quite variable, does not represent a determinant parameter in the denominator increment.

5.4.3 Idles misdetection

Once stated that idle periods are the most retardant factors among IUMPR denominator general conditions, it was necessary to have further insights on their detection mechanisms and on the duration of not sufficiently-lasting idles. These ones can be particularly affecting, because while performing in-use driving cycles it is not so common to be in idle for at least 30 seconds: measuring the number of times that the vehicle is in idle conditions for less than 30 seconds helps to understand what is the incidence of the standardized prescription on the idle recognition. The conditions for idle recognition have been already illustrated in Section 5.2.

To filter out every time window — larger or smaller than the prescribed threshold of 30 seconds — in which the idle conditions have been respected, without leading to a settled idle recognition, the following trigger has been implemented:

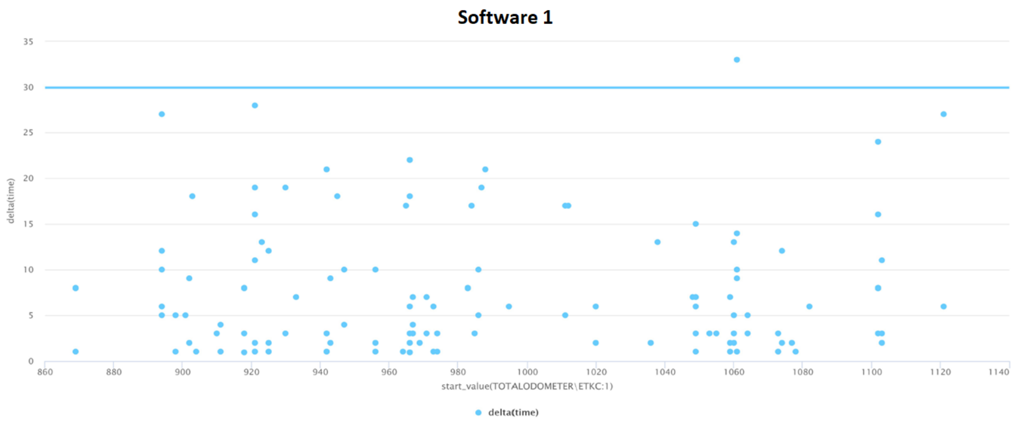
```
EXT_TRIGGER := detect/time {KEYON\ETK = 1 AND RPM\ETK > 0};
EXT_TRIGGER += delta(time);
EXT_TRIGGER += {
    NESTED_TRIGGER_2 := detect/time {KEYON\ETK = 1; diff(IUMPR_GE-
                        NDEN) = 2};
    NESTED_TRIGGER_2 += delta(time);
    NESTED_TRIGGER_2 += start_value ODOMETER\ETK, RPM\ETK;
    NESTED_TRIGGER_2 += end_value ODOMETER\ETK, RPM\ETK;
    NESTED_TRIGGER_2 += {
        SUBNESTED_TRIGGER_2 := detect/time {VEHSPEED\ETK < 1 AND
                               PEDAL\ETK = 0; VEHSPEED\ETK > 2 OR
                               PEDAL\ETK != 0};
        SUBNESTED_TRIGGER_2 += delta(time);
        SUBNESTED_TRIGGER_2 += start_value ODOMETER\ETK, RPM\ETK;
        SUBNESTED_TRIGGER_2 += end_value ODOMETER\ETK, RPM\ETK;
        };
    };
```

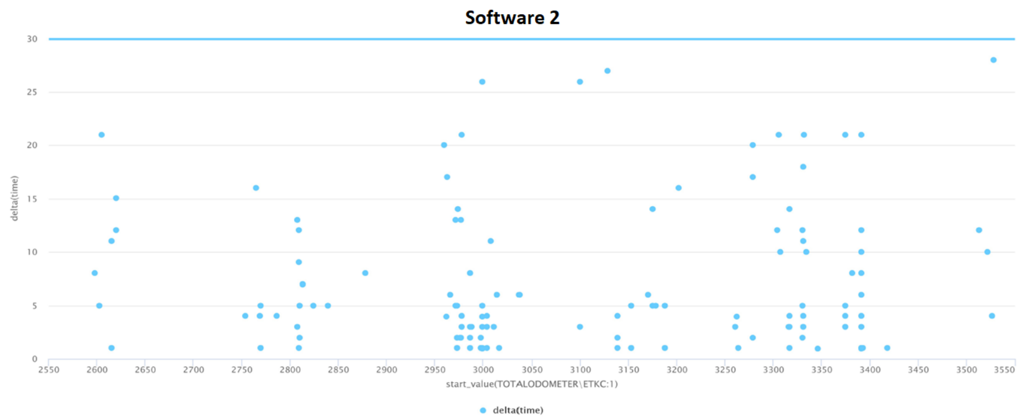
`EXT_TRIGGER` and `NESTED_TRIGGER_2` have the same logic encountered in Sections 5.4.1, 5.4.2, hence will not be discussed anymore. The only important tip to recall is that they trigger every period lasting from the start of trip and the moment in which the idle condition of the general denominator is verified (`diff(IUMPR_GENDEN) = 2`, since the increment to which the general denominator channel is subdued when an idle is detected). Nested under the two, the trigger `SUBNESTED_TRIGGER_2` detects all the time windows that accomplish the following conditions:

- They start whenever the accelerator pedal is totally released (`PEDAL\ETK = 0`) and the vehicle speed is under one km/h (`VEHSPEED\ETK < 1`);

- They close whenever one of the two conditions stops to be verified, i.e. either when the accelerator pedal is not released anymore (`PEDAL\ETK` $\neq$ `0`), or when the vehicle speed exceeds a certain threshold (`VEHSPEED\ETK` $>$ `2`).

In this way, every idle period — regardless their duration with respect to prescription — is recorded. The results, always divided by software update, are the following:





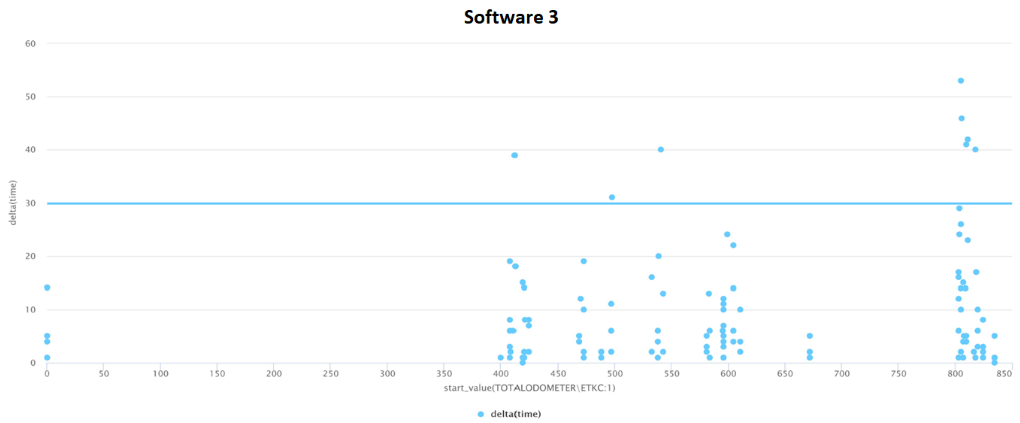


Figure 5.5: Results of Section 5.4.3 statistic, divided by software release: duration of every idle condition window, with highlighted threshold.

In Fig. 5.5 charts, which display the duration of every idle window — preceeding an idle detection — on the base of a temporal axis, in this case represented by the odometer. Two relevant information can be noted:

- There are a lot of periods in which the vehicle has been in true idle conditions, but for a time interval smaller than required. This means that the actual driving conditions requirement to consider a driving cycle as an in-use cycle are quite demanding from the point of view of idle recognition, hence the engine control software has to be resilient to this drawback.
- There are some misdetections, in which the two prescribed conditions have been established for durations larger than the requirement (i.e. 30 seconds), but they did not correspond to an idle recognition.

Before starting analyzing the reasons for the latter point, which will be discussed in Section 5.4.4, a quantitative estimation of the delay imposed to the idle recognition is led. Two kinds of histograms were produced: in the first one (Fig. 5.6), always divided by software, the counter of every idle window — with duration smaller or larger than the required — has been calculated for any measurement file.

