Implementation of EOQ and Lambert W function in 1-D Engine simulation model for optimizing fuel injection in GDI engine

Pablo Santirso, Stephen Samuel[1,]

Oxford Brookes University, School of Engineering, Computing and Mathematics, Wheatley Campus, Oxford, United Kingdom

Abstract

The aim of this study was to implement Economic Order Quantity method (EOQ) together with the Lambert W function in a 1-D engine simulation model in order to develop a fuel control strategy for a Gasoline direct injection (GDI) engine. Previous work of the co-author demonstrated the possibility of optimizing fuel injection quantity in GDI engine using the EOQ that is commonly used in supply chain of perishable products. This work extends the previous work and implements it in a 1-D, crank angle resolved, engine simulation model for the application of model based calibration process. The present work uses a validated engine simulation model, which is based on predictive combustion modelling approach, and couples the 1-D engine simulation model with SIMULINK to add the evaporation, wall- wetting and heat transfer models. It employs FORTRAN subroutines to modify the internal code of the 1-D simulation software in order to add crank angle resolved evaporation model. Finally, EOQ with Lambert W function was added to the model using MATLAB with special attention to the decimal control for the solution. This study demonstrated that EOQ and Lambert W functions together are a suitable method to develop fuel control strategy for a model based calibration procedure when implemented in crank angle resolved 1-D simulation model.

Keywords: Economic Order Quantity, Lambert W function, 1-D engine simulation model, Predictive Combustion modelling, Fuel injection,

Email address: s.samuel@brookes.ac.uk (Stephen Samuel)
[1]Corresponding Author

1. INTRODUCTION

One of the most important decisions taken by the engine design teams for optimizing the performance of the engine and for meeting the emission target is choosing appropriate injection strategy for gasoline injection engines. This, in turn, decides the strategy for developing fuel injection mapping of the engine and powertrain calibration process. Improving the efficiency with the Engine Control Unit (ECU) of the engine enables the designer to develop fast and relatively less expensive modifications, which are easily adaptable to different types of engines. Likewise, 1-D simulation modelling tools allow the designer to develop complex algorithms for improving the fuel consumption, numerically observe the behaviour of the engine and to find possible errors before developing the testing schemes for calibration optimization.

Historically, the fuel injector had been placed in the intake system of a gasoline engine. The quantity of the fuel was metered based on the desired air-to-fuel ratio in the cylinder considering wall-wetting and other losses. Later, this system evolved into a multi-point port fuel injection (MPFI) system where individual injectors were employed for each cylinder located immediately before the intake valve. This system could meter the quantity of the fuel more accurately than the single point injectors for all cylinders. Modern automotive engines have their injectors in the combustion chamber, where fuel metering is done more precisely than the MPFI system and there is more freedom to vary the air to-fuel-ratio for different operating conditions if required.

However, this additional flexibility offered by the direct injection system also brings additional challenges in relation to quantifying the amount of fuel that is not available for combustion processes. For example, in a single point throttle body injection system (TBI), the temperature of the air in the manifold upstream and downstream of the injection system will indicate the amount of fuel evaporated in the manifold. Therefore, it was possible to correlate the mixture temperature with the quantity of the fuel lost in the system for that cycle [1, 2, 3].

Similarly, for MPFI engines the parameters required for estimating the portion of the fuel available in the cylinder could be estimated using the transient wall wetting model [4, 5, 6, 7, 8, 9]. Additional measurements could be carried out for validating the model. The accuracy required for predicting the correction is very critical since the time available for fuel to evaporate and mix with the incoming air stream is not long enough when compared to TBI system. However, the temperature and pressure, mass flow rate of air in the manifold, parameters required for describing the fuel puddle could be measured or observed using various methods in order to model the transient fuel correction [4, 10, 9, 11, 5]. 3-D computational fluid dynamic models could be employed to visualise and also gain detailed insight into fuel puddle and evaporation dynamics [12, 13] at different operating temperatures. These studies could be validated using experimental observations [14, 15] such as Shadowgraph [13] since the physical location of interest is accessible for instrumentation.

However, in GDI engines the time available for fuel injection, evaporation and mixing is significantly lower than that of MPFI engine [5]. In addition, most of the variables used for estimating fuel quantity available in gaseous form for combustion and fuel lost during the injection and evaporation process change constantly as a function of crank position. The variables such as the charge temperature, fuel jet penetration distance, fuel velocity will change as a function of crank position. It is also difficult to use any direct measurement or observation for verifying these instantaneously [16, 17, 18, 19] Therefore, developing physical models, which will inherently reflect the main characteristics of the injection, wall wetting, evaporation and mixing, is paramount for predicting the amount of fuel available for combustion, which is directly linked to the torque output of the engine.

The concept of model based engine calibration is not a new field of study [30, 69]; however, it has received more attention recently for the above stated reasons. Some of the models rely heavily on experimental data and statistical inference, some rely on a semi-empirical model, [20, 21, 49] and some are based on experimental data and statistical inference for training a neural network based model [22]. The ultimate aim of these models is to predict the injection quantity for achieving the required torque output from the engine and emission levels, since this torque value determines the effort required to drive the vehicle at a desired vehicle speed. If the predicted quantity of the

fuel is different from the quantity of the fuel actually available for combustion at the start of the combustion process, it will have an adverse effect on vehicle drivability, overall emission levels and vehicle performance. These models should be able to predict the fuel quantity for steady-state as well as transient operating conditions [23, 20].

1-D simulation models to develop sufficient details for engine calibration is one of the approaches used by engine manufacturers to reduce the calibration time and generate calibration sheets for achieving required performance and emission control process [24] using software such as GT-Power. These types of simulations also use neural network for optimization using design of experiments done numerically [25].These types of models carry inherent limitations because they are not capable of capturing flow field characteristics in the engine combustion chamber and have only limited capability for capturing any physical process that is 3-D in nature. However, the use of 3-D computational model for developing engine calibration sheets has not been fully realised so far mainly because of the computational cost. Therefore, computational fluid dynamic (CFD) models are commonly used to study the individual process such as wall wetting in the fuel injection engines [12]. In addition, coupling a 1-D model with the 3-D model and using CFD for only the critical components and control volume area where 3-D details are required for capturing physical processes is being tried by various researchers [26, 27]. However, validating a 3-D model for a combustion chamber and flow field characteristics in the combustion chamber is a major challenge for production type engines.

Therefore, if detailed physics based models [28, 29] to capture the characteristics of fuel injection and spray formation, wall wetting and evaporation are chosen then it is possible either using 1-D or 3-D to estimate the fuel loss in the system and the amount of fuel available for combustion. Therefore, it is possible to correlate the injected fuel mass with the torque output of the engine provided a suitable crank angle resolved combustion model [30, 31] is chosen. The choice of the combustion model is also crucial for relating injected fuel quantity with the torque output of the engine. If a forced combustion model such as Wiebe function [32] based models are chosen then the energy release characteristics are already imposed for a given quantity of fuel and engine state. Any changes in engine temperature that will affect the wall wetting and evaporation characteristics of the fuel which determine

the quantity of the fuel available for combustion cannot be linked to output torque if Wiebe combustion model is used, unless a detailed map of Wiebe constants [33, 32] which include the corrections for these types of changes are used. However, if a suitable predictive combustion model is used which can take the flow field characteristics and chemical kinetics of the fuel into account then correlating the injected mass with the torque output will be more accurate provided the physics model of injection, wall wetting and evaporation are correctly coupled [14, 30, 34, 35, 34, 37, 38].

Hence, a model based calibration has been considered a viable option by various researchers for reducing the calibration time and to develop design of experiments for capturing essential experimental data in order to optimize the performance of the engines [39, 40, 41, 42, 43, 44]. Moreover, crank angle resolved engine models [31] with varying degree of complexities [45] in conjunction with statistical insight derived from experimental data and Artificial Neural network based models are getting more attention for vehicle calibration and optimization recently [37]. These are for predicting cycle-by-cycle indicated torque [46], detecting and estimating spark timing [47, 48] to employ fuel injection strategy [50, 51, 45, 52, 53, 54], for cold start control strategy [55] and for effective implementation of other controls such as knock and for monitoring the performance of catalytic converters.

Recently researchers are exploring the possibilities of combining various simulation packages to capture essential characteristic of the flow field and combustion but at the same time using the strength of different modelling approaches to reduce the run time significantly. For example, Rask *et al.* [24] have demonstrated the use of GT-Power, a 1-D gas dynamic engine modelling approach with experimental data for developing high fidelity model for engine calibration. Sika *et al.* [56] have tried to couple GT-Power with MATLAB Simulink for developing control strategies. Similarly, Omekanda *et al.* [28] have used 1-D thermal model of the engine using MATLAB Simulink and Python in order to develop thermal management control. Recently Turkson *et al.* [25] have attempted to couple GT-Power with MATLAB for developing optimization strategy.

However, the present study has taken completely different approach when compared to the existing optimization strategies available in the public domain. The previous work of the author [57] demonstrated that fuel injection

and optimization could be treated as a supply chain problem and therefore, the concepts such as Economic Order Quantity (EOQ) [73,59] commonly used in determining the cost in the supply chain discipline could be used for optimizing the injected quantity of the fuel in Gasoline direct injection engine. Recently, researchers have already demonstrated that borrowing concepts from thermodynamics for modelling inventory systems and from economics and industrial engineering for modelling thermodynamic systems can enable us to gain more insight for optimizing the performance of systems [70,71,72]. Therefore, in order to develop schemes for reducing the development time, especially for fuel injection and calibration process, the present study aims to implement EOQ based optimization in a 1-D crank angled resolved engine model.

2. SCHEME OF THE WORK

This study is a continuation of the research work of Ventura *et al.* [57] which demonstrated that EOQ and Lambert W function could be used for optimizing fuel injection process in GDI engine. It aims to implement EOQ and Lambert W function by developing sub-model that can be linked to crank angle resolved 1-D engine simulation software. The sub-models such as fuel spray model, spray evaporation models and wall wetting and heat transfer are developed in SIMULINK. A FORTRAN subroutine is used for interacting with 1-D simulation model in order to estimate crank angle resolved quantities. A script for the EOQ and a script to solve the Lambert W function are developed in MATLAB based on Disney *et al.* [58]. Finally, the results are correlated with the cylinder pressure data of the engine simulated and the EOQ function for different engine operating conditions.

3. ECONOMIC ORDER QUANTITY

EOQ model is a well known model in inventory management [73, 74] and it is used for estimating near-optimal order quantities. A brief summary of EOQ concept commonly used in supply chain management for optimizing the cost [59] is given here for the sake of completeness. This work uses EOQ principle for perishable products such as melons proposed by [58]. The model

minimises the cost of this supply chain using the following equation.

$$TC = \frac{KD}{Q} + DV - \frac{D}{Q\alpha}\left(pV\exp(-\beta t_j)\exp(\alpha t_r)\right)\left(1 - \exp\frac{\alpha Q}{P}\right) + cD + C_j \tag{1}$$

Where TC is the total cost for harvest period, K is the batch transfer time in currency, D is the total annual harvest, c is the scalar multiplier, Q the transfer batch size in cartons, V is the maximum value that a carton can achieve at time $t = 0$, p is the picking rate in cartons/hour, α is the deterioration rate of the product per hour outside the coolant storage, t_r the transfer time in hours from field to the cooling facility, β deterioration rate of the value in the cooling facility, t_j the time in the cooling facility and C_j the cost of transportation to the retailer. If the cost is minimized and the equation is adjusted:

$$Q = \left(\frac{p}{\alpha} - k\right)\exp(\frac{Q\alpha}{p}) - \frac{p}{\alpha} \tag{2}$$

$$k = \exp(\alpha t_r + \beta t_j).\frac{K}{V} \tag{3}$$

If the Lambert W function is applied in the equation, then:

$$Q^* = -\frac{p}{\alpha}\left[W_{-1}\left(\frac{k\alpha}{pe} - \frac{1}{e}\right) + 1\right] \tag{4}$$

The solution exists for equation (4) if p, K and α are positives and $V > \frac{\exp(\alpha t_r + \beta t_j)}{p} K\alpha$. In this study SIMULINK is used to estimate the solution of Lambert W function for the EOQ equation (4).

