Politecnico di Torino

Politecnico di Torino

Signal processing of an internal combustion engine using MATLAB

Propulsion systems and their application to vehicles

Students:

Dennis Bettinsoli (s357720)
Matteo Canestrini (s349238)
Antonio Bisaccia (s359955)
Alessandro De Blasi (s352294)

Academic Year 2025–2026

Department of Mechanical and Aerospace Engineering

Contents

1 Introduction

This report presents a comprehensive post-processing analysis of an internal combustion engine (ICE) utilizing MATLAB to evaluate experimental data and combustion characteristics. The study focuses on a Compression Ignition (CI) engine equipped with Exhaust Gas Recirculation (EGR).

The primary objectives of this analysis include:

- Steady-State Evaluation: Normalizing measured performance metrics to standard reference conditions according to the ISO 1585 standard.
- Harmonic Analysis: Utilizing Fourier Transforms to reconstruct in-cylinder pressure and torque signals, determining the global contributions of various harmonic orders.
- Flywheel Dimensioning: Calculating the necessary rotational inertia for both single-cylinder and multi-cylinder configurations to regulate angular velocity fluctuations.
- Combustion analysis: Performing a Heat Release Rate (HRR) analysis using the Rassweiler-Withrow method to identify critical combustion indices, such as mass fraction burned and ignition delay.

2 Steady-State Test Evaluation and In-Cylinder Pressure Analysis

This chapter details the post-processing procedures applied to the experimental data obtained from steady-state engine testing and the analysis of the in-cylinder pressure cycle. The evaluation was performed using MATLAB to compute performance parameters, apply standard corrections, and analyze combustion characteristics.

2.1 Steady-State Performance and ISO 1585 Correction

The initial dataset, comprising variables such as engine speed, torque, fuel mass flow, and intake conditions, was imported from the datasheet for processing. A primary objective was to normalize the measured performance metrics to standard reference conditions to ensure comparability. The correction was performed according to the ISO 1585 standard, suitable for internal combustion engines.

The reference atmospheric conditions were defined as a dry pressure of 99 kPa and a temperature of 298 K. For the Compression Ignition (CI) engine under analysis, the correction factor (μ_c) was calculated based on the atmospheric factor (f_a) and the engine factor (f_m). The atmospheric factor accounts for variations in dry air pressure and intake temperature, while the engine factor f_m is determined by the corrected fuel delivery parameter (q_c). As implemented in the code, specific thresholds for q_c (37.2 mg/cycle and 65 mg/cycle) were utilized to determine the appropriate value of f_m, distinguishing between different load regions.

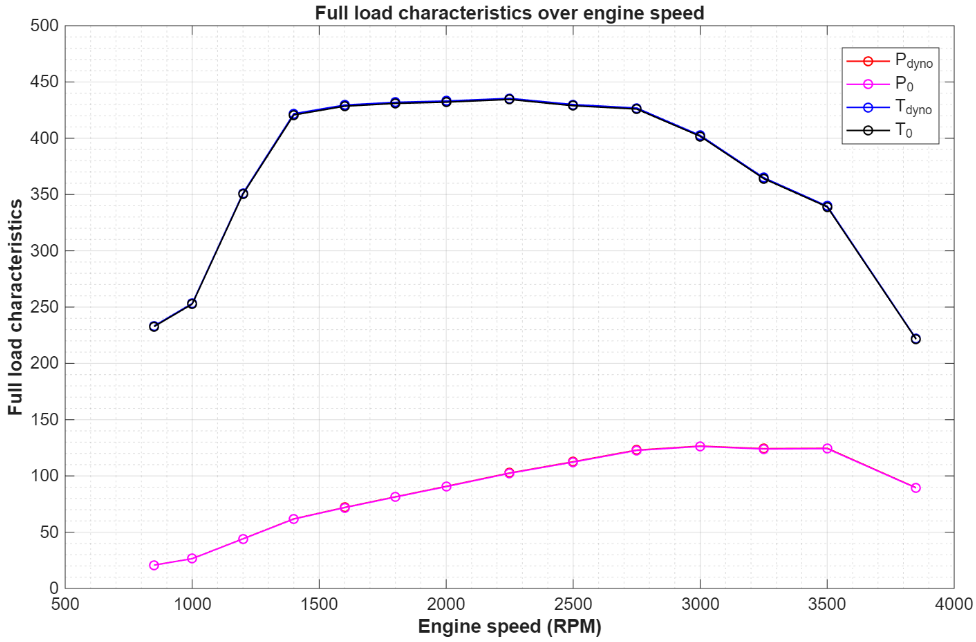

Figure 1: Full load characteristic.

The corrected power (P_0) and torque (T_0) were subsequently calculated by multiplying the measured dynamometer values by the correction factor μ_c. The correction factor trend across the engine speed range was also analyzed to verify the validity of the tests, ensuring that the values remained within the acceptable range defined by the standard.

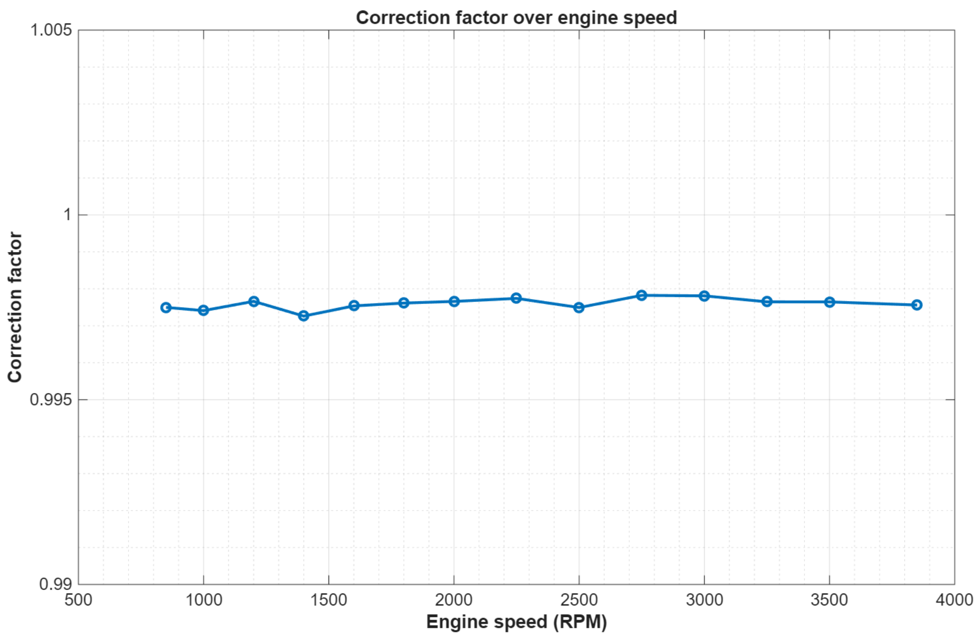

Figure 2: Correction factor μ_c.

Following the power correction, the fuel conversion efficiency (η_f) was computed using the corrected power and the lower heating value of the fuel (Q_{LHV}), set to 42.5 MJ/kg. The Brake Specific Fuel Consumption (BSFC) was derived inversely from the efficiency to quantify the fuel mass consumed per unit of energy produced.

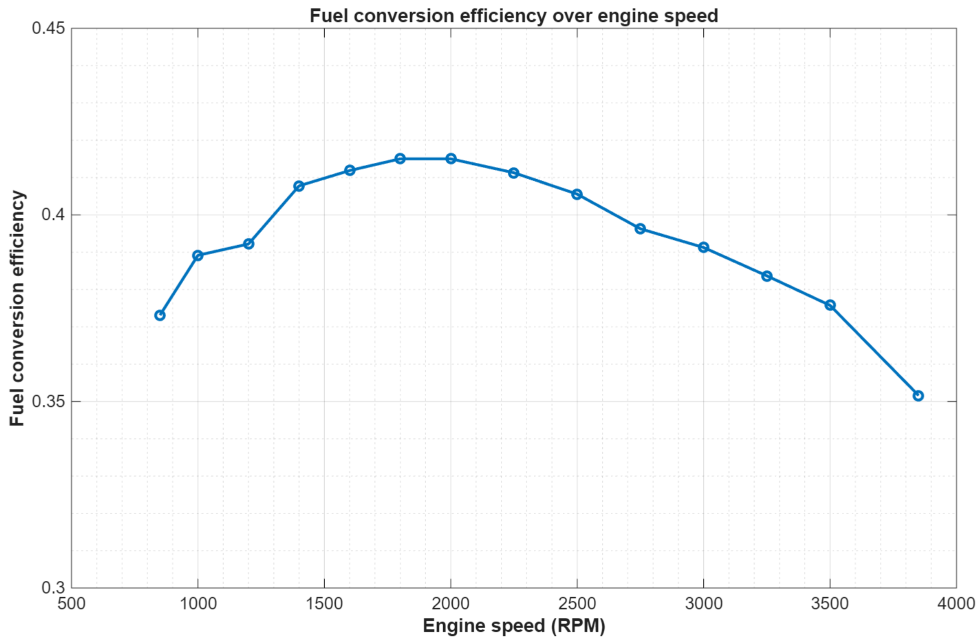

Figure 3: Fuel conversion efficiency η_f.

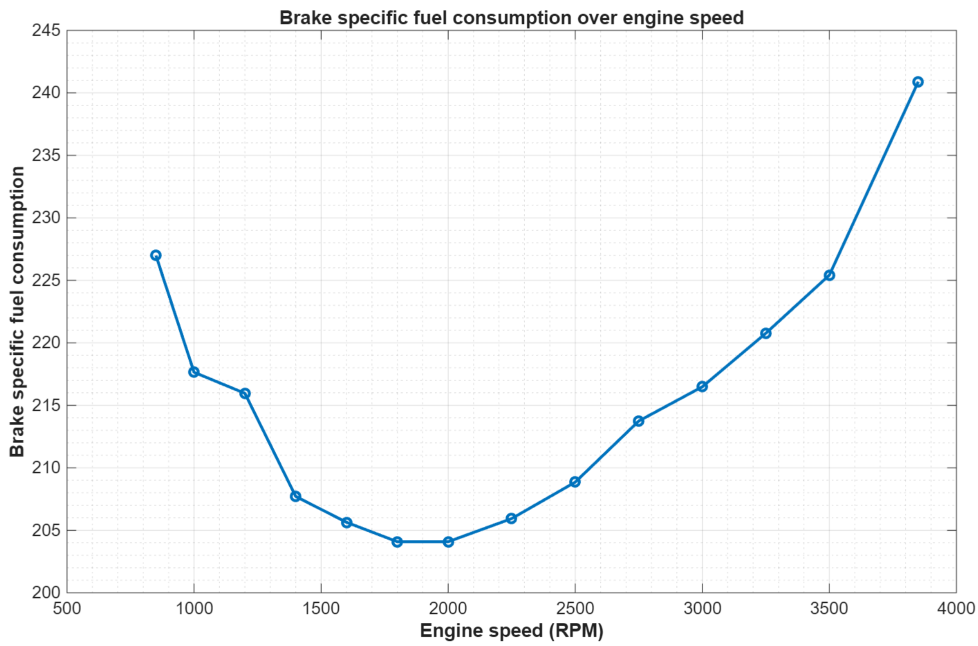

Figure 4: Brake specific fuel consumption.

Volumetric efficiency (λ_v) was calculated to evaluate the engine breathing characteristics. Given the presence of Exhaust Gas Recirculation (EGR) in the tested engine configuration, the calculation incorporated both the air mass flow and the EGR mass flow. The density of the intake mixture was determined using a weighted gas constant (R_{mix}), derived from the respective mass fractions and gas constants of air and EGR.

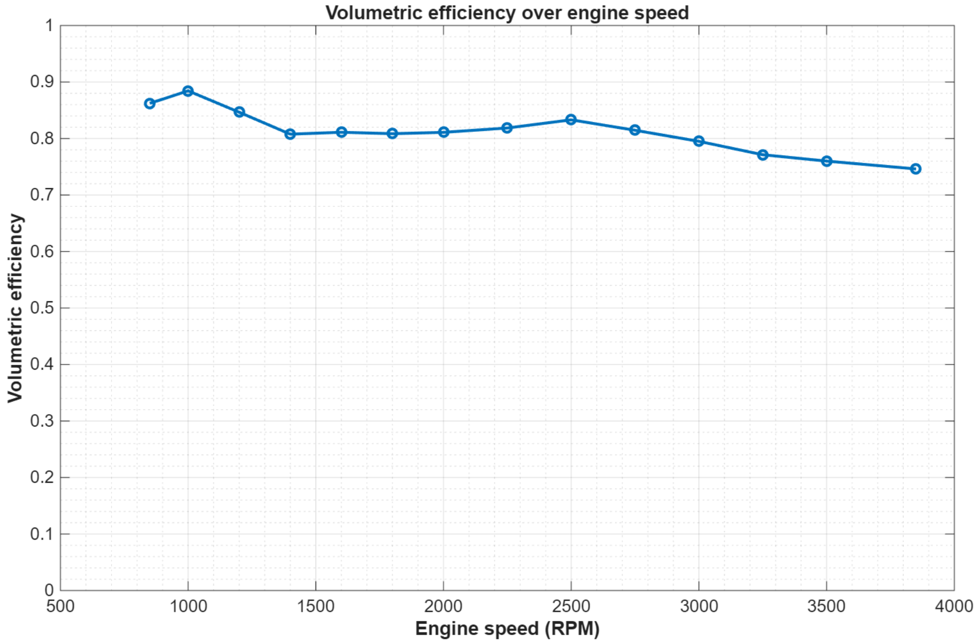

Figure 5: Volumetric efficiency λ_v.

2.2 In-Cylinder Pressure Processing

Detailed combustion analysis was performed on a specific dataset acquired at 2000 rpm under full-load conditions. The raw data consisted of in-cylinder pressure traces from 100 consecutive cycles.

The piezoelectric transducers used for these measurements provide a relative pressure signal rather than an absolute value. Consequently, a pegging procedure was necessary to reference the signal. The ensemble average of the raw pressure was computed over the 100 cycles to mitigate cycle-to-cycle variations. The signal was then pegged by aligning the mean cylinder pressure during the bottom dead center (BDC) of the intake stroke (between 175 and 185 degrees crank angle) with the mean absolute manifold pressure measured during the same interval.

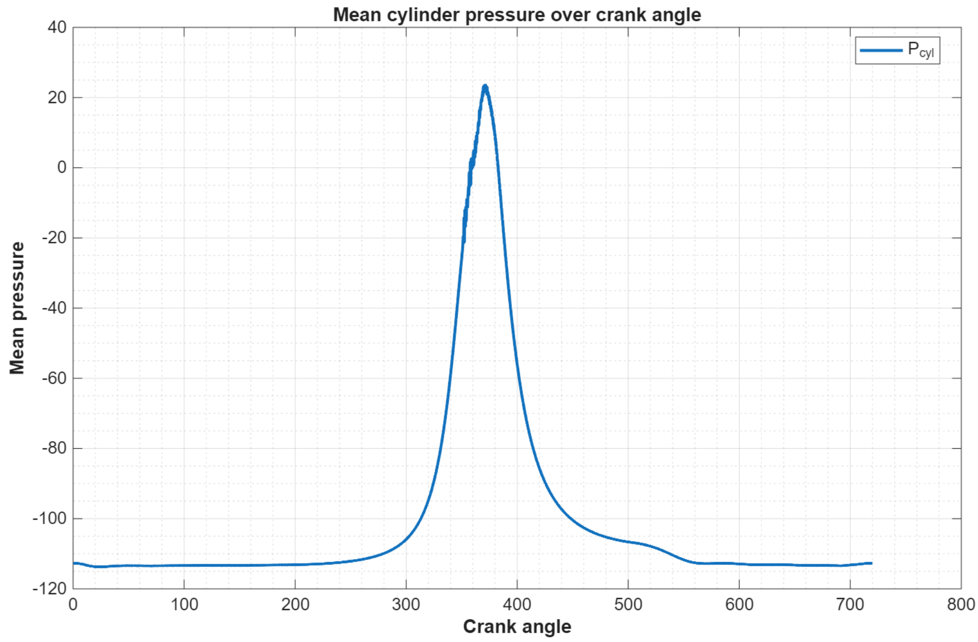

Figure 6: Mean in-cylinder pressure (raw signal).

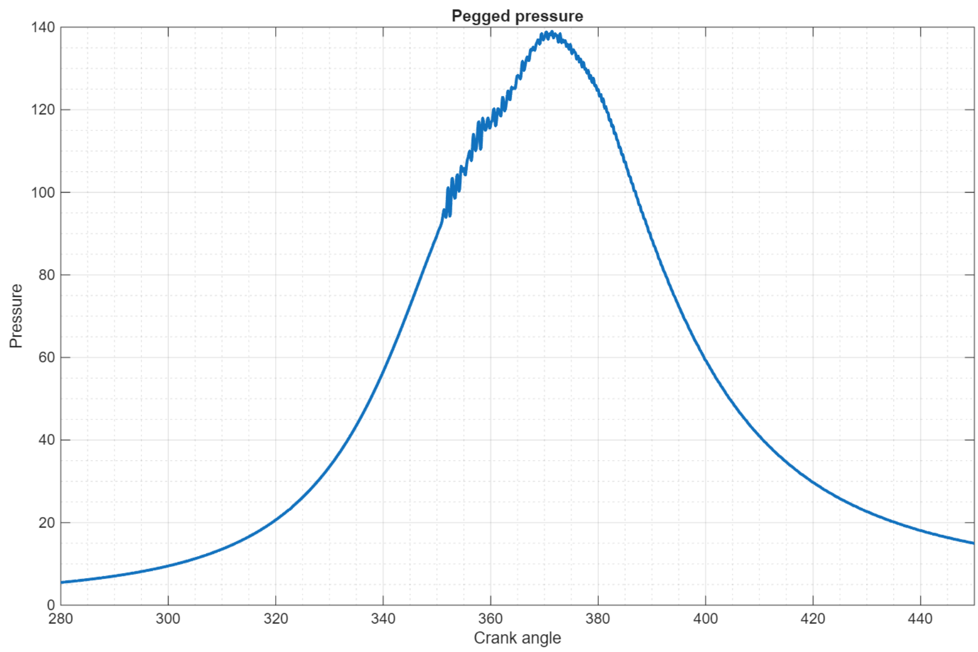

Figure 7: Pegged pressure signal.

To eliminate high-frequency noise inherent in the measurement, specifically resonance effects within the combustion chamber, digital filtering was applied. Three filtering methods were implemented and compared: a Finite Impulse Response (FIR) moving average filter, a standard moving mean (`movmean`), and a Butterworth low-pass filter. A second-order Butterworth filter with a cutoff frequency of 4000 Hz was selected for the final analysis, as it effectively attenuated high-frequency noise while preserving the peak pressure characteristics without introducing a significant phase shift.