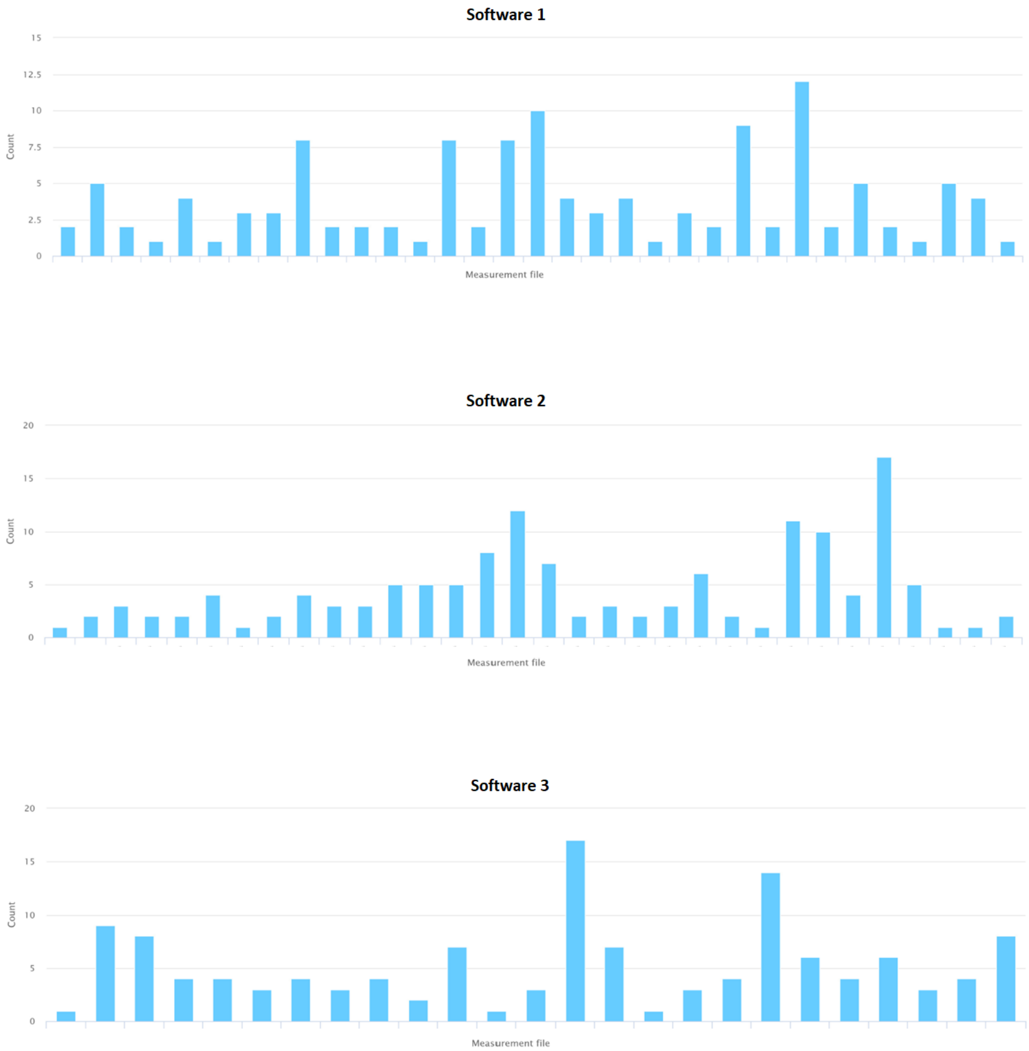


Figure 5.6: Counter of detected idle conditions for any measurement file, divided by software.

On the other hand, in the second ones (Fig. 5.7), it is possible to have a glimpse on the contribution that every unrecognized idle period has on delaying the idle recognition. According to the software, the average delay computed from those charts is:

- Software 1: $27.49s$
- Software 2: $29.35s$
- Software 3: $46.77s$

Figure 5.7: Histograms giving quantitative indications about the cumulative time lost in uncompliant idles per trip, divided by software.

5.4.4 Software check

The pinnacle of this analysis is gaining more insights about the reasons why there are some misdetected idle periods, in which both the vehicle speed and the accelerator pedal conditions are respected for more than 30 seconds, but no idle has been recognized by the software. With reference to Fig. 5.5 – 3 — where idle periods of not-prescribed durations have been detected, in relation to Software 3 — it can be noted the presence of many points over the detectability threshold. By scoping through one of those points, the occurences that are found out correspond to the situation represented, in form of signals, in Fig. 5.8:

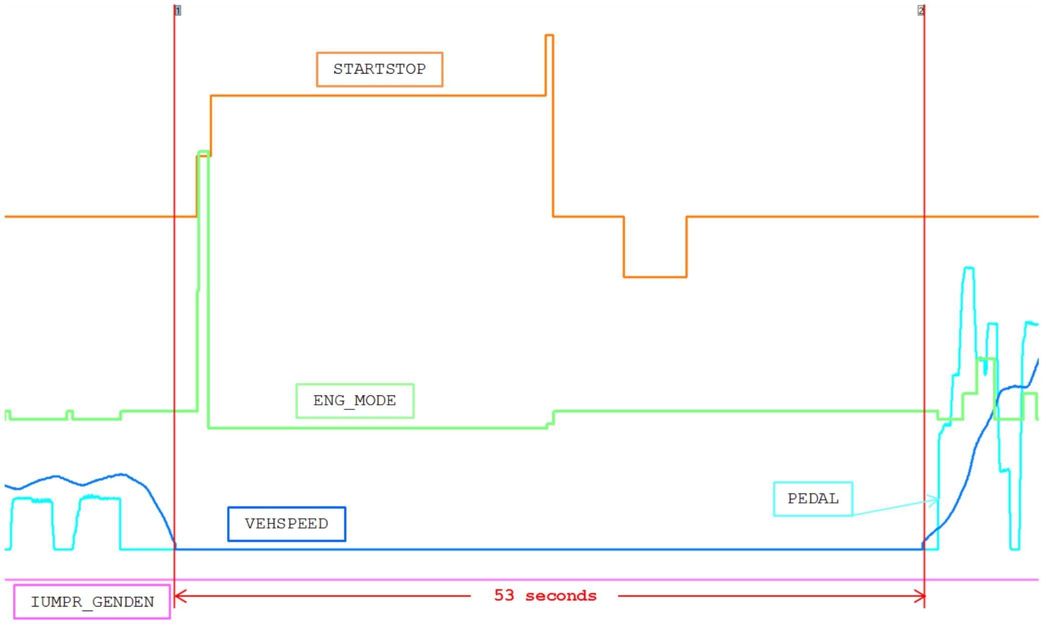


Figure 5.8: Occurrence relative to the highest point of Fig. 5.5 – 3.

As it can be seen — while the external condition (`NESTED_TRIGGER_2`) is satisfied, since the channel `IUMPR_GENDEN` has not incremented yet — the internal condtion (`SUBNESTED_TRIGGER_2`) is verified as soon as both the vehicle speed (`VEHSPEED < 1`) and the accelerator pedal (`PEDAL = 0`) conditions are satisfied (left red cursor), until at least one of them becomes not satisfied anymore (right red cursor). This window, as indicated in the figure above, lasts indeed 53 seconds, which is largely above the detectability threshold of 30 seconds. Furthermore, this does not lead to any idle recognition.

This undesired behavior is not casual, since in this window the vehicle has undergone a S&S condition: as it can be seen from Fig. 5.9, the red cursors delimit the start and the end of a Stop&Start maneuver, in which the engine is switched off — pursuant of adequate calibrated conditions that will not be discussed in this document — the engine rpm goes to zero as well, and the vehicle is stopped. The signal `STARTSTOP` records three different stages to properly identify each vehicle status: the maneuver starts in correspondence of *engine stopping* status, goes through *engine stopped* and ends with *cranking* status. At the end of the latter, the S&S maneuver is finished, as the engine has been meanwhile switched on again.

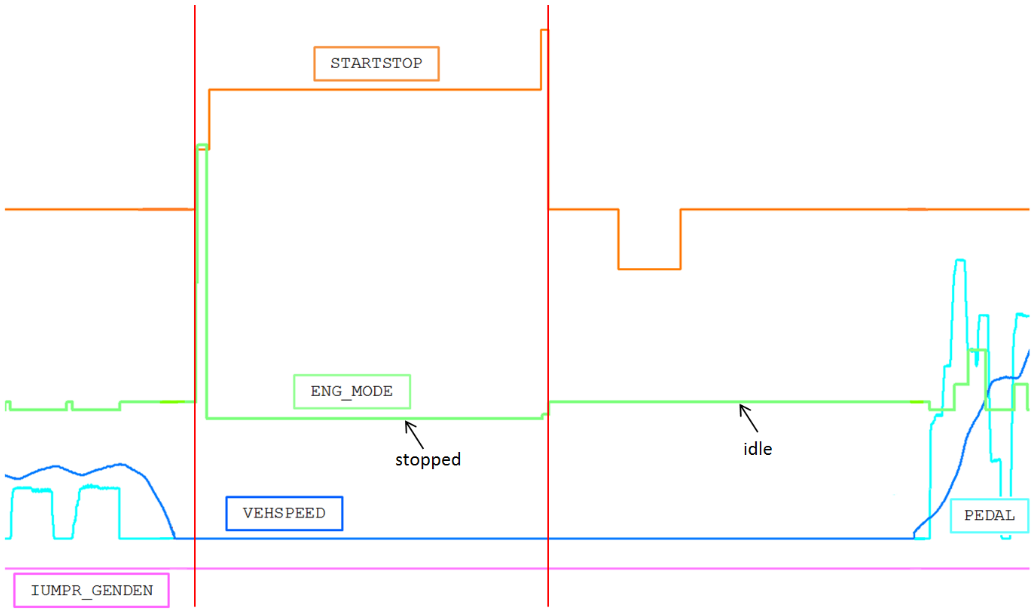


Figure 5.9: S&S maneuver highlighted within the occurrence of Fig. 5.8.

