2014

Investigation of Emission Characteristics during Low Temperature Combustion using Multivariate Adaptive Regression Splines

Mario Velardi
West Virginia University

Investigation of Emission Characteristics during Low Temperature Combustion using Multivariate Adaptive Regression Splines

by

Mario Velardi

Dissertation Thesis submitted to the
College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy
in
Mechanical Engineering

Mridul Gautam, Ph.D., Chair
Mario Perhinschi, Ph.D.
Andrew Nix, Ph.D.
Hailin Li, Ph.D.
Arvind Thiruvengadam, Ph.D.
Lorenzo Mariti, Ph.D.

Department of Mechanical and Aerospace Engineering

Morgantown, West Virginia
2014

ABSTRACT

Investigation of Emission Characteristics during Low Temperature Combustion using Multivariate Adaptive Regression Splines

Mario Velardi

Exhaust emissions from diesel engines operating in a low temperature combustion (LTC) regime are significantly affected by fuel composition and injection strategy. The starting point of this study is a collection of data correlating injection system parameters, and fuel characteristics, to response parameters such as engine-out emissions (oxides of nitrogen (NOx), total particulate matter (TPM), carbon monoxide (CO), hydrocarbons (HC)) and brake thermal efficiency (BTE).

The purpose of this work is to develop a statistical analysis tool to assist the emission analyst in modeling problems in which a response of interest is influenced by several variables and the objective is to optimize this response. The experimental data produced during LTC operation have been analyzed using an approach commonly known as Response Surface Methodology (RSM). Since the system under study may be responding to hidden inputs that are neither measured nor controlled, regression analysis must be performed via a flexible procedure. The methodology that will be used in this sense is called Multivariate Adaptive Regression Splines (MARS), which allows to approximate functions of many input variables given the value of the function at a collection of point in the input space.

Data was collected at West Virginia Universitys Engine and Emissions Research Laboratory for the project CRC AVFL-16. The test engine was a turbo-charged GM 1.9L operated in the LTC mode utilizing a split injection strategy. Main and pilot SOI timing and fuel split were varied per a 5 X 3 X 3 full factorial design. Advanced Vehicle Fuel Lubricants (AVFL) Committee of the Coordinating Research Council (CRC) defined a matrix of nine test Fuels for Advanced Combustion Engines (FACE) based on the variation of three properties: cetane number, aromatic content, and 90 percent distillation temperature. The experimental data was used has a platform for the code development, and for its validation.

Using multivariate data analysis is not only useful in visualizing correlations that otherwise would be hidden by the large amount of experimental data points, but it is also capable to predict the behavior of those points inside the domain where no data are available. As suggested by the name this is a regression methodology capable of adapting the shape of the regression splines to the data analyzed. Validation datasets which were independent of the calibration datasets were used to check the accuracy of the model predictions.

Contents

List of Figures

List of Tables

Nomenclature

ANOVA	Analysis of variance
ATDC	After top dead center
BMEP	Brake mean effective pressure
BTDC	Before top dead center
CA50	50 percent mass fraction burned
CFR	Code federal regulations
CI	Compression ignition
CLD	Chemi-luminescence
CLD	Chemiluminescence detector
CN	Cetane number
CO	Carbon monoxide
CR	Compression ratio
CVS	Constant volume sampling
DAQ	Data acquisition system
DMS	Differential Mobility Spectrometer
EEPS	Exhaust emissions particulate sizer
EERL	Engine and emission research laboratory
EGR	Exhaust gas recirculation
FACE	Fuels for advanced combustion
FID	Flame ionization detection
GCV	Generalized Cross Validation
HC	Hydrocarbon
HFID	Heated flame ionization detector
HPN	Heptamethylnonane
HTHR	High temperature heat release

IC	Internal combustion
LOF	Lack of fitting
LTC	Low temperature combustion
LTHR	Low temperature heat release
MARS	Multivariate adaptive regression splines
MS	Mass spectrometry
NDIR	Non-dispersive infrared
NDIR	Non-dispersive infrared
NOx	Nitrogen oxide
OEM	Original equipement manufacturer
PID	Proportional-integral derivative
PM	Particulate matter
PRF	Primary reference fuels
PRR	Pressure rise rate
RSM	Response surface methodology
RSS	Residual sum of squares
SOI	Start of injection
SOI	Start of injection
SSV	Subsonic venturi
STD	Standard deviation
T90	Distillation temperature
TDC	Top dead center
UNIBUS	Uniform Bulky combustion System
VTG	Variable turbine geometry

Introduction

Diesel engines represent a convenient choice in terms of power density, reliability and durability compared to the gasoline counterpart. The combustion process in diesel engines on the other hand contributes to significantly high PM and NOx emissions , while HC and CO are relatively low. Diesel exhaust gases were identified in the past to be a serious threat to human health and the environment, and regarded as the main source of air pollution.On 17 October 2013, the World Health Organizations International Agency for Research on Cancer (IARC) announced that it has classified Outdoor Air Pollution as carcinogenic to humans [1]. After a review of available literature, IARCs expert group concluded that there is sufficient evidence that exposure to outdoor air pollution causes lung cancer [2]. They also noted a positive association with an increased risk of bladder cancer. Particulate matter, as a major component of outdoor air pollution, was evaluated separately and was also classified as carcinogenic to humans.

Due to tightening emissions regulations, both within the US and Europe, including concerns regarding greenhouse gases, next-generation combustion strategies for diesel engines have drawn increasing attention during recent years. One way to minimize NOx emissions is to limit the in-cylinder temperature during the combustion process by means of fuel injection strategies. However, fuel chemistry plays a significant role in the ignition delay; hence, influencing the overall combustion characteristics and the resulting emissions. Much of the existing technical literature has focused on the interaction of fuel properties and conventional diesel combustion, whereas studies of fuel property effects on advanced combustion are less prevalent.

The present study analyze the data collected during the project AVFL-16 [3]. These data refer to a light duty compression-ignition engine, operated at a fixed engine speed and load. Several split

injection strategy were investigated by varying the start of fuel injection and fuel injection quantity for two injection events. The fuel chemistry was taken into account by operating the engine with a matrix of nine different fuels. The matrix of nine fuels for advanced combustion engine research was defined by the Fuels for Advanced Combustion Engines (FACE) Working Group of the Advanced Vehicle, Fuel, and Lubricants Committee (AVFL) of the Coordinating Research Council (CRC) [4]. Development of the FACE diesel fuel matrix centered around three important fuel characteristics: auto-ignition quality, boiling range, and chemical composition. Respectively, these characteristics are represented by CN, T90, and AC.

The analysis of the data collected is particularly challenging due to the nature of the problem. Various engine operating parameters have conflicting effects on the NOx and PM emissions, for example retarding the injection timing to reduce NOx production will result in higher soot levels. The variations of NOx, and PM emissions introduced above involve tradeoffs that make achieving this goal especially difficult [5]. Considering also different fuel characteristics makes the phenomenon even more complex to analyze. Hence, only a systematic multivariate study can provide a clear understanding of the combustion characteristics of the engine. Such multivariate problems require a continuous quantitative mathematical model to link the factors to responses. In engine applications the relationship between the design or operating factors and the responses is highly nonlinear. The underlying mechanisms in most cases cannot be condensed to one equation that still reflects the physical laws and can be used as the objective or constraint function. Moreover, sometimes the true functional relationships or physical models are not even understood or available, and the governing equations of the processes are simply lacking.

A viable solution to approximate the underlying process is to create an empirical model. Such model does not contain any physical meaning; it is instead obtained through regression of the available data. This approach to the problem is commonly referred to as response surface methodology (RSM). This is a procedure in which the output variables are called responses and the independent input variables are called factors. The response function forms a surface or hyper-surface (if there are more than two factors) in the factor space. RSM refers to the analysis to be performed once that an emulator model is obtained by data fitting. The determination of an emulator model is not a trivial task in the case of multivariate, nonlinear problems.

Commonly used linear regression techniques fail to assess the actual value of the operating parameters in engine testing. This can be considered as one of the causes why the RSM approach is not a standard analysis procedure for emission testing. In the present study a code based on MARS algorithm, described by Friedman [6], has been developed in order to describe engine performance and emission characterization. The main advantage of MARS is that this is a nonparametric regression modeling procedure. Unlike better known linear regression techniques, MARS does not assume that coefficients are stable across the entire domain of each variable and instead uses splines to fit the response. By combining classical linear regression, and mathematical construction of splines, MARS is ideally suitable for problems with high input dimensions where the course of dimensionality would likely create problems for other techniques. This method has been successfully employed for various prediction and data mining applications both in recent years [7].

The use of multivariate data analysis is not just confined to visual correlations of hidden data, but also to predict behavior of data points within a domain. The MARS algorithm used for analyzing the data will be developed in this study, and the development of the algorithm will constitute a major portion and a significant outcome.

Objectives The focus of this work is to produce a statistical tool specifically designed to support the analysis of empirical data related to emissions and combustion. The main objective is to develop a regression tool capable to accommodating up to 6 engine and fuel parameters. In particular, the tool will address highly nonlinear high-dimensional problems and background noise, maintain a low computational cost and offer ease of interpretation. The analysis support code has to evaluate an equation for each emission or performance output that needs to be studied.

Once that the regression equations are proved to be reliable in describing the data, meaning that the fitting of the data is considered acceptable, these can be used to determine the importance of the governing factors. The evaluation of the importance of each factor over the response is both quantitative and graphical. The quantitative analysis is obtained by recasting the regression equation in a form that will allow to separate the contribution of each factor and their join interactions, the output of this analysis will be an index that will quantify the importance of the given independent variable (or group of variables) in relation to the given response. The data available on

LTC have been used has a platform to develop the analysis code described in this work. The main reasons that led to the choice of a set of data related to LTC, can be summarized in the following two points:

- A large set of data was available, in which multiple factors have been varied and different responses have been sampled. This offers the opportunity to challenge the regression code in a high dimensional domain, where the independent variables have conflicting effects on the responses. Moreover having to perform regression for several types of responses is useful in determining the flexibility of the modeling capability for different phenomenon.

- LTC represents a particularly challenging field of operation, characterized by high nonlinearity in the relationship between injection and fuel factors with emission responses. Considering that the purpose of the code described in this work will be to assess combustion and emission problems, proving its capability in analyzing LTC data will guarantee the applicability of the same method to conventional combustion problems.

The given data set has been used to perform an investigation on LTC and specifically the code described in this work determined regression equations correlating engine and fuel parameters to emission and performance values. In order to retain the investigation successful, and hence prove the capability of RSM has analysis methodology, the following specific objectives should be meet:

- To show the existing interaction between the fuel chemistry and the injection strategies, to further understand the phenomena that dominate Low Temperature Combustion.

- Represent the engine test data collected as surface plots that underline the correlations between the single variable and the output, and at the same time show the existing interactions between the different variables.

- As knowledge is gained about the response surface, major interest can be directed toward regions that appear to have greatest potential in terms of emission reduction. This will ultimately lead to the determination of the optimal operating conditions to minimize emissions while maintaining an acceptable thermal efficiency.

Literature Review

Introduction

Conventional diesel combustion is broadly used because of its high thermal efficiency, at the cost of high emissions in terms of PM and NOx. The significant NOx production is a direct consequence of the high temperature zones in the flame due to the necessity of spontaneous ignition which characterize diesel engines. Particulate matter formation instead is due to heterogeneous air fuel mixtures, which in turn leads to locally fuel rich zones. The majority of energy released during conventional combustion is at high temperatures, as a consequence of a diffusion flame [8]. Cooling the flame reduces the formation of NOx at the expenses of higher PM , HC, and CO production.

Studies have attempted to determine which is the minimum temperature required to achieve complete combustion [9]. Flyn et al. performed a study where the fuel specific NOx production was measured at different levels of exhaust gas recirculation (EGR). Their results show a decrease in NOx proportional to increased EGR rates, the limit is reached when combustion starts to deteriorate. The authors pointed out the minimum peak temperature for conventional diesel combustion is 2300K.

