POLITECNICO DI TORINO

Department of Mechanical, Aerospace, Automotive, and Production Engineering

Master of Science in Automotive Engineering

Master's degree Thesis

Development of an algorithm for the control of lambda in 3D-CFD simulation of gasoline engines

Professor:

Prof. Ezio Spessa

Supervisors:

Dr. Ing. Marco Chiodi

M.Sc. Mario Pipolo

Candidate:

Giuseppe Nestola

Student ID:291413

ACADEMIC YEAR 2022-2023

A te

che mi guidi

dall'alto

Abstract

This master thesis describes the development and validation of a real-time regulator tool for 3D-CFD simulations of internal combustion engines.

The aim of the project was to build an algorithm that regulates key engine parameters during 3D-CFD simulations performed using QuickSim, a 3D-CFD tool developed by Dr-Ing. Marco Chiodi during his PhD at FKFS and Institut für Verbrennungsmotoren und Kraftfahrwesen (IVK) of University of Stuttgart. The tool guarantees high performance, speed and reliability of evaluation thanks to appropriate models. The project was carried out during an internship period at the Forschungsinstitut für Kraftfahrwesen und Fahrzeugmotoren Stuttgart (FKFS) in Stuttgart, Germany.

The developed regulator tool focuses on critical engine parameters such as indicated mean effective pressure, the combustion center of gravity tuning an appropriate ignition point and the lambda by adjusting the boost pressure and the amount of fuel injected.

The tool is designed to ensure that the engine parameters are within pre-defined target values set during the pre-processing phase of the simulation. During the simulation, the regulator tool checks the engine parameters at selected points of interest during the engine cycle. If any parameter is found to be outside the target range, the tool modifies the boundary conditions to adjust the parameter.

The regulator tool was validated on two different engines: a multi-cylinder engine with post-oxidation technology with single and multiple injections, and a high-performance engine. The first engine was tested using two different fuels: normal gasoline and synthetic fuel. The obtained results have demonstrated the effectiveness and reliability of the developed algorithm, as well as the reduction of time wasted during this crucial phase of project development.

In conclusion, the regulator tool developed in this thesis proves to be a valuable asset for conducting accurate and efficient 3D-CFD simulations of internal combustion engines. By automatically reaching target values and reducing the need for manual adjustments, the tool streamlines the simulation process, ultimately saving time and increasing productivity. Therefore, the regulator tool not only improves the accuracy of the simulation but also provides substantial advantages to the overall development phase.

Acknowledgments

Grazie a *tutti* coloro che mi hanno accompagnato in questo straordinario cammino.

Ma soprattutto … (inquadra QR code)

Contents

Index of figures

Index of tables

Chapter 1

1 Introduction and State of Art

The main goal of this project is to create a real-time regulator tool that automates and streamlines 3D-CFD simulations, reducing time wastage while achieving accurate results. In the ever-evolving era of mobility, companies play a critical role in developing easily manageable tools that align with the European Commission's emission reduction targets. QuickSim [1], a 3D-CFD software developed at Forschungsinstitut für Kraftfahrwesen und Fahrzeugmotoren Stuttgart (FKFS) and Institut für Verbrennungsmotoren und Kraftfahrwesen (IVK) by Dr.-Ing. Marco Chiodi, is used to conduct these simulations. QuickSim's unique features, such as the use of specific combustion and heat exchange models, allow for the accurate simulation of internal combustion engines, from intake to exhaust, in a virtual test bench setting. However, the large amounts of data generated by QuickSim can be time-consuming to analyze, and the simulations must converge before reliable results can be obtained [2].

During the internship at FKFS, under the supervision of Dr.-Ing. Marco Chiodi and his team, the regulator tool was developed to address these issues. The main goal of the tool is to reduce the time required for simulations and to minimize the number of repeated simulations that must be conducted when an engine operating point has not been correctly simulated. The tool operates by automating the management of simulations, setting target values for key parameters, and applying control strategies to fuel injection.

The development of the internal combustion engine has been a major focus of technological development in the automotive industry in recent years. In this context, improving the efficiency of internal combustion engines is one of the most promising and cost-effective approaches to increasing highway vehicles' fuel economy. The use of 3D-CFD simulations has proven to be an effective method for optimizing engine design, and QuickSim's ability to conduct simulations in a virtual test bench setting further enhances its utility.

The regulator tool was validated using two different engines: a multi-cylinder engine with post-oxidation technology and a high-performance engine. Additionally, two different fuels were used during testing: normal gasoline and synthetic fuel. The results obtained demonstrated the effectiveness and reliability of the developed algorithm and the reduced time required for simulation. As a result, the regulator tool improves the accuracy of the simulation process and increases productivity during the development phase.

In summary, the development of the regulator tool provides a valuable asset for obtaining accurate results from 3D-CFD simulations of internal combustion engines in a shorter amount of time. The tool not only improves the accuracy of the simulation process but also increases the productivity of the development phase by reducing the time spent on manual adjustments.

The following chapters outline the development of the study and the techniques employed to develop the algorithm. Beginning with Chapter 1, it provides an overview of the current state of the internal combustion engine (ICE) and the efforts undertaken by governments and companies to strike a balance and prevent the fading of ICE technology.

In Chapter 2, the use of computational fluid dynamics (CFD) in the field of internal combustion engines is described, highlighting its effectiveness in obtaining results that closely resemble real-world conditions. In Chapter 3, the development of the algorithm and its integration with the QuickSim software [1] are described.

In Chapter 4, the algorithm is validated using a wide range of engines and fuel types to ensure its reliability and accuracy.

1.1 General aspects of internal combustion engines

An internal combustion engine (ICE) is a type of heat engine that converts the chemical energy of fuel into mechanical energy through the process of combustion. The engine typically consists of a fixed cylinder and a moving piston, where the combustion of fuel and air mixture occurs inside the cylinder. The fuel and air are mixed in the intake manifold and drawn into the cylinder during the intake stroke. During the compression stroke, the piston compresses the fuel-air mixture, raising its temperature and pressure. In a spark ignition engine, a spark plug ignites the mixture, causing it to burn and rapidly release heat. The resulting expansion of hot gases drives the piston downward, producing mechanical work. In a compression ignition engine, the fuel ignites spontaneously due to the high pressure and temperature of the compressed air. In the figure 1.1.1 below is showed a typical working cycle point out the different phases.

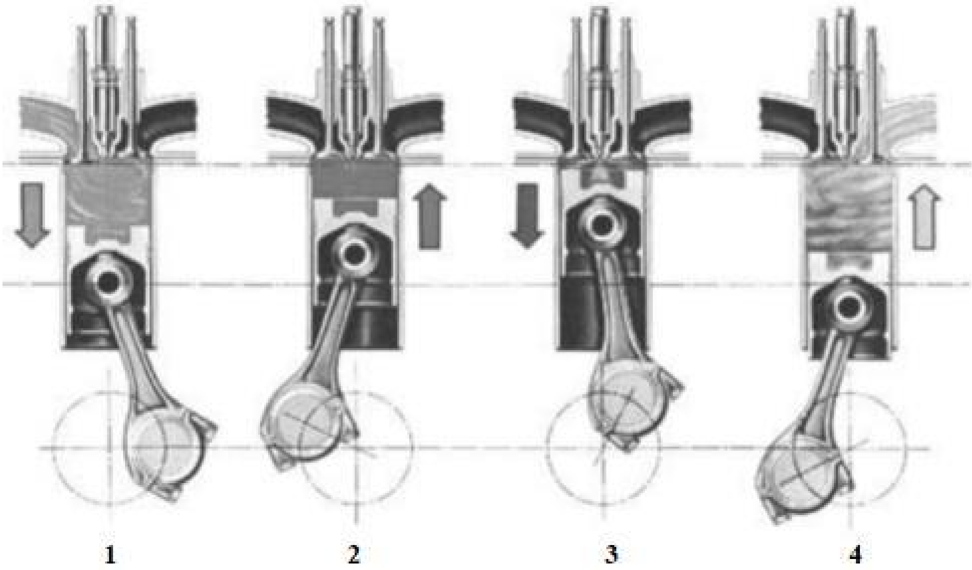


Figure 1.1.1 - *Phases of an internal combustion engine [5]*

All heat engines, including ICEs, can be characterized by a pressure-volume (p-V) diagram (see figure 1.1.2), which shows the variation of pressure and volume inside the cylinder during a complete engine cycle. The four-stroke cycle, which is commonly used in ICEs, includes the intake stroke, compression stroke, power stroke, and exhaust stroke. During the intake stroke, the piston moves downward, drawing in a fresh mixture of fuel and air. During the compression stroke, the piston moves upward, compressing the fuel-air mixture and raising its temperature and pressure. The power stroke is where the fuel-air mixture is ignited, producing a rapid expansion of hot gases that drive the piston downward, producing mechanical work. Finally, during the exhaust stroke, the piston moves upward, expelling the exhaust gases from the cylinder.

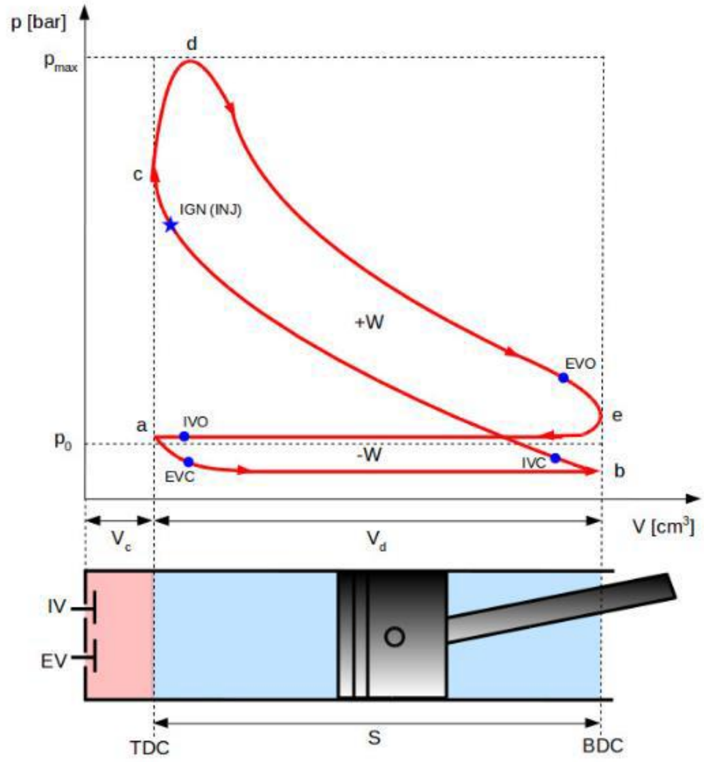


***Figure 1.1.2** – p-V diagram for a typical 4 stroke ICE [14]*

The performance of an ICE is determined by its efficiency, power output, and emissions. The efficiency of an ICE is determined by its thermal efficiency, which is the ratio of the net work produced by the engine to the heat input from the fuel. The power output of an ICE is determined by its torque and rotational speed, which depend on factors such as engine displacement, compression ratio, and combustion chamber design.

To evaluate the quality and the performance of the cycle in figure 1.1.2, a parameter called imep (indicated mean effective pressure pressure) is used and it's expressed by the formula 1.1.1:

$$IMEP = \frac{1}{V}\oint pdV$$

Equation 1.1.1 - *Indicated mean effective pressure*

Where:

- V is the volume displaced by the piston;
- p is the pressure inside the combustion chamber;
- dV is the infinitesimal volume displaced by the piston.

The mean indicated pressure is the hypothetical constant pressure that would produce the same amount of work as the actual variable pressure during the volume change equal to the engine displacement. In other words, it is the average pressure that would result in the same amount of work being produced as the actual varying pressure. This measure is useful for comparing the performance of different engines or for evaluating the efficiency of an individual engine over its entire operating range. The mean indicated pressure can be calculated using the measured pressure data from the engine's cylinder pressure sensors and is an important parameter in engine design and optimization.

The average effective pressure, commonly referred to as break mean effective pressure (BMEP), is a measure of the actual work produced by the engine per unit of engine displacement. The BMEP takes into account the engine's mechanical efficiency and is calculated by dividing the total torque produced by the engine by its displacement volume. The relationship between BMEP and IMEP is obtained through mechanical efficiency, which can be influenced by factors such as friction, combustion efficiency, and heat transfer losses. It is important to note that the BMEP is expressed in units of pressure.

$$BMEP = \frac{T}{V} 2\pi\epsilon$$

Equation 1.1.2 - *Brake mean effective pressure [16]*

Where:

- T is the engine torque;
- V is the engine displacement;
- ϵ is a frequency factor (one for 2-strokes, two for 4-strokes).

During an engine test, various performance parameters such as the air mass m_a and the fuel mass m_f can be measured. The air mass flow rate is typically measured using a mass flow meter located in the engine's air intake system, while the fuel mass flow rate is measured using a fuel flow meter located in the fuel supply system. These measurements allow for the calculation of other important parameters such as the air-fuel ratio. The air-fuel ratio is the ratio of the mass of air to the mass of fuel used in the combustion process and is an important factor affecting engine performance and emissions. It’s rapresent in the following formula provided:

$$\alpha = \frac{m_a}{m_f}$$

Equation 1.1.3 - *Air-to-fuel ratio*

Where:

- m_a air mass intaked for each cycle;
- m_f fuel mass injected for each cycle.

In addition to mass flow rates, other parameters such as engine speed, engine load, and exhaust gas temperature can also be measured during an engine test to provide a more comprehensive understanding of the engine's performance characteristics.

The stoichiometric air-to-fuel ratio α_s is the minimum quantity of air required to fully oxidize one unit mass of fuel. In other words, it is the ideal air/fuel ratio for achieving complete combustion of the fuel. However, in actual engine operation, the air/fuel ratio used can be greater, equal to, or smaller than α_s, depending on the specific operating conditions and design characteristics of the engine.

To account for these variations in air-to-fuel ratio, an important parameter for defining the mixture composition is lambda (λ).

$$\lambda = \frac{\alpha}{\alpha_s}$$

Equation 1.1.4 - *Equivalence air-to- fuel ratio*

Lambda is defined as the ratio of the actual air/fuel ratio to the stoichiometric air-to-fuel ratio. A lambda value of 1 indicates a stoichiometric mixture, while values greater than 1 indicate a lean mixture (excess air) and values less than 1 indicate a rich mixture (excess fuel).

As mentioned in the notes by prof. StefanoD'ambrosio [14], the lambda value is an important factor affecting engine performance and emissions. For example, a lean mixture can result in higher fuel efficiency and lower emissions of carbon monoxide (CO) and hydrocarbons (HC), but may also lead to higher nitrogen oxide (NO_x) emissions. A rich mixture can improve engine power and reduce NO_x emissions, but can also result in lower fuel efficiency and higher emissions of CO and HC. Therefore, maintaining the proper air-to-fuel ratio is critical for optimizing engine performance and minimizing emissions.

One of the main challenges of an internal combustion engine (ICE) is to improve its efficiency, defined as:

$$\eta_g = \frac{P_u}{m_f H_i} = \frac{1}{bsfc \cdot H_i}$$

Equation 1.1.5 - *Fuel conversion efficiency*

Where:

- P_u is the engine power;
- H_i is the lower heating value;
- m_f is the fuel injected for each cycle;
- $bsfc$ is the brake specific fuel consumption.

The efficiency of an internal combustion engine (ICE) is typically measured as the ratio of the mechanical work produced to the thermal energy released by fuel combustion, common best values are η_g= 37-42 % for SI engine, η_g = 42-50 % for CI engines. As showed in the relation the fuel conversion efficiency is inversally proportional to the brake specific fuel consumption, defined as:

$$bsfc = \frac{\dot{m}_f}{P_u}$$

Equation 1.1.6 *Brake specific fuel consumption*

Where:

- $\dot{m}_f$ is the fuel mass flow rate injected for each cycle;
- P_u is the power produced by the engine in each cycle.

The equation 1.1.6 is useful to monitor how efficiently the engine uses the supplied fuel.

If we consider 100% of the energy supplied by the fuel, the losses in an internal combustion engine (ICE) are typically shared in the following ways:

1. Exhaust losses: More than 20% of the energy is lost through the exhaust gases due to the exhaust valve opening before the end of the expansion stroke. This loss is due to the fact that the high-pressure gases are not fully utilized before they are expelled from the cylinder.
2. Cooling losses: Approximately 20% of the energy is lost due to engine cooling. Engine cooling is necessary to prevent overheating and failure of the engine components, and is caused by the technological limits of the materials used and the need to maintain lubricating oil characteristics.
3. Pumping and friction losses: These losses are due to the work required to move the air and fuel mixture into the cylinder (pumping losses) and the friction between the moving engine components (friction losses). Throttle valves and auxiliary components also contribute to these losses.

When we consider all of these losses, the overall efficiency of the engine is typically around 30% [9].

The main losses of an ICE are described in the figure 1.1.3:

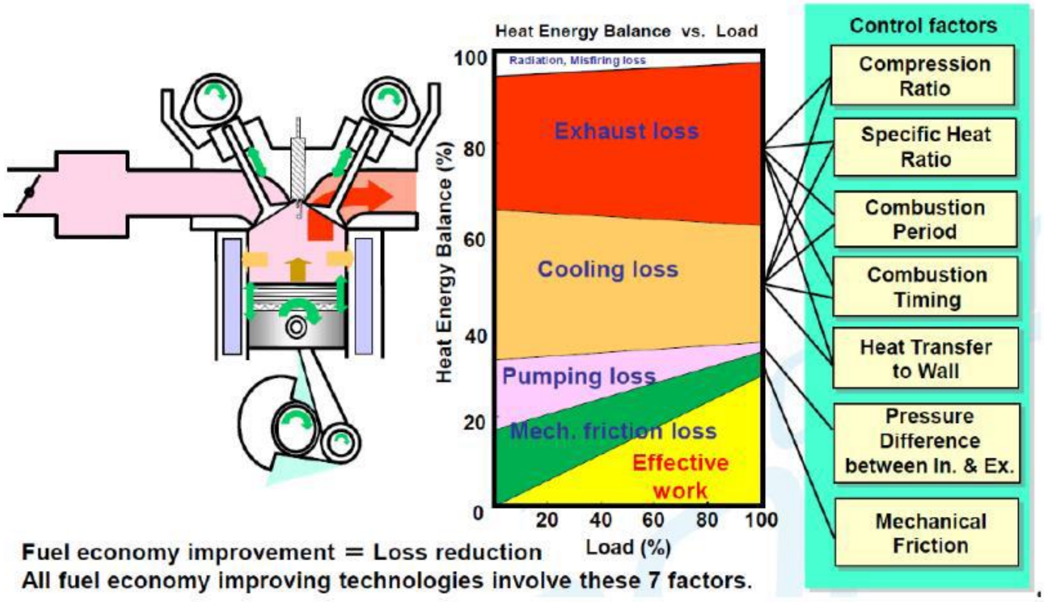


Figure 1.1.3 – *Thermal losses of an ICE [9]*

1.2 Combustion in SI engines

The primary fuel used in ICEs is a hydrocarbon mixture of C_xH_y, which contains varying percentages of carbon and hydrogen. The combustion process in a spark-ignition (SI) engine is achieved through a series of elementary binary chemical reactions that occur on the flame front. When the spark plug delivers activation energy to the surrounding premixed charge, it creates a flame core that propagates inside the cylinder at a certain speed. The parameter that represents the rate at which oxidation reactions occur on the flame front is known as the laminar flame speed (SL). A correct evaluation of SL can be carried out through the analysis of specific kinetic schemes. In literature, there are various correlations available to determine the value of SL. One of the most commonly used correlations is:

$$S_L(p_u, T_u, x_r, \Phi) = S_{L,0}(\Phi) \cdot (\frac{p_u}{p_0})^{\alpha_s} \cdot (\frac{T_u}{T_0})^{\beta_s} \cdot EGR_{factor}$$

Equation 1.2.1 – *Laminar flame front speed*

Where:

- p_u is the unburned zone pressure;
- T_u is the unburned zone temperature;
- T_0 ambient temperature in standard conditions;
- p_0 ambient pressure in standard conditions;
- EGR_{factor} takes into account the slowing of the flame front due to the presence of the inert gasses;
- α_s and β_s depends on the equivalent ratio;
- $S_{L,0}$ also depends on the equivalent ratio and has formulations characteristic of each;
- Φ is the equivalent ratio;
- x_r residual inert gas fraction.

The speed of flame propagation is an important consideration in combustion processes, and it is influenced not only by the effective speed but also by turbulence phenomena. Turbulence is necessary to complete the combustion process quickly and efficiently. In spark-ignition (SI) engines, two types of flow motions occur within the combustion chamber: swirl and tumble. Tumble motion, in particular, is utilized in SI engines to create a swirling motion of the air-fuel mixture during the intake stroke, which improves combustion efficiency and reduces emissions. This is achieved through a combination of intake port design, valve timing, and piston shape. The goal of tumble motion is to create a homogeneous mixture, which is essential for efficient and clean combustion. At the top dead center (TDC), the collapse of the tumble motion generates turbulent motions that play a critical role in maximizing the burning speed by increasing the flame front area. This phenomenon can be described mathematically by the following formula:

$$S_T = S_L \cdot \frac{A_T}{A_L} = S_L \cdot \Xi$$

Equation 1.2.2 – *Turbulent flame front speed*

Where:

- S_T is the turbulent flame front speed;
- S_L is the laminar flame front speed;
- A_T is the turbulent area of the flame front;
- A_S is the laminar area of the flame front;
- Ξ is the wrinkling factor, which measures the corrugation of the flame front. His evaluation may be done through specific models like fractal geometry.

The generation of organized motion fields during the intake stroke, described by the Mean Kinetic Energy (MKE), is essential for the efficient operation of internal combustion engines. The MKE is a measure of the average kinetic energy of the inducted air, and its level is influenced by factors such as the geometry of the intake system, the valve timing, and the engine speed. During the compression stroke, the MKE is conserved, leading to an increase in air temperature and pressure [9].

In the figure 1.2.1 is reported the trend of the kinetic energy as a function of the intake and compression phases.

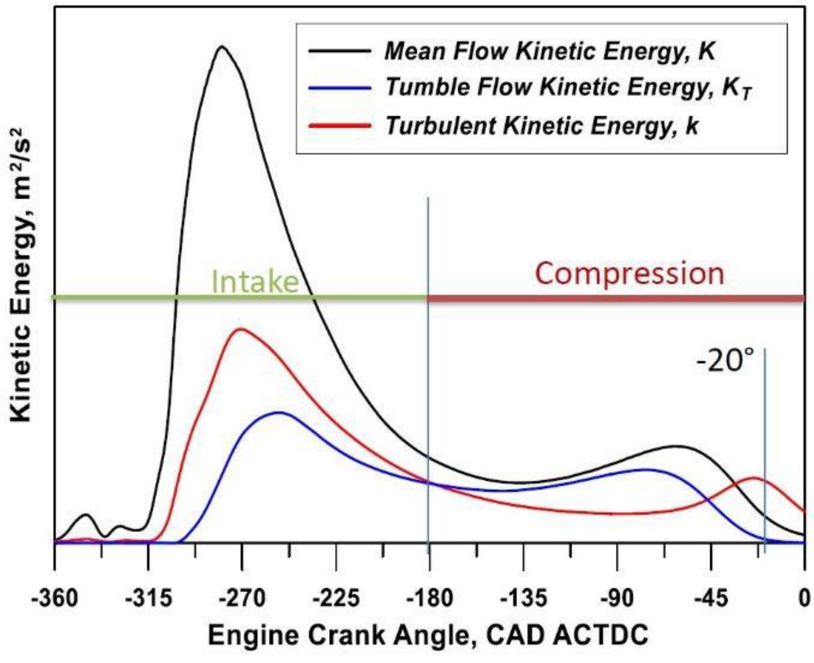


Figure 1.2.1 - *MKE, TKE and Tumble motion during intake and compression phases [9]*

1.3 Polluntant emissions in SI

The formation of pollutants in SI engines is a complex process that is closely related to the combustion process. Automotive engines are one of the major sources of urban air pollution, discharging a significant amount of pollutants into the atmosphere at the end of each operation cycle. These pollutants can be harmful to both plant and animal life and can disrupt the natural ecological balance.

The most common fuels used in ICE are hydrocarbon mixtures, which have varying percentages of carbon and hydrogen. Ideally, the combustion process should only produce carbon dioxide (CO_2) and water (H_2O) as byproducts. However, the reality is that the combustion process also produces harmful substances such as hydrocarbons (HC), nitrogen oxides (NO_x), carbon monoxide (CO), and particulate matter.

The formation of pollutants in spark ignition (SI) engines is a complex process that is closely linked to combustion. These engines are one of the primary sources of urban air pollution, discharging a range of harmful pollutants into the atmosphere at the end of each operation cycle. The most common fuels used in SI engines are mixtures of hydrocarbons (C_xH_y) with varying percentages of carbon and hydrogen. Ideally, the combustion products of these fuels are and H_2O, but the combustion process also generates other harmful substances such as unburned hydrocarbons (HC), carbon monoxide (CO), and nitrogen oxides (NO_x), as shown in Figure 1.3.1.

- Unburned HC emissions have several sources, including elements of the charge that enter crevices in the cylinder during the compression phase or go into the lubrication oil film and are released during the expansion phase. HC formation can also occur during the scavenging phase, when some of the charge goes directly from the intake to the exhaust. Additionally, incomplete combustion or quenching can cause HC formation in certain load conditions where temperature and pressure are insufficient for the flame front to advance. The fuel/air ratio of the mixture plays a significant role in determining the level of HC emissions, as it affects the velocity and completeness of the combustion process. HC concentrations decrease rapidly when leaner mixtures are used, until the combustion quality becomes poor;
- Carbon monoxide (CO) is formed as a product of incomplete oxidation of hydrocarbons due to combustion with a rich mixture or partial dissociation at the high temperatures of the CO_2 already formed. CO levels are strongly related to the air/fuel equivalent ratio, with concentrations being high when the mixture is rich and decreasing rapidly with an impoverishment of the mixture. However, since SI engines typically operate at stoichiometric conditions, CO emissions must be controlled through after-treatment processes;

- Nitrogen oxides (NO_x) are formed at high temperatures in the reaction zone, where temperatures exceeding 2000 K can cause the thermal dissociation of N_2 and O_2 molecules into atoms that combine to form NO_x. The Zeldovich model is the most common model used to predict NO production, which strongly depends on local oxygen concentration and temperature. Lambda affects the combustion temperature, which reaches its maximum with rich mixtures (λ=0.9). As the mixture becomes leaner, the decrease in gas temperature is initially compensated by the rise in oxygen concentration. Figure 1.2.1 clearly shows that NOx emissions have their maximum level for lean mixtures.

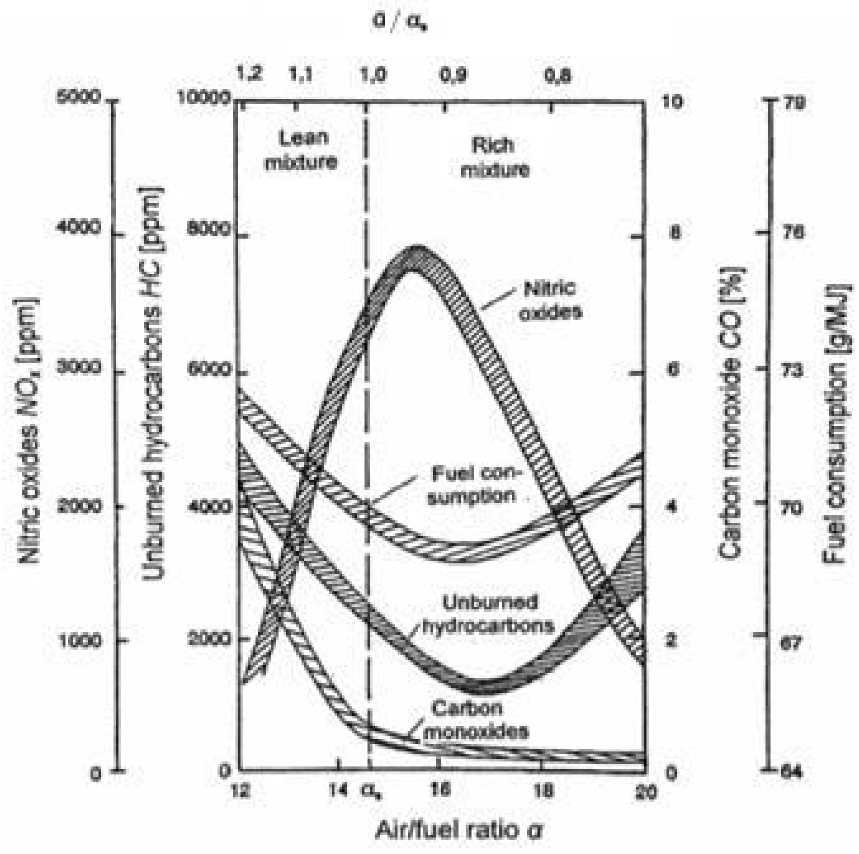


Figure 1.3.1 - *Dependence of pollutant production on equivalence ratio [5]*

Controlling pollutants in SI engines is a significant challenge, and the most efficient way to clean the exhaust gases is to force them through a catalytic converter. In the following paragraph will explain the typical characteristics of one of the most commonly used systems in spark ignition (SI) engines to reduce emissions.

1.4 Aftertreatment systems for SI engines

Aftertreatment systems are devices used in spark ignition engines to reduce exhaust emissions. These systems are necessary because even with the best combustion technology, there will always be some unburned fuel and other emissions that need to be cleaned up before being released into the atmosphere. As mentioned in the previous paragrapher, a spark-ignition engine produces mainly three pollutants. The most common aftertreatment system used in SI engines is the Three-Way Catalytic Converter (TWC).

The three-way catalyst (TWC) is a crucial technology used in spark ignition engines to reduce emissions of harmful pollutants such as carbon monoxide (CO), nitrogen oxides (NO_x), and hydrocarbons (HC). The TWC operates by oxidizing CO and HC into CO_2 and water and reducing NO_x to N_2 and O_2.

In the figure 1.4.1 are reported the traces of the TWC efficiency with respect to the three pollutants as a function of the air-fuel ratio. The definition of catalyst efficiency is reported in the equation 1.4.1:

$$\eta_X = \frac{\dot{m}_{X,in} - \dot{m}_{X,out}}{\dot{m}_{X,in}}$$

Equation 1.4.1 - *Catalyst conversion efficiency*

Where X is a generic chemical substance.

As it reported in the figure 1.4.1, the efficiency of the Three-Way Catalyst (TWC) in converting NO_x is highest on the rich side of the air-fuel ratio, and gradually decreases as the mixture becomes leaner. On the other hand, the TWC's conversion efficiency for HC and CO is highest on the lean side, and shows a rapid decrease as the mixture becomes richer. This is due to the fact that TWCs require a specific air-fuel ratio to function optimally, known as the stoichiometric point, where there is just enough oxygen to react with all of the fuel.

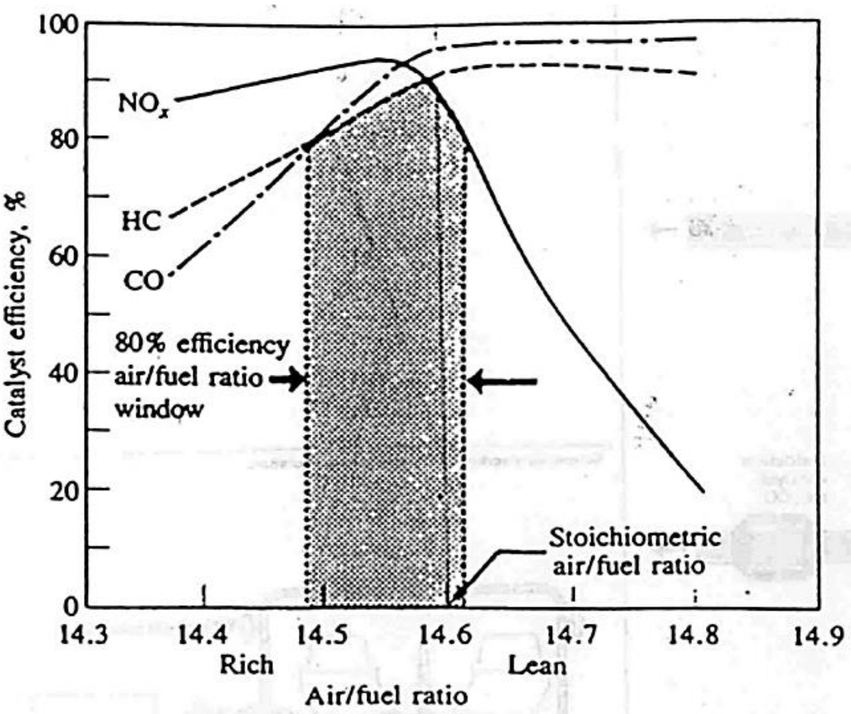


Figure 1.4.1 - *Catalyst efficiency [13]*

Another important parameter in the design and optimization of catalytic converters in order to ensure efficient pollutant conversion is the, so called, light-off temperature. It used to describe the minimum temperature required to initiate the catalytic reaction. However, due to the gradual increase of the reaction rate, this definition is not very precise. A more precise definition of the light-off temperature os the temperature in which the conversion efficiency reaches 50%, which is frequently referred to as T50. When comparing the activities of different catalysts, the most active catalyst will be characterized by the lowest light-off temperature for a given reaction. However, it is important to note that increasing the temperature in some catalyst systems may only increase the conversion efficiency up to a certain point. Further temperature increase may cause a decrease in the catalyst conversion efficiency, as show in the figure 1.4.2, due to other competing reactions that deplete the concentrations of reactants or thermodynamic reaction equilibrium constraints. In such cases, it is possible to define a catalyst window as the range of temperature in which the conversion efficiency is higher than a given value, such as 50% in the example given.

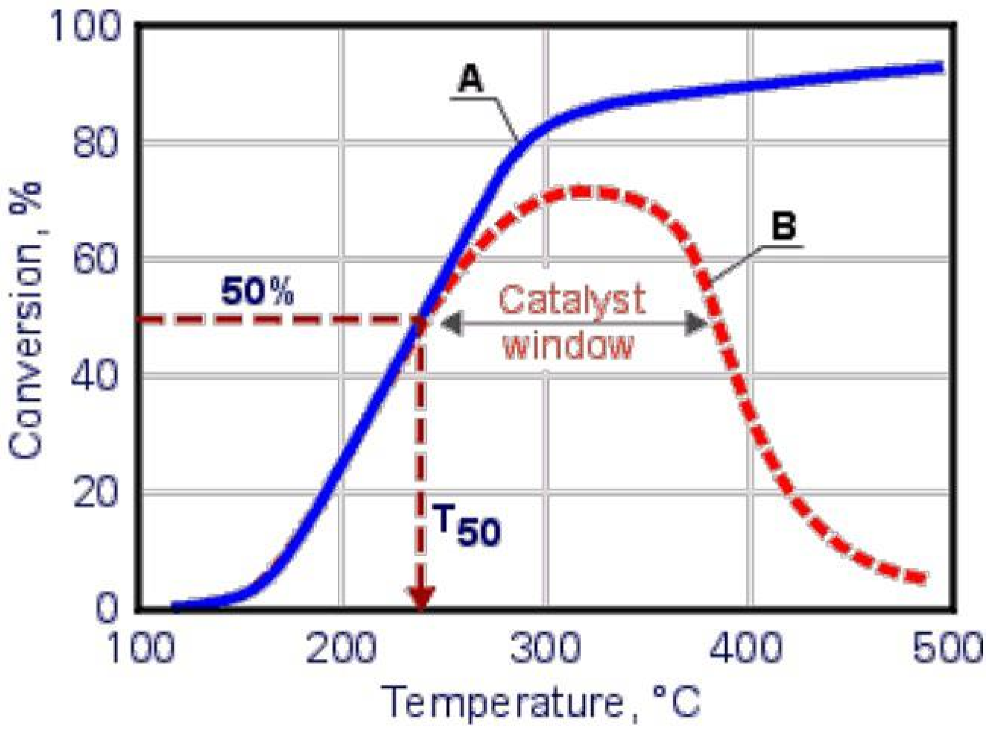


Figure 1.4.2 - *Conversion efficiency as a function of the operating temperature [13]*

- *Catalyst layout*

There are two main configurations for catalysts in spark ignition engines: close-coupled and underfloor. The choice of configuration depends on the requirements that the catalyst system must meet. The close-coupled layout shortens the warm-up phase, but imposes limitations on the size of the catalyst due to space constraints in the engine compartment. Moreover, having a large pre-catalyst in a close-coupled layout can lead to longer warm-up times.

The catalyst system has two contrasting requirements to fulfill:

1. A fast warm-up, which is beneficial when the catalyst is small and as close as possible to the engine. This aspect is crucial during the urban portion of the type-approval cycle;
2. A sufficiently low space velocity at peak exhaust flow rate (high-speed, high-load engine operation), which is beneficial when the catalyst is large. This aspect is crucial during the extra-urban portion of the type-approval cycle.

For larger engine displacements, a double catalyst layout is necessary to meet these contrasting needs. In this layout, a small pre-catalyst is used to rapidly warm up, and an underfloor main catalyst provides the additional residence time required for high-speed high-load operation. The main catalyst takes longer to reach the light-off temperature, but this is not a problem during the type-approval cycle, as the exhaust flow rate in the urban part of the driving cycle is low and the pre-catalyst can provide all the required conversion efficiency. In the figure 1.4.2 is shown the two different configurations.

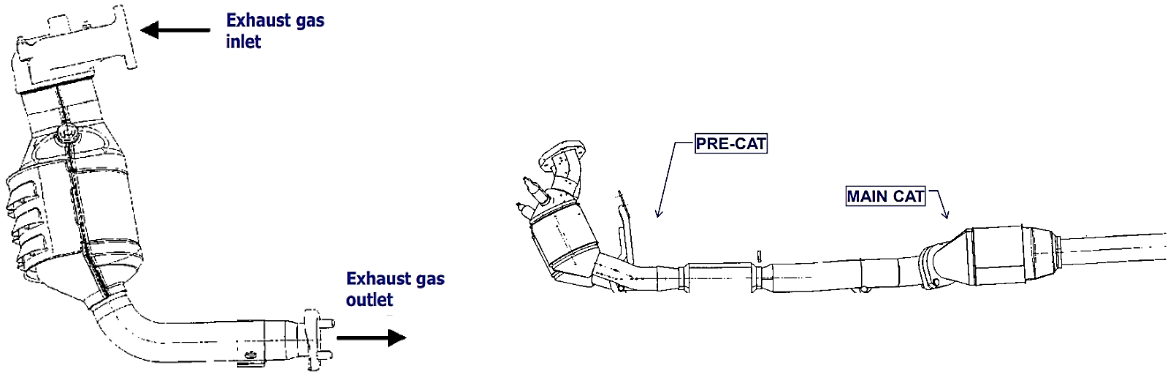


Figure 1.4.3 - ***Left***: *Close-coupled configuration,* ***Right***: *Pre-catalyst with underfloor catalyst [13]*

The Euro emissions standards have become increasingly stringent with each iteration, setting stricter limits on the emissions of pollutants such as nitrogen oxides (NO_x), particulate matter (PM), and carbon monoxide (CO). As a result, the design of catalytic converters has evolved to meet these more stringent requirements.