3.1. Analogy with Internal combustion engines

Ventura and Samuel [57] established an analogy between the supply chain of a perishable product and the fuel injection phenomenon in a GDI engine. The analogy proposed by Ventura and Samuel with additional details linked to the present work is shown in Figure 1.

The final output when the equations are solved is the decision variable Q*, it will be the mass of fuel that the injector has to reduce from the base fuel mass quantity. The final equation in the GDI engine will be:

$$Q^* = -\frac{p}{\alpha}\left[W_{-1}\left(\frac{k\alpha}{pe} - \frac{1}{e}\right) + 1\right] t_j \tag{5}$$

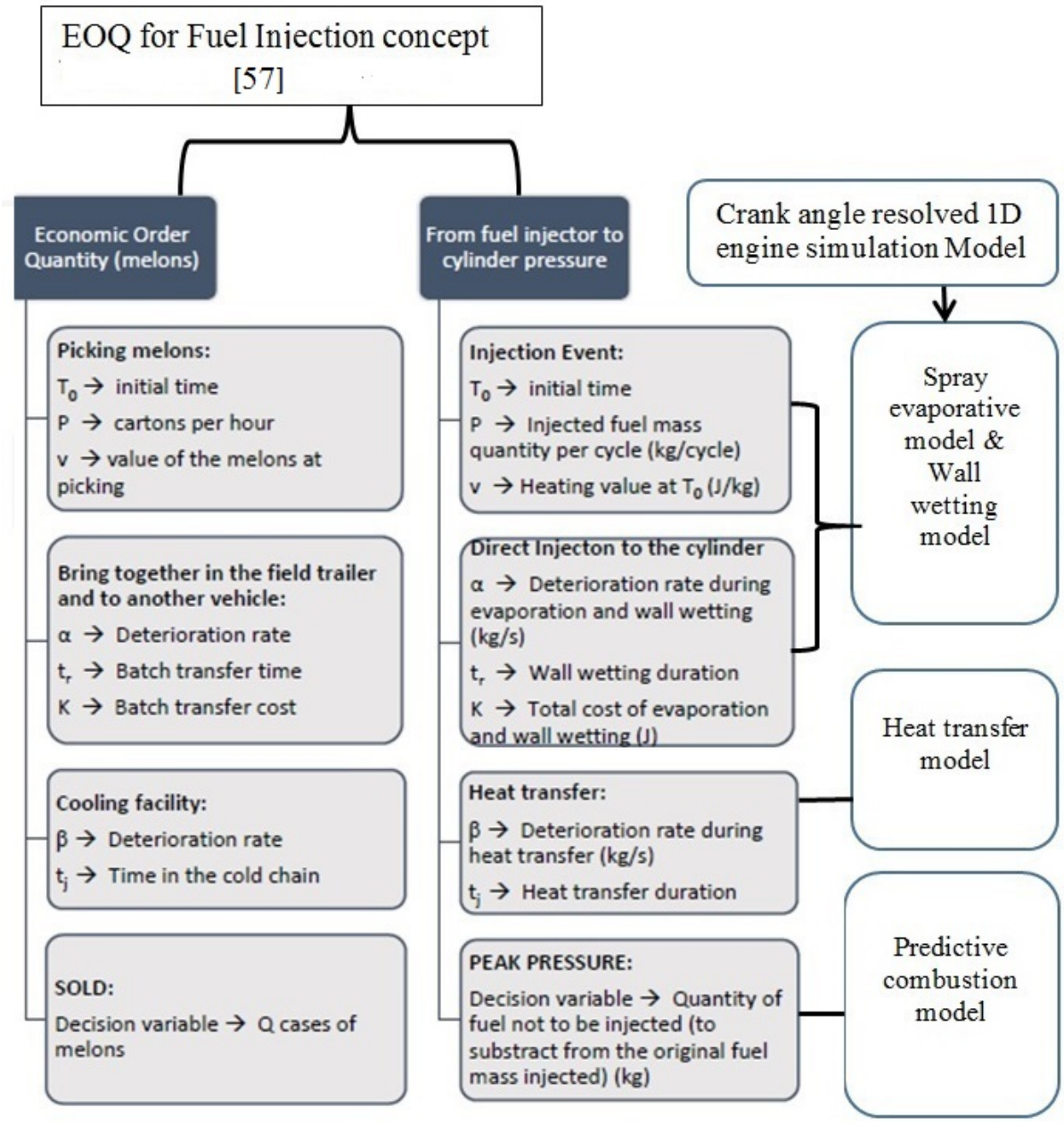

Figure 1: Introducing EOQ and Lambert W function in crank angle resolved 1-D engine simulation software (part of the figure is redrawn from [57]

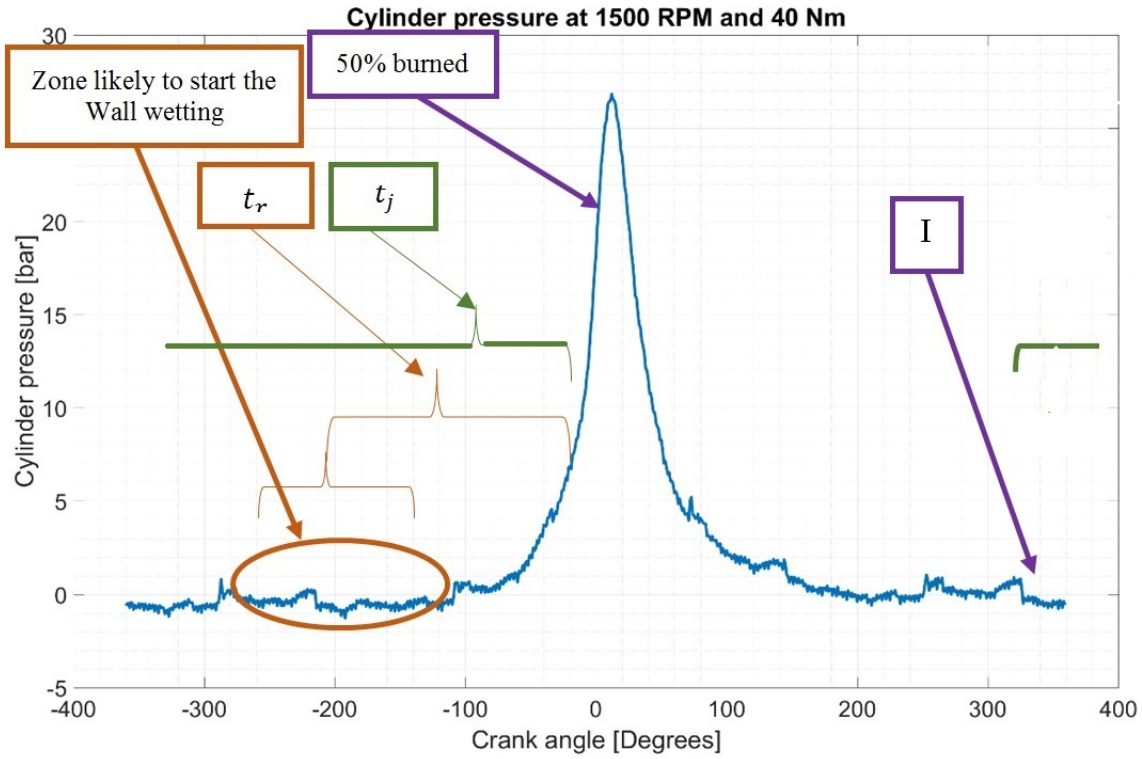

Figure 2: Cylinder pressure with the variables required for implementing EOQ parameters in 1-D, crank angle resolved engine model

Where t_j is the duration of injection. Figure 2 shows the phases of the injection, combustion and the main variables used in the EOQ function. For estimating deterioration rate in evaporation, α, using EOQ it is essential to include the spray evaporation model and wall-wetting model and heat transfer model to the main engine model. It is also essential to define injection event, start of combustion event in order to estimate the amount of fuel evaporated before the start of combustion. The injection rate, p, was estimated with the mass injected and the time of injection. The heating value, v, of the fuel is assumed as constant and is 4.395 x 10^7 J/kg , t_j and t_r are the duration of the injection process and wall wetting process and β the deterioration rate of the heat transfer process. Therefore, the total cost of evaporation and wall wetting, K, is the amount of energy lost because of the unburned fuel quantity in kJ.

$$K = m_{UB} * v \tag{6}$$

where m_{UB} is the mass of fuel unburned fuel. Now, it is possible to calculate the Q^* as shown in equation (5), this will be the amount of fuel that should be subtracted from the original mass of fuel to obtain the most optimal amount of fuel to be injected. Therefore, the iteration process will continue until the system is stabilized and Q^* become negligible as shown in Figure A1 in Appendix A.

4. PHYSICS MODELS

The EOQ function requires crank angle resolved quantities from spray evaporation, wall wetting and heat transfer and combustion models. The evaporation of the spray is the rate of amount of fuel, which evaporates from liquid to gas state in the spray jet. In addition, this spray can impinge on the piston surface and create a puddle of fuel; this puddle has a different evaporation rate, which will be calculated in the wall-wetting model, finally, the heat transfer model finds the amount of energy transfer by the walls of the cylinder.

4.1. Spray evaporation model

The direct injection system injects a jet of spray into the combustion chamber. The evaporation rate of the spray is governed by the equations proposed by Yoshizaki *et al.* [60]. These equations assume that the spray is constituted by individual packages as shown in Figure 3.