Approaches that allow engines to successfully operate at combustion temperatures lower than those encountered in conventional diesel are usually referred as LTC. Work on this family of strategies proved that the NOx-PM trade off curve can change dramatically at low temperature levels Figure 1 [10]. Altering the engine parameters that result in lower NOx and soot production will invariably result in lower combustion temperatures. The main idea behind LTC is to produce spontaneous ignition throughout the combustion chamber at low temperature, i.e. by means of very

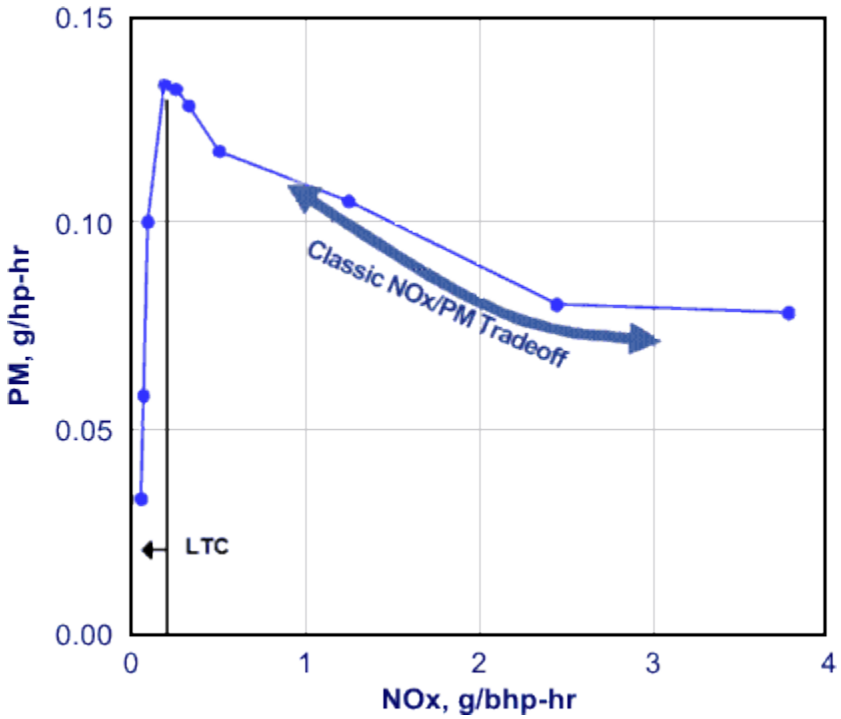


Figure 1: NOx PM trade off curve [10].

lean homogeneous air-fuel mixtures. The low cylinder temperature is useful in reducing NOx and soot production, but, since it involves slow oxidation reactions that cause misfiring and quenching, will result in higher HC and CO emissions compared with conventional diesel combustion.

Low Temperature Combustion

In order to generate the conditions to perform LTC an increased premixing between air and fuel must be achieved, compared to conventional combustion. Higher levels of premixing can be obtained by increasing the injection delay, which can be controlled by increasing the amount of EGR. Longer ignition delay allows for more time available to fuel and air mixture to properly homogenize.

Start of Injection Timing One common method of promoting better mixing of the air and fuel charge is to advance the start of injection (SOI) timing. This will increase the homogeneity of the mixture allowing more time for air and fuel to mix. In conventional combustion NOx emissions increase with advanced SOI, but researchers have demonstrated that advanced SOI timing pared

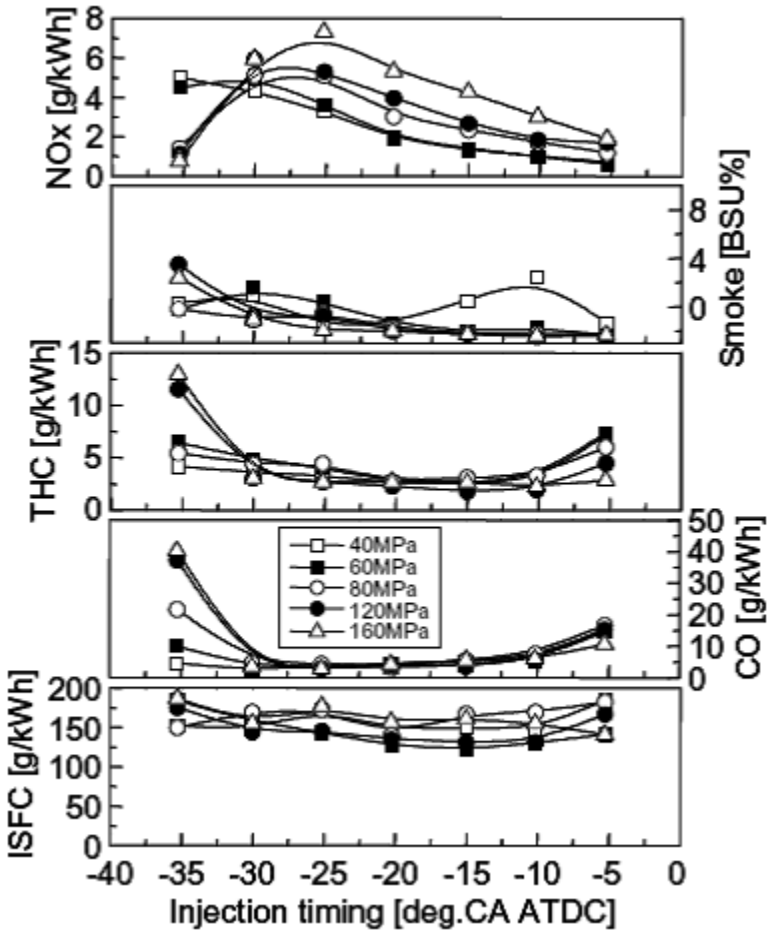


Figure 2: Emission characteristics at different injection timings [11]

with increased injection pressure can reduce NOx emissions for SOI timings greater than 30° before top dead center (BTDC) [11]. This effect is likely a result of reduced diffusion flame zones, due to the absence of near stoichiometric air and fuel mixtures. Figure 2 shows the trend of NOx, CO, HC and fuel consumption with varying injection timing for different rail pressures. A decrease in NOx is obtained by anticipating the SOI timing at high pressure as a consequence of low temperature combustion. Marked increase in CO and HC is found under the same condition. The early fuel injection causes the spray interaction with the cylinder liners or piston walls; the result is wall quenching with resulting fuel problems and high concentrations of HC in the exhaust. Furthermore low temperature combustion is a major cause for high CO emissions.

Fuel Split Modern compression ignition (CI) engines equipped with electronically controlled direct fuel injection system commonly use multiple pulse injections. The basic idea is to carry out a pilot injection of a small quantity of fuel before the main injection event. The primary advantage in using pilot injection is the ability to retard the main injection further than it would be possible with a single injection event. This is related to the reduction in ignition delay consequent to a higher temperature in the combustion chamber at the moment of the main injection. An important parameter in this sense is the time interval between the pilot and main injection event. When this time interval is too long the products of pilot injection mix sufficiently with the ambient gas in the cylinder, consequently no difference is produced with respect to the single injection case. On the other hand, when this interval is shorter a local high temperature zone is created near the fuel nozzle with a consequent shorter ignition delay [12]. It is still important to maintain a sufficient separation between pilot and main injection to ensure the main injection occurs after the injection delay of the pilot event. Under a NOx perspective the pilot injection can produce an effect similar to the one of an internal EGR, especially where the quantity of fuel injected during the pilot event is higher than 10% of the total fuel [13]. The burned gas produced from the combustion of the pilot injection will dilute the concentration of oxygen inside the combustion chamber for the main injection, acting basically like an EGR system.

Hasegawa and Yanagihara explored the effects of a multiple injection strategy on a concept engine named Uniform Bulky Combustion System (UNIBUS) [14]. During the development of the UNIBUS concept, a number of tests were performed to compare and quantify the effects of a double injection strategy with varying SOI timing versus a conventional diesel combustion strategy. For the double injection tests, the main SOI was fixed at 13° after top dead center (ATDC) while the pilot SOI was varied. The injection volume of fuel per cycle during the double injection strategy was held at 15 mm^3/st for both injection events. The plot of brake mean effective pressure (BMEP) in Figure 3 shows that the UNIBUS strategy, as well as the majority of double injection strategy tests, are capable of achieving a BMEP close to that of conventional combustion with the same fuel quantity injected and while producing significantly lower NOX and smoke emissions. The main idea behind the UNIBUS system is to disperse fuel in small droplets that, upon evaporation, would form small fuel rich pockets uniformly distributed. An adequate spacing of these pockets will allow

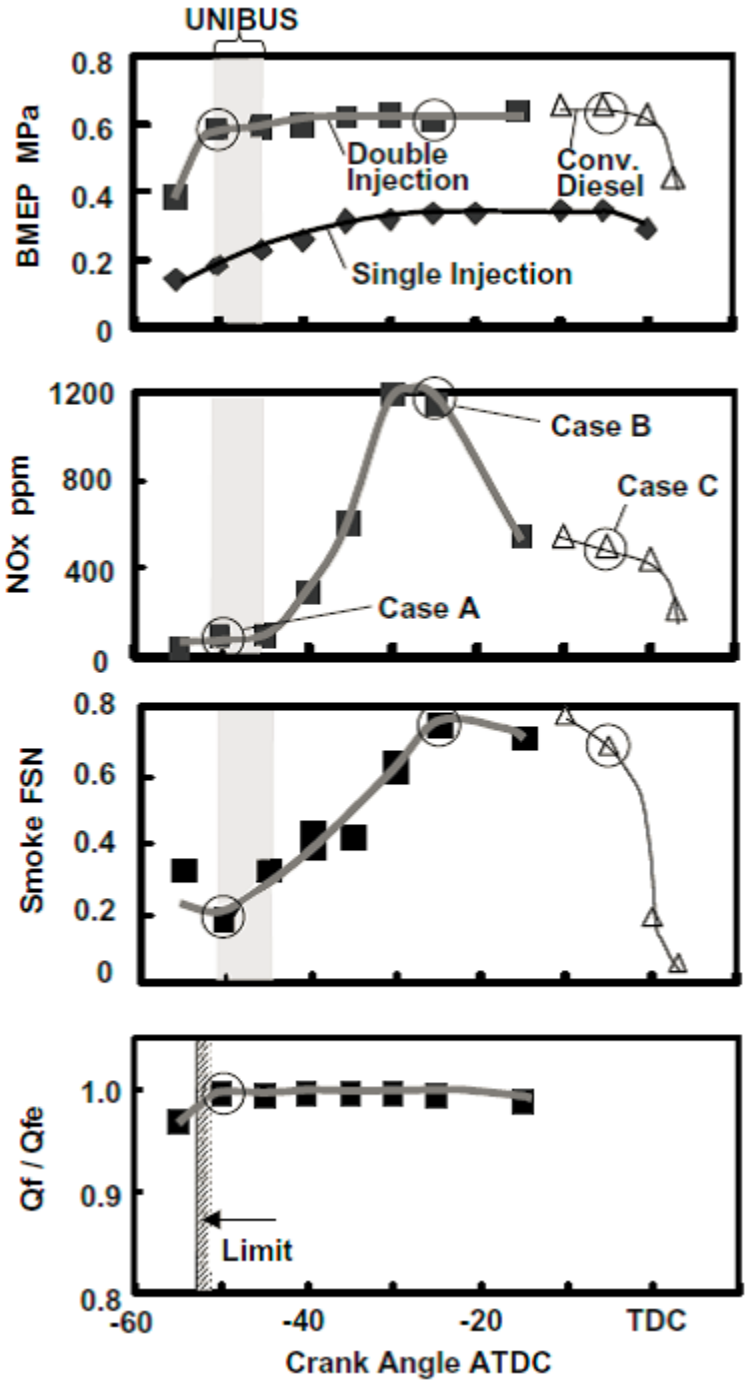


Figure 3: Performance and emissions comparison at different SOI [14]

a first low temperature heat release (LTHR) stage in the compression stroke, during which the combustion of each pocket will not interact with the adjacent ones. Under these conditions the LTHR stage will be followed by a pause before the rest of the fuel is consumed. Further heating of the mixture, due to compression, will then produce a second more rapid high temperature heat release (HTHR) stage of combustion. The objective is to produce a premixed lean combustion during the HTHR that complete the oxidation of the remaining fuel hence producing little NOx and soot. An important feature to achieve the described combustion is early injection. It allows adequate time for LTHR to complete before the beginning of the second stage of combustion. It also ensures for maximum fuel penetration and dispersion as a consequence of the low density inside the cylinder at these early injection timings.

A way to promote the desired combustion process also at high engine loads, where early injection is not convenient, is using pilot injection. It was discovered that with the injection of about 30% of the fuel 30° BTDC, combined with a second injection around TDC, the combustion process remain premixed allowing for the tipical advantages of LTC without deficit in engine output [15]. The fuel injected in the pilot event undergoes LTHR and the main event will complete the combustion in the HTHR stage, as shown in Figure 4.

Figure 4 shows how the pilot injection allows postponing the main event, which in turn improves the premixing of air and fuel. Injection timings of the main fuel as late as 13° ATDC have been reported [14].

Control of combustion Early injection diesel LTC combustion has not been employed yet in production engines, despite of its low NOx and PM emissions, principally for the difficulty to control combustion under transient modes. Since the start of ignition is determined by in-cylinder conditions, there is no direct control on the combustion phasing. There are several indirect ways to influence the start of combustion, as excess air ratio, compress ratio, EGR rate and intake air charge temperature. However, those parameters are difficult to tightly control under transient modes.