Prior to Euro 2, most vehicles utilized one or more underfloor catalysts, which were able to meet the relatively lenient emissions standards in place at the time. However, with the introduction of Euro 3 and subsequent standards, the need for a faster light-off time became crucial in order to meet the more stringent requirements.

It should also be noted that with the introduction of Euro 6, which mandates even stricter emissions limits, new technologies such as selective catalytic reduction (SCR) and gasoline particulate filters (GPFs) have been implemented to further reduce emissions [13].

1.5 Current era of mobility

The increasing stringency of exhaust emission regulations coupled with the need for a significant reduction in CO_2 emissions and competition from alternative powertrain technologies, has sparked discussions about the future of internal combustion engines. Historically, the primary focus of engine development was on maximizing performance. However, since the 1990s, the introduction of Euro regulations shifted the focus towards reducing pollutant emissions. This has led to a significant research effort by car manufacturers and oil companies to develop new engines and fuels that meet increasingly stringent emissions limits. For this reason, a new euro standard should be come out in the 2025 [13]. The Euro 7 is a future emission standard that is currently being developed buy the European Union (EU). It will be the latest and the most stringent emission standard for road vehicles in the EU. Euro 7 will introduce more stringent limits for (NO_x) and PM emissions, and for the first time, it will also introduce limits on ammonia (NH_3) emissions, which are a byproduct of modern selective catalytic reduction (SCR) systems, mentioned for Euro 6. The standard will also place limits on particulate number (PN) emissions, which are particularly important for gasoline direct injection (GDI) engines.

As the automotive industry continues to face pressure to reduce the environmental impact of internal combustion engines, the focus has shifted towards compliance with low-emissions regulations. In response to this pressure, car manufacturers and oil companies have made significant efforts to reduce harmful emissions by up to a quarter of their initial levels. However, with the need to limit CO2 emissions, it is now imperative to burn less fuel, thereby improving the engine's specific fuel consumption and efficiency.

In order to achieve this goal, internal combustion engines must be a component of a more complex propulsion system. Improving the engine's efficiency is the most important parameter to achieve this objective. While the internal combustion engine alone cannot achieve this goal, it can play a critical role in a more comprehensive propulsion system.

The current era of mobility is characterized by a transition phase in which no single solution exists as the optimal choice. The best solution depends on the consumer's needs and requirements.

Thus, the need to comply with low-emissions regulations and reduce CO2 emissions has spurred a significant research effort in the field of internal combustion engines, making it a highly motivated and active area of development. In the figure 1.4.1 is shown the trend of restrictions in the last two decades.

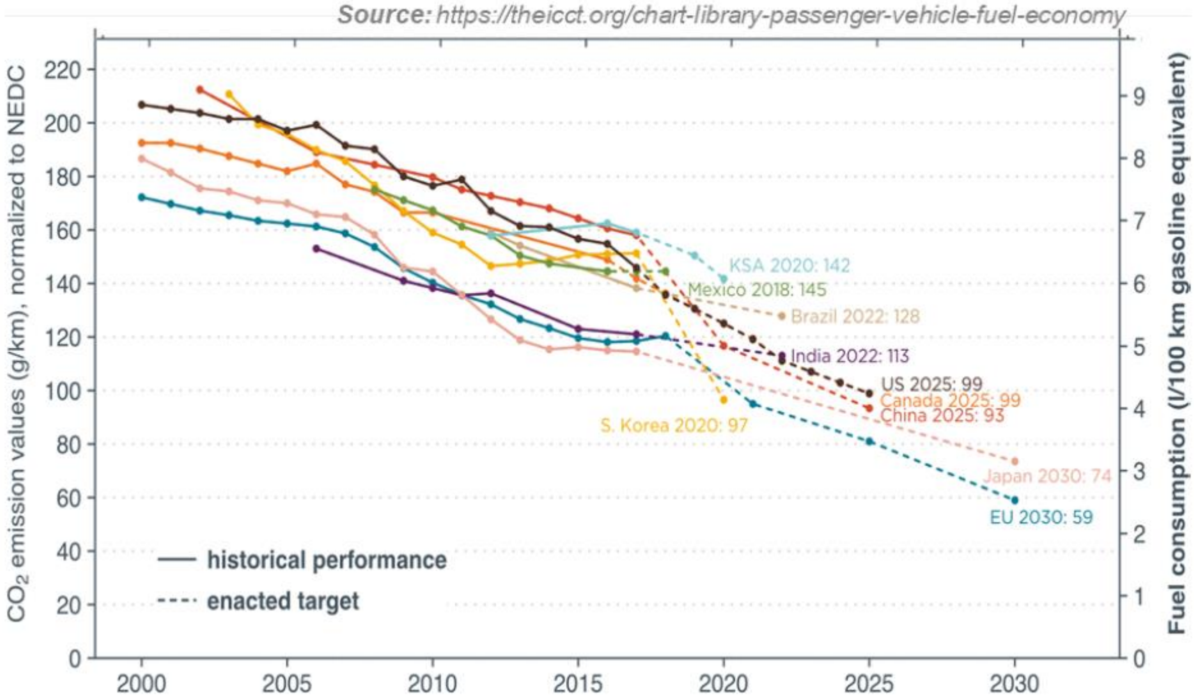


Figure 1.5.1 - *Euro CO2 standards [13]*

It is clear that the level of CO_2 emissions has decreased significantly over the years. In Europe, it is worth noting that the limit for CO_2 emissions for vehicles has decreased from 175 g/km in 2000 to approximately 120 g/km currently, with further reductions expected to bring it down to 60 g/km by 2030.

As point out from the figure 1.5.2., the trend in the automotive industry shows that the sales of gasoline and diesel engines are gradually decreasing, giving way to the development of gasoline DI hybrid, gasoline DI, E-vehicles, and fuel cell vehicles. It is becoming increasingly evident that the future of the internal combustion engine lies in hybrid vehicles, which combine an ICE propulsion system with a battery-electric drive system to enhance fuel consumption, emissions, and performance in comparison to conventional cars.

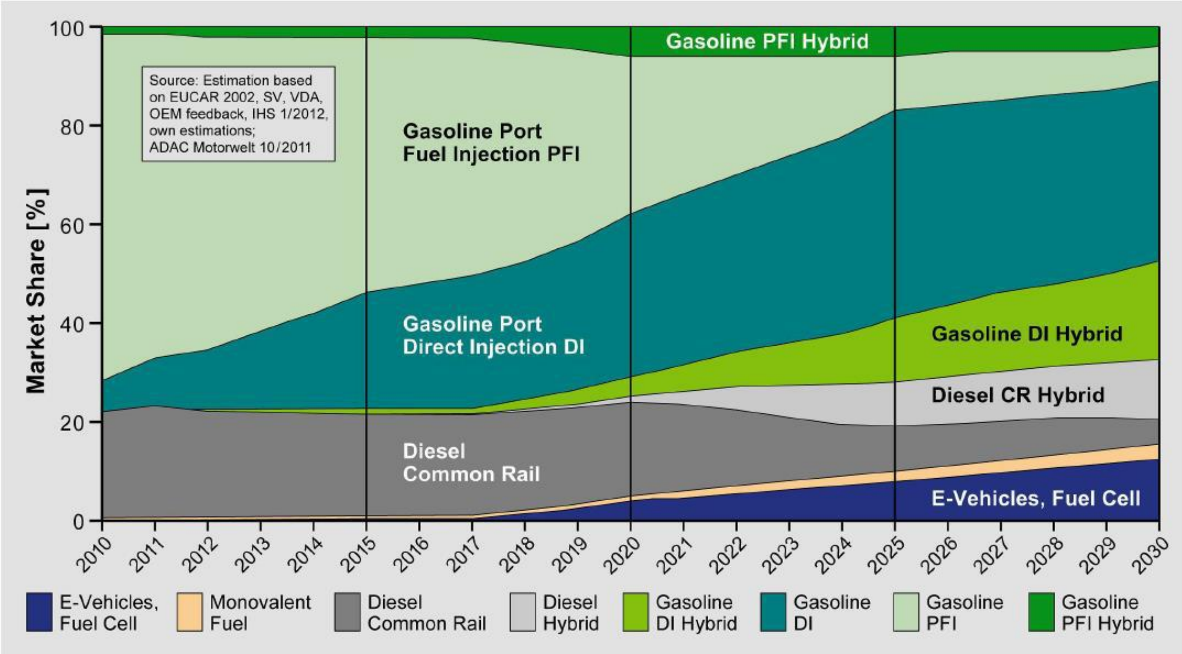


Figure 1.5.2 - *Market share of gasoline and Diesel engine [9]*

Hybrid vehicles are typically classified based on the power division between sources, which can operate in parallel, series, or both. They can also be classified according to their degree of hybridization, which includes full hybrids and mild hybrids. Full hybrids are capable of running on the engine, the batteries, or a combination of both, with different power division between sources that can operate in parallel, series, or both. Mild hybrids are essentially conventional cars with some degree of hybrid hardware, but with limited hybrid feature utilization, typically in the form of start-stop only or modest levels of engine assist or regenerative braking features.

The current situation of hybridization is depicted in figure 1.5.3. As of 2020, the most common solution is the Micro Hybrid, which is based on the use of Start and Stop systems that reduce fuel consumption. By 2024, there will be an increase in Mild Hybrid sales, in which a battery pack and an electric motor provide part of the energy needed to complete the driving cycle in addition to the primary thermal engine. We will eventually reach Full-Hybrid with various configurations, including series, parallel, and regenerative braking. However, this technology will lead to complications in energy flow management.

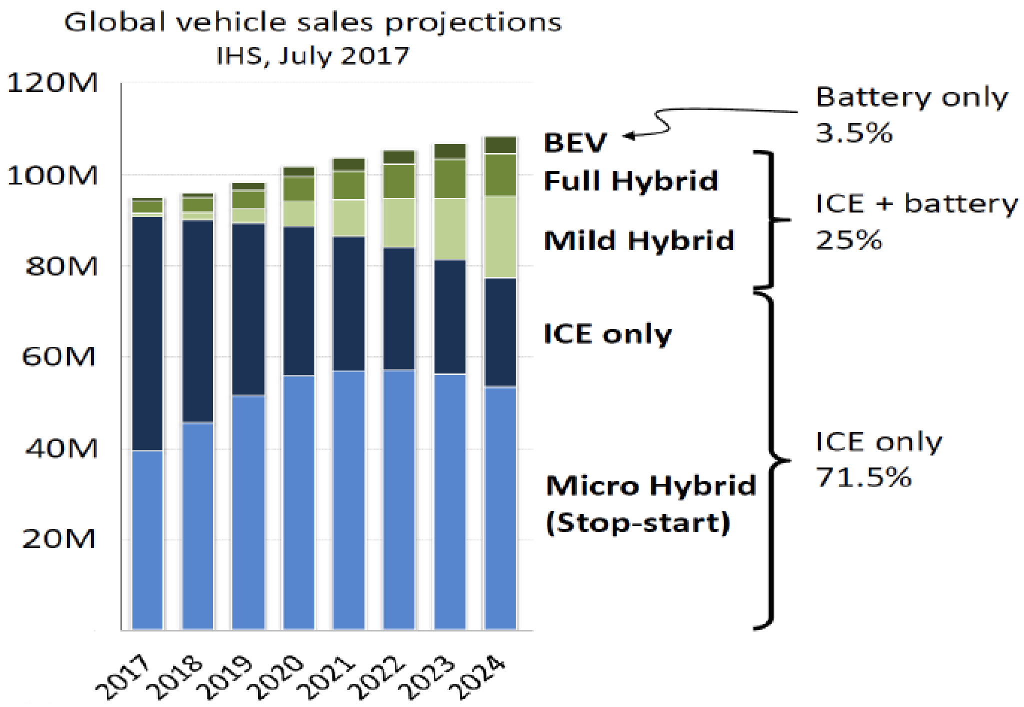


Figure 1.5.3 - *Market share of hybrid and electric vehicles [9]*

To fully assess the environmental impact of hybrid or electric vehicles, it is important to conduct a Life Cycle Assessment (LCA) which provides a detailed analysis of the entire production process, from raw materials extraction to end-of-life disposal. LCAs can help to identify the true environmental impact of a propulsion system. For instance, in the case of hybrid or electric systems, the batteries used in these vehicles contain substances that can be difficult to dispose of at the end of their life cycle. The LCA analysis typically includes several stages, as shown in Figure 1.5.4. These stages include defining the scope and goal of the study, inventory analysis, impact assessment, and interpretation. In the inventory analysis stage, the inputs and outputs of each process involved in the production and use of the propulsion system are identified, including raw material extraction, manufacturing, and use. The impact assessment stage involves assessing the environmental impacts of each process, which can include factors such as air pollution, water pollution, and greenhouse gas emissions. The final stage involves interpreting the results and identifying areas where improvements can be made to reduce the environmental impact of the propulsion system.

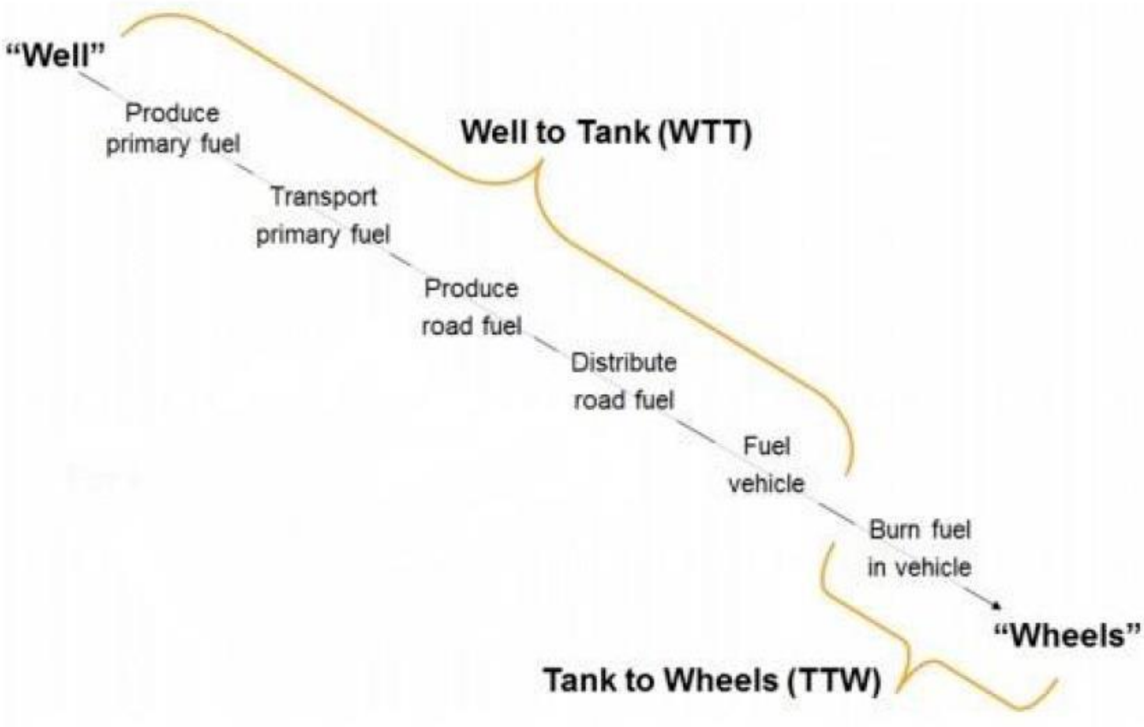


Figure 1.5.4 - *Life cycles analysis of a vehicle [9]*

The figure provides a global view of the energy problem by analyzing all energy conversions starting from the primary energy source up to the energy available at the wheels, which will be significantly lower and highly dependent on the driving profile. It is evident that all these conversions contribute to the production of CO_2.

The first step in this process is the "Well to Tank" analysis, which considers the primary energy sources, including fossil fuels, solar energy, and nuclear energy. Of these three sources, only fossil fuels contribute to the production of CO_2. It is essential to understand the efficiency of extraction and storage of this energy, as different forms of energy require various storage systems (such as tanks, batteries, etc.) characterized by different performances.

Finally, "Tank to Wheels" is the analysis of converting the energy stored in the above-mentioned systems into mechanical energy, which propels the vehicle. This step is crucial for understanding the overall energy efficiency of the propulsion system and its environmental impact. It is also important to note that the efficiency of the conversion process depends heavily on the driving profile and conditions [9].

1.6 Pathways to decarbonization in European Union (EU)

As mentioned by *Peter Mock* and *Sonsoles Diaz* [16], the European Union's (EU) climate and green growth strategy, known as the European Green Deal, aims to achieve net-zero greenhouse gas (GHG) emissions by 2050. In order to achieve this goal, the European Commission plans to propose revised carbon dioxide (CO_2) targets for new passenger cars and vans by mid-2021. The current targets require a 15% reduction in type-approval CO_2 emissions for new passenger cars by 2025, compared to 2021, and a 37.5% reduction by 2030. However, meeting these targets alone will not be sufficient to achieve the EU's envisioned economy-wide GHG reduction target of at least 55% by 2030, as well as the EU Green Deal target for transport of -90% GHG emissions by 2050. Therefore, the EU sectoral policies and regulations need to be strengthened and revised to meet the necessary targets. The international council on clean transportation [16] has investigated the CO_2 reduction potential on four different scenarios:

- *Adopted Policies scenario*

The Adopted Policies scenario envisions a future in which manufacturers only meet the established targets of -15% by 2025 and -37.5% by 2030 for reducing CO_2 emissions, without making any additional efforts to exceed these levels through the deployment of electric vehicles. This scenario leaves the remaining potential of internal combustion engine (ICE) vehicles untapped and results in stagnation of electric vehicle market shares from 2030 onwards. In other words, this scenario assumes that manufacturers will simply comply with existing regulations and not take any further action to reduce emissions, such as investing in electric vehicle technology. As a result, the potential benefits of transitioning to more sustainable forms of transportation will not be fully realized, and progress towards reducing greenhouse gas emissions will be slower than what is necessary to meet climate goals.

- *Lower Ambition scenario*

The Lower Ambition scenario envisions an increase in the current CO_2 reduction targets to -20% by 2025 and -50% by 2030, along with the implementation of a new target of -70% by 2035. By doing so, this scenario aims to unlock some of the remaining potential of internal combustion engine (ICE) vehicles, resulting in an annual CO_2 reduction of around 1%. Additionally, it promotes the growth of the electric vehicle market by increasing the market share of battery electric vehicles (BEVs), with the goal of having them account for approximately half of new car sales by 2035.

- *Moderate Ambition scenario*

In the Moderate Ambition scenario, the target reductions for CO_2 emissions are significantly higher than in previous scenarios, with a goal of -30% by 2025, -70% by 2030, and ultimately, a complete reduction of 100% by 2035. This requires vehicle manufacturers to make significant advancements in the reduction of ICE emissions, targeting a 4% reduction annually in the Worldwide Harmonized Light Vehicle Test Procedure (WLTP) between 2021 and 2025. Achieving these targets involves reducing vehicle mass and transitioning to mild hybrid vehicles, thereby tapping into most of the remaining potential of ICEs.

In addition, plug-in hybrid electric vehicles (PHEVs) are phased out and replaced by more cost-efficient battery electric vehicles (BEVs) faster than in previous scenarios. This helps to increase the market penetration of BEVs, with a target of around 50% of new car sales by 2030 and full market saturation by 2035. This ambitious approach could potentially drive significant progress towards the goal of achieving net-zero greenhouse gas (GHG) emissions by 2050, as set out in the European Green Deal. However, this scenario would require significant investments and efforts from both industry and policy makers alike.

- *Higher Ambition scenario*

In the Higher Ambition scenario, the target for CO_2 reduction is set at -100% in WLTP by 2030. This is achieved by rapidly transitioning towards battery electric vehicles (BEVs) and fully exploiting the remaining potential of internal combustion engines (ICEs) in the transition years. This scenario requires a significant increase in the deployment of BEVs, with a market penetration of about 75% by 2030 and 100% by 2035. In addition, vehicle manufacturers must focus on reducing the weight and increasing the efficiency of BEVs to achieve the necessary CO_2 reductions.

In the figure 1.6.1 is summarized the market share evolution for the four different scenarios, it's worth to notice how in the higher ambition scenario the internal combustion engine goes out by 2025.

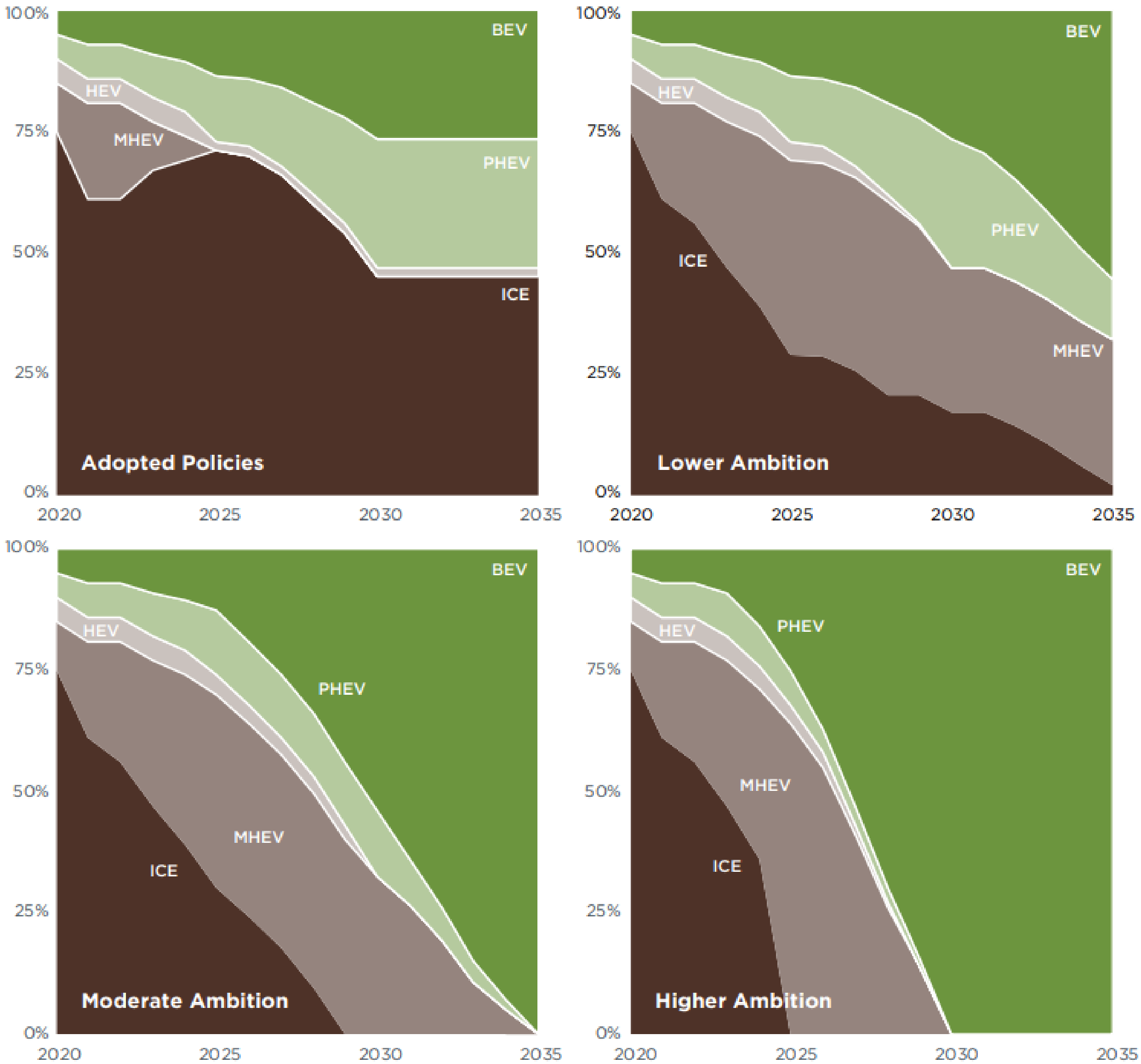


Figure 1.6.1 – *New registrations technology market share evolution for the four main policy scenarios [16]*

1.6.1 Zero emissions from new cars and vans by 2035

After the European agreement on the Green Deal and the CO_2 reduction targets set by the European Union, the European Parliament met in October 2022 to vote on a regulation that imposes a ban on the production and registration of automobiles and vans powered by internal combustion engines starting from 2035. This decision has significant implications for both the automotive industry and consumers.

The goal of this regulation is to reduce the CO_2 emissions produced by the transportation sector, which represent about 25% of total emissions in the European Union. The decision to ban the production and sale of vehicles with internal combustion engines by 2035 is a significant step towards achieving the European Union's goal of reducing CO2 emissions by 55% by 2030.

The automotive industry will face several challenges to comply with the new regulation. They will need to invest in new technologies, such as the production of high-capacity batteries, the construction of charging stations, and the implementation of energy management solutions. Additionally, they will need to modify their supply chains to meet the demand for electronic materials and reduce their dependence on fossil fuels.

To meet the growing demand for electric vehicles, the automotive industry will need to produce high-capacity batteries at affordable prices. High-capacity batteries are essential for increasing the range of electric vehicles and making them more competitive than internal combustion engine vehicles. The construction of fast-charging stations will be another critical aspect to ensure the spread of electric vehicles, as the ability to quickly recharge vehicles is essential to ensure their practicality and daily usability.

Furthermore, the automotive industry will need to work to reduce its dependence on fossil fuels. The use of electronic materials and batteries requires the production of rare materials such as lithium and cobalt, which are currently extracted from mines around the world. The automotive industry will need to invest in alternative technologies for the production of electronic materials, such as recyclability and circular economy, to reduce the environmental impact of their production.

On the other hand, consumers will need to adapt to the new reality of electric vehicles. They will need to purchase vehicles equipped with advanced technologies, which require a completely different understanding and driving experience. Additionally, they will need to address the issue of vehicle charging, which will require the installation of charging stations [16].

1.7 Advanced techniques for mitigating CO_2 emissions in SI-engines

The previous paragraph highlighted the production of CO_2 as the primary issue with ICEs. To address this problem, new technological solutions have been developed in recent years to improve engine efficiency and minimize fuel consumption. This chapter will discuss some of the most important techniques for achieving these goals. So, how exactly can engine efficiency be improved? In the figure 1.7.1 is reported an overview of the possible short-term solutions. As highlighted, differences solutions can be adopted.

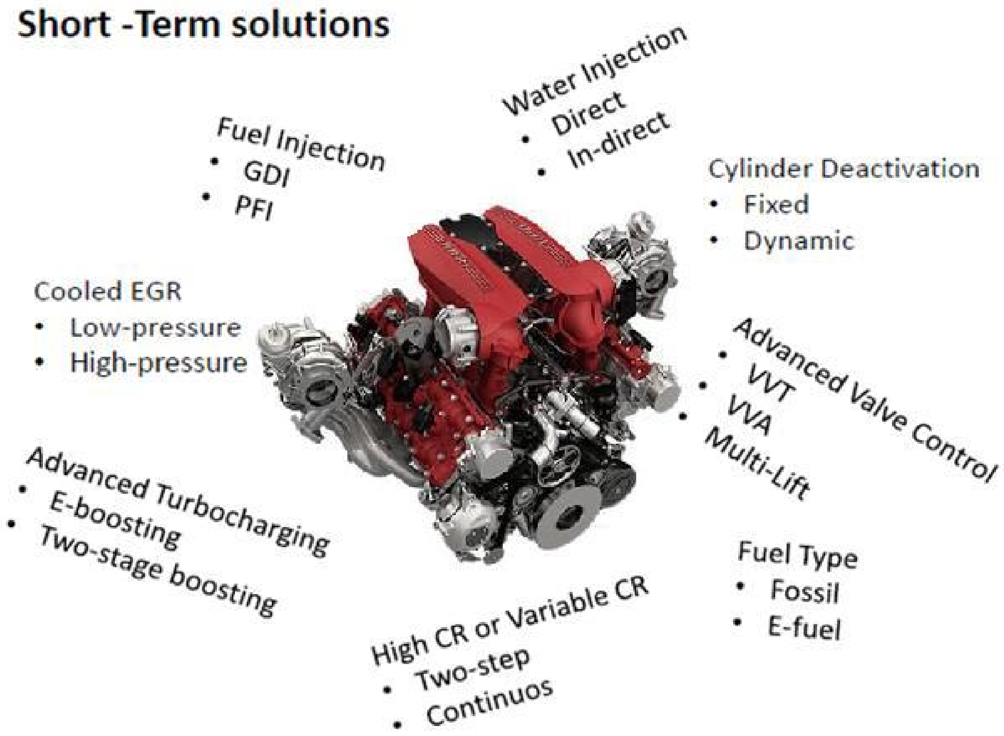


Figure 1.7.1 - *Short-Terms solutions to improve ICE efficiency [9]*

The evolution of the engine over time has witnessed numerous technological advancements aimed at enhancing its performance in accordance with current regulations. Initially, the focus was on increasing power output and improving fuel consumption, leading to solutions such as turbocharging, downsizing, and the adoption of higher compression ratios and variable valve actuation systems. However, these solutions caused an increase in in-cylinder pressure and temperature, resulting in a high knock tendency. To overcome this, it was necessary to delay spark timing and enrich the mixture at high load, which led to a decrease in efficiency. This in turn created new challenges in achieving high efficiency across the entire engine map.

1.7.1 Downsizing and Turbocharging

Usually, the internal combustion engine works most of the time in his low-efficiency zone, the figure 1.7.2 shows the operating point in green and yellow on the engine map.

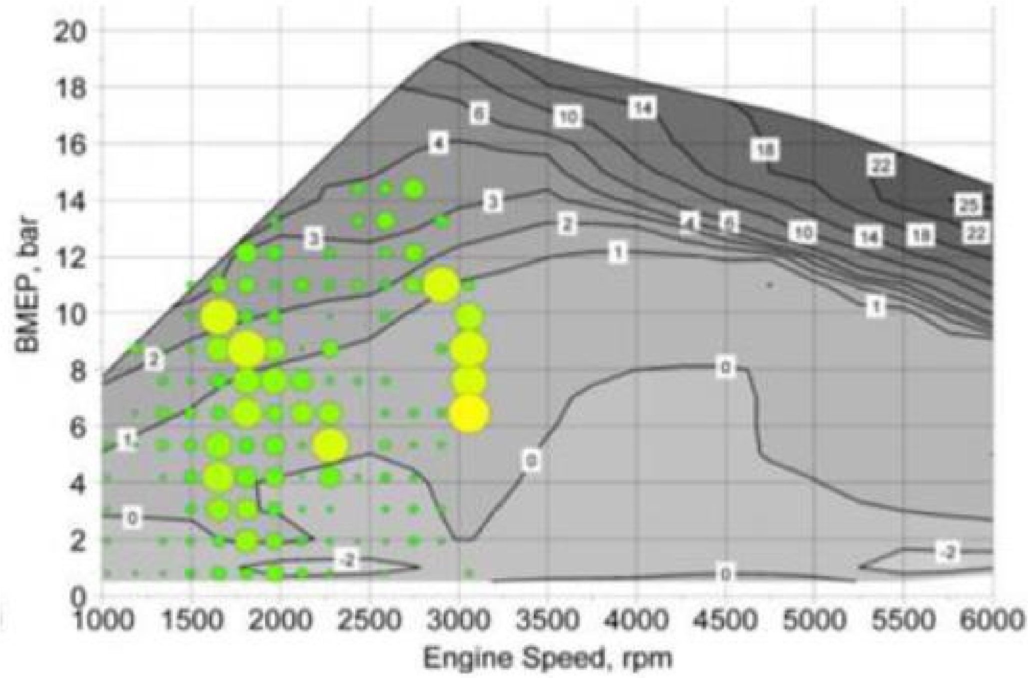


Figure 1.7.2 – *Engine Map [9]*

The term "downsizing" refers to the replacement of a larger naturally aspirated engine with a smaller, forced-induction engine on the same vehicle with the same transmission. The main goal of this strategy is to operate the engine more efficiently at part loads, resulting in better overall efficiency during driving. Turbocharging is then used to maintain the same performance level as the vehicle with the full-size naturally aspirated engine. Therefore, combining downsizing and turbocharging allows for improved fuel consumption with a smaller engine while maintaining the same performance through the assistance of the turbocharger.

The improvement in terms of fuel consumption is mainly related to an improvement of the mechanical efficiency and of the thermofluid-dynamics efficiency thanks to the fact that the engine is operated at higher loads.

Downsizing is typically combined with gasoline direct injection (GDI) technology to operate at higher efficiency operating points, while also increasing power density through turbocharging. However, downsizing can cause a delay in torque at low rpm due to a low mass flow rate through the turbine, resulting in low boost pressure. This turbo-lag effect can lead to poorer dynamic behavior compared to a naturally aspirated engine.

1.7.2 Variable Valve Actuation (VVA)

Conventional engines typically have a fixed valve timing, meaning that the angles of valve opening and closing do not change with engine speeds and loads. As a result, valve timing can only be optimized for a specific range of speed and loads. However, advanced technologies have been developed to address this limitation and optimize valve timing across the engine map, leading to improved performance and efficiency.

In detail, there are two generations of variable valve timing systems. The first-generation mechanism is able to change only the timing, as shown in the figure 1.7.3. In this mechanism, the lift laws for intake and exhaust remain the same, but they can be shifted earlier or later than TDC. This technology offers several advantages, such as:

- Anticipated IVC at low RPM to avoid backflow of the charge in the intake channel;
- Delayed IVC at high RPM to exploit the inertia of the air in the intake channel and improve the volumetric efficiency.

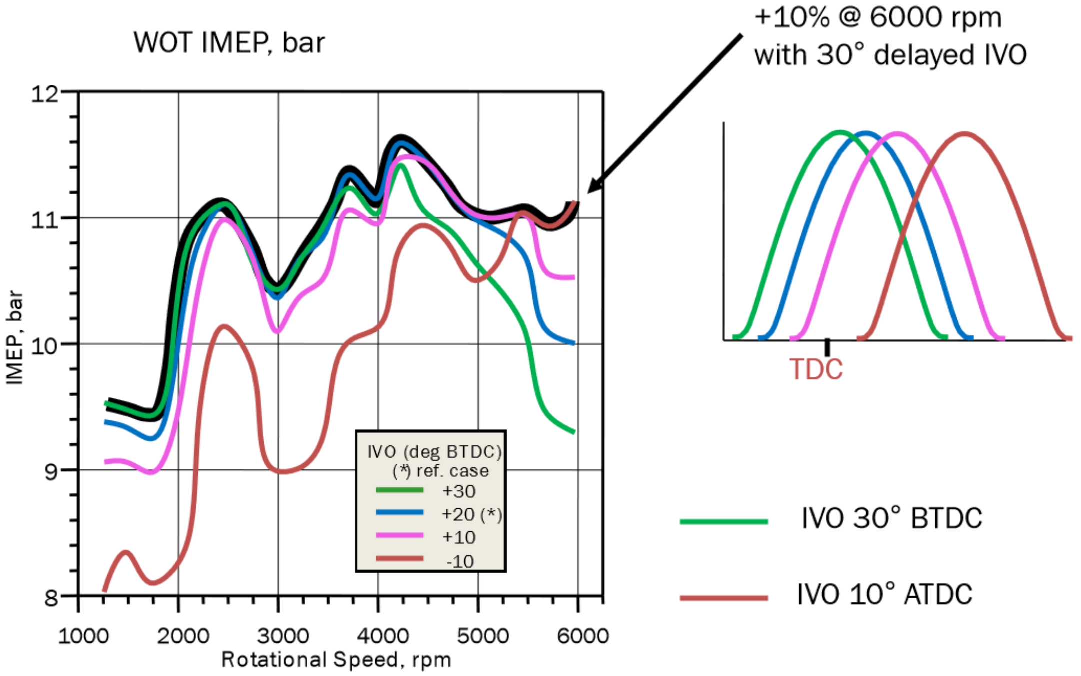


Figure 1.7.3 – *Effects of VVT on the IMEP [9]*

The second-generation mechanism, known as Variable Valve Actuation (VVA), not only changes the timing of the valve but also its lift, as shown in the figure 1.7.4. With this technology, two different working cycles can be achieved. The Early Intake Valve Closure (EIVC) results in the Miller cycle, while the Late Intake Valve Closure (LIVC) produces the Atkinson cycle.

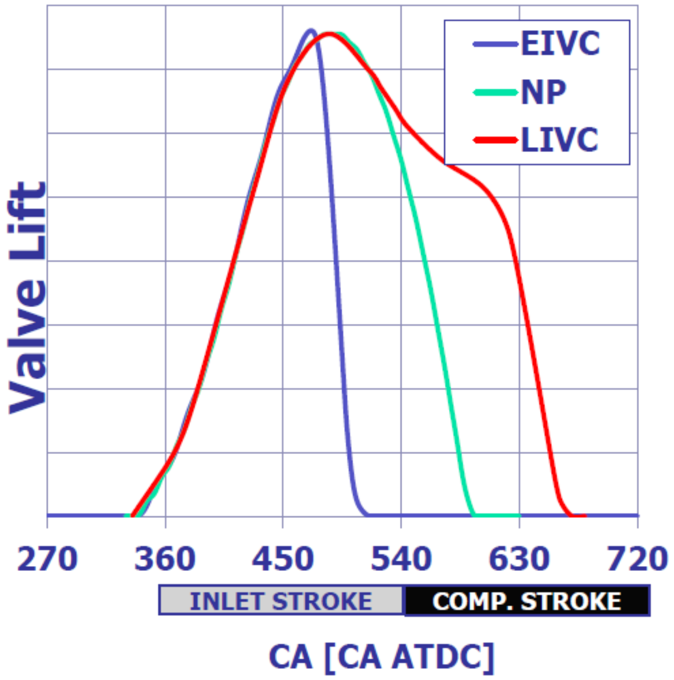


Figure 1.7.4 – *EIVC and LIVC [13]*

The Miller cycle, achieved with EIVC, is a modified version of the Otto cycle where the intake valve is closed earlier than usual, resulting in a shorter effective compression stroke. This reduces the amount of air in the cylinder and increases the expansion ratio, improving thermodynamic efficiency. The Miller cycle is particularly useful in engines with turbocharging, as it reduces the work required to compress the intake air.

The Atkinson cycle, achieved with LIVC, is another modification of the Otto cycle where the intake valve is closed later than usual, resulting in a longer effective expansion stroke. This decreases the amount of air in the cylinder and reduces the compression ratio, but increases the expansion ratio even further, resulting in improved efficiency. The Atkinson cycle is often used in hybrid engines, as it sacrifices power output for improved fuel efficiency.