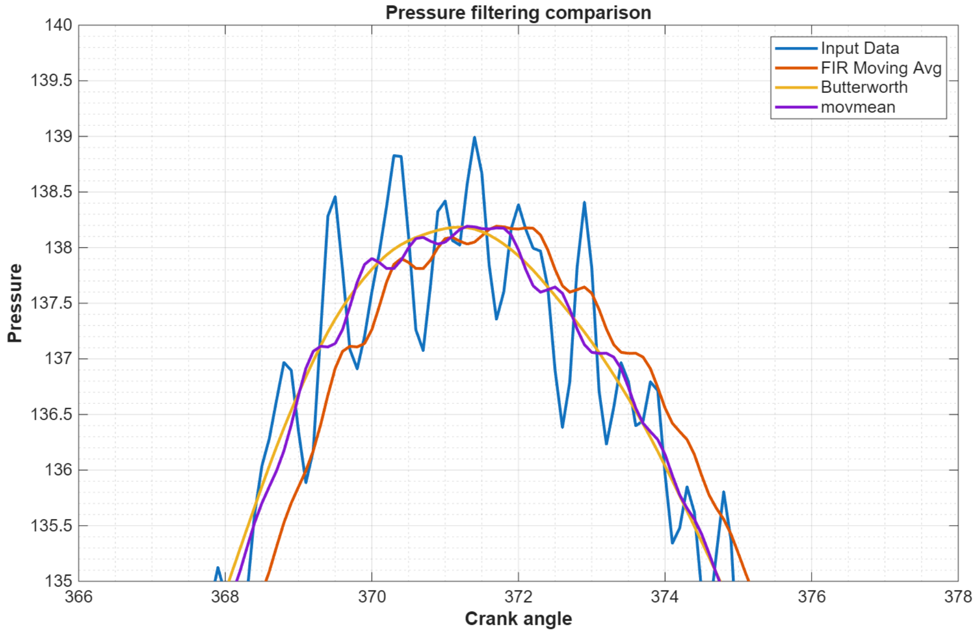

Figure 8: Comparison of the raw pressure signal with different filterings.

2.3 Thermodynamic Analysis and IMEP Calculation

The instantaneous cylinder volume (V_x) was calculated as a function of the crank angle using the engine geometric parameters, including bore, stroke, connecting rod length, and compression ratio. This kinematic relationship allowed for the transformation of the pressure data from the time domain (crank angle) to the volume domain.

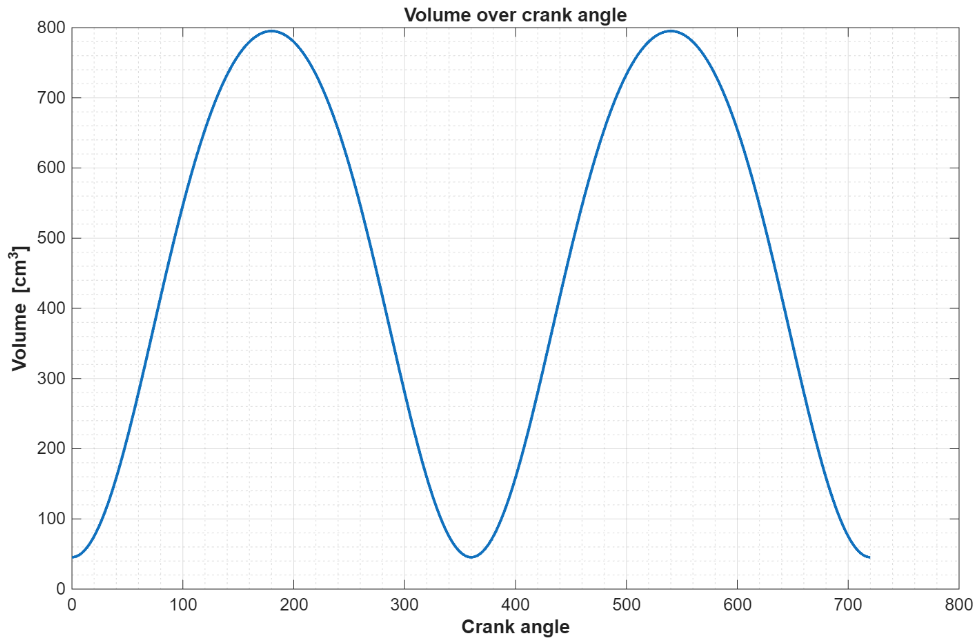

Figure 9: Instantaneous cylinder volume over crank angle.

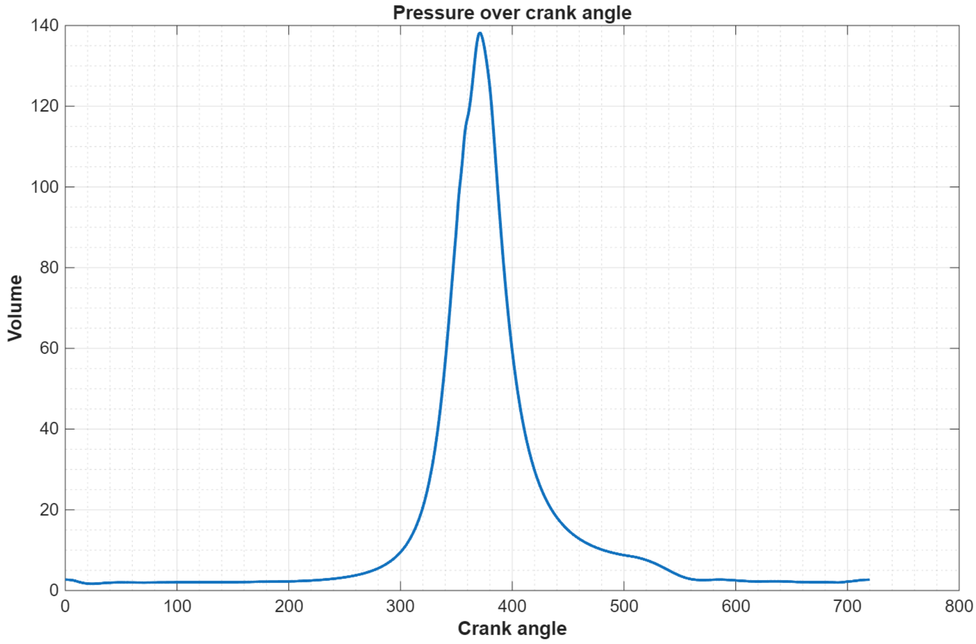

Figure 10: Filtered in-cylinder pressure over crank angle.

The correlation between pressure and volume was visualized through p–V diagrams. These were plotted in linear coordinates to visualize the work loop and in logarithmic coordinates to analyze the compression and expansion polytropic exponents. A normalized diagram (p vs. $V/V_{\max}$) was also generated for non-dimensional analysis.

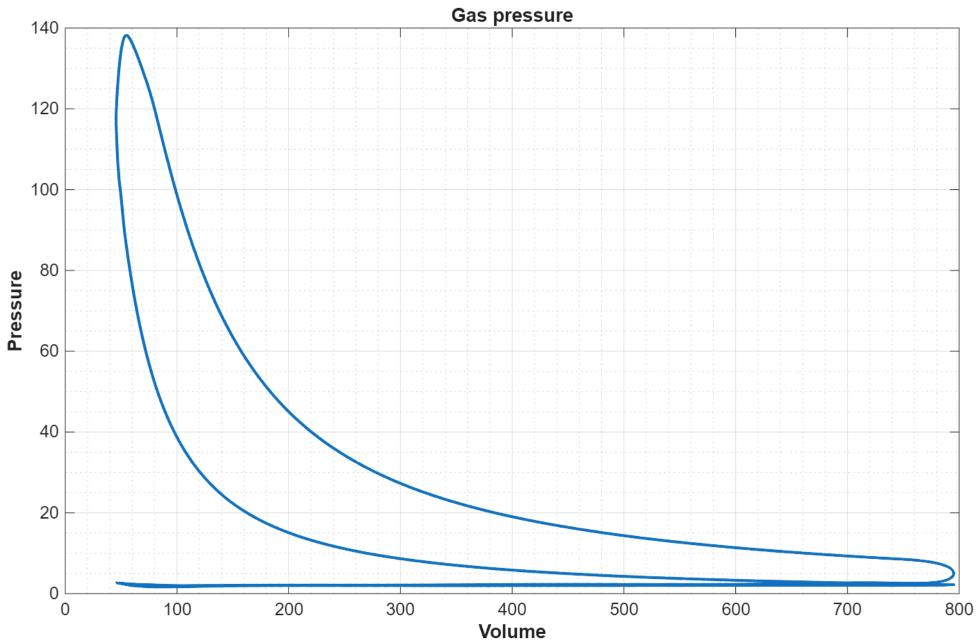

Figure 11: Linear p-V diagram.

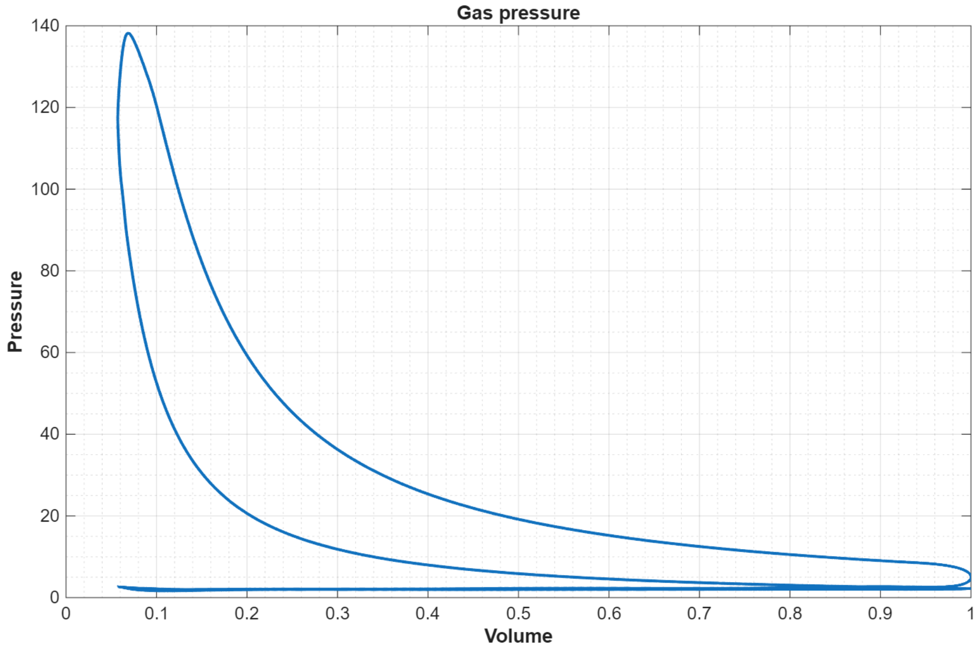

Figure 12: Normalized p-V diagram (pressure vs. V/Vmax).

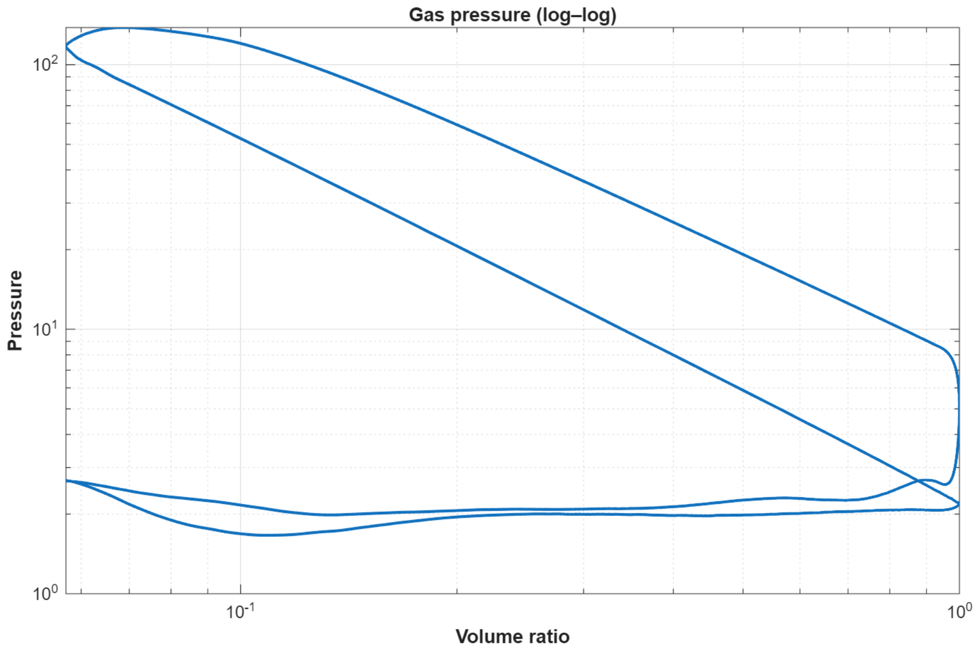

Figure 13: Logarithmic p-V diagram.

Finally, the Indicated Mean Effective Pressure (IMEP) was calculated by integrating the pressure over the volume change using the trapezoidal numerical integration method. Two distinct IMEP values were computed:

- **Gross IMEP ($IMEP_g$):** Calculated over the compression and expansion strokes (180° to 540°), representing the work delivered to the piston during the high-pressure part of the cycle.

- **Net IMEP ($IMEP_n$):** Calculated over the entire 720° cycle, accounting for the pumping loop losses.

The Brake Mean Effective Pressure (BMEP) was derived from the corrected dynamometer torque. The gross mechanical efficiency (η_m) of the engine at this operating point was subsequently determined by the ratio of BMEP to Gross IMEP, quantifying the mechanical losses due to friction and auxiliary drives.

The following are the results obtained from this analysis:

- imep gross = 19.63 bar
- imep net = 19.33 bar
- Bmep = 18.10 bar
- Mechanical efficiency = 0.92

3 Fourier analysis and calculations

3.1 Introduction to Fourier analysis

The second part of this report is based on the in-cylinder pressure data pegged and filtered by the butterworth filter, related to steady-state conditions, considering the full load test at 2000 rpm.

Initially, the in-cylinder pressure signal is reconstructed using the Fourier Transform. the In-cylinder pressure harmonic spectrum is described by their amplitude and their order.

Then, instantaneous torque is computed as a function of the crank angles for both single-cylinder and multi-cylinder engines. The rotating vectors of the different orders for the four cylinders are plotted to understand the global contribution of each harmonic.

3.2 Fourier analysis

Fourier analysis is the study of the way in which general functions can be approximated by sums of trigonometric functions. It can be used to determine the components of a signal, called harmonics. Generally, Fourier analysis is applied to signals that depends on time sampled at equal time intervals of period T. The Fourier transform can convert these samples into a sequence of amplitudes for sine and cosine functions with a frequency that is a multiple of $\frac{1}{T}$. This is useful for identifying periodic components of periodic or non-periodic signals.

In general, any periodic waveform can be expressed as :

$$f(t) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n cos(n\omega t) + b_n sin(n\omega t)$$

where the first term $a_0/2$ is the average value of the function, while the terms a_n and b_n represent the cosine and sine components of each signal harmonic.

In this report, the Fourier transform is used to approximate the pressure and torque trends over the crank angle, and a plot is used to show the comparison between them. The number of harmonics considered will be crucial for finding an accurate approximation.

3.3 In-cylinder pressure analysis

The data are provided with a resolution of 0.1 °CA. To reduce the computational time for the Fourier analysis, it can be helpful to downsample the signal, which means that only one measurement every DSF is taken. Thus, the new encoder resolution is equal to 1 °CA.

When the function fft (Fast Fourier Transform) is applied to the signal, the output is a vector Y of complex values with the same length as the signal vector. In the script, the first 10 harmonics are considered; this number is sufficient to obtain an accurate representation of the original pressure trend.

Given Y, the magnitude and phase of each harmonic are evaluated. It is important to compute the first harmonic as it represents the average value of the pressure over the period.

$$Magnitude\ c_0\ (order\ 0) = |Y(1)| \tag{1}$$

$$Magnitude\ c_n\ (order\ n) = 2 \cdot |Y(1)| \tag{2}$$

$$phase\ \phi = angle(Y(n)) + \pi/2 \tag{3}$$

n is the harmonic order, while N is the total number of harmonics.

To obtain the signal approximation, the contribution of each harmonic should be summed:

$$f(t) = c_0 + \sum_{n=1}^{N} c_n sin(n\theta/m + \phi_n) \tag{4}$$

where θ is the crank angle in radians, and $m = 2$ is the parameter that refers to 4-stroke engines.

3.4 Instantaneous torque single-cylinder harmonic analysis

To evaluate the instantaneous torque produced by the engine, the effective pressure acting inside the cylinder must be defined. It is the sum of the gas pressure, the inertia pressure, and the crankcase pressure. The gas pressure is the pressure filtered in the first part, the crankcase pressure is the atmospheric pressure and is equal to 1.01325 bar. The inertia pressure is due to the inertia force of the piston and has a negative

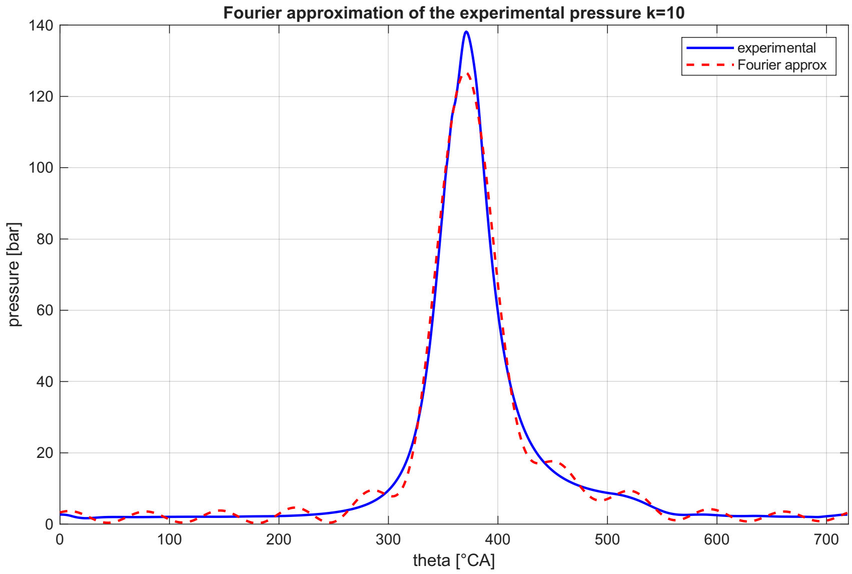

Figure 14: the first 10 harmonics approximate well the pressure trend.

value:

$$F_{in} = -m_{rec} * \omega^2 r \cdot (cos\theta + \lambda \frac{cos2\theta}{cos\beta}$$

$$p_i = \frac{F_i}{A_p}$$

$$P_{eff} = p_g - p_c + p_i$$

The effective pressure acting on the piston is transmitted to the connecting rod with a magnitude F, which depends on the angle β according to the scheme below. The projection of F in the direction orthogonal to the shaft F_T produces a rotational moment M_s, which is responsible for the crankshaft rotation. The rotational moment is the result of the product of the tangential force and r, which is half the engine stroke, while the tangential pressure is the ratio of M_s to half of the volume displacement.

$$M_t = F_t r \tag{5}$$

$$p_t = \frac{p_t}{V_d/2} \tag{6}$$

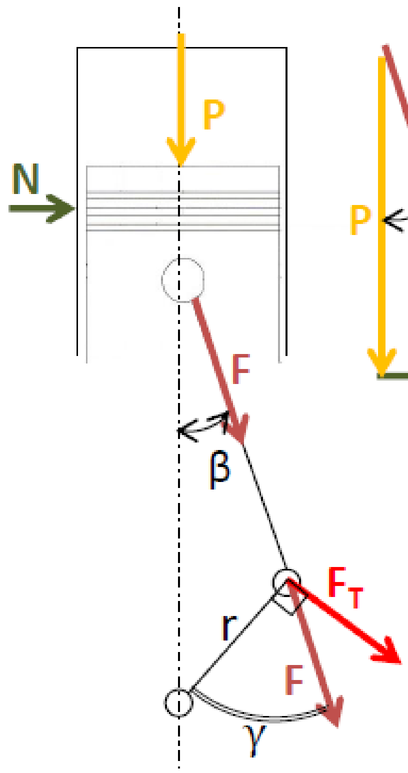

Figure 15: Scheme representation of force acting on the piston and the shaft.