High load limit High load conditions are critical for LTC combustion; this constrain is related to the premature start of heat release during compression stroke. A common approach to this problem is the so called dual mode operation, which consists in a switch between advanced combustion and

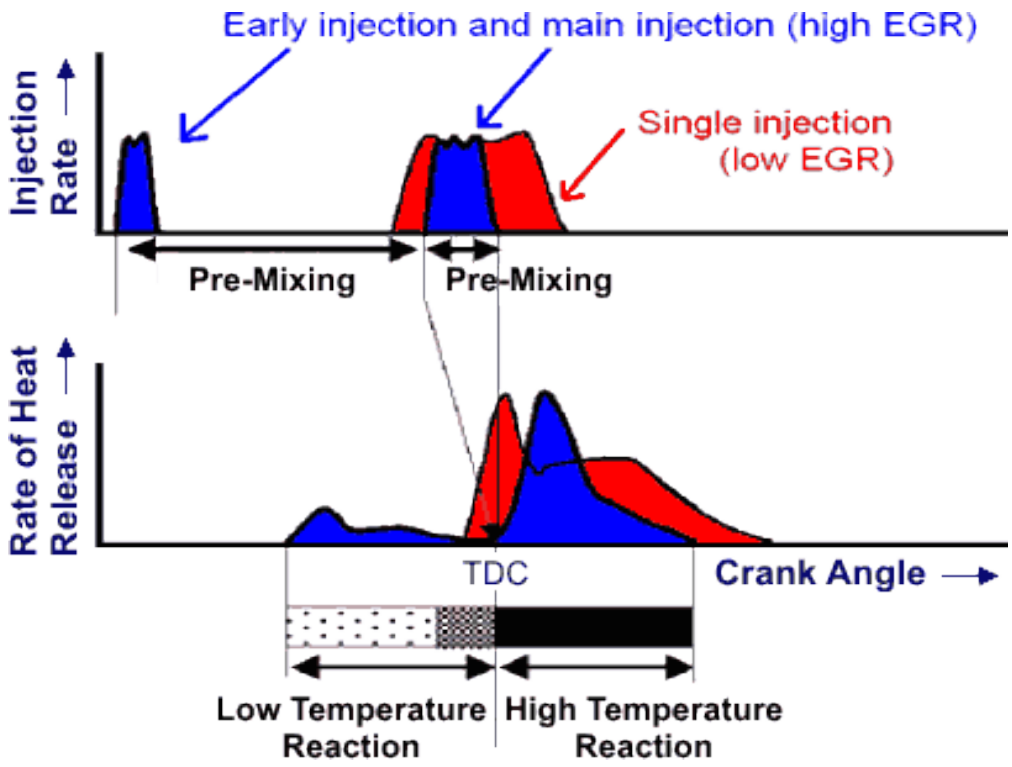


Figure 4: Injection Rate and Rate of Heat Release of Conventional Combustion vs. UNIBUS Combustion [14]

conventional combustion. It is still desirable to be able to perform a broad range of loads with the advanced combustion in order to avoid frequent transition between the two modes of operation. Several paths are viable to expand the range of operation of advanced combustion during high loads, for example applying high cooled EGR can postpone the ignition timing to crank angles near TDC as shown in Figure 5 [16].

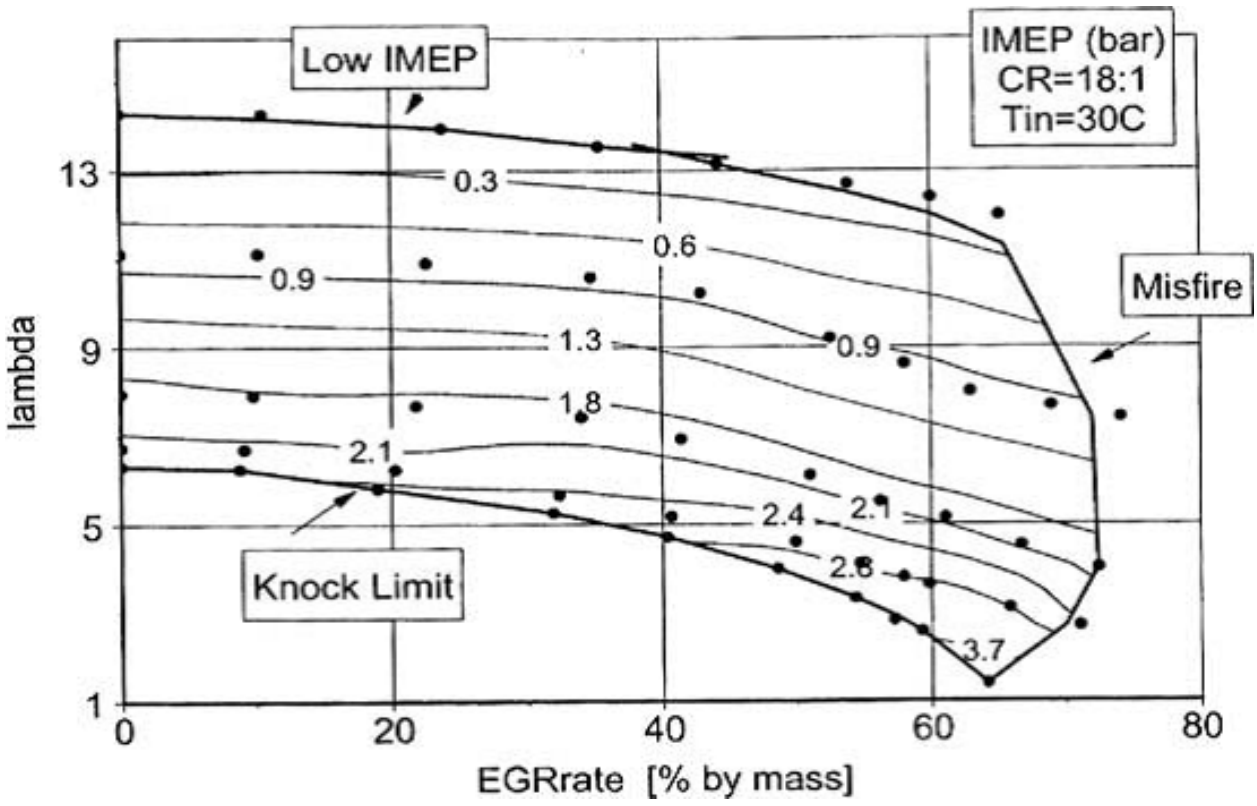
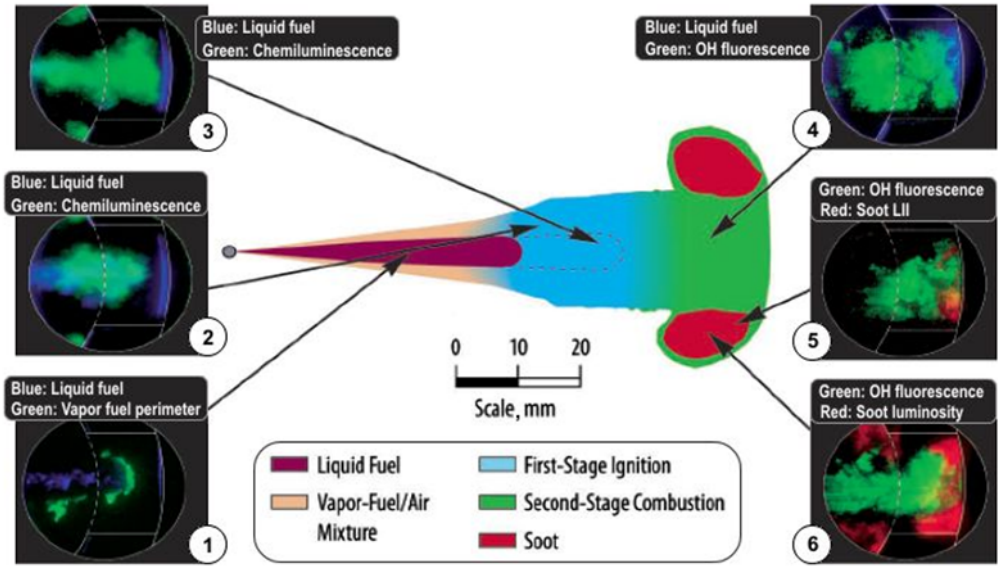


Figure 5: Operating region and IMEP for a compression ratio of 18 and intake charge temperature of 30° C [16]

Conceptual Combustion Model Results obtained through optical engine testing are useful in visualizing of the combustion takes place during LTC [17]. The results showed in Figure 6 are obtained on a heavy-duty diesel engine operated with a single early fuel-injection and with 12.7% intake oxygen to simulate EGR. Panel 1 in Figure 6 shows how the low ambient temperature and density during the early-injection condition produces a long liquid jet penetration (blue) before a clear separation with the vapor phase becomes evident. The jet penetration during LTC can reach a length which is twice the one that occurs in conventional diesel engine, this may impinge in-cylinder surfaces and potentially degrade combustion efficiency and emissions. Panel 2 shows the vaporization due to the energy release as a consequence of the start of ignition, it proceeds up to panel 3. As the subsequent premixed combustion commences (panel 4), fluorescence appears in the laser sheet indicating that the mixture is almost stoichiometric. Panel 5 and 6 shows the areas where soot formation actually happens, i.e. in those areas where the fuel-air equivalence



1. Under the low-density conditions of early injection, liquid fuel (blue) penetrates much farther, even as it vaporizes (green).
2. Chemiluminescence from ignition reactions (green) envelopes the region containing liquid fuel (blue).
3. As the chemical energy of the fuel is released (green), the liquid fuel (blue) rapidly vaporizes.
4. OH (green) fills the jet cross-section shortly after ignition, indicating leaner mixtures than conventional diesel combustion.
5. Soot (red) is first detected near the head of the jet, in regions that are deficient in OH (green).
6. The strongest soot luminosity (red) is often observed in the roll-up vortices at the head of the jet, enveloped by OH (green).

Figure 6: Schematic diagram of LTC combustion [17]

ratio is greater than 2. The presence of soot in the head vortex of the jet indicates that the mixing is poorest in these regions.The long jet penetration contributes to a better mixing between fuel and air, which is the key feature leading to lower PM emissions compared to conventional combustion. The combustion is also more evenly distributed, leading to lower peak temperatures and consequently lower NOx emissions.

Fuel Influence on Combustion

Fuel properties are a key parameter in the combustion process, and consequently have a profound impact on the performance and emissions of engines employing LTC. Methods of control and necessary modification of engine hardware can be solely dependent on the properties of a selected fuel. This has resulted in a considerable amount of research to determine which fuels are best suited for advanced combustion [4]. Especially the capability of the fuel to mix with air and how easily the mixture will ignite affect the combustion. The ignitability is related to the chemical kinetic characteristics of the fuel when it mixes with air. CN is commonly used as a measure of fuel's ignitability. The fuel volatility is the main index of how the fuel is able to vaporize in the air, combined with the temperature of the air flow interacting with the fuel. Higher levels of volatility guarantee a better mixing, and this parameter is described with distillation characteristics.

Auto Ignition In LTC engines the delay between injection of fuel into the air and ignition is much longer than that typically encountered in conventional diesel engines. A longer delay allows for higher rate of mixing and homogenizing, at the same time the temperature of the mixture reaches a point where it can auto-ignite making the process of conversion of the fuel air mixture into combustion product particularly rapid. The ignition delay is the main parameter of control for mixture homogeneity and combustion phasing, which has a strong impact on performance and emissions. The factors that mostly affect ignition delay are the temperature and pressure field, and the fuel characteristics. Figure 7 shows a comparison between diesel (CN=54) and gasoline (RON=95) engines in terms of heat release profiles [19]. It is interesting to notice how a LTHR stage is experienced by the diesel engine at 25°BTDC while gasoline does not. According to Shibata and Urushiahara [20] he LTHR stage can be inhibited, in some fuels, due to the following reasons:

1. Some fuels may contain only compounds that form stable radicals upon abstraction and do

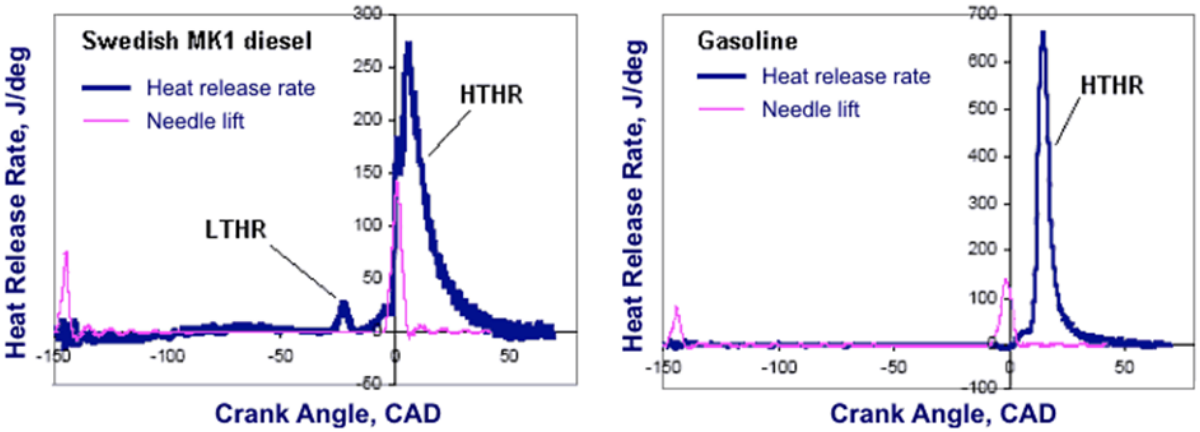


Figure 7: Heat Release Rate comparison between Diesel and Gasoline [19]

not undergo LTHR.

2. Radicals that form fuel components, such as olefins and oxygenetes, react with OH radicals that are important for the start of LTHR.

In most cases, LTHR can be attributed mainly to n-paraffins and to a lesser extent to branched paraffins. Aromatics have a LTHR inhibiting effect while compounds such as olefins and naphthenes both undergo LTHR and have a LTHR inhibiting effect [20]. Fuels that exhibit LTHR include diesel fuels with a cetane number above about 35-40. Fuels with a higher tendency to auto-ignite will generally have a more pronounced LTHR (Figure 8) [21].

While LTHR is not a requirement for successful premixed LTC operation, it can have a significant impact on the HTHR. Impacts include:

- a lower HTHR rates (see Figure 7). The HTHR rate can influence the combustion noise and the maximum load that can be achieved by the engine.
- a lower initial mixture temperature requirement to achieve optimal combustion phasing.
- a need for more advanced combustion phasing to achieve maximum power.
- increased tendency to misfire at retarded combustion phasing.

A fuel with a large LTHR tend to knock easily and is better suited for high speed and low torque operation. Fuels with a small LTHR are less likely to knock, hence they are better suited for low

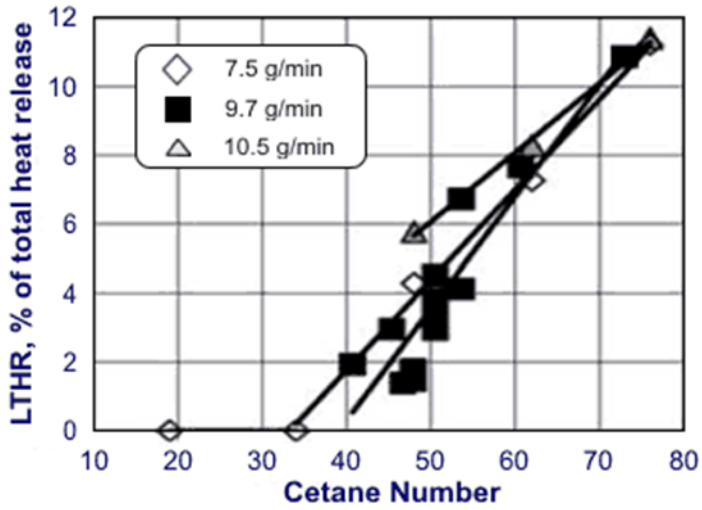


Figure 8: Dependency of LTHR on Fuel Ignitability for Different Fueling Rates [21]

speed and high torque operation [22].

Measures of fuel Ignitability Cetane number is the most commonly used index to correlate well diesel-like fuels and ignition quality. Is widely proved that a decrease in Cetane number produce a delay in the ignition of the air-fuel mixture [23]. Other factors have strong influence on ignition delay and have to be considered when comparing fuel properties, for example [24] showed in their work that the relationship between different Cetane number diesel fuels and ignition delay was not constant at different temperatures and intake pressure. Other researchers [25] showed the ignition delay did not increase with low Cetane number when the engine is running at high loads without pilot injection.

Fuel volatility Fuel volatility is another key parameter in LTC; it has to be high enough to guarantee fuel evaporation before impingement of the combustion chamber surfaces occurs. This explains the difference in behavior that occurs between gasoline and diesel when the start of injection is advanced. Emission strongly increase for early injection strategy that use diesel fuel because this kind of combustible is not enough volatile at low temperature. When the injection start earlier than 50° BTDC the temperature in the combustion chamber is not high enough to vaporize diesel fuels, this brings to impingement on the surfaces with consequent high values of emissions. On the

other hand gasoline fuels high volatility guarantee for vaporization also at the lower temperatures that characterize early injection [26].

Emissions

The main reasons that lead to the study of LTC engines is their capability to lead to significant reduction in thermal NO formation. While total NOx emissions from engines using LTC are lower than conventional diesel engines; it was reported that the fraction of NO_2 in NOx is consistently higher [27].

Soot Describing soot production in internal combustion engines can be extremely complex. Tao et al [28] were able to incorporate a soot generation model into KIVA-3V simulation code to model LTC engine operation with high EGR rates. The results suggested that the model is able to predict soot controlling mechanisms over a wide range of operating conditions including a EGR sweep from 0% to 68% combined with a split injection around TDC. As depicted from Figure 9; sufficiently high EGR rates will significantly lower the soot formation rate by mean of lower temperature combustion.

Increasing moderately EGR rates brings to increase in soot production especially in the final part of the crank motion. The increase is mainly due to lower soot oxidation, which is related to lower temperatures due to EGR flow. At EGR rates in excess of 65% the trend is reversed and a significant decrease in soot production is observed. This is probably due to lower rates of soot formation, as we can deduce observing that both the initial and final decrease has the same magnitude. Increase in the amount of EGR, combined with other measures, is used to lower emission, for example high injection pressure, high intake boost pressure, variable valve timing and variable compression ratio [29].

Carbon Monoxide LTC systems exhibit high carbon monoxide (CO) emissions, particularly at high dilution levels. CO is one of the intermediate species generated from the burning process of hydrocarbon in the fuel, when complete combustion is achieved this is oxidized to CO2. The efficiency of the oxidation depends on the local temperature and oxygen concentration [30].

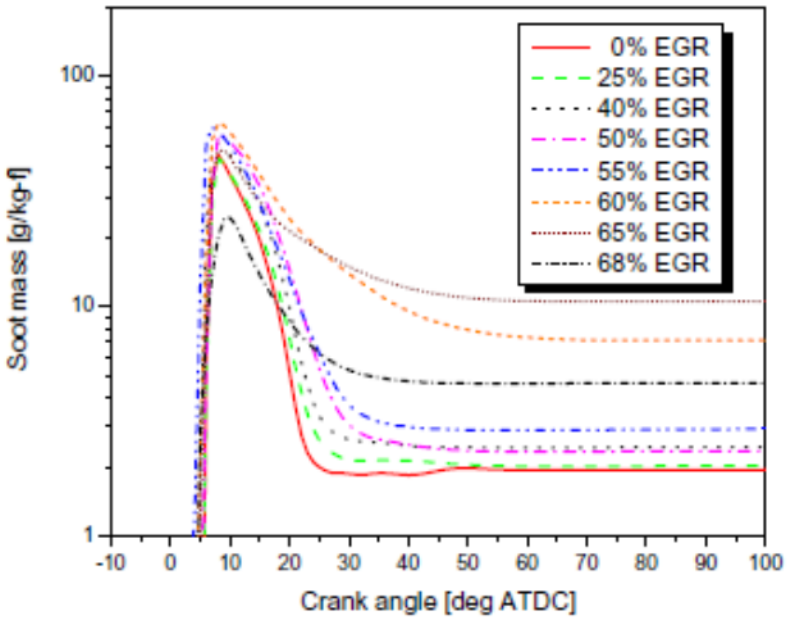


Figure 9: Smoke production for different level of EGR [28]

Hydrocarbons Like for CO the hydrocarbons (HC) produced by LTC are higher than those produced by conventional diesel operations [8]. The factors that cause this phenomenon can be listed has follows:

- The long spray penetration which characterize LTC operation bring fuel to accumulate on the combustion chamber surfaces
- Fuel lean zones inside the combustion chamber are more common in LTC operations rather than in conventional diesel. Fuel in these areas is more prone to escape combustion leading to the production of unburned HC.
- Since the local peak temperatures are lower than conventional diesel, the fuel near the wall does not burn especially at low loads.

Experimental Setup

All measurements in this study were conducted at the Engine and Emission Research Laboratory (EERL) at West Virginia University for the project CRC AVFL-16 [31] [3]. The EERL test cell follows the recommendations outlined in the Code of Federal Regulations (CFR), Title 40, Part 1065 [32]. Figure 10 shows an overview of the EERL sampling capabilities for regulated and unregulated diesel exhaust emissions.

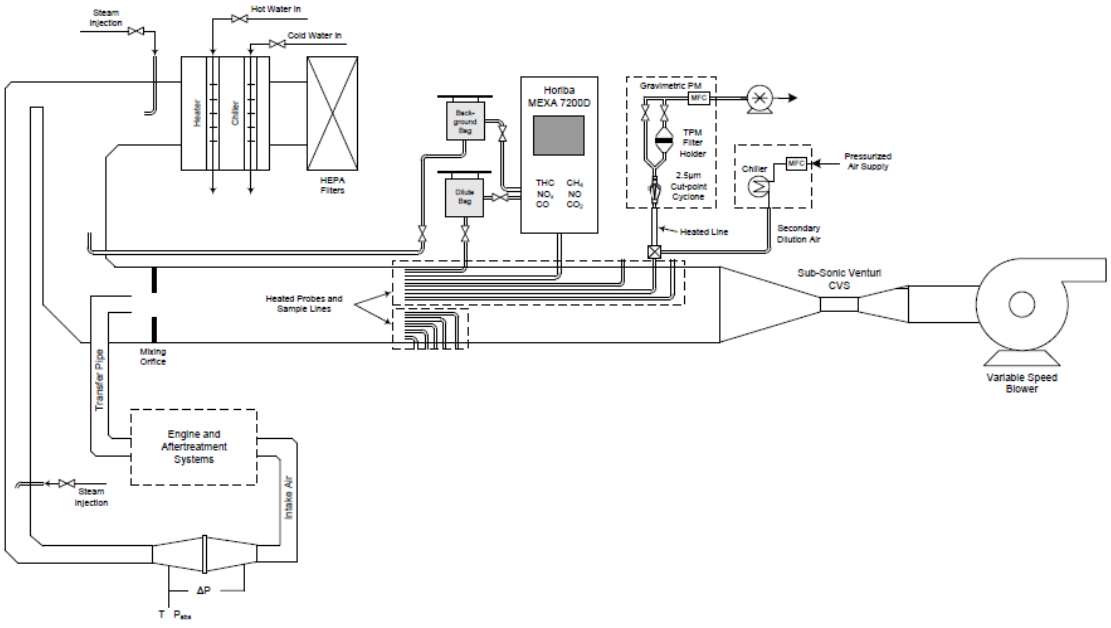


Figure 10: Schematic Overview of EERL Measurement System

The sampling system consists of a variable speed blower and a subsonic venture constant volume sampling system (CVS). Diluted gaseous sampling were analyzed using the Horiba® MEXA 7200D system. Primary data collected by the Horiba system comprised CO and CO_2 acquired by the non-dispersive infrared (NDIR) method, NO and NOx using the Chemi-luminescence (CLD) method, HC by using a heated flame ionization detector (HFID). Particulate matter characterization downstream the aftertreatment system was quantified both using the gravimetric method as outlined in Title 40 CFR Part 1065 [32], and by photo-acoustic method [33]. Diluted exhaust gas was sampled from the main dilution tunnel through a secondary dilution system and subsequently sampled onto a Pallflex 47 mm Teflon filter. Filters were weighed in an environmentally controlled clean room (Class 1000) using a Sartorius microbalance with an accuracy of 1 μg .

Test Engine

The test engine used for this study was an in-line 4-cylinder common rail diesel engine model Z19DTH from General Motors, depicted in Figure 11. The intake air pressure was controlled via a variable turbine geometry (VTG) turbocharger. The engine was instrumented with thermocouples measuring engine lubricant, coolant, intake manifold and exhaust manifold temperatures. Inlet depression, intake manifold pressure and exhaust backpressure were measured as well by pressure transducers.

Since EGR rates that were adopted for the advanced combustion research on this engine were higher than the original equipment manufacturer (OEM) specifications, a larger EGR cooler was fitted to the engine in order to reduce inherent higher intake manifold temperatures. An overview of the test engine specifications are shown in Table 1.

Figure 11: Test Engine GM Z19DTH in the EERL

Type	CDTi Diesel Engine	**Bore**	82mm
Manufacturer	General Motors	**Stroke**	90.4mm
Model	Z19DTH	**Compression Ratio**	17.5:1
Valve Configuration	4 Valves per Cylinder	**Turbocharger**	Garret VGT
Year	2005	**Injection System**	Common Rail
Configuration	In-Line 4 Cylinder	**EGR**	Cooled, External
Displacement	1.9L	**Rated Power**	110 kW at 4000 rpm

Table 1: Test Engine Specifications

Laboratory Instrumentation

This section describes the laboratory instrumentation in the EERL that was used for this study. The exhaust dilution system, gaseous emissions measurement instrumentation and particulate matter sampling system and technique, as well as in-cylinder pressure measurement and the engine and

dynamometer control system are presented.

Constant Volume Sampling Dilution Tunnel A total-exhaust dilution, CVS tunnel, designed to simulate the mixing of exhaust gas with ambient air conditions, maintains a nominally constant total molar flow rate of the diluted exhaust, as outlined in the Code of Federal Regulations (CFR), Title 40, Part 1065, Subpart 140 [32]. To accurately measure and actively control the flow rate maintaining proportional sampling of the exhaust constituents, a subsonic venturi (SSV) flow meter is used, see Figure 12. The SSV was calibrated for a Reynolds number at the throat greater than the maximum Reynolds number expected during testing and used only between the minimum and maximum calibrated flow rates.

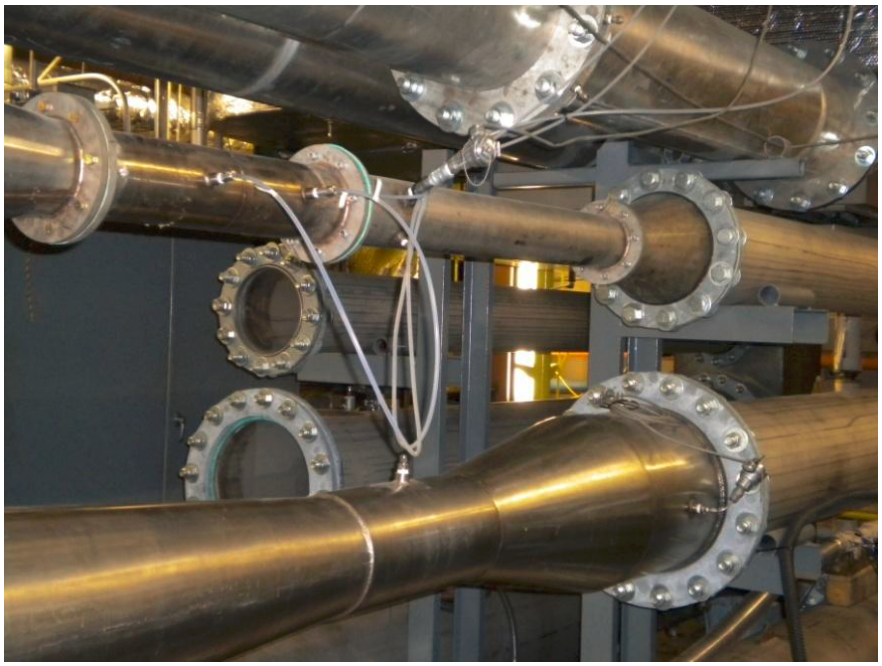


Figure 12: CVS tunnel in the EERL

Gaseous Emission Measurement Diluted exhaust gas emissions extracted from the CVS tunnel were measured continuously using a HORIBA MEXA 7200D gaseous emission analyzer and included HC, CO as well as CO_2 and NO_x and O_2. The same emission characteristics were also sampled in the raw exhaust and intake manifold still using an HORIBA MEXA 7200D NO_x analyzer.

Horiba Automotive Emission Analyzer System MEXA 7200D The Horiba automotive emission analyzer system MEXA 7200D is a modular components system consisting of a main control unit, an interface controller, an analyzer rack with up to five analyzer modules, a heated analyzer module for THC, THC/CH_4, NO_x and $NO/N0_x$ analyzers, a power supply unit, as well as a solenoid supply unit, which routes to zero, span and samples gas to the analyzer modules, and a sample handling system dehumidifying the sample gas and directing it to the analyzers. A separate heated oven unit cart contains up to three heated analyzers (THC, NO_x and CH_4) with heated lines, pumps and solenoid valves. The analyzer system is shown in Figure 13.

Figure 13: Horiba Automotive Emission Analyzer System MEXA 7200D

The analyzer modules measuring the concentration of CO and CO_2 (cold dry sample) use the non-dispersive infrared (NDIR) principle. Infrared energy at specific wavelengths is absorbed by a molecule consisting of different atoms and the degree of absorption is proportional to the concentration at constant pressure [34]. In a NDIR analyzer, an infrared beam is passed through a sample and a comparison cell by a light source. The comparison cell is filled with a gas non-absorbent to infrared radiation (such as nitrogen). Figure 14 shows an example of an NDIR

configuration.

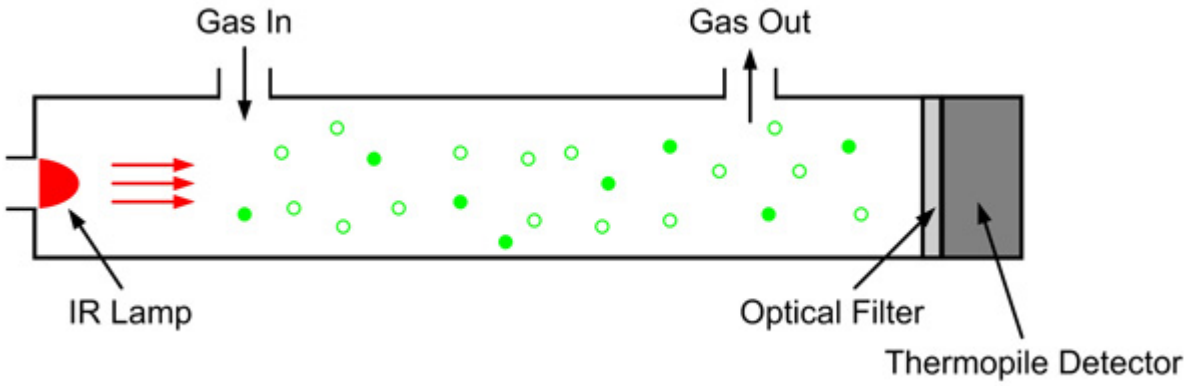


Figure 14: Example of NDIR Configuration

A sealed capacitor type detector, consisting of two cells separated by a movable membrane and filled with the gas to be measured, transforms the displacement of the membrane, which moves as a result of the differential pressure generated by the difference in quantity of radiation each gas received and therefore heated up and expanded, into an electrical output signal. To prevent interference with another gas component absorbing infrared radiation in the same wavelength range, an optical filter in front of the detector is used to eliminate the absorption area of the interfering component [35].

The concentration of total hydrocarbons is measured employing hydrogen (H_2) flame ionization detection (FID) (heated wet sample). Hydrocarbons introduced into a hydrogen flame generate ions, which are proportional to the number of carbon atoms in the sample. This measurement principle is sensitive to almost all hydrocarbon compounds [36]. Figure 15 shows a FID configuration.

The sample gas is mixed with H_2 and directed into the H_2 flame. Ions in the high-temperature area are generated according to the following the reaction:

$$\mathrm{CH}^* + O^* \rightarrow CHO^* + e^- \quad (1)$$

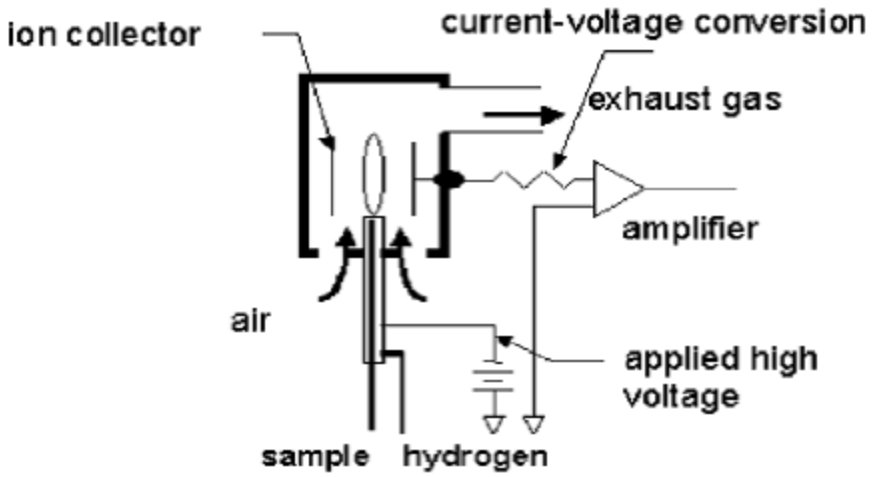


Figure 15: Example of FID Configuration

where the stars (*) denote radicals. A DC voltage is applied on two electrodes in the ion collector which causes a migration of ions towards them and a current can be measured and amplified. Due to the proportionality of the current to the number of carbon atoms, this is a measure of the total hydrocarbons (THC), but no information of different hydrocarbon components can be obtained by this method [36]. The analyzer module measuring NO and NO_x concentration (heated dry atmospheric sample) uses a chemiluminescence detector (CLD). Sample gas containing NO is mixed with ozone gas (O_3) in a reactor to be oxidized into nitrogen dioxide (NO_2), whereas part of the NO_2 is in excited state, releasing excited energy as light (radiation) when returning to the ground state, as shown in Equations 2 and 3:

$$NO + O_3 \rightarrow NO_2^* + O_2 \tag{2}$$

$$NO_2^* \rightarrow NO_2 + hv \tag{3}$$

where the star (*) denotes the NO_2 molecules in excited state. The light released is directly proportional to the NO molecule quantity before the reaction and therefore a measure of the NO concentration. This chemiluminescence signal is detected photo-electrically. Excited NO_2 molecules can also return to ground state without radiation emission due to collision with other molecules (H_2O, CO_2, N_2 or O_2). This interference can be reduced by reducing the pressure in the reaction chamber. Since there is also NO_2 in the initial sample that does not have chemiluminescence, it has

to be converted to NO by means of a NO_x converter. The measurement of the converted NO_2 and the aforementioned NO measurement can therefore be added up to yield the NO_x concentration.

Horiba MEXA-720 NO_x Analyzer Two heated zirconia-ceramic (ZrO_2) sensors were installed in the intake manifold and exhaust pipe directly as parts of MEXA-720 NOx analyzer units (see Figure 16), which are capable of measuring NO_x concentrations, air/fuel ratio, excess air ratio (lambda) and O_2 concentrations simultaneously. The measurement principle of a zirconia sensor is depicted in Figure 51. It is based on the oxygen conducting properties of zirconia [37]. Zirconia is used as an ion pump lowering the oxygen concentration from the sample gas to approximately 10ppm in the first internal cavity, where nitrogen dioxide is reduced to nitric oxide and oxygen. The pump current, depending on the amount of oxygen pumped, is measured and used to calculate the oxygen concentration in the sample gas. Further lowering the oxygen concentration to nearly zero ppm (1ppb) is achieved by an auxiliary oxygen pump (not showed in schema). Nitric oxide is further split into nitrogen and oxygen. A measure of the oxygen generated is the current created by the zirconia ion pump and can be used to calculate the NO concentration, which is an indicator of the NO_x concentration in the exhaust stream, since NO_2 is reduced to NO in the first internal cavity [38]. Intake oxygen concentration was one of the primary controlled operating parameters

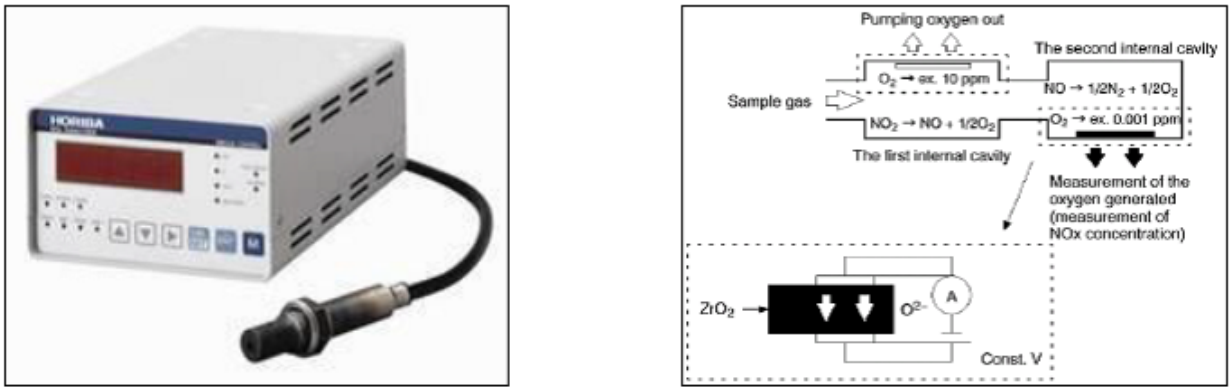


Figure 16: Horiba MEXA-720 NO_x and principle of Zirconia Sensor

in this study and raw exhaust O_2 concentration was used along with intake O_2 concentration to determine actual EGR fractions.

Particulate Matter Sampling

Nanoparticle number concentration and size distributions were determined using the Exhaust Emissions Particle Sizer (EEPS), a spectrometer from TSI Inc. (model 3090) as well as the Differential Mobility Spectrometer (DMS) from Cambustion (model DMS500). Continuous exhaust gas samples were extracted from the CVS tunnel (dilution ratio DR = 10) and routed through a double stage dilution system using ejector type dilutors, see Figure 17. The first stage was maintained at $140°C$ (DR = 6) in order to suppress condensation and particle nucleation phenomena, while the second stage utilized dilution air at ambient temperatures (25C, DR = 11).

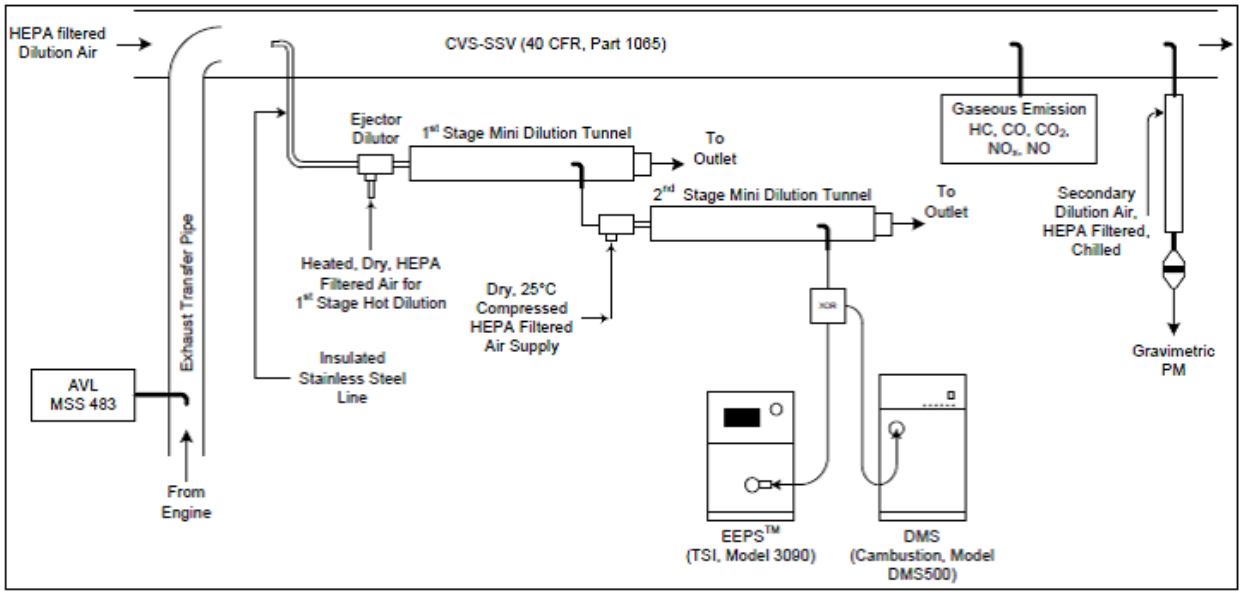


Figure 17: Experimental Setup for Nanoparticle Sampling

Control of Engine Operating Parameters

In order to have full control on the engine operating parameters, an open engine controller from Drivven Inc. was used. The controller is based on National Instruments hardware. Operating parameters such as main SOI, number and duration of fuel injection events, turbocharger boost (by controlling VGT vane position), EGR rate, rail pressure, throttle, and more, were accessed and controlled to obtain advanced combustion regimes.

Laboratory and Dynamometer Control The GM Z19DTH engine was coupled to a Medsker Electric Inc. (MEI) alternate current (AC) dynamometer operated in speed mode (see Figure 18). Engine torque was controlled by means of a proportional-integral-derivative (PID) throttle controller integrated into the laboratorys data acquisition system (DAQ). The EERLs DAQ system is an in-house solution using NI hardware and software developed by CAFEE with high-grade automation capabilities for engine testing as well as calibration and quality control. The DAQ system follows the recommendations outlined in 40 CFR, Part 1065 [32].

Figure 18: Medsker Electric Inc. alternate current dynamometer

Fuel Properties

The Advanced Vehicles, Fuels, and Lubricants committee of the Coordinating Research Council specified and formulated a matrix of nine test fuels for advanced combustion engines (FACE) [4] based on the variation of three properties:

- Cetane number, a measure of ignition quality;
- Aromatic content, a measure of chemistry; and
- The 90 percent distillation temperature, a measure of volatility.

Table 2 displays the nine fuels used in this study.

Property	FACE 4	FACE 1	FACE 3	ULSD	FACE 7	FACE 9	FACE 8	FACE 6	FACE 5
Cetane Number	28.4	29.9	32.0	44.0	44.3	45.0	50.0	53.3	54.2
Aromatic Content (Mass %)	40.7	26.1	50.0	34.7	46.2	37.0	43.5	21.1	22.2
90 % Distillation Temperature ($^{\circ}F$)	639	517	518	582	513	610	648	646	528
Specific Gravity	0.8355	0.8084	0.8401	0.8496	0.8375	0.8465	0.8682	0.8411	0.8086
HC Ratio	1.819	1.956	1.749	1.796	1.773	1.788	1.704	1.871	1.967
Net Heat of Combustion (BTU/lb)	0.8355	0.8084	0.8401	0.8496	0.8375	0.8465	0.8682	0.8411	0.8086

Table 2: Analysis of FACE fuels characteristics

These properties were obtained by fuel blending and determined to be of primary importance to the performance of advanced combustion engines. The target values built up a full factorial statistical design with a center run representing average marketplace values of the design variables. This design theoretically allows investigation of each combination of the fuel properties and the injection strategies. A graphical representation of the design matrix is shown in Figure 19, where target values (in blue) of the three factors at two levels build a design cube. The actual values of the formulated fuels are represented, as well (in red).

Repeatability Study

As part of the AVFL-16 project [3] a study of the repeatability of test data generated from two split injection control strategies was performed with the ULSD fuel. The purpose of this effort

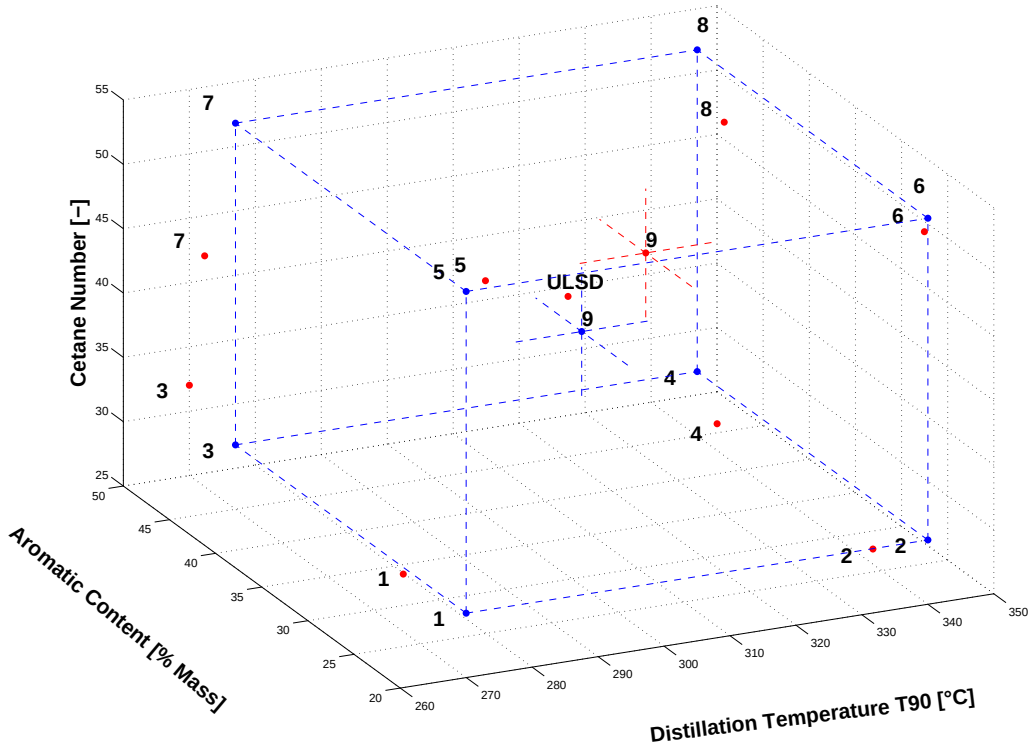


Figure 19: FACE Diesel Fuels Design Matrix: Target (Blue) vs. Actual (red)

was to develop a standard by which emissions and performance changes among the fuels could be attributed to fuel property differences and not to the variability associated with the equipment or control strategy.

The selected test was repeated three times in the morning and evening. This daily routine was repeated for three days providing data for 18 repeats of each test. From this data, the percent difference of the original test and average of the repeated tests was quantified. The standard deviation and coefficient of variation (COV) was also calculated for the repeat tests and is presented in the subsequent section. It should be noted that the original test was conducted four months prior to the 18 repeat tests and consisted of one test.

Characteristic	NOx (g/kW-hr)	PM (mg/kW-hr)	CO (g/kW-hr)	HC (g/kW-hr)	BTE %
Original Test	0.352	104.5	14.5	2.74	29.7
Repeted Test	0.350	88.4	13.47	2.72	29.7
Standard Deviation	0.010	7.73	0.48	0.06	0.3
COV	2.81 %	8.75 %	3.58 %	2.35 %	0.99 %
Original vs Repeated	0.7 %	18.2 %	7.4 %	0.7 %	0.04 %

Table 3: Repetability analysis

Based on the results of this repeatability study, it was determined that any difference in HC emissions or NOX emissions greater than 3% could be deemed significant and thus attributed to differences in the fuels rather than variability associated with the data. While BTE should be greater than 1%. Similarly for CO emissions and soot emissions, a difference greater than 8% and 19% respectively, could also be deemed significant. These values were determined by selecting the maximum between COV and the percent difference between original and repeated study, and then rounded.

Regression Analysis

Introduction

When selecting the engine parameters for testing, it is not always known what impact each will have on performance and emissions, because the underlying function is implicit. Obtaining a regression model plays an important role in the data analysis, providing prediction and classification rules, and data analytic tools to understand the importance of different inputs. This may be a difficult task when the underlying function is non-smooth and highly non-linear. This is often the case when performing an experimental emission characterization in IC engines.

For each data point there is a set of variables that might be denoted as inputs (also called predictors or in general independent variables), which are measured or preset. These have some influence on one or more outputs (also called responses or dependent variables). The goal of regression analysis is to use the inputs to predict values of the outputs. The branch of statistics which study how to obtain prediction functions is also called machine learning or data mining.

This section gives a description of the theory behind the algorithm used in this work, based on [6] and [42]. In the statistical literature plenty of methods are described. In order to motivate why a certain approach have been preferred over others a brief description regarding the theory of machine learning is given in this chapter.

Two Simple Approaches to Prediction

Linear models and Least Square Linear model has been a mainstay of statistics, and still remain the basic tool for many prediction models [43]. Given a vector of inputs X the output Y is

predicted using the following model:

$$\widehat{Y} = \widehat{\beta}_0 + \sum_{j=1}^{p} X_m \widehat{\beta}_j \tag{4}$$

The coefficients of equation 4 can be determined using several methods, but by far the most popular is the method of least squares. The idea is to pick the coefficients that minimize the residual sum of squares:

$$RSS(\beta) = \sum_{i=1}^{n} (y_i - X_i^T \beta)^2 \tag{5}$$

Since this is a quadratic function the solution to the minimization problem always exist, even if might not be unique. The resulting equation will be characterized by the p parameters $\widehat{\beta}$. This method will produce smooth equations, but it does appear to rely on the assumption that the degree of the model is appropriate to fit the data. In the language of machine learning it is described as a low variance, and high bias method.

Nearest-Neighbor Methods The model function $\widehat{Y}$ is determined by the closest observation it he input space to x. The k-nearest method is defined as follows:

$$\widehat{Y} = 1/k \sum_{x_i \in N_k(x)} y_i \tag{6}$$

Where $N_k(x)$ is the set of k closest points to x_i. The output is then the average of the k closest observations to x_i in the input space. The parameter to be selected when utilizing this method is the number of neighborhoods k. It is clear that off the shelf least square methods cannot be used to determine k, since we would always pick $k = 1$. The k-nearest-neighbor procedures do not appear to rely on any stringent assumption about the underlying data, and can adapt to any situation. The drawback of this approach is that every sub region depends strongly on a handful of input points and their positions, and is thus wiggly, unstable, high variance and low bias.

Each method has its own situation for which it works best. The linear models are suited for those scenarios where the data are affected by large variability; hence a general behavior wants to be detected at the cost of lost in local accuracy. Linear methods are often referred as model based, because the general model selection determines the fitting over the entire domain. The nearest neighbors methods are more appropriate when the information deriving from each training data

point is considered very relevant. Hence a locally accurate fitting is necessary, and that is why this family of mythology is often referred to as local. Large subsets of the most popular techniques are variants of these two simple procedures, and often are a combination of the two.

Curse of Dimensionality

Local methods like k-nearest-neighbor seem able to approximate the data optimally, as long as we are able to find a fairly large neighborhood of observations close to every x. It can be proved that with an infinite number of samples the nearest neighbor method will produce the best possible function in term of average squared error. In dealing with high dimensional problems this intuition breaks, as a consequence of the phenomenon commonly refereed to as course of dimensionality [39]. In a p-dimensional problem suppose we are interested in a fraction r of the observations. Assuming that the inputs are uniformly distributed in a p-dimensional unit hypercube, r will also be the fraction of volume captured by our neighborhood. The expected length of the edge we are sampling will be proportional to $r^{1/p}$. So to cover just a small portion of the data in a high dimensional space we need to span on a very wide range of each input variable, as in Figure 20.

Another manifestation of the course is that the sampling density is proportional to $N^{1/p}$, where N is the sample size. Thus in higher dimension the density of the data will be naturally lower. As the complexity of functions of many variables grow exponentially with the dimension, the number of responses necessary to determine those functions accurately grows exponentially too. We have seen that although local methods focus directly on estimating the function at a point, they face problems at high dimensions. It is also possible that local methods are inappropriate in low dimensions cases where a more efficient use of the data can be achieved by structured approaches.

Tree Based Methods

These, conceptually simple, family of methods partition the feature space into a set of rectangles, and then fit a model in each subspace. These methods use an approach which is intermediate between model based, and local methods. The simplest method belonging to this family is the recursive binary partitions method. The input space is first divided into two regions, and the

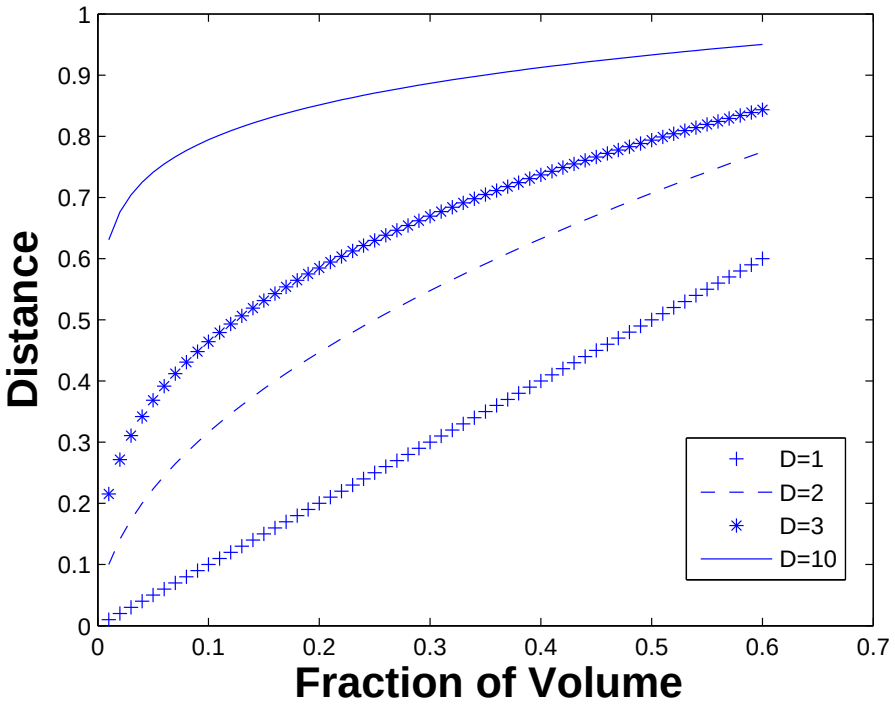


Figure 20: Length of the subcube edge needed to capture a fraction r of the volume of the data, for different dimensions p.

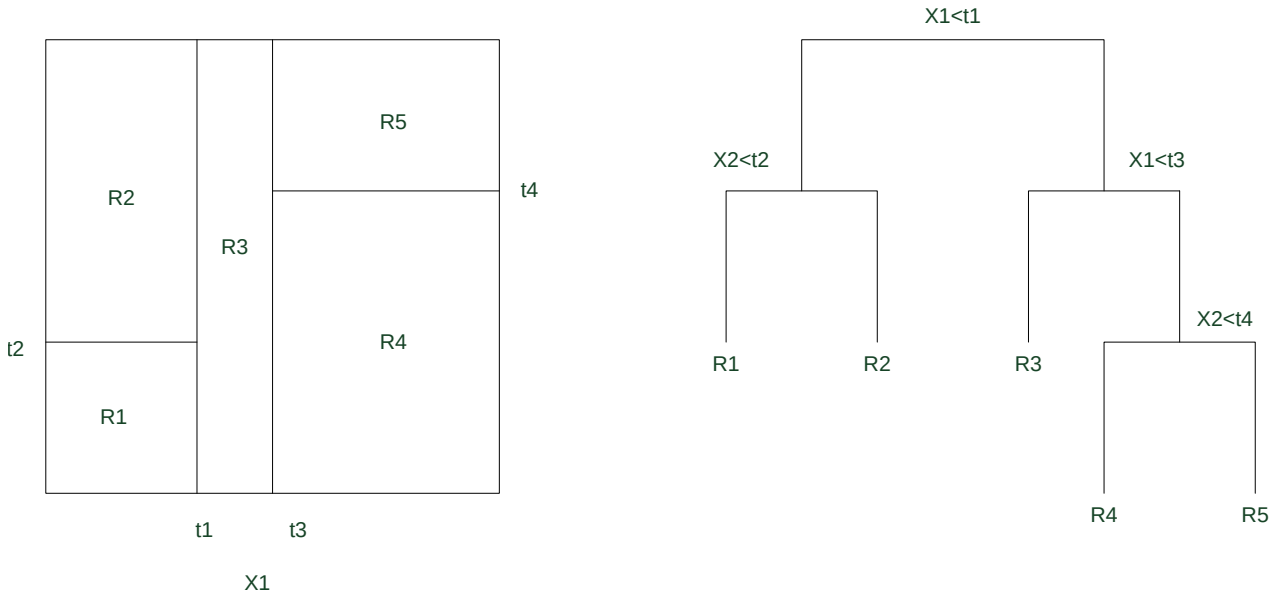


Figure 21: Partition of a two dimensional space, and corresponding binary tree diagram

response is modeled by the mean Y in each region. The variable and the split point are selected to achieve the best fit. Then one or both of these regions are split into two or more regions, and so on until some stopping rule is applied. A simplified bi-dimensional case is displayed in Figure (21) . In the example the entire domain is divided into five regions, the diagram on the right side of Figure (21) shows the binary selection tree. The upper part of the binary tree is where the entire set is located. Observation satisfying at each junction are assigned to the left branch, and the sub division is arrested when the region satisfy a certain criteria. The partitioning algorithm needs to determine which splitting variable and splitting point to use, hence to determine the shape of the final tree. Starting from the entire data set a splitting variable and a splitting point will define the two regions:

$$R_1(j,s) = \{X|X_j \leq s\} \quad and \quad R_2(j,s) = \{X|X_j > s\} \tag{7}$$

Which are determined by solving:

$$\min_{j,s}[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2] \tag{8}$$

Which consist in identifying two areas of the domain where the data are as less inhomogeneous as possible. For any choice of j and s, the inner minimization is solved by:

$$\widehat{c_1} = ave(y_i | x_i \in R_1(j,s)) \quad and \quad \widehat{c_2} = ave(y_i | x_i \in R_2(j,s)) \tag{9}$$

The tree size is determined by a tuning parameter, which represent the maximum number of branches.

MARS: Multivariate Adaptive Regression Splines

MARS is a nonparametric method for flexible regression modeling of multidimensional data, this means that the predictor does not take a predetermined form but is constructed according to information derived from the data. An expansion in product spline basis functions is carried over, but differently from other regression methodologies the number of basis functions as well as the parameters associated with each one are related to the data. Mars approximation takes the following form:

$$\widehat{f}(x_1, x_2, ..., x_p) = \sum_{m=0}^{M} a_m B_m(x_1, x_2, ..., x_p) \tag{10}$$

Where $B_m(x_1, x_2, ..., x_p)$ are the basis functions. They take the form:

$$B_m(x_1, x_2, ..., x_p) = \prod_{k=1}^{K_m} b_{km}(x_{v(k,m)} | P_{km}) \tag{11}$$

This is the product of elementary functions $b_{km}(\cdot)$, each one using just one input variable and characterized by a set of parameters P_{km}.

The basis functions in the MARS method have the following form:

$$b_{km}(x|s,t) = [s(x-t)]_+ \tag{12}$$

Where the subscript + indicates that only the positive part of the argument is taken, i.e.:

$$(x-t)_+ = \begin{cases} x-t, & \text{if } x > t \\ 0, & \text{if } x \leq t \end{cases} \quad and \quad (t-x)_+ = \begin{cases} t-x, & \text{if } x < t \\ 0, & \text{if } x \geq t \end{cases} \tag{13}$$

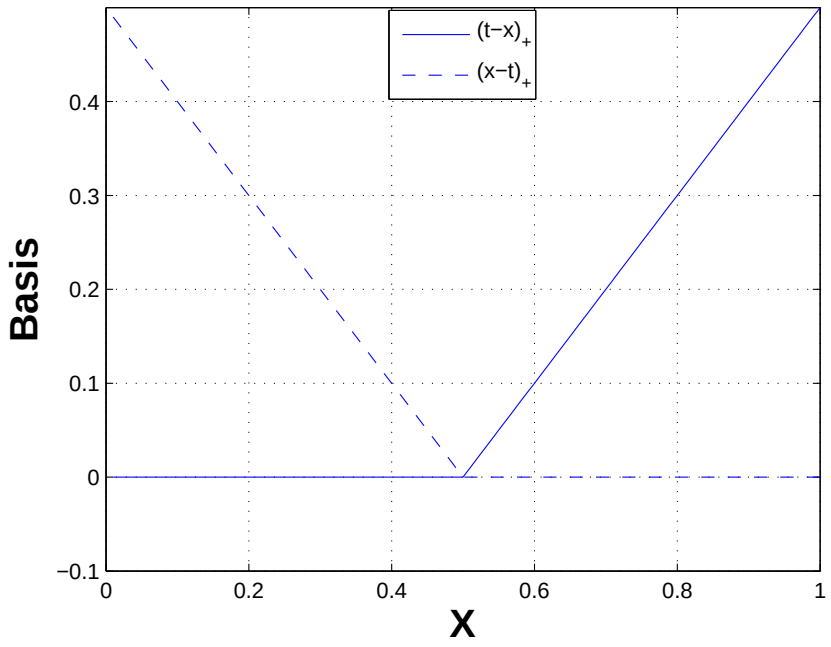


Figure 22: Basis functions $(x-t)_+$ (broken) and $(t-x)_+$ (solid)

The sample case with $t = 0.5$ is shown in Figure 22. The goal of the algorithm is to produce a minimal set of basis functions B_m for approximating each output function. This is accomplished through a two phase iterative approach. The first phase, also called forward part, generate a super set of basis functions. The data are then over-fitted and the number of basis functions is larger than optimal. The second phase, called backward procedure, selectively deletes basis functions with the goal of producing an acceptable fitting with the minimum number of basis.

Forward Phase A collection of basis functions is generated by assigning a reflected pair for each input X_j with knots at each observed value x_ij of that input. This set of basis functions will be called C and has the following form:

$$C = \{(X_j - t)_+, (t - X_j)_+\}_{t \in \{x_{1j}, x_{2j}, \ldots, x_{Nj}\}, j=1,2,\ldots,p} \tag{14}$$

Although each basis functions depends only on a single input it will represent a function over the entire domain $\mathbb{R}^p$. The model-building strategy adds one by one a function from the set C and their products. Each term in equation 10 is a function in C, or a product of two or more functions

contained in C. The main scope of the forward phase is to select which of those functions will be included in the model.

The selection is performed iteratively. Starting with a constant function b_m and all the set C as candidate as shown in Figure 23. The pair of basis function selected from the set C will be inserted in the model set M. At each stage the candidate functions for M are composed by all the products between the reflected functions in C and the function already present in M. The term that will be added to M will have the following form:

$$a_{M+1}b_l(X)(X_j - t)_+ + a_{M+2}b_l(X)(t - X_j)_+, \quad b_l \in M \tag{15}$$

The selection among all the possible outcomes is performed by minimization of the squared error. The coefficients that will multiply the new basis function are selected by least square method. Once the winning products are added to the model the process is continued until the model set M reach the maximum number of terms. Figure 23 is a scheme of the forward model building procedure. On the left are the basis functions currently in the model M, at the first iteration this consists of only a constant. On the right are all the candidate basis functions belonging to C. These are reflected pairs of linear splines with knots at all observed values x_{ij} of each predictor X_j. At each stage we consider all products of a candidate pair with a basis function already in the model. The logic for selecting which candidate function to use is described by the inner loop of the forward phase algorithm, Figure 24. Once a reflected pair is added to M the algorithm must determine which of the basis functions already present in the model must be multiplied by the new term. The outer loop in Figure 24, determines which basis function $b_l(X)$ in equation 15 must be multiplied by the new term in the model. There is one restriction regarding the formation of the model set M; each input variable can appear at most once in a product. This is in order to avoid the formation of higher order powers of an input, which would be complex to govern in the forward model building phase. At the end of this process the result is a large model, which typically overfits the data, hence a deletion procedure is needed.

The MARS forward procedure approach can be seen as an evolution of the tree growing algorithm previously discussed. In the tree based methods at each iteration a node was split, a similar result is obtained in MARS multiplying a step function by a pair of reflected step functions. The

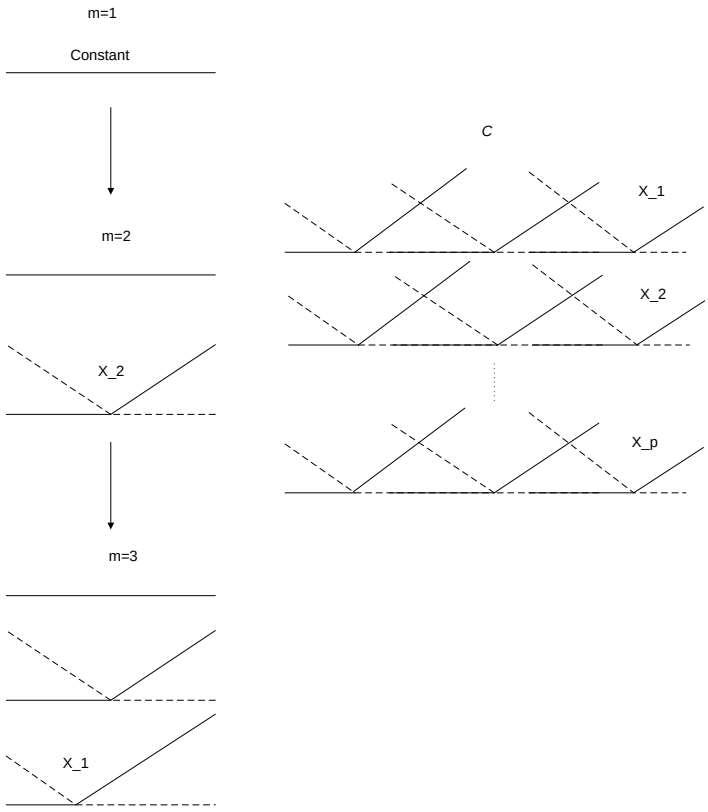


Figure 23: Schematic of the MARS forward model-building procedure.

main improvement achieved by MARS with respect to tree based methods is the ability to capture additive effects. This is due to the fact that a knot location can be used as many times as necessary, while tree methods cannot split a node more than once. A key property of the functions shown in Figure 22 is their local behavior. They have a zero value over a large part of their range and nonzero only over a small part of the feature space. As a result the regression surface can be built up parsimoniously, because adding a basis function to improve the fitting in a certain area does not affect the entire domain. The use of higher order basis functions would produce a nonzero product everywhere, and would not work as well.

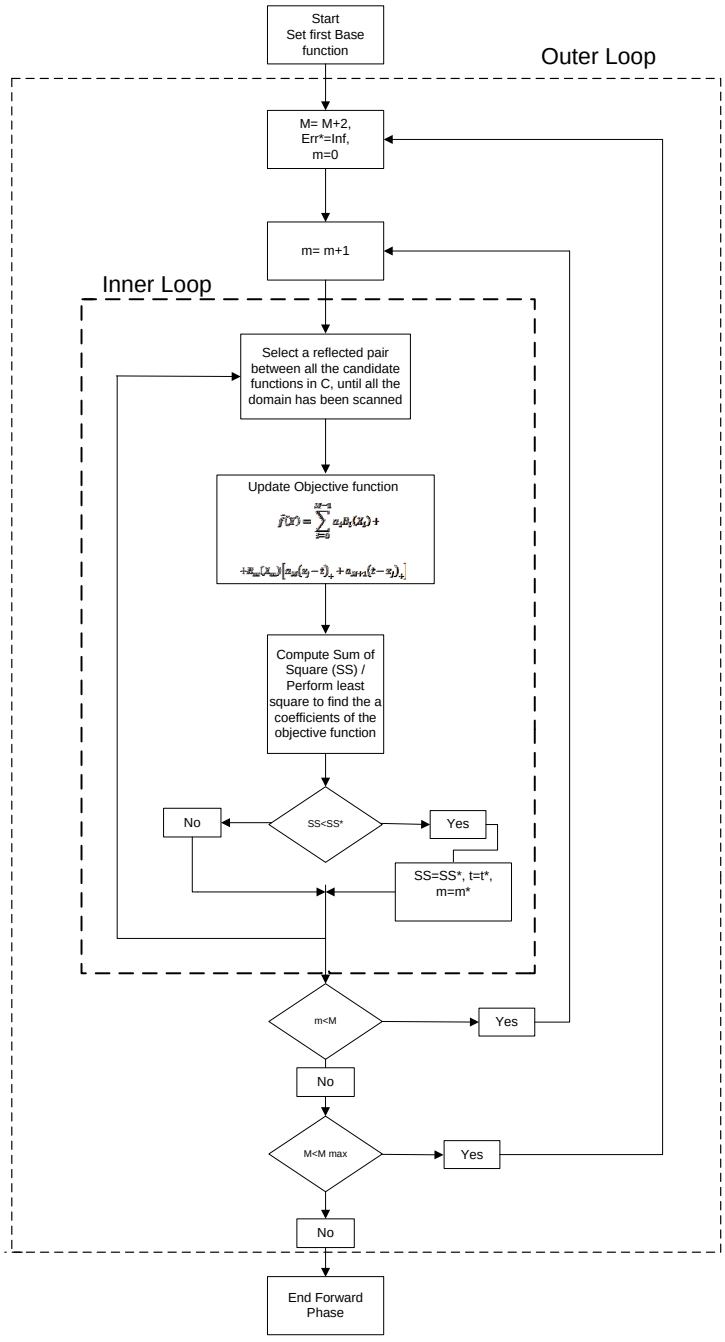


Figure 24: Block diagram of the MARS forward model-building procedure.

Backward phase The backward stepwise procedure is applied to the final set produced by the forward phase in order to selectively delete individual basis functions whose contribution to the fitting is judged (by a certain criterion) to be negligible.

The criterion used to judge the contribution of each basis function is the Generalized Cross Validation (GCV). This method is used instead of the more common Residual Sum of Squares (RSS) because otherwise the most complex model would always be considered the best. In any case the RSS is included in the GCV valuation, as follow:

$$GCV = \frac{RSS}{f(N)} \tag{16}$$

Where $f(N)$ is a function function which value is proportional to the number of basis functions, and RSS is defined as follow:

$$RSS = \sum_{i=1}^{n} (y_i - f(x_i))^2 \tag{17}$$

In order to avoid an excessively noisy fit, i.e. roughness in the response function, an auxiliary term is added to the standard RSS. With this extra parameter the modified RSS function will look as follows:

$$RSS^* = \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda J(f) \tag{18}$$

Where $J(f)$ is a function describing the roughness of the response function, and λ is a roughness penalty parameter which value can vary between 0 and 1. The roughness penalty parameter is left as a tuning degree of freedom to the user depending on the kind of fitting desired; in the following work it has been set to 0.5. At the end of the backward phase the model having the lowest GCV is selected as the final one.

ANOVA

The representation of the model given by equation 10 is useful in understanding the construction logic of the algorithm but does not provide very much insight into the nature of the approximation. The objective function can be rearranged into a form that reveals useful information about the predicted relationship between the response y and the covariates x. This is done by collecting together all basis functions that pertain to the same predictor variable, so that the model can be

recast in the following form:

$$\begin{aligned}\widehat{f}(x_1, x_2, ..., x_p) &= a_0 + \sum_{K_m=1} f_i(x_i) + \sum_{K_m=2} f_i j(x_i, x_j) + \\ &+ \sum_{K_m=3} f_i jk(x_i, x_j, x_k) + ... \end{aligned} \tag{19}$$

In equation 19 the first sum includes all the basis functions that involve only a single variable. The second sum does the same thing for the two variables interaction, and so on until the maximum degree of interaction present in the model is represented. The ANOVA decomposition is then performed by grouping each of the particular variables that enter into the model. Depending on the degree of interaction present in the final model the output can be related to one or more variables. For example the first sum in equation 19 can be expressed as:

$$f_i(x_i) = \sum_{K_m=1} a_m B_m(x_i) \tag{20}$$

Equation 20 is a sum over a single variable basis functions involving only x_i, similarly each bivariate function in the second sum of equation 19 can be expressed as:

$$f_{ij}(x_i, x_j) = \sum_{K_m=2} a_m B_m(x_i, x_j) \tag{21}$$

Equation 21 is a sum over all two variable basis functions involving the particular pair of variables x_i and x_j. Adding this to the corresponding univariate functions provides the joint contribution of these variables to the model. Terms involving more variables can be collected together and represented similarly.

Interpretation of the MARS model is greatly facilitate through this ANOVA decomposition. This representation identifies the particular variables that enter into the model, whether they enter purely additively or are involved in interactions. The ANOVA tables presented in this work address the importance of the single variables, or of their interactions, composing the model by reporting the value of the coefficient a_m appearing in equations like 20 21.

Case Study

To determine that the code developed here works properly a case study has been performed. The purpose of this study is to test the ability of MARS algorithm, and in particular the specific code

here produced, to uncover interaction effects that are present in data. The test example is taken from [40]. They considered trying to model the function:

$$F(x) = 10 * sin(pi * x_1 * x_2) + 20(x_3 - 1/2)^2 + 10 * x_4 + 5 * x_5 \tag{22}$$

In the n=6 dimensional hypercube using N=200 points. The covariates were randomly generated from a uniform distribution and the responses were assigned using the given equation with the addition of a standard error. In this section we consider the same function but with an increase in dimensionality; n=10. Instead of one noise variable, there are now five such variables that are independent of f(x).The following table summarize the ANOVA decomposition obtained by modeling the given equation with MARS.

Function	**STD**	**GCV**	**# of basis**	**Variables**
1	4.761	62.542	2	1
2	5.512	153.263	1	2
3	1.514	104.319	2	3
4	2.739	33.806	2	4
5	1.415	3.164	2	5
6	5.155	44.224	5	1 2

Table 4: ANOVA Decomposition

The first column lists the function number. The second gives the value of standard deviation of the function. This gives one indication of its relative importance to the overall model and can be interpreted in a manner similar to the regression coefficient in a linear model. The third column gives another indication of the importance of the corresponding ANOVA function, by listing the GCV score for a model with the entire basis functions corresponding to that particular ANOVA function removed. This is used to judge if this function is making an important contribution to the model. The fourth columns gives the number of basis functions comprising the ANOVA function while the last column gives the particular predictor variables associated with the ANOVA function.

From Table 4 we see that the first five functions involve only one variable, and just one involves two variables. Judging from the second and third column of the ANOVA table all the indicated variables are important, and there is no indication in the model to the last five variables (pure noise). With respect to the originating function we see that MARS is able to uncover the presence of an interaction between the x_1 and x_2 variables. It also rank the variables in terms of importance, x_4 has twice a stronger effect than x_5. Figure 25 shows a graphical representation of the functions that have a stronger impact. The three additive contributions are plotted in the first three frames, while the joint contribution of x_1 and x_2 is plotted in the surface plot on the bottom. The graphical representation is a key feature of the MARS analysis because together with the ANOVA analysis allows to uncover the nature of the effect that each independent variable has on the analyzed output. For example we can determine that x_4 and x_5 have a linear effect while x_3 is parabolic.

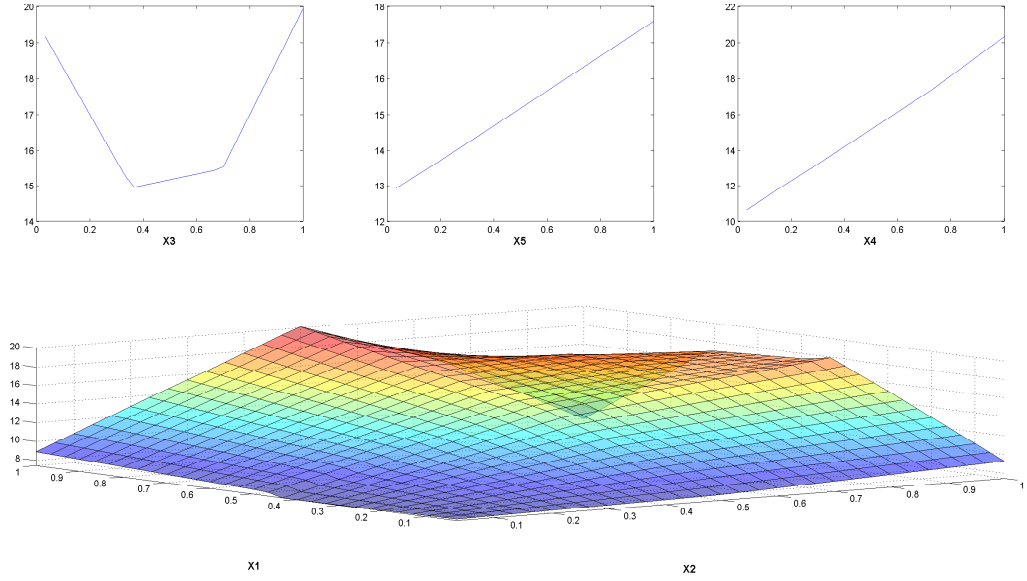


Figure 25: Single variable plot and interaction surface

Comparing the results of the MARS fit to these data with the true underlying function, shows that the resulting model provides a fairly accurate and interpretable description. This is especially noteworthy given the high dimensionality (n=10) and the small sample size (N=100).

Preliminary Results

A preliminary version of the code has been developed using data that was generated on the FACE 5 fuel, and the experimental study was focused on injection strategy. The following plots (Figures 26, 27, 28, 29, 30) are produced to demonstrate the capability of this method to generate flexible functions that are capable to follow the trend of the experimental values even when the behavior is far from monotone. This result is a direct consequence of the nature of the functions used for approximation; unlike linear regression the objective function is composed of Heaviside functions rather than polynomial functions. A source of flexibility for this model is the possibility to perform tuning on the objective function. That is, a penalty parameter to avoid excessive roughness in the objective functions, and the maximum number of basis functions used for regression are set before performing the analysis.

Figure 26 shows the trend of NOx concentrations versus two injection characteristics at the time. The plots are organized in such a way that those in the same row are characterized by the same variables. The x and y axes represent two set of injection characteristics, with the third one used as a parameter to which are assigned three equally spaced values. Both the regression surfaces and the experimental points are displayed in order to visually evaluate the fitting capability of the MARS model. In all the panels composing Figure 26 the regression surfaces fit the experimental values except for the subplot (H), where the predicted value at 40 % fuel split and 40 Pilot SOI is not consistent with the experimental correspondent. The local lack of fitting (LOF) is a consequence of MARS being a compromise between a model based, and local methods. The model based algorithm are more oriented in finding a regression function which globally reproduce the trend of the data, consequently paying a price in terms of local fitting.