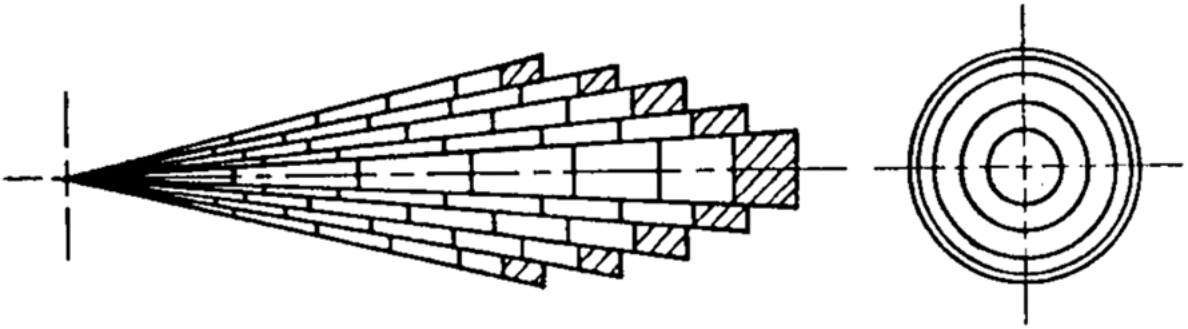

Figure 3: Individual packages of spray as illustrated by model

The evaporation ratio of each droplet in a spray jet based on Lefebvre [61] as a function of Sauter Mean diameter, D, (SMD) and the fluid properties are as follows :

$$\frac{\partial m_D}{\partial t} = 2\pi D \frac{K_a}{C_{p,a}} ln(1 + B_m) \tag{7}$$

where K_a and $C_{p,a}$ are the conductivity and specific heat at constant pressure of the air and B_m the mass transfer number. Similarly, the number of droplets in the spray could also be estimated by knowing the volume of fuel injected, SMD and the fuel density.

$$\dot{m}_{spray} = \frac{\partial m_D}{\partial t} . \left(\frac{\frac{m_{inj}}{\rho_{fuel}}}{\frac{3}{4}\pi . (D/2)^3} \right) \tag{8}$$

And the relationship between the droplet diameter and the temperature of the droplet is:

$$\frac{\partial D}{\partial t} = \frac{4K_a ln(1 + B_m)}{\rho_f C_{p,a} D} \tag{9}$$

$$\frac{\partial T}{\partial t} = \frac{\partial m_D}{\partial t} \frac{L_f}{m_D C_{p,f}} (\frac{B_T}{B_m} - 1) \tag{10}$$

Where ρ_f, $C_{p,f}$ and L_f are the density, specific heat and latent heat of fuel respectively and B_T is the heat transfer number, the vapour concentration, Y_{Fs}, at the surface of the droplet. According to [62]:

$$B_m = \frac{Y_{Fs}}{1 - Y_{Fs}} \tag{11}$$

$$B_t = \frac{C_{p,a}(T_{air} - T_{droplet})}{1 - Y_{Fs}} \tag{12}$$

$$Y_{Fs} = \left[1 + (\frac{P}{P_{Fs}} - 1)\frac{M_a}{M_f}\right]^{-1} \tag{13}$$

Where M_a and M_f are the molecular weight of the fuel and air, P is the pressure inside the cylinder and P_{Fs} is the fuel vapour pressure. Fuel vapour pressure is estimated using the Antoine and Clausius-Claypeyron relation [63]:

$$P_{Fs} = 10^{(A - \frac{B}{C+T})} \tag{14}$$

Where A, B and C are the Antoine coefficients. And the latent heat as a function of temperature [62]is:

$$L_f = L_{f,reference}. \left(\frac{T_{cr} - T_{droplet}}{T_{cr} - T_{bn}}\right)^{-0.38} \tag{15}$$

Where T_{cr} and T_{bn} are the critical temperature and boil temperature. By combining these principles, according to Hiroyasu *et al.* [64] the SMD is:

$$D = SMD = d.max\{\frac{SMD^{LS}}{d}, \frac{SMD^{HS}}{d}\} \tag{16}$$

$$\frac{SMD^{LS}}{d} = 4.12Re^{0.12}We^{-0.75}(\frac{\mu_f}{\mu_a})^{0.54}(\frac{\rho_f}{\rho_a})^{0.18} \tag{17}$$

$$\frac{SMD^{HS}}{d} = 0.38Re^{0.25}We^{-0.32}(\frac{\mu_f}{\mu_a})^{0.37}(\frac{\rho_f}{\rho_a})^{-0.47} \tag{18}$$

Where, d is the injector diameter, Re the Reynolds number, We the Webber number and μ_f and μ_a the static viscosity of fuel and air. The superscript LS and HS mean low speed and high speed and represents the incomplete and complete spray behaviour.

4.2. Wall wetting model

The wall wetting and evaporation happens when the fuel jet impinges on the surface and creates a puddle of fuel on top of the piston surface. The rate of evaporation of this fuel puddle is determined by the surface temperature [4]. Curtis *et al.* [4] defines the evaporation rate of fuel puddle for a port injection engine, and the same equation could be adapted for a direct injection engine:

$$\dot{m}_{wall} = Sh[\frac{A_{ls}}{D_p}\rho_{gm}D_{fa}Ln(1+\frac{\Delta MFF}{1-MFFs})] \tag{19}$$

Where Sh is the non-dimensional Sherwood number, A_{ls} is the surface area of the fuel puddle, ρ_{gm} is the density of the fuel, D_{fa} the diffusion coefficient between the fuel and the air, ΔMFF the difference in mass fraction of fuel in the vapour phase at the liquid surface and free stream, $MFFs$ is the mass fraction of fuel in the vapour phase and D_p is the port diameter. In this work, the port diameter will be substituted by the diameter of the cylinder.

The Sherwood number as function of Reynolds number, Re, and Schmidt number, Sc [4] is:

$$Sh = 1 + 0.023Re^{0.83}Sc^{0.33} \tag{20}$$

A MATLAB script was developed for estimating the surface of the puddle A_{ls}. This script assumes the spray jet as a cone and its penetration angle was estimated with the equations of [65,61]as follows:

$$\theta_{spray} = arctg(\frac{4\pi\sqrt{3}/6\sqrt{\rho_a/\rho_f}}{3+0.28.l_n/d}) \tag{21}$$

Where l_n is the length of the nozzle and d is its diameter.

In the penetration model, the spray has two steps, the first is called pre-breakup phase and the jet travels freely at constant velocity, when the fluid achieves the break-up time ($t_{b,k}$) the penetration of the spray will change and it will be in the post-breakup phase:

$$x(t)_{pre} = C_d\sqrt{\frac{2(P_{inj}-P_a}{\rho}}\ t \tag{22}$$

$$x(t)_{post} = 2.95(\frac{P_{inj} - P_a}{\rho_a})^{1/4}\sqrt{d.(t - t_{b,k})} + C_d\sqrt{\frac{2(P_{inj} - P_a)}{\rho_f}}\, t_{b,k} \quad (23)$$

$$t_{b,k} = 4.351\frac{\rho_f d}{C_d^2\sqrt{\rho_a(P_{inj} - P_a)}}(\frac{6-k}{5}) \quad (24)$$

Where P_{inj} is the pressure that the fluid is injected, P_a the pressure in the cylinder d the droplet diameter, k the radial index that it will be assumed as 1 and C_d is the discharge coefficient. This script calculates the surface from the parametric equation of a cone and the general and implicit equations of a plane. The equation for cone [66] :

$$x = \frac{h-u}{h}r.\cos\theta, \quad y = \frac{h-u}{h}r.\sin\theta, \quad z = u, \quad 0 \le u \le h, \quad 0 \le \theta \le 2\pi \quad (25)$$

The implicit equation of a plane [66]:

$$x = x_0 + u_1\lambda + v_1\mu$$

$$y = y_0 + u_2\lambda + v_2\mu$$

$$z = z_0 + u_3\lambda + v_3\mu$$

$$-\infty \le \lambda \le \infty, \quad -\infty \le \mu \le \infty \quad (26)$$

The general equation of a plane [66]:

$$Ax + By + Cz + D = 0 \quad (27)$$

The script assumes that the fuel spray is in the shape of perfect cone and the top of piston is horizontal plane. The depth, angle of spray cone and other relevant geometric characteristics of the spray cone are estimated using spray model and simple trigonometric functions. Finally, MATLAB returns the surface of the cut between the cone and the plane. Also, the script can return a visual solution of the script to check that the results make sense:

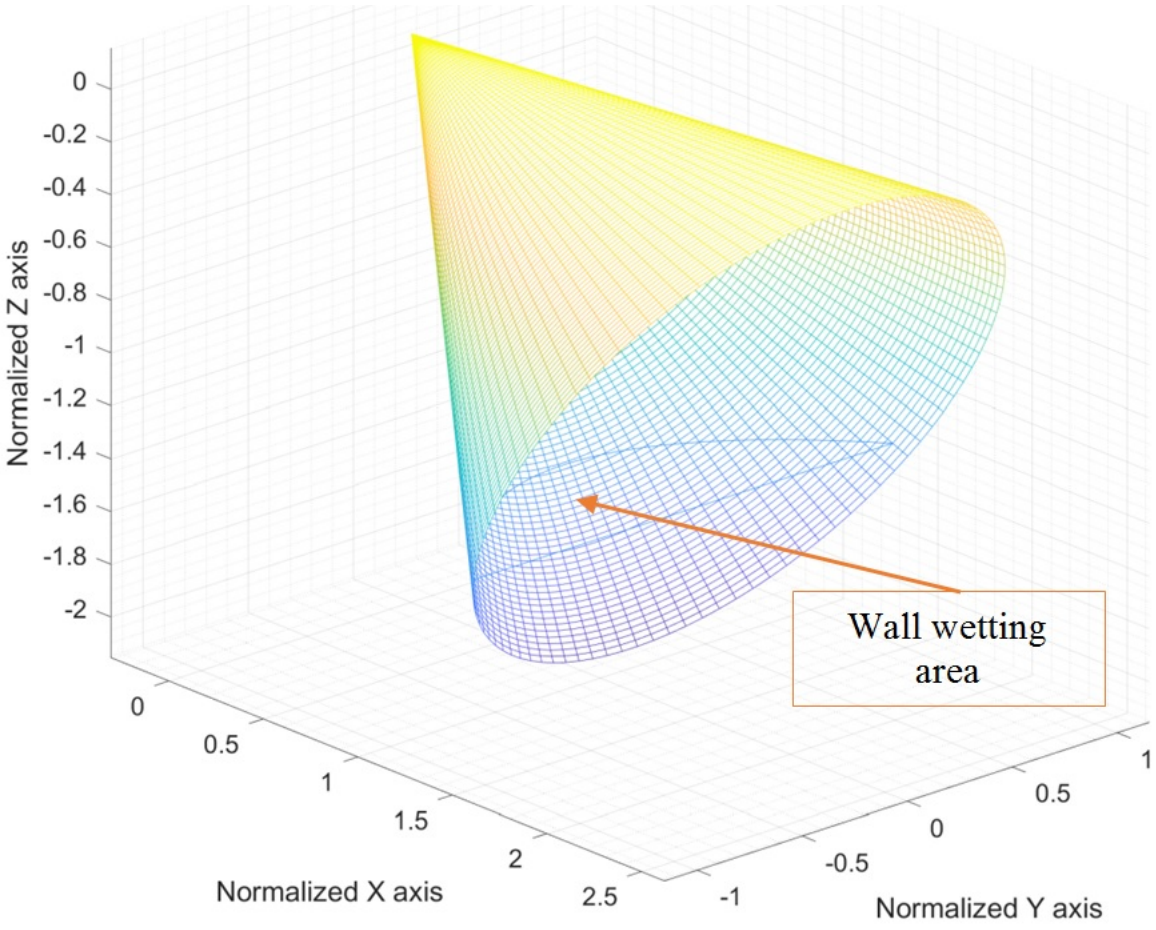

Figure 4: Wall wetting area found with MATLAB

4.3. Heat transfer model

The Woschni [67] heat transfer model was used to estimate the heat transfer ratio:

$$Q = h_c A(T_g - T_w) \tag{28}$$

Where A is the exposed area, T_g and T_w the temperatures of the gas in the combustion chamber and wall of the cylinder and h_c is the convection coefficient [32] which obey the following equation:

$$h_c = CB^{m-1}p^m w^m T_g^{-0.87} \tag{29}$$

Where C and m are empirical coefficients which take the values of 0.0035 and 0.8 respectively (Heywood 1988). Also, P and T_g are the pressure and temperature of the gas in the combustion chamber, finally w is the average gas speed in the cylinder and it is estimated using [32]:

$$w = C_1 S_p + C_2 \frac{V_d T_r}{P_r V_r}(P - P_m) \tag{30}$$

Where C_1 and C_2 are coefficient and their value depends on the stroke , V_d is the displacement volume, S_p is the mean speed of the piston and P_m is the motored pressure.