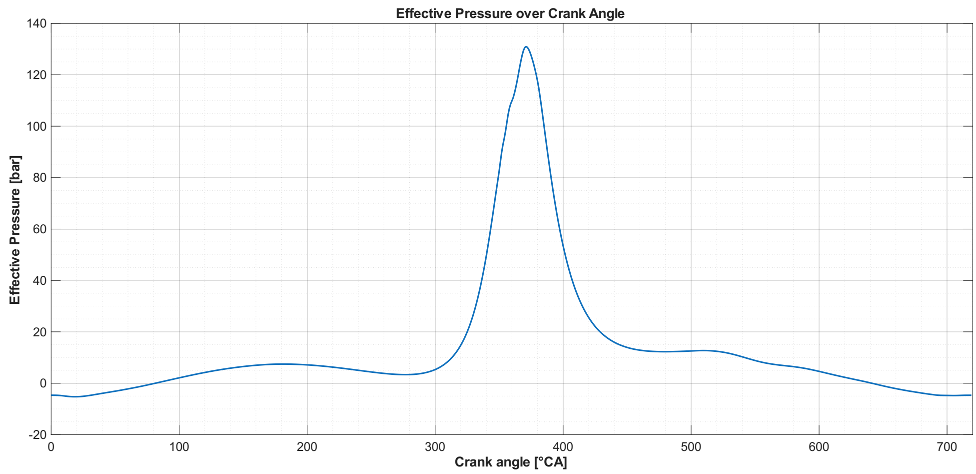

Figure 16: Effective pressure acting on the piston.

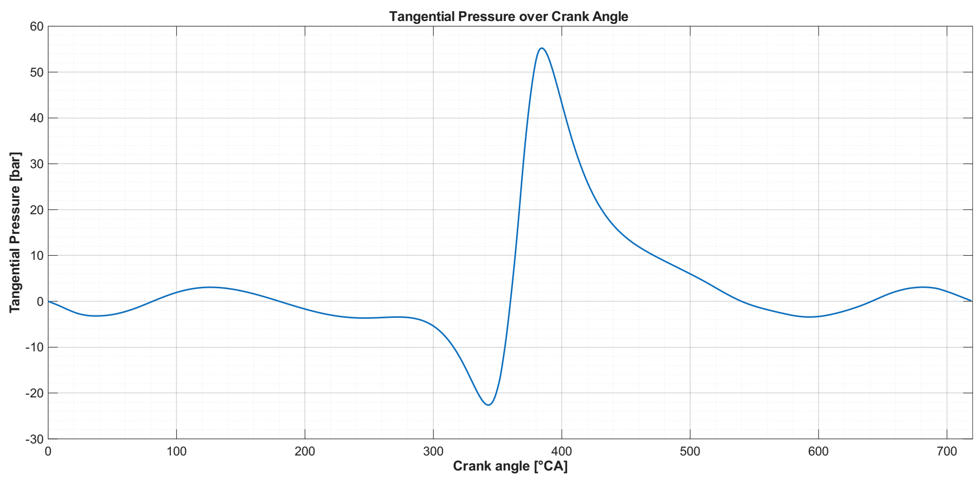

Figure 17: The tangential pressure that produces rotational moment.

The Fourier analysis conducted for the in-cylinder pressure can also be applied to the instantaneous torque for the single-cylinder engine. The first 10 harmonics are plotted. In figures 16 and 17, it can be observed that the effective pressure has a higher magnitude than the tangential pressure. Their curve shapes are also different: the effective pressure starts increasing in proximity to the start of combustion, nearly after 300℃A, then it reaches a peak at around 370℃A at over 130 bar. Finally, it decreases as the piston moves down. On the other hand, the tangential pressure spans both positive and negative values and reaches its maximum value at around 55 bar.

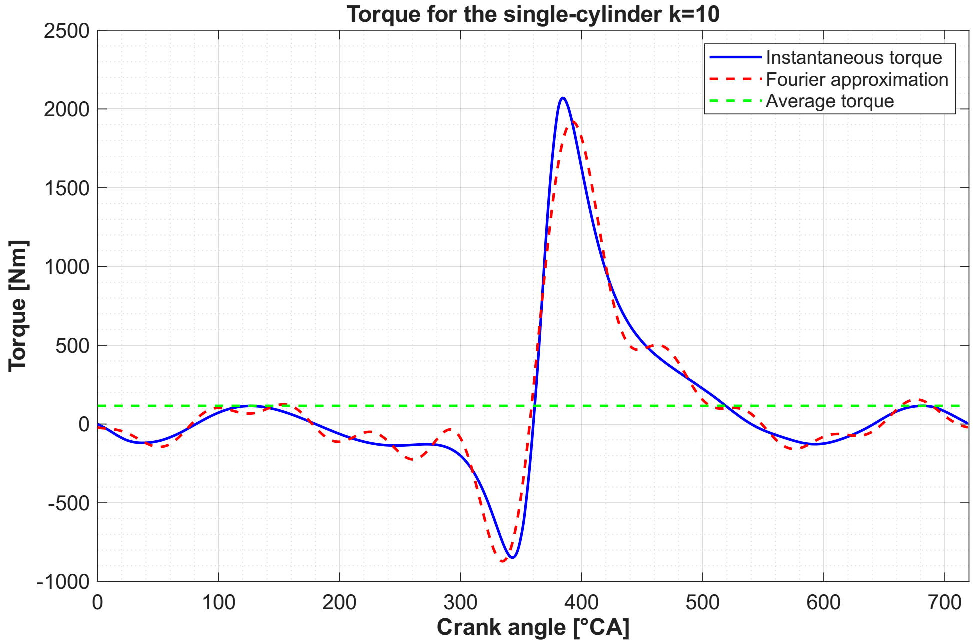

Figure 18: The FFT can approximate correctly the pressure behaviour including 10 harmonics.

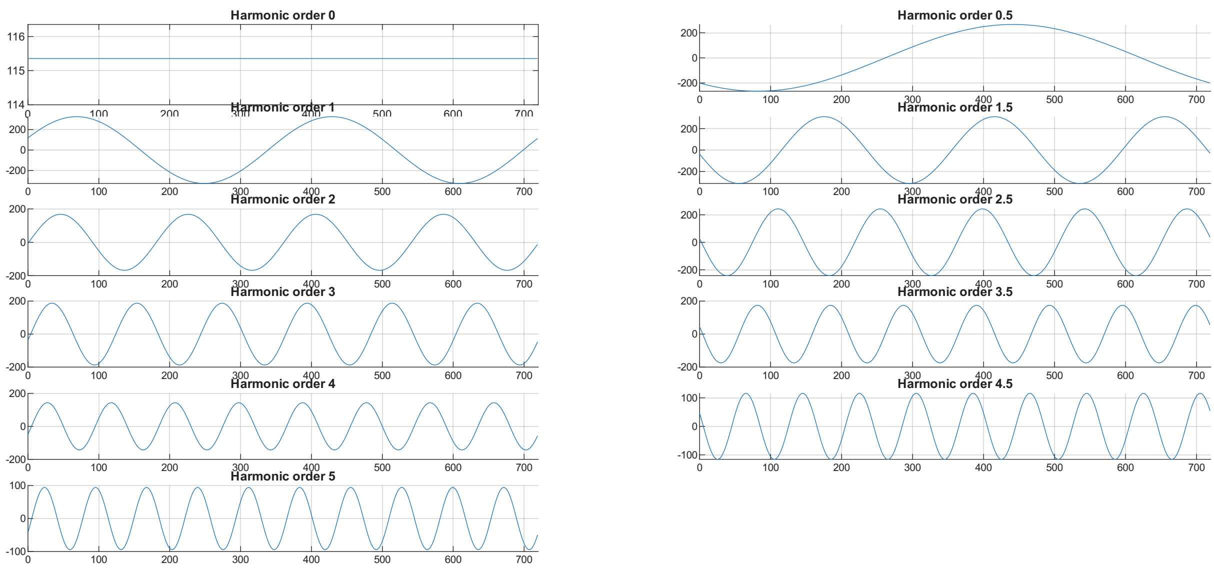

Figure 19: The first 10 harmonic orders of the torque signal.

The instantaneous torque for the single-cylinder configuration is obtained through the equation 5. By using the FFT approach, this signal can be easily reconstructed. In the figure 18, the Fourier approximation is shown. The approximation has a correct shape using the first 10 harmonics; therefore, there is a discrete ripple when the torque is around 0. The actual torque peak is over 2000 Nm, while the Fourier curve reaches a lower value.

In figure 19, the first 10 harmonics are plotted. Harmonic 0 is the average value, while as the harmonic number increases, the number of zero-crossings also increases, as expected.

It is also possible to show the spectrum of the amplitude of the harmonics, as shown in figure 20. The harmonic order k is equal to $n/2$, where n is the harmonic number. The highest contribution is given by the order 1 harmonic; then, the amplitude of higher orders decreases linearly.

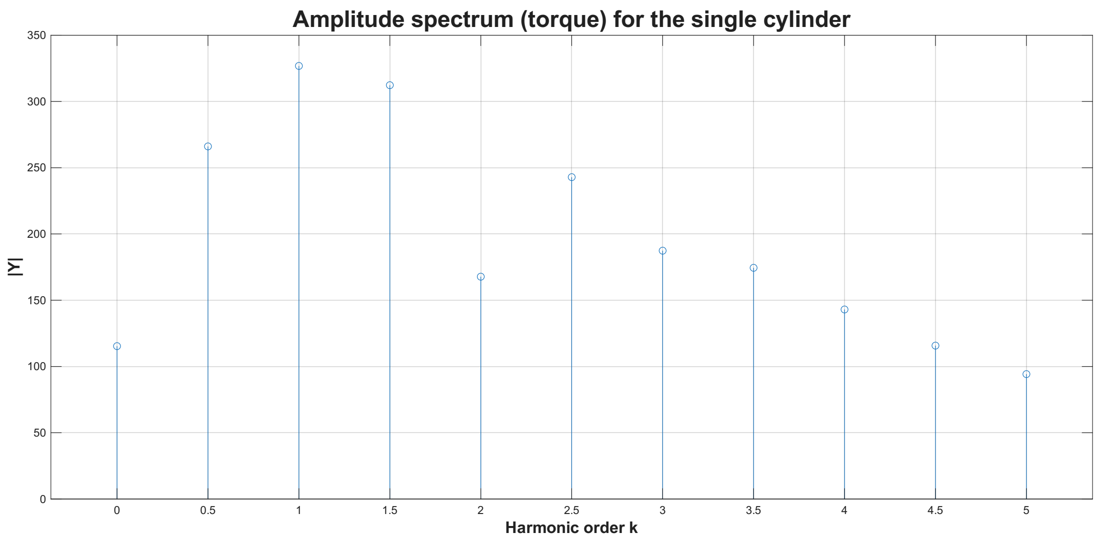

Figure 20: The contribution level of each harmonic.

3.5 Instantaneous torque multi-cylinder harmonic analysis

It is possible to evaluate the torque in the other three cylinders, considering that the torque in each cylinder is shifted by an angle $\Delta\phi$

$$\Delta\phi = \frac{360 \cdot m}{i} \tag{7}$$

in which $m = 2$, is a parameter referring to a 4-stroke engine and i is the number of cylinders. In this engine, it is equal to 4. As a result, each torque plot is shifted by 180℃A with respect to the previous one.

The firing order in a 4-cylinder engine is usually 1-3-4-2, so the torque of cylinder 3 will be shifted by 180℃A with respect to cylinder 1 and so on. In the following plots, the instantaneous torque for the remaining three cylinders is plotted.

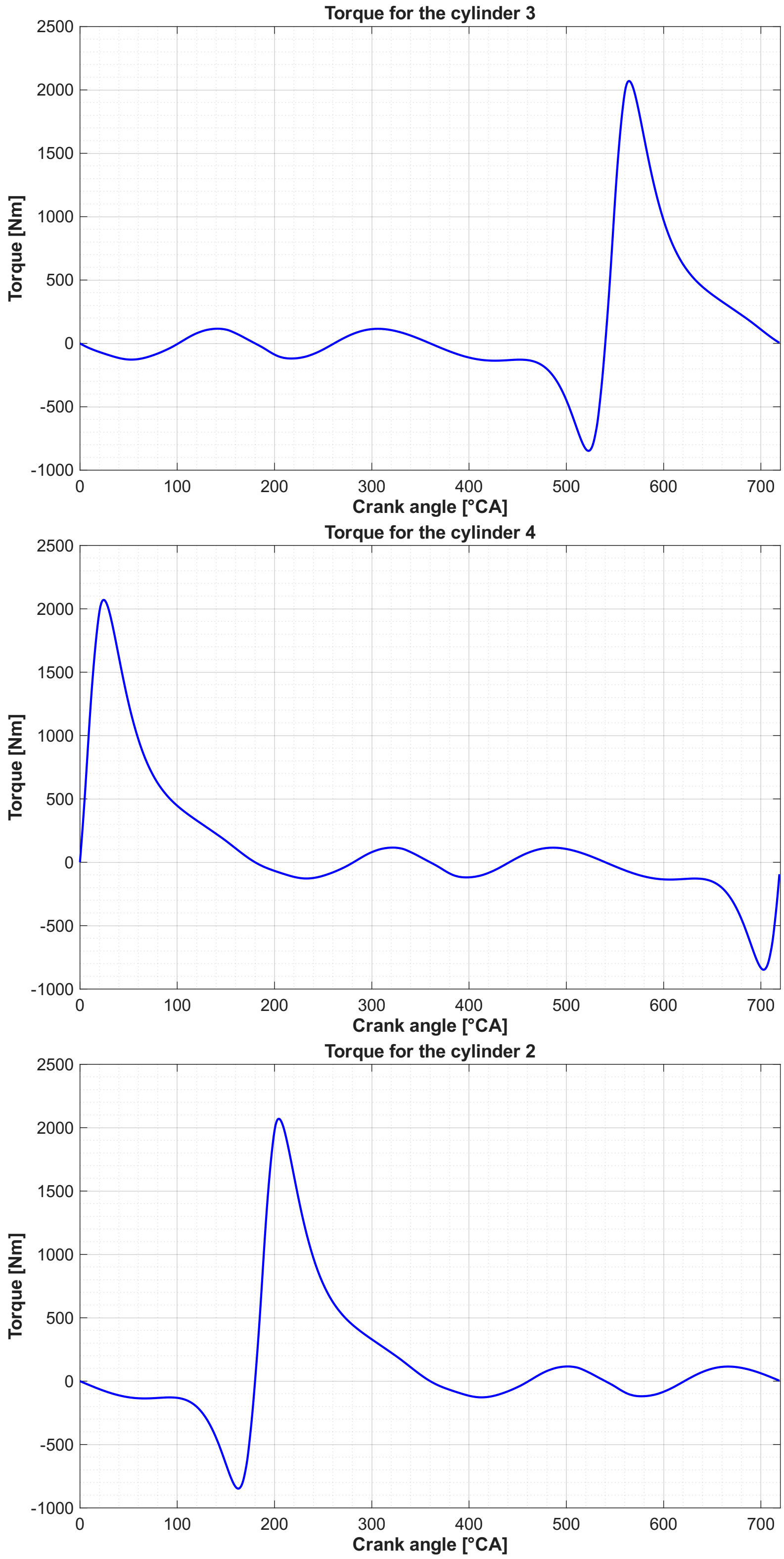

Figure 21: Torque trend over crank angle for cylinders 2,3 and 4.

The instantaneous torque for the multi-cylinder engine is the sum of the torque of each cylinder, while the average torque value is $461\ Nm$. Figure 22 shows that the period is one-fourth of the single cylinder period.

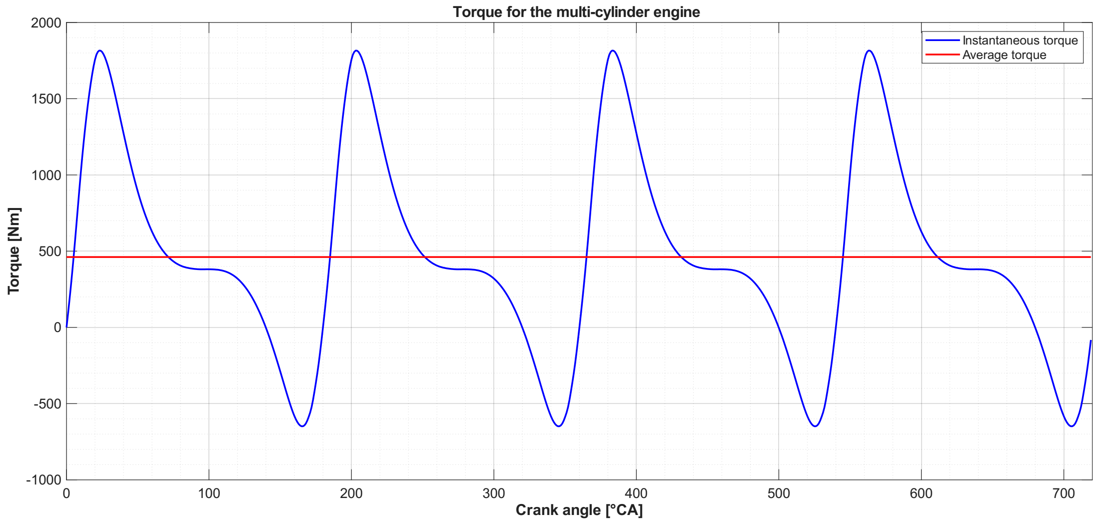

Figure 22: Torque for the multi cylinder engine and average torque

In figure 23, rotating vectors are plotted. They are phasors with a fixed-length arrow that spins at the angular frequency of the wave, allowing phase differences and combinations (addition/subtraction) to be handled like simple vector mathematics. The length of the phasor shows the maximum (amplitude) value, its angle shows the phase, and its projection onto an axis gives the instantaneous value.

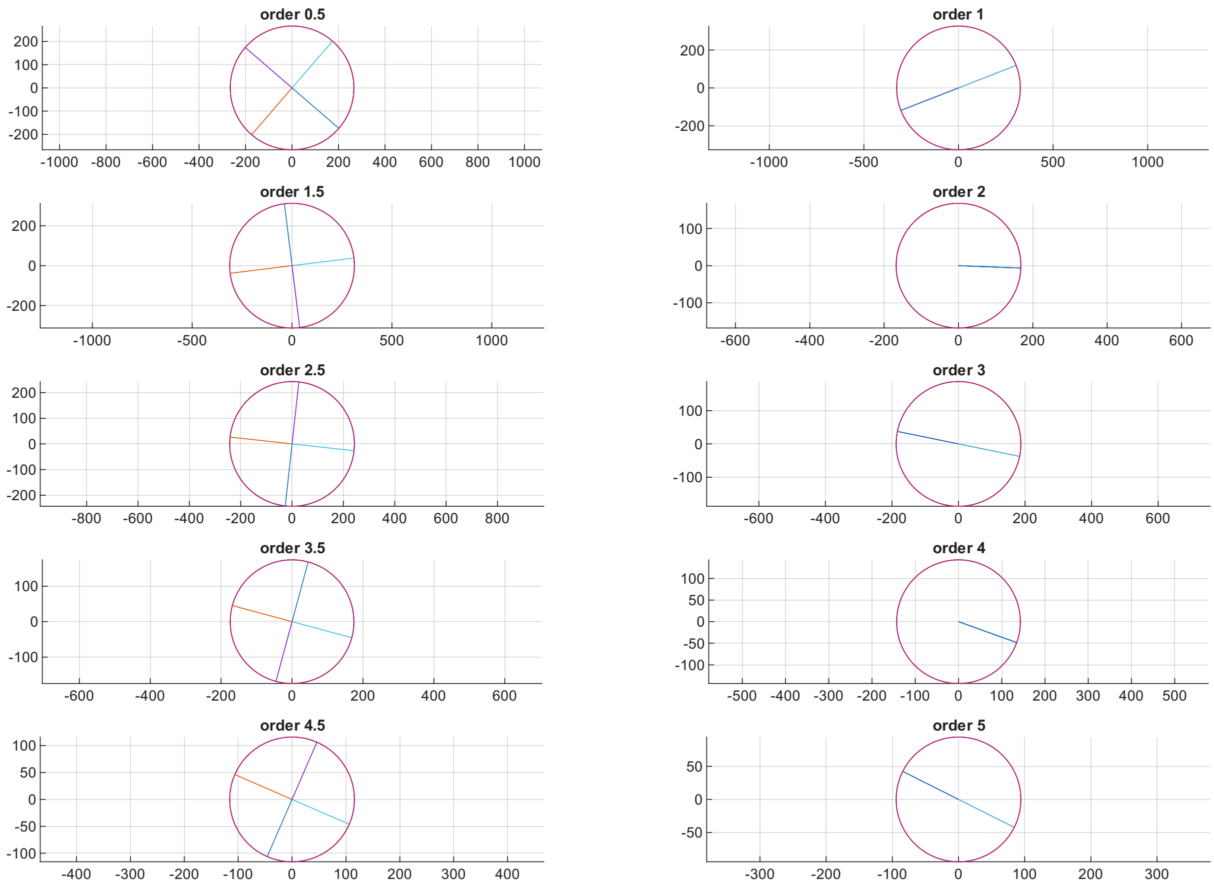

Figure 23: The rotating vectors show the global contribution of each harmonic.