The duration of this maneuver is, as it can be seen, less than half of the whole duration of the triggered window. On the other hand, next to the cursors an idle period starts, in which the engine is running, but the vehicle is at standstill and no power is requested: this last period's duration is more than half of the previously mentioned window, but anyway smaller than 30 seconds.

Together with the `STARTSTOP` signal, also the signal describing the engine current modality (`ENG_MODE`) has been included. Neglecting the enumeration of all the statuses that can describe and assume the abovementioned channel, which is not of interest, it is anyway important to highlight that as soon as the S&S maneuver starts — as the engine enters the engine stopping S&S phase — the engine mode goes to cut-off (the green spike in correspondence of the left red cursor) mode, in which the fuel injection is dropped and the engine is consequently switched off. Then, as soon as the engine is stopped, `ENG_MODE` assumes the value *stopped*, until it returns to *idle* after the transition experienced during the cranking phase. The fact that the so triggered window is not recognized is a software problem, since the S&S maneuvers are not considered in the duration of 30 seconds requested by the norm, even if they should be, pursuant of the respected conditions upon vehicle speed and pedal position. As a consequence, a new trigger to test the software compliancy had to be assembled:

```
TRIP := detect/time {KEYON\ETK = 1; KEYON\ETK = 0};
TRIP += {
    GENDEN_INCREMENT := detect/time {KEYON\ETK = 1; diff(IUMPR_GEN-
                        DEN) = 2};
    GENDEN_INCREMENT += delta(time);
    GENDEN_INCREMENT += start_value ODOMETER\ETK, RPM\ETK;
    GENDEN_INCREMENT += {
        IDLE_WINDOW := detect/time :30:100 {VEHSPEED\ETK < 1 AND
                       PEDAL\ETK = 0; VEHSPEED\ETK > 2 OR PEDAL\ETK
                       != 0};
        IDLE_WINDOW += delta(time);
        IDLE_WINDOW += start_value ODOMETER\ETK, RPM\ETK;
        IDLE_WINDOW += end_value ODOMETER\ETK, RPM\ETK;
        IDLE_WINDOW += {
            ENG_MODE_STOPPED := detect/time {ENG_MODE\ETK = 0};
            ENG_MODE_STOPPED += delta(time);
            ENG_MODE_STOPPED += start_value ODOMETER\ETK, RPM\ETK;
            ENG_MODE_STOPPED += end_value ODOMETER\ETK, RPM\ETK;
            };
        IDLE_WINDOW += {
            ENG_MODE_IDLE := detect/time {ENG_MODE\ETK = 4};
            ENG_MODE_IDLE += delta(time);
            ENG_MODE_IDLE += start_value ODOMETER\ETK, RPM\ETK;
            ENG_MODE_IDLE += end_value ODOMETER\ETK, RPM\ETK;
            };
        };
    };
```

Although the names have changed, the substance of the trigger remained the same of the ones already explained in previous sections. In the following, each subsection of the trigger is explained:

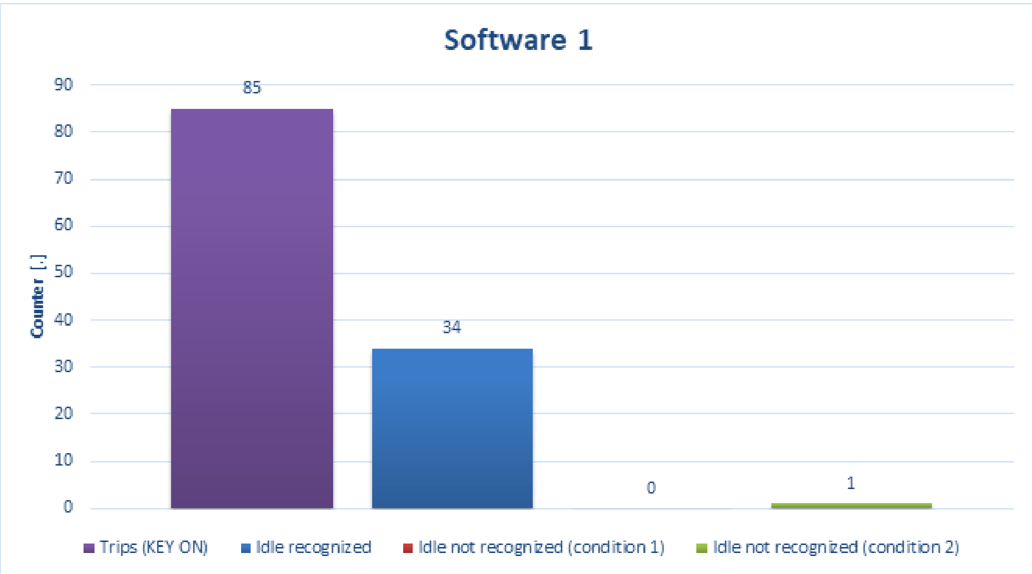
- `TRIP` is the global conditions implying that the key is inserted. Whenever the key is switched off, all the nested conditions fade. Notice that in case of S&S, even if the engine is switched off, the key remains inserted for the whole maneuver, i.e. the driving cycle is not considered as interrupted (i.e. the bit defining the key status remains still TRUE).

- `GENDEN_INCREMENT` identifies, within and from the start of each trip, the time windows that are spent before an idle recognition occurs (i.e. the channel `IUMPR_GENDEN` increments of 2).

- `IDLE_WINDOW` triggers all the windows — lasting more than 30 seconds (indicated by the syntax `detect/time` : 30 : 100, which stands detecting windows lasting at least 30 seconds up to a duration of 100 seconds) — for which the conditions for an idle recognition have been formally established. In this way, just the pathologic cases are isolated.

- `ENG_MODE_STOPPED` and `ENG_MODE_IDLE` detects the time frames in which the engine is either in stopped (`ENG_MODE = 0`) conditions, or in idle (`ENG_MODE = 4`) conditions. This last operation is needed to discriminate which of the two pathological behaviors had determined an idle misdetection:
  - *S&S lasting more than 30 seconds*: if the engine lies in the stopped condition for more than 30 seconds and this does not induce an idle recognition, a misdetection has occurred. This event will be addressed as Condition 1.
  - *S&S + idle windows lasting more than 30 seconds*: regardless of the duration of both the S&S and the idle periods, the combination of the two leads to an undetected idle. This other event will be addressed as Condition 2.