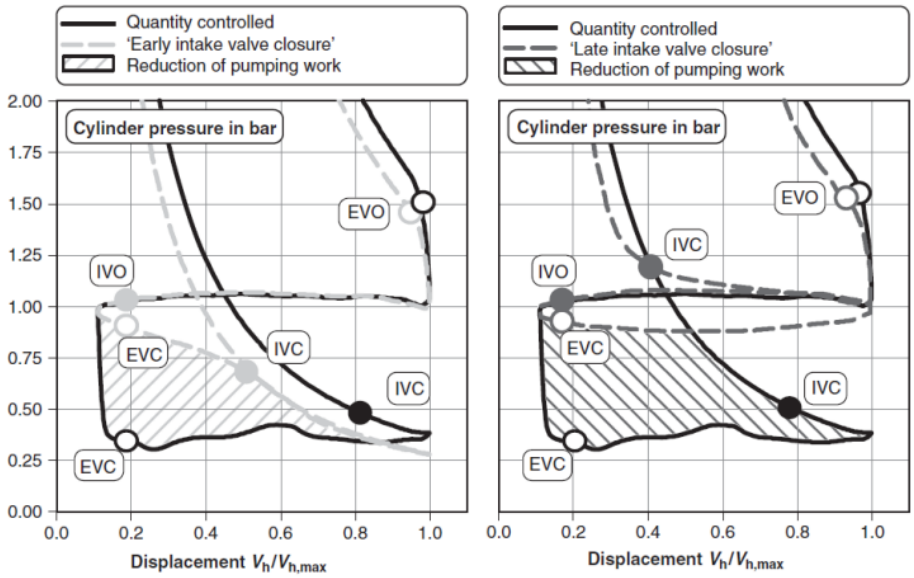


Figure 1.7.5 - ***Left:*** *EIVC* ***Right:*** *LIVC [14]*

Overall, VVA provides a way to optimize engine performance for different operating conditions, resulting in improved fuel economy and reduced emissions.

1.7.3 Gasoline Direct Injection (GDI)

Gasoline Direct Injection (GDI) is an injection system for gasoline engines that differs from the classic Port Fuel Injection (PFI) because the fuel spray is injected directly into the combustion chamber, as shown in Figure 1.7.6, rather than into the intake channel. This allows for more precise control over the fuel-air mixture, resulting in improved combustion efficiency and reduced fuel consumption. However, GDI systems also present some challenges, such as higher particulate emissions and increased susceptibility to engine deposits, which require proper maintenance and care.

Figure 1.7.6 - *Gasoline Direct Injection (GDI) [17]*

Thanks to the GDI system is possible to control the power of the engine changing the mass of fuel injected into the cylinder for each cycle, similar to what happens in the diesel engines, that give as result the reduction of pumping losses. In addition, the GDI system is able to control the charge mixture in two ways:

- At part load, it is possible to use the charge stratification. The mixture near the spark is kept close to stoichiometric in order to maximize laminar speed flame. As the distance from the spark increases, the mixture becomes leaner in order to save fuel;
- At full load, it is possible to achieve a homogenous mixture thanks to arly fuel injection.

However, GDI systems present some disadvantages, the most important one is the formation of the particulate matter. Another issue is the impingement phenomena, which is the impact of liquid fuel on the cylinder walls, resulting in the production of hydrocarbons and particulates.

Nowadays, this technology is widely used in new engines and when combined with other techniques, it is an effective way to reduce emissions and save fuel.

1.7.4 Post-oxidation

Post-oxidation is employed in a Three-Way Catalyst (TWC) system to reduce the light-off temperature, which refers to the minimum temperature required for the catalyst to start its efficient operation. Lowering the light-off temperature is important because it allows the catalyst to reach its optimal working conditions more quickly, leading to improved emissions control as mentioned in 1.4.

Post-oxidation contributes to reducing the light-off temperature in a TWC system through two main mechanisms:

- *Catalyst Coating*: The TWC is coated with a layer of precious metals, such as platinum, palladium, and rhodium, which act as catalysts for the oxidation reactions. These catalysts facilitate the oxidation of unburned hydrocarbons and carbon monoxide (CO) present in the exhaust gases. By promoting these oxidation reactions, the catalyst coating helps to generate heat, which raises the temperature in the catalytic converter. This heat generation aids in achieving the desired light-off temperature more rapidly.
- *Air-Fuel Ratio Control*: Achieving the optimal air-fuel ratio (stoichiometric ratio) is crucial for efficient TWC operation. The TWC is designed to work most effectively when the air-fuel mixture is near stoichiometry. Post-oxidation is used to ensure that any excess fuel or hydrocarbons remaining in the exhaust gases after combustion are oxidized in the presence of oxygen. This additional oxidation process raises the temperature in the catalytic converter, helping to reach the light-off temperature faster.

Recent studies have proposed different ways to achieve post-oxidation. The latest method involves a second air injection into the exhaust manifold, as shown in Figure 1.7.7. This is accomplished by using an air pump to inject fresh air in the proximity of the exhaust valves.

Secondary air injection requires careful design and testing to ensure that it is reliable and effective, but it has the potential to significantly reduce emissions and improve the performance of a wide range of engines. A detailed explanation of the secondary air injection is presented in [18].

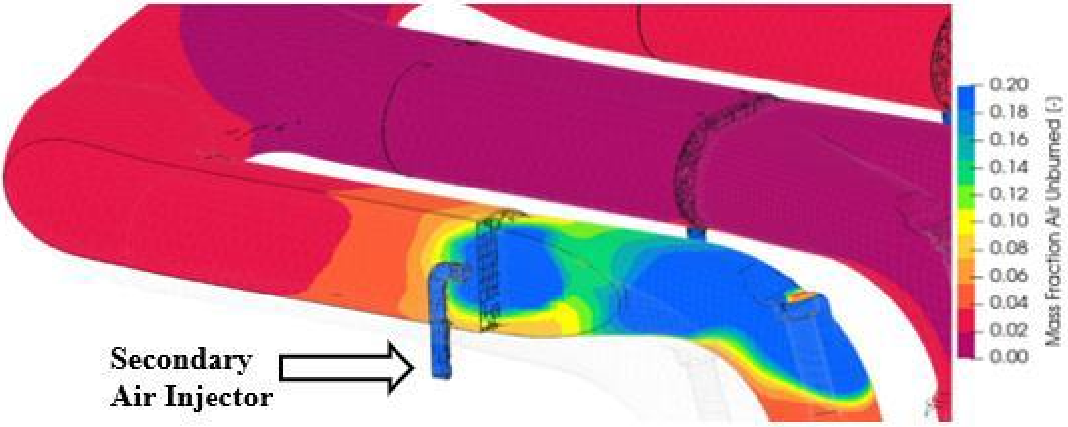


Figure 1.7.7 - *Secondary air injection [18]*

1.7.5 Alternative fuels

The concept of synthetic fuels involves the conversion of renewable energy sources into liquid or gaseous fuels that can be used in internal combustion engines. This can be achieved through various processes, such as power-to-liquid (PtL) or power-to-gas (PtG). These synthetic fuels have the potential to significantly reduce greenhouse gas emissions, as they can be produced using renewable energy sources, such as wind or solar power.

While electric powertrains are often seen as a solution to reduce emissions, there are limitations to their widespread adoption, particularly in terms of infrastructure and the availability of renewable electricity. On the other hand, synthetic fuels have the advantage of being easily transportable and can be distributed using the existing infrastructure, making them a more practical solution for reducing emissions in the transportation sector.

The development of innovative fuels, such as the one mentioned by the German automotive company, can lead to significant improvements in fuel efficiency and emission reduction, while still allowing for the use of conventional internal combustion engines. The fourth chapter of this work will provide a detailed analysis of the behavior of this fuel in a four-cylinder engine.

The main reasons why synthetic fuels are considered a viable mobility solution for the future are:

1. Timeliness: Synthetic and renewable fuels have advanced beyond basic research and are technically feasible to produce. Electricity generated from renewable sources can be used to extract hydrogen from water, after which carbon is added to create synthetic fuels such as petrol, diesel, gas, or kerosene. However, while the production process is technically feasible, there is a need to rapidly increase production capacity to meet demand.
2. Compatibility with existing infrastructure: Synthetic fuels can be used in existing vehicles and infrastructure, such as filling stations and distribution networks. This makes them a more

practical solution than transitioning to entirely new forms of mobility, such as electric vehicles, which would require significant investments in charging infrastructure.

3. Lower carbon emissions: Synthetic fuels can be produced from renewable energy sources, resulting in lower carbon emissions than traditional fossil fuels. They can also be blended with fossil fuels to reduce emissions, making them a more flexible solution for reducing carbon emissions in the transportation sector.
4. Energy security: Synthetic fuels can be produced domestically, reducing dependence on foreign oil imports and enhancing energy security.
5. Cost: The production costs of synthetic fuels are still high. However, it is expected that the cost of producing synthetic and renewable fuels will decrease as the production capacity increases and the cost of electricity from renewable sources falls. According to current studies, the price of these fuels is projected to reach 1.20-1.40 euro/litre (excluding excise duties) by 2030 and 1 euro/litre by 2050 [9].

1.7.6 Cylinder Deactivation

Cylinder deactivation, also known as variable displacement, is a technology used in internal combustion engines to improve fuel efficiency by deactivating some of the engine's cylinders under light load conditions, such as when cruising or idling. By doing so, the engine can operate more efficiently, reducing fuel consumption and emissions.

In modern engines, cylinder deactivation is often achieved by using hydraulic lifters that can deactivate the valves on selected cylinders. The engine control unit (ECU) determines which cylinders to deactivate based on the engine load and other factors, such as vehicle speed and throttle position. When a cylinder is deactivated, the intake and exhaust valves remain closed, and the fuel injector is shut off, preventing fuel from being injected into the cylinder.

Most engines with cylinder deactivation can switch between multiple modes, allowing for different numbers of cylinders to be deactivated, depending on the driving conditions. For example, some engines may deactivate two or four cylinders when cruising at a steady speed, while others may deactivate just one cylinder when idling.

Cylinder deactivation is used in a variety of modern engines, including V8 and V6 engines, as well as some four-cylinder engines. The technology has been adopted by many automakers, including General Motors, Honda, and Mercedes-Benz. It is often used in conjunction with other technologies, such as turbocharging and direct injection, to further improve fuel efficiency and reduce emissions.

1.8 Future Developments

In this subchapter, future developments for spark-ignition engines will be discussed. One way to improve engine efficiency is to enhance the combustion process, which can be achieved through various techniques. In the next paragraphers will be shown some techniques to improve combustion processes.

1.8.1 Turbulent Jet Ignition (TJI)

In Turbulent Jet Ignition (TJI), the main combustion chamber's mixture is ignited by a hot turbulent jet generated by burning a small quantity of stoichiometric or near-stoichiometric fuel/air mixture in a separate small volume pre-chamber. The higher pressure resulting from pre-chamber combustion pushes the combustion products into the main chamber in the form of a hot turbulent jet, which then ignites the ultra-lean premixed fuel/air in the main chamber. Compared to a conventional spark plug, the hot jet has a much larger surface area, leading to multiple ignition sites on its surface, which can enhance the probability of successful ignition and cause faster flame propagation and heat release, as shown in the figure 1.8.1.

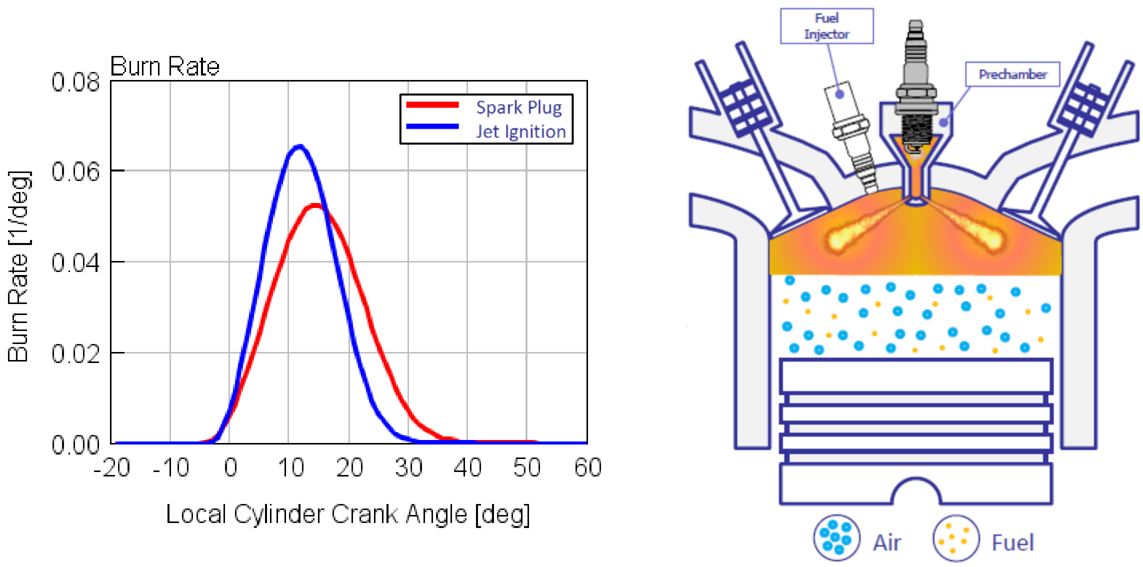


Figure 1.8.1 - ***Left:*** *Burn rate* ***Right:*** *TJI scheme [13]*

The faster combustion process enabled by Turbulent Jet Ignition (TJI) may allow for ultra-lean operation, which could lead to nearly complete elimination of in-cylinder NOx emissions and significant improvements in efficiency and fuel economy. TJI involves two possible ignition mechanisms, depending on the diameter of the pre-chamber nozzles: jet ignition and flame ignition.

Jet ignition produces a jet of hot combustion products, which extinguishes the pre-chamber flame when passing through the orifice. Flame ignition, on the other hand, produces a jet of wrinkled turbulent flames, which consist of incomplete combustion products containing flames. As the orifice diameter is increased, the ignition mechanism tends to switch from jet ignition to flame ignition (figure 1.8.2).

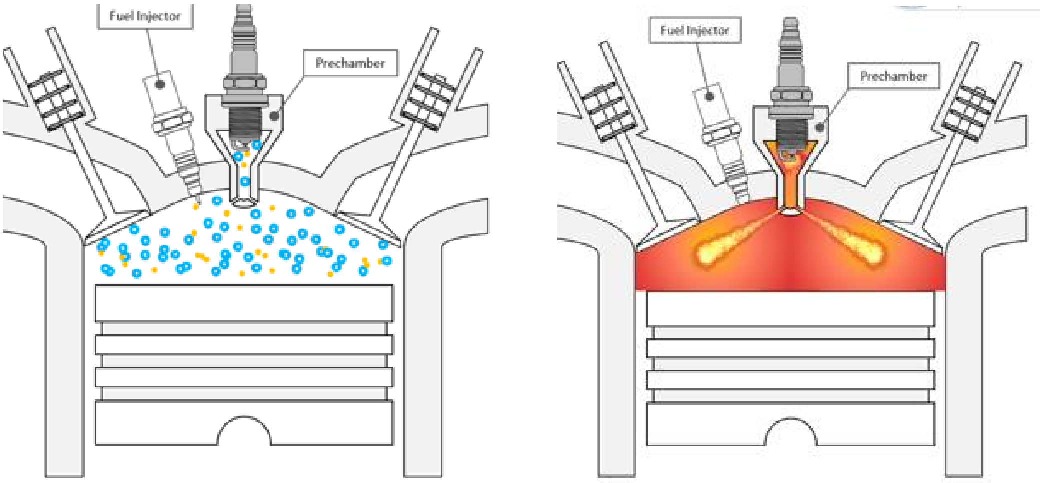


Figure 1.8.2 - ***Left:*** *Jet ignition* ***Right:*** *Flame ignition [13]*

Typically, jet ignition is preferred. When the nozzle diameter is small enough, the pre-chamber flame extinguishes while passing through the nozzle due to the high stretch rate and heat loss through the walls. As a result, the jet entering the main chamber contains only hot combustion products. Very little intermediate species and radicals are present in the jet.

Additionally, main chamber ignition does not occur as soon as the hot jet penetrates into the main chamber. The hot jet first accelerates and then decelerates. Main chamber ignition, however, always takes place during the jet deceleration process. Furthermore, ignition usually occurs from the side surface of the hot jet at a location downstream of the nozzle exit [13].

1.8.2 Spark-Controlled Compression Ignition (SCCI) and Homogeneous Charge Compression Ignition (HCCI)

Homogeneous Charge Compression Ignition (HCCI) and Spark-Controlled Compression Ignition (SPCCI) are combustion modes that aim to achieve higher efficiency and lower emissions compared to conventional spark-ignition engines. In HCCI, the air/fuel mixture is homogeneous and auto-ignites due to high compression temperatures, while in SPCCI, a spark plug ignites a small portion of the mixture to initiate combustion.

HCCI engines operate in a narrow range of air/fuel ratios, and the combustion is controlled by the amount of fuel injected and the compression ratio. The combustion is characterized by a rapid pressure rise and a low-temperature heat release, resulting in low NOx emissions. HCCI has been primarily used in experimental and research engines due to its limitations in terms of controllability and operating range.

SPCCI engines, on the other hand, operate similarly to conventional spark-ignition engines at low loads and switch to compression ignition at high loads. A spark plug ignites a small amount of the mixture, creating a flame kernel that propagates and initiates combustion. The timing of the spark plug and the amount of fuel injected control the combustion process. SPCCI has been implemented in production engines, such as Mazda's Skyactiv-X, to achieve higher efficiency and lower emissions.

Both HCCI and SPCCI have their challenges and limitations, such as difficulty in controlling combustion, achieving high load operation, and controlling emissions. However, they show promise in achieving higher efficiency and lower emissions, and further research and development are ongoing to improve their viability in production engines.

Figure 1.8.3 shows a diagram of the typical combustion processes used in an engine. The y-axis represents the local equivalence ratio, while the x-axis represents the temperature. It is worth noting that there are two regions where soot and NOx are formed and how HCCI and SPCCI can avoid them.

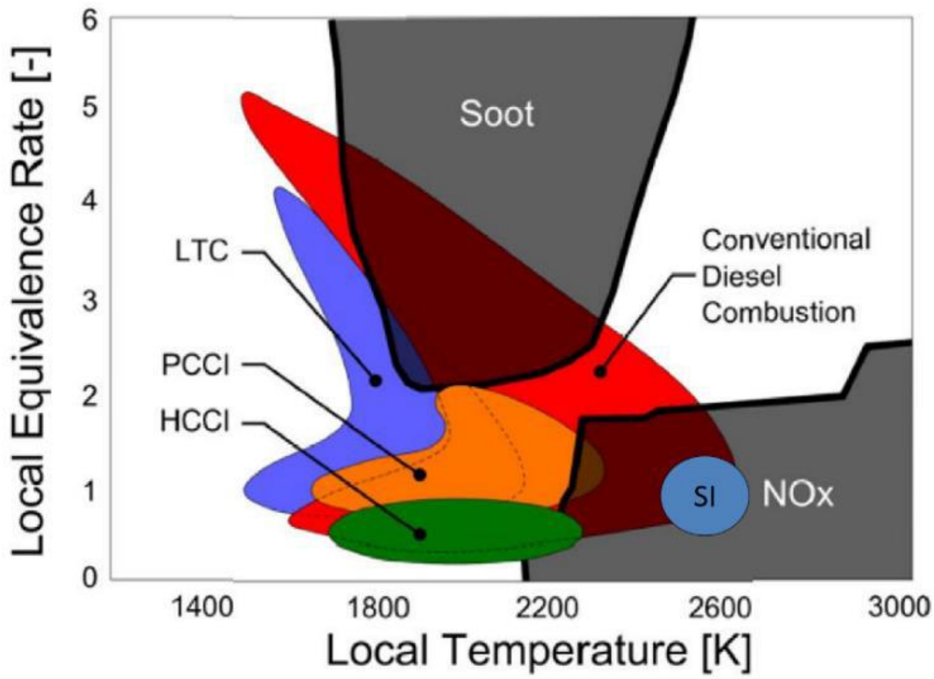


Figure 1.8.3 - *T- ϕ diagram of pollutants [9]*

1.8.3 Multiple injections

In GDI engines, fuel is injected directly into the engine combustion chamber instead of the intake manifold as in indirect injection engines. This allows for better control over the amount of fuel entering the combustion chamber, resulting in improved engine efficiency and reduced emissions.

An alternative ways described by the research conducted in the University of Michigan [20] can be used to reduce the engine emissions.

Multiple injections are used to distribute the amount of fuel injected into the combustion chamber over multiple moments during the combustion cycle, instead of in a single injection. This allows for more efficient and complete combustion, reducing emissions of nitrogen oxides (NO_x) and particulate matter.

The number and timing of injections depend on engine specifications and operating conditions. In general, injections can be divided into three categories: pre-injection, main injection, and post-injection.

Pre-injection occurs just before the main injection and serves to prepare the combustion chamber for the subsequent combustion, increasing turbulence and improving air-fuel mixing. This results in more efficient and complete combustion during the subsequent injection.

The main injection provides most of the fuel needed for combustion. This injection can be divided into multiple injections to ensure better fuel distribution in the combustion chamber and even greater combustion efficiency.

Finally, post-injection occurs after the main injection and serves to reduce emissions of NOx and particulate matter. Post-injection cools the exhaust gases, reducing the temperature in the combustion chamber and limiting the formation of NO_x. It also provides enough fuel to burn any particulate matter that was not burned during the main combustion.

In summary, multiple injections in GDI engines allow for more efficient and complete combustion, reducing emissions [20].

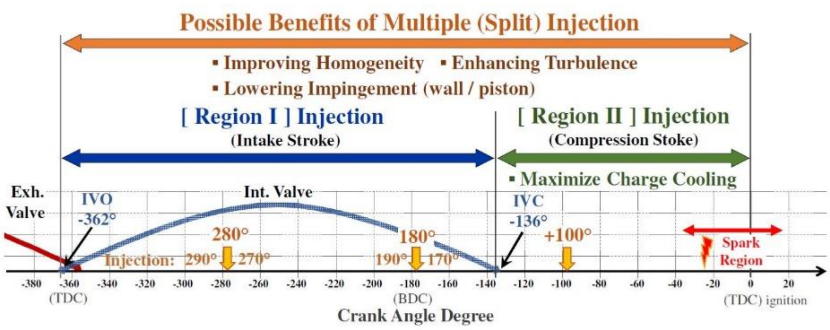


Figure 1.8.4 - *Possible Multiple Injections Benefit [20]*

Chapter 2

2 3D-CFD Simulation of Internal Combustion Engine

As described in Chapter 1, engine development approaches can be divided into two main areas: experiments and simulations. In the early days of internal combustion engines, development was based only on experimental testing due to limited computing power. However, as computational models with increasing levels of detail and accuracy were developed over the years, simulations have become crucial in the research and development of internal combustion engines. Simulations allow for the prediction of engine behavior in terms of emissions, efficiency, and performance, and can significantly reduce engine development times and costs. Additionally, simulations can evaluate a range of parameters that may not be experimentally detectable.

Some examples of engine simulations include Computational Fluid Dynamics (CFD) simulations, which allow for the analysis of fluid flow and heat transfer in the engine, and Finite Element Analysis (FEA) simulations, which can be used to study the engine's mechanical behavior and stress distribution. These simulations can be applied to a wide range of engines, including gasoline, diesel, and hybrid electric powertrains.

2.1 Overview of 3D-CFD Simulation

Internal combustion engines will undoubtedly continue to play a crucial role in transportation and energy generation, especially with the ongoing advancements in virtual modeling technologies. However, to meet the ever-increasing emission regulations while ensuring affordability, reliability, drivability, and excitement, future engine architectures will become more complex. Managing this complexity will require a proper combination of various development tools, including the powerful combination of one-dimensional (1D) and three-dimensional computational fluid dynamic (3D-CFD) simulations. Furthermore, a single-cylinder test bench, as illustrated in figure 2.1.1, will also be essential to fully validate and optimize these engines.

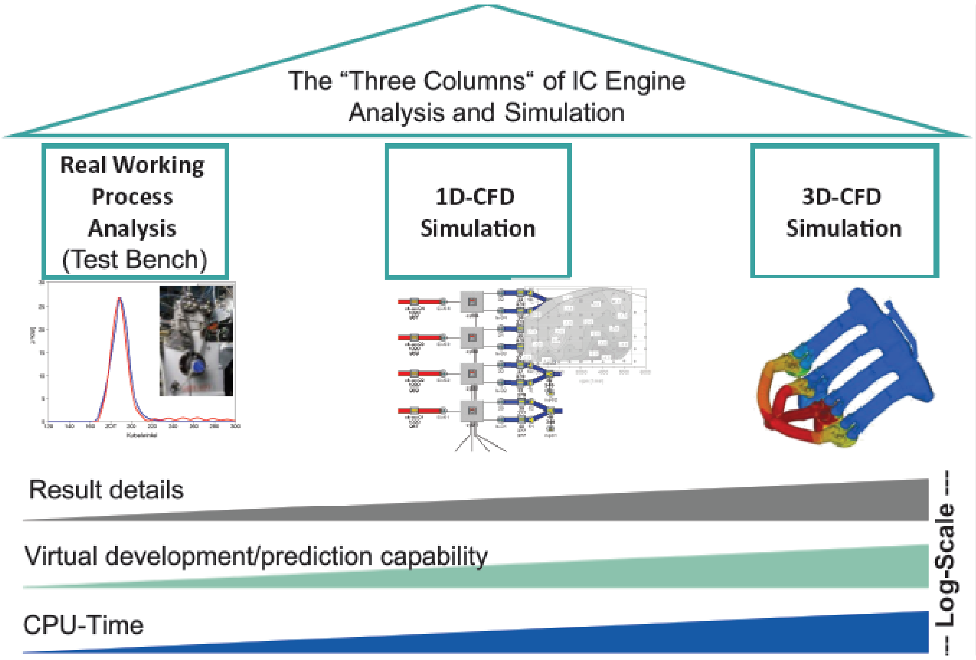


Figure 2.1.1 - *Development methodologies of an internal combustion engines [1]*

Models can be classified based on the predicted space details as follows:

1. Zero-dimensional (0D) models with concentrated parameters, which assume no fluid motion and uniformity of quantities in the considered volume. These models are typically used to describe overall system behavior, such as emissions or fuel economy, and are often used for optimization studies. In figure 2.1.2, an example of a 0D modeling approach is shown.

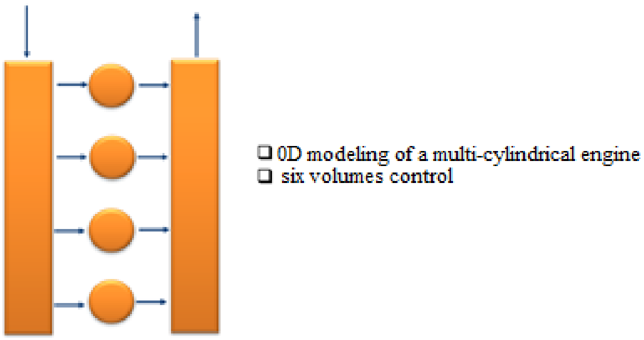


Figure 2.1.2 - *0D modelling of a four cylinders engine [1]*

2. One-dimensional (1D) models, which assume the existence of a main dimension over other two. 1D simulations are a crucial tool for successful engine development processes. In particular, their relatively low CPU-time and capability for comprehensive and reliable analysis have increased the popularity of 1D simulations in engine development. A model layout of 1D simulation is reported in figure 2.1.3.

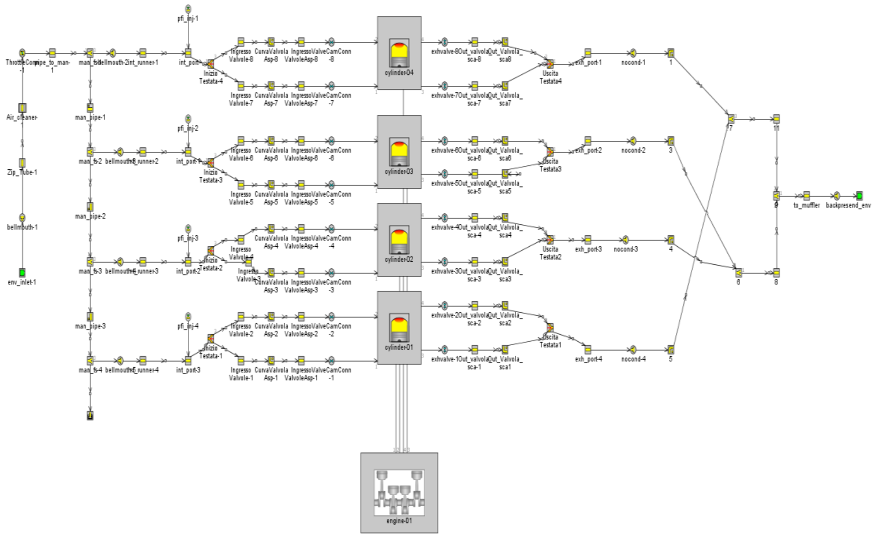


Figure 2.1.3 - *1D-CFD simulation: model layout of a four cylinders engine*

3. Three-dimensional (3D) models, which study the spatial and temporal evolution of the motion field. The most powerful tool for the solution of thermo-fluid-dynamical phenomena are 3D-CFD simulations, which provide accuracy and detail far greater than 1D simulations, but require far higher computational times. The current challenge is to improve the performance of 3D-CFD simulations in order to reduce computational times and maximize the number of evaluations and virtual testing of engines. In figure 2.1.4, an example of a 3D-CFD simulation's mesh can be found.

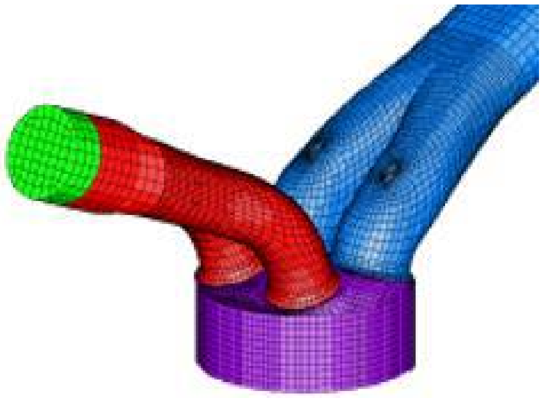


Figure 2.1.4 - *3D-CFD simulation: model layout of a cylinder [1]*

As mentioned before, the range of applications for 1D-CFD simulations includes:

1. Intake system layout optimization: 1D-CFD simulations can be used to optimize the geometry of the intake system, including the shape and dimensions of the intake manifold and the air filter. This can help to improve the air flow and distribution within the engine, resulting in increased performance and efficiency.

2. Exhaust system layout optimization: 1D-CFD simulations can also be used to optimize the geometry of the exhaust system, including the exhaust manifold, catalytic converter, and muffler. This can help to reduce backpressure and improve exhaust gas flow, which can in turn improve engine performance and reduce emissions.
3. Valve-timing optimization: 1D-CFD simulations can be used to optimize the timing and duration of the engine's intake and exhaust valves. This can help to improve the engine's volumetric efficiency, resulting in increased power and torque.
4. Turbocharging layout optimization including intercooler: 1D-CFD simulations can be used to optimize the turbocharging system, including the turbocharger and intercooler. This can help to improve boost pressure and reduce turbo lag, resulting in increased power and torque.
5. Analysis and optimization of the engine transient behavior: 1D-CFD simulations can be used to analyze and optimize the engine's transient behavior, including its response to changes in load and speed. This can help to improve drivability and reduce emissions, while also ensuring that the engine operates within safe and reliable limits.

The main drawback of the zero-dimensional and one-dimensional models is the need for a complex model calibration process, which is heavily reliant on accurate experimental data to ensure successful implementation for investigations based on parameter variations [1]. However, the 3D-CFD simulation is considered the most powerful tool for the solution of thermo-fluid-dynamical phenomena, providing a high degree of accuracy and detail, although requiring far higher computational times compared to 1D models.

To improve the performance of 3D-CFD simulations, reducing computational times is crucial in order to increase the number of engine evaluations and virtual tests. Figure 2.1.1 shows that the level of detail and the ability to predict virtual engine development results can be compromised by the high CPU time and complexity of the simulations, which can be expensive and time-consuming.

However, as computing power continues to grow and simulation software evolves, numerical simulations are becoming more integrated into the engine development process. By using 3D-CFD simulations, it is possible to analyze the internal processes of the engine with great accuracy, providing information that may be difficult or even impossible to obtain via measurement, such as turbulence and mixture formation.

This approach can investigate flow-dependent processes with high temporal and spatial resolution, providing a deeper understanding of how variations in geometry, valve timing, and other factors affect combustion processes. Additionally, simulations can be used to analyze the effects of such variations without the need for expensive prototyping.

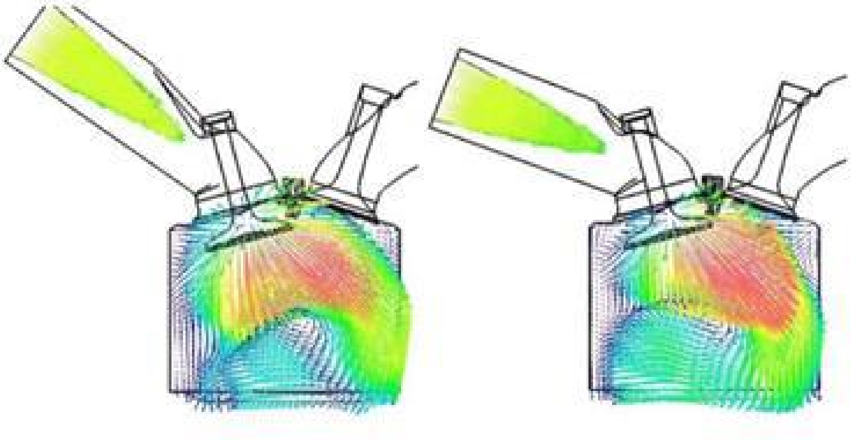


Figure 2.1.5 - *Virtual analysis of the effects on tumble motion by variations of intake runner angle [1]*

The description above highlights the dual role of 3D-CFD simulation as a measurement tool and simulation device, while 1D simulations are useful in concept studies and transient calculations of engine processes, especially when considering other drivetrain components. However, there comes a point where 3D-CFD becomes the more powerful tool, such as when trying to optimize the geometry of inlet pipes and valves for a gasoline engine to achieve high tumble generation and low coefficients of discharge. This is an example of how the whole process can have iterations and how 1D and 3D approaches can complement each other.

In this context, 3D-CFD results can be used for point-dependent validation of zero-dimensional (0D) to one-dimensional (1D) models regarding turbulence, residual gas content, and other factors. Nevertheless, experimental investigations remain essential for validation and will not be entirely replaced by simulations.

The combination of targeted and high-quality experimental investigations on the test bench with comprehensive numerical simulations provides a general overview of engine behavior.

2.2 Virtual Development of Internal Combustion Engines through 3D-CFD Simulation

Computational fluid dynamics (CFD) is a computer-based simulation technique used to analyze systems involving fluid flow, heat transfer, and related phenomena such as chemical reactions. CFD is a powerful tool with a wide range of industrial and non-industrial applications. It has been extensively used in research and development of internal combustion engines (ICE) due to their complex behavior, characterized by unsteadiness, presence of reactive flows, combustion process, multi-phase flows, and heat exchange. Since the 1990s, the availability of affordable high-performance computing hardware and user-friendly interfaces has led to a growing interest in CFD, and it has entered into the wider industrial community [1]. CFD codes are structured around numerical algorithms that solve fluid flow problems.

The CFD process involves three main steps:

- pre-processing, where input is set;
- 3D-CFD calculation, where conservation equations are solved;
- and post-processing, where simulation output data is analyzed.

During the pre-processing phase of a CFD simulation, the user sets up the simulation using a user-friendly interface provided by the software. The first step is the generation of the engine CAD model, which serves as the basis for the entire simulation. The CAD model is used to create a computational domain that represents the physical space in which fluid flow and associated phenomena are to be studied.

The next step is mesh generation, which involves dividing the computational domain into a large number of cells in which the Navier-Stokes equations will be solved. The meshing process is crucial, as it affects the accuracy and stability of the numerical solution. The mesh should be refined in areas where high gradients in flow properties are expected, such as near walls or in regions of high turbulence.

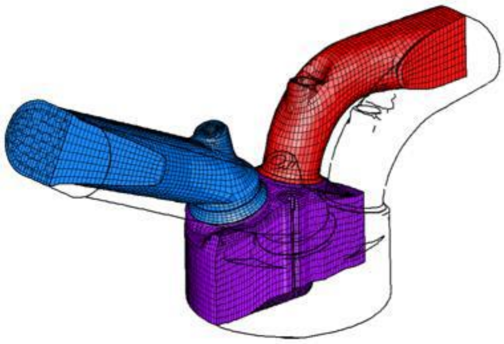


Figure 2.2.1 - *Computational mesh of an internal combustion engine [1]*

After meshing, the user selects the physical and chemical phenomena that have to be modelled. These phenomena can include laminar or turbulent flow, heat transfer, combustion, and chemical reactions. The choice of models depends on the specific application and the level of accuracy required.

Finally, the user sets the boundary conditions for the simulation, which describe the flow properties at the boundaries of the computational domain. Boundary conditions include values for velocity, pressure, temperature, and species concentrations. They are essential for obtaining meaningful results from the simulation.

In summary, the pre-processing phase of a CFD simulation involves setting up the simulation by creating a CAD model, meshing the computational domain, selecting the physical and chemical phenomena to be modelled, and setting boundary conditions. Proper setup of the simulation is essential for obtaining accurate and meaningful results from the simulation.

The solution to a flow problem, such as velocity, pressure, and temperature, is defined at nodes located inside each cell, forming a grid of points as shown in figure 2.2.2.

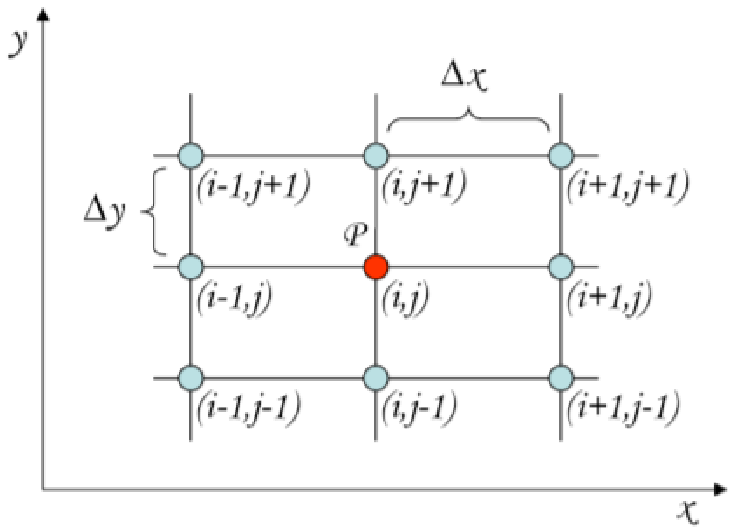


Figure 2.2.2 - *Spatial discretization of the computational domain [1]*

The solution at each node is obtained by solving the Navier-Stokes equations, which describe the motion of fluids. The equations are solved numerically using iterative methods such as finite volume or finite element methods. The accuracy of the solution depends on the quality of the mesh, i.e., the number of cells and their distribution in the computational domain. A well-designed mesh should have a sufficient number of cells in regions of high flow gradients and geometric complexity, while having fewer cells in regions of low gradients to reduce computational cost. Once the numerical solver has computed the solution, the post-processing stage involves analysing and interpreting the simulation results to gain insights into the flow behavior and make informed design decisions.