The harmonics are plotted in figure 24, where it is evident that many harmonics have a cancelling effect, given the contribution of all the cylinders. By increasing the harmonic order, the frequency increases, as expected. In addition, it can be noted that in fractional orders, the contribution of each cylinder is shifted, resulting in mutual cancellation. On the other hand, in integer orders, some curves overlap. In particular, odd orders have 2 contributions that cancel each other out, while in even orders, there is only 1 curve.

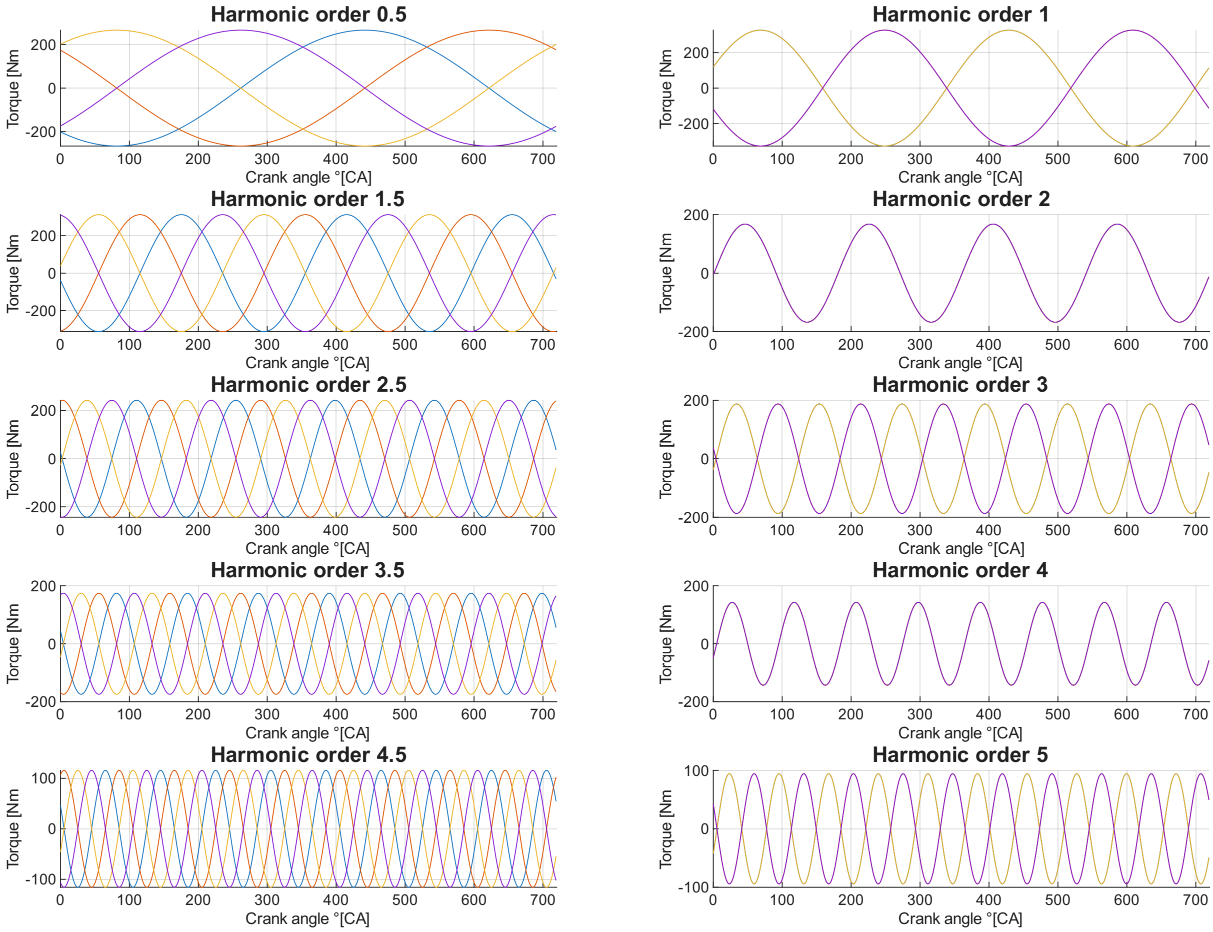

Figure 24: Some harmonics provide an erasing effect.

In the following plot, it is clear that only even harmonic orders provide a significant contribution to the spectrum of the multi-cylinder instantaneous torque. The harmonic with the highest amplitude is order 2, followed by order 4. All the other harmonics have no effect on the signal spectrum.

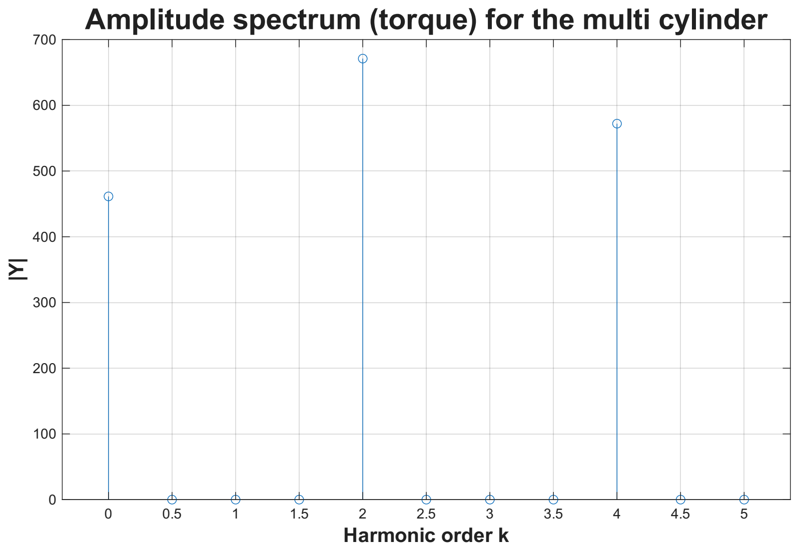

Figure 25: In the multi-cylinder spectrum only the harmonic orders 2, 4 and the average value remain.

In conclusion, figure 26 represents how the sum of the first 10 harmonics can approximate the instantaneous torque. This curve follows the overall trend of the original signal but appears smoother, capturing the dominant harmonic content while filtering out higher-frequency fluctuations. The approximation reproduces the main peaks and troughs with reduced sharpness, indicating that only a limited number of harmonics are used in the reconstruction. Anyhow, the approximation is sufficient to capture the general trend and the peaks of the torque.

A horizontal magenta line indicates the average torque. This line remains constant across the entire crank angle range and represents the mean value around which the instantaneous torque oscillates.

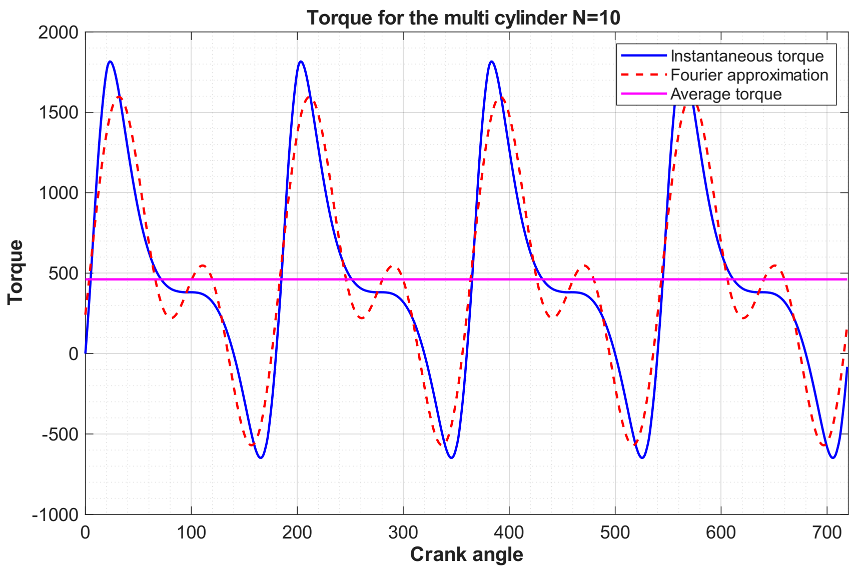

Figure 26: Fourier approximation for multi cylinder instantaneous torque.

4 Flywheel dimensioning

4.1 Definition of the Flywheel

A flywheel is a mechanical energy storage device designed to store rotational kinetic energy and release it when required. In internal combustion engines (ICEs), the flywheel is typically a rotating mass rigidly connected to the crankshaft, whose primary function is to regulate angular velocity fluctuations and ensure smoother engine operation.

From a physical standpoint, the flywheel exploits the principle of rotational inertia: a body with a sufficiently high moment of inertia resists variations in angular speed when subjected to fluctuating torques. This characteristic makes the flywheel an essential component in systems where torque delivery is inherently non-uniform, such as internal combustion engines.

4.2 Role of the Flywheel in Internal Combustion Engines

Internal combustion engines are characterized by cyclic and discontinuous torque production. In four-stroke engines, positive torque is generated only during the power stroke, while the remaining strokes (intake, compression, and exhaust) require energy input to overcome resistive forces. As a consequence, the instantaneous torque delivered to the crankshaft exhibits significant oscillations over each engine cycle.

The flywheel mitigates these oscillations by alternately absorbing and releasing energy:

- During phases of excess torque, the flywheel stores energy by increasing its rotational kinetic energy.
- During phases of torque deficit, the flywheel releases part of the stored energy, maintaining crankshaft rotation.

This energy exchange results in a more uniform angular velocity, improving the overall dynamic behavior of the engine.

4.3 Functions of the Flywheel

The flywheel performs several critical functions within an internal combustion engine system:

1. **Reduction of Speed Fluctuations**
   By increasing the rotational inertia of the crankshaft assembly, the flywheel reduces angular speed variations, ensuring smoother operation and minimizing vibrations.

2. **Energy Storage and Redistribution**
   The flywheel stores excess energy during the power stroke and redistributes it during non-powered strokes, enabling continuous rotation of the crankshaft.

3. **Improvement of Engine Stability**
   A properly dimensioned flywheel helps prevent engine stalling at low rotational speeds and improves idle stability.

4. **Assistance During Engine Start-Up**
   During starting, the flywheel contributes to overcoming compression resistance and inertial loads, facilitating engine ignition.

5. **Mechanical Interface Functions**
   In many applications, the flywheel also serves as a mounting surface for the clutch assembly and, in some cases, the starter ring gear, playing a structural role in power transmission.

4.4 Physical Principles Governing Flywheel Operation

The behavior of a flywheel is governed by the laws of rotational dynamics. The rotational kinetic energy stored in a flywheel is expressed as:

$$E_k = \frac{1}{2} J \omega^2 \tag{8}$$

where:

- J is the mass moment of inertia of the flywheel,
- ω is the angular velocity.

The ability of the flywheel to reduce speed fluctuations depends primarily on its moment of inertia, which in turn is influenced by its mass distribution relative to the axis of rotation. For this reason, flywheels are typically designed with a significant portion of their mass concentrated at larger radii.

The dimensioning of a flywheel represents a critical design task, as it directly affects engine performance, efficiency, and comfort. An undersized flywheel may lead to excessive speed fluctuations, vibrations, and unstable engine operation. Conversely, an oversized flywheel increases mass and inertia unnecessarily, negatively impacting engine responsiveness, acceleration, and fuel efficiency.

Therefore, the flywheel must be dimensioned as a compromise between:

- acceptable angular speed fluctuation,
- required energy storage capacity,
- mass and geometric constraints,
- mechanical strength and material limitations.

The analytical and numerical methods used for flywheel dimensioning will be discussed in the following chapter.

4.5 Flywheel Calculation

4.5.1 Single-cylinder Engine

The Analysis starts with the computing of flywheel for single-cylinder engine. The starting point of analysis is shifting of 360° the cycle up to expansion stroke using the Matlab function 'circshift' in order to consider the phase where work is developed from gas pressure to piston.
The images below show the gas pressure and tangential pressure shifted of 360°.

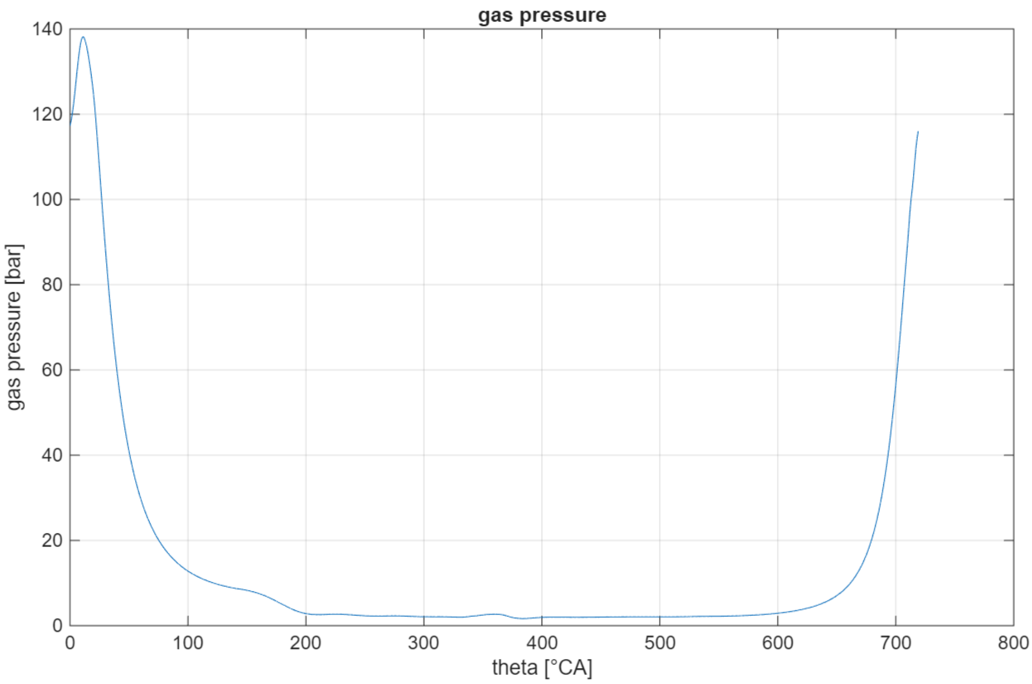

Figure 27: Gas pressure on piston

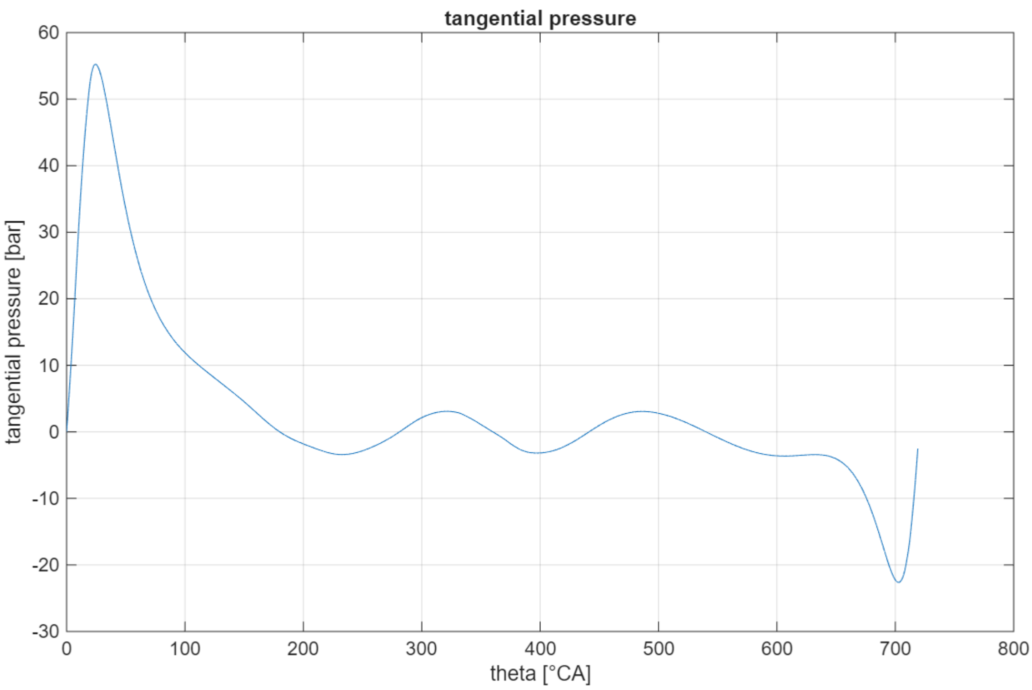

Figure 28: Tangential pressure

In the following, the engine will be considered as disconnected from the transmission and therefore, only the engine and flywheel inertia will be taken into account. The resistant moment can be considered constant over an engine cycle, since steady state condition are assumed, the instantaneous rotational speed of the engine at the beginning of the cycle is equal to the speed at the end of the cycle.

$$\omega(0) = \omega(4\pi) \tag{9}$$

Then, all parameters for computing the flywheel parameters are evaluated:

- density of flywheel material ($\rho = 7700kg/cm^3$)
- equivalent mass that creates an alternative motion (m_{alt})
- equivalent mass that creates a rotational motion (m_{rot})
- computing max and min angular speeds (ω_{max}) , (ω_{min})
- computing average angular speed (ω_{avg})
- computing kinematic irregularity (δ)

The kinematic irregularity is defined as:

$$\delta = \frac{\omega_{max} - \omega_{min}}{\omega_{avg}} \tag{10}$$

This parameter has a value limited to 0.01 to limit the maximum allowed speed fluctuation.
All calculations for the flywheel diameter are based on the fundamental equation of Dynamic:

$$M_s(\theta) - M_r(\theta) = J\frac{d^2\theta}{dt^2} \tag{11}$$

Then, the total Inertia is calculated, considering the flywheel and the engine:

$$J = J_{eng} + Jflyw \tag{12}$$

$$J_{eng} = (\frac{m_{rot}}{iV})V * r^2 \tag{13}$$

$$J_{flyw} = \int_{mass} r^2 * dm = \int_{mass} r^2 * 2\pi r * dr * w_{flyw} * \rho = \int_0^{D/2} r^3 * dr \tag{14}$$

$$J_{flyw} = 2\pi\rho * \omega_{flyw} * \frac{r^4}{4} = \frac{\pi}{32} * \rho * \omega_{flyw} * D^4 \tag{15}$$