5. SIMULATION

A 1-D simulation of a gasoline direct injection engine developed in GT-Power was used in this study. The 1-D model is based on predictive combustion modelling approach and is a validated model [68] and the physics models were coupled with this 1-D model. Detailed description of this model and the strength of this model could be found in [68]. This coupling was carried out modifying the internal code of the 1-D simulation software through FORTRAN.

5.0.1. Coupling methodology

The 1-D simulation allows FORTRAN code to modify the internal code for adding evaporation rate and the heat transfer coefficient for estimating crank angle resolved quantities and couples with SIMULINK models as shown in Figure 5. The FORTRAN code is used to read the variables produced by

SIMULINK. For the evaporation model, the FORTRAN subroutine will receive the evaporation rate from SIMULINK and the internal code needs to know the amount of fuel evaporated in this time step for each component. Fortunately, the valour of the time step is an internal variable in the FORTRAN code and the operation of multiplying the evaporation rate by the length of time step is trivial. On the other hand, for the heat transfer, the 1-D simulation only allows modification of the convection coefficient. Then, SIMULINK estimates the coefficient and the FORTRAN subroutine reads it directly. Finally, the FORTRAN subroutines are compiled in a dynamic link library (dll) file, this file is called by the 1-D simulation software at the beginning of the simulation and it is used by the subroutines in the 1-D simulation at each time step. MATLAB code developed for implementing EOQ is given in Appendix A.

Two different SIMULINK models were developed in order to estimate the evaporation rate, the convection coefficient and for applying EOQ and the Lambert W function. The first model calculates the physics phenomena and the second the EOQ solution with the Lambert W function. These two models use variables from the 1-D simulation to achieve the solution. These models were compiled in C code to create a dynamic link library (dll) file. These files are called from the 1-D simulation software for each time step. The EOQ is referred to the whole cycle and not for each time step, therefore, the solution is calculated between 50 and 100 degrees before the firing TDC, since the injection event should have been completely finished and therefore, all the parameters required for estimating Q* will be available at this stage.

The 1-D simulation was created from a gasoline direct engine with the specifications shown in Table 1. The angles are referred to the top dead centre (TDC) in the firing stroke in Table 1.

After building the engine simulation, the 1-D variables are received by coupling FORTRAN and SIMULINK subroutines through sensors and results variables (RLT) in the model. Also, this study assumes that the conditions do not vary between the cylinders, therefore, the evaporation rate and heat transfer in each cylinder is the same but the signal is delayed by the firing order.

Bore	77mm
Stroke	85.8mm
Compression Ratio	10.5
displacement	1598 cc
Rated Power	171 bhp at 5500 rev/min
Rated Torque	240 Nm at 1700-4500 rev/min
fuel injection type	direct injection and wall-guided
maximum lift point for intake valve	-450 CA
Duration of intake	211 CA
Maximum lift point for exhaust valve	250 CA
Duration of Exhaust	205 CA
Rated Power	171 bhp at 5500 rev/min
Rated Torque	240 Nm at 1700-4500 rev/min

Table 1: Specification of experimental engine

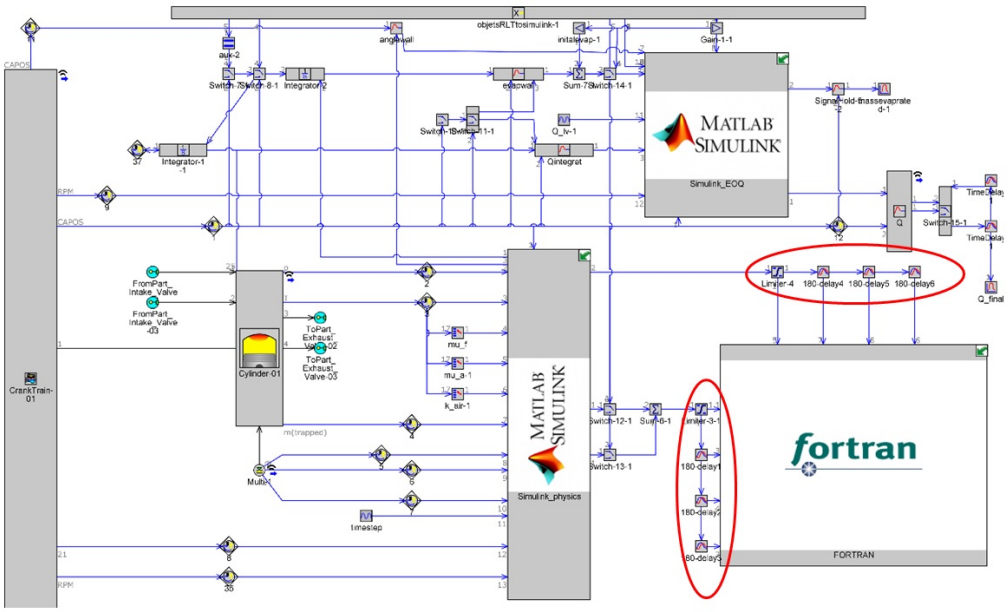

Figure 5: 1-D engine simulation model coupled with Economic Order Quantity (EOQ) and LambertW function using SIMULINK and FORTRAN

Speed(rev/min)/Torque(Nm)	1500	2000	2400	2800
20			91.6	93.1
40	97.8	96.7	92.1	94.8
60	94.0	98.3	93.5	94.1
80	95.9	98.9	95.8	95.2
100			94.0	94.5
120			94.4	94.2

Table 2: Correlation coefficients in percentage for the model validated against the experimental in-cylinder pressure at different engine operating conditions.

6. RESULTS AND DISCUSSION

After coupling FORTRAN and SIMULINK models with 1-D crank angle resolved engine simulation model, the results were obtained from the simulation and compared with the data obtained in a test bench in the first stage. In the second stage, the EOQ results were evaluated.

6.1. Model Validation

To validate the model, the measured cylinder pressure data obtained from the engine was used to compare with the results of the 1-D simulation model. Cylinder pressure is the most important data (Heywood, 1988) in the engine and, it enables us to derive significant amount of information in relation to combustion process. Experimental data from two different engine speeds (1500 and 2000 rpm) and three loading conditions (40, 60 and 80 Nm) were used for validating this approach. The model was also tested for two other engine speeds (2400 and 2800 rpm) and five loading conditions (20, 40, 60, 80 and 120 Nm).

Two different approaches to validate the data were used in the (Ventura and Samuel 2016) study: the peak pressure correlation and the area correlation. Peak pressure and area under the Pressure-crank angle are chosen for correlation since they are directly linked to energy release characteristics and indicated mean effective pressure from combustion process. The correlations were compared between the data obtained in the laboratory and the data of the 1-D simulation. In addition, the correlation between each sets of data was estimated with MATLAB for each engine condition separately as shown in Figure 6 and Table 2.

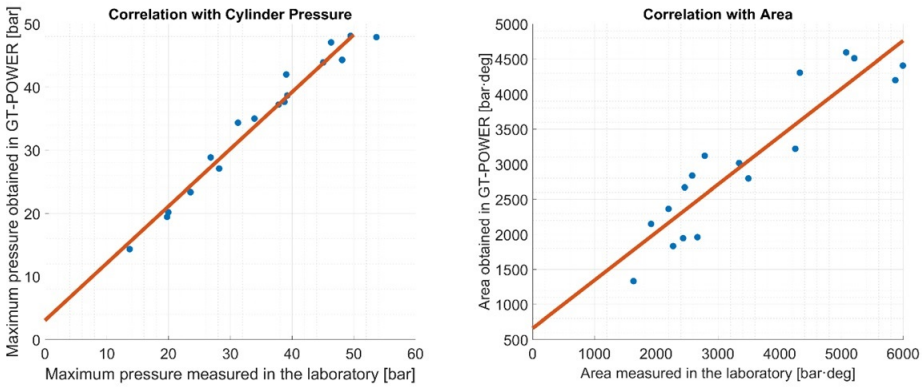

Figure 6: Model validation using Maximum Cylinder pressure and Area of pressure Vs Crank angle trace

The correlation coefficients for the two first approaches are 99.6% and 89.6%. Also, all the correlation coefficients evaluated in each engine condition separately have values above 91%. This level of correlations is considered sufficient for combustion analysis.

6.2. Results

Finally, the fuel injection strategy is evaluated for each engine operating condition. The final amount of fuel evaporated considering all the loses was assessed as shown in Table 3. The Table 3 shows that the least fuel evaporated is for high speed and low torque (2400 rpm and 40 and 60 Nm) operating condition. Although the amount of fuel evaporated is related to the fuel saved in the EOQ function through the parameter, and it does not hold a linear relationship, it is possible to see that the maximum fuel saved is for 2400 rpm and 20 Nm with a percentage of evaporation of 97.77%. The mean fuel saving is 4.7% for all conditions. The Table 3 shows that, when all the fuel is evaporated the solution of EOQ function is zero. Then, if the average is estimated only for the conditions that the fuel is not completely evaporated, this average of fuel saved will be 6.91%. Table 4 shows inputs and outputs of Lambert W function for each engine condition.

Speed (rev/min)	Torque (Nm)	mass of fuel (mf) injected (kg)	mf evaporated (kg)	mass evaporated %	Q	mf reduced %
1500	40	1.31E-05	1.31E-05	100	0.00E+00	0.00
	60	1.84E-05	1.50E-05	99.51	-1.08E-06	-5.83
	80	2.40E-05	2.39E-05	81.42	0.00E+00	0.00
2000	40	1.43E-05	1.43E-05	100.00	0	0.00
	60	1.65E-05	1.56E-05	94.45	-1.82E-06	-11.05
	80	1.92E-05	1.54E-05	80.26	-1.04E-06	-5.44
2400	20	1.82E-05	1.77E-05	97.77	-2.82E-06	-15.52
	40	6.77E-06	4.18E-06	61.80	-3.28E-07	-4.85
	60	7.79E-06	4.10E-06	52.61	-3.18E-07	-4.09
	80	1.23E-05	1.03E-05	83.56	-6.92E-07	-5.62
	100	1.68E-05	1.67E-05	99.46	0	0.00
	120	2.04E-05	2.02E-05	99.34	0	0.00
2800	20	1.89E-05	1.62E-05	85.89	-9.56E-07	-5.06
	40	9.93E-06	7.46E-06	75.10	-5.01E-07	-5.04
	60	1.14E-05	9.02E-06	79.13	-5.99E-07	-5.25
	80	1.46E-05	1.30E-05	88.80	-9.52E-07	-6.52
	100	2.20E-05	2.11E-05	95.82.80	-1.92E-06	-8.71
	120	2.20E-05	2.20E-05	100.00	0	0.00

Table 3: Percentage of fuel saved with EOQ and Lambert W function

7. Conclusion

This work has demonstrated that the EOQ and Lambert W function can be implemented in a 1-D engine simulation model for developing fuel injection strategy for estimating optimum quantity of the fuel that should be injected for achieving desired torque. The present work demonstrated that by coupling 1-D engine simulation model with Simulink and Fortran sub-routines one can estimate the evaporation rate of fuel injected, and heat tranfer rate in the engine as a function of crank angle in order to estimate the amount of fuel available in the gas phase for combustion process.

This novel approach based on the principles derived from EOQ for melon picking has the potential for developing software in loop calibration process for optimizing fuel injection quantity in gasoline direct injection engines and other liquid fuel and fuel neutral combustion engines.