The most common types of numerical solution techniques are finite difference, finite element, and finite volume methods. The three fundamental equations of conservation of mass, momentum, and energy can be derived from the Navier-Stokes equations. These equations are the foundation of the finite volume method used in 3D-CFD of ICE.

The three equations are described by the following expressions:

1. Mass conservation:

$$\frac{\partial p}{\partial t} + div(\rho u) = 0$$

Equation 2.2.1 - *Mass conservation equation*

It describes the conservation of mass within a control volume and is expressed as a continuity equation, where:

- ρ is the density;
- u is the tensor of the velocities in the three dimensions.

2. Momentum conservation:

$$\rho \frac{Du}{Dt} = -\frac{\partial p}{\partial x} + div[\mu \nabla(u)] + S_x$$

$$\rho \frac{Dv}{Dt} = -\frac{\partial p}{\partial y} + div[\mu \nabla(v)] + S_y$$

$$\rho \frac{Dw}{Dt} = -\frac{\partial p}{\partial z} + div[\mu \nabla(w)] + S_z$$

Equation 2.2.2 - *Equations of momentum conservation in space domain*

It describes the conservation of momentum and the forces acting on a fluid within a control volume, where:

- S_x, S_y, S_z are the source terms;
- μ is the fluid diffusivity.

3. Energy conservation:

$$p\frac{Di}{Dt} = -pdiv(\rho u) + div[k\nabla(T)] + \Phi + S_i$$

Equation 2.2.3 - *Energy conservation equation*

It describes the conservation of energy within a control volume and includes the effects of heat transfer and work done by external forces, where:

- S_i are the source terms;
- i is the interna energy of the fluid;
- Φ is the dissipation function and represents the source of the internal energy due to the deformation work on the fluid particle;
- k is the thermal conductivity of the fluid.

In addition to the definition of geometrical parameters of turbulence structures, it's worth mentioning that the evaluation of turbulent characteristics parameters is based on statistical averages. This means that the variations of turbulence quantities are averaged over time and/or space.

To close the system of equations, an equation of state is added which expresses the pressure as a function of density and temperature. These equations typically have a property variation on the left side due to convection and variations on the right side due to diffusion and sources. However, the system is complex as some variables such as pressure cannot be separated from the fluid flow solution. To address this, different approaches have been developed, with the main challenge being the mathematical definition of terms that arise from the stresses on the control volume through momentum exchange. These terms are responsible for producing turbulence, which is a critical factor in the analysis of internal combustion engines using 3D-CFD simulation. There are different models available for describing turbulence:

1. Direct Numerical Simulation (DNS);
2. Large Eddy Simulation (LES);
3. Reynolds Averaged Navier-Stokes (RANS) equations.

The in-cylinder air flow structures are essential for the preparation of the mixture and the development of the flame in a spark-ignition engine. Different approaches for simulating turbulence are based on different methodologies for evaluating the characteristic parameters of turbulence. However, before discussing these approaches, it is important to define the geometric parameters of the turbulence structures:

1. Integral length scale (LI), which is the dimension of large-scale turbulence structures and a measure of the biggest turbulence structures. It is closely related to the geometrical computational domain;
2. Kolmogorov scale (Lk), which is the dimension of micro-scale turbulence structures and a measure of the smallest turbulence structures of the motion field. In IC-engines, it is around 0.015 mm on average over the operating cycle. The evaluation of the Kolmogorov length scale is strictly related to the scale of molecular diffusion at which the turbulent kinetic energy (k) dissipates directly into internal energy due to viscosity.

Refers to the models available for describing turbulence mentioned before, DNS provides the most accurate solution of the equations without introducing any turbulence models, but it requires very detailed grids and small time-steps, making it a very slow method. To evaluate every turbulence structure, the computational mesh must have cells with a dimension much smaller than the Kolmogorov length scale, making it impractical due to the required computational resources.

On the other hand, LES is a numerical technique used to solve the differential equations that govern turbulent fluid flow. It evaluates the integral length scale LI while modeling the effects of the Kolmogorov scale. LES requires less computational effort than DNS but more than RANS equations.

Finally, RANS equations average the Navier-Stokes equations in both time and space. Solving the RANS equations involves nonlinear terms called Reynolds stresses, which cannot be solved directly. A turbulence model capable of describing these stresses, such as the k-epsilon model, must be associated with the equations of motion. This model adds two more transport equations to the RANS equations, one for the turbulent kinetic energy and one for the turbulent eddy dissipation rate, as well as an algebraic formula to determine the eddy viscosity.

RANS equations are commonly used in 3D-CFD due to their good trade-off between accuracy and computational cost.

In the figure 2.2.3 is shown a comparison between the three models.

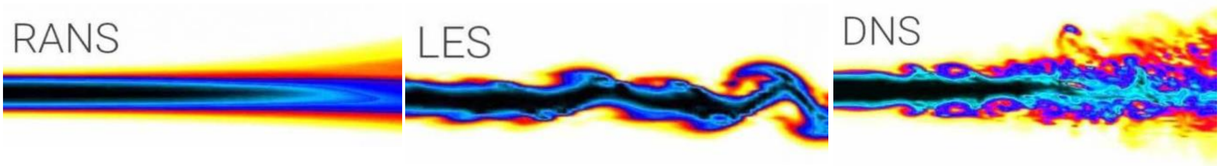


Figure 2.2.3 - *Models comparison [9]*

Hence, 3D-CFD simulations allow for a detailed investigation of physical phenomena. However, a limiting factor of simulations is the requirement for reliable boundary conditions. In a 3D-CFD simulation, it is important to define the model input and connect the model with the surrounding environment.

2.2.1 Boundary Conditions

Boundary conditions are necessary constraints for solving a boundary value problem, which define the input of a simulation model and link it with the surrounding environment. A boundary value problem is a differential equation or a system of differential equations that need to be solved within a domain with known boundary conditions. These problems are crucial as they model a vast range of phenomena and applications, including solid mechanics, heat transfer, fluid mechanics, and acoustic diffusion. They naturally arise in every problem based on a differential equation in space, whereas initial value problems usually refer to problems to be solved in time. Choosing realistic boundary conditions is one of the most crucial and challenging parts of setting up a simulation. It is not necessarily difficult to find a combination that works, but it is essential to produce meaningful results instead of random numbers. Boundary conditions can be constraints, which are preserved during simulation, or variables, which change during simulation and are usually used to simulate a cyclic phenomenon.

Boundaries in 3D-CFD simulations of internal combustion engines (ICE) can take on different forms, such as inlets, outlets, walls, symmetric planes, or axes for axial-symmetric geometries. The setting of boundary conditions for such simulations can be achieved through various methodologies:

1. *Coupled 1D-3D simulation*

   As described in paragraph 2.1, one-dimensional (1D) and three-dimensional (3D) approaches have different levels of computational costs and degree of detail. By coupling their features, it is possible to simplify and maximize the reliability of the simulation. For instance, 1D-CFD models can be applied to turbocharging layout and intake systems, allowing for the evaluation of the boundary conditions at the inlet of a 3D-CFD model. Naturally, the 1D model must be calibrated with the corresponding boundary conditions that can be experimentally detectable. This approach typically requires a significant amount of engine cycles to obtain meaningful solutions;

2. *Experimental approach*

   The evaluation of boundary conditions can also be carried out through test bench analyses where they are detected experimentally.

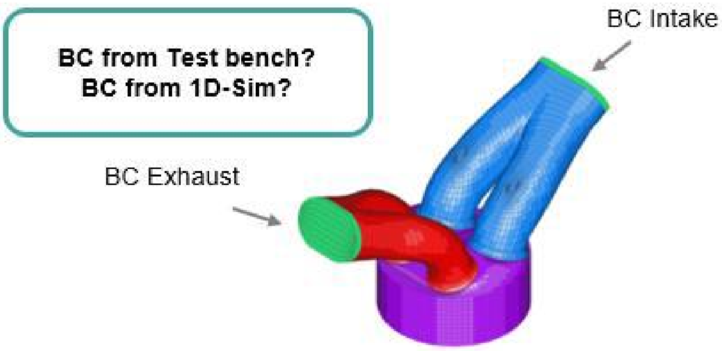


Figure 2.2.4 - *Reduced computational domain of an ICE*

3. *Extension of the computational domain*

   One possible solution is to extend the computational domain to simulate all the processes that happen before and after the cylinder, returning them to environmental conditions. This approach is often used in *QuickSim* environment because of the computational cost savings it provides. In Figure 2.2.5, an example of a computational domain for a single-cylinder engine is shown, including the air-tank, which is typical of test benches and used for reproducing air compression, intake manifolds, cylinder, and exhaust line. In this case, the boundary conditions are environment pressure at the exhaust and boost pressure at the inlet. A possible development of this solution is to couple a 0D-1D model for turbocharger to simulate air compression, resulting in inlet boundary conditions as environment pressure.

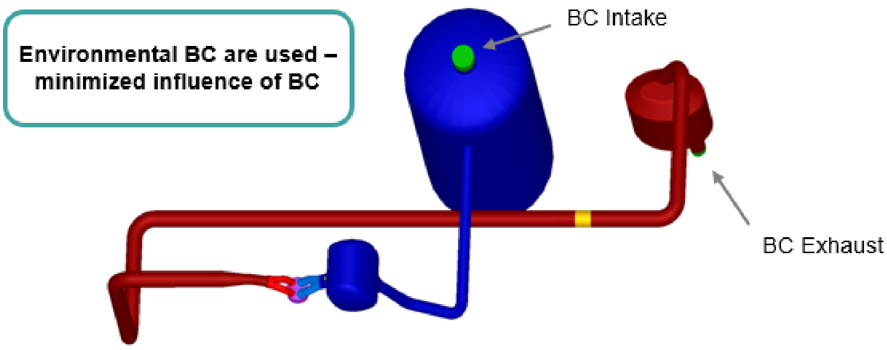


Figure 2.2.5 - *Extended computational mesh for a 3D-CFD simulation*

2.2.2 Convergence of the simulation

Three types of errors can arise from the numerical solution of Navier-Stokes equations:

1. *Discretization errors:*

   These arise because a finite number of discrete points or nodes are used to approximate a continuous function. The discretization error is defined as the difference between the result obtained by the differential operator and the result obtained by the numerical operator, both with infinite significant digits.

2. *Round-off errors:*

   These are due to inexactness in the representation of real numbers and the arithmetic operations done with them. Rounding errors occur because on a digital computer, real numbers are stored only approximately up to a certain number of decimal places, often referred to as the precision of the number. This is a form of quantization error and is a manifestation of the fact that computers can only represent real numbers to a finite precision.

3. *Truncation errors:*

   These are the difference between the true (analytical) derivative of a function and its derivative obtained by numerical approximation. The truncation error can be expressed as the difference between the numeric operator and the exact solution, or as the difference between the true derivative and its numerical approximation.

Truncation errors are unavoidable in any discretization scheme unless the higher derivatives are zero, which rarely occurs in problems of practical interest. Therefore, these errors must be kept as small as possible. One way to achieve this goal is to refine the mesh to a point where the truncation error becomes negligible, which means smaller than the round-off error. Such a solution is known as a grid-independent solution. In practice, this formal criterion cannot always be strictly met. Most often, a much more relaxed criterion is employed.

To assess the accuracy of a numerical simulation, it is important to monitor the convergence of the solution. Convergence is the process by which the solution of the numerical scheme approaches the exact solution as the spatial and temporal resolutions are refined. In order to achieve a converged solution, the truncation error must be smaller than the round-off error. One way to estimate the truncation error is to perform the simulation with different spatial and temporal resolutions and observe how the solution changes. If the solution changes significantly with changes in the resolution, the simulation is not converged, and the truncation error is likely to be larger than the round-off error.

2.3 QuickSim Tool

The processes that occur within the working fluid of an internal combustion engine are numerous and complex. In recent years, there has been an increasing demand for a predictive, fast, and reliable 3D-CFD tool that can be efficiently integrated into the internal combustion engine development process. This demand led to the introduction of an innovative concept in the simulation of internal combustion engines, which is the fundamental idea behind the 3D-CFD-program QuickSim.

QuickSim is a fast-response software developed by Dr-Ing. *Marco Chiodi* at FKFS (Forschungsinstitut für Kraftfahrwesen und Fahrzeugmotoren Stuttgart) and IVK (Institut für Verbrennungsmotoren und Kraftfahrwesen) of the University of Stuttgart. The tool uses the traditional CFD-code StarCD® from Adapco in the background and is able to solve CFD problems applied to ICE in a significantly shorter time than traditional software, as shown in figure 2.3.1.

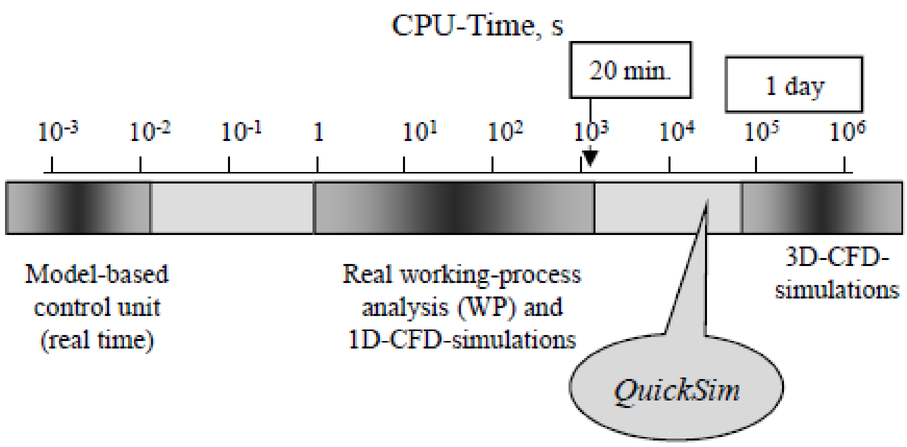


Figure 2.3.1 - *The 3D-CFD tool QuickSim in the spectrum of CPU-time among different calculation/simulation tools [1]*

QuickSim developed by Dr-Ing. *Marco Chiodi,* offers several unique features, including fast and reliable analysis calculations, clear representation of results, full-engine 3D-CFD simulation, reproduction of real test bench conditions, cost efficiency, and simulation flexibility.

This simulation tool is seamlessly integrated into the existing engine development process, enabling efficient comparisons with experimental data and other simulation programs.

QuickSim also allows for simulations of the entire engine domain, starting from the design phase and even before prototyping. In addition, the tool enables a reduction in the number of cells used for solving equations, leading to significant reductions in CPU-time, without compromising on accuracy.

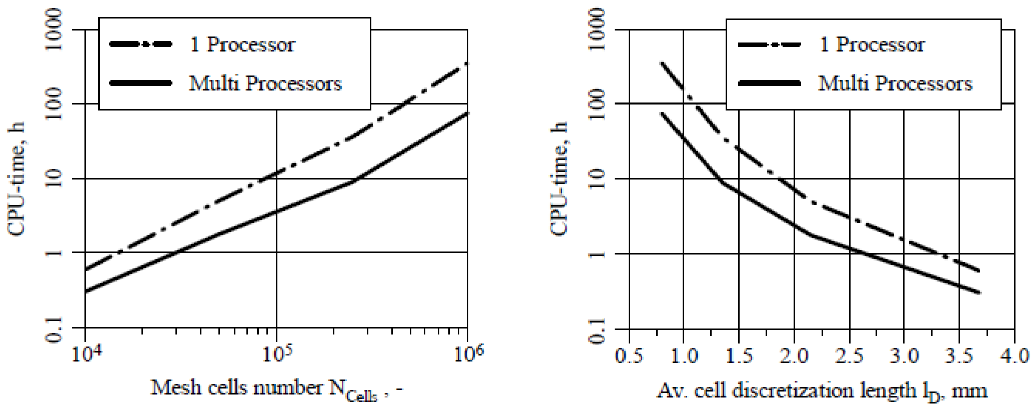


Figure 2.3.2 - ***Left***: *CPU-time as function of the total number* ***Right:*** *CPU-time as function averaged cell-discretization length [1]*

Figure 2.3.2 provides an estimate of the CPU-time required for a simulation over a full operating cycle, as a function of the total number of cells in the mesh and its averaged cell-discretization-length. It is evident that reducing the cell-discretization-length leads to an increase in the number of cells, and consequently, the CPU-time required for the simulation.

For example, halving the cell-discretization-length from 1.5 mm to 0.75 mm results in an increase of CPU-time by approximately a factor of 25.

Thanks to its flexibility, it is possible to investigate the following with QuickSim:

1. Engine layout: QuickSim can simulate any cylinder number, combustion chamber geometry, intake and exhaust system geometry, and injection system with arbitrary injector geometry;
2. Ignition type: QuickSim can simulate spark-ignition, compression ignition, homogeneous charge compression ignition (HCCI), and spark-assisted compression ignition (SACI);
3. Fuel type: QuickSim can simulate gasoline, diesel, compressed natural gas (CNG), biofuels, and e-fuels;
4. Valve and piston motion: QuickSim allows for simulation of valve and piston motion.

Furthermore, QuickSim minimizes the influence of boundary conditions (BC) by extending the simulation domain to a full 1-cylinder or full-engine test bench, eliminating the need for external simulations to provide boundary conditions.

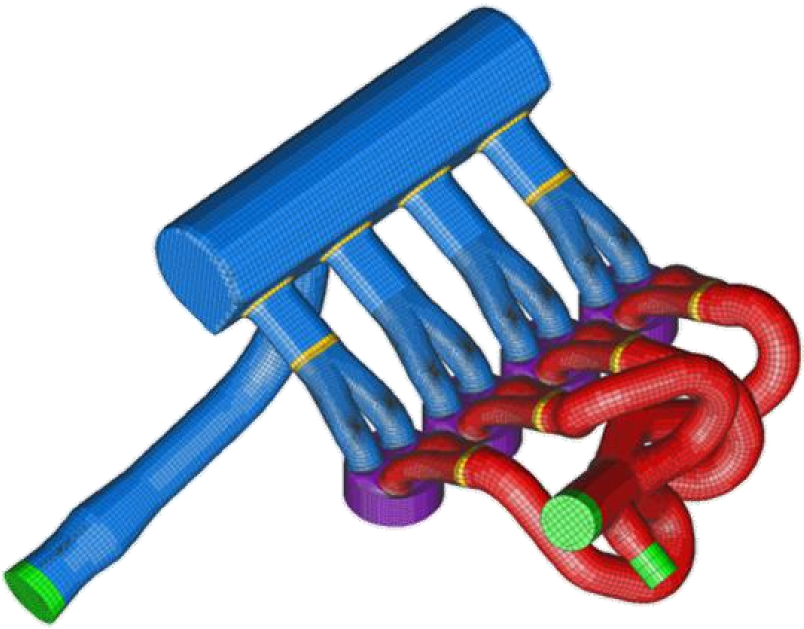


Figure 2.3.3 - *Typical QuickSim computational domain*

Figure 2.3.3 shows a typical *QuickSim* computational domain, where the 3D domain extends to the full engine, including the airbox, all the cylinders, and the exhaust system. This enables a detailed analysis of the differences among the cylinders, with a focus on volumetric efficiency, residual gas concentration at IVC, fuel mixture formation, turbulence profile up to the end of combustion, combustion progression, and more. By performing a full-engine simulation with *QuickSim,* a high level of predictability can be achieved, allowing investigation into the influence of design modifications, different injection strategies, and valve timings on the performance of the "entire" engine, making it a critical step towards virtual engine development.

As previously mentioned, the 3D-CFD models used in QuickSim are not universally applicable for all thermodynamic investigations. Instead, their formulation is costumized to optimize the solution of engine processes and minimize the computational resources required for the calculation. These models rely on a combination of different approaches, including traditional local 3D-CFD models, engine-specific phenomenological relationships, trained neural networks, databases, and empirical relationships. They also explicitly consider the cell dimensions and structures, enabling a reliable analysis of the process and its relevant behavior for practical applications in the engine design process.

The primary goal of performing rapid calculations for the thermodynamic properties of the working fluid in QuickSim is to closely approximate the characteristics of real gas, ensuring a thermodynamically accurate representation. However, it should be noted that the calculated fluid composition, which may include pollutant species such as NO_x and HC, cannot be properly used for determining exhaust emissions. Dedicated models for emission calculation must be implemented for this purpose.

Chapter 3

3 Development of the Algorithm

3.1 Introduction

3D-CFD simulations are characterized by a very long computational times, which is why the purpose of this work is to develop a code that can optimize the engine development time by automating the simulation setup, which is normally done manually. The *QuickReg* package serves as a real-time regulator of 3D-CFD simulations conducted by the 3D-CFD tool *QuickSim*, developed by Dr-Ing. *Marco Chiodi*.

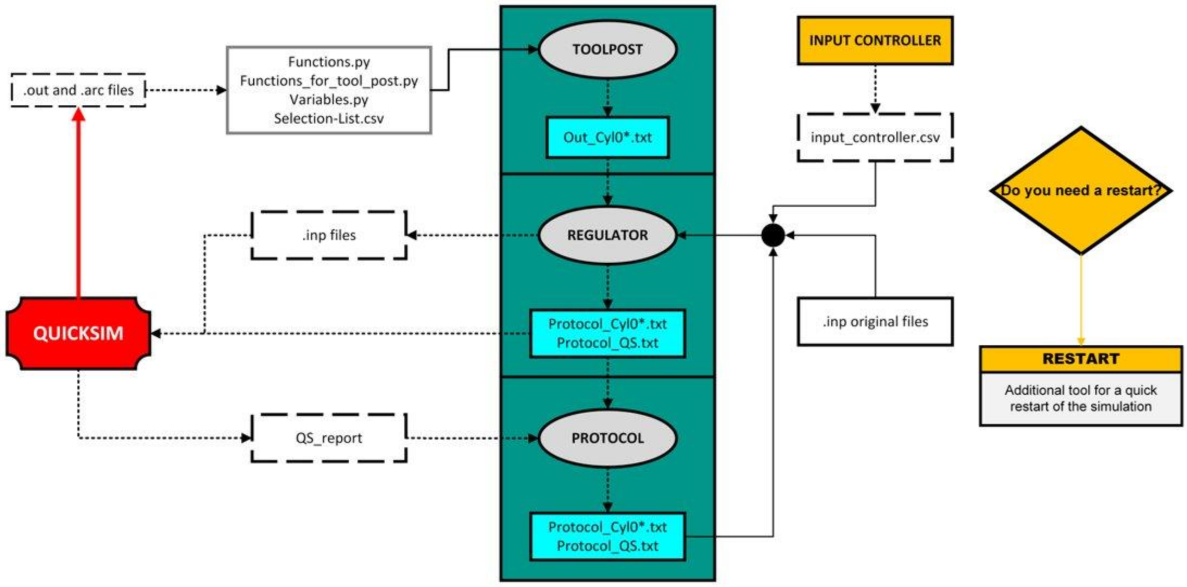


Figure 3.1.1 - *QuickReg flowchart*

The *QuickReg* package consists of several tools that work together to optimize engine development time through the automation of 3D-CFD simulations in the 3D-CFD tool *QuickSim*. The package includes three main algorithms, namely *ToolPost*, *Regulator*, and *Protocol*, which are displayed in Figure 3.1.1. In addition, an user interface developed in *Excel* generates the input controller file required for the regulator tool. More details about the individual tools will be provided in the following paragraphs.

The pre-processing step is crucial to set up the simulation correctly. In this step, input files (*QSI*) are generated for *QuickSim*, which is used to solve the governing equations of CFD. The simulation output a large amount of data, which are stored in *QuickSim Output* files (*QSO*). These files are then post-processed by the user to analyze the simulation results.

The simulation setup requires the imposition of boundary conditions and this step is done in the pre-processing phase, as described in subparagraph 2.2.3. These conditions represent the thermo-fluid dynamic conditions at the interface of the mesh. Selecting the appropriate boundary conditions is a non-trivial process that requires careful consideration, as errors during this stage could compromise the reliability of the results.

Boundary conditions, evaluated experimentally or through 1D-CFD codes, are specified in QuickSim Input (*QSI*) files. *QSI* file needed for the simulation are:

1. *Qs-exh-caloric.inp* = In this file are present informations about the fuel properties;
2. *Qs-fuel-flamind.inp* = Information about the flame front velocity of the fuel;
3. *Qs-injection-fu.inp* = It referes to injection characteristics;
4. *Qs-spark-ign-sp.inp* = It presents information about the spark characteristics;
5. *Qs-liq-fuel-properties.inp* = In this file are present informations on the trend of the fuel to vaporization;
6. *Qs-manifolds-em.inp* = information about exhaust manifolds;
7. *Qs-manifolds-im.inp* = It refers to intake manifolds characteristics;
8. *Qs-setting.inp* = overall engine, valves ecc. informations.

Sometimes, the simulation setup intended to replicate a specific engine operating point is not executed correctly, leading to the need for manual interventions by the user. As a result, valuable time is wasted, and the simulation produces unreliable results that must be repeated. This poses a challenge to meeting the company's time-to-market requirements, which can be particularly demanding. To address these challenges, there is a growing need to develop a code that can automate the regulation of the simulation.

As mentioned before, the *Regulator* tool serves as a real-time controller for the most important parameters of the engine. Its steps are as follows:

1. *Checking:* the tool verifies if the analyzed engine parameters are within the predefined target values set by the user;
2. *Regulation:* if the checking has a poor response, the tool proceeds with the regulation phase and modifies boundary conditions to ensure the target values are met. This creates a new boundary conditions file, which is fed to the simulation in real-time.

The development of the algorithm was made possible thanks to the support of Dr-Ing. Marco Chiodi and his team. The algorithm was written in *Python*, a high-level, general-purpose programming language created by Guido van Rossum in 1991 [12]. Its language constructs and object-oriented approach help programmers to write clear and logical code for small and large-scale projects.

The libraries used in the development of the *QuickReg* package are:

1. *Pandas*, one of the most popular data science tools used in *Python* for data collection and analysis;
2. *NumPy*, the fundamental package for scientific computing with *Python*, which provides an abundance of useful features for operations on n-arrays and matrices in *Python*.

QuickReg is capable of controlling three different engine parameters: Lambda, Center of the Combustion, and Indicated Mean Effective Pressure. More details about each of these regulations will be provided in the following sections.

3.2 Pre-Processing of QuickReg

As mentioned earlier, the output files obtained from *QuickSim* are called *QSO* (*QuickSim Output*) files. The *QSO* files are databases in which all the simulation outputs are stored and they are discretized crank-angle to crank-angle based on the user's choice. Additionally, the *QSO* files are divided into two types of files: *qso.arc*, which stores all the variables for each cycle, and *qso.out*, which stores only the variables of the current and completed cycle. It's worth noticing that the *.out* file has a smaller memory size because it contains less data.

The code interacts with the 3D-CFD software in real-time, and as a result, *.out* files are used to collect and analyze output data from the simulation. As stated earlier, these files are lighter and easier to manage, which helps to keep the computational time of the algorithm low.

The inputs provided by QuickSim to the controller during the simulation are:

1. Cylinder number;
2. Actual simulated crank angle;
3. Cycle number;
4. Crank angle index: QuickSim uses numerical digits to represent the crank angles of greater motor interest. The index is necessary to evaluate which parameter needs to be checked. Table 3.2.1 shows the selected points.

0	Start of the cycle + 1 CA	**13**	1% burned mass
1	IVO	**14**	5% burned mass
2	EVC	**15**	10% burned mass
3	IVC	**16**	50% burned mass
4	IP/SOI	**17**	90% burned mass
5	FTDC	**18**	95% burned mass
6	1% burned fuel	**19**	99% burned mass
7	5% burned fuel	**20**	Combustion end
8	10% burned fuel	**21**	EVO
9	50% burned fuel	**22**	End of cycle
10	90% burned fuel	**23**	40bFTDC
11	95% burned fuel	**24**	450
12	99% burned fuel		

Table 3.2.1 - *Selected points in QuickSim*

To use the QuickReg package, it is necessary to activate the ECU external regulator by changing its value from zero to one in the *qs-setting.inp* file. This setting is indicated in row shown in Figure 3.2.1.

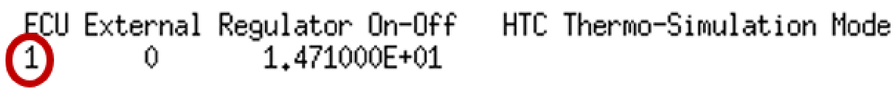


Figure 3.2.1 - *qs-setting.inp row to active the QuickReg package*

Besides, to ensure everything is in order, the user should verify that all necessary tools are present in the main simulation folder. The main simulation folder, as shown in the figure 3.2.2, should include two important sub-folders, namely *02_Regulator* and *04_Tool-Post*. The former contains all the files required for the *REGULATOR.py* to function properly. On the other hand, the latter comprises files that are essential for computing the main engine parameters cycle-by-cycle. Moreover, the user should ensure that the names are written correctly.

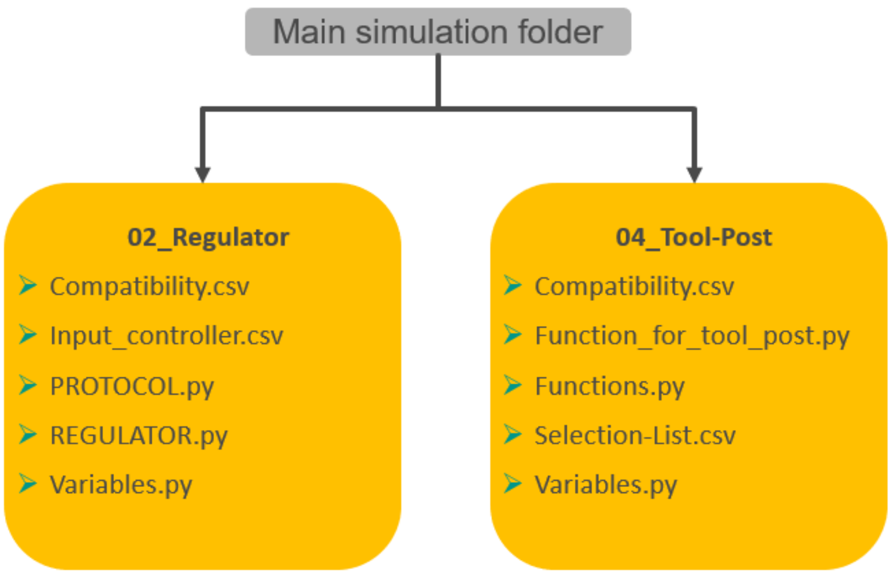


Figure 3.2.2 - *Main simulation folder*

After setting up all the folders and the ECU correctly, the *input_controller.csv* file must be generated. In the next section will explain how the input controller works and how the file *.csv* is generated.

3.3 Input Controller

The input controller is a user-friendly interface developed in the *Excel*, and it can generate the *input_controller.csv* file. This file is one of the most critical files for the *QuickReg* package because it contains all the information necessary for the simulation to function correctly. Furthermore, the *Regulator* tool can identify the engine parameters that it must regulate.

The input controller presents the interface show in the figure 3.2.1.

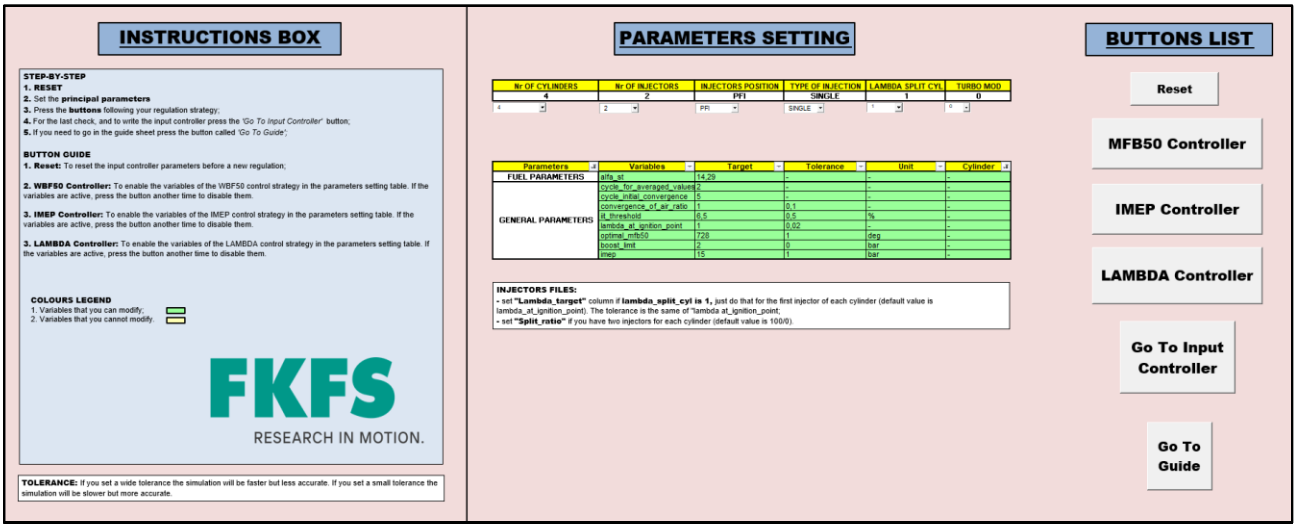
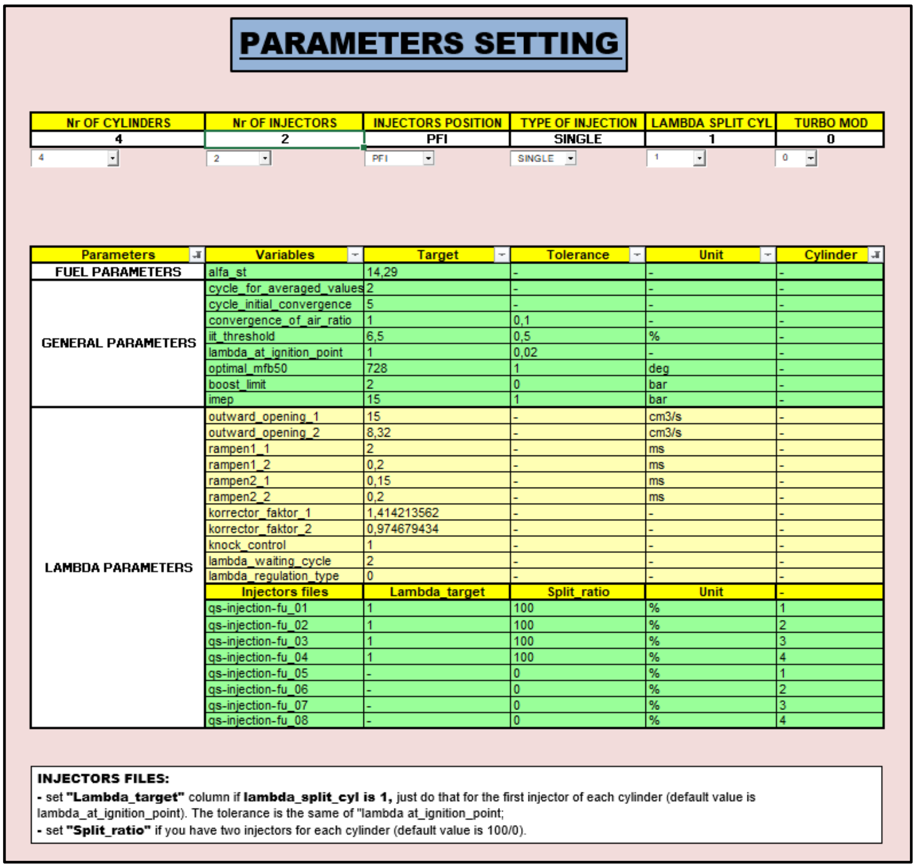


Figure 3.3.1 - *Input controller interface*

As can be seen, on the left side of the interface, there is an instruction box that explains how to use the input controller, while on the right side there are two sections: parameter settings, in which the user can configure their simulation parameters, and the button list, where the user can select the regulation that *QuickReg* should perform. All the informations are present in the *QuickReg – User guide documentation* [19] developed during the internship.

An example is provided below to explain in detail how the interface works. If the user decides to enable the regulation of lambda, they must press the *lambda* button. Once the button is pressed, a new section called *Lambda parameters* appears below the *General parameters* table. In this section, the user can set the characteristics of the injectors and the target value of lambda, along with the acceptable range of values.



PARAMETERS SETTING

Nr OF CYLINDERS	Nr OF INJECTORS	INJECTORS POSITION	TYPE OF INJECTION	LAMBDA SPLIT CYL	TURBO MOD
4	2	PFI	SINGLE	1	0
4	2	PFI	SINGLE	1	0

Parameters	Variables	Target	Tolerance	Unit	Cylinder
FUEL PARAMETERS	alfa_st	14,29	-	-	-
GENERAL PARAMETERS	cycle_for_averaged_values	2	-	-	-
	cycle_initial_convergence	5	-	-	-
	convergence_of_air_ratio	1	0,1	-	-
	iit_threshold	6,5	0,5	%	-
	lambda_at_ignition_point	1	0,02	-	-
	optimal_mfb50	728	1	deg	-
	boost_limit	2	0	bar	-
	imep	15	1	bar	-
LAMBDA PARAMETERS	outward_opening_1	15	-	cm3/s	-
	outward_opening_2	8,32	-	cm3/s	-
	rampen1_1	2	-	ms	-
	rampen1_2	0,2	-	ms	-
	rampen2_1	0,15	-	ms	-
	rampen2_2	0,2	-	ms	-
	korrector_faktor_1	1,414213562	-	-	-
	korrector_faktor_2	0,974679434	-	-	-
	knock_control	1	-	-	-
	lambda_waiting_cycle	2	-	-	-
	lambda_regulation_type	0	-	-	-
	Injectors files	Lambda_target	Split_ratio	Unit	-
	qs-injection-fu_01	1	100	%	1
	qs-injection-fu_02	1	100	%	2
	qs-injection-fu_03	1	100	%	3
	qs-injection-fu_04	1	100	%	4
	qs-injection-fu_05	-	0	%	1
	qs-injection-fu_06	-	0	%	2
	qs-injection-fu_07	-	0	%	3
	qs-injection-fu_08	-	0	%	4

INJECTORS FILES:
- set **"Lambda_target"** column if **lambda_split_cyl is 1,** just do that for the first injector of each cylinder (default value is lambda_at_ignition_point). The tolerance is the same of "lambda at_ignition_point;
- set **"Split_ratio"** if you have two injectors for each cylinder (default value is 100/0).