The results of the analysis illustrated above gave the following results, divided by software (Fig. 5.10):



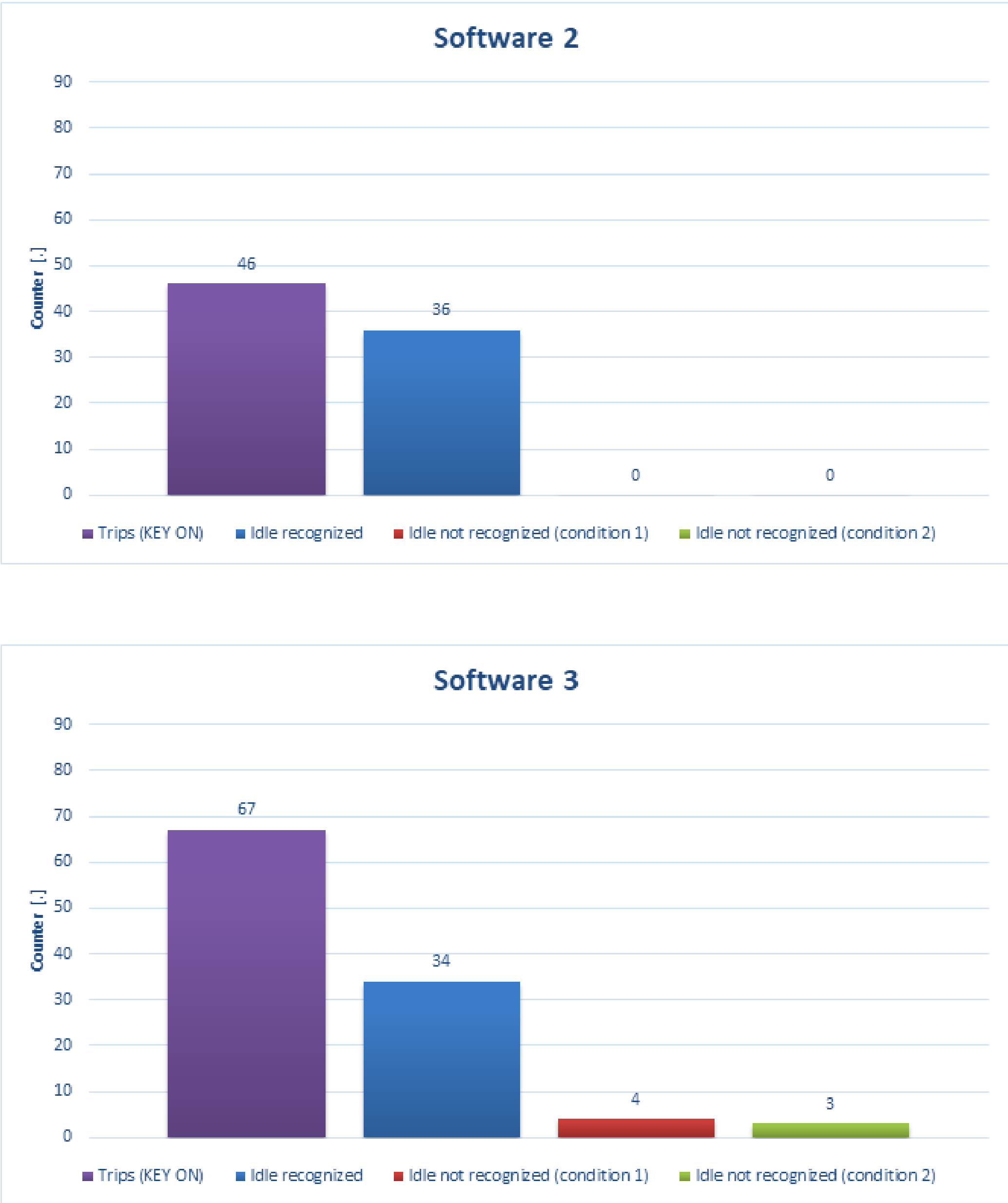


Figure 5.10: Outcomes of the software check inquiry divided by software.

5.4.5 Conclusions

On the base of the just mentioned results, it is quite evident that the only software that presents the problem of idle misdetection is the latest, in spite of its newer updates. Moreover, it cannot be considered a secondary problem, since:

- In the face of an idle recognition rate of nearly 51% per trip (i.e. the outcome — in percentage — of the number of idle recognized divided by the number of trips), which is anyway questionably low, there is an alarming **17% of unrecognized idles** out of the total number of idles that should have been recognized (i.e. the number of not recognized idles, 7, divided by the total number of idles that would have been recognized with a perfectly compliant software, 41). This not negligible portion affects negatively the EVAP in Boost IUMPR calculation, since the ECU loses some conditions that would else be conducive.

- The mutual proportions between the two fault modes can be considered halfway distributed, in spite of a low sample size. However, the misleading contribution of the S&S is decisive for the happening of the issue.

This statistic was clearly confirmative of suspicions that were already circulating through the various Powertrain Departments, since both at the calibration level, and at control system level, was already known. However, it also thoroughly showed – with quantitative and expendable data – that the problem was real and quite affecting, too. In a context so strict and legislated – as the emissions control one is – the continuous improvement philosophy is a must-have spirit, otherwise it could not be feasible to meet the required performance targets. Moreover, it should be taken into account not only the impulse of the authorities' challenging requests, but also the market competition for what concerns winning benchmarking comparisons.

5.5 EVAP system diagnoses chain

In the perspective of increasing the high flow purge valve IUMPR for benchmarking reasons and to obtain fresh emission reduction credits from CARB, the expiring time of the diagnoses chain leading to EVAP in Boost should be monitored and analyzed. The high flow purge valve diagnosis, in fact, comes as the final output of a cascade of preliminary diagnoses that should be made to be sure about the integrity of the system and the reliability of the information it provides.

The complete diagnoses chain leading to the diagnosticability of EVAP in Boost condition establishment is here omitted for confidentiality reasons, since the logic and the arrangement is property of Magneti Marelli Powertrain S.p.A and of Maserati S.p.A. Anyway, my analysis was led on four backbone diagnoses around which all the diagnostic chain pivots:

- *Purge valve learning*: it is a diagnosis that is made on the purge valve to verify if it is ready to accept the duty cycle imposed by the ECU and if the information it provides are reliable.

- *Clean Routine diagnosis*: it verifies the completion of the clean routine procedure. Simply, due to hardware faults that were evidenced in past experience vehicles, it happened that in boost conditions some water might introduce in the EVAP system from the CCV valve, due to the favorable pressure gradient it was realized with respect to the external environment while purging — to whom the CCV is directly connected (see Section 4.2.5). This was mainly due to a bad positioning of the CCV valve with respect to the ground, which unfortunately aided the event. At first, since a hardware modification was not yet feasible, it was decided to implement a strategy to clean the EVAP system ducts upstream the purge, to avoid blocked lines or worse to introduce dirt into the engine itself. This strategy was called **cleaning routine**, and it basically consists in sequentially operating the purge valve, regardless of the established boost condition, to make air purging from the EVAP ducts and in this way to remove dirt from the ducts. The completion of this procedure is a preliminary step to initiate the diagnoses of other monitorings, such as:

  - CCV stuck closed monitoring;
  - High-flow purge monitoring (i.e. EVAP in Boost);
  - ESIM switch rationality.

  Which are enabled if and only if the two following conditions — characterizing the cleaning routine completion — are verified:

  - Cumulative air mass purged through opened CCV is larger than a calibrated threshold, basically to be sure to have removed all water infiltration from the CCV-to-CPV circuit branch.
  - Cumulative dynamic purge flow from canister is larger than a calibrated threshold, to be sure alike that water infiltrations have been removed from canister-to-CPV circuit branch.

  This should allow to drain water infiltrations inside the EVAP system, that could lead to misdetections and may impair its functionality.

- *CCV stuck closed monitoring*: the aim of this test is to detect if the Canister Check Valve (i.e. the CCV) has been left **stuck closed**. This diagnosis is needed because if the CCV remains closed while the purge valve is experiencing a particularly heavy duty cycle, no fresh air can be drawn from the CCV and the pressure in the fuel tank and in the EVAP circuit upstream the purge valve subsequently decreases to unsafe values, which could ultimately lead to the collapse of the fuel tank.

- *Purge flow dual-path monitoring*: As already mentioned, the purge monitor test is needed to determine if the evaporative fuel vapors are really flowing into the engine when the CPV is opened. As explained, the canister purging can occur in two different fashions: it can be actuated in naturally aspirated engine mode — owing to the vacuum inside the intake manifold — and in boosted mode, as well, because of the presence of the ejector tee. Hence, two different tests exist to determine which of the EVAP circuit branches is blocked. The monitoring for Boost operations — which is, for our purposes, the most interesting one — shall fulfill a set of enable conditions to be started:
  - The relative manifold pressure — i.e. the difference between the pressure within the intake manifold and the ambient pressure — must be positive and higher than a calibrated threshold. If this threshold is too high, a poor detectability will be experienced, conversely if it is too low the vehicle could actually not be in Boost conditions.
  - The intake mass airflow must be within a specified range. This is done to guarantee that a sufficient airflow is indicating that the Boost conditions have been established.
  - The environmental temperature should be larger than a minimum threshold.
  - The canister load coefficient — which is a modelled variable that estimates the amount of fuel vapors currently stored within the canister — should be lower than a threshold. This is requested because, in Boost conditions, the canister should be purged by the EVAP in Boost branch, unless it is blocked from some undesired obstacle.

  When the test is enabled, the CPV is closed and the CCV is commanded open to stabilize the system at the ambient pressure. After a calibratable timeout, a rationality check on fuel tank pressure sensor is performed (it should measure a value close to zero, as the system has been put in communication with the external environment, i.e. at ambient pressure). If the pressure signal reaches the atmospheric value, it means no fault on sensor is present, otherwise the monitoring is interrupted. Following the former case, the CPV is opened to a target and the CCV is closed. During a second calibratable timeout, the minimum value of the fuel tank pressure drop is monitored and at the end of the timeout the delta pressure is calculated: if it is inside a plausible range of vacuum, the test is passed. Otherwise, different type of faults can be set according to the measured delta pressure (Fig. 5.11).