The proposed approach could be extended across all type of combustion devices provided the suitable physical models are identified.Therefore, EOQ and Lambert W function based calibration process offer promising direction

Speed(rev/min)	Torque(Nm)	Z	W_{-1}	W_{-1}+1
1500	40	-0.36788	-1	0
	60	-0.36788	-1.00022	-2.23E-04
	80	-0.36788	-1	0
2000	40	-0.36788	-1	0
	60	-0.36788	-1.00008	-8.47E-05
	80	-0.36788	-1.00017	-1.74E-04
2400	20	-0.36788	-1.00005	-5.14E-05
	40	-0.36788	-1.00016	-1.58E-04
	60	-0.36788	-1.00019	-1.86E-04
	80	-0.36788	-1.00014	-1.37E-04
	100	-0.36788	-1	0
	120	-0.36788	-1	0
2800	20	-0.36788	-1.00014	-1.38E-04
	40	-0.36788	-1.00013	-1.34E-04
	60	-0.36788	-1.00013	-1.27E-04
	80	-0.36788	-1.0001	-1.04E-04
	100	-0.36788	-1.00008	-8.14E-05
	120	-0.36788	-1	0

Table 4: Inputs and outputs of Lambert W function for different engine operating conditions

for improving the fuel economy, and to reduce combustion generated CO2 and harmful combustion generated pollutants.

The current work clearly demonstrated the potential of EOQ and Lambert W function for improving the fuel economy, improving the combustion control and therefore, reduce the pollutants, however, the implementation using real hardware and hardware in loop is yet to be done. Therefore, the future work will implement it in real-hardware and thereby aim to reduce the development time for meeting the emission regulations significantly.

References

[1] Bowler, L.L., 1980. Throttle Body Fuel Injection (Tbi) an Integrated Engine Control System. SAE.800164. http://dx.doi.org/10.4271/800164.

[2] Kirwan, J. E., Drallmeier, J.A., Coverdill, R. E., Crawford, R. R., Peters, J. E., 1989. Spray Characteristics of Throttle Body Fuel Injection. SAE.890318. http://dx.doi.org/10.4271/890318.

[3] Ali Khan,M., Watson, H., Baker, P., Liew, G.,Johnston, D.,2006. SI Engine Lean-Limit Extension through LPG Throttle-Body Injection for Low CO2 and NOx..SAE. 2006-01-0495. http://dx.doi.org/10.4271/2006-01-0495.

[4] Curtis, E.W., Aquino, C.F., Trumpy, D.K., Davis, G.C., 1996. A New Port and Cylinder Wall Wetting Model to Predict Transient Air/ Fuel Excursions in a Port Fuel Injected Engine. SAE.961186. http://dx.doi.org/10.4271/961186.

[5] Z.Han, Z. Xu, N. Trigui. Spray/Wall Interaction Models for Multidimensional Engine Simulation. International Journal of Engine Research 1, no. 1 (2000/02/01): 127-146. Accessed 2017/03/09. http://dx.doi.org/10.1243/1468087001545308.

[6] Senda, J., Ohnishi, M., Takahashi,T., Fujimoto, H., Utsunomiya, A., Wakatabe, M.,1999. Measurement and Modeling on Wall Wetted Fuel Film Profile and Mixture Preparation in Intake Port of Si Engine. SAE.1999-01-0798. http://dx.doi.org/10.4271/1999-01-0798.

[7] Dunne, W.J., Brace, C.J., Stodart, A., 2007. Automated Calibration of an Analytical Wall-Wetting Model SAE.2007-01-0018. http://dx.doi.org/10.4271/2007-01-0018.

[8] Yu, J., Abe, M., Sukegawa, J., 2017. A Numerical Method to Simulate Intake-Port Fuel Distribution in Pfi Engine and Its Application. SAE.2017-01-0565. http://doi.org/10.4271/2017-01-0565.

[9] Ye, Z., Mohamadian. H., 2013. Model Predictive Control on Wall Wetting Effect Using Markov Chain Monte Carlo. IEEE Latin-America Conference on Communications.

[10] J. Zhang, T. Shen, G. Xu, J. Kako, Wall-Wetting Model Based Method for Air-Fuel Ratio Transient Control in Gasoline Engines with Dual Injection System. International Journal of Automotive Technology 14, no. 6 (2013): 867-873. http://dx.doi.org/10.1007/s12239-013-0095-y.

[11] B.Wang, S.Mosbach, S.Schmutzhard, S.Shuai, Y.Huang, M.Kraft, Modelling Soot Formation from Wall Films in a Gasoline Direct Injection Engine Using a Detailed Population Balance Model. Applied Energy 163 (2/1/): 154-166 (2016) http://dx.doi.org/10.1016/j.apenergy.2015.11.011.

[12] Malaguti, S., Cantore, G., Fontanesi, S., Lupi, R., Rosetti, A., 2009. CFD Investigation of Wall Wetting in a Gdi Engine under Low Temperature Cranking Operations. SAE.2009-01-0704. http://dx.doi.org/10.4271/2009-01-0704.

[13] Luijten, B., Adomeit, P., Brunn, A., Somers, B.,2013. Experimental Investigation of in-Cylinder Wall Wetting in Gdi Engines Using a Shadowgraphy Method. SAE. 2013-01-1604. http://dx.doi.org/10.4271/2013-01-1604.

[14] Ates, M., Matthews, R.D., Hall, M.J., 2017. A Full-Cycle Multi-Zone Quasi-Dimensional Direct Injection Diesel Engine Model Based on a Conceptual Model Developed from Imaging Experiments. SAE.2017-01-0537. http://doi.org/10.4271/2017-01-0537.

[15] Chen, L., et al., 2011. Spray Imaging, Mixture Preparation and Particulate Matter Emissions Using a GDI Engine Fuelled with Stoichiometric Gasoline/Ethanol Blends. In Internal Combustion Engines: Improving

Performance, Fuel Economy and Emission IMecHE, ISBN 978-0-85709-20502. 43-52: Woodhead Publishing.

[16] Zhao, F.Q., Lai, M.C., Harrington, D.L., 1997. A Review of Mixture Preparation and Combustion Control Strategies for Spark-Ignited Direct-Injection Gasoline Engines. SAE.970627. http://dx.doi.org/10.4271/970627.

[17] M. Castagn, J.P. Dumas, S. Henriot, Ph. Pierre, Use of Advanced Tools for the Analysis of Gasoline Direct Injection Engines. Oil & Gas Science and Technology - Rev. IFP 58, no. 1 (2003) 79-100.

[18] J. F. Le Coz, J. Cherel, S. Le Mirronet, Fuel/Air Mixing Process and Combustion in an Optical Direct-Injection Engine. Oil & Gas Science and Technology - Rev. IFP 58 (2003) no. 1 : 63-78.

[19] Pischinger, S., Schernus, C., Ltkemeyer, G., Theuerkauf, H.J., Winsel, T., Ayeb, M.,2004. Investigation of Predictive Models for Application in Engine Cold-Start Behavior. SAE.2004-01-0994. http://dx.doi.org/10.4271/2004-01-0994.

[20] Indranil, B., 2011. Development of Dynamic Constraint Models for a Model Based Transient Calibration Process. SAE.2011-01-0691. http://dx.doi.org/10.4271/2011-01-0691.

[21] Kitamura, Y., Sekikawa, A., Tokoro, M., Tomatsu, N., Uchimoto, T., Yuasa, H., Kato, A.,2011. Investigation About Predictive Accuracy of Empirical Engine Models Using Design of Experiments. SAE.2011-01-4271. http://dx.doi.org/10.4271/2011-01-1848.

[22] Wilhelm, C., Winsel, T., Ayeb, M., Theuerkauf, H.J., Brandt, S., Busche, E., Longo, C., Knoll, G.D., 2007. A Combined Physical / Neural Approach for Real-Time Models of Losses in Combustion Engines. SAE.2007-01-1345. http://dx.doi.org/10.4271/2007-01-1345.

[23] E.Corti, N.Cavina, A.Cerofolini, C.Forte, G.Mancini, D.Moro, F.Ponti, V.Ravaglioli, Transient Spark Advance Calibration Approach. Energy Procedia 45 (2014/01/01/): 967-976 (2014) https://doi.org/10.1016/j.egypro.2014.01.102.

[24] Rask, E., Mark, S.,2004. Simulation-Based Engine Calibration: Tools, Techniques, and Applications. SAE.2004-01-1264. http://dx.doi.org/10.4271/2004-01-1264.

[25] R.F.Turkson, F.Yan, M.K.A.Ali, B.Liu, J.Hu., Modeling and Multi-Objective Optimization of Engine Performance and Hydrocarbon Emissions Via the Use of a Computer Aided Engineering Code and the Nsga-Ii Genetic Algorithm. Sustainability 8, no. 1.(2016) http://dx.doi.org/10.3390/su8010072.

[26] T.Luigi, A.Roberto, N.Fabio, A 1D/3D Methodology for the Prediction and Calibration of a High Performance Motorcycle SI Engine. Energy Procedia 82 (2015/12/01/): 936-943 (2015) https://doi.org/10.1016/j.egypro.2015.11.842.

[27] Torre, D., Montenegro, A.G., A. Onorati.A., 2017. Coupled 1d-Quasi3d Fluid Dynamic Models for the Simulation of Ic Engine Intake and Exhaust Systems. In 17. Internationales Stuttgarter Symposium: Automobil- Und Motorentechnik, edited by Michael Bargende, Hans-Christian Reuss, and Jochen Wiedemann, 1461-1476. Wiesbaden: Springer Fachmedien Wiesbaden.

[28] Omekanda, S.O., Rahman, R., Lott, E.M., Rahman, S. S., Hornback, D.E., 2017. Optimal Parameter Calibration for Physics Based Multi-Mass Engine Model. SAE. 2017-01-0214. http://doi.org/10.4271/2017-01-0214.

[29] Lumsden, G., Browett, C., Taylor, J., Kennedy, G., 2004. Mapping Complex Engines. SAE.2004-01-0038. http://dx.doi.org/10.4271/2004-01-0038.

[30] R.Finesso, O.Marello, D.Misul, E.Spessa, M.Violante, Y.Yang, G.Hardy, C.Maier, Development and Assessment of Pressure-Based and Model-Based Techniques for the MFB50 Control of a Euro VI 3.0L Diesel Engine, SAE Int. J. Engines 10(4):1538-1555 (2017) https://doi.org/10.4271/2017-01-0794.

[31] Wurzenberger, J., Bartsch, P., and Katrasnik, T., 2010. Crank-Angle resolved Real time Capable Engine and Vehicle Simulation-Fuel consumption and Driving Performance. SAE.2010-01-0784. http://dx.doi.org/10.4271/2010-01-0784.

[32] J.Heywood, Internal Combustion Engine Fundamentals.McGraw-Hill, New York, 1988.

[33] J.I.Ghojel, Review of the development and applications of the Wiebe function: A tribute to the contribution of Ivan Wiebe to engine research,International Journal of Engine Research, Vol 11, Issue 4, pp. 297 - 312, First Published June 23 (2010), https://doi.org/10.1243/14680874JER06510

[34] Kumar, P. and Makki, I., 2017. Model Based Control of a Three-Way Catalytic Converter Based on the Oxygen Storage Level of the Catalyst. SAE.2017-01-0960.

[35] Lee, H., Lee, B., Kim, S., Kim, N., Rousseau, A., 2017. Model-Based Fuel Economy Technology Assessment. SAE.2017-01-0532. http://doi.org/10.4271/2017-01-0532.

[34] Watanabe, S., Miyata, Y., Ogata, Y., and Ivosic, V., 2017. Application of Model-Based Development to Engine Restart Vibration after Idling Stop. SAE .2017-01-1053. http://dx.doi.org/10.4271/2017-01-1053.

[37] C.Friedrich, M.Auer, G.Stiesch, Model Based Calibration Techniques for Medium Speed Engine Optimization: Investigations on Common Modeling Approaches for Modeling of Selected Steady State Engine Outputs, SAE Int. J. Engines 9(4):1989-1998 (2016), https://doi.org/10.4271/2016-01-2156.