Figure 3.3.2 - *Lambda controller interface*

For the specific case of lambda regulation, the user must also choose other characteristics related to the type of regulation, such as the knock controller, and the number of cycles to wait before a new regulation is performed.

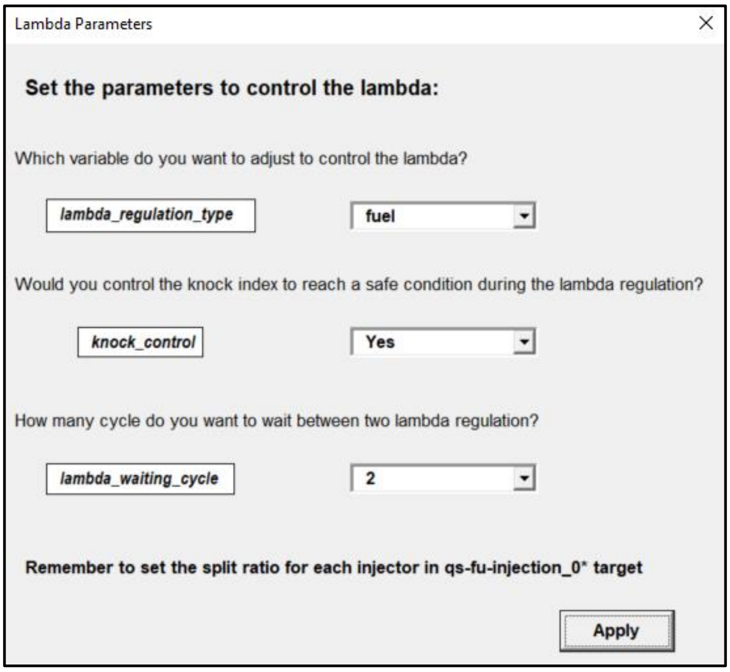


Figure 3.3.3 - *Other parameters present for lambda regulation*

The user must also set how many cylinders the engine has, which is the position of the injector, and the type of injection. Figure 3.3.4 shows explain how to do that.

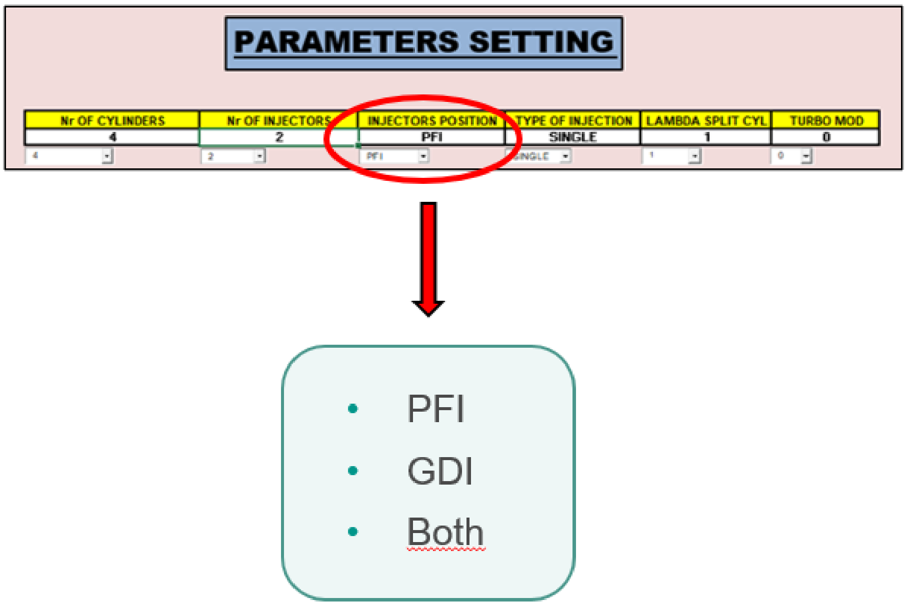


Figure 3.3.4 - *Main engine details*

Once the users have set all the necessary parameters, they can use the input controller tool to write the input_controller.csv file. This file contains all the variables set by the user, divided into four columns:

1. *Variables:* which lists all the variable names.
2. *Target:* which contains the target values that the user wants to achieve.
3. *Range:* which specifies the acceptable range for each variable.
4. *Unit:* which lists the unit of measurement for each variable.

An example of the *input_controller.csv* file is shown in the figure 3.2.5.

Variables	Target	Range	unit
lambda_split	0	NaN	NaN
alfa_st	17.206	NaN	NaN
turb_mod	0	NaN	NaN
cycle_convergent	2	0	NaN
cycle_initial_convergence	3	0	NaN
ratio_initial	1	0.2	NaN
iit_threshold	5	0.5	%
l_at_IP	1	0.02	NaN
optimal_wbf50	735	2	deg
imep	24	1	bar
cycle_BP_change	2	0	NaN
boost_limit	2.5	0	bar
outward_opening_1	800	0	cm3/s
rampen1_1	0.15	0	ms
rampen2_1	0.15	0	ms
korrector_faktor_1	1.414213562	0	NaN
cycle_L_AT_IP_change	1	0	NaN
qs-injection-fu_01	100	1	%
type_inj	0	NaN	NaN

Figure 3.3.5 - *Input_controller.csv*

Where:

- *lambda_split* indicates if the lambda will be equal in all-cylinders or different;
- *alfa_st* is the alpha stoichiometric;
- *turbo_mod* is the variable that indicates if the regulation of the turbo is present;
- *cycle_convergent* indicates after how many cycles an adjustment is made;
- *cycle_initial_convergence* indicates how many cycles the regulator wait before to start a regulation;
- *ratio_initial* indicates the ratio between the value of air consumption between the current and the previous cycle;
- *iit_threshold* indicates the knock limit;
- *l_at_IP* is the desiderable lambda at ignition point;
- *optimal_wbf50* is the value of MFB50 that the user would reach;
- *imep* is the indicate mean effective pressure;
- *cycle_BP_change* indicates after how many cycles an adjustment of BP is made;
- *boost_limit* is the threshold impose by the user in order to avoid damage;
- *outward, rampen and correct factor* are parameters that involves the injector characteristics;
- *cycle_L_at_IP_change* indicates after how many cycles an adjustment of lambda is made;
- *qs-injection-fu_0** is the file that will be modify by the regulator during a regution of lambda;
- *type_inj* indicates if the injection is single (0) or multiple (1);

Finally, once the input_controller.csv file is written, the simulation can start.

In the following section, it will be explained how the other tools work and how they interact with each other.

3.4 Tool-Post

As discussed earlier, the *QuickReg* package is composed by three main algorithms, that interact with each other following the logical flow shown in the figure 3.3.1. In this section will be described the *Tool-Post* and how it works.

Figure 3.4.1 - *Logical flow of the algortims*

The *Tool-Post* is an algorithm developed in python that collects all the parameters evaluated during the simulation and writes them into a text file, allowing the user to easily keep an overview of the engine parameters. The file provided by the *Tool-Post* is called *out_cyl0** and they are written as many as the cylinder numbers. An example is shown un the figure 3.4.2.

====== Results ======

VARIABLE	UNIT					
cycle	No	10	9	8	7	6
Parameter	-	Target_Knock_Index_reached	Waiting_after_IP_change	IP_change	Waiting_after_IP_change	IP_change
Engine_air_cons	kg/h	56,8427	-1	-1	-1	-1
cyl_air_cons	kg/h	56,8427	56,8198	56,7843	56,7299	56,6821
cyl_m_air_IP	mg	946,5527	946,2028	945,6394	944,8069	943,988
Engine_fuel_cons	kg/h	3,3168	-1	-1	-1	-1
cyl_fuel_cons	kg/h	3,3168	3,3167	3,3168	3,3167	3,3167
inj_m_fuel	mg/Cyc	55,2792	55,2777	55,2793	55,2788	55,2784
cyl_m_fuel_IP	mg	55,1123	55,0448	55,0919	55,1036	55,07
Engine_power	kW	19,3825	-1	-1	-1	-1
cyl_power	kW	19,3825	19,3689	19,1916	19,2163	18,9524
ind_eff_abs	%	47,8131	47,7809	47,3421	47,4035	46,7528
imep	bar	23,3176	23,3013	23,088	23,1177	22,8002
cyl_p_max	bar	138,7498	138,9958	129,6188	129,3382	120,5411
cyc_iit09m_max	%	5,33	5,65	1,09	1,1	0,05
cyl_egr_IVC	%	0,026	0,0258	0,026	0,0262	0,0261
cyl_max_wbf	%	0,99	0,99	0,99	0,99	0,99
cyl_10_wbf	deg	723,808	723,7767	725,4741	725,5118	727,1576
cyl_50_wbf	deg	734,8949	734,8164	736,644	736,7155	738,4383
cyl_90_wbf	deg	747,8628	747,8016	749,5289	749,4932	751,3237
cyl_10_90_wbf	deg	24,0548	24,0249	24,0548	23,9814	24,1661
cyl_lambda_IP	-	1,2019	1,2029	1,2012	1,1999	1,1996
prec_lambda_IP	-	1,0252	1,0192	1,0276	1,0322	1,0317
sp_lambda_IP	-	1,028	1,0216	1,0305	1,0345	1,0352
prec_egr_IP	%	4,3228	4,2835	4,2269	4,2607	4,1845
sp_egr_IP	kg/kg	4,1542	4,1015	4,0249	4,0617	3,9456
ip	deg	712,0	712,0	713,5	713,5	715,0
sp_r_max	mm	1,6	1,6	1,6	1,6	1,6
sp_dca_max	deg	1,5	1,5	1,5	1,5	1,5
cyc_A_Weller	-	3,7	3,7	3,7	3,7	3,7
imb_p_01	bar	2,1868	2,1883	2,1883	2,1882	2,1883
imb_T_01	K	313,1512	313,3807	313,3999	313,4238	313,4494
emb_p_01	bar	2,0069	2,0083	2,0083	2,0083	2,0083

Figure 3.4.2 - *Out_cyl0* format*

The *Out_Cyl0*.txt* file is written by the *Tool-Post* at the end of each cycle and plays a crucial role in the regulation process. Firstly, it interacts with the regulator to check whether the variables selected by the user for regulation are within range or not. Secondly, it allows the user to verify if the main engine parameters have physical meaning.

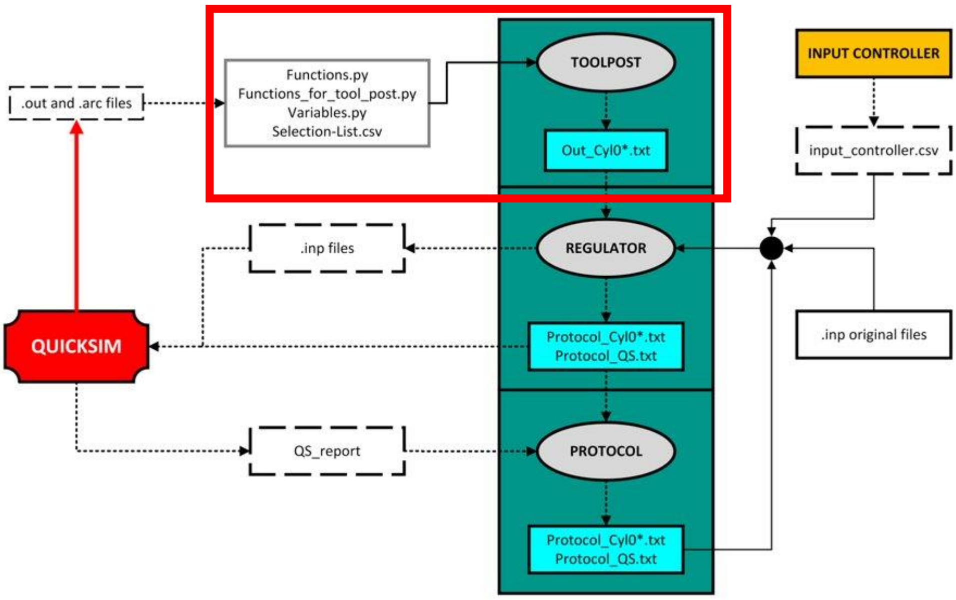


Figure 3.4.3 - *Logical scheme of the Tool-Post*

The *Tool-Post* needs of four tools to work correctly and they are the following, as show the figure 3.4.3:

- *Selection-List.csv:* it contains a list of variables that the Tool-Post will write in the out_Cyl files;

- *Functions_for_tool_post.py & Functions.py:* these python tools process the .out files from QuickSim in order to allow the Tool-Post to write the out_Cyl files correctly;
- *Variables.py:* it contains a list of variables needed for making the interface between *QS* and *QuickReg* possible.

In the next section, it will explain how the most important part of the *QuickReg* package works: the *regulator*, which contains all the control strategies.

3.5 Regulator

In this subsection, it is analyzed the *Regulator* tool, which is one of the most crucial components of the *QuickReg* package. The *Regulator*, like most of the other tools, is written in *Python* language and contains the control strategies. The figure 3.5.1 shows the main strategies present inside it.

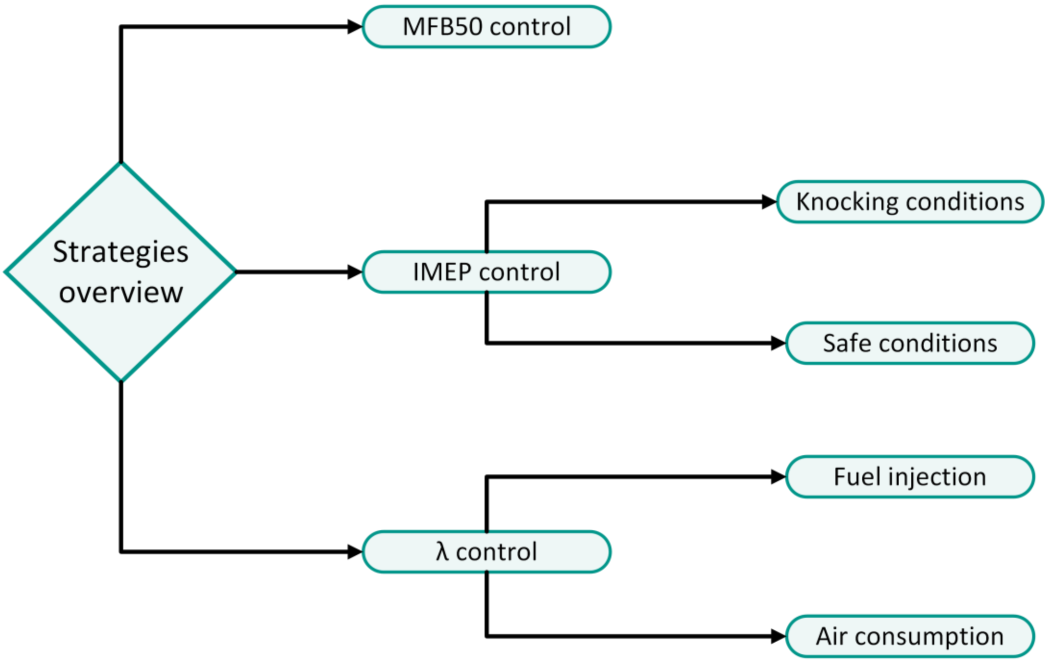


Figure 3.5.1 - *Strategies overview*

More details about the strategies will be given during the subsection.

As highlighted in Figure 3.5.2, the *Regulator* tool takes as input the *Out_Cyl0*.txt* file provided by the *Tool-Post*, the input_controller.csv set by the user, the *Protocol_Cyl0*.txt* file (which will be discussed later on), and the original *.inp* files such as the boundary conditions provided by *QuickSim*, which should be modified by the *Regulator*.

After a regulation is completed, the Regulator writes new boundary conditions files (*.inp*) that will be read by QuickSim in the next iteration. However, the effects resulting from the new boundary condition will only be noticeable in the next cycle, after the completion of the cycle.

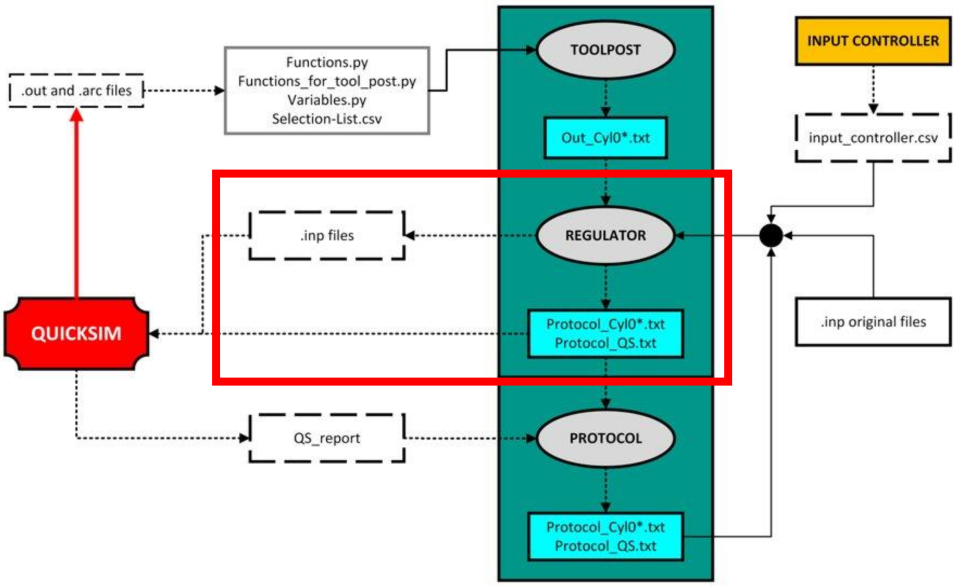


Figure 3.5.2 - *Logical Scheme of the regulator tool*

Before beginning the checking process, the simulation needs to reach convergence. The key parameter used to determine whether the simulation is convergent is the air consumption, which is evaluated with other critical engine parameters using the *ToolPost*.

The condition that determines the initial convergence of the simulation is based on the cyclical variability of air consumption, which is evaluated with respect to the range specified in the *input controller*. This evaluation is performed by calculating the ratio between the air consumption of the current cycle (i-cycle) and that of the last cycle (n-cycle), where i ranges from 1 to n-1. If the variation in air consumption exceeds the specified target range, it is assumed that convergence has not been achieved yet.

As mentioned before, the *Regulator* tool works on the principle of real-time regulation of the simulation. The *QuickSim* simulation progresses through iterations, with each step being either fixed or variable. A shorter angular step is preferred for higher accuracy during certain processes such as combustion or injection. Whenever *QuickSim* evaluates a crank angle where a parameter needs to be controlled, the simulation is paused and the tool is automatically called to take over the control.

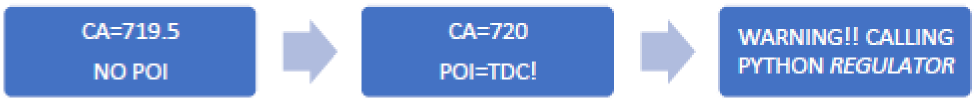


Figure 3.5.3 - *Step by step interaction between Regulator tool and QuickSim*

3.5.1 Mass Fraction Burned 50 control strategy

Mass Fraction Burned 50, usually called *MFB50*, is a term used in engine combustion analysis, which refers to the point in the combustion process where 50% of the fuel mass has been burned. It is often used as a metric for measuring engine performance and can be used as a target for combustion control strategies.

This control strategy involves adjusting the ignition timing to achieve a specified optimal center of combustion, while also adhering to the user-imposed knock threshold. If the engine begins knocking while trying to achieve the desired mass fraction burned at 50% of the cycle (*MFB50*) target, the tool adjusts the *MFB50* target value to avoid unnecessary changes in ignition timing. The tool reads the *Out_Cyl0** files, and if there are cycles within the knock threshold range, it selects the *MFB50* value for each of those cycles and sets the target as the value closest to the optimal *MFB50* specified by the user.

In the figure 3.5.4 is shown the logical scheme of the working process.

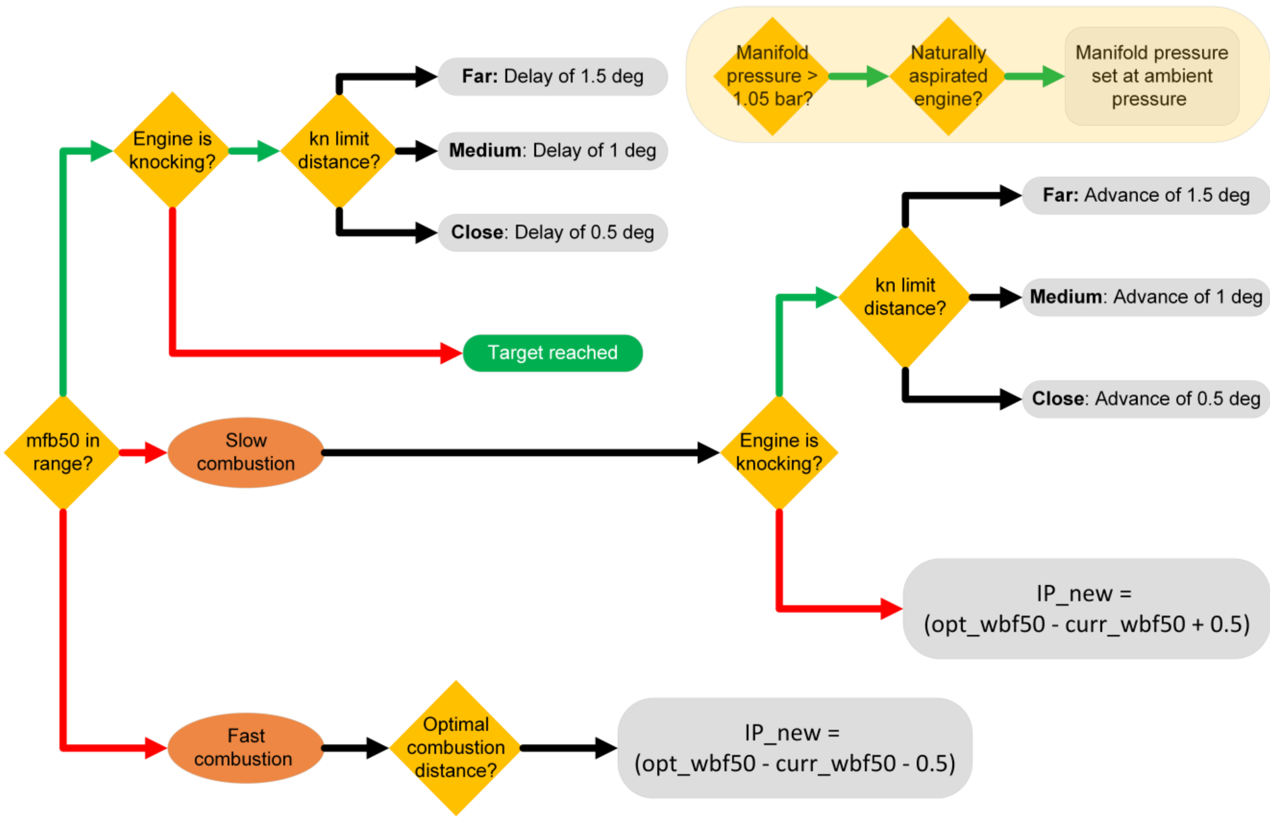


Figure 3.5.4 - *Logical scheme of the MFB50 regulation*

As shown, firstly, the regulator checks if the *MFB50* value is within the range or not, it reads the value from the *out_cyl0*.txt* file. If it falls within the range, the regulator then checks for engine knocking. If the engine is not knocking, the regulator achieves the set target. However, if the engine is knocking, the regulator takes action on the ignition point (IP) shifter as a function of the knock limit distance.

On the other hand, if the *MFB50* value falls outside the range, there are two options: slow combustion and fast combustion. For slow combustion, the regulator checks for engine knocking and acts on the IP shifter as a function of the knock limit distance. However, if the engine is not knocking, the regulator uses a dynamic IP shifter. This formula is also used in the case in which the engine has fast combustion. It's worth noting that the IP shifter is calibrated for different engines during previous simulations, while the formulation of the dynamic IP has been provided by a member of the 3D-CFD team.

One important detail about this strategy is related to the knock conditions. The priority is given to knock control because it is a crucial condition for engines, and the regulator must be as realistic as possible. Therefore, controlling knock is a top priority.

Once the correct value of the IP shifter is chosen, the *Regulator* modifies the original boundary conditions related to the ignition point, named *qs-spark-ign-sp.inp,* with the new value of ignition point.

3.5.2 Indicated Mean Effective Pressure control strategy

The Indicated Mean Effective Pressure, usually called IMEP, is a metric used in the field of internal combustion engines to evaluate the performance of an engine during a cycle. It represents the average pressure that acts on the piston during the power stroke and is calculated by dividing the net work done by the engine during a cycle by the total volume swept by the piston. IMEP is used as an indicator of the engine's efficiency, as well as its power output.

As shown in the figure 3.5.1, the Regulator includes two main conditions for controlling the IMEP in case of turbocharger engines: one when the engine is experiencing knocking and the other when the engine is operating safely, i.e., without any knocking.

The two strategies will be explained separately below.

- *Strategy in knocking condition*

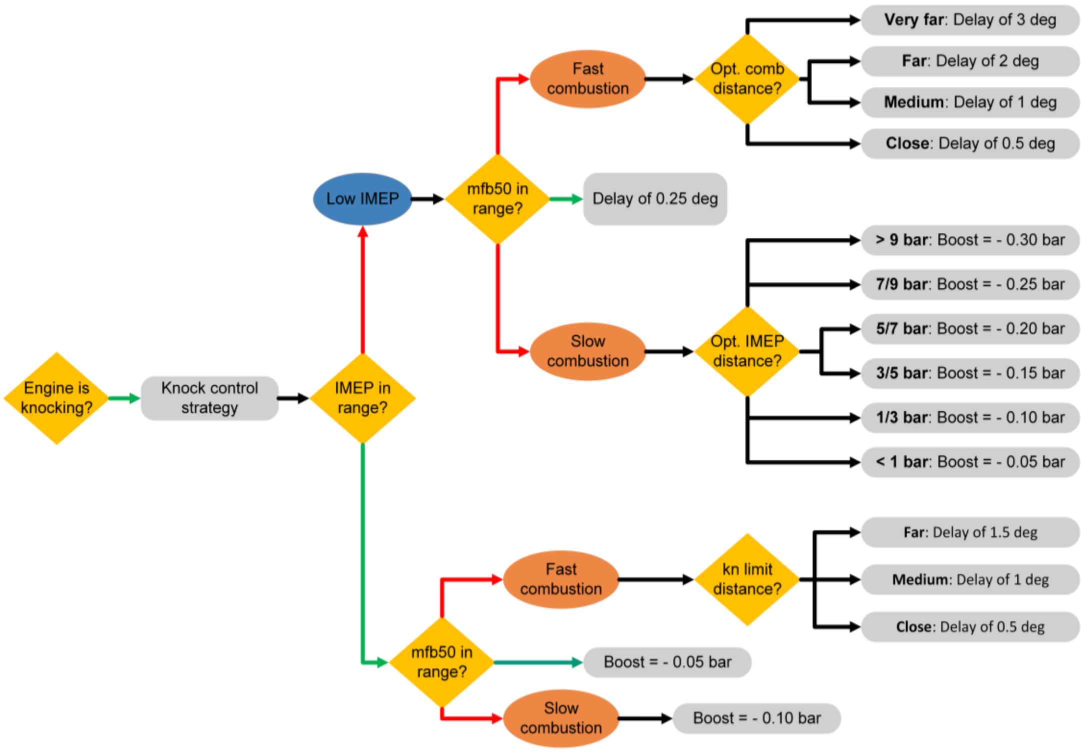


Figure 3.5.5 - *Logical scheme for Imep control strategy in knocking conditions*

The figure 3.5.5 shows the logical scheme for imep control strategy in knocking conditions.

The *Regulator* tool initially verifies the presence of engine knocking before proceeding to assess the IMEP value. If the IMEP value is found to be too low, the tool evaluates whether the MFB50 is within the desired range or not. In case of fast combustion or MFB50 within the range, an IP shifter will be evaluated as a function of the optimal combustion distance from the value imposed by the user. However, in case of slow combustion, the regulator will act on the boost pressure shifter to increase the speed of combustion.

On the other hand, if the IMEP value is within the desired range, the Regulator evaluates whether the MFB50 is within the desired range or not. If the combustion is too fast, the Regulator evaluates an IP shifter as a function of the engine's knock limit to slow down the combustion. However, if the MFB50 is within the desired range or the combustion is slow, the tool evaluates a boost pressure shifter. Once the necessary evaluations have been completed by the regulator and the appropriate value of shifter has been selected, the regulator modifies the boundary conditions acting on the file named *Qs-manifolds-im.inp*.

It's important to note that the regulation of IMEP can only be done in the case of a turbocharged engine, as this strategy primarily acts on the boost pressure, which is present only in this type of engine.

- *Strategy in safety conditions*

Figure 3.5.6 - *Logical scheme for Imep control strategy in safe conditions*

In the figure 3.5.6 is shown the IMEP control strategy for safe engine conditions, meaning that the engine is not experiencing knocking.

The regulator first checks the IMEP value. If it is too low, the regulator evaluates an increase in the boost pressure shifter to raise the IMEP value, based on the optimal IMEP distance.

If the IMEP value is within the desired range, the regulator checks the combustion speed. In this case, an advance or a delay of the IP shifter is evaluated based on the optimal combustion distance, which is determined by the user during the pre-processing phase.

Finally, if the IMEP value is found to be too high, the regulator evaluates a decrease in the boost pressure shifter to lower the IMEP value, based on the optimal IMEP distance.

As with the previous strategies, once the regulator selects the appropriate values for the shifters, it will evaluate the new boundary conditions and proceed to modify the *Qs-manifolds-im.inp* file.

3.5.3 Lambda control strategy

As mentioned in the first chapter, λ, indicates the ratio of the actual air-to-fuel ratio to the stoichiometric air-to-fuel ratio.

Controlling λ is one of the goals of the *Regulator*, given its importance as a parameter in engine calibration. The control strategy provides two methods of regulation: one involves changing the air consumption acting on the boost pressure shifter while keeping the mass of fuel injected for each cycle constant, while the other involves keeping the air consumption constant and changing the mass of fuel injected acting on the injection timing. More details are provided in the following section.

- *λ control changing the mass of fuel injected*

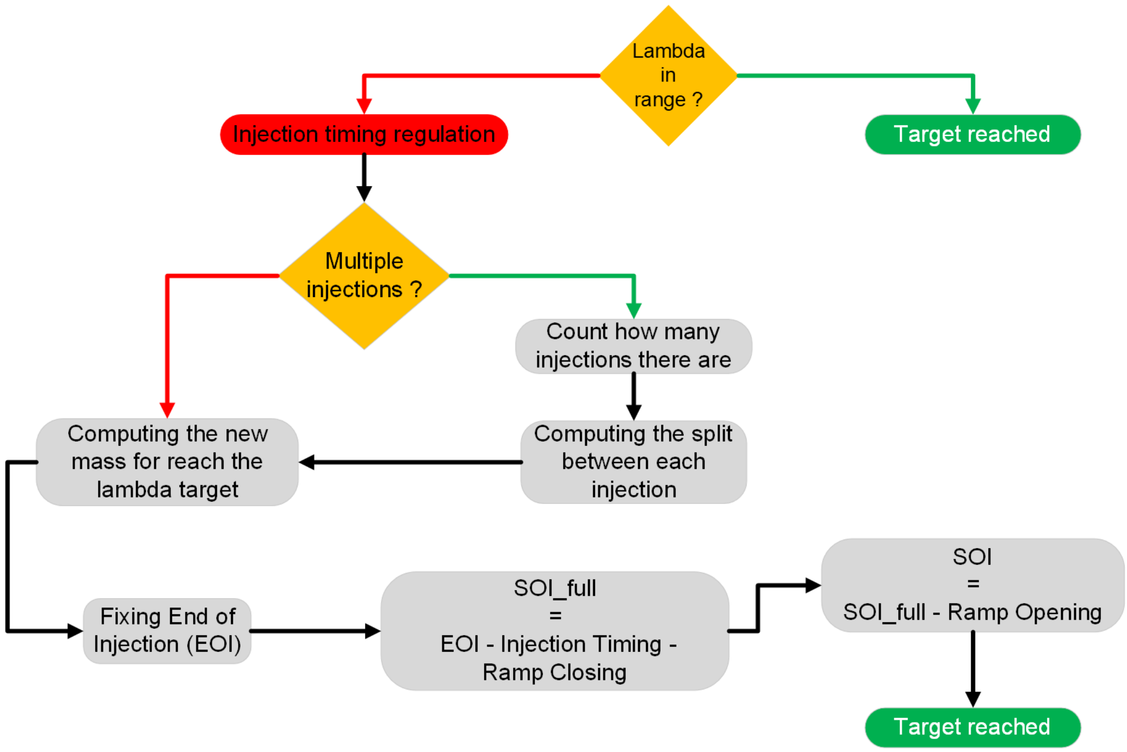


Figure 3.5.7 - *Logical scheme for lambda regulation changing the mass of fuel injected*

As stated earlier, this strategy regulates the air-to-fuel ratio at the ignition point. This is achieved by controlling the injection timing, which refers to the angular interval during which fuel is injected into the cylinder. The figure 3.5.7 shows the logical scheme of the regulation.

Firstly, the *Regulator* checks, thanks to the *out_cyl0** written by the *Tool-Post,* if the lambda is within the range previously set by the user. If the lambda is within range, the regulator proceeds with other regulations. However, if it is out of range, the regulator evaluates a new injection mass.

The regulation is based on the assumption that the air mass at the ignition point of the current cycle is in line with the expected values. Therefore, the target fuel mass to be injected is evaluated using the following equation:

$$Fuel_{target} = \frac{am_{IP}}{\lambda_{target} \cdot \alpha_{st}}$$

Equation 3.5.1 *- Fuel target equation*

Where:

- am_{IP} is the air mass at ignition point;
- λ_{target} is the lambda target imposed by the user in the input controller;
- α_{st} is the alpha stoichiometric for the fuel get in consideration.

As stated earlier, the regulation is done modyfing the injection timing.

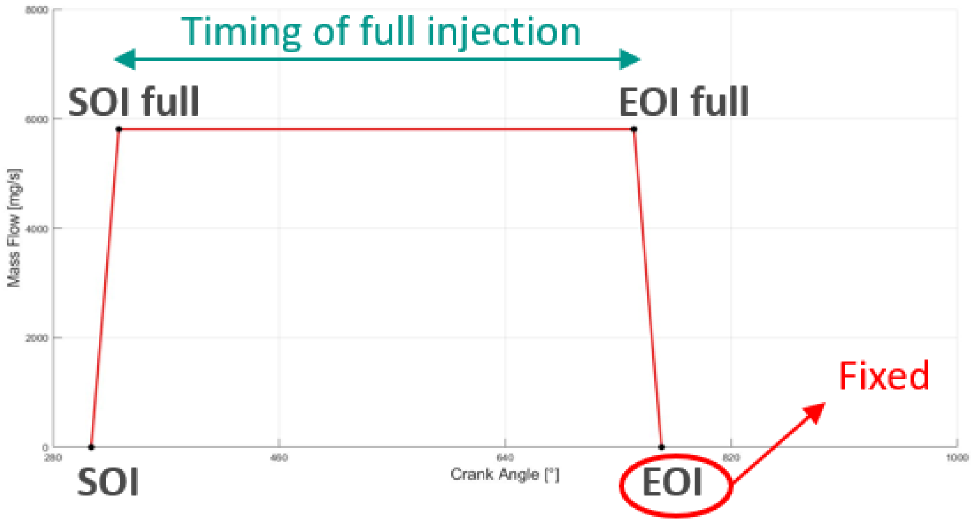


Figure 3.5.8 - *Injection law*

The figure 3.5.8 shows a typical injection law of an injector. The regulation strategy involves keeping the EOI (end of injection) fixed, and evaluating the crank angle corresponding to the full EOI, which is the point where the injector is fully open. The area between the points EOI full and EOI can be approximated by a triangle and should be constant since it depends on the geometrical parameters of the injector.

The formula to calculate the mass presents inside this area is:

$$Fuel_{ramp} = \frac{(ramp \cdot Fuel\ for\ CA)}{2} \quad [mg]$$

Equation 3.5.2 - *Fuel ramp formula*

That is the area of triangle with base ramp and height the fuel for CA.

The terms inside the formula are:

- *Ramp*: it is measure in *ms* and it rapresents the time needed to fully opening/closing the injector;
- *Fuel for CA*: measured in *mg/CA* and it rapresents the fuel injected for CA.

Once the fuel inside the rampen is computed, it’s possible to get the mass of fuel when the injector is fully opens with the easy formula:

$$Fuel_{full\,injection} = Fuel_{target} - 2 \cdot Fuel_{ramp} \quad [mg]$$

Equation 3.5.3 - *Fuel full injection*

Thanks to this two parameters the injection timing can be evaluated with the following equation:

$$Injection\ timing = \frac{Fuel_{full\,injection}}{Fuel\ for\ CA} \quad [deg]$$

Equation 3.5.4 - *Injection timing*

After that, the start of injection full (SOI full) can be evaluated by means of the formula:

$$SOI\ full = EOI\ full - injectiontiming \ \ [deg]$$

Equation 3.5.5 - *Start of injection full*

And then the start of injection (SOI):

$$SOI = SOI\ full - ramp$$

Equation 3.5.6 - *Start of injection*

Therefore, once the necessary variables are obtained by the *Regulator* tool, it can proceed to modify the boundary conditions to achieve the target set by the user. The file modified by this regulation is called *qs-injection-fu.inp.*

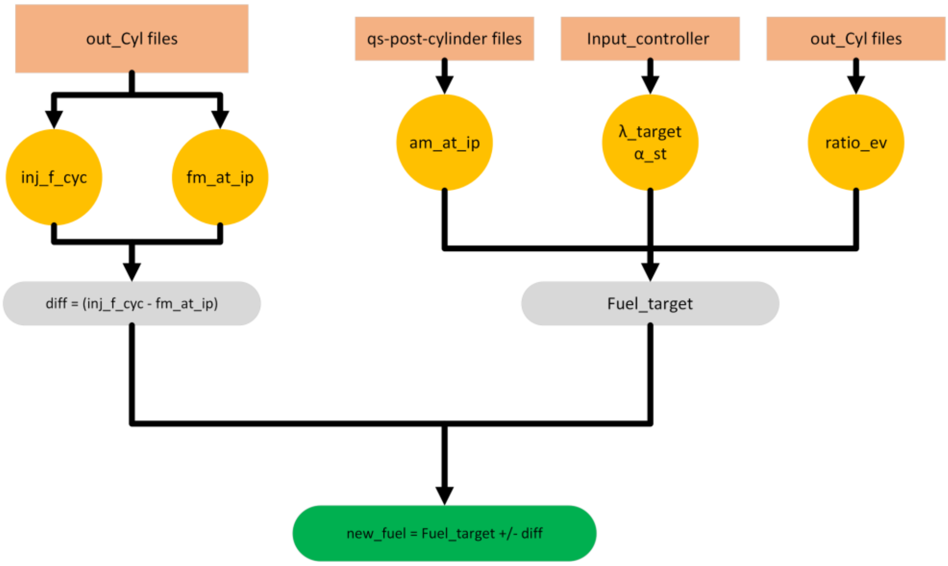


Figure 3.5.9 - *Injection strategy improvement for PFI engines*

The figure 3.5.8 shows an improvement on fuel injection strategy dedicated for port fuel injection (PFI) engines. In PFI engines, fuel is injected into the air stream before it enters the combustion chamber, allowing for a more homogeneous air-fuel mixture. The fuel is typically delivered to the engine by a fuel pump, which sends it through a fuel line to the fuel rail and then to the injectors, which are mounted in the intake manifold.