When the timeout is expired, a check on the delta fuel tank pressure is done:

- If the delta pressure is inside a range centered on the atmospheric pressure it means that CV2 or CPV are stuck closed and no purge flow is allowed;
- If the delta pressure is negative (i.e. the pressure has increased) a reverse flow from the intake manifold is detected; the possible cause is that CV1 stuck opened;
- If the pressure drops below a minimum threshold an excessive vacuum has been generated;
- If the delta pressure is inside an acceptable range the test is passed. Each of the previous cases is considered a fault.

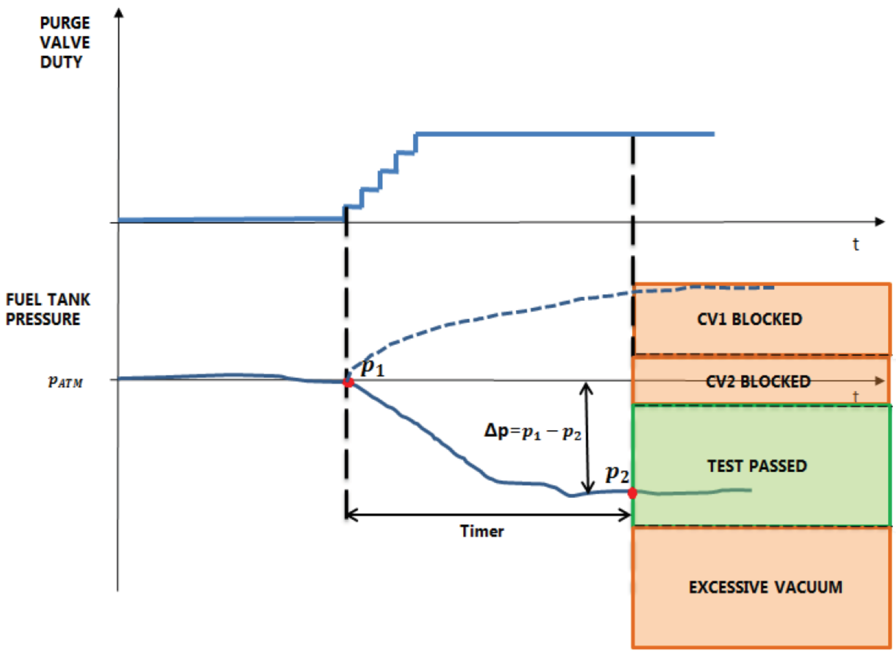


Figure 5.11: Scheme of the EVAP in Boost fault diagnosis.

To evaluate the importance of the delaying contribution of all the above mentioned monitorings upon the purge flow dual-path's one, which is strictly bounded to unsatisfactory values of the relative IUMPR index, the following trigger has been implemented:

```
TRIP := detect/time {KEYON\ETK = 1};
TRIP += {
    CPV_LRND := detect/time {CPV_LRND\ETK = 0; diff(CPV_LRND\ETK)
                != 0};
    CPV_LRND += delta(time);
    };
TRIP += {
    CCV_STCL := detect/time {CCV_STCL\ETK = 0; diff(CCV_STCL\ETK)
                != 0};
    CCV_STCL += delta(time);
    };
TRIP += {
    CLEAN_ROUTINE := detect/time {CLEAN_ROUTINE\ETK = 0; diff(CLE-
                     AN_ROUTINE\ETK) > 1};
    CLEAN_ROUTINE += delta(time);
    };
TRIP += {
    HIGH_PRG_FLOW := detect/time {HIGH_PRG_FLOW\ETK = 0; diff(HIGH_
                     PRG_FLOW\ETK) > 2 AND HIGH_PRG_FLOW\ETK = 6};
    HIGH_PRG_FLOW += delta(time);
    };
```

As it can be seen, this trigger is far simpler than the previous Sections ones. In the following, its syntax is thoroughly explained:

- `TRIP` is the global conditions implying that the key is inserted. Whenever the key is switched off, all the nested conditions fade.

- `CPV_LRND` triggers each window opening at the start of the trip and closing as well as the purge learning has been completed. The syntax `diff(CPV_LRND\ETK) != 0` indicates that the difference between the channel `CPV_LRND` current sample and one sample before is different from zero (i.e. the channel has incremented). Under those conditions, the time spent (`delta(time)`) is measured.

- `CCV_STCL` embodies the same logic of the previous nested trigger (`CPV_LRND`), but implemented on the channel `CCV_STCL`. Under those conditions, the time required to complete the CCV stuck closed diagnosis is measured.

- `CLEAN_ROUTINE` triggers each window opening at the start of the trip and closing as well as the clean routine is completed. As it can be seen from Fig. 5.12, this one is considered completed as soon as the channel CLEAN_ROUTINE increments of at least 2 between two consecutive samples. This occurrence verifies only when the clean routine channel reaches the value 3, corresponding to routine completed.

- HIGH_PRG_FLOW triggers each window opening at the start of the trip and closing when the channel HIGH_PRG_FLOW increments of at least 3 and simultaneously the value it assumes is 6, corresponding to high purge flow diagnosis completed. The two closing conditions have been inserted together because — in case the bare conditions about the increment had been defined — there would have been an ambiguity between preceding increments, which equally satisfied the condition.

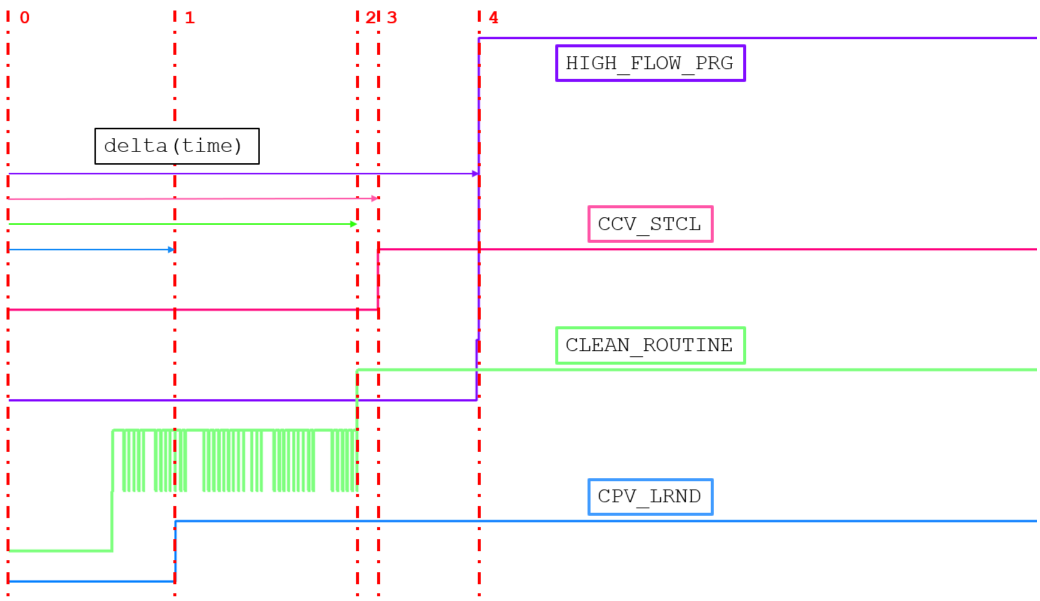


Figure 5.12: Rise time of each diagnosis. The cursors are placed in the exact instant in which each channel increments.

The results of the trigger processing — relative to just one engine control software, whose name and characteristics will not here be specified — with the measurement files already at dispose from the analysis led in Section 5.4, are the ones displayed in the histograms of Fig. 5.13.