[38] Poetsch, C. and Katrasnik, T., 2016. Crank-Angle Resolved Modeling of Fuel Injection, Combustion and Emission Formation for Engine Optimization and Calibration on Real-Time Systems. SAE.2016-01-0558 http://dx.doi.org/10.4271/2016-01-0558.

[39] Schlosser, A., Kinoo, B., Salber, W., Werner, S., Ademes, N., 2006. Accelerated Powertrain Development through Model Based Calibration. SAE.2006-01-0858. http://dx.doi.org/10.4271/2006-01-0858.

[40] Stuhler, H., Kruse, T., Stuber, A., Gschweitl, K., Piock, W., Pfluegl, H., Lick, P.,2002. Automated Model-Based Gdi Engine Calibration Adaptive Online Doe Approach. SAE .2002-01-0708. http://dx.doi.org/10.4271/2002-01-0708.

[41] Jiang, S., Nutter, D., Gullitti, A., 2012. Implementation of Model-Based Calibration for a Gasoline Engine. SAE.2012-01-0722. http://dx.doi.org/10.4271/2012-01-0722.

[42] Suzuki, K., Nemoto, M., Machida, K., 2006. Model-Based Calibration Process for Producing Optimal Spark Advance in a Gasoline Engine Equipped with a Variable Valve Train. SAE.2006-01-3235. http://dx.doi.org/10.4271/2006-01-3235.

[43] Beattie, T., Osborne, R., Graupner, W., 2013. Engine Test Data Quality Requirements for Model Based Calibration: A Testing and Development Efficiency Opportunity. SAE .2013-01-0351. http://dx.doi.org/10.4271/2013-01-0351.

[44] Onder, C., Geering, H., 1995. Model-Based Engine Calibration for Best Fuel Efficiency. SAE.950983. http://dx.doi.org/10.4271/950983.

[45] Wurzenberger, J. and Poetsch, C., 2015. Plant Modeling for Closed Loop Combustion Control - a Thermodynamic Consistent and Real-Time Capable Approach. SAE. 2015-01-1247. http://dx.doi.org/10.4271/2015-01-1247.

[46] A.S.Abbas, S.Saraswati, Cycle-by-Cycle Estimation of Cylinder Pressure and Indicated Torque Waveform Using Crankshaft Speed Fluctuations. Transactions of the Institute of Measurement and Control 37,(2015) no. 6: 813-825.

[47] Zhu, G., Daniels, C., and Winkelman, J., 2003. MBT Timing Detection and Its Closed-Loop Control Using in-Cylinder Pressure Signal. SAE.2003-01-3266. http://dx.doi.org/10.4271/2003-01-3266.

[48] Itshak,G., Powell, J.D., 1981. Optimal Closed-Loop Spark Control of an Automotive Engine. SAE.810058. http://dx.doi.org/10.4271/810058.

[49] Kampelmhler, F. T., Paulitsch, R., Gschweitl, K., 1993. Automatic Ecu-Calibration - an Alternative to Conventional Methods. SAE .930395. http://dx.doi.org/10.4271/930395.

[50] Rivard, J. G. Closed-Loop Electronic Fuel Injection Control of the Internal Combustion Engine. SAE.730005.(1973) http://dx.doi.org/10.4271/730005.

[51] Schnorbus, T., Pischinger, S., Krfer, T., Lamping, M., Tomazic, D., Tatur, M., 2008. Diesel Combustion Control with Closed-Loop Control of the Injection Strategy. SAE.2008-01-0651. http://dx.doi.org/10.4271/2008-01-0651.

[52] Franz, J., Schwarz, F., Guenthner, M., Reissing, J., Mueller, A., Donn, C., 2009. Closed Loop Control of an Hcci Multi-Cylinder Engine and Corresponding Adaptation Strategie. SAE.2009-24-0079. http://dx.doi.org/10.4271/2009-24-0079.

[53] Serizawa, K.,Ueda, D., Mikami, N., Tomida, Y., Weber, J., 2017. Realizing Robust Combustion with High Response Diesel Injector with Controlled Diffusive Spray Nozzle and Closed Loop Injection Control. SAE.2017-01-0845. http://doi.org/10.4271/2017-01-0845.

[54] John, C., Rachel, T., 1975. Closed-Loop Electronic Fuel and Air Control of Internal Combustion Engines. SAE.750369. http://dx.doi.org/10.4271/750369.

[55] Yinbo, C., Li, L., 2012. A Novel Closed Loop Control Based on Ionization Current in Combustion Cycle at Cold Start in a GDI Engine. SAE.2012-01-1339. http://dx.doi.org/10.4271/2012-01-1339.

[56] ika, Z., Valek, M., Florin, M., Macek, J., Polek, M.,2008. Multilevel Predictive Models of Ic Engine for Model Predictive Control Implementation. SAE.2008-01-0209. http://dx.doi.org/10.4271/2008-01-0209.

[57] V.Robert, S.Samuel, Optimization of Fuel Injection in GDI Engine Using Economic Order Quantity and Lambert W Function. Applied Thermal Engineering 101 (2016/05/25/): 112-120.(2016) http://dx.doi.org/10.1016/j.applthermaleng.2016.02.024.

[58] D.Stephen, R.Warburton, On the Lambert W Function: Economic Order Quantity Applications and Pedagogical Considerations.International Journal of Production Economics Volume 140, Issue 2, (2012), 756-764.

[59] B.Joseph, G.Scudder, Supply Chain Strategies for Perishable Products: The Case of Fresh Produce. Production and Operations Management 18, (2009). no. 2: 129-137. http://dx.doi.org/10.1111/j.1937-5956.2009.01016.x.

[60] Takuo, Y., Nishida, K., Hiroyuki Hiroyasu, H.,(1993). Approach to Low Nox and Smoke Emission Engines by Using Phenomenological Simulation.SAE.930612. http://dx.doi.org/10.4271/930612, 1993.

[61] A.Lefebvre, *Atomization and Sprays*. CRC Press, 1988.

[62] Robert, M., Heywood, J.B.,1999. Evaporation of in-Cylinder Liquid Fuel Droplets in an Si Engine: A Diagnostic-Based Modeling Study. SAE.1999-01-0567. http://dx.doi.org/10.4271/1999-01-0567.

[63] C.Antoine, Tensions Des Vapeurs; Nouvelle Relation Entre Les Tensions Et Les Tempratures. Comptes Rendus des Sances de l'Acadmie des Sciences (1888) 107: 681-684, 778-780, 836-837.

[64] Hiroyuki, H., Arai, M., Tabata, M., 1989. Empirical Equations for the Sauter Mean Diameter of a Diesel Spray. SAE.890464. http://dx.doi.org/10.4271/890464.

[65] Dohoy, J., Assanis. D.N., 2001. Multi-Zone Di Diesel Spray Combustion Model for Cycle Simulation Studies of Engine Performance and Emissions. SAE.2001-01-1246. http://dx.doi.org/10.4271/2001-01-1246.

[66] I.N.Bronshtein , K.A.Semendyayev, G.Musiol , H.Mhlig, Handbook of Mathematics. 5th ed.: Springer, 2007.

[67] Woschni, G., 1967. A Universally Applicable Equation for the Instantaneous Heat Transfer Coefficient in the Internal Combustion Engine. SAE.670931. http://dx.doi.org/10.4271/670931.

[68] Divakera, Arjun, D., Samuel, S.,2011. Numerical Simulation of Adaptive Combustion Control for Fuel-Neutral Smart Engines. SAE.2011-01-0848. http://dx.doi.org/10.4271/2011-01-0848.

[69] Spessa, E., D'Ambrosio, S., Iemmolo, D., Mancarella, A., Vitolo, R., Hardy, G., 2017. Steady-State and Transient Operations of a Euro Vi 3.0l Hd Diesel Engine with Innovative Model-Based and Pressure-Based Combustion Control Techniques. SAE.2017-01-0695. http://dx.doi.org/10.4271/2017-01-0695.

[70] M. Y. Jaber, R. Y. Nuwayhid, M. A. Rosen, Price-driven economic order systems from a thermodynamic point of view, International Journal of Production Research, 42:24, 5167-5184,(2007) DOI: 10.1080/00207540412331281971.

[71] H.Jawad, Y.M.Jaber, M.Bonney, The Economic Order Quantity model revisited: an Extended Exergy Accounting approach. Journal of cleaner production, 105, 64-73, (2015). doi: 10.1016/j.jclepro.2014.06.079.

[72] H.Jawad, M.Y.Jaber, M.Bonney, M.A.Rosen, Deriving an exergetic economic production quantity model for better sustainability. Applied Mathematical Modelling, 40(11-12),(2016) 6026-6039.

[73] F.M.Harris, How Many Parts to Make at Once, Factory, The Magazine of Management,10:2,(1913) 135136, 152. Reprinted in Operations Research 38:6 (1990), 947950.

[74] L.B.Schwarz, The economic order-quantity (EOQ) model, chapter 8 from Building Intuition: Insights From Basic Operations Management Models and Principles,(2008) doi:10.1007/978-0-387-73699-0-8, ISSN:0884-8289.