Supposing $\omega_{flyw} = (1/10)D$, it is possible to calculate the required flywheel diameter that allows one to obtain the desired kinematic irregularity at the considered working point. The calculated diameter should stay within the range 2L<D<5L.
If this condition is not verified:

- if D>5L, it is possible to change the flywheel section (i.e. «C» shaped section instead of a planar disk)

- if D<2L, then we should reduce the flywheel width. Repeat the calculation of the diameter supposing w_{fly} = (1/15) D

To compute the total Inertia and evaluate the total diameter of the flywheel, the total resistant Torque must be computed in order to evaluate the resistant work as well as the shaft work.
For the evaluation of resistant and shaft work , the resistant and shaft torque must be integrated over the interval 0-720° using the Matlab function 'cumtrapz'.

$$dW_s(\theta) = M_s(\theta)d\theta \tag{16}$$

$$dW_r(\theta) = M_r(\theta)d\theta \tag{17}$$

Normalizing the shaft and the resistant work with respect to the unitary displacement V_d, it is possible to plot these quantities with the tangential and resistant torque (the two "work" terms are the integrals of the two "pressures")

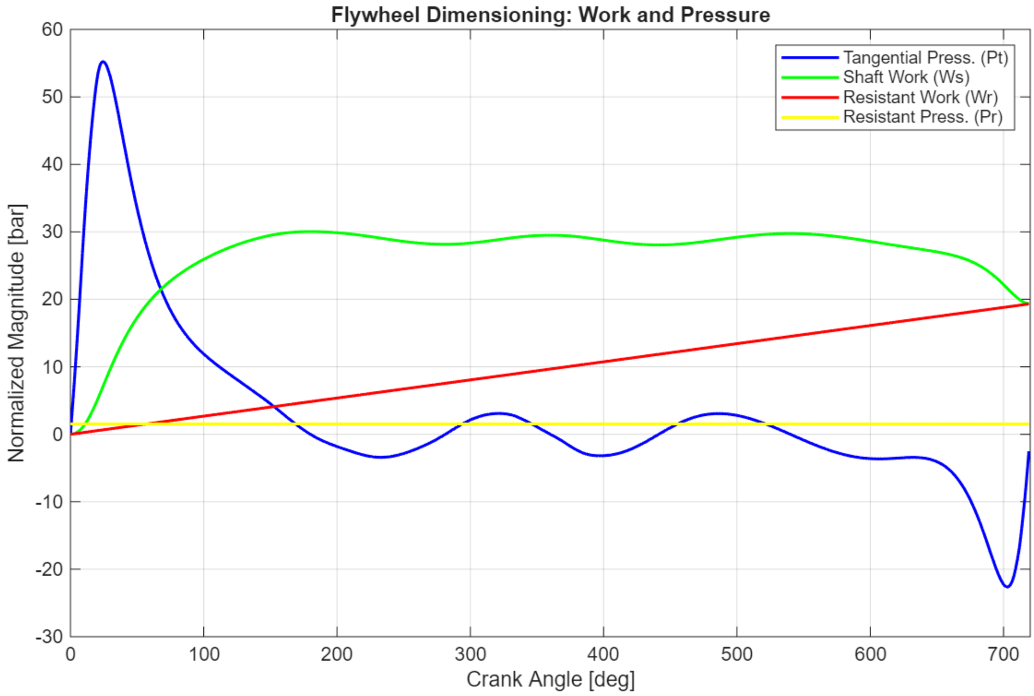

Figure 29: tangential and resistant pressure

It is important to see that at the end of the cycle, the shaft work and the resistant work are equals, as they must be.
The next step is to evaluate the maximum and minimum work variation in order to compute the Dynamic irregularity:

$$\Delta W(\theta) = W_s(\theta) - W_r(\theta) \tag{18}$$

$$\zeta = \frac{\Delta W_{max} + |\Delta W_{min}|}{W(4\pi)} \tag{19}$$

$$\zeta = \frac{\Delta W_{tot}}{imep * V_d} \tag{20}$$

Once the dynamic irregularity has been computed, it is possible to evaluate the total inertia in order to get the flywheel inertia:

$$J_{tot} = \frac{\zeta \cdot imep \cdot V_d}{\delta \cdot \omega_{avg}^2} \tag{21}$$

where ω is taken from dataset at 2000 rpm

$$J_{flyw} = J_{tot} - J_{eng} \tag{22}$$

From equation 22, flywheel diameter is then computed:

$$D_{flyw} = (320 \cdot \frac{J_{flyw}}{\pi \cdot \rho})^{0.2}; \tag{23}$$

The Flywheel diameter computed for the single-cylinder engine is 0.565 m.

Once the total inertia is computed, it is possible to calculate and plot the instantaneous crankshaft angular speed with or without the flywheel; from equation 23, integrating:

$$\int_0^\theta M_s(\theta)d\theta - \int_0^\theta M_r(\theta) = \int_{\omega_0}^{\omega} Jwdw \tag{24}$$

$$W_s(\theta) - W_r(\theta) = J[\frac{w^2 - w_0^2}{2}] \tag{25}$$

$$\omega(\theta) = \sqrt{\omega_0^2 + \frac{2}{J}[W_s(\theta) - W_r(\theta)]} \tag{26}$$

Where J is the J_{tot} when the instantaneous angular speed with the flywheel is computed; otherwise, J is just J_{eng}.

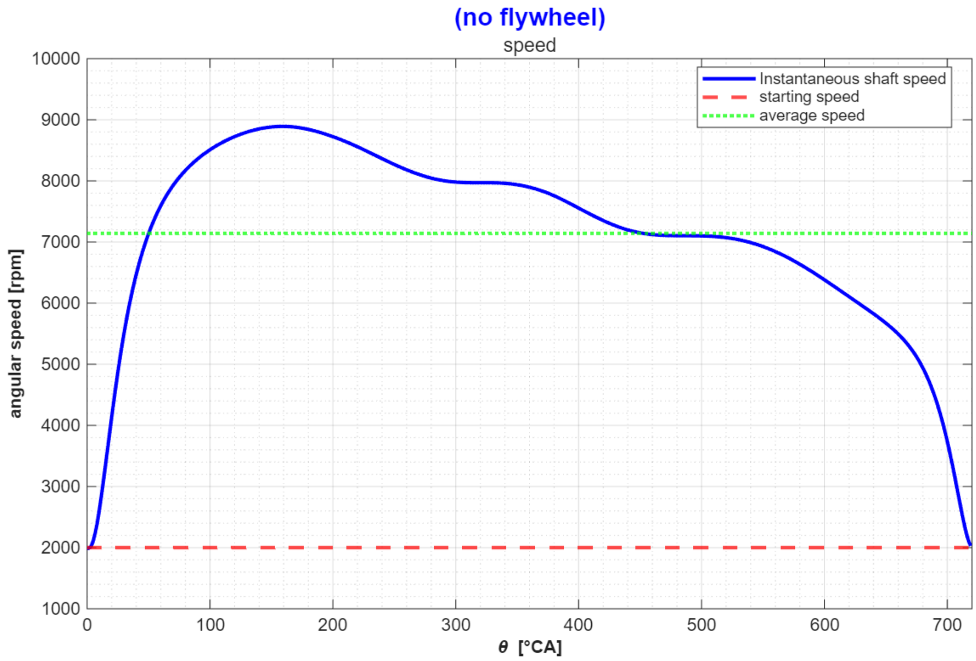

Figure 30: instantaneous crankshaft angular speed without flywheel

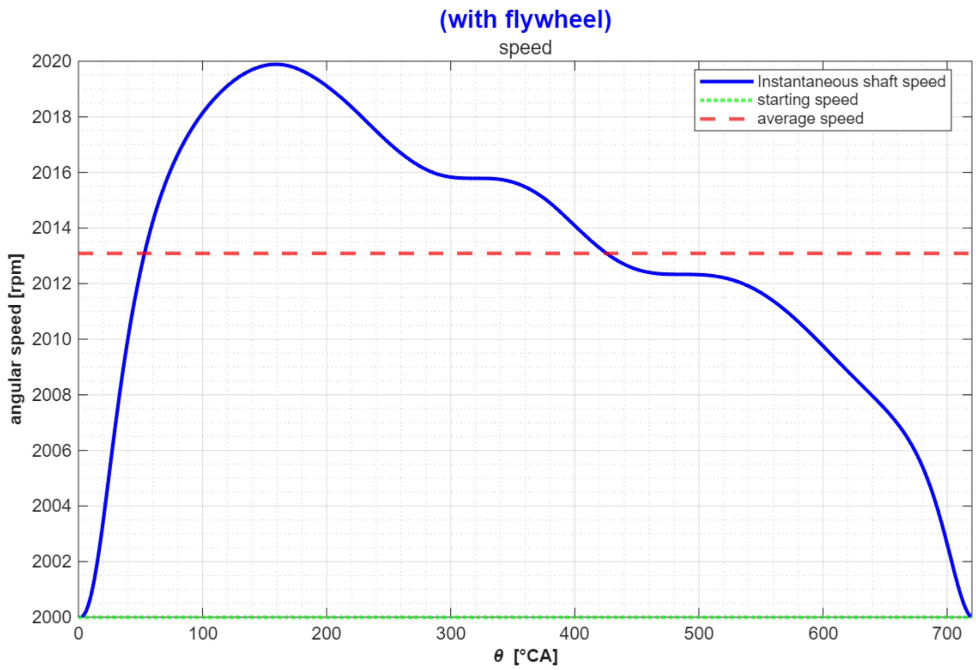

Figure 31: instantaneous crankshaft angular speed with flywheel

In this way, omega is expressed as a function of the shaft and resistant work and of the shaft speed at $\theta = 0$

$$\omega(\theta) = f(\omega(0), W_s(\theta), W_r(\theta)) \tag{27}$$

As a first approximation, we can consider $\omega(0) = \omega_{avg}$ and therefore we will calculate a first-attempt crankshaft speed ω_I.

The estimation can be improved by iterating the process: we can calculate the average value of ω_I

4.5.2 Multi-cylinder Engine

In order to evaluate the total moment acting on the crankshaft in a multi-cylinder engine, it is necessary to sum up the contributions provided by each cylinder, considering the phase shift among them and the firing order.

	FIRING ORDER
4 CYLINDERS ENGINE	1-3-4-2
5 CYLINDERS ENGINE	1-3-5-4-2
6 CYLINDERS ENGINE	1-4-2-6-3-5

Figure 32: Firing order for multi-cylinder engine

$$M_{s,\text{multi-cyl}}(\theta) = \sum_{j=1}^{i} M_{s,j}(\theta - \varphi_j) \tag{28}$$

$$p_{t,\text{multi-cyl}}(\theta) = \frac{M_{s,\text{multi-cyl}}(\theta)}{\frac{i\,V_d}{2}} \tag{29}$$

To obtain this value of M_s for the multi-cylinder engine, it is necessary to integrate the total torque obtained by previous calculations for multi-cylinder engine over the entire interval (0-720°) using the Matlab function 'cumtrapz'.

The tangential pressure for a multi-cylinder engine is also plotted, making the ratio between the torque of each cylinder and the displacement (V_d)

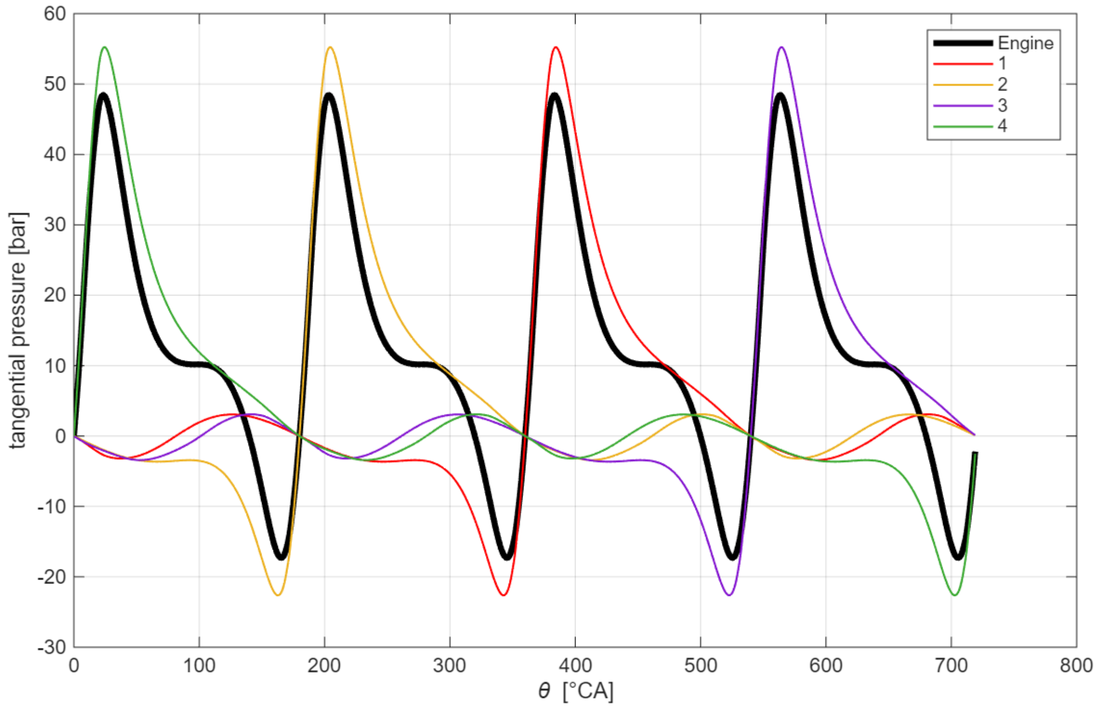

Figure 33: tangential pressure multi-cylinder engine

Integrating the previous equations over the complete engine cycle 4π, and taking into account the considerations already mentioned, we can obtain the resistant torque acting on the shaft.

As with the previous cases, it is also possible to calculate the shaft and the resistance works in the multi-cylinder engine by integrating the shaft and the resistant torque.

It is now possible to calculate $\Delta W_{max,multi-cyl}$, and $\Delta W_{min,multi-cyl}$. Through the definition of dynamic irregularity, it is possible to obtain the total rotational inertia J_{tot} in order to derive J_{flw} and compute the flywheel diameter in the multi-cylinder engine.

Once calculated, the value of the flywheel diameter is 0.471 m

At the end, it is possible to compute and plot the instantaneous crankshaft speed in multi-cylinder as well as made in previous case applying the same procedure used for the single cylinder, considering in this case the shaft and resistant work due to all the cylinders.

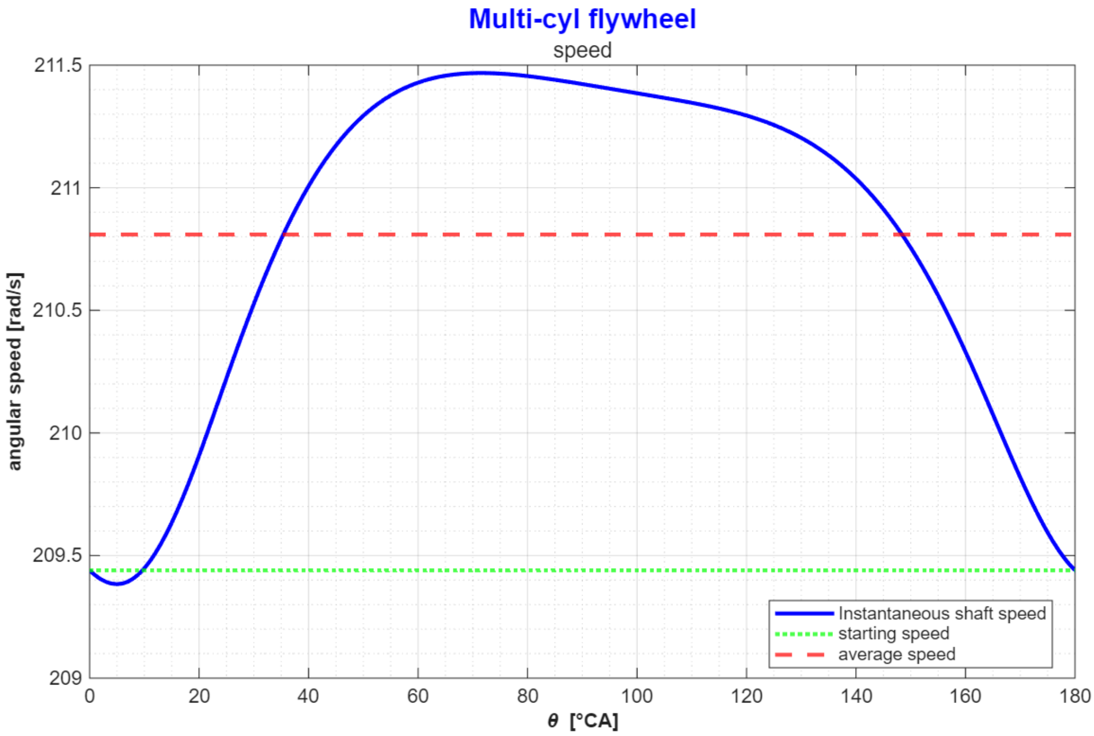

Figure 34: instantaneous angular speed in multi-cylinder engine

5 Heat Release Rate (HRR) analysis

In relation to this part, an analysis of the *Heat Release Rate* (HRR) was performed based on experimental in-cylinder pressure data acquired from a steady-state Compression Ignition (CI) engine test. The analysis is conducted using a pressure-based ***Rassweiler-Withrow (RW) method*** and a ***single-zone model***, assuming that the gas inside the combustion chamber is homogeneous. As for the assumptions, mass flows in and out of the control volume (by blow-by and injection before combustion) and heat losses through the cylinder walls are neglected.

The purpose of this analysis was to compute, in MATLAB, the curve of the mass fraction burned (x_b) and to identify key combustion parameters such as the Start of Combustion (*SOC*), the Start of Ignition (*SOI*), the Ignition Delay (*ID*) and the mass fraction burned at 10% of the crank angle *(MFB10)*, at 50% of crank angle *(MFB50)* and at 90% of crank angle *(MFB90)*.

5.1 Methodology

This chapter outlines the theoretical framework used for combustion diagnostics. Two main approaches are generally considered to estimate the mass fraction burned and analyze the combustion process: the pressure-based RW method and the sigle-zone HRR analysis.

5.2 Rassweiler-Withrow (RW) Method

The *Rassweiler-Withrow method* is a classic approach based on the analysis of in-cylinder pressure data. The fundamental assumption is that the total pressure variation (Δp) observed within a crank angle interval ($\Delta\theta$) is the sum of two contributions: the pressure change due to volume variation (Δp_v) and the pressure rise due to combustion (Δp_c), as shown below:

$$\Delta p = \Delta p_c + \Delta p_v \tag{30}$$

The term Δp_v represents the pressure change that would occur if the engine undergoes a polytropic compression or expansion without combustion (motoring). Despite ignoring heat transfer and leakages, this component is modeled using a polytropic process $pV^m = \textbf{\textit{const}}$. For a step from initial instant i to final instant j, the pressure change due to volume is calculated as:

$$\Delta p_v = p_j - p_i = p_i \left[\left(\frac{V_i}{V_j} \right)^m - 1 \right] \tag{31}$$

Consequently, the pressure increase solely due to combustion is derived as follows:

$$\Delta p_c = \Delta p - \Delta p_v \tag{32}$$

Furthermore, the method assumes that the mass of burned fuel ($m_{f,b}$) is directly proportional to the increase in pressure due to combustion ($m_{f,b} \propto \Delta p_c$). Therefore, the mass fraction burned (x_b) in the i-th interval is given by the ratio of the cumulative pressure increase up to that point to the total pressure increase over the entire combustion duration (N intervals):

$$x_b(\theta_i) = \frac{m_{f,b}(\theta_i)}{m_{f,inj,TOT}} = \frac{\sum_0^i \Delta p_c}{\sum_0^N \Delta p_c} \tag{33}$$

A critical aspect of this method is the selection of the polytropic index m. Although m varies during the cycle, it is typically recommended to have a constant value in this range $[1.30, 1.35]$ for CI engines to fit the experimental data.