As the figure suggests, the fulfillment of the cleaning routine has actually an impact on the completion of the CCV stuck closed diagnosis, which is logically in sequence. Since the diagnosis upon EVAP in Boost is subsequent to both of them, it can be actually concluded that the presence of the clean

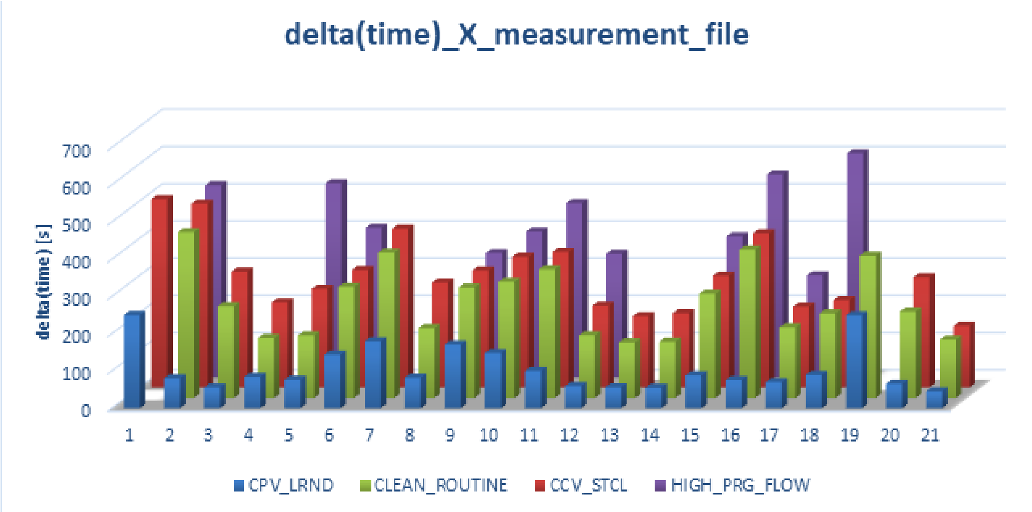


Figure 5.13: Overall rise times of each diagnosis, per measurement file.

routine hinders the rate at which the EVAP in Boost diagnosis is completed. As a consequence, the IUMPR index evaluation is hindered, too.

A further confirmation of the trend is relatable from Fig. 5.14, in which the average rise times for each diagnosis are displayed. Those numbers have been calculated simply as the average of each column of Fig. 5.13 per indicator:

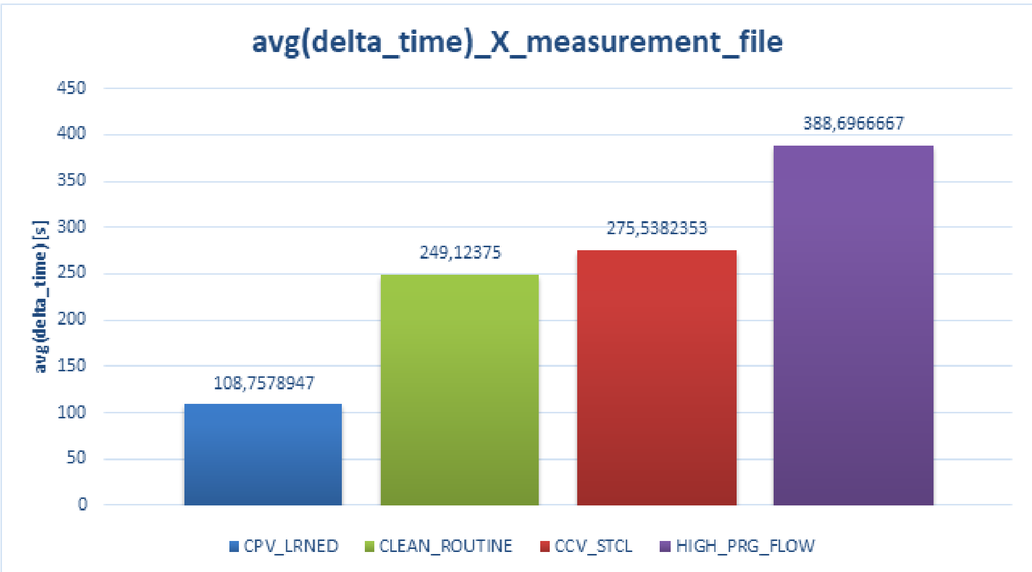


Figure 5.14: Average completion time of each diagnosis.

It could be concluded that without the cleaning routine procedure, the two following diagnoses would be completed faster, with expendable benefits in terms of IUMPR improvement. To completely remove the cleaning routine, it is necessary to rearrange the layout of the CCV branch, in order to avoid water and dirt intrusion during ordinary vehicle usage. This modification is currently under development, and will be presented in following months.

Chapter 6

Future horizons

As introduced in Chapter 3, my internship experience was not just bounded to a mere drafting and composition of statistics and technical reports — although they actually required a great mole of work — but also with the development of Moogle in cooperation with ETAS technicians. This task included the management of the contacts and interpersonal relationship with them, pursuing the establishment of a clear and structured communication, searching for the best improvements to enhance and facilitate the user experience. In this last chapter, the newest updates that I succeeded in achieving will be presented, as well as a new personal and hypothetic view of what could be the development of Moogle inside Maserati as aiding instrument to engine calibration. Eventually, intimate and solemn considerations about my internship experience will conclude this discussion.

6.1 Latest developments

The main advantage of this developing activity was that an everyday user is surely aware of all the troubles and problematic behaviors characterizing the software usage, with respect to other company entities. From ETAS' side, this represents an incredible opportunity to have a deep, articulated and motivated technical feedback from their potential customers, being the best *litmus test* for their corporate development. On the other hand, Maserati could exploit the potentials of such a powerful instrument to hugely enhance the comprehension of the phenomena occurring in its vehicles, as well as having a finer development tuning. On reflection, a better and more aware development translates into better market performances, regardless of the financial strategies adopted. Furthermore, with this activity Maserati could properly form fresh engineers to insert in corporate teams, allowing them to understand and to manage in time all the knowledge that should be digested to work in this company. As a third counterpart, this activity is also an important formation occasion for stagiaires, because allows them to be trained as already-hired engineers and to get insights on many arguments that could be expendable for their Master's Thesis. Eventually, live and telematic meetings with ETAS are an important occasion to learn to interlace and renewing contacts with a foreign supplier.

Before my internship experience, several stagiaires had already alternated to the role, improving the software in many ways and making it globally stable and usable. Moreover, a proven collaboration between ETAS and Maserati already existed, although impoverished by a recent slack period in which no substantial development had been carried out. The reason of this — together with a more mature software, requiring less and less attention and development — might be searched in the changes in commercial agreements between the two companies, which saw Maserati downgraded from the privileged position of unique developer in favor of other software developer companies recently taking the lead. This is at least logic: Maserati is a car maker, hence interested in making and selling vehicles to specific customer segments; it is not a software developer company. On one hand, Moogle already works in a way that makes us satisfied, thus there is not a constant and strong pulse towards further developments. On the other hand, problems and improvements are gradually becoming subtler and harder to be conceived or even explained, because they may be related to programming logic issues or to intrinsic mechanisms in which the applicative operates. The development has reached a point in which it is legit to ask themselves if we actually have the competencies to develop the software as usefully and thoroughly as a software developer company. It is probable that ETAS made the same reasoning: from their point of view, it was more convenient to have a car maker, rather than a software maker company, as developer in the prior stages of the software's life. Who could have been a better source of hints and requests about how Moogle should have worked, from the point of view of a car maker who simply needed to analyze their vehicles accumulation history? At the same time — in a more mature stage of Moogle's development, once the needs of a car maker had been properly shaped out and addressed — a more cognizant and technically aware aid could have been more useful to perfect the software and lead it to the commercialization.

However, despite the background conditions that characterized Moogle development from the start of my trainee period, there was anyway room for improvement. There have been principally two issues that we managed during this time:

- *OPL 86*: when a trigger was requesting a specific channel from the measurement files, but this channel was just present in a few of them, the crawling ended up with no results because the trigger was disabled for all measurement files. As a consequence, it was not even possible to plot anyone of the occurrences that would have been detected in the portion of measures actually containing those missing channels. This was a huge problem, because it may happen that — in the complete accumulation history of a vehicle — two or more different experiments had been used. For instance, if a particular issue or behavior occurred at an already advanced stage of the vehicle life, and this issue implied the inclusion of new channels in the experiment, to read this new behavior, Moogle would not be able to generate any report because of the late introduction of those channels of interest. Furthermore, the lack of channels could also be not intentional, and it would be highly undesired to miss the possibility of analyzing the whole accumulation period just because in few weeks there had been errors with the experiments. Thus, the problem was discussed with ETAS both via e-mail and via Skype meeting, in which we had the possibility to properly and furtherly illustrate the pathologic behaviors. Moreover, with the shared screens it was possible to demonstrate in real time every criticality, from the trigger drafting to the complete crawling.