As stated earlier, in port fuel injection engines, fuel is injected into the manifold, and some of it is lost along the way to the combustion chamber. This lost fuel represents a discrepancy between the fuel target and the actual fuel quantity that reaches the combustion chamber. The goal of the improvement strategy is to reduce this discrepancy and minimize the number of cycles required for the simulation to converge to the new fuel target.

As can appreciate, first the regulator compute the difference between the fuel injecyed for cycle and the fuel mass present at ignition point reading the *out_cyl* file.

$$diff = inj\_f\_cyc - fm\_at\_ip$$

Equation 3.5.7 - *Fuel losses*

Where:

- *Inj_f_cyc* is the quantity of fuel injected in each engine cycle;
- *Fm_at_ip* is the quantity of fuel present at ignition point.

The difference will add or subtract to the fuel target (equation 3.5.1) compute previously by the regulator.

Regarding the PFI engines, the regulator is capable also to control the lambda in case the injector is still opened between two cycle, usually called injection over-cycle end. The logic is the same used in the first case, there are just some considerations on the respective injection law.

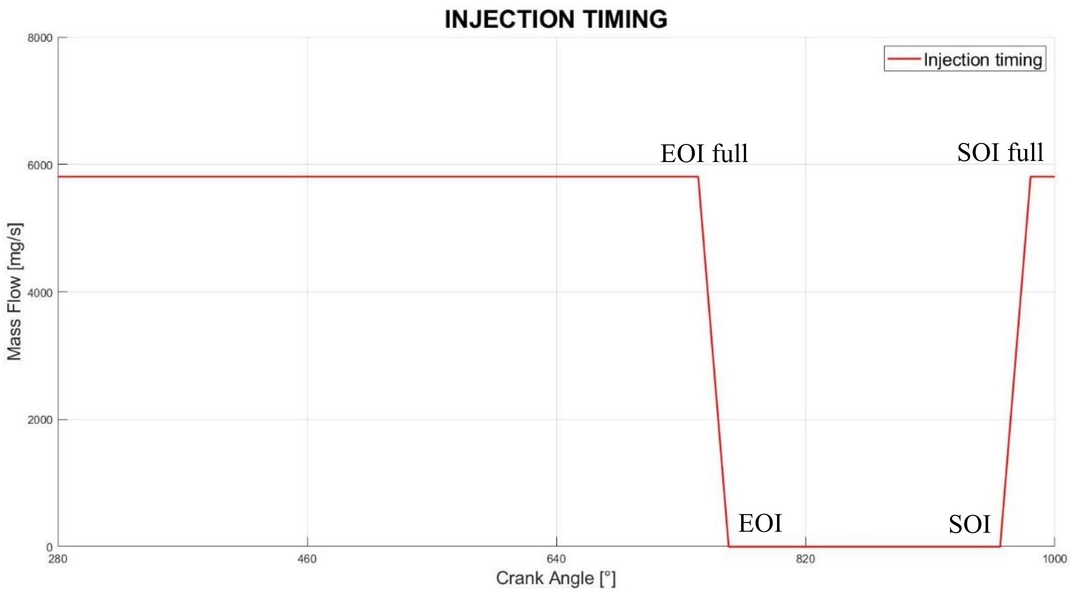


Figure 3.5.10 - *Injection law for over cycle end strategy*

Figure 3.5.10 illustrates a typical case of injection beyond the end of the cycle. As previously mentioned, in high-speed engines, the injector may remain open for a longer period of crank angle degree due to the reduced time available to complete a cycle compared to the time required for the injector to fully inject the fuel.

Once the new start of injection is computed using equation 3.5.6, a comparison is made to determine if the injection will occur over cycle end. If the new start of injection is less than zero, it indicates that the injection law exceeds the cycle end.

The Regulator can also control the lambda in case of multiple injections. The user should specify beforehand whether the injection will be single or multiple. As shown in figure 3.5.7, the Regulator counts the number of end-of-injections present using a counter, and then evaluates the split ratio between each injection that remain constant between the original and the regulation injection.

In the figure 3.5.11 is shown a comparison between the original law and the regulated one. The latter are commonly used in gasoline direct injection (GDI) engines to reduce NO_x and particulate matter, as well as to improve combustion stability. More details on this topic are provided in the first chapter.

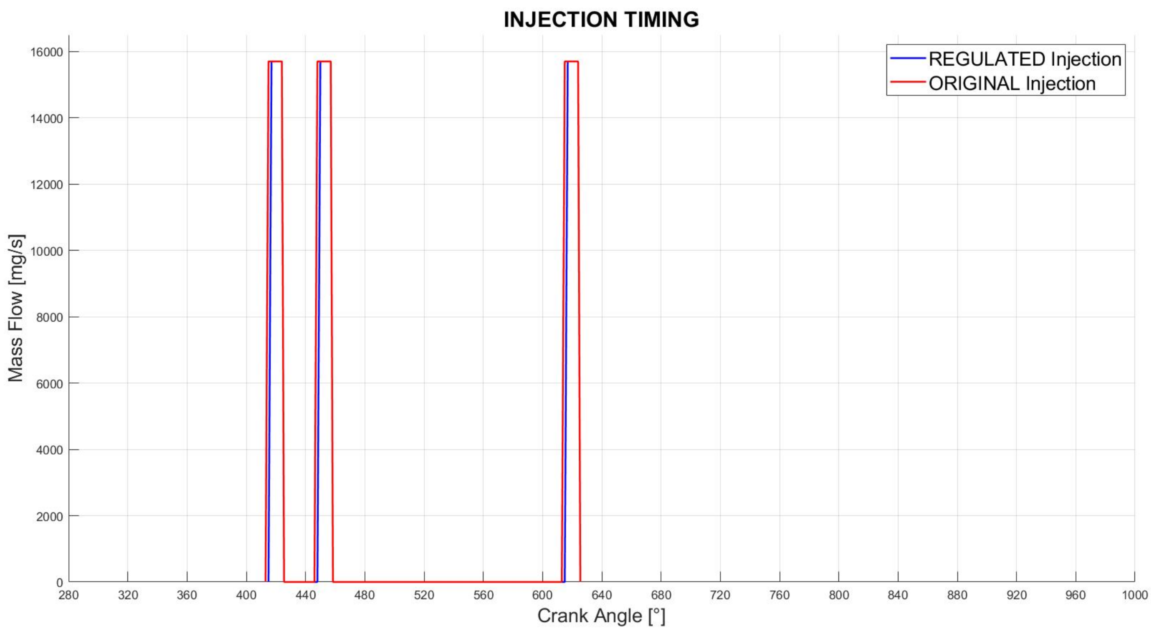


Figure 3.5.11 - *Multiple injections law original vs regulated*

- *λ control changing air consumption*

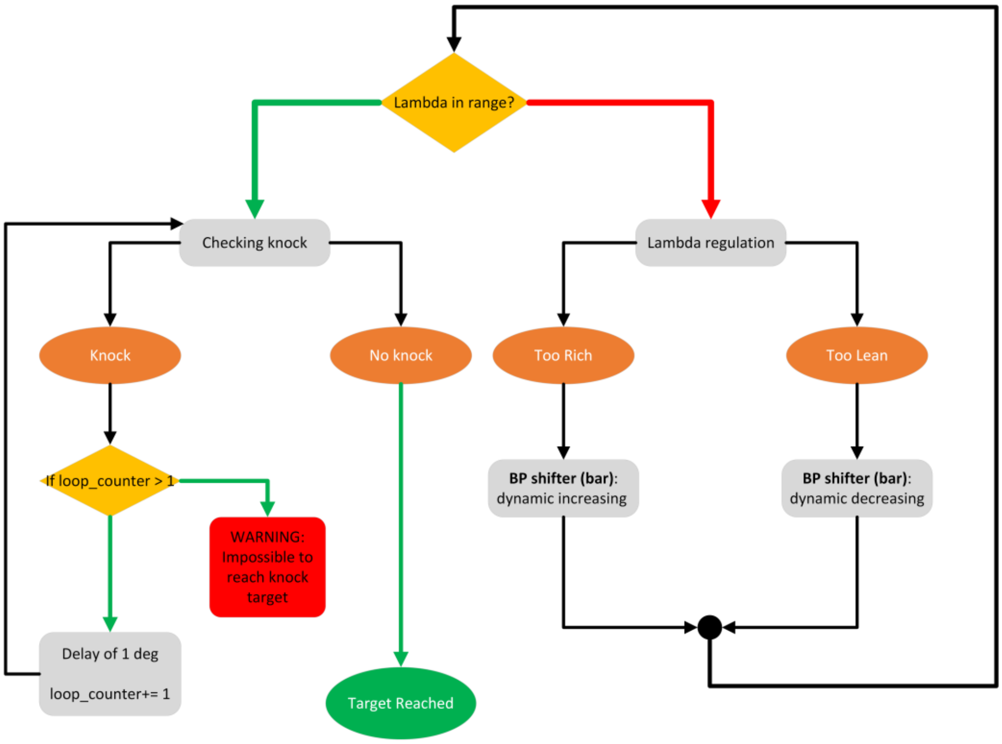


Figure 3.5.12 - *Logical scheme for lambda regulation changing the air mass by means of boost pressure*

The *regulator* can also control the λ, exclusively in a turbocharger enginge, adjusting the air consumption by means of a boost pressure shifter.

As shown in Figure 3.5.9, the regulator first checks if the lambda is within the range set by the user. If it is out of range, the regulator evaluates a boost pressure shifter as a function of the air-fuel ratio.

If the mixture is too rich, the formula used by the regulator to evaluate the boost pressure shifter is:

$$BP_{shifter} = \frac{(\lambda_{target} - \lambda_{range}) - \lambda}{\lambda_{target} + (\lambda_{target} - \lambda)}$$

Equation 3.5.8 - *Boost Pressure shifter for too rich mixture*

Where:

- λ_{target} is the lambda value set by the user;
- λ_{range} is the corresponding range of acceptability;
- λ is the current value of lambda reads from the *out_cyl0**.

Instead if the mixture is too lean, the formula used by the regulator to evaluate the boost pressure shifter is:

$$BP_{shifter} = -\frac{\lambda - (\lambda_{target} - \lambda_{range})}{\lambda}$$

Equation 3.5.9 – *Boost Pressure for too lean mixture*

The meaning of the values inside the equation 3.5.6 is the same shown in the previous one.

Once the boost pressure shifter has been evaluated, the *regulator* checks the new λ value. If it is within range, a knock check is performed. If the engine is not knocking, the target value has been reached. However, if the engine is knocking, the regulator will delay the ignition point by one degree to reduce the knock occurrence. If the engine is still knocking, it means that the target value set by the user cannot be reached.

Finally, depending on the case in which the *regulator* falls the files that will be modified are respesctivetly *qs-spark-ign-sp.inp,* to change the ignition point and *qs-manifolds-im.inp* to change the boost pressure.

3.6 Protocol

The previous paragraphs discussed the *Regulator* tool in detail. To maintain traceability of the actions performed by the code, a text file named *Protocol_Cyl0*.txt* is generated to record the history of the actions taken and also in this case the number of protocol is the same to the number of cylinders.

The main objectives of the *Protocol_Cyl0*.txt* file are:

1. To provide a timeline that enables users to track the actions performed by the *Regulator* during the simulation, which is useful for obtaining a quick summary of what has been done;
2. To serve as an interface between *QuickSim* and the *Regulator* tool.

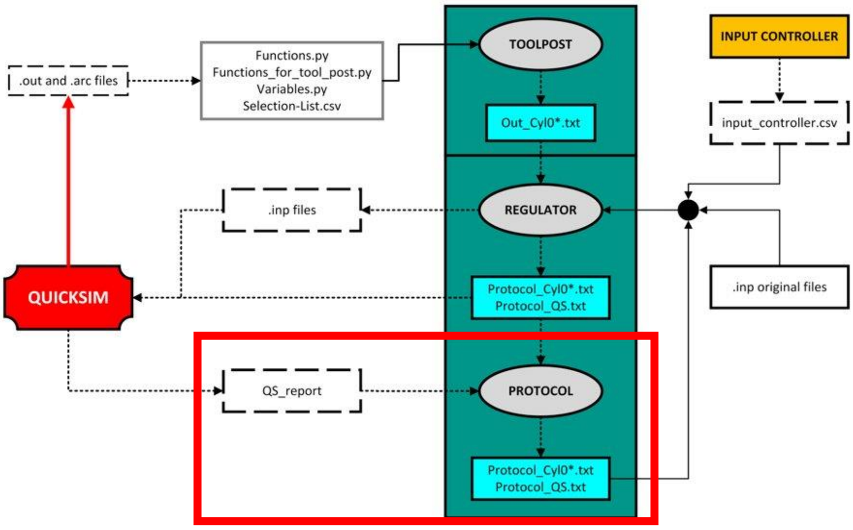


Figure 3.6.1 - *Logical scheme of the protocol*

As shown in Figure 3.6.1, the Protocol tool takes as input the previous Protocol_Cyl0*.txt files and updates them with the actions performed by the Regulator and QuickSim. If the Protocol_Cyl0*.txt files do not exist yet, for example if it is the first cycle of simulation or the user has deleted them, the tool is capable of creating them.

Index	Cylinder	Crank_Angle	Cyc	Element_category	Element_sub	Execute	Parameter
1012171448	1	999.5	10	0	1	1	Target_Knock_Index_reached
1012041315	1	999.5	9	0	1	1	Waiting_after_IP_change
1011151051	1	999.5	8	7	1	1	IP_change
1011015819	1	999.5	7	0	1	1	Waiting_after_IP_change
1010123927	1	999.5	6	7	1	1	IP_change
1008084136	1	999.5	5	0	1	1	Waiting_after_IP_change
1007194321	1	999.5	4	7	1	1	IP_change
1001023418	1	999.5	3	0	1	1	Waiting_initial_convergence
930142154	1	999.5	2	0	1	1	Waiting_initial_convergence
0	1	999.5	1	7	1	1	Start_IP
-1	-1	-1.0	-1	-1	-1	-1	-1

Actual_value_wbf50	Target_value_wbf50	Actual_value_iit	Actual_value_IMEP	Shifter	New_mf	Edited_File	Command
734.85565	734.8	5.190	23.30945	0.0	-1	No_Edited_File	REGULATOR.PY
735.73020	708.0	3.370	23.19465	0.0	-1	No_Edited_File	REGULATOR.PY
736.68000	708.0	1.095	23.10290	-1.5	-1	qs-spark-ign-sp_01	REGULATOR.PY
737.57700	708.0	0.575	22.95900	0.0	-1	No_Edited_File	REGULATOR.PY
738.31600	708.0	0.075	22.80500	-1.5	-1	qs-spark-ign-sp_01	REGULATOR.PY
737.68300	708.0	0.355	22.91800	0.0	-1	No_Edited_File	REGULATOR.PY
733.79400	708.0	8.460	23.33900	1.0	-1	qs-spark-ign-sp_01	REGULATOR.PY
730.43200	735.0	15.975	23.64410	0.0	-1	No_Edited_File	REGULATOR.PY
730.74800	735.0	15.350	23.83150	0.0	-1	No_Edited_File	REGULATOR.PY
0.00000	735.0	0.000	0.00000	-1.0	-1	-1	REGULATOR.PY
-1.00000	-1.0	-1.000	-1.00000	-1.0	-1	-1	-1

Figure 3.6.2 - *Protocol_Cyl0*.txt example*

The figure 3.6.2 shows an example of the *Protocol_Cyl0*.txt* file, as can be seen a generic row of the file is composed by:

1. *Index:* it represents the matrix code of the action structured as shown in the figure 3.6.3:

2020	07	27	1440	52
Year	Month	Day	Hour	Second

Figure 3.6.3 - *Matrix code of the action performed by the Regulator tool*

   Thanks to the matrix code is always possible to trace the simulation regulation history;
2. *Cylinder*;
3. *Crank_Angle*, is the crank angle in which the regulator is called by QuickSim;
4. *Cyc* is the actual cyle of the simulation;
5. *Element_Category,* is the number convention based on the regulation type;
6. *Element_Sub* referred to cylinder number;
7. *Execute,* needed for QS – Regulator interface. When QS read Execute = 0 understands that a regulation must be done, otherwise if reads 1 there is no regulation to be done*;*
8. *Paramter,* regulation type;
9. *Actual_value_wbf50,* actual value of the centre of the combustion in CA (in QS convention);
10. *Target_value_wbf50,* target value of mfb50 for optimal combustion (set by user or re-evaluated by the tool);
11. *Actual_value_iit,* actual value of the knock index in percentage;
12. *Actual_value_IMEP,* actual value of IMEP in bar;
13. *Shifter,* shifter for new values. :
    - IP_change: CA of delay or advance of the ignition point;
    - BP_change: bar added or subtracted to the intake manifold pressure;
14. *New_mf,* new mass of fuel evaluated. If the value is equal to -1 ther is no fuel/lambda regulation;
15. *Edited_file,* .inp file modified based on the regulation type;
16. *Command,* python tool called by QS.

During the initial cycles, the tool waits for the simulation to converge before proceeding with parameter checks. If a parameter specified in the *Input_Controller.csv* falls outside the specified range, a new input file is generated and the name of the created file is reported in *Protocol_Cyl0*.txt*.

In the next section, the final tool of the QuickReg package will be explained.

3.7 Restart

QuickSim offers the unique advantage of allowing users to restart simulations from any previously analyzed point, without the need to start from the beginning the simulation. This capability is highly advantageous as it saves a significant amount of time. The *Restart* tool is an essential feature of the *QuickReg* package as it enables the management of input files for the simulation. The restart function is accomplished by exploiting the traceability information recorded in *Protocol_Cyl0*.txt*, which allows the user to reconstruct the simulation history and choose the appropriate QSI file to impose on the restarted simulation.

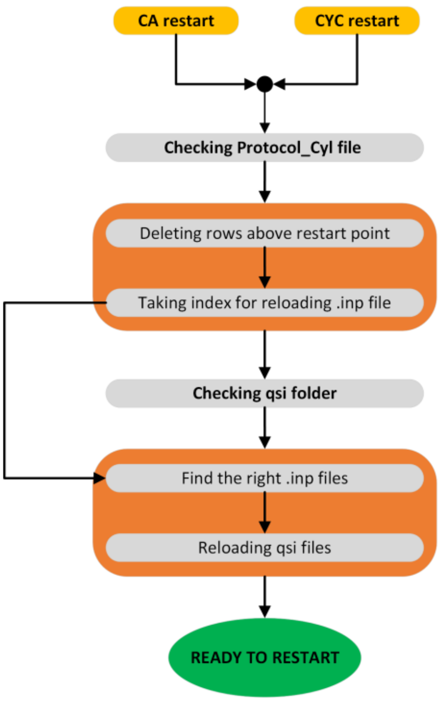


Figure 3.7.1 - *Logical scheme of the restart tool*

As shown in Figure 3.7.1, the Restart tool requires as input the crank angle and cycle at which the user wants to perform the restart. Once these two variables are set by the user in the *QuickSim* command window, the *Restart* tool is ready for use.

First, the tool checks the *Protocol_Cyl0** file and deletes all the rows above the specified restart point. Then, the tool takes all the indices present in the protocol in order to reload the corresponding boundary conditions (.inp files) at the restart point. Using these indices, the tool navigates to the qsi folder where all the backup boundary condition files are stored and retrieves the correct file to reload it in the simulation. At this point, the restart can be executed.

Chapter 4

4 Validation of the tool

Once the off-line development of the QuickReg tool was completed, it was coupled with the 3D-CFD tool QuickSim as mentioned in the previous chapter for the validation and verification. Validation is intended to ensure results have a physicical meaning. Verification is intended to check that the algorithm meets a set of design specifications.

To guarantee stability and validity, a series of tests were conducted, starting with simple simulations and gradually progressing to more complex scenarios. Various engines were used in this study, including a mono-cylinder engine and three multi-cylinder engines operating under different conditions.

The first engine was a Gasoline Direct Injection (GDI) engine commonly used in passenger cars,

the second engine was a high-performance Port Fuel Injection (PFI) engine, while the last one was a Gasoline Direct Injection (GDI) powered by methane and with pre-combustion chamber.

The validation of the tool with different engines involved regulating the lambda at the Ignition Point (IP) while maintaining a constant air mass and adjusting the fuel through injection timing regulation and using different injection strategy.

4.1 Multi-cylinder SI direct injection engine

The lambda regulation by means of injected fuel mass was applied to a simulation of a turbocharged multi-cylinder engine with the specification outlined in Table 4.1.1. The engine has been used for the investigation of post oxidation in the exhaust manifold in the FVV project *1456 Innovative RDE Engine-Out Emission Reduction*:

Engine	**Direct injection spark ignition**
Bore	79.7 mm
Stroke	81.1 mm
Number of cylinders	4
Displacement	1618 cm^3
Compression ratio	10.5
Valves	16

Table 4.1.1 - *Engine Propierties*

In the figure 4.1.1 is shown the mesh of the engine:

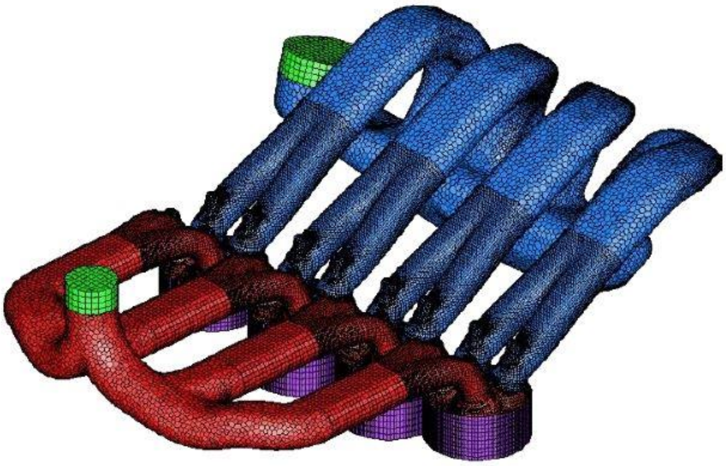


Figure 4.1.1 – *Mesh of multi-cylinder model*

The figure depicts the essential components of the engine, including the air-box, intake channels, four cylinders, and exhaust channels. The computational domain, spanning from the air-box to the turbine inlet, simplifies the selection of boundary conditions.

As mentioned earlier, two types of injection strategies were employed. Initially, the simulation was conducted using a single injection technique with different lambda values for each cylinder. Once convergence was achieved, the simulation was then run using a multiple injection strategy with the same lambda value for all cylinders.

4.1.1 Single Injection

The operating point of the engine used for validate the single injection is shown in table 4.1.2:

Engine	Direct injection spark ignition
RPM	1600
Load	High load

Table 4.1.2 - *Operating Point of Multi-Cylinder GDI engine*

The first part of the simulation was developed with different values of lambda target for each cylinder and with a single injection, as shown in figure 4.1.2:

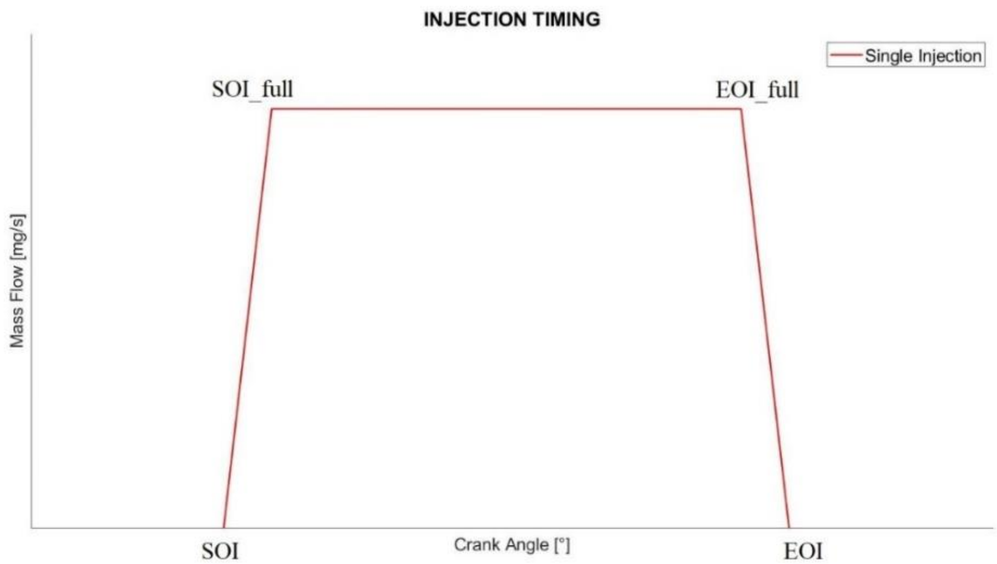


Figure 4.1.2 – *Cylinder one single injection law of multi-cylinder engine*

The figure highlights the main points of an injection law, respectively:

- SOI = Start of injection;
- SOI_full = Start of full injection, in which the injector is completely open;
- EOI_full = End of full injection;
- EOI = End of injection.

More details are described in the chapter 3.

During the pre-processing of the simulation an input controller file must be set. In the next table (4.1.3) are shown the variables needed for the *regulator tool* to know which strategy inside the code is enabled:

Variables	Target
Num_inj	1
Lambda_split_cyl	1
Pos_inj	GDI
Type_injection	SINGLE
Cycle_L_at_IP_Change	1

Table 4.1.3 – *Boolean variables for a single injection*

For the validation, a "*lambda_split_cyl*" value equal to one has been selected, allowing the regulator tool to evaluate a different lambda value for each cylinder by modifying the corresponding injection files. The variable "pos_inj" is set to "GDI", indicating that the engine utilizes a direct injection system. Additionally, the variable "type_injection" is set to "SINGLE," indicating that only a single injection occurs during the engine cycle. Furthermore, the variable "Cycle_L_at_IP_change" is set to one, indicating that the regulator waits for one cycle before performing another regulation.

In the table 4.1.4 there are other input variables that need to be configured, such as the injector characteristics required to compute the injected mass flow and the target lambda to achieve set by the user.

Target Variables				
Variables	**Target**	**Range**	**Unit**	**Cyl.**
L_at_ip	1	0.02	-	01
L_at_ip	0.9	0.02	-	02
L_at_ip	1.1	0.02	-	03
L_at_ip	1.3	0.02	-	04
Injectors Characteristics				
Outward opening	15	cm^3/s		
Ramp opening	0.15	ms		
Ramp closing	0.15	ms		
Corrector factor	1.4142	-		

Table 4.1.4 – *λ target and injector characteristics*

In the section of the injectors characteristics in the table 4.1.4 are shown the value of the outward opening that correspond to the max fuel that flows when the injectors is fully opened, the ramp opening and closing that is instead the time needed to the injectors to be fully open, finally the corrector factor is a parameter used to compensate for any discrepancies between the actual amount of fuel delivered by the injector and the desired amount.

In the pre-processing phase of the simulation, different boundary conditions files were set for each cylinder, in order to achieve the lambda targets imposed by the user keeping constant the air mass.

Table 4.1.5 provides a summary of the settings.

Cylinder	File	Starting lambda	Action	Regulated lambda
01	*'qs-injection-fu_01'*	1.21	Enrichment	1
02	*'qs-injection-fu_02'*	1.21	Enrichment	0.9
03	*'qs-injection-fu_03'*	0.9	Impoverishment	1.1
04	*'qs-injection-fu_04'*	1	Impoverishment	1.3

Table 4.1.5 – *File and action where the tool acts*

During the first six cycles of the simulation, the focus is on achieving the convergence, and therefore, the regulator tool remains inactive. Once the convergence is reached, the tool begins evaluating the air consumption, checking if its variability is within ±2% so that the lambda regulation process can start.

Following the initial convergence phase, the regulator tool starts the evaluation of the new air-to-fuel ratio by adjusting the fuel injection timing. In the tool, the end of injection (EOI) is fixed, as delaying the fuel injection can influence negatively the combustion process. The opening and closing ramps are also predetermined based on the injector design. Consequently, the regulator calculates a new injection law by modifying the start of injection, ensuring it does not coincide with the exhaust valve closure, which, for this case, is set at 370.5 degrees in *QuickSim* convention.

In table 4.1.6 the original and the regulated fundamental points of the injection law are shown.

ORIGINAL				
Cylinder	**SOI [deg]**	**SOI_full [deg]**	**EOI_full [deg]**	**EOI**
01	433	435	454.5	456.5
02	433	435	454.5	456.5
03	425	427	454.5	456.5
04	428	430	454.5	456.5
REGULATED				
Cylinder	**SOI [deg]**	**SOI_full [deg]**	**EOI_full [deg]**	**EOI**
01	428	430	454.5	456.5
02	426	428	454.5	456.5
03	430.5	432.5	454.5	456.5
04	434	436	454.5	456.5

Table 4.1.6 – *Fundamentals points of injection law before and after the regulation*

In figure 4.3, the different injection laws for each cylinder are depicted, starting from cylinder one to four. It is noteworthy to observe how the regulator tool reaches to the target set by the user.

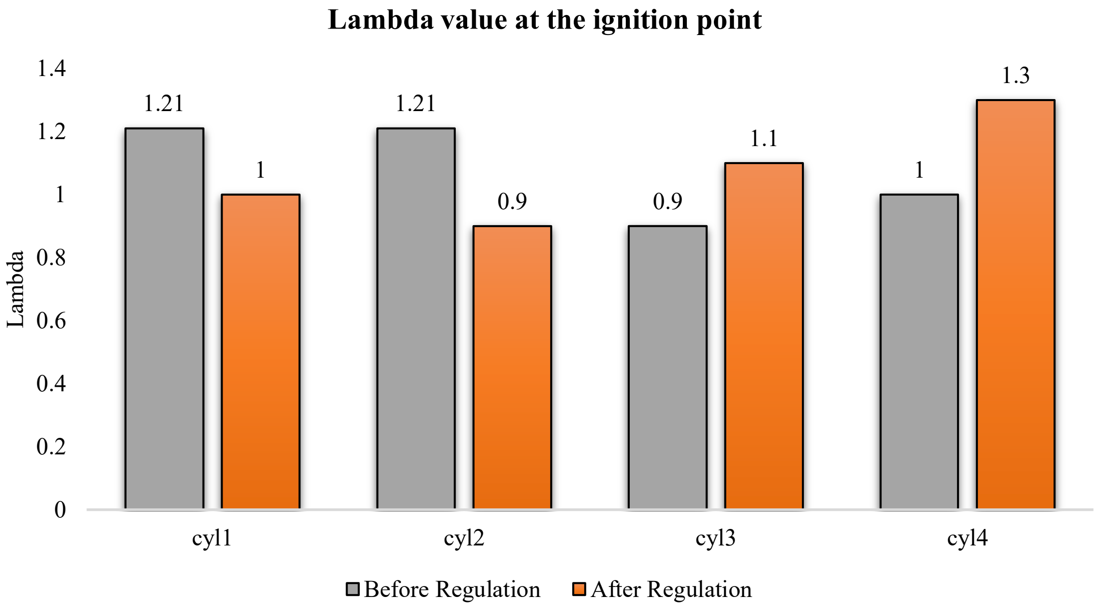


Figure 4.1.3 – *Lambda before and after regulation of the multi-cylinder engine*

In cylinder one, as shown in figure 4.3, the lambda was regulated from a lean condition to a stoichiometric condition, while in cylinder two, it was regulated from a lean to a rich condition. Both regulations showed an increase in the injected fuel mass to enrich the mixture. However, both cylinders four and three show a leaner mixture after regulation, resulting in a decrease in the injection law area from the original to the regulated state.

It is important to note that these regulations, involving different lambda values for each cyclinder, were carried out to test the stability and versatility of the tool.

Upon reaching cycle eleven, the injection timing regulation was stopped as the targets set in the input controller were achieved, and the simulation had reached stable conditions.

The subsequent sections analyze the impact of lambda changes on the main engine parameters for each regulation. Cycles seven and eleven were chosen for the study since cycle seven represents the last cycle without regulation, while cycle eleven provides more stable numerical results.

The figure 4.1.4 presents the main engine results obtained from the *tool post*. Additionally, a protocol, figure 4.1.5, details the actions performed by the regulator, using cylinder one as an example.

====== Results ======

VARIABLE	UNIT											
cycle	No	11	10	9	8	7	6	5	4	3	2	1
cyl_air_cons	kg/h	29.7417	29.73075	29.7261	29.69435	29.4152	29.41645	29.43495	29.4361	29.4626	29.4007	28.0821
cyl_m_air_IP	mg	619.97	619.7382	619.6612	619.1965	613.3767	613.3411	613.6005	613.7338	614.2583	612.8983	586.0208
cyl_fuel_cons	kg/h	2.06445	2.06445	2.0644	2.0645	1.68915	1.68915	1.68915	1.68925	1.6892	1.6893	1.6891
inj_m_fuel	mg/Cyc	43.0094	43.0089	43.0088	43.0103	35.191	35.1907	35.1907	35.1922	35.1911	35.1929	35.1906
cyl_m_fuel_IP	mg	43.021	43.0126	43.0098	43.0691	35.2094	35.2343	35.1707	35.2287	35.2017	35.5177	34.5553
cyl_power	kW	8.6295	8.6625	8.62375	8.5587	7.98435	8.00425	8.0621	8.0332	8.0718	8.0907	7.5502
ind_eff_abs	%	34.2003	34.3315	34.1781	33.9191	38.674	38.7707	39.0506	38.909	39.0972	39.1866	36.5715
imep	bar	16.0014	16.0626	15.9908	15.8702	14.8052	14.8421	14.9493	14.8957	14.9673	15.0023	14.0002
cyl_p_max	bar	59.2402	59.6252	58.9209	58.055	55.4032	55.9228	56.6507	56.109	56.5902	57.1861	52.4727
cyc_iit09m_max	%	0.01	0.01	0.0	0.0	0.28	0.39	0.26	0.0	0.0	0.0	0.0
cyl_egr_IVC	%	0.0426	0.0432	0.0437	0.0461	0.0467	0.0463	0.0459	0.0455	0.0463	0.0463	0.065
cyl_max_wbf	%	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
cyl_10_wbf	deg	727.545	727.5079	727.6053	727.6935	727.1636	727.1043	727.0408	727.1295	727.0573	727.04	727.1299
cyl_50_wbf	deg	739.7037	739.7216	739.8269	740.1203	739.2424	739.1196	738.9516	739.0062	738.9555	738.6634	739.6994
cyl_90_wbf	deg	751.8898	751.6777	751.9376	752.6924	752.1044	751.7152	751.2793	751.8074	751.3129	751.1357	752.9006
cyl_10_90_wbf	deg	24.3448	24.1698	24.3323	24.9989	24.9408	24.6109	24.2385	24.6779	24.2556	24.0957	25.7707
cyl_lambda_IP	-	1.0085	1.0083	1.0082	1.0061	1.2191	1.2182	1.2209	1.2191	1.2211	1.2076	1.1868
sp_lambda_IP	-	0.8458	0.869	0.8461	0.8444	0.8737	0.8751	0.879	0.866	0.8771	0.8515	0.9694
sp_egr_IP	kg/kg	4.6397	4.6513	4.7657	5.0294	5.3765	5.3181	5.2862	5.2466	5.3621	5.3401	6.9781
ip	deg	714.0	714.0	714.0	714.0	714.0	714.0	714.0	714.0	714.0	714.0	714.0
sp_r_max	mm	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
sp_dca_max	deg	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0
cyc_A_Weller	-	3.9	3.9	3.9	3.9	3.9	3.9	3.9	3.9	3.9	3.9	3.9
imb_p_01	bar	1.3778	1.3778	1.3778	1.3778	1.3778	1.3778	1.3778	1.3778	1.3778	1.3778	1.3769
imb_T_01	K	303.9906	303.9907	303.9905	303.9916	303.9909	303.9906	303.9916	303.9941	303.9942	303.9948	303.8772
emb_p_01	bar	1.2694	1.269	1.2695	1.2694	1.2701	1.2699	1.2689	1.2695	1.2695	1.2692	1.2785

Figure 4.1.4 - *Tool Post @Cylinder 1*

Index	Cylinder	Crank_Angle	Cyc	Element_category	Element_sub	Execute	Parameter	New_mf	Edited_File	Command
1116155256	1	999.75	11	0	1	1	WAIT_LAMBDA	0	No_Edited_File	REGULATOR.py
1115093143	1	999.75	10	0	1	1	WAIT_LAMBDA	0	No_Edited_File	REGULATOR.py
1114034358	1	999.75	9	0	1	1	WAIT_LAMBDA	0	No_Edited_File	REGULATOR.py
1112213503	1	999.75	8	0	1	1	WAIT_LAMBDA	0	No_Edited_File	REGULATOR.py
1111035357	1	713.75	7	4	1	1	L_AT_IP_REGULATION	43.01	gs-injection-fu_01	REGULATOR.py
1110083025	1	999.75	6	0	1	1	Waiting_initial_convergence	-1	No_Edited_File	REGULATOR.PY
1109012830	1	999.75	5	0	1	1	Waiting_initial_convergence	-1	No_Edited_File	REGULATOR.PY
1029142514	1	999.75	4	0	1	1	Waiting_initial_convergence	-1	No_Edited_File	REGULATOR.PY
1025040249	1	999.50	3	0	1	1	Waiting_initial_convergence	-1	No_Edited_File	REGULATOR.PY
1012144504	1	999.50	2	0	1	1	Waiting_initial_convergence	-1	No_Edited_File	REGULATOR.PY
0	1	999.50	1	7	1	1	Start_IP	-1	-1	REGULATOR.PY
-1	-1	-1.00	-1	-1	-1	-1	-1	-1	-1	

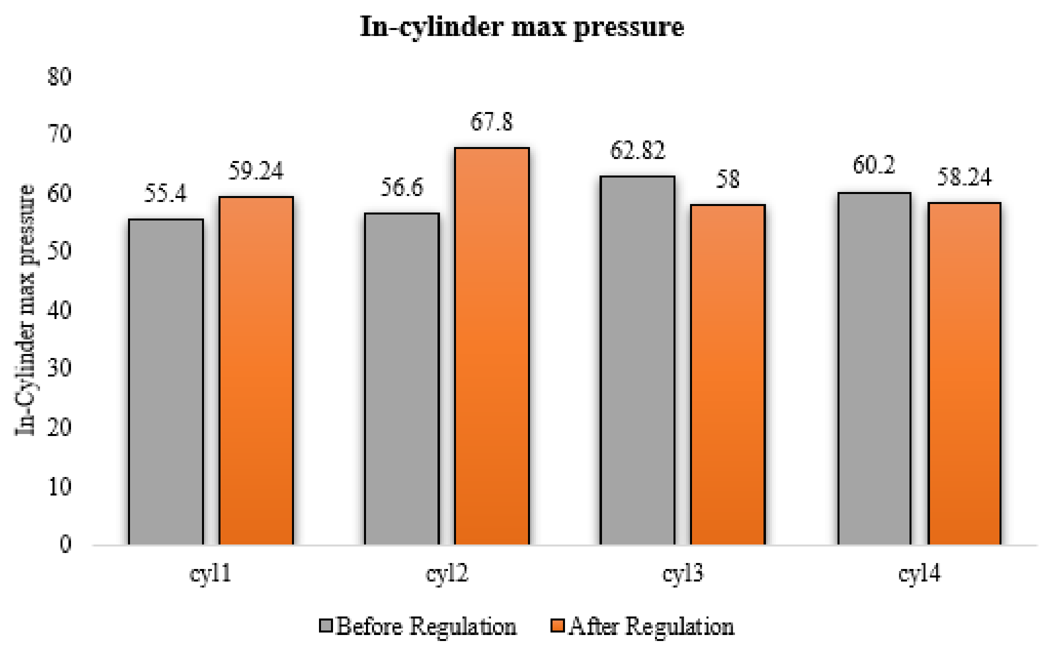
Figure 4.1.5 – *Protocol @Cylinder 1*

- *Influence of lambda regulation on main engine parameters*

Once the regulation is stopped, several important engine parameters are analyzed to observe the influence of lambda variation on each cylinder.