5.3 Single-Zone Heat Release Rate (HRR) Analysis

A more physically comprehensive approach is the *single-zone model*, which applies the first law of thermodynamics to the combustion chamber region.

In this model, the cylinder charge is assumed to be a homogeneous ideal gas mixture (single-zone hypothesis). The control volume is treated as an open system to account for work, internal energy changes, and heat transfer, although mass flows (blow-by and injection) are often simplified in basic analyzes. The Net heat release rate ($\dot{Q}_n$), also referred to as apparent HRR, represents the difference between chemical heat release and heat losses in the wall ($\dot{Q}_{ch} - \dot{Q}_{ht}$). Using the differential form with respect to the crank angle θ, the energy balance equation is derived as follows:

$$\frac{dQ_n}{d\theta} = \frac{\gamma}{\gamma - 1} p \frac{dV}{d\theta} + \frac{1}{\gamma - 1} V \frac{dp}{d\theta} \tag{34}$$

where:

- p and V are the instantaneous cylinder pressure and volume.
- γ is the specific heat ratio (c_p/c_v), which is temperature-dependent.

Finally, the profile of mass fraction burned $x_b(\theta)$ is obtained by integrating the heat release rate and normalizing it by the total fuel energy introduced:

$$x_b(\theta) = \frac{\int_{SOI}^{\theta} HRR \, d\theta}{m_{f,inj,TOT} \cdot Q_{LHV}} \tag{35}$$

where Q_{LHV} is the Lower Heating Value of the fuel. This formulation allows for a direct evaluation of combustion phasing parameters such as *MFB10*, *MFB50*, or *MFB90* as requested.

5.4 Computational Analysis

In this section, the MATLAB-based post-processing procedure is described. The analysis focuses on deriving the x_b profile and identifying critical timings such as *the SOC*, *SOI*, *ID*, and *MFB percentiles*. Furthermore, the calculated parameters are presented graphically, analyzed, and then discussed.

5.5 In-Cylinder pressures comparison

First, the experimental data sets were retrieved from the reference table (*T*), focusing on the operating point of the engine at *2000 rpm* (the seventh row). The motored pressure trace was computed iteratively, as shown in Figure 35. The calculation was initialized at 350° CA by setting the motored pressure equal to the filtered in-cylinder pressure (*p_cyl_butter_filtfilt*) at that specific crank angle and the polytropic exponent *m* was set to 1.3.

```
delta_p = zeros(size(theta_DSF));
delta_pV = zeros(size(theta_DSF));
p_motored = p_cyl_butter_filtfilt;
m = 1.3;
idx = 3500;
V_ref= V_x(idx);
p_ref = p_cyl_butter_filtfilt(idx);

for h= 3500:7199
    delta_p(h) = p_cyl_butter_filtfilt(h+1) - p_cyl_butter_filtfilt(h);
    delta_pV(h) = p_cyl_butter_filtfilt(h)*((V_ref/V_x(h+1))^m -1);
    p_motored(h+1) = p_ref*((V_ref/V_x(h+1))^m);
end
```

Figure 35: *Motored pressure (p_motored) and delta pressures computations.*

In the plot below, the measured in-cylinder pressure (*pCYLINDER*) was compared with the calculated motored pressure (*pMOTORED*):

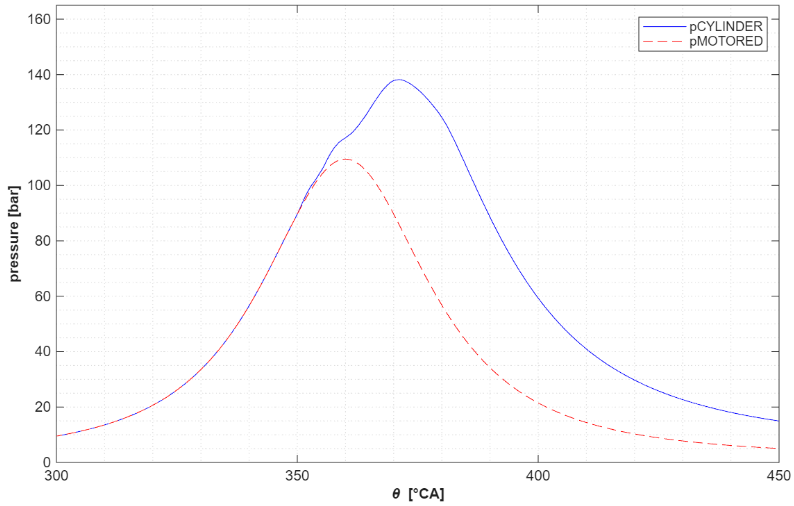

Figure 36: *Comparison between measured in-cylinder pressure and motored pressure.*

After that, the thermodynamic parameters and the net HRR were computed.

5.6 Thermodynamic parameters and net HRR

To compute the net HRR as defined in Equation (34), it was necessary first to determine the specific heat ratio γ as shown in Figure 37.

First, as shown in the equations, the temperature throughout the cycle was calculated according to the law of the ideal gas state, where the mass of the mixture (*m_mix*) and the gas constant of the mixture (*R_mix*) were defined as shown in the first two rows of the script in Figure 37 using data extracted from Table *T* for the operating point of *2000 rpm*.

$$T = \frac{pV}{m_{mix} R_{mix}} \tag{36}$$

$$\gamma(T) = 1.338 - 6 \cdot 10^{-5} \cdot T + 1 \cdot 10^{-8} \cdot T^2 \tag{37}$$

For the calculation of γ, Equation (37) was applied in MATLAB.

```
% Thermodynamic parameters
m_mix = T.m_air(7)+T.m_EGR(7);
R_mix = T.R_mix(7);
temperature = (p_cyl_butter_filtfilt .* V_x'*10^-1) ./ (m_mix * R_mix);   % [K]
gamma = 1.338 -6 * 10^(-5).*temperature + 10^(-8).*temperature.^2;
```

Figure 37: *Thermodynamic parameters.*

Once γ was defined, the Net Heat Release Rate (HRR) was calculated for each crank angle θ, as shown in Figure 38, according to Equation (34).

```
% compute net HRR
HRR = ones(length(theta),1);
for j = 2:length(theta)
    HRR(j) = 1.05*gamma(j)/ (gamma(j)-1) * p_cyl_butter_filtfilt(j) * ...
            (V_x(j)-V_x(j-1))+1/(gamma(j)-1)* V_x(j)*(p_cyl_butter_filtfilt(j)-p_cyl_butter_filtfilt(j-1));
end
```

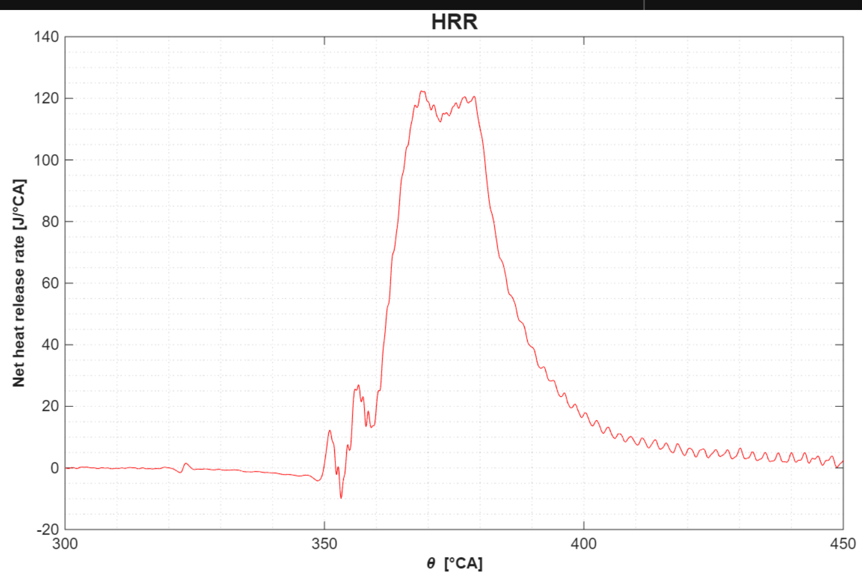

Figure 38: *The net Heat Release Rate computation (upper) and its plot (lower).*

5.7 Mass Fraction Burned and Combustion Indices

Once the Heat Release Rate profile was obtained, the mass fraction burned profile was calculated as shown in Equation (35) using numerical integration called *trapezoidal method* (`cumtrapz` function in MATLAB). Integration was performed within a specific crank angle window starting from the Start of Injection (SOI) up to the End of Combustion (EOC, which was supposed to be 450 °CA). The SOI was determined on the basis of the injector energizing current. It is identified as the CA where the mean current signal first exceeds a threshold of *1 Ampère*.

```
% Window indices
EOC = 450;
theta_deg= theta_deg';
% Indices inside the integration window
SOI = find(inj>1,1,"first")/10;
idx = (theta_deg >= SOI) & (theta_deg <= EOC);
Q_lhv = 42.5 * 10^6;      %J/kg
theta_w = theta_deg(idx);
HRR_w   = HRR(idx);
% Compute Xb only in the combustion window
Xb = zeros(size(theta_deg));
Xb(idx) = cumtrapz(theta_w, HRR_w) / (T.m_fuel(7)* Q_lhv);
```

Figure 39: *Calculus of SOI and x_b.*

As simply depicted in Figure 40, the other parameters were estimated. In Figure 41 the results of the profile of the mass fraction burned (*x_b*) and the injection current were also plotted.

```
% MFB10/50/90 and SOC
MFB10 = theta_deg(find(Xb >= 0.10, 1));
MFB50 = theta_deg(find(Xb >= 0.50, 1));
MFB90 = theta_deg(find(Xb >= 0.90, 1));

% SOC: first point where XB>small threshold (ex: 0.02)
SOC_idx = find(Xb > 0.01, 1);
SOC = theta_deg(SOC_idx);
ID = SOC - SOI;      % ignition delay
```

Figure 40: *Calculus of SOC, ID and MFB percentiles.*

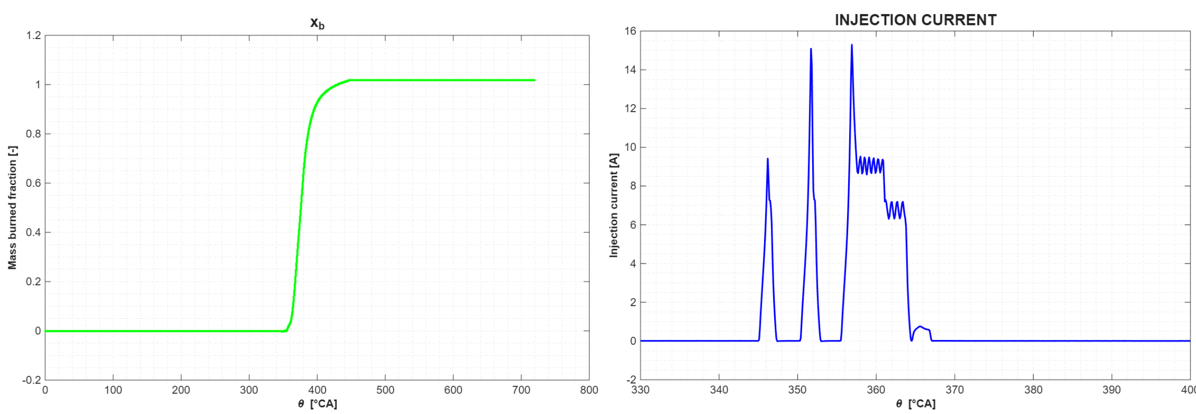

Figure 41: *The plots of* x_b *(left-side) and injection current (right-side).*

Finally, the key parameters identified from the analysis were listed below.

Parameter	Value [° CA]
Start of Injection (SOI)	345.40
Start of Combustion (SOC)	356.50
Ignition Delay (ID)	11.10
MFB10	364.50
MFB50	376.00
MFB90	395.60

Table 1: Combustion parameters derived from Heat Release Rate analysis.

5.8 Results and comments

The results obtained from the analysis are clearly summarized in the figure below. The relationship between the injection command and the heat release were illustrated in the upper-side in the Figure below (referencing the *HRR and INJ CURRENT* paths).

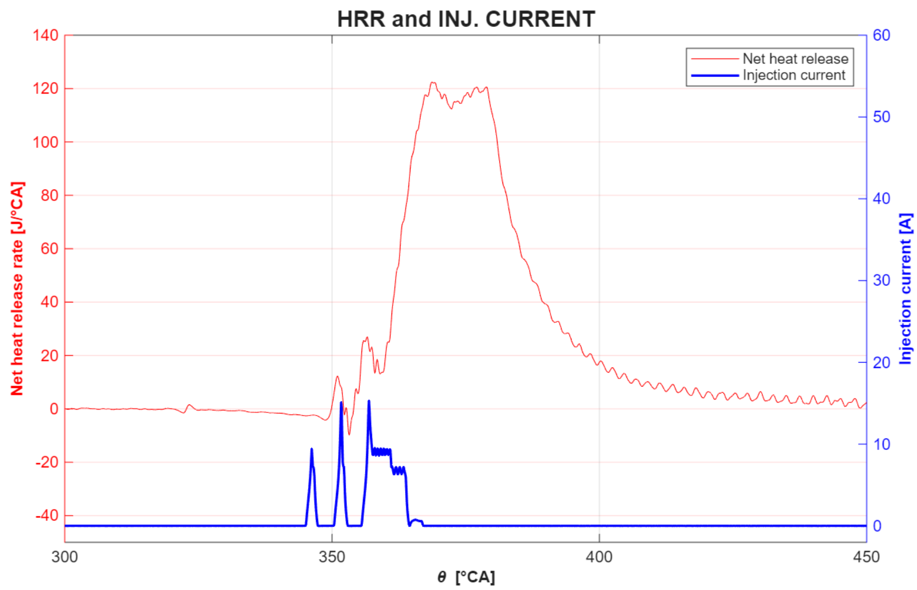

Figure 42: *Comparison between the -net Heat Release Rate and the Injection current.*

The blue path represents the injector energizing current. It clearly highlights a multiple-injection strategy consisting of a short **pilot injection** (starting around *345 °CA*) followed by the **main injection** pulse.The red trace shows the Net Heat Release Rate ($dQ_n/d\theta$). The subsequent rapid increase in HRR reaches a peak of approximately *120 J/° CA*, marking the premixed combustion phase.

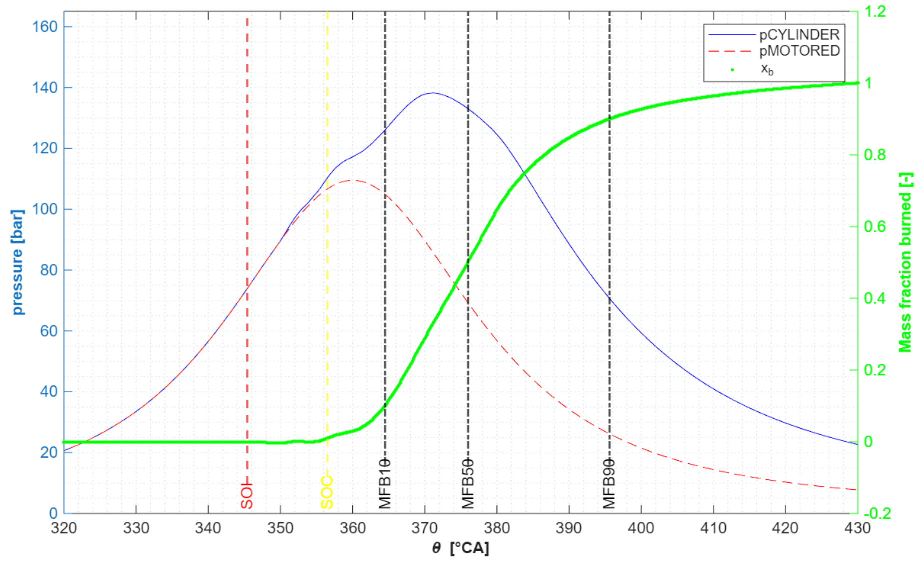

Figure 43: *Evolution of In-Cylinder Pressure and Mass Fraction Burned (x_b).*

Figure 43 provides a comprehensive view of the thermodynamic cycle and the progress of combustion. The graph compares the measured in-cylinder pressure (p_{cyl}, blue line) with the calculated motored pressure (p_{mot}, red dashed line). The deviation between the two curves starts at the Start of Combustion (SOC), confirming that the pressure rises due to heat release. The peak firing pressure reaches approximately *140 bar*, significantly higher than the motored peak. The green curve represents the cumulative Mass Fraction Burned (x_b).

To conclude, also vertical markers identify the key angular phases computed in the analysis:

- **Ignition Delay (ID):** Clearly visible as the interval between the Start of Injection (SOI, red dashed line) and the Start of Combustion (SOC, yellow dashed line).
- **Combustion Phasing:** The MFB50 point (where 50% of fuel is burned) occurs after the Top Dead Center, properly positioned to optimize the expansion work.
- **Combustion Duration:** The interval from SOC to MFB90 characterizes the overall duration of the combustion process, showing the typical S-curve profile that saturates at $x_b = 1$.

6 Conclusions

The computational analysis confirms the successful integration of experimental steady-state testing with advanced thermodynamic modeling. The application of the ISO 1585 correction ensured that performance trends remained within acceptable standard ranges throughout the engine speed range.

The thermodynamic analysis at the 2000 rpm full-load operating point yielded a Gross IMEP of 19.64 bar and a Net IMEP of 19.34 bar. The BMEP was calculated at 18.11 bar, resulting in a high mechanical efficiency of 92%.

Furthermore, the Fourier analysis demonstrated that using the first 10 harmonics provides a sufficiently accurate representation of the original pressure and torque trends.

For flywheel sizing, to maintain a kinematic irregularity below 0.01, the required flywheel diameter was determined to be 0.565 m for the single-cylinder engine and 0.471 m for the multi-cylinder configuration.

The HRR analysis identified a Start of Combustion (SOC) at 356.50° CA and an Ignition Delay (ID) of 11.10° CA.Phasing: The MFB50 point was correctly positioned after Top Dead Center to optimize expansion work, while the identified multiple-injection strategy was clearly reflected in the premixed combustion phase peaks.

Ultimately, these results provide a detailed map of the engine's mechanical efficiency and thermodynamic behavior, validating the MATLAB-based methodology for engine performance diagnostic applications.