- *OPL 87*: a different behavior between Moogle Test environment and the Productive one was outlined. It occurred that processing some triggers in Moogle Test, with a subset of measurement files — stemming from the complete set in the relative vehicle folder — some occurrences were detected, while processing the same trigger with the complete set of measures — together with other older triggers — those occurrences disappeared. Furthermore, when this happened, the trigger list available in the report page was missing of a significant amount of triggers that should have given occurrences, if the crawler had worked properly. Since the way in which the two applications extract data are the same, there should have been issues related either to the size of the crawled folder, or to the number and/or the types of triggers simultaneously present in the folder. We established a thick and fruitful dialogue with ETAS to get insights on the problem, including one troubleshooting session in which it was replicated in real time via Skype and shared screens. Excluding gradually all the possible causes, we finally managed to hit the real cause, which was related to not tolerated syntaxes that were causing an indexing loop. This issue regarded two out of the 60 triggers actually present in the complete folder, but the number was enough to impair the whole crawler functionality. This problem was not known by ETAS technicians, also because the abovementioned pathologic syntax was correct from the C++ coding point of view, but it was not correctly interpreted by the software. In this way, we managed to not only improve significantly the everyday experience of the software — achieving a perfected crawlability of every folder of interest — but also to help ETAS to discover important and disqualifying bugs.

The typology and the degree of challengingness of the two abovementioned problems let intuiting the hardship of developing a software at this level of reached complexity. However, there is still large room for improvement, since the hindering and annoying behavior of Moogle in the everyday usage certainly abound. There have been many times, for instance, in which we needed to make the data post-processing on Excel, because Moogle did not allow the flexibility or the data representation features that we were searching for. Two examples are the statistics I have introduced in Sections 5.4 and 5.5, in which many charts have been realized with Excel. In the latter, for instance, the post-processing became necessary because the rise time of the four diagnoses were calculated in four different nested triggers, and Moogle do not allow to represent simultaneously many nested triggers in the same chart, without for example representing the results of the external trigger, which is not of interest. Usually, we felt cramped to aggregation and representation logics that could not promote a didactic and explanatory visualization of the results. The capability of signaling, reporting, and asking for development of all the lacking features we have actually spotted out during our journey is not only a matter of ability, but also of opportunity and politics. Whoever has ever worked within a company knows what I am addressing to.

6.2 A new role

On the base of my experience within the company, I tried to imagine how to make such a powerful instrument as Moogle more and more incisive and pervasive. In my opinion, the most interesting characteristic of this tool is that it allows calibrators to have a wider and more detailed perspective of the evolution of the behaviors of a theoretically infinite number of variables. As I have illustrated in Chapters 2 and 3, the only way calibrators have to validate their reasonings is to take a vehicle, to imagine the driving maneuvers it shall replicate to observe the behavior of interest, recording the measures through one or more driving cycles and finally analyzing data with Excel post-processing. The problem is not only that one or more tests are required for each behavior to be verified — i.e. money and time — but also that the outcoming set of measures is very small and not indicative from a mere statistical point of view. Since statistics is the only instrument to improve the robustness of a company process, this makes Moogle users an important feedback source for calibrators' comprehension of the systems onto which they are working and about the phenomena that are going on. In the Validation Team, although there are many data analysts, only a few of them plays a role of connection between the calibration and the validation spheres, also because there are many other duties that the team as a whole has to accomplish. Sometimes there is a perceived lack of circularity — that risks to solely burden on those ones — that prevents the various teams to provide a unified and more reasoned solution. Even though an intern takes 2-3 months to become fully autonomous and aware about all the powertrain systems and components that are running through the company, and fully capable of managing the software for every demand as well, it is just in the very last weeks of its experience that it becomes sufficiently skilled and trained to be helpful in a technical meeting and to add something of merit to the discussion. In the last month of my internship, in fact, me and my colleagues have been able to improve the discussion about many calibration and software updates that had in someway to be revised and/or enhanced, on the base of the results of our daily data analysis activity. A high degree of circularity was on the point to be established, not only between me and my colleagues of the Validation team, but also between our team and other teams. We were slowly becoming a reference point for anyone needing a detailed and proven analysis inside the company. Unfortunately, my internship period finished shortly thereafter, and my acquired competences could not be expended for the improvement anymore. This habit is henceforth consolidated, since every stagiaire follows more or less the same path; because of this cyclic workforce renewal, there is not the possibility to establish a data analysis pole, which any company agent could take benefit from.

My idea starts from the previous foundational reasoning: it could be a huge enhancement to stably hire a person who could:

- Manage the existent ones and weave new relationships with ETAS. Without this cyclic change, it could be possible to have more time, more experience and more leverage to ask for improvements, in a structured and settled way. This could hugely enhance the software everyday usage, expanding data analysis and representation possibilities and cutting off the time required to produce them. Ultimately, it could utterly improve the communication, the comprehension and the collaboration between the various Powertrain Testing teams.

- Propose as interface between Validation and Calibration teams, albeit remaining closely part of the former. In this case, every clarification request from both the two entities would pass from this figure, with the immediate result of reducing the effort and the diligence of other workers, who could have the opportunity to focus on just their mansion. Furthermore, every request would be managed in less time, with higher efficiency and finer level of detail, to achieve a complete and steady comprehension of all the phenomena.

- Increase the leverage capability of the Validation team as a whole, because all other entities should actually address this one for every clarification or support request. Moreover, supposing that this figure was already settled, the Validation team could become the only custodian of a global and at the same time aware perspective on the widest range of problems inside the company.

- Train better new stagiaires: differently to what happens nowadays, in which the formation of young engineers is mainly addressed to the data analysts of the Validation team, there would be the possibility to dedicate more time and resources to daily train youngster engineers. In this way, not only they would gain the required level of preparation in a remarkably lower amount of time, but they would also be able to participate and to leave their mark in technical meetings after few months. Eventually, in spite of a lower amount of educated interns per year, they would be more prepared and skilled at the end of their internship period, becoming more attractive for hiring purposes.

Obviously, to implement a figure of this kind an initial investment has to be considered. Ultimately, also considering the numerous benefits that I have conceived, it can be said that this economic effort could worth.

6.3 Ending

With this paragraph the discussion of my Master's Thesis ends. These six months in Maserati have been very fascinating, as well as full of unknowns, sometimes discomforts and challenges that I had to overcome entirely on my own. I consider the Company Internship a very educational project, as it allows students to measure themselves in an unprotected context and often devoid of references: from this side, it appears to be majorly distant from what happens in the University everyday life. Starting this path as students means having the opportunity to naturally bring the scientific and learning-oriented mentality, characteristic of the University experience, in a context in which it is much more important to *do* and operate rather than personal erudition. Sometimes the student point of view can be constructive, since limiting wit, the free usage of imagination and the possibility to investigate problems can sometimes lead to superficial conclusions and works. On the other hand, this experience may allows students to be more prepared and competitive once the moment to choose the job arrives, pursuant of their gained experience in terms of corporate logic, hiring mechanisms, job placement, and so on. Moreover, it also allows to get a preliminary idea of what it takes from the job, what you want to become, what you do not like, what you would like to improve. In my experience I have found very talented colleagues, willing to offer me a hand for whatever I needed, or to show me the right way.

I consider my Thesis work as a first significant step towards a complete modernization of business processes, which should pass from the analysis of Big Data. Considering the always increasing volume of data characterizing transversally every human activity in today's society, in absence of a tool allowing to extract valid information from the jungle of available data to which everyone of us is exposed, such data would be useless. The transformation towards a society permeated by *control* and data analytics phylosophy was so rapid and sudden, that developing a data analysis expertise became soon a need for market survival. Maserati is the first automotive company in Italy to undertake a path of this type: this choice is starting to bear fruits from the point of view of the continuous improvement of its products, and in the near future the impact could also be assessed in relation to the *quantity* and *quality* of the data analysts thus formed. The validation, in fact, turns out to be the completion ring of the testing process of the entire Powertrain division, and just in presence of a department that can collect and resolve the criticalities proposed by all development teams it is possible to get a full feedback and a cyclicity on development processes. As it can be guessed, a cyclical and constantly validated development would have the immediate effect of reducing the testing times and with them the deriving costs.

In Chapter 5, I had the opportunity to deal in depth with the topic of the IUMPR indexes, which constitute an efficiency parameters of a vehicle's OBD system, as well as its performance in daily use, in relation to specific and predefined limits. The quantity of IUMPR indexes that can be evaluated in a vehicle is enormous, of the same order of magnitude as the number of components overall present in that vehicle. Since the subsequent evaluation that they offer turns out to be very specific and detailed, it is very probable that the improvement of a specific index requires a complex activity of root causes identification, plus a detailed knowledge of the system and of the correlations between the various systems involved. This was also the case of the EVAP system, with which — in addition to the statistics I have presented in this discussion — I have worked for most of my internship. Often and willingly, none of us was preliminarily aware of the way we should have followed to improve a function that could positively affect the IUMPR index: many technical meetings became necessary to exchange ideas, to give its own contribution and to offer a different knowledge of the system and decide together on the next steps. This working method — other than being very stimulating and training — allowed us to thoroughly investigate all the possible unwanted behaviors that could have negatively affected the IUMPR indexes from the bencharking point of view.