The parameters considered are the maximum in-cylinder pressure, the indicated mean effective pressure, the combustion duration, and the indicated efficiency for each lambda regulation.

Starting with the maximum in-cylinder pressure, figure 4.1.6, it is important to observe the influence of different lambda values on this parameter.



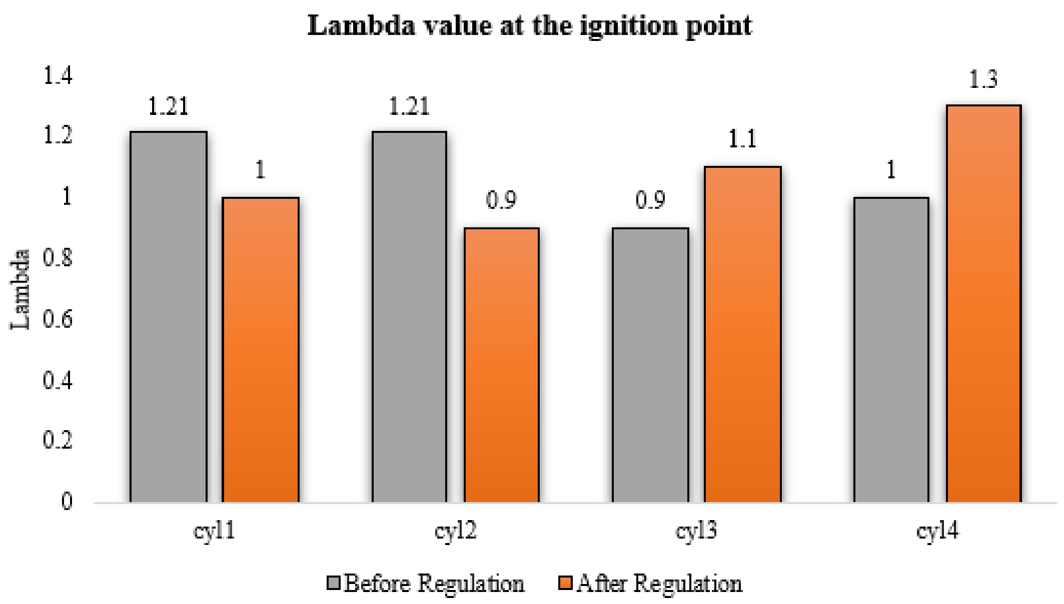


Figure 4.1.6 – *Maximum In-cylinder pressure and lambda before@cycle7 and after@cycle11 regulation*

Starting with cylinder one, the lambda value changes from lean to stoichiometric conditions. As a result, there is a slight increase in the maximum in-cylinder pressure. This is directly related to the amount of fuel burned, as more fuel combustion leads to the generation of more heat, ultimately resulting in an increase of the maximum pressure. This effect is more pronounced in cylinder two where the lambda value changes from 1.21 (lean condition) to 0.9 (slightly rich condition). On the other hand, for cylinders three and four, a decrease in the maximum in-cylinder pressure is observed as a result of a leaner mixture, resulting in less fuel being ignited.

As a result of the change in the lambda value, the indicated work also changes, following the same trend as the in-cylinder pressure.

Fixing the displacement, the indicated work is strictly linked to the in-cylinder pressure as shown the formula 4.1.1:

$$W_i = \oint p dV$$

Equation 4.1.1 - *Indicated Work per cycle*

Where:

- p is the pressure inside the cylinder developed on each cycle;
- dV is the infinitesimal volume.

The trend of the indicated work is shown in Figure 4.1.7, demonstrating its correlation with the in-cylinder pressure.

An enrichment of the mixture leads to an increase of the pressure and so an increase of the indicated work, while on the other hand, a leaner mixture leads to a decrease in the indicated work.

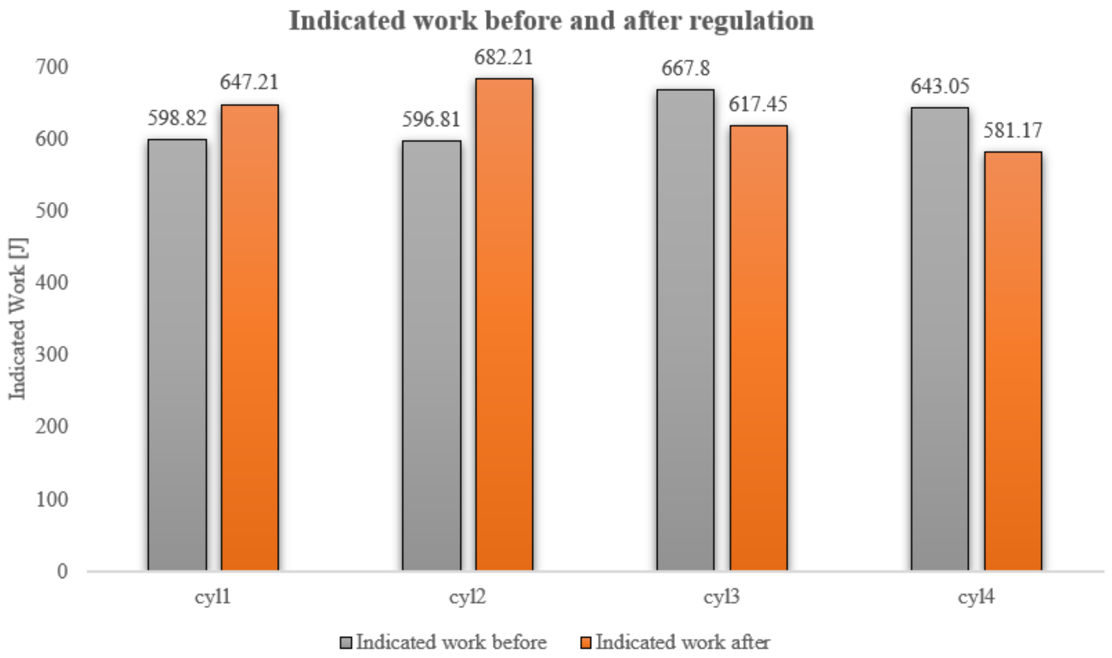


Figure 4.1.7 - *Indicated work before@cycle7 and after@cycle11 regulation*

Usually, it is better to refer to the indicated mean effective pressure (figure 4.8), obtained by dividing the indicated work by the displacement.

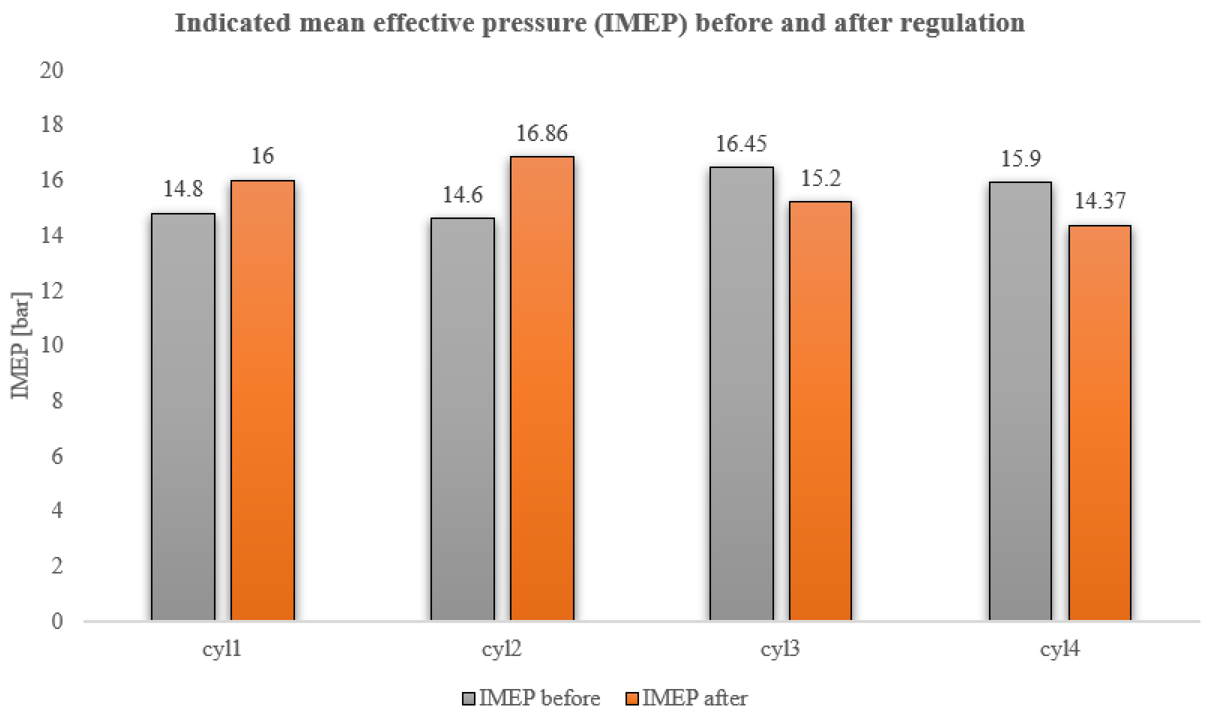


Figure 4.1.8 - *Indicated mean effective pressure (IMEP) before@cycle7 and after@cycle11 regultion*

Finally, a parameter that links the indicated work with the fuel mass injected and so the lambda (keeping constant the air mass) is the indicated efficiency shown in the figure 4.1.9.

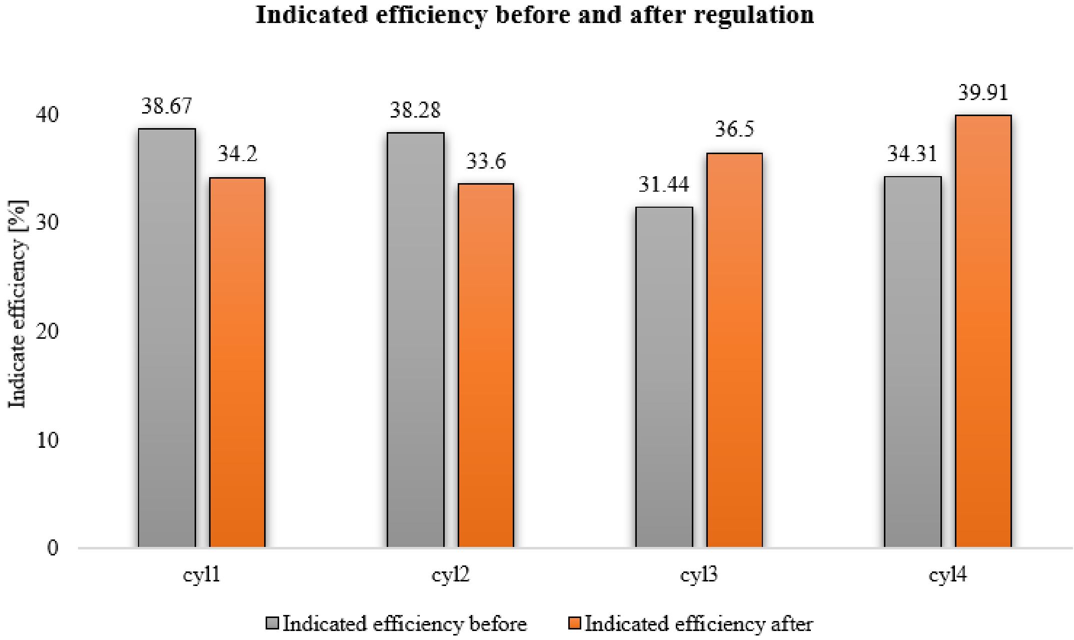


Figure 4.1.9 - *Indicated efficiency before @cycle7 and after@cycle11 regulation*

The equation 4.1.2 demonstrates the relationship between pressure, fuel injection, and indicated efficiency.

$$\eta_i = \frac{W_i}{m_f \cdot LHV}$$

Equation 4.1.2 - *Indicated efficiency*

Where:

- W_i is the indicated work;
- m_f is the mass of fuel injected for each cycle;
- LHV is the lower heating value of the fuel kept in consideration.

The indicated efficiency, equation 4.1.2, is closely correlated to the indicated work, the amount of fuel injected per cycle, and the lower heating value of the fuel (LHV). Since the LHV remains constant due to the use of the same fuel, it is important to understand how the fuel injection affects the indicated efficiency.

When the mixture is enriched, it leads to an increase in the indicated work. However, this also requires a larger amount of gasoline to be injected, resulting in a higher denominator in equation 4.1.2.

On the other hand, regulating the fuel injection in order to leaner the mixture can lead to a more efficient engine, as depicted in figure 4.9. Most gasoline engines are operated with a stochiometric mixture to optimize the efficiency of after-treatment catalysts and reduce the emission of NO_x. For this reason, the cylinder one, where the regulation achieves a stoichiometric mixture, will be used for comparison between single and multiple injections.

4.1.2 Multiple Injections

The algorithm also incorporates a strategy for regulating multiple injections as stated in chapter 3, which was developed and validated using the same engine as before at the same operating point. In the simulation, at cycle twelve, the injection law was switched from single to multiple injections, demonstrating the versatility of the regulator tool. Therefore a multiple injection strategy was configured starting with all cylinders in rich conditions. The target imposed by the user was set to stoichiometric value and in this case QuickReg tool is expected to reduce the fuel injected per cycle to achieve conditions.

After allowing a few cycles without regulation for the simulation to stabilize with the multiple injections, the regulator can be activated to achieve the user-imposed lambda.

To adjust multiple injections, certain input variables and the input controller need to be modified. Table 4.1.7 displays the new input variables.

Variables	Target
Num_inj	1
Lambda_split_cyl	1
Pos_inj	GDI
Type_injection	MULTI
Cycle_L_at_IP_Change	1

Table 4.1.7 - *Boolean variables for a multi injection*

Instead, the table 4.1.8 shows the injection characteristics, and in this case, the lambda target has been uniformly set for all cylinders, while the injector characteristics remain the same as in the single injection case.

Variables	Target	Range	Unit	Cyl.
L_at_ip	1	0.02	-	01
L_at_ip	1	0.02	-	02
L_at_ip	1	0.02	-	03
L_at_ip	1	0.02	-	04
Injectors Characteristics				
Outward opening	15	cm^3/s		
Rampen opening	0.15	ms		
Rampen closing	0.15	ms		
Korrector factor	1.4142	-		

Table 4.1.8 - *λ target and injector characteristics*

In this case, three injections have been selected as shown in the figure 4.1.10. They have the same split ratio of mass but differ in timing.

The first two injections are performed during the intake phase to achieve better mixture homogeneity. The third injection is carried out during the first part of the compression stroke this can have some pro and contro in the engine combustion process.

The drawback is that the mixture may not have enough time to vaporize, resulting in an increased quantity of unburned fuel and higher levels of unburned hydrocarbons. As pro, by adding multiple injections, it can help to improve the vaporization of the mixture, as the additional injections contribute to cooling the mixture surrounding the fuel droplets. Furthermore, the reduction in fuel inertia minimizes fuel impingement, leading to improved combustion efficiency.

Therefore, injecting the fuel during the compression phase provides a more favorable environment for vaporization. Additionally, the higher engine speed ensures better fuel homogenization, resulting in improved combustion performance.

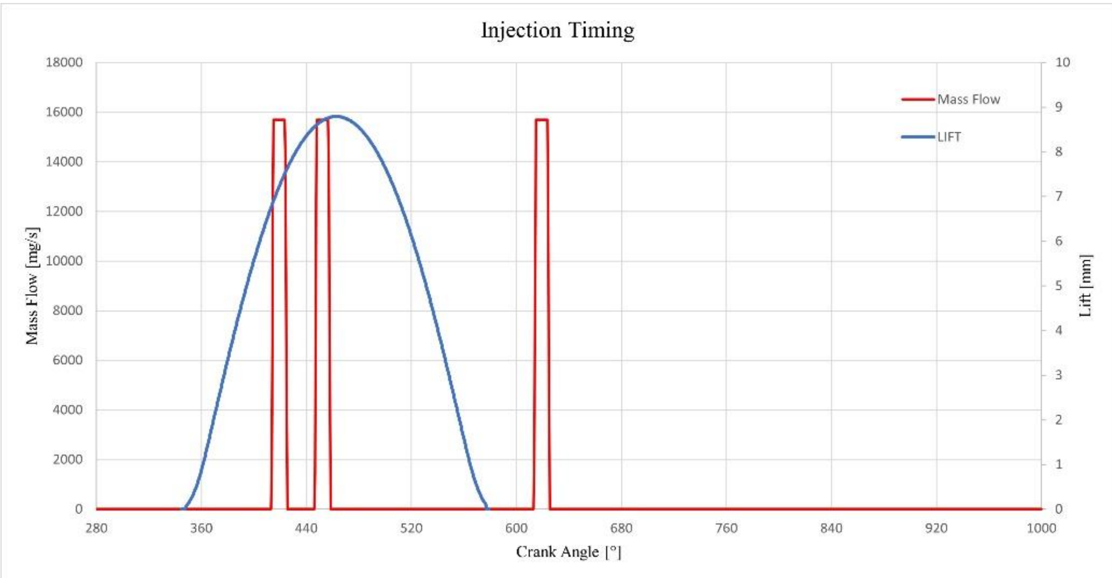


Figure 4.1.10 – *Injection Timing for multiple injections*

The key parameters of the injection law were carefully selected during the pre-processing stage to optimize the combustion performance and take the advantages of multiple injections.

The table 4.1.9 displays the original and regulated injection law for each injection, highlighting the key points. It can be observed that injections 1 and 2 occur during the intake phase, while injection 3 takes place during the compression stroke. The interpretation of each point in this case is the same as in the single injection scenario (refer to figure 4.1.2) and the values are referred to the *QuickSim* convention.

Original					
Injection	**SOI [deg]**	**SOI_full [deg]**	**EOI_full [deg]**	**EOI**	**Fuel injected [mg]**
1	413	415	423.5	425.5	51.59
2	446	448	456.5	458.5	
3	613	615	623.5	625.5	
Reguated					
Injection	**SOI [deg]**	**SOI_full [deg]**	**EOI_full [deg]**	**EOI**	**Fuel injected [mg]**
1	415	417	423.5	425.5	43.19
2	448	450	456.5	458.5	
3	615	617	623.5	625.5	

Table 4.1.9 - *Fundamentals points of injection law before and after the regulation*

The figure 4.1.11 illustrates the injection laws before and after the regulation is applied. It can be observed that after the regulation, the fuel mass injected is reduced, confirming that the regulator applies an impoverishment.

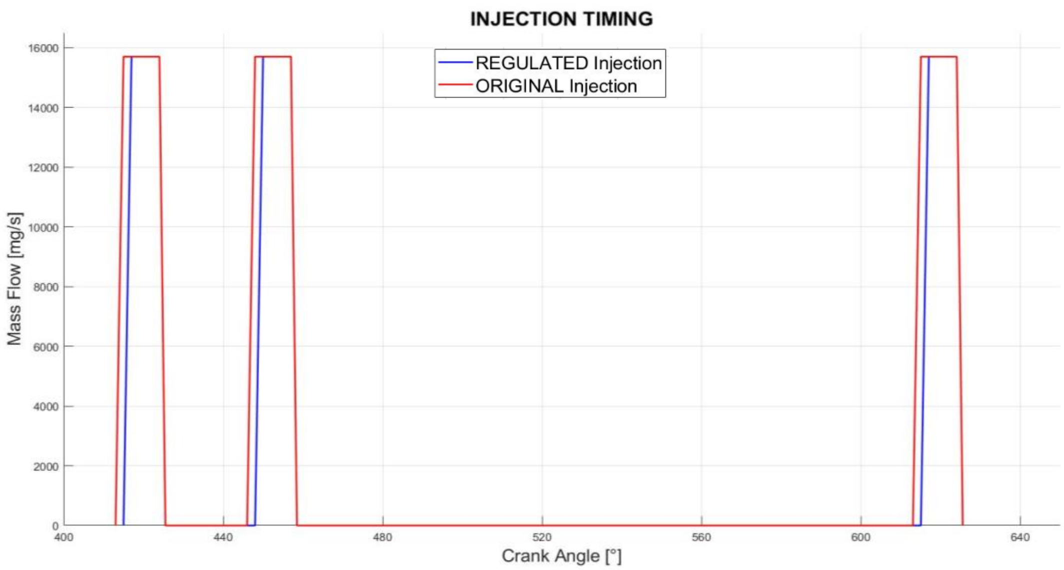


Figure 4.1.11 - *Injection laws before and after the regulation*

The figure 4.1.12 presents the engine results obtained from the post-processing tool. Additionally, a protocol file, figure 4.1.13, provides a detailed record of the operations carried out by *the regulator*, with cylinder one serving as an example. It is evident that the regulator remains inactive from cycle twelve to cycle sixteen, allowing the simulation to reach the convergence with the new injection law.

After the initial four cycles of waiting, the regulation takes place in cycle seventeen, and the outcomes of the regulation can be observed in cycle eighteen.

====== Results ======

VARIABLE	UNIT							
cycle	No	18	17	16	15	14	13	12
Parameter	-	LAM_wait	LAM_change	CONV_wait	CONV_wait	CONV_wait	CONV_wait	CONV_wait
cyl_air_cons	kg/h	29.4347	29.6282	29.6381	29.6173	29.6334	29.6536	29.6536
cyl_m_air_IP	mg	613.6763	617.7044	617.8496	617.4859	617.7863	618.1024	618.1024
cyl_fuel_cons	kg/h	2.0726	2.4773	2.4773	2.4773	2.4773	2.4773	2.4773
inj_m_fuel	mg/Cyc	43.1797	51.6107	51.6106	51.6113	51.611	51.6112	51.6112
cyl_m_fuel_IP	mg	43.1896	51.5977	51.6131	51.6106	51.61	51.2868	51.2868
cyl_power	kW	9.0719	9.6056	9.5628	9.5267	9.5424	9.5624	9.5624
ind_eff_abs	%	35.812	31.7243	31.5831	31.4634	31.5156	31.5815	31.5815
imep	bar	16.8218	17.8114	17.7321	17.6651	17.6943	17.7313	17.7313
cyl_p_max	bar	69.6146	72.5413	72.9214	72.1719	72.1982	72.2559	72.2559
cyc_iit09m_max	%	0.87	3.31	3.04	2.48	2.9	2.68	2.68
cyl_egr_IVC	%	0.043	0.0425	0.0428	0.043	0.0428	0.043	0.043
cyl_max_wbf	%	0.99	0.99	0.99	0.99	0.99	0.99	0.99
cyl_10_wbf	deg	726.7431	727.1837	727.0427	727.1295	727.1901	727.1192	727.1192
cyl_50_wbf	deg	737.2405	737.5653	737.3755	737.456	737.5135	737.4533	737.4533
cyl_90_wbf	deg	746.3116	746.0811	745.7526	746.0596	746.107	746.0939	746.0939
cyl_10_90_wbf	deg	19.5685	18.8974	18.7099	18.9301	18.9169	18.9747	18.9747
cyl_lambda_IP	-	0.9943	0.8378	0.8377	0.8373	0.8377	0.8434	0.8434
prec_lambda_IP	-	-1000000.0	-1000000.0	-1000000.0	-1000000.0	-1000000.0	-1000000.0	-1000000.0
sp_lambda_IP	-	0.9657	0.8726	0.8933	0.9074	0.8894	0.8782	0.8782
prec_egr_IP	%	-100000000.0	-100000000.0	-100000000.0	-100000000.0	-100000000.0	-100000000.0	-100000000.0
sp_egr_IP	kg/kg	4.7023	4.6093	4.6057	4.5877	4.6076	4.6724	4.6724
ip	deg	714.0	714.0	714.0	714.0	714.0	714.0	714.0
sp_r_max	mm	2.0	2.0	2.0	2.0	2.0	2.0	2.0
sp_dca_max	deg	4.0	4.0	4.0	4.0	4.0	4.0	4.0
cyc_A_Weller	-	3.9	3.9	3.9	3.9	3.9	3.9	3.9
imb_p_01	bar	1.3771	1.3771	1.3771	1.3771	1.3776	1.3777	1.3777
imb_T_01	K	303.8805	303.8804	303.8802	303.8802	303.9863	303.9868	303.9868
emb_p_01	bar	1.2709	1.2717	1.2706	1.2717	1.27	1.2698	1.2698

Figure 4.1.12 - *Tool Post Multiple injections*

Index	Cylinder	Crank_Angle	Cyc	Element_category	Element_sub	Execute	Parameter	New_mf	Edited_File	Command
1112213503	1	999.75	18	0	1	1	WAIT_LAMBDA	0	No_Edited_File	REGULATOR.py
1111035357	1	713.75	17	4	1	1	L_AT_IP_REGULATION	43.17	qs-injection-fu_01	REGULATOR.py
1110083025	1	999.75	16	0	1	1	Waiting_initial_convergence	-1	No_Edited_File	REGULATOR.PY
1109012830	1	999.75	15	0	1	1	Waiting_initial_convergence	-1	No_Edited_File	REGULATOR.PY
1029142514	1	999.75	14	0	1	1	Waiting_initial_convergence	-1	No_Edited_File	REGULATOR.PY
1025040249	1	999.50	13	0	1	1	Waiting_initial_convergence	-1	No_Edited_File	REGULATOR.PY
1012144504	1	999.50	12	0	1	1	Waiting_initial_convergence	-1	No_Edited_File	REGULATOR.PY

Figure 4.1.13 - *Protocol Multiple injections*

After the conclusion of the regulation process, a comparison between single and multiple injections was conducted to highlight the key improvement in the engine parameters.

4.1.3 Comparison between single and multiple injections

In this subsection, a comparison has been performed to emphasize the primary differences in the engine resulting from the two types of injection laws. The operating point used for this comparison is present in the table 4.1.10.

Engine	**Direct injection spark ignition**
RPM	1600
Load	High load

Table 4.1.10 - *Operating Point used for the comparison between single and multiples injection*

Due to the stable operating point and consistent data across the cylinders, the comparison primarily focuses on cylinder one, as the other cylinders exhibit similar trends for the variables. The cycles considered for analysis are the cycle 11 and cycle 18. Figure 4.1.14 displays a section of the tool-post output, presenting the main variables for the comparison.

```
====== Results ======
VARIABLE          UNIT

cycle             No           18              11
Injection_law     -            Multiple        Single

cyl_air_cons      kg/h         29.4347         29.7417
cyl_m_air_IP      mg           613.6763        619.97

cyl_fuel_cons     kg/h         2.0726          2.06445
inj_m_fuel        mg/Cyc       43.1797         43.0094
cyl_m_fuel_IP     mg           43.1896         43.021

cyl_power         kW           9.0719          8.6295
ind_eff_abs       %            35.812          34.2003
imep              bar          16.8218         16.0014
cyl_p_max         bar          69.6146         59.2402
cyl_50_wbf        deg          737.2405        739.7037
cyl_10_90_wbf     deg          19.5685         24.3448

cyl_lambda_IP     -            0.9943          1.0085
sp_lambda_IP      -            0.9657          0.8458
sp_egr_IP         kg/kg        4.7023          4.6397

ip                deg          714.0           714.0
sp_r_max          mm           2.0             2.0
sp_dca_max        deg          4.0             4.0
cyc_A_Weller      -            3.9             3.9
```

Figure 4.1.14 – *Output of cylinder 1: multi (cycle 18) vs single (cycle 11) injection*

Figure 4.1.14 illustrates the output data of cylinder 1 for both single and multiple injections. Multiple injection proves to be effective in reducing the combustion duration significantly, thereby enhancing engine performance. As mentioned earlier, a shorter combustion duration results in higher combustion efficiency when employing multiple injection. Additionally, it is noteworthy that the lambda value at the spark (*sp_lambda_IP*) approaches stoichiometric levels with multiple injection, indicating better fuel mixing compared to the single injection technique.

The first variable to be analyzed is the in-cylinder fuel mass, represented as *'inj_m_fuel'* in Figure 4.1.14.

As depicted in Figure 4.1.15, in the case of single injection, the entire fuel mass is injected at once, as indicated by the light blue line. The injection law exhibits a single step, and the fuel mass remains constant until combustion occurs. On the other hand, in the case of multiple injections, there are three stages in the injection profile before it stabilizes. By ensuring that the lambda value remains consistent, as a condition for comparing the two injection profiles, the figure demonstrates that the fuel mass inside the cylinder remains the same for both cases.

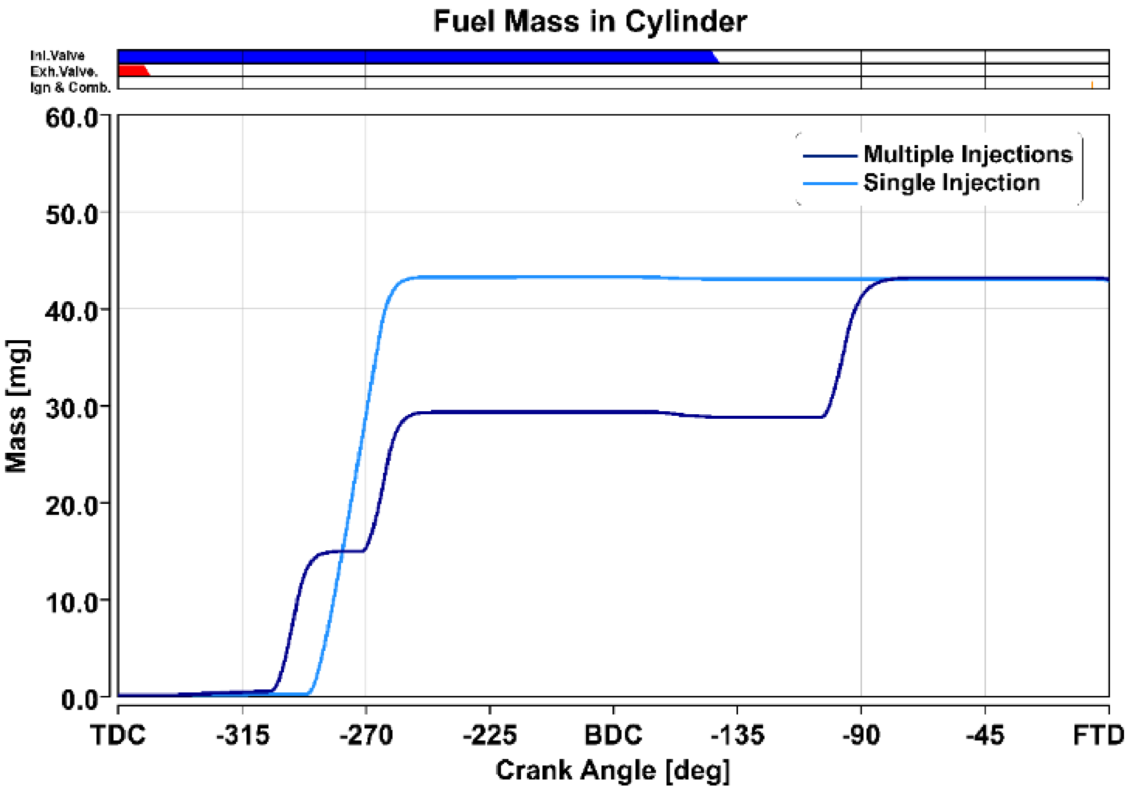


Figure 4.1.15 - *Fuel mass in-cylinder @1600 rpm_high load* - ***Light blue***: *single injection (cyc. 11)* - ***Dark blue***: *multiple injection (cyc. 18)*

As a result, figure 4.1.16 illustrates a variable closely associated with the mass of fuel injected into the cylinder, namely air-to-fuel ratio. The trend shows two different behaviors for both single and multiple injections. In the case of single injection, the air-to-fuel ratio begins to shift towards a rich zone once the fuel is vaporized. When the valves are closed, the air-to-fuel value stabilizes, typically after a few crank angle degrees from the bottom dead center.

This cannot be observed in the case of multiple injections. After the initial injection, the air-to-fuel ratio returns to the lean zone as the air continues to be drawn into the cylinder. Then, during the second injection, while the intake valves are still open, the trend becomes similar to the single injection but with a shift towards the lean zone, indicating that there is more air than fuel in the cylinder. Finally, when the intake valves close and the compression phase begins, the third injection occurs, and the air-to-fuel ratio converges to the stoichiometric value.

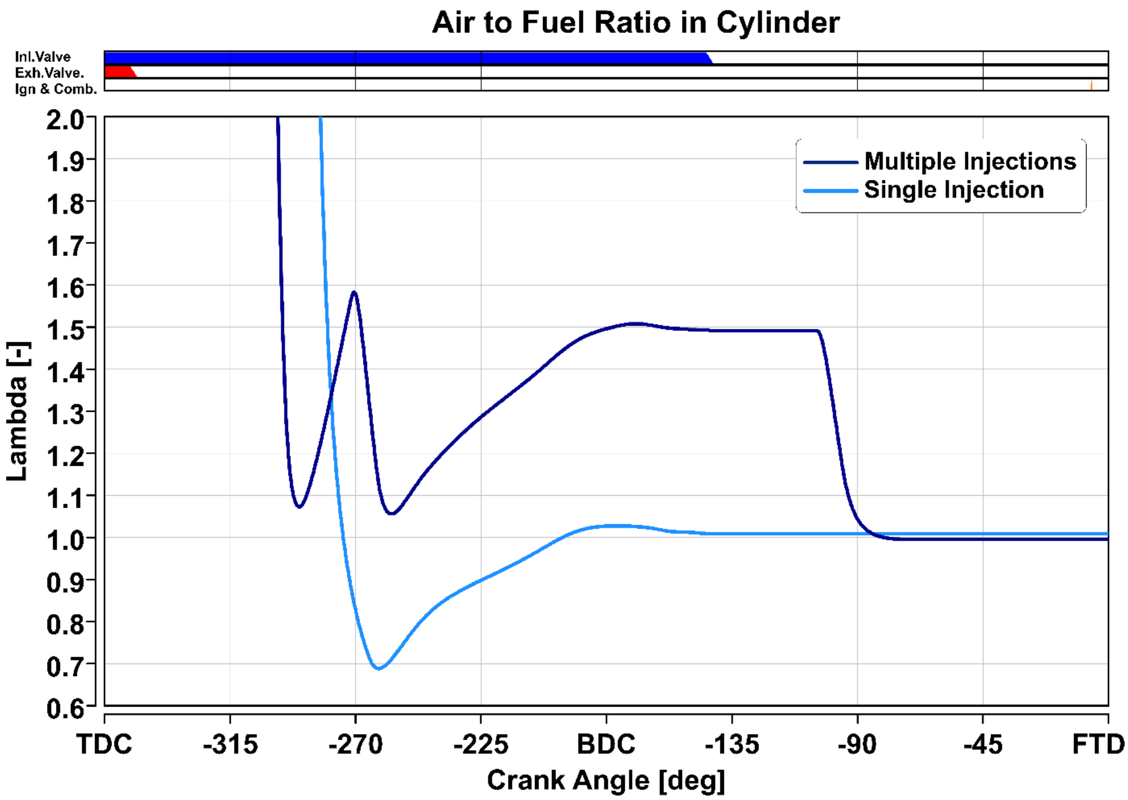


Figure 4.1.16 - *Air-to-fuel ratio in-cylinder @1600 rpm_high load* - ***Light blue****: single injection (cyc. 11)* - ***Dark blue****: multiple injection (cyc. 18)*

In Figure 4.1.17, a comparison is presented between the turbulence kinetic energy (TKE) of the flame-front for both multiple and single injections.

The turbulent kinetic energy is generally calculated using the formula 4.1.3 and is a crucial parameter for ensuring better combustion efficiency:

$$TKE = \frac{1}{2}\left[\overline{(u')^2} + \overline{(v')^2} + \overline{(w')^2}\right]$$

Equation 4.1.3 - *Tubulent kinetic energy*

Where:

- $\overline{(u')^2}$ is the standard deviation of the velocity along x;
- $\overline{(v')^2}$ is the standard deviation of the velocity along y;
- $\overline{(w')^2}$ is the standard deviation of the velocity along z.

As depicted in the figure 4.1.17, the TKE value for the multiple injection strategy is higher than that for the single injection. Having a higher turbulence kinetic energy results in a more efficient combustion phase and a shorter combustion duration.