8. Appendix A

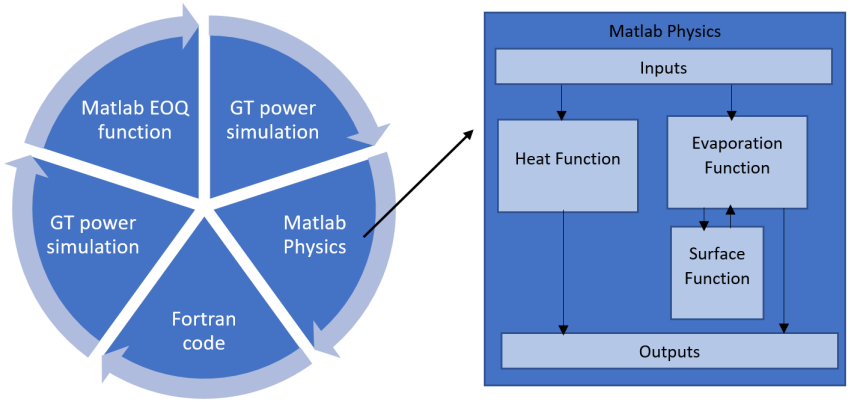

Figure A1 Overview of Flow of information and iterative process

Matlab Code

```
function [evap,angle_spray,x_spray,m_wall]=calculates(
    theta,P_cylinder,T_cylinder,mu_f,mu_a,k_a,m_traped,
    fuel_rate_injected,P_rail,v_injector,
    average_gas_velocity,dt,surface)
     global time
     global m_spray_acumulated
     global m_wall_acumulated
     global Pa
     global T_drop_previous
     global flag
     global d_previous
     global v_previous
     global m_fuel_traped
     global m_spray
     global time_injected
     global t_bk
    % global m_wall_traped
     global D_droplet
     global surface_saved

     %definition of numerical variables (required by
        MATLAB to compile the script)
      d_v=0; d=0; v=0; m_wall=0; d_T_drop=0; C_drag=0;
         Re=0; We=0; L_fs=0; Bt=0; incr_MFF=0; MFF=0;
         Pfs=0; Yfs=0; Bm=0; C_drag=0; Sc=0; Sh=0;
         SMD_ls=0; SMD_hs=0; T_drop=0; d_d=0; x_spray=0;
          angle_spray=0; time_from_injection=0;
         m_spray_total=0; m_suspended=0; N_droplets=0;
         vol_suspended=0; Pfs_free=0; Yfs_free=0; r=0;
         Sc=0; A_ls=0; m_air_traped=0;

     %constants of the model
     bore=77/1000;                  %bore of the cylinder
     diamter_injector= 0.3/1000; %diameter of the
        injector
     Cd=0.7;                        %Nozzle Discharge
```

```
    Coefficient of the injector
T_injector=350;                 %Injected Fluid
    Temperature
Ma=29.97;                       %Molar weith of air
Mf=114.2285;                    %Molar weith of fuel
    http://webbook.nist.gov/cgi/inchi?ID=C111659&
    Mask=4#Thermo-Phase
Rg_air=287.058;                 %Constant gas of air
Cp_a=1024;                      %Specific heat of air
    at constante pressure
Cp_f=2341;                      %Specific heat of fuel
    at constante pressure http://webbook.nist.gov/
    cgi/inchi?ID=C111659&Mask=2#Thermo-Condensed

k=1;                            %Radial index for time
    of break up
sigma=0.02162;                  %Surface tension of
    http://www.surface-tension.de/
Tcr=568.9;                      %Critial temperature of
     octane (fuel) http://webbook.nist.gov/cgi/inchi
    ?ID=C111659&Mask=4#Thermo-Phase
Tbn=398.7;                      %Boil temeperature of
    Octane (fue)    http://webbook.nist.gov/cgi/
    inchi?ID=C111659&Mask=4#Thermo-Phase
LTBN=350000;                    %Heat of Vaporization
    at 298K of fuel GT-POWER
D_fa =7.353E-7;                 %Diffusion coeficient
    between octane and air  http://www.jocet.org/
    papers/012-J30011.pdf
rho_f=750;                      %Density of fuel

rho_a=P_cylinder/(T_cylinder*Rg_air); %density of
    air
Ap=(pi/4)*bore^2;               %Area of piston
% = 0.3*P_rail;                %Pressure lost in the
    injection
ln=diamter_injector/10;         %Lenght of nozzle