Con queste parole, la mia avventura di student si conclude. Segue…

References

Informatic Resources

[1] Venturella, Antonino, “*La storia della Maserati*”, Last modified june 05, 2008.

`http://www.targaflorio-1906-1977.it/paginesommario/Pagine-costruttori/La%20storia%20della%20Maserati.htm`

[2] —, “*Maserati, un viaggio lungo 100 anni*”, Icon Wheels magazine, Last modified december 17, 2014.

`https://wheels.iconmagazine.it/auto-classiche/auto-story/storia-maserati`

[3] —, “*Il marchio Maserati*, Museo del Marchio Italiano.

`https://www.museodelmarchioitaliano.it/marchi/maserati.php`

[4] —, “*Una storia italiana, un'icona globale*”, Maserati site.

`https://www.maserati.com/it/it/Maserati-Life/company`

[5] Campignotto, Antonio (2016), “*La Ruota e l'Incudine la memoria dell'Industria Meccanica bolognese in Certosa*”, Minerva.

`https://www.storiaememoriadibologna.it/maserati-osca-automobili-442-organizzazione`

[6] Paganoni, Giulia, “*Trofeo e Gts, per il suv Levante una cura di potenza e prestazioni*”, Il Sole24Ore Motori, Last modified august 17, 2019.

`https://www.ilsole24ore.com/art/trofeo-e-gts-il-suv-levante-cura-potenza-e-prestazioni-ACxhaQZ?refresh_ce=1`

[7] —, “*OBD-II Background*”, ®2011 B&B Electronics.

`http://www.obdii.com/background.html`

[8] —, "*Il CAN Bus*", Elettronica Open Source, Last modified august 03, 2018.

https://it.emcelettronica.com/il-can-bus

[9] —, "*CAN Bus Explained - A Simple Intro (2019)*", CSS Electronics site, 2019.

https://www.csselectronics.com/screen/page/simple-intro-to-can-bus/language/en

[10] —, "*Controller Area Network (CAN): A Serial Bus System - Not Just For Vehicles*", esd Electronics Inc.

http://www.esd-electronics-usa.com/Controller-Area-Network-CAN-Introduction.html

[11] —, "*XETK – XCP-compliant ECU Interface*", ETAS GmbH, 2018.

https://www.etas.com/en/products/xetk.php

[12] —, "*ASAM MCD-1 XCP*", Association for Standardization of Automation and Measuring systems (ASAM), 2018.

https://www.asam.net/standards/detail/mcd-1-xcp/wiki/

[13] —, "*Connected Car Challenge? Big Data. Meet Delphi's Solution.*", Aptiv PLC, Last modified april 17, 2017.

https://www.aptiv.com/newsroom/article/2017/11/27/connected-car-challenge-big-data.-meet-delphi-s-solution

[14] —, "*Process Capability Index (Cpk).*", WhatisSixSigma.net.

https://www.whatissixsigma.net/process-capability-index-cpk/

[15] —, "*MDA – Measure Data Analyzer*", ETAS GmbH.

https://www.etas.com/en/products/mda.php

[16] —, "*INCA Software Products*", ETAS GmbH.

https://www.etas.com/en/products/inca_software_products-details.php?printThisDoc=true

[17] —, "*Control-Tec, Qualifier®: Automated data acquisition & analysis*", Delphi Automotive, 2014.

[18] —, "*The definition of Big Data*", Oracle Corp. site.

https://www.oracle.com/big-data/guide/what-is-big-data.html

[19] Pickell, Devin (2018), "*Structured vs Unstructured Data — What's the Difference?*", G2.com site, Last modified november 16, 2018.

https://learn.g2.com/structured-vs-unstructured-data

[20] —, "*Carbon Canister and Evaporative Emission Control System Modeling*", GT, The leading CAE platform for Multi-Physics system simulations site.

https://www.gtisoft.com/gamma_news/carbon-canister-and-evaporative-emission-control-system-modeling/

[21] MECA (Manufacturers of Emission Controls Association) (2010), "*Evaporative Emission Control Technologies for Gasoline Powered Vehicles*", MECA site.

http://www.meca.org/technology/technology-details?id=10&name=Evaporative%20Emission%20Controls

[22] Muscoplat, Rick (2019), "*ESIM on late model Chrysler vehicles*", Rick's free auto repair advices, Last modified january 01, 2019.

https://ricksfreeautorepairadvice.com/esim-on-late-model-chrysler-vehicles/

[23] Lyons, Allen (2015), "*On-Board Diagnostics (OBD) Program Overview*", California Air Resources Board (CARB), Mexico City, April 2015.

https://theicct.org/sites/default/files/6_ARB_OBD.pdf

Papery resources

[24] Buzzonetti, Daniele, Barbolini Ferrari, E. (ed. by), Bulgarelli, A. (ed. by) (2014), "*1914-2014 Maserati. 100 anni di storia attraverso i fatti più significativi*", Artioli Editore.

[25] Multiple Authors, "*Prove e test speciali di affidabilità in ottica cliente*", FCA Italy S.p.A. ®, ed. 6, 2015.

[26] Multiple Authors, "*ETAS GmbH, ETAS ECU Interfaces Solutions: the ETK Product family*", ETAS GmbH, 2018.

[27] Markl, Patrick, Albrecht, Stefan (2011), "*Analyzing AUTOSAR ECU Software with XCP*", Hanser Automotive.

[28] Elgendy, Nada, Elragal, Ahmed (2014), "*Big Data Analytics: a literature review paper*", Conference Paper in Lecture Notes in Computer Science, Springer International Publishing, Department of Business Informatics & Operations, German University in Cairo (GUC), Cairo, Egypt.

DOI: 10.1007/978-3-319-08976-8_16

`https://www.researchgate.net/profile/Ahmed_Elragal/publication/264555968_Big_Data_Analytics_A_Literature_Review_Paper/links/541e9b9a0cf203f155c0655a/Big-Data-Analytics-A-Literature-Review-Paper.pdf`

[29] Robert Bosch GmbH (2019), "*Automotive Handbook*", 10th edition, Wiley.

[30] Maserati Academy (2015), "*Motore 3.0L V6 biturbo, diagnosi avanzata e riparazione*", Maserati Training Network, Modena.

[31] Maserati Academy (2016), "*Motore 3.8L V8 biturbo, diagnosi avanzata e riparazione*", Maserati Training Network, Modena.

[32] Genta, Giancarlo et al. (2014), "*The Motor Car: Past, Present and Future*", Springer, Mechanical Engineering Series, Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Torino (Italy).

[33] Williams, R. and Clontz, C. (2001), "*Impact and Control of Canister Bleed Emissions*", SAE Technical Paper 2001-01-0733.

`https://doi.org/10.4271/2001-01-0733.`

[34] D. H. Edmund, R. C. Sager, W. B. Blomquist, M. R. Grote, F. A. Gibb, C. G. Hadre (2015), "*Boost purge ejector tee arrangement*", United States Patent Application Publication, US 2015/0096541 A1, 9 April 2015.

[35] Multiple Authors (2019), "*Canister purge valve duty cycle calculation*", Magneti Marelli Powertrain S.p.A., May 2019.

[36] Multiple Authors (2017), "*Canister air and fuel flow estimation*", Magneti Marelli Powertrain S.p.A., May 2017.

[37] Dixon, Matthew (2008), "*Chrysler Evaporative Leak Detection Systems*", Southern Illinois University Carbondale, OpenSIUC Presentations, Department of Automotive Technology, 2008.

`http://opensiuc.lib.siu.edu/auto_pres/7`

[38] Standler, D. (2017), "*Chrysler engine OBD-II requirements for passenger cars, light-duty trucks and medium-duty vehicles*", FCA US, OBD Cert. and Compliance Department, 2017.

[39] Multiple Authors (2019), "*in-Use Monitoring Performance Ratio calculation*", Magneti Marelli Powertrain S.p.A., May 2019.

[40] Multiple Authors (2019), "*EVAP diagnosis*", Magneti Marelli Powertrain S.p.A., May 2019.

[41] Multiple Authors (2019), "*Evaporative Emissions Diagnostics*", FCA US, 4 April 2019.

[42] California ARB (2016), "*EOBD II regulation, section 1968.2 of title 13*", Barclays Official California Code of Regulations, as approved by OAL on July 25, 2016.

[43] Multiple Authors (1988), "*The control of vehicle evaporative and refuelling emissions - the on-board" system*", CONCAWE AEISTF- I, The Hague, November 1988.