7 MATLAB script

```
clc;
clear;
close all;
%% import data
file_name='datasheet_2025.xlsx';
sheet_name = 'Steady-state tests';
T=readtable(file_name, 'Sheet', sheet_name,'Range', 'A:J');
file_name='datasheet_2025.xlsx';
sheet_name = 'Engine data';
E=readtable(file_name, 'Sheet', sheet_name);
%% Initial parameters kPa and K
p0d=99;
T0d=298;
Q_lhv = 42.5; % MJ/kg
B = 95.8; % mm
S= 104; % mm
R_air = 287.05; % J/kgK
cp = 1038; % J/kgK
gamma = 1.4;
R_EGR = cp * (1 - 1/gamma);
i = 4;
% Calculate the displacement volume
V_d = (pi/4) * (B^2) * S;

% convert pressure from mbar to kPa
T.p_baro=T.p_baro./10;
T.p_i_MF=T.p_i_MF./10+101.325;
% convert temperature in K
T.T_snorkel = T.T_snorkel +273.15 ;
T.T_i_MF = T.T_i_MF +273.15;
% convert Q_lhv in kwh/kg
Q_lhv = Q_lhv * 0.2778;
%% Corrected power and torque (in kW and Nm)
T.qc=T.qm_fuel./T.p_i_MF.*T.p_baro;
T.fm= zeros(height(T),1);
T.fm(T.qc<37.2) = 0.2;
T.fm(T.qc >= 37.2 & T.qc<65) = 0.036 .*T.qc (T.qc>=37.2 & T.qc<= 65) - 1.14;
T.fm (T.qc>65)= 1.2;
T.fa= (p0d./T.p_baro).^0.7.*((T.T_snorkel)./T0d).^1.2;
T.mu_c = T.fa.^T.fm;
```

```
T.P = T.T_dyno .* 2 .* pi .* T.n_engine./60 /1000;
T.P_0 = T.P .* T.mu_c;
T.T_0 = T.T_dyno .* T.mu_c;

%% Fuel conversion efficiency
T.efficiency = T.P_0 ./ (Q_lhv * T.qm_fuel);

%% Brake specific fuel consumption in g/kWh
T.bsfc = 1 ./ (T.efficiency.* Q_lhv) .*1000;


%% Volumetric efficiency
T.m_air = T.qm_air *60 * 2 ./ (3600* i* T.n_engine);
T.m_EGR = T.qm_EGR *60 * 2 ./ (3600* i* T.n_engine);
T.m_fuel= T.qm_fuel *60 * 2 ./ (3600* i* T.n_engine);
T.R_mix = (T.qm_air.*R_air+ T.qm_EGR* R_EGR)./(T.qm_air+T.qm_EGR); % J/kgK
T.vol_eff = ((T.m_air+T.m_EGR)./V_d).*((T.R_mix.*T.T_i_MF)./T.p_i_MF);
T.vol_eff = T.vol_eff* 10^6;

%% Plot Full load parameters
figure
plot(T.n_engine, T.P, '-o', 'Color','r', 'LineWidth',1, 'MarkerSize',6); hold on;
plot(T.n_engine, T.P_0, '-o', 'Color','g', 'LineWidth',1, 'MarkerSize',6);
plot(T.n_engine, T.T_dyno, '-o', 'Color','b', 'LineWidth',1, 'MarkerSize',6);
plot(T.n_engine, T.T_0, '-o', 'Color','y', 'LineWidth',1, 'MarkerSize',6);
grid on
grid minor
box on
ylim([0,500])
xlabel('Engine speed (RPM)', 'FontSize', 14, 'FontWeight', 'bold')
ylabel('Full load characteristics', 'FontSize', 14, 'FontWeight', 'bold')
title('Full load characteristics over engine speed', 'FontSize', 20)
set(gca, 'FontSize', 12)
legend({'P_{dyno}','P_0','T_{dyno}','T_0'}, 'Location','best', 'FontSize',12);

%% Plot correction factor
figure
plot(T.n_engine, T.mu_c, '-o', 'LineWidth', 2, 'MarkerSize', 6, 'Color', [0 0.4470 0.7410])
grid on
grid minor
box on
```

```
ylim([0.99, 1.005])
xlabel('Engine speed (RPM)', 'FontSize', 14, 'FontWeight', 'bold')
ylabel('Correction factor', 'FontSize', 14, 'FontWeight', 'bold')
title('Correction factor over engine speed', 'FontSize', 20)
set(gca, 'FontSize', 12)

%% Plot fce
figure
plot(T.n_engine, T.efficiency, '-o', 'LineWidth', 2, 'MarkerSize', 6, 'Color', [0 0.4470 0.7410])
grid on
grid minor
box on
ylim([0.30, 0.45])
xlabel('Engine speed (RPM)', 'FontSize', 14, 'FontWeight', 'bold')
ylabel('Fuel conversion efficiency', 'FontSize', 14, 'FontWeight', 'bold')
title('Fuel conversion efficiency over engine speed', 'FontSize', 20)
set(gca, 'FontSize', 12)

%% Plot BSFC
figure
plot(T.n_engine, T.bsfc, '-o', 'LineWidth', 2, 'MarkerSize', 6, 'Color', [0 0.4470 0.7410])
grid on
grid minor
box on
ylim([200, 245])
xlabel('Engine speed (RPM)', 'FontSize', 14, 'FontWeight', 'bold')
ylabel('Brake specific fuel consumption', 'FontSize', 14, 'FontWeight', 'bold')
title('Brake specific fuel consumption over engine speed', 'FontSize', 20)
set(gca, 'FontSize', 12)

%% Plot volumetric efficiency
figure
plot(T.n_engine, T.vol_eff, '-o', 'LineWidth', 2, 'MarkerSize', 6, 'Color', [0 0.4470 0.7410])
grid on
grid minor
box on
ylim([0,1])
xlabel('Engine speed (RPM)', 'FontSize', 14, 'FontWeight', 'bold')
ylabel('Volumetric efficiency', 'FontSize', 14, 'FontWeight', 'bold')
title('Volumetric efficiency over engine speed', 'FontSize', 20)
```

```matlab
set(gca, 'FontSize', 12)

%% Load data
load('ifile_2000FL.mat');
%% Crank angle theta
theta_deg = linspace(0,719.9,7200);

%% Cylinder pressure: compute mean over first 100 cycles
p_raw  = double(ifile.PCYL1.data);
p_mean = mean(p_raw(:, 1:100), 2);

%% Manifold pressure: compute mean over first 100 cycles
p_man       = double(ifile.PMAN1.data);
p_man_mean  = mean(p_man(:, 1:100), 2);

%% Plot cylinder mean pressure
figure
plot(theta_deg, p_mean, 'LineWidth', 2)
grid on; grid minor; box on
xlabel('Crank angle', 'FontSize', 14, 'FontWeight', 'bold')
ylabel('Mean pressure', 'FontSize', 14, 'FontWeight', 'bold')
title('Mean cylinder pressure over crank angle', 'FontSize', 20)
set(gca, 'FontSize', 12)
legend({'P_{cyl}'}, 'Location','best', 'FontSize',12);


%% Compute pegged signal
ifile.PMAN1.axis(:,2) = p_mean;

%% Interval for reference (175-185 deg)
idx = (theta_deg > 175) & (theta_deg < 185);
p_mean_interval  = mean(ifile.PMAN1.axis(idx, 2));

% Global manifold mean
p_mean_manifold  = mean(p_man_mean);

% Offset
p_offset = p_mean_manifold - p_mean_interval;

% Pegged pressure
p_pegged = (+p_mean + p_offset);   % segno coerente

figure
```

```
plot(theta_deg, p_pegged, 'LineWidth', 2)
grid on; grid minor; box on
title('Pegged pressure')
xlabel('Crank angle')
ylabel('Pressure')
xlim([280 450])


%% FIR moving average filter
windowSize = 10;
b = (1/windowSize) * ones(1, windowSize);
a = 1;
p_filtered = filter(b, a, p_pegged);

%% Butterworth filter
enc_res = 0.1;
fs = 2000 ./ 60 .* 360 ./ enc_res;    % sampling frequency

fc = 4000;              % cutoff [Hz]
Wn = fc / (fs/2);       % normalized cutoff
n  = 2;                 % order

[b,a] = butter(n, Wn);

p_cyl_butter_filtfilt = filtfilt(b, a, p_pegged);     % zero-phase
p_cyl_butter_filt     = filter(b, a, p_pegged);       % causal

%% moving mean
p_moving = movmean(p_pegged, 10);

%% Plot comparison
figure(100)
plot(theta_deg, p_pegged,                'LineWidth', 2); hold on
plot(theta_deg, p_filtered,              'LineWidth', 2);
plot(theta_deg, p_cyl_butter_filtfilt,       'LineWidth', 2);
plot(theta_deg, p_moving,                'LineWidth', 2);

grid on; grid minor; box on
xlabel('Crank angle', 'FontSize', 14, 'FontWeight', 'bold')
ylabel('Pressure',      'FontSize', 14, 'FontWeight', 'bold')
title('Pressure filtering comparison', 'FontSize', 20)
set(gca, 'FontSize', 12)
xlim([366 378])
```

```
ylim([135 140])
legend({'Input Data', 'FIR Moving Avg', 'Butterworth', 'movmean'}, ...
       'Location','northeast', 'FontSize',12)

%% Volume calculation
A_piston = pi * (ifile.engine.bore)^2 /4;       % mm^2
r = 0.5 * ifile.engine.stroke;
l = ifile.engine.conrod_length;
theta = deg2rad(theta_deg);
lambda = r/l;
x = r.*((1 - cos(theta)) + 1./lambda*(1 - sqrt(1 - (lambda^2 .* sin(theta).^2))));
V_d = A_piston * ifile.engine.stroke * 10^-3 * i;
V_c = V_d / (ifile.engine.compression_ratio  -1)/i;  % Assuming V_c is the clearance volume
V_x = V_c + A_piston *10^(-3) .* x;



%%
% Plot p-theta and p-V
figure
plot (theta_deg, V_x,  'LineWidth', 2)
grid on; grid minor; box on
xlabel('Crank angle', 'FontSize', 14, 'FontWeight', 'bold')
ylabel('Volume [cm^3]',      'FontSize', 14, 'FontWeight', 'bold')
title('Volume over crank angle', 'FontSize', 20)
set(gca, 'FontSize', 12)
figure
plot (theta_deg, p_cyl_butter_filtfilt,  'LineWidth', 2)
grid on; grid minor; box on
xlabel('Crank angle', 'FontSize', 14, 'FontWeight', 'bold')
ylabel('Volume',      'FontSize', 14, 'FontWeight', 'bold')
title('Pressure over crank angle', 'FontSize', 20)
set(gca, 'FontSize', 12)

figure
plot (V_x, p_cyl_butter_filtfilt,  'LineWidth', 2)
grid on; grid minor; box on
xlabel('Volume', 'FontSize', 14, 'FontWeight', 'bold')
ylabel('Pressure',      'FontSize', 14, 'FontWeight', 'bold')
title('Gas pressure', 'FontSize', 20)
set(gca, 'FontSize', 12)