```

```
45      angle_of_injection=360;
46
47      %Antonie coefficients http://webbook.nist.gov/cgi/
            inchi?ID=C111659&Mask=4#Thermo-Phase
48      if T_cylinder<297.1
49          A = 5.2012;
50          B =1936.281 ;
51          C = -20.143 ;
52      else
53          A = 4.04867;
54          B =1355.126 ;
55          C = -63.633 ;
56      end
57
58      %inicializating the global variables in each cycle
59      if 20< theta && theta < 60 %we use a range of
            angles to avoid that a big timestep jumps the
            iniciaization
60          flag=0;                 %flag used for save the
                variables at start of injection
61          m_fuel_traped=0;        %fuel acumulated in the
                cyinder
62          m_spray_acumulated=0;   %Acumulated fuel
                evaporated by the spray jet
63          m_wall_acumulated=0;    %Acumulated fuel
                evaporated by the wall wetting
64          %m_wall_traped=0;       %fuel acumulated in the
                cyinder
65          D_droplet=1E-3;         %Diameter of the
                droplet
66          surface_saved=0;        %Surface of wall
                wetting in the previous timestep (is used to
                be sure that the surface only will increase
                )
67      end
68
69      %A_ls is the wall wetting surface
70      %This part avoid that the wall wetting surface
```

```
        decreasing in the same cycle, we are assuming
        that once the spray hits the piston, the puddle
        only will be able to increase
71      if surface_saved>surface
72          A_ls= surface_saved;
73      else
74          A_ls= surface;
75          surface_saved=surface;
76      end
77
78      %acumuated variables
79      time=time+dt;                   %time of simulation
80      m_fuel_traped=fuel_rate_injected*dt + m_fuel_traped
            ; %amount of fuel in the cylinder
81      m_air_traped=m_traped-m_fuel_traped;
                            %amount of air in the cylinder
82      evaporated_mass_acumulated=m_spray_acumulated+
            m_wall_acumulated; %amount of fuel evaporated
83
84      if m_fuel_traped>0 %if we are in the evaporation
            phase, we do not want to estimate anything is
            the exahust, this start when the fuel is
            injected
85
86          %variabes marked in the injection point
87          if flag==0
88              Pa=P_cylinder;          %Pressure at
                    injection
89              d= D_droplet;           %Diameter of the
                    droplet
90              d_previous=d;           %Diameter of the
                    droplet in the previous timestep
91              v=v_injector;           %Speed of the
                    injection
92              v_previous=v;           %Speed of the
                    injection in the previous timestep
93              T_drop=T_injector;      %Temperature of the
                     droplet
```

```
94              T_drop_previous=T_drop; %Temperature of the
                    droplet
95              time_injected=time;     %Time when the
                   spray are injected, after we will
                   correect it with time_from_injection to
                   stimate the total time from the
                   injection
96              t_bk=4.251*((rho_f*diamter_injector)/(Cd^2*
                   sqrt(rho_a*abs(P_rail-P_cylinder))))
                   *((6-k)/5); %time for break-up
97              flag=1;
98          end
99      %Very high Re numbers in combustion, but the model is
            validated!
100         Re=rho_a*average_gas_velocity*bore/mu_a;
101         %Re=(flow_re*bore)/(Ap*mu_a);            %
                Reynols number
102         C_drag=(24/Re)*(1+((Re^(2/3)/6)));
                                %Drag coefficient of a
                droplet
103         L_fs=LTBN*(((Tcr-T_drop_previous)/(Tcr-Tbn))
                ^-0.38); %Latent heat of evaporation
104
105         %considerer to change all Pa by P_cylinder
106         Pfs_free=1E5*10^(A-(B/(T_cylinder+C)));     %
                Pfs of the droplet fuel evaporated
107         Yfs_free=1/(1+((Pa/Pfs_free)*(Ma/Mf)));     %
                Yfs of the droplet fuel evaporated
108         Pfs =1E5*10^(A-(B/(T_drop_previous+C)));    %
                Pfs of the droplet fuel in the liquid state
109         Yfs =1/(1+((Pa/Pfs)*(Ma/Mf)));              %
                Yfs of the droplet fuel in the liquid state
110
111         incr_MFF=abs(Yfs-Yfs_free);
112         MFF=Yfs;
113
114         %Bt and Bm we have a loop here because the
                change of the droplet deperature depens on
```

```
            Bt and Bm and, at the same time Bt and Bm
            depends on the temperature of the droplet
115         if Yfs==1
116             Bm=1;
117         else
118             Bm=Yfs/(1-Yfs);
119         end
120
121         Bt=Cp_a*(T_cylinder-T_drop_previous)/L_fs;
                        %Mass transfer number of
            temperature
122
123         %Droplet Temperature
124         %we have a loop here because the change of the
            droplet Temperature depens on Bt and Bm and,
             at the same time Bt and Bm depends on the
            temperature of the droplet
125         if m_spray_acumulated>0
126             d_T_drop=m_spray*(L_fs/(Cp_f*
                    m_spray_acumulated))*((Bt/Bm)-1); %
                    Differencial of temperature
127             %control that the differencial is not very
                    big due to incrrect timesteps
128             if d_T_drop>10000 ||   d_T_drop<-10000
129                 T_drop =T_drop_previous;
130             else
131                  T_drop=T_drop_previous-(d_T_drop*dt);
132             end
133             %Control that the temperature of the
                    droplet is not less than 0
134             if T_drop <0
135                  T_drop=T_injector;
136             end
137             T_drop_previous=T_drop;
138          end
139
140         %Droplet Diameter
141         d_d=(4*k_a*log(1+Bm))/(rho_f*Cp_a*d_previous);
```

```
142         d=d_previous-d_d*dt;
143         if d>0 %no droplet negatives diameter or zero
144             d_previous=d;
145         end
146
147         %Droplet Velocity
148         if d_previous>0 %no droplet negatives diameter
                or zero
149             d_v=((3*rho_a)/(4*rho_f))*(C_drag)*(((mu_f-
                    mu_a)^2)/(d_previous));
150          v=v_previous-(d_v*dt);
151         v_previous=v;
152          if v<0
153             v=0;
154             v_previous=v;
155          end
156         end
157
158         %Dimensionless numbers
159         Sc = mu_f/(rho_f*D_fa);
160         We =rho_a*(v^2)*D_droplet/sigma;
161         Sh=1+(0.023*(Re^(0.83))*(Sc^(0.33)));
162         %Sauter mean diameter
163         SMD_ls=4.12*(Re^0.12)*(We^-0.75)*((mu_f/mu_a)
                ^0.54)*((rho_f/rho_a)^0.18);
164         SMD_hs=0.38*(Re^0.25)*(We^-0.32)*((mu_f/mu_a)
                ^0.37)*((rho_f/rho_a)^-0.47);
165         D_droplet= diamter_injector*max(SMD_ls, SMD_hs)
                ;
166
167         %Number of Droplets
168         if m_wall_acumulated>0
169             m_suspended=m_fuel_traped*0.95; %Correction
                    if we have wall wetting
170         else
171             m_suspended=m_fuel_traped;
172         end
```

```
173         vol_suspended=m_suspended/rho_f;
174         N_droplets=vol_suspended/((4*pi/3)*(D_droplet
                /2)^3);
175
176         if Bm<0 %Control to avoid negative log
177             Bm=0;
178         end
179
180         %evaporation rates
181         m_spray=2*3.1415*D_droplet*(k_a/Cp_a)*log(1+Bm)
                ; %it is necessary for stimate the droplet
                temperature, for this reason it is a global
                variable
182         m_spray_total=m_spray*N_droplets;
183         m_wall=Sh*(rho_f*D_fa*(A_ls/bore)*log(1+(
                incr_MFF/(1-MFF))));
184
185         evap=m_spray_total+m_wall; %Final evaporation
                rate
186
187         %Acumulated evaporations
188         m_spray_acumulated=m_spray_acumulated+
                m_spray_total*dt;
189         m_wall_acumulated=m_wall_acumulated+m_wall*dt;
190
191         %If all fuel were evaorated, we can not
                continue evaorating
192         if m_fuel_traped<(m_spray_acumulated+
                m_wall_acumulated)
193             evap=0;
194             m_spray_acumulated=m_spray_acumulated-
                    m_spray_total*dt;
195             m_wall_acumulated=m_wall_acumulated-m_wall*
                    dt;
196         end
197
198         %Penetration and angle for the spray to stimate
                the A_ls in other script (see simulink
```

```
            blocks)
199         time_from_injection=time-time_injected;
200         if (time_from_injection)<t_bk
201             x_spray=(Cd*sqrt(abs(P_rail-P_cylinder)/
                    rho_f))*(time_from_injection);
202         else
203             x_spray=2.95*((abs(P_rail-P_cylinder)/rho_a
                    )^0.25)*sqrt(diamter_injector*abs(
                    time_from_injection-t_bk))+(Cd*sqrt(abs(
                    P_rail-P_cylinder)/rho_f))*(t_bk);
204         end
205         angle_spray=atan((3.6276*sqrt(rho_a/rho_f))
                /(3+0.28*(ln/diamter_injector)))*180/pi;
206
207     else
208         evap=0;
209     end
210 %{
211 subplot(4,4,1);
212 hold on
213 plot(time,theta,'*'), title('theta')
214 subplot(4,4,2);
215 hold on
216 plot(time,m_wall,'*'), title('m_wall')
217 subplot(4,4,3);
218 hold on
219 plot(time,m_spray_total,'*'), title('m_spray_total')
220 subplot(4,4,4);
221 hold on
222 plot(time,m_spray,'*'), title('m_spray')
223 subplot(4,4,5);
224 hold on
225 plot(time,N_droplets,'*'), title('N_droplets')
226 subplot(4,4,6);
227 hold on
228 plot(time,angle_spray,'*'), title('Bm')
229 subplot(4,4,7);
230 hold on
```

```
231 plot(time,x_spray,'*'), title('Bt')
232 subplot(4,4,8);
233 hold on
234 plot(time,average_gas_velocity,'*'), title('
    average_gas_velocity')
235 subplot(4,4,9);
236 hold on
237 plot(time,incr_MFF,'*'), title('incr_MFF')
238 subplot(4,4,10);
239 hold on
240 plot(time,MFF,'*'), title('MFF')
241 subplot(4,4,11);
242 hold on
243 plot(time,A_ls,'*'), title('A_ls')
244 subplot(4,4,12);
245 hold on
246 plot(time,Sh,'*'), title('Sh')
247 subplot(4,4,13);
248 hold on
249 plot(time,Re,'*'), title('Re')
250 subplot(4,4,14);
251 hold on
252 plot(time,rho_a,'*'), title('rho_a')
253 subplot(4,4,15);
254 hold on
255 plot(time,m_spray_acumulated,'*'), title('
    m_spray_acumulated')
256 subplot(4,4,16);
257 hold on
258 plot(time,m_wall_acumulated,'*'), title('
    m_wall_acumulated')
259 %}
260 end
261
262 function surface = fcn(angle_spray,x_spray,x_piston)
263 %This script stimate the wall wetting area with the
    parametrics equations of a cone and a plane
264
```

```
265     x_piston=abs(x_piston);
266     %clerance=0.001;
267     x0=77/1000; y0=0; z0=0;%position of the inyector
268     inclination_degree=70;%angle of inclination of the
            inyector
269
270     tita = 90-angle_spray; %angle of spray
271     h = x_spray; %penetration of the spray
272     D=6.25/1000+x_piston; %position of the piston,
            be carefull with the coordinates
273
274     r=h/tan(tita*pi/180); %radio of the cone
275     m = h/r;
276     phi = -inclination_degree*pi/180 ;
277     [u,A] = meshgrid(linspace(0,r,100),linspace(0,2*pi
            ,100));
278     [tt]=meshgrid(linspace(-h,h,100));
279
280     %parametrization of the cone
281     X_cone = (u .* cos(A)*cos(phi) -  (h/r)*u*sin(phi)+
            x0);
282     Y_cone = u .* sin(A)+y0;
283     Z_cone = u .* cos(A)*sin(phi) +  (h/r)*u*cos(phi)+
            z0;
284
285     %paramezitration of the cone's cut with the plane,
            but, the cone has not end
286     k=(D-z0)./(cos(A)*sin(phi) + (h/r)*cos(phi));
287     x1=(k).*cos(A)*cos(phi)-(h/r)*(k)*sin(phi)+x0;
288     y1=(k).*sin(A)+y0;
289     z1=D.*(ones(100,100));
290     %vector of the hight of the cone
291     p=[-h*sin(phi)+x0,0+y0,h*cos(phi)+z0];
292     origin=[x0,y0,z0];
293     vd=p-origin;%vector director of plane of the final
            cone
294
295     D2=p(1)*vd(1)+p(2)*vd(2)+p(3)*vd(3);
```

```
296     %line created by the cut with this plane and the
            piston plane
297     vd2=cross(vd,[0,0,D]);
298     p2=[(D2-D)/vd(1),y0,D];
299
300     x_line2=p2(1)+vd2(1).*tt;
301     y_line2=p2(2)+vd2(2).*tt;
302     z_line2=p2(3)+vd2(3).*tt;
303
304     %create surface delimitated by this line and the
            previous cut of the cone
305     cut=[x1(:,1) y1(:,1) z1(:,1) x_line2(1,:)' y_line2
            (1,:)' z_line2(1,:)' ];
306     %inizialization of final matrix
307     cut_final_x1=ones(100); cut_final_y1=ones(100);
            cut_final_z1=ones(100);
308
309     j=1; k=1;
310     %The loop find the surface of the cut until the
            line,
311     %But, the line is not painted because we have the
            inicialization of the
312     %matrx cut_final
313     for i=1:100
314         if cut(i,1)<=cut(i,4)
315         cut_final_x1(i,:)=cut(i,1);
316         cut_final_y1(i,:)=cut(i,2);
317         cut_final_z1(i,:)=cut(i,3);
318         j=i;
319         k=k+1;
320         end
321     end
322     %we use the point of cut the line and extend to the
            first and final matrix
323     %matlab will joint this points automatically
324     cut_final_x1(1:j-k+1,:) = cut(j,1);
325     cut_final_y1(1:j-k+1,:) = cut(j,2);
326     cut_final_z1(1:j-k+1,:) = cut(j,3);
```

```
327         cut_final_x1(j:100,:) = cut(j,1);
328         cut_final_y1(j:100,:) = cut(j,2);
329         cut_final_z1(j:100,:) = cut(j,3);
330        %Finally, we will calculate the matrix
331         surface1=polyarea(cut_final_x1,cut_final_y1);
332         surface=surface1(1);
333         if isnan(surface)
334             surface=0;
335         end
336
337 end
338
339
340
341 function [hc,w]=convection(theta,P_cylinder,T_cylinder,
      rpm,x_piston)
342 %Estimate the heat transfer coefficient and the average
       speed inside the cylinder
343 global Pi
344 global Tr
345 global Vr
346 global time
347 %definition of numerical variables (required by MATLAB
      to compile the script)
348 w=0; Pm=0;
349
350 %Constants
351 bore=77/1000;
352 stroke=85.8/1000;
353 C=0.0035;              %Empirical coefficient
354 m=0.8;                 %Empirical coefficient
355 gamma=1.4;             %Adiabatic coefficient
356 r=10.5;                %Compresion ratio
357 x_piston=abs(x_piston);
358
359 Sp=rpm*stroke/30;          %Mean speed of the piston
360 Vd=pi/4*stroke*bore^2;     %Displacement volume
361
```

```
362 %C1 and C2 coefficients (Heywood)
363 if theta>148 && theta<554 %exchange between 148 and 554
364     C1=6.18;
365     C2=0;
366 elseif theta>=554 && theta<=698 %compression between
      554 and 698
367     C1=2.28;
368     C2=0;
369 else %combustion and expansion
370     C1=2.28;
371     C2=3.24e-3;
372 end
373
374 %Pressure temperature and volumen for a reference state
375 if theta>300 && theta<350 || Vr==0
376     Pi=P_cylinder;
377     Tr=T_cylinder;
378     Vr=pi/4*(x_piston+(6.25/1000))*bore^2;
379 end
380
381 Pm=((Vr/(pi/4*(x_piston+(6.25/1000))*bore^2))^gamma)*Pi
      ; %Isentropic pressure
382 if Pm==P_cylinder
383     w=1;
384 else
385     w=C1*Sp+C2*((Vd*Tr)/(Pi*Vr))*(P_cylinder-Pm); %
        average speed inside the cylinder
386 end
387 w=abs(w);
388
389 hc=C*(bore^(m-1))*(P_cylinder^m)*(w^m)*(T_cylinder
      ^(0.75-1.62*m)); % heat transfer coefficient
390 %{
391 subplot(4,4,1);
392 hold on
393 plot(time,theta,'*'), title('theta')
394 subplot(4,4,2);
395 hold on
```

```
396 plot(time,P_cylinder,'*'), title('P_cylinder')
397 subplot(4,4,3);
398 hold on
399 plot(time,Vr,'*'), title('Vr')
400 subplot(4,4,4);
401 hold on
402 plot(time,w,'*'), title('w')
403 subplot(4,4,5);
404 hold on
405 plot(time,hc,'*'), title('hc')
406 subplot(4,4,6);
407 hold on
408 plot(time,x_piston,'*'), title('x_piston')
409 subplot(4,4,7);
410 hold on
411 plot(time,P_cylinder,'*'), title('P_cylinder')
412 subplot(4,4,8);
413 hold on
414 plot(time,T_cylinder,'*'), title('T_cylinder')
415 subplot(4,4,9);
416 hold on
417 plot(time,Pm,'*'), title('Pm')
418
419 subplot(4,4,10);
420 hold on
421 plot(time,We,'*'), title('We')
422 subplot(4,4,11);
423 hold on
424 plot(time,Re,'*'), title('Re')
425 subplot(4,4,12);
426 hold on
427 plot(time,A_ls,'*'), title('A_ls')
428 subplot(4,4,12);
429 hold on
430 plot(time,T_drop_previous,'*'), title('T_drop_previous'
    )
431 subplot(4,4,13);
432 hold on
```

```
433 plot(time,v_previous,'*'), title('v_previous')
434 subplot(4,4,14);
435 hold on
436 plot(time,d_previous,'*'), title('d_previous')
437 subplot(4,4,15);
438 hold on
439 plot(time,Yfs,'*'), title('Yfs')
440 subplot(4,4,16);
441 hold on
442 plot(time,Pfs,'*'), title('Pfs')
443 %}
444 end
445
446 function [z,p_alfa,time_injection2] = EOQ(theta,
      mass_injected_rate,Q_lv,evap,m_wall,
      burned_fuel_fraction,Q_heat,dt,angle_spark)
447 global time
448 global time2
449 global p
450 global flag
451 global flag2
452 global flag3
453 global t_wall
454 global time_injection
455 global tr
456 global tj
457 global mass_not_evaporated
458 k=0; K=0;
459 %variables
460 v=Q_lv; alfa=0;
461 %alfa=evap;            %alfa coefficient
462 %beta=Q_heat;          %beta coefficient
463 time=time+dt;         %time from injection to spark
464 time2=time2+dt;       %time of simulation for control
465 e=2.71828182;
466 Q_rate=0; z=0; w_lambert=0;
467 angle_spark=720+angle_spark;
468 %inizialization of global variables
```

```
469 if 20 < theta && theta < 60
470     p=0;                    %total fuel injected
471     t_wall=0;               %time at start of wall wetting
472     tr=0;                   %duration of wall wetting
473     tj=0;                   %duration between start of
            injection and spark
474     time_injection=0;
475     flag=0;
476     flag2=0;
477     flag3=0;
478 end
479 %injection
480 if mass_injected_rate>0 && flag==0
481     time=0;                 %duration between start of
            injection and spark
482     flag=1;
483 end
484 %time from wall wetting
485 if m_wall>0 && flag2==0
486     t_wall=time;            %time at start of wall wetting
487     flag2=1;
488 end
489
490 %spark
491 if theta>angle_spark && flag3==0
492     flag3=1;
493     tj=time;            %duration of wall wetting
494     tr=time-t_wall;     %duration of injection
495 end
496
497 if mass_injected_rate>0
498     p=p+mass_injected_rate*dt;                  %total
            mass injected
499     time_injection=time_injection+dt;       %duration
            of the injection
500 end
501 %between spark and combustion
502 if theta<angle_spark && p>0
```

```
503         mass_not_evaporated=p-evap*dt;
504         if     mass_not_evaporated<0
505             mass_not_evaporated=0;
506         end
507     end
508
509     if  p==0  ||  tr==0  ||  tj==0
510         z=-1/e;
511     else
512     %Parameters for the input of the lambert W fuction (z)
513     alfa=mass_not_evaporated/tr;
514     beta=Q_heat/(Q_lv*tj);
515     K=burned_fuel_fraction*p;
516     k=(K/v)*e^(alfa*tr+beta*tj);
517     z=((alfa*k)/(p*e))-(1/e);
518     end
519     if  alfa==0
520         p_alfa=0;
521     else
522     p_alfa=(p/alfa);
523     end
524     end
```