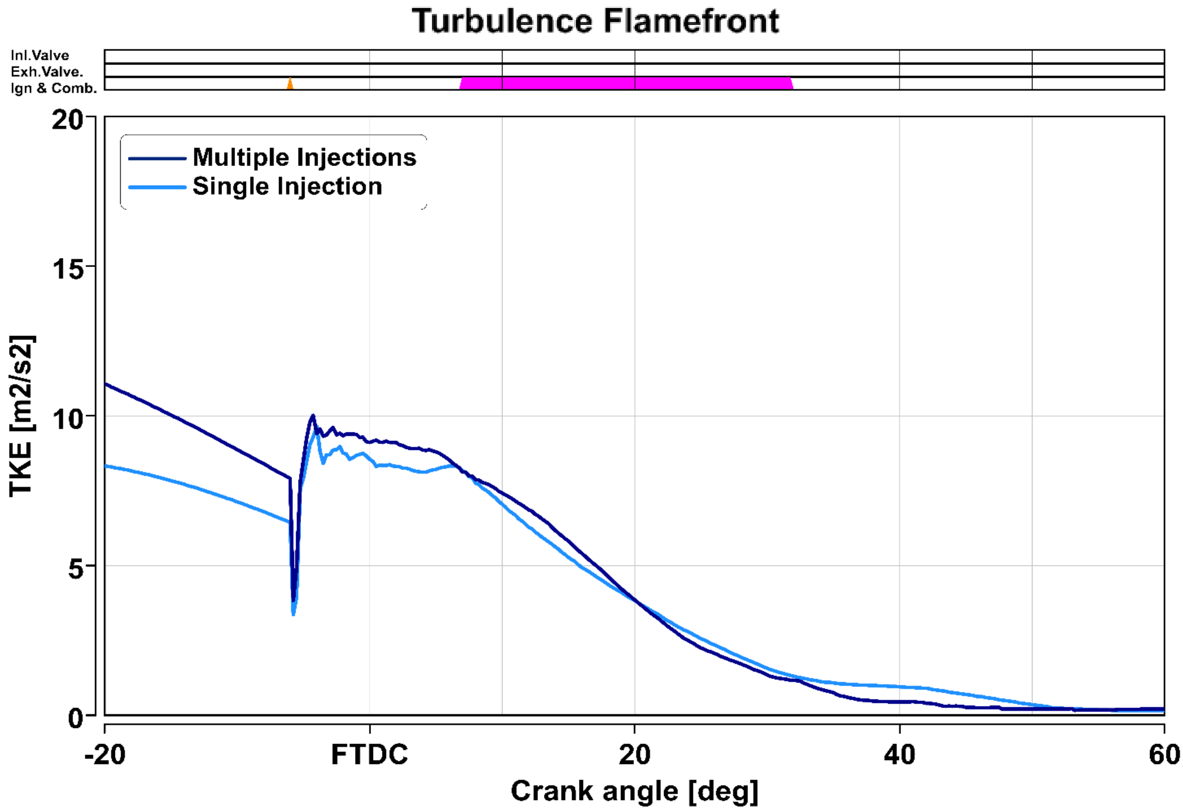


Figure 4.1.17 - *Turbulence Flamefront @1600 rpm_high load* - ***Light blue****: single injection (cyc. 11)* - ***Dark blue****: multiple injection (cyc. 18)*

Analyzing the flame speed and the burned mass fraction, as shown in figure 4.1.18, it can be observed that the flame starts to propagate at the same time during the engine cycle, considering the same ignition point. However, due to the higher turbulent kinetic energy inside the combustion chamber in the case of multiple injections, the flame front moves with greater strenght. This results in a higher flame speed compared to the single injection case, leading to a faster combustion duration. As depicted in the figure 4.1.18, the curve of the burned mass fraction is shifted towards the left for multiple injections, indicating that the MFB90 (the point at which 90% of the fuel has been burned) is reached earlier compared to the single injection case.

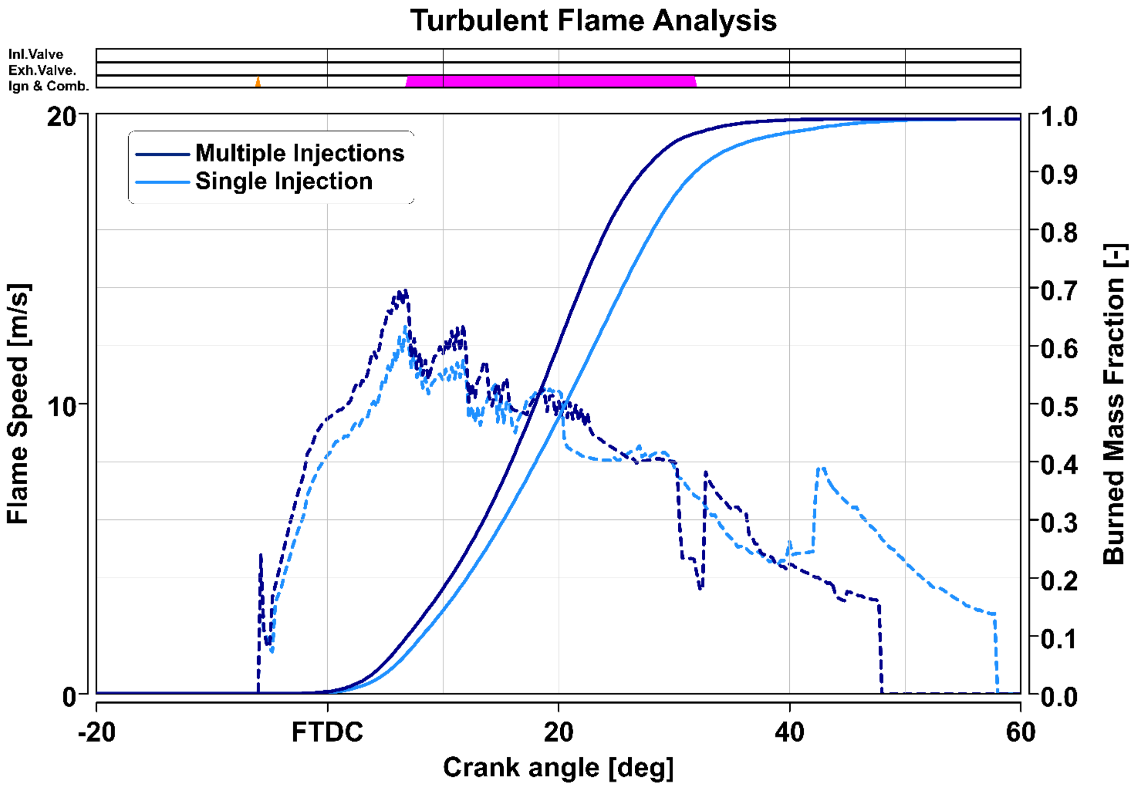


Figure 4.1.18 - *Turbulent flame analysis @1600 rpm_high load* - ***Light blue***: *single injection (cyc. 11)* - ***Dark blue***: *multiple injection (cyc. 18)*

As a result of higher turbulence and a faster combustion process, the in-cylinder pressure within the combustion chamber increases, as shown in figure 4.1.19. In-cylinder pressure is closely related to the combustion process, and a more efficient combustion leads to an increase in pressure within the chamber. Performing a multiple injection strategy, a higher maximum in-cylinder pressure is reached.

However, it is important to consider the potential drawbacks of increased pressure, particularly the sensitivity to knock. High pressure inside the cylinder is one of the factors that can trigger knock phenomena, which can damage the engine. Therefore, following the lambda regulation in the multiple injection strategy, the regulator will evaluate the engine's proximity to knock and make necessary adjustments to the ignition point in order to prevent this issue. This aspect will be further explored in the thesis of my colleague.

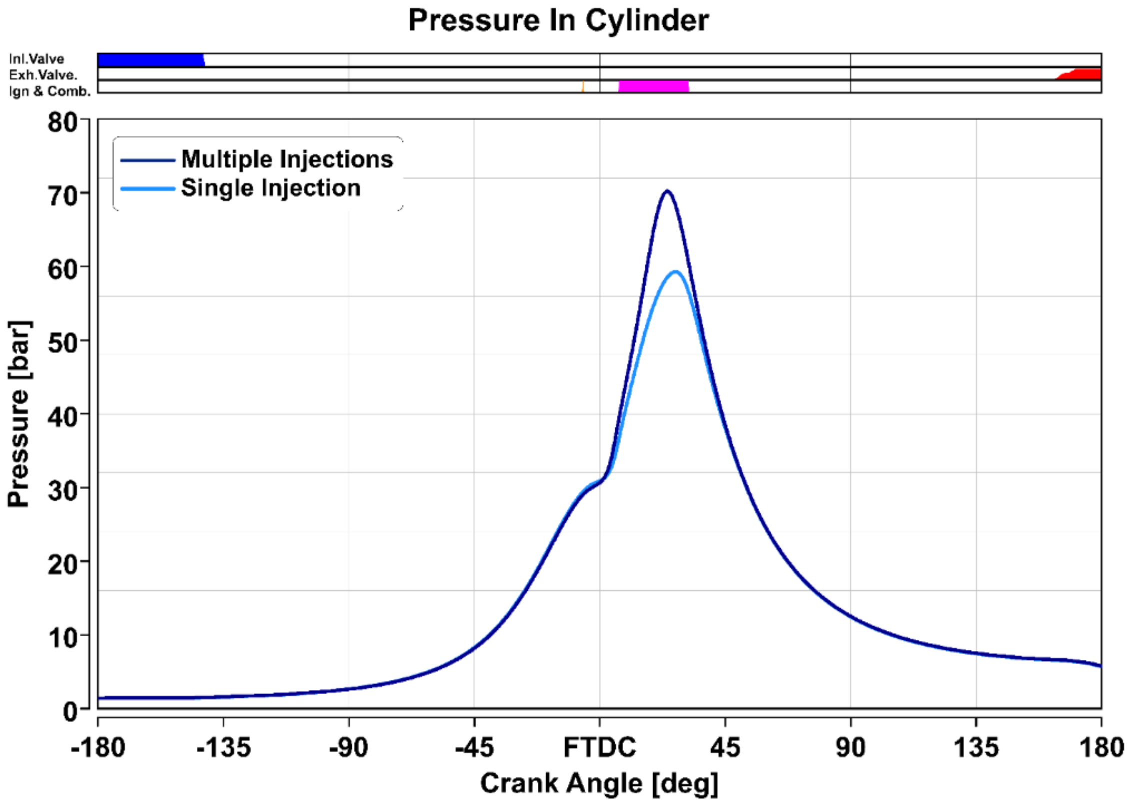


Figure 4.1.19 - *In-cylinder pressure @1600 rpm_high load* - ***Light blue****: single injection (cyc. 11)* - ***Dark blue****: multiple injection (cyc. 18)*

Increasing in-cylinder pressure leads to an increase in temperature, as these two parameters are closely related according to equation 4.1.4:

$$\frac{dT}{d\theta} = \frac{p}{mR}\frac{dV}{d\theta} + \frac{V}{mR}\frac{dp}{d\theta}$$

Equation 4.1.4 – *Temperature equation*

Where:

- dT is the infinitesimal temperature;
- $d\theta$ is the infinitesimal crank angle;
- p is the pressure inside the cylinder;
- dp is the infinitesimal pressure inside the cylinder;
- m is the mixuture mass in the cylinder;
- V is the volume of the combustion chamber;
- dV is the infinitesimal volume of the cylinder;
- R is the elastic osntant of the gasses.

The equation is derived from the state equation of gases. In this case, it is specifically related to the pressure within the cylinder. Consequently, an increase in cylinder pressure results in a corresponding increase in temperature within the cylinder.

As shown in figure 4.1.20, there is a corresponding increase in temperature inside the cylinder with an increase in pressure. This relationship is due to the close connection between these two parameters and in the case of multiple injection, the in-cylinder temperature is higher compared to the single injection.

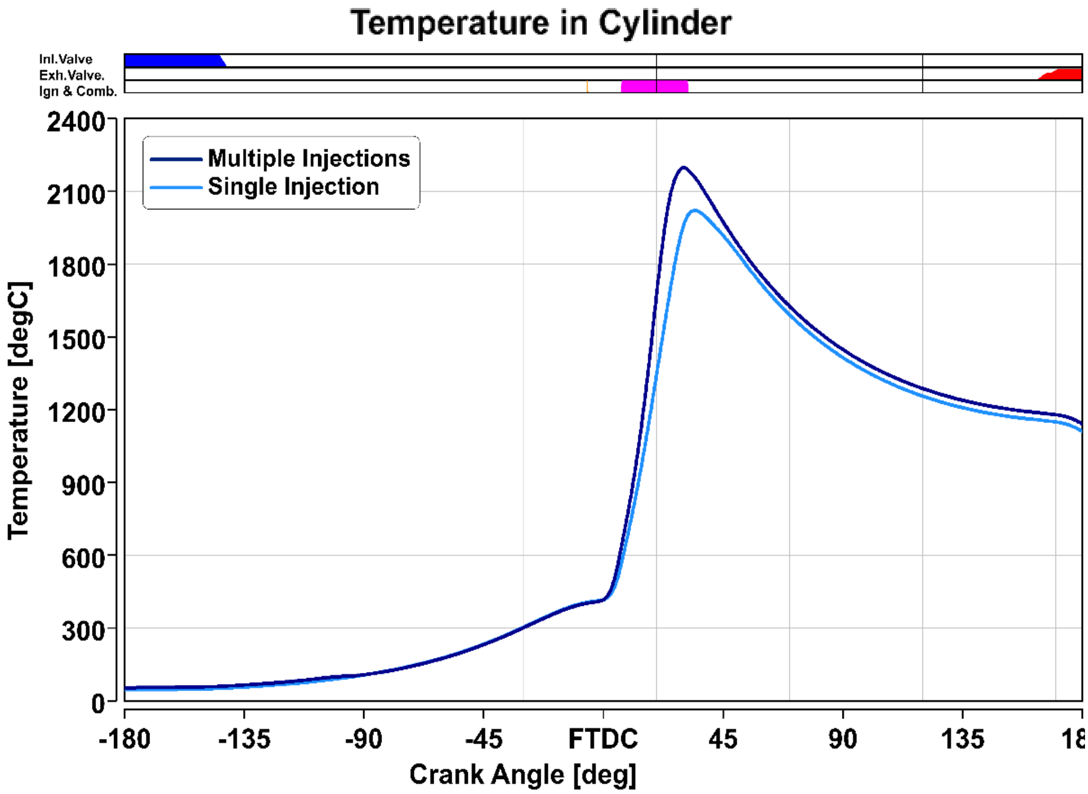


Figure 4.1.20 - *Temperature in-cylinder @1600 rpm_high load* - ***Light blue****: single injection (cyc. 11)* - ***Dark blue****: multiple injection (cyc. 18)*

Figure 4.1.21 depicts the heat release and heat release rate for both single and multiple injection laws. It is evident that multiple injections result in higher values for both parameters. This can be attributed to the improved stability and faster combustion achieved with multiple injections. In addition to that, the increase in pressure and temperature results in an increase of the heat release rate and heat release.

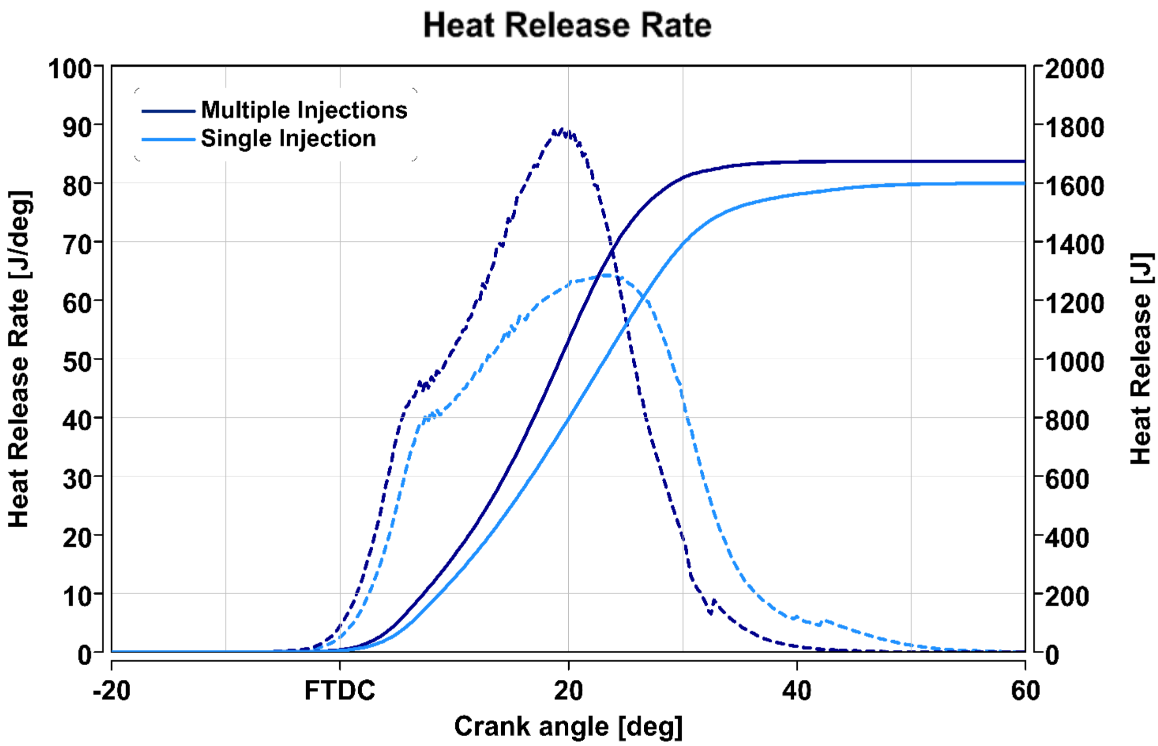


Figure 4.1.21 - *Heat Release Rate @1600 rpm_high load* - ***Light blue****: single injection (cyc. 11)* - ***Dark blue****: multiple injection (cyc. 18)*

In summary, the comparison highlights how multiple injections can improve combustion and, consequently, enhance engine efficiency under the same operating conditions compared to single injections. However, implementing multiple injection strategies can be challenging as it requires an ECU and injectors that operate with high speed and precision, in addition, these technology are expensive. Despite these challenges, the advantages offered by multiple injections outweigh the drawbacks, prompting companies to invest in this technology to achieve greener engines and meet the emission targets set by the European Union.

4.2 Multi-cylinder high-performance engine

The lambda regulation strategy has also been validated using a high-performance engine commonly used in racing competitions. The simulation was conducted at a high RPM with a high compression ratio (CR). This engine is equipped with a port fuel injection system, where the fuel is injected into the intake port rather than directly into the combustion chamber. Further details about this type of injection can be found in the first chapter.

The engine characteristics are outlined in the table 4.2.1 as follows:

Engine	High-performance engine
Rpm	High rpm
Power	High power
Type of injection	Port Fuel Injection (PFI)
Number of cylinders	4
Compression Ratio	High CR

Table 4.2.1 – *Engine Characteristics*

The validation process for the high-performance engine involved beginning an injection with a timing that occupied 85% of the cycle duration, starting from rich conditions. One of the challenges encountered with this engine was maintaining a constant air mass at the ignition point, resulting in the need for multiple regulation cycles.

To begin the validation, the input controller had to be configured. Table 4.2.2 shows the variables utilized in the input controller, while the injector characteristics and the specific operating point of the engine during the simulation are not depicted.

Variables	Target
Num_inj	1
Lambda_split_cyl	1
Pos_inj	PFI
Type_injection	SINGLE
Cycle_L_at_IP_Change	3

Table 4.2.2 – *Boolean variables for a PFI engine*

First the regulation was done with the same injection strategy as for gasoline direct injection not considering the fuel losses and the time needed by the fuel to reach the combustion chamber. The lambda target value was set equal to 1 with a range of ± 0.02. Subsequently, a modification has been implemented to decrease the time required to reach the target set by the user.

It is important to note that in this case, the number of waiting cycle between each lambda regulation and the subsequent one is longer compared to the GDI simulation discussed earlier. This increased in the number of the waiting cycle is necessary because, in a port fuel injection engine, especially at high RPM, it becomes more challenging to achieve a one-shot regulation of lambda. This difficulty arises from frequent fluctuations in air mass and the possibility of a small percentage of fuel getting trapped inside the intake manifold as described before.

Therefore, a greater number of waiting cycles is necessary between each regulation to ensure accuracy and stability.

As shown in figure 4.2.1, the first five cycles are dedicated to ensuring the convergence of the simulation before the control and regulation of lambda can start. The first regulation of lambda is performed in cycle six, and its effect can be observed in the following cycle. After the initial regulation, not all cylinders are within the range imposed by the user, so additional regulations must be performed independently for each cylinder. Specifically, the regulator adjustes the lambda values for cylinders one, two, and four, while cylinder three is skipped because it is already in the range of lambda imposed by the user.

As mentioned earlier, the regulation is not performed in a single step due to the fluctuations in air mass and the omission of fuel losses in this strategy. Therefore, three cycles of waiting are considered to achieve convergence. After these cycles, a new evaluation and regulation of lambda is conducted in cycle ten.

In cycle fourteen, another check is performed. In the case of cylinders one and three, the lambda values reach the target specified by the user in the input controller. However, for the remaining cylinders, the lambda values are outside the acceptable range, necessitating additional checks and adjustments to bring them within the desired range. This process can be time-consuming before all cylinders are regulated. To address this issue, an improvement in the injection strategy is implemented, as described in section 3.5.3.

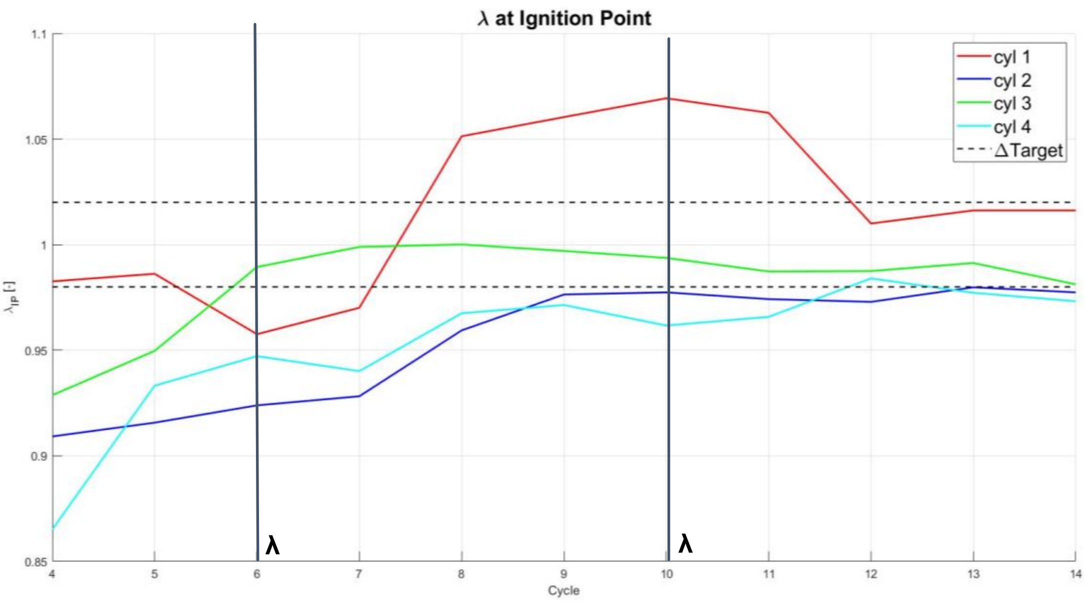


Figure 4.2.1 - *Lambda at ignition point before the improvement of the injection strategy*

All the action made by the regulator are displayed in figure 4.2.2:

Index	Cylinder	Crank_Angle	Cyc	Element_category	Element_sub	Execute	Parameter	New_mf	Edited_File	Command
***	1	IP	14	4	1	0	L_AT_IP_IN_RANGE	-	No_Edited_File	REGULATOR.PY
***	1	EOC	13	0	1	0	WAITING_BEFORE_NEW_LAMBDA_REGULATION	-	No_Edited_File	REGULATOR.PY
***	1	EOC	12	0	1	0	WAITING_BEFORE_NEW_LAMBDA_REGULATION	-	No_Edited_File	REGULATOR.PY
***	1	EOC	11	0	1	0	WAITING_BEFORE_NEW_LAMBDA_REGULATION	-	No_Edited_File	REGULATOR.PY
***	1	IP	10	4	1	0	L_AT_IP_REGULATION	-4.4%ORIG.	qs-injection-fu_01.inp	REGULATOR.PY
***	1	EOC	9	0	1	0	WAITING_BEFORE_NEW_LAMBDA_REGULATION	-	No_Edited_File	REGULATOR.PY
***	1	EOC	8	0	1	0	WAITING_BEFORE_NEW_LAMBDA_REGULATION	-	No_Edited_File	REGULATOR.PY
***	1	EOC	7	0	1	0	WAITING_BEFORE_NEW_LAMBDA_REGULATION	-	No_Edited_File	REGULATOR.PY
***	1	IP	6	4	1	0	L_AT_IP_REGULATION	-10%ORIG.	qs-injection-fu_01.inp	REGULATOR.PY
***	1	EOC	5	0	1	1	Waiting_initial_convergence	ORIGINAL	No_Edited_File	REGULATOR.PY
***	1	EOC	4	0	1	1	Waiting_initial_convergence	"	No_Edited_File	REGULATOR.PY
***	1	EOC	3	0	1	1	Waiting_initial_convergence	"	No_Edited_File	REGULATOR.PY
***	1	EOC	2	0	1	1	Waiting_initial_convergence	"	No_Edited_File	REGULATOR.PY
0	1	EOC	1	0	1	1	Start_IP	-	-1	REGULATOR.PY
-1	-1	-1.0	-1	-1	-1	-1	-1	-1.0	-1	-1

Figure 4.2.2 – *Protocol*

As stated earlier, the simulation starts with rich conditions and a stoichiometric lambda value is imposed thus a reduction in the fuel mass injected must be evaluated by the regulator.

Figure 4.2.3 depicts the injection law before and after the actions of the regulator tool for *'cylinder_01'*:

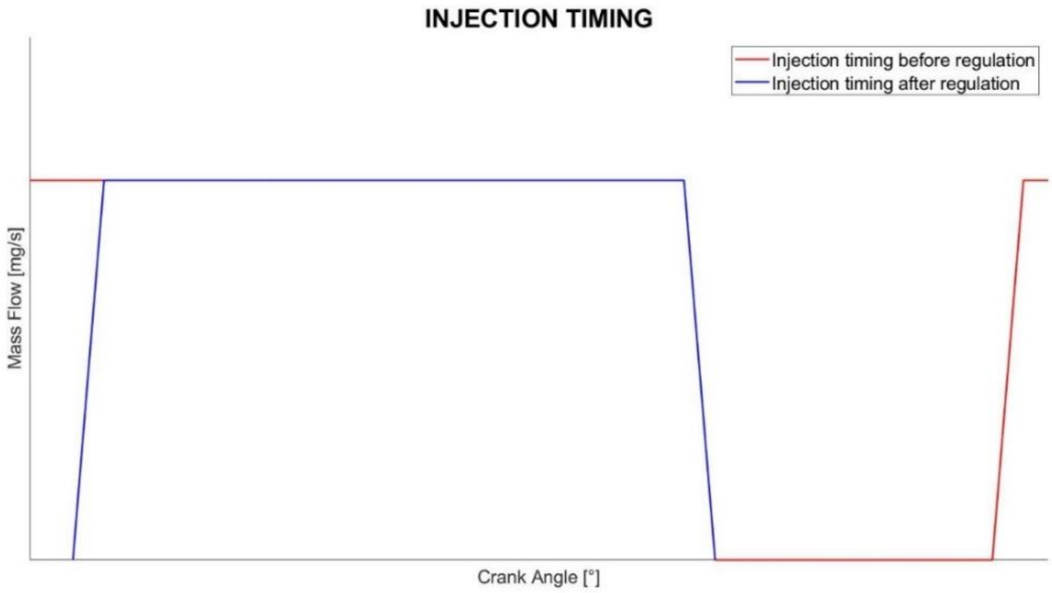


Figure 4.2.3 – *Injection over cycle end*

As noticeable from the figure, the fuel injected after the regulation of the injection timing is reduced compared to the fuel injected before, resulting in a leaner mixture.

Consequently, the proposed improvement was validated using the same engine, but with a change in the user-imposed lambda target. In this case, a target of 0.95 was chosen, with a more stringent range of variation of ± 0.01, thus the regulation becomes more challenging compared to previous tests.

As depicted in figure 4.2.4, the regulation of lambda occurs at cycle twenty-three, while the preceding cycles were dedicated to controlling other parameters. When cycle twenty-two is reached, the regulator tool starts the regulation process. The effects of the regulation can be observed in cycle twenty-five, saving two cycles of waiting. In this scenario, cylinder four is already within the specified range, thus the regulation is performed for the remaining cylinders. It can be observed that, in this case, the regulation occurs in a single step, except for cylinder two, which deviates from the range by a value that can be considered negligible.

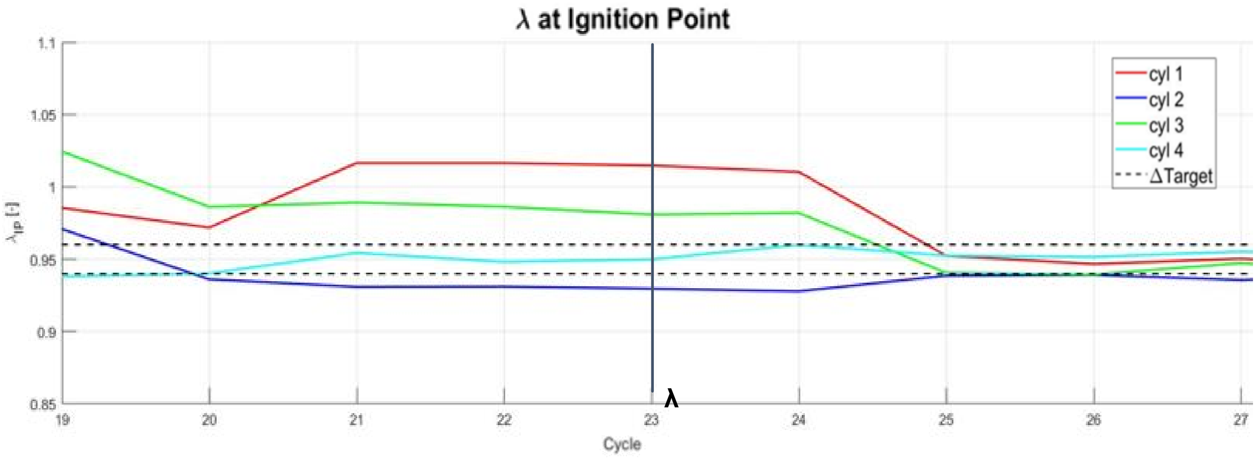


Figure 4.2.4 - *Lambda at ignition point with the new injection strategy*

In the figure 4.2.5 is shown the protocol file from the cycle twentytree to twentysix.

Index	Cylinder	Crank_Angle	Cyc	Element_category	Element_sub	Execute	Parameter	New_mf	Edited_File	Command
***	1	EOC	26	0	1	0	L_AT_IP_IN_RANGE	-	No_Edited_File	REGULATOR.PY
***	1	EOC	25	0	1	0	WAITING_BEFORE_NEW_LAMBDA_REGULATION	-	No_Edited_File	REGULATOR.PY
***	1	EOC	24	0	1	0	WAITING_BEFORE_NEW_LAMBDA_REGULATION	-	No_Edited_File	REGULATOR.PY
***	1	IP	23	4	1	0	L_AT_IP_REGULATION	-5.3%ORIG.	qs-injection-fu_01.inp	REGULATOR.PY

Figure 4.2.5 - *Protocol*

Thanks to the optimization achieved through the improved injection strategy, there has been a significant reduction in the number of waiting cycles during the regulation process. This reduction not only improves the accuracy and efficiency of the lambda regulation but also translates into a substantial reduction in overall simulation time. By minimizing the waiting periods between regulation adjustments, the *QuickReg* package enables faster convergence of the simulation and facilitates quicker attainment of the desired lambda values also for the engine that are tricky to control.

The reduced simulation time is very importance as it allows for more efficient and streamlined engine development processes. With shorter simulation durations, engineers and researchers can iterate through different scenarios and test various parameters more rapidly. This accelerated workflow enhances productivity and enables quicker assessment of engine performance under different operating conditions.

In summary, the improvement in the injection strategy not only enhances the accuracy of lambda regulation but also contributes to significant time savings in simulation, making the QuickReg package a valuable tool for efficient and time-effective engine development.

Chapter 5

5 Conclusions and Outlook

The objective of this study was to enhance the efficiency and automation of 3D-CFD simulations in the QuickSim environment by developing an algorithm for simulation regulation. In the pre-processing phase, establishing accurate boundary conditions is crucial. If these conditions are set incorrectly, it can lead to inaccuracies in the simulations. Additionally, 3D simulations are computationally expensive, necessitating a tool to correct these conditions.

The initial phase of the work involved the offline development of a regulation strategy for key engine parameters. Subsequently, the tool's reliability was tested by applying it to various simulations, demonstrating improvements in simulation management. The flexibility of the code was also assessed by modifying settings during the simulation, such as fuel changes. Currently, the regulation process is manual, resulting in significant time wastage as the user needs to frequently monitor and adjust the simulation to ensure correct values.

The implementation of this straightforward regulation method has yielded promising results, optimizing the performance of the QuickSim software and reducing engine development time. Future steps will focus on refining the regulation methods for the existing parameters and expanding the regulation to other engine parameters.

Additionally, efforts will be made to collect and organize all boundary conditions files, using a matrix code introduced in this work, to create a comprehensive database encompassing various engine configurations (e.g., pre-chamber, innovative fuels, advanced combustion modes).

The obtained results were highly reliable as they were supported by physical feedback, as discussed in Chapter 4. The initial simulation was performed using a multi-cylinder engine with a single injection law to demonstrate the functionality of QuickReg. The regulation of lambda through fuel adjustment proved to be highly precise and adaptable. After the successful regulation, the boundary conditions related to the injection law were modified to incorporate the multiple injection strategy, showcasing the versatility of the tool.

Furthermore, another type of engine with different characteristics and operating points was tested to validate the tool's performance. The tests conducted in various scenarios demonstrated that the tool met the validation requirements, including accuracy, precision, repeatability, and reproducibility. The regulator seamlessly integrated into the simulation, even in restart situations, owing to the traceability of the protocol, enabling it to determine its position in the simulation and accurately read the appropriate inputs.

In the future, there are several potential developments which could enhance the capabilities of the tool even further.

One of the future development is to implement a strategy ables to control the lambda changing the air mass and keeping constant the fuel mass.

The latter approach can be more challenging, as it currently only applies to turbocharged engines where the regulation of air can be achieved by controlling the boost pressure. However, in the case of naturally aspirated engines, where the air mass is regulated by the throttle valve, implementing this type of control becomes challenging.

At present, the control of air in naturally aspirated engines is achieved by imposing a pressure drop between the upstream and downstream of the throttle valve, but this approach is not efficient and results in a significant amount of time waste during the simulation. Therefore, improvements are needed to enhance the control of air for naturally aspirated engines within the regulator.

Currently, research is underway to establish a correlation between the pressure drop across the throttle valve and the percentage of porosity. However, it is challenging to find a linear relationship and identify a strong correlation between the two variables.

The goal is to determine a functional relationship that accurately represents the behavior of the throttle valve and enables effective control of the air mass in naturally aspirated engines. This correlation is crucial for achieving precise lambda regulation and optimizing engine performance.

Although the process may be complex and time-consuming, it is an essential area of research and development to enhance the control strategies for naturally aspirated engines and enable efficient lambda regulation.

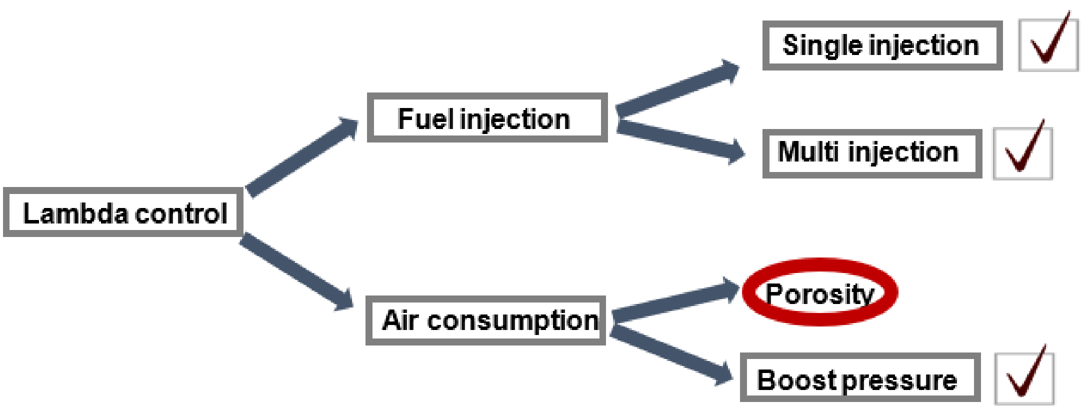


Figure 4.2.1 – *Future developments strategy*

Chapter 6

6 References

[1] *An innovative 3D-CFD-Approach towards virtual development of internal combustion engines,* M. Chiodi, Vieweg+Teubner Research, 2011.

[2] *Virtual Engine Development: 1D- and 3D-CFD up to Full Engine Simulation,* Rodolfo Tromellini, M. Chiodi, M. Bargende et al.

[3] *Internal combustion engines fundamentals*, J.B. Heywood, McGrawHill, 1988.

[4] *Analysis of scavenging air post-oxidation by means of 3D-CFD simulation including reaction mechanism*, Rodolfo Tromellini, J. Przewlocki, F. Cupo, M. Bargende & M. Chiodi.

[5] *Internal combustion engines*, G. Ferrari, Esculapio, 2016.

[6] *An introduction to computational fluid dynamics*, H.K. Versteeg, W. Malalasekera, Pearson Education Limited, 2nd edition, 2007.

[7] *Analysis of Water Injection Strategies to Exploit the Thermodynamic Effects of Water in Gasoline Engines by Means of a 3D-CFD*, Virtual Test Bench A. Vacca, M. Bargende, M. Chiodi, T. Franken et al., SAE Technical Paper 2019-24-0102, 2019.

[8] *1D and Multi-D Modeling Techniques for IC Engine Simulation*, A. Onorati, G. Montenegro, SAE international, 2020.

[9] *Automotive propulsion systems*, notes of prof. F. Bozza, University of Naples Federico II, 2019.

[10] *Robert Bosch GmbH*, [Online] https:// /www.bosch-presse.de, 2013.

[12] *Python*, [Online] https://it.wikipedia.org/wiki/Python.

[13] *Engine emissions control*, notes of prof. F. Millo, Polytechnic of Turin, 2022.

[14] *Propulsion systems for automotive applications*, notes of prof. S. D'ambrosio, Polytechnic of Turin, 2022

[15] *BCG forecasts electrified vehicles to take half of auto market by 2030*, https://www.greencarcongress.com/2017/11/20171103-bcg.html

[16] *Pathways to decarbonization: the European passenger car market in the years 2021-2035*, Peter Mock, Sonsoles Diaz, 2021

[17] *Gasoline Direct Injection* [online] https://www.bosch-mobility.com/en/solutions/powertrain/gasoline/gasoline-direct-injection/

[18] *3D-CFD full engine simulation of post-oxidation by means of secondary air injection*, M. Pipolo, A. Casal Kulzer, M. Chiodi, M. Kumar, Y. Moriyoshi, 2023

[19] *QuickReg – User guide* documentation

[20] *Multiple Injection for Improving Knock, Gaseous and Particulate Matter Emissions in Direct Injection SI Engines*, Taehoon Han1, Ripudaman Singh1, George Lavoie1, Margaret Wooldridge1, 2 and André Boehman1, Department of Mechanical Engineering, University of Michigan

Never give up,

without commitment will never start,

but more importantly, without consistency you'll never finish,

it's not easy.

So, keep working,

keep striving,

fall down 7 times get up 8,

keep moving,

keep growing,

keep learning.

See you at work.

Torino, 2023