```

```
V_ratio = V_x ./ max(V_x);
figure
plot (V_ratio, p_cyl_butter_filtfilt,  'LineWidth', 2)
grid on; grid minor; box on
xlabel('Volume', 'FontSize', 14, 'FontWeight', 'bold')
ylabel('Pressure',      'FontSize', 14, 'FontWeight', 'bold')
title('Gas pressure', 'FontSize', 20)
set(gca, 'FontSize', 12)


figure
loglog(V_ratio,p_cyl_butter_filtfilt, 'LineWidth', 2)
grid on; grid minor; box on
xlabel('Volume ratio', 'FontSize', 14, 'FontWeight', 'bold')
ylabel('Pressure', 'FontSize', 14, 'FontWeight', 'bold')
title('Gas pressure (log-log)', 'FontSize', 20)
set(gca, 'FontSize', 12)


% calculate Bmep - Imep (bar)
% values added on V_x and p_pegged for computation of imep_net and gross
p_pegged = p_pegged';
V_x = [V_x V_x(1)];
p_pegged = [p_pegged p_pegged(1)];
A=size(V_x);
B=size(p_pegged);
%-----------------------------------------------------------------------------%
w_i = trapz(V_x,p_pegged);
imep_n =i* w_i / V_d;
% id = (theta_deg >= 1801) & (theta_deg <= 5401);
theta_g =(theta_deg(:,1801:5401));
theta_g_rad=deg2rad(theta_g);
x_gross = r.*((1 - cos(theta_g_rad)) + 1./lambda*(1 - sqrt(1 - (lambda^2 .*
    ↪ sin(theta_g_rad).^2))));
V_gross = V_c + A_piston .* x_gross *10^-3;
p_gross = p_pegged(:,1801:5401);      %id);
w_g = trapz(V_gross, p_gross);
imep_g = w_g / V_d*4;
bmep = 4*pi * T.T_0(7) /(V_d)* 10;
mec_eff = bmep/imep_g;
fprintf ('imep gross = %f\n',imep_g)
fprintf ('imep net = %f\n',imep_n)
fprintf ('Bmep = %f\n',bmep)
```

```
fprintf ('Mechanical efficiency = %f',mec_eff)

%% Part 2: FOURIER ANALYSIS

% downsampling from 7200 to 720
DSF = 10;
p_DSF=p_cyl_butter_filtfilt(1:DSF:end);
theta_DSF= theta_deg(1:DSF:end);

%% pressure FFT
len = length(p_DSF);
Y = fft(p_DSF/len);
N = 10;

% harmonic 0
c0 = abs(Y(1));

% preallocation
cn = zeros(N,1);
phi = zeros(N,1);

for n = 1:N
    cn(n) = 2*abs(Y(n+1));
    phi(n) = angle(Y(n+1)) + pi/2 ;
end

% Conversions
theta_rad = deg2rad(theta_DSF);
rpm = 2000;
omega = rpm * 2*pi/60;              % engine speed
t = theta_rad / omega;
p_fourier = c0 * ones(size(theta_rad));
m = 2 ;  % 4 strokes
for n = 1:N
    p_fourier = p_fourier + cn(n)*sin(n/m* omega .*t+ phi(n));
end
%% plots
figure
plot(theta_DSF, p_DSF, 'b', 'LineWidth',1.5); hold on
plot(theta_DSF, p_fourier, 'r--', 'LineWidth',1.5);
xlabel('Theta [deg]')
ylabel('pressure [bar]')
title('Fourier approximation of the experimental pressure k=10')
```

```
legend('experimental','Fourier approx')
xlim([0 720])
grid on

%%
V_d = V_d * 10^-6 /4;
A_piston = A_piston *10^(-6);  % in m^2
r = r * 10^-3; % in m
m_pis = 1.46 ;  %Piston mass
m_rod = 1.40  ; %Conrod mass
m_crk = 0.85 ;  %Crankpin mass
m_rec = m_pis + 0.38 *m_rod;  %Recipr. masses
m_rot = m_crk + 0.62 *m_rod;     %Rotating masses
pc=1.01325*ones(size(theta_DSF))'; %Crankcase pressure
cos_beta = sqrt(1-lambda^(2)*(sin(theta_rad)).^2); % Calculate the angle beta
F_i = -m_rec * omega ^2 * r * (cos(theta_rad)+ lambda * cos(2.* theta_rad) /
    ↪ cos_beta);
bet_rad = asin(lambda * sin(theta_rad));  % Calculate the angle beta in radians
p_i = F_i / (A_piston)/10^5;  % bar
p_eff = p_DSF + p_i'-pc;   % bar
P = p_eff * A_piston *10^5 ; % N
F = P' ./ cos (bet_rad);                 % force along conrod
Ft = F .* sin(bet_rad + theta_rad);      % force tangent to the shaft
Ms = Ft * r ;
Pt = Ms ./ ((V_d)/2)  /10^5 ;  % bar

figure(20)
plot (theta_DSF, p_eff ,'LineWidth',1.5)
xlabel('Crank angle [deg]', 'FontSize', 14, 'FontWeight', 'bold')
ylabel('Effective Pressure [bar]', 'FontSize', 14, 'FontWeight', 'bold')
title('Effective Pressure over Crank Angle', 'FontSize', 20)
set(gca, 'FontSize', 12)
xlim ([0 720])
grid on; grid minor; box on

figure
plot (theta_DSF, Pt,  'LineWidth',1.5)
xlabel('Crank angle [deg]', 'FontSize', 14, 'FontWeight', 'bold')
ylabel('Tangential Pressure [bar]', 'FontSize', 14, 'FontWeight', 'bold')
title('Tangential Pressure over Crank Angle', 'FontSize', 20)
set(gca, 'FontSize', 12)
xlim ([0 720])
grid on; grid minor; box on
```

```
%% FFT torque
len_T = length(Ms);
Y = fft(Ms/len_T);
N = 10;

% harmonic 0
c0_T = abs(Y(1));

% preallocation
ncols = 2;
nrows = ceil(N/ncols);
cn_T = zeros(N,1);
phi = zeros(N,1);
T_fourier = c0_T * ones(size(theta_rad));
c0_T = c0_T * ones(size(theta_DSF));

% Plots
figure
subplot (nrows,ncols,1)
plot (theta_DSF, c0_T)
title('Harmonic order 0', 'FontSize', 10)
xlim([0 720])
grid on
for n = 1:N
    cn_T(n) = 2*abs(Y(n+1));
    phi_T(n) = angle(Y(n+1)) + pi/2 ;
    subplot (nrows+1,ncols,n+1)
    hold on
    harm = cn_T(n)*sin(n/m* omega .*t+ phi_T(n));
    T_fourier = T_fourier +  harm;
    plot (theta_DSF, harm)
    title(sprintf('Harmonic order %g', (n)/2), 'FontSize', 10)
    xlim([0 720])
    grid on
end
% avg torque
T_avgFourier = mean(T_fourier)*ones(size(theta_DSF));
%% torque
figure
plot(theta_DSF, Ms, 'b', 'LineWidth',1.5); hold on
plot(theta_DSF, T_fourier, 'r--', 'LineWidth',1.5);
plot(theta_DSF, T_avgFourier, 'g--', 'LineWidth',1.5);
```

```
xlabel('Crank angle [deg]', 'FontSize', 14, 'FontWeight', 'bold')
ylabel('Torque [Nm]', 'FontSize', 14, 'FontWeight', 'bold')
title('Torque for the single-cylinder k=10', 'FontSize', 20)
set(gca, 'FontSize', 12)
legend('Instantaneous torque', 'Fourier approximation', 'Average torque')
xlim ([0 720])
grid on; grid minor; box on

%% Stem
figure
cn_T = cn_T(:).';
cn_T = [c0_T(1), cn_T(1:end)];
harmonic_order = (0 : N)/2;
stem ( harmonic_order, cn_T)
xlabel('Harmonic order k', 'FontSize', 14, 'FontWeight', 'bold')
ylabel('|Y|', 'FontSize', 14, 'FontWeight', 'bold')
title('Amplitude spectrum (torque) for the single cylinder', 'FontSize', 20)
grid on
%% Torque of other cylinders
Ms_3 = circshift(Ms, 180);
figure
plot(theta_DSF, Ms_3, 'b', 'LineWidth',1.5); hold on
xlabel('Crank angle [deg]', 'FontSize', 14, 'FontWeight', 'bold')
ylabel('Torque [Nm]', 'FontSize', 14, 'FontWeight', 'bold')
title('Torque for the cylinder 3', 'FontSize', 20)
set(gca, 'FontSize', 12)
xlim ([0 720])
grid on; grid minor; box on
% 3rd
Ms_4 = circshift(Ms_3, 180);
figure
plot(theta_DSF, Ms_4, 'b', 'LineWidth',1.5); hold on
xlabel('Crank angle [deg]', 'FontSize', 14, 'FontWeight', 'bold')
ylabel('Torque [Nm]', 'FontSize', 14, 'FontWeight', 'bold')
title('Torque for the cylinder 4', 'FontSize', 20)
set(gca, 'FontSize', 12)
xlim ([0 720])
grid on; grid minor; box on
% 4 th
Ms_2 = circshift(Ms_4, 180);
figure
plot(theta_DSF, Ms_2, 'b', 'LineWidth',1.5); hold on
xlabel('Crank angle [deg]', 'FontSize', 14, 'FontWeight', 'bold')
```

```
ylabel('Torque [Nm]', 'FontSize', 14, 'FontWeight', 'bold')
title('Torque for the cylinder', 'FontSize', 20)
set(gca, 'FontSize', 12)
xlim ([0 720])
grid on; grid minor; box on


%% Multi cylinder torque analysis
T_4cyl = Ms + Ms_2 + Ms_3 + Ms_4; % Total torque of multi Cylinder engine
T_avg = mean(T_4cyl);
figure
plot(theta_DSF, T_4cyl, 'b', 'LineWidth',1.5); hold on
plot(theta_DSF, T_avg *ones(size(theta_DSF)),'r', 'LineWidth',1.5); hold on
xlabel('Crank angle [deg]', 'FontSize', 14, 'FontWeight', 'bold')
ylabel('Torque [Nm]', 'FontSize', 14, 'FontWeight', 'bold')
title('Torque for the multi-cylinder engine', 'FontSize', 20)
set(gca, 'FontSize', 12)
xlim([0 720])
legend('Instantaneous torque', 'Average torque' )
grid on; grid minor; box on


% FFT torque multi cylinder
T_4 = zeros(720,4);
T_4(:,1) = Ms;
T_4(:,2) = Ms_2;
T_4(:,3) = Ms_3;
T_4(:,4) = Ms_4;

Y_T4 = zeros(720,4);
for j = 1:4
    Y_T4(:,j) = fft(T_4(:,j))/720;
end

N = 10;

% 0 Components
c0_T4 = abs(Y_T4(1,:));

% Output Fourier
T_fourier4 = zeros(720,4);
for j = 1:4
    T_fourier4(:,j) = c0_T4(j) * ones(720,1);
```

```
end
ncols = 2;
nrows = ceil(N/ncols);
fig_phasor = figure('Units','pixels','Position',[100 100 1200 800]);
hold on
axis equal
grid on
title('Phasor diagrams')

fig_harm = figure('Units','pixels','Position',[100 100 1200 800]);
hold on
grid on
title('Harmonic contributions','FontSize', 12)
xlim([0 720])

cn_T4  = zeros(N,4);
phi_T4 = zeros(N,4);
harm_T4 = zeros(720,4);
for n = 1:N
    figure(fig_phasor)
    subplot(nrows, ncols, n)
    hold on
    title(sprintf('order %g',n/2))
    axis equal
    for j = 1:4

        cn_T4(n,j)  = 2 * abs(Y_T4(n+1,j));
        phi_T4(n,j) = angle(Y_T4(n+1,j)) + pi/2;
        harm_T4(:,j) = cn_T4(n,j) * sin(n/m * omega .* t + phi_T4(n,j));
        T_fourier4(:,j) = T_fourier4(:,j) + harm_T4(:,j);

        x  = [0, cn_T4(n,j)*cos(phi_T4(n,j))];
        y  = [0, cn_T4(n,j)*sin(phi_T4(n,j))];

        ang = 0:0.0001:2*pi;
        xp  = cn_T4(n,j) * cos(ang);
        yp  = cn_T4(n,j) * sin(ang);

        plot(xp,yp); hold on;
        plot(x,y); grid on;

    end
    figure(fig_harm)
```

```
    subplot(nrows,ncols,n)
    hold on
    title(sprintf('Harmonic order %g', n/2), 'FontSize', 12)
    xlabel('Crank angle [deg]')
    ylabel('Torque [Nm')
    grid on
    for k = 1:4
        plot (theta_DSF, harm_T4 (:,k))
        xlim([0 720])
    end
end

%% Multi cylinder
T_multi = Ms + Ms_2 + Ms_3 + Ms_4;
% FFT torque
len_T = length(T_multi);
Y_multi = fft(T_multi/len_T);
N = 10;

% Calculate the average torque for the multi-cylinder engine
T_avg_multi = mean(T_multi)*ones(size(theta_DSF));

% harmonic 0
c0_Tmulti = abs(Y_multi(1));

% preallocation
cn_T_multi = zeros(N,1);
phi_multi = zeros(N,1);
T_fourier_multi = c0_Tmulti * ones(size(theta_rad));
for n = 1:N
    cn_T_multi(n) = 2*abs(Y_multi(n+1));
    phi_multi(n) = angle(Y_multi(n+1)) + pi/2 ;
    harm_multi = cn_T_multi(n)*sin(n/m* omega .*t+ phi_T(n));
    T_fourier_multi = T_fourier_multi +  harm_multi;
end
%% torque
figure
plot(theta_DSF, T_multi, 'b', 'LineWidth',1.5); hold on
plot(theta_DSF, T_fourier_multi, 'r--', 'LineWidth',1.5); hold on
plot(theta_DSF, T_avg_multi, 'm', 'LineWidth',1.5);
xlabel('Crank angle', 'FontSize', 14, 'FontWeight', 'bold')
ylabel('Torque', 'FontSize', 14, 'FontWeight', 'bold')
title('Torque for the multi cylinder N=10', 'FontSize', 20)
```

```
set(gca, 'FontSize', 12)
legend('Instantaneous torque', 'Fourier approximation', 'Average torque')
xlim([0 720])
grid on; grid minor; box on

%% Stem
figure
cn_T_multi = cn_T_multi(:).';
cn_T_multi = [c0_Tmulti, cn_T_multi(1:end)];
stem ( harmonic_order, cn_T_multi)
xlabel('Harmonic order k', 'FontSize', 14, 'FontWeight', 'bold')
ylabel('|Y|', 'FontSize', 14, 'FontWeight', 'bold')
title('Amplitude spectrum (torque) for the multi cylinder', 'FontSize', 20)
grid on

%% REPORT 3 FLYWHEEL DIMENSIONING
p_DSF=circshift(p_DSF,360);
% Shift Pressure
Pt=circshift(Pt,360);
% Shift Torque and Theta
Ms_shifted = circshift(Ms, 360); % Shift left to start at expansion

figure
plot(theta_DSF,p_DSF)
title('gas pressure')
xlabel('theta [deg]')
ylabel('gas pressure [bar]')
grid on, zoom on
figure
plot(theta_DSF,Pt)
title('tangential pressure')
xlabel('theta [deg]')
ylabel('tangential pressure [bar]')
grid on, zoom on
rho_met= 7700; %value of density of the flywheel
m_alt= m_pis+m_rod*0.38; %reciprocating mass [kg]
m_rot= m_crk+m_rod*0.62; % equivalent rotating masses []kg/dm^3]
w_max= 3850*2*pi/60; %max angular speed
w_min= 1400*2*pi/60; %min angular speed
w_avg= 2000*2*pi/60; % average angular speed
delta= ((w_max-w_min)/w_avg)/100; %Kinematic irregularity
if delta>0.01
    delta=0.01;
```

```
end
V_xDSF=(V_x(1:DSF:end))';
V_xDSF = V_xDSF(1:720);
% 2. Calculate Resistant Torque (Mr)
% For steady state, Mr is constant and equals the average Shaft Torque
Mr = mean(Ms);
Mr = Mr * ones(size(Ms));
delta_W=ones(size(theta_rad));

% calculate works
Ws = cumtrapz(theta_rad,Ms_shifted);
Wr = cumtrapz(theta_rad, Mr); % Calculate work done for each angle
delta_W=Ws-Wr;

% find min and max work variation
delta_Wmax=max(delta_W);
delta_Wmin=min(delta_W);
csi=(delta_Wmax+abs(delta_Wmin))/(imep_n*V_d);


J_eng= m_rot*(r)^2;
J_tot=csi*imep_n*V_d/(delta*w_avg^2);
J_flyw=J_tot-J_eng;
D_flyw=(320*J_flyw/(pi*rho_met))^0.2;
w_flyw= (1/10)*D_flyw;
fprintf('Required Flywheel Inertia: %.4f kg m^2\n', J_flyw);
fprintf('Calculated Flywheel Diameter: %.4f m\n', D_flyw);

%Table

tr=1.708856*ones(size(theta_DSF));
V_x=V_x(1:end-1);
p_filtered=p_filtered(1:DSF:end);
theta_DSF = theta_DSF';
bet_deg = (rad2deg(bet_rad))';
p_i = p_i';
Ms = Ms';
Mr = Mr';
Pt = Pt';
tr = tr';
Ws = Ws';
Wr = Wr';
delta_W = delta_W';
```

```
p_resistance = (Mr / (V_d) / 10^5);
% Convert to bar for plotting (divide by 10^5)
Ws_norm = (Ws ./ V_d) / 10^5;
Wr_norm = (Wr ./ V_d) / 10^5;
DeltaW_norm = (delta_W ./ V_d) / 10^5;
%%
Q = table( theta_DSF, bet_deg, V_xDSF, p_filtered, pc, p_i, p_eff, Ms,Pt,Mr,
    ↪ tr,   Ws, Wr, p_resistance);
%%
figure
plot (theta_DSF,Pt, 'b', 'LineWidth',1.5)
hold on
plot (theta_DSF,Ws_norm, 'g', 'LineWidth',1.5)
hold on
plot (theta_DSF,Wr_norm, 'r', 'LineWidth',1.5)
hold on
plot (theta_DSF,p_resistance, 'y', 'LineWidth',1.5)
hold on
legend('Tangential Press. (Pt)', 'Shaft Work (Ws)', 'Resistant Work (Wr)',
    ↪ 'Resistant Press. (Pr)');
xlabel('Crank Angle [deg]');
ylabel('Normalized Magnitude [bar]');
title('Flywheel Dimensioning: Work and Pressure');
xlim([0 720])
grid on
% (Starting Speed) in rad/s
%  w_avg as initial speed
w_start_rpm = 2000;
w_start_rad = w_start_rpm * (2*pi/60);

% 2. speed (rad/s)
% NO FLYWHEEL
omega_nofly_rad = sqrt(w_start_rad^2 + (2/J_eng) .* delta_W);

% WITH FLYWHEEL
omega_fly_rad   = sqrt(w_start_rad^2 + (2/J_tot) .* delta_W);

% 3. in RPM
omega_nofly_rpm = omega_nofly_rad * (60/(2*pi));
omega_fly_rpm   = omega_fly_rad   * (60/(2*pi));

% mean values
mean_nofly = mean(omega_nofly_rpm);
```

```
mean_fly    = mean(omega_fly_rpm);

%% PLOT
figure
% PLOT 1: NO FLYWHEEL
plot(theta_DSF, omega_nofly_rpm, 'b', 'LineWidth', 2); hold on;
% Starting speed
yline(w_start_rpm, 'r--', 'LineWidth', 2);
% Average speed
yline(mean_nofly, 'g:', 'LineWidth', 2);

title('(no flywheel)', 'FontSize', 14, 'Color', 'b');
subtitle('speed');
ylabel('angular speed [rpm]', 'FontSize', 10, 'FontWeight', 'bold');
xlabel('\theta [deg]', 'FontSize', 10, 'FontWeight', 'bold');
legend('Instantaneous shaft speed', 'starting speed', 'average speed', ...
       'Location', 'best');
grid on; grid minor; box on;
xlim([0 720]);
ylim([1000 10000]);

% --- PLOT 2: WITH FLYWHEEL ---
figure

plot(theta_DSF, omega_fly_rpm, 'b', 'LineWidth', 2); hold on;
% Starting speed
yline(w_start_rpm, 'g:', 'LineWidth', 2);
% Average speed
yline(mean_fly, 'r--', 'LineWidth', 2);

title('(with flywheel)', 'FontSize', 14, 'Color', 'b');
subtitle('speed');
ylabel('angular speed [rpm]', 'FontSize', 10, 'FontWeight', 'bold');
xlabel('\theta [deg]', 'FontSize', 10, 'FontWeight', 'bold');
legend('Instantaneous shaft speed', 'starting speed', 'average speed', ...
       'Location', 'best');
grid on; grid minor; box on;
xlim([0 720]);
ylim([1990 2030]);


%  T_avg is the average torque

```

```
% T_4cyl is the total torque

Ws_multi = cumtrapz(theta_rad, T_4cyl);
Wr_multi = cumtrapz(theta_rad, T_avg*ones(size(theta_rad)));
DeltaW_multi = Ws_multi - Wr_multi;
csi_multi = max(DeltaW_multi) - min(DeltaW_multi);
J_tot_multi = csi_multi / ((w_avg)^2 * delta);
D_fly_multi = ((J_tot_multi * 320) / (pi * rho_met))^(1/5);
fprintf('Required Flywheel Inertia multi-cylinder: %.4f kg m^2\n', J_tot_multi);
fprintf('Calculated Flywheel Diameter multi-cylinder: %.4f m\n', D_fly_multi);
pt_4cyl=T_4cyl/V_d/2 *10^(-5) *i;
pt_2=Ms_2/V_d/2 *10^(-5) *i;
pt_3=Ms_3/V_d/2 *10^(-5) *i;
pt_4=Ms_4/V_d/2 *10^(-5) *i;
figure
plot(theta_DSF,pt_4cyl,'k','LineWidth',3)
hold on
plot(theta_DSF,Pt,'LineWidth',1)
hold on
plot(theta_DSF,pt_2,'LineWidth',1)
hold on
plot(theta_DSF,pt_3,'LineWidth',1)
hold on
plot(theta_DSF,pt_4,'LineWidth',1)
hold on
grid on
zoom on
legend('Engine','1','2','3','4')

%% Plot Multi-cyl

%omega flywheel multi-cyl
omega_flyMulti_rad    = sqrt(w_start_rad^2 + (2/J_tot_multi) .* DeltaW_multi);
mean_flyMulti    = mean(omega_flyMulti_rad);

figure
plot(theta_DSF, omega_flyMulti_rad, 'b', 'LineWidth', 2); hold on;
% Starting speed
yline(w_start_rad, 'g:', 'LineWidth', 2);
% Average speed
yline(mean_flyMulti, 'r--', 'LineWidth', 2);

```

```
title('Multi-cyl flywheel', 'FontSize', 14, 'Color', 'b');
subtitle('speed');
ylabel('angular speed [rad]', 'FontSize', 10, 'FontWeight', 'bold');
xlabel('\theta [deg]', 'FontSize', 10, 'FontWeight', 'bold');
legend('Instantaneous shaft speed', 'starting speed', 'average speed', ...
       'Location', 'best');
grid on; grid minor; box on;
xlim([0 180]);
% INJection
inj = mean(ifile.INJ1.data, 2);

% delta Work
[max_deltay,index_max] = max(delta_W);
max_deltax =  theta_DSF (index_max);
figure
plot(theta_DSF, delta_W)
title('Delta work', 'FontSize', 14, 'Color', 'b');
subtitle('speed');
ylabel('Delta work', 'FontSize', 10, 'FontWeight', 'bold');
xlabel('\theta [deg]', 'FontSize', 10, 'FontWeight', 'bold');
grid on; grid minor; box on;
hold on
plot(max_deltax,max_deltay,'ro','MarkerSize',10,'LineWidth',2)

%% Heat Release - part 4

% Start of injection
SOI = find(inj>1,1,"first")/10;
delta_p = zeros(size(theta_DSF));
delta_pV = zeros(size(theta_DSF));
p_motored = p_cyl_butter_filtfilt;
m = 1.3;
idx = 3500;
V_ref= V_x(idx);
p_ref = p_cyl_butter_filtfilt(idx);

for h= 3500:7199
    delta_p(h) =p_cyl_butter_filtfilt(h+1) - p_cyl_butter_filtfilt(h);
    delta_pV(h) = p_cyl_butter_filtfilt(h)*((V_ref/V_x(h+1))^m -1);
    p_motored(h+1) = p_ref*((V_ref/V_x(h+1))^m);
end
mf_tot = 0;
delta_pc = delta_p - delta_pV;       % combustion pressure
```

```
figure(99)
yyaxis left
plot(theta_deg,p_cyl_butter_filtfilt,'b-'); hold on
plot(theta_deg,p_motored,'r--')
xlim ([320 430])
ylabel('Pressure [bar]', 'FontSize', 10, 'FontWeight', 'bold');
xlabel('\theta [deg]', 'FontSize', 10, 'FontWeight', 'bold');
yyaxis right


m_mix = T.m_air(7)+T.m_EGR(7);
R_mix = T.R_mix(7);
%R_mix = mean(T.R_mix);
temperature = (p_cyl_butter_filtfilt .* V_x'*10^-1) ./ (m_mix * R_mix);    % [K]
gamma = 1.338 -6 * 10^(-5).*temperature + 10^(-8).*temperature.^2;
% 4) derivative of p and V
dV_dtheta = gradient(V_x*10^-6, theta)';     % m^3 / deg
dp_dtheta = gradient(p_cyl_butter_filtfilt*10^5, theta);  % MPa / deg

% compute HRR

HRR=ones(length(theta),1);
for j= 2:length(theta)
    HRR(j) = 1.05*gamma(j)/ (gamma(j)-1) * p_cyl_butter_filtfilt(j)
    ↪ *(V_x(j)-V_x(j-1))+1/(gamma(j)-1)*
    ↪ V_x(j)*(p_cyl_butter_filtfilt(j)-p_cyl_butter_filtfilt(j-1));
end

% Window indices
EOC = 450;
theta_deg= theta_deg';
% Indices inside the integration window
idx = (theta_deg >= SOI) & (theta_deg <= EOC);
Q_lhv = 42.5 * 10^6;       %J/kg
theta_w = theta_deg(idx);
HRR_w   = HRR(idx);
% Compute Xb only in the combustion window
Xb = zeros(size(theta_deg));
Xb(idx) = cumtrapz(theta_w, HRR_w) / (T.m_fuel(7)* Q_lhv);
figure
plot(theta_deg,HRR,'r')
title('HRR', 'FontSize', 14, 'FontWeight', 'bold')
```

```
xlabel('Crank angle [deg]', 'FontSize', 10, 'FontWeight', 'bold')
ylabel('Net heat release rate', 'FontSize', 10, 'FontWeight', 'bold')
grid on
% 9) EMFB10/50/90 and SOC/SOI
MFB10 = theta_deg(find(Xb >= 0.10, 1));
MFB50 = theta_deg(find(Xb >= 0.50, 1));
MFB90 = theta_deg(find(Xb >= 0.90, 1));

% SOC: first point where XB>small threshold (ex: 0.02)
SOC_idx = find(Xb > 0.01, 1);
SOC = theta_deg(SOC_idx);
ID = SOC - SOI;      % ignition delay

fprintf('MFB10=%.2f deg, MFB50=%.2f deg, MFB90=%.2f deg, SOC=%.2f
    ↪ deg\n,SOI=%.2f deg\n,ID=%.2f deg\n', MFB10, MFB50, MFB90, SOC, SOI, ID);


% MFB10 line
figure(99)
plot(theta_deg,Xb,'g--')
xlim ([320 430])
ylabel('Mass burned fraction', 'FontSize', 10, 'FontWeight', 'bold');
xlabel('\theta [deg]', 'FontSize', 10, 'FontWeight', 'bold');
set(gca, 'YColor', 'g');
hold on, grid on, box on
% Plot injection data
xline(MFB10, 'k--', 'LineWidth', 1.2, ...
       'Label', 'MFB10', 'LabelVerticalAlignment', 'bottom', ...
       'LabelHorizontalAlignment', 'center');

% MFB50 line
xline(MFB50, 'm--', 'LineWidth', 1.2, ...
       'Label', 'MFB50', 'LabelVerticalAlignment', 'bottom', ...
       'LabelHorizontalAlignment', 'center');

% MFB90 line
xline(MFB90, 'c--', 'LineWidth', 1.2, ...
       'Label', 'MFB90', 'LabelVerticalAlignment', 'bottom', ...
       'LabelHorizontalAlignment', 'center');
% SOI line
xline(SOI, 'c--', 'LineWidth', 1.2, ...
       'Label', 'SOI', 'LabelVerticalAlignment', 'bottom', ...
       'LabelHorizontalAlignment', 'center');
```

```
% SOC line
xline(SOC, 'c--', 'LineWidth', 1.2, ...
       'Label', 'SOC', 'LabelVerticalAlignment', 'bottom', ...
       'LabelHorizontalAlignment', 'center');

figure
plot(theta_deg, inj, 'r:', 'LineWidth', 1.5);
xlabel('Crank angle', 'FontSize', 10, 'FontWeight', 'bold')
ylabel('Injection current', 'FontSize', 10, 'FontWeight', 'bold')
title('Injection current', 'FontSize', 14, 'FontWeight', 'bold')
xlim([320 430])
grid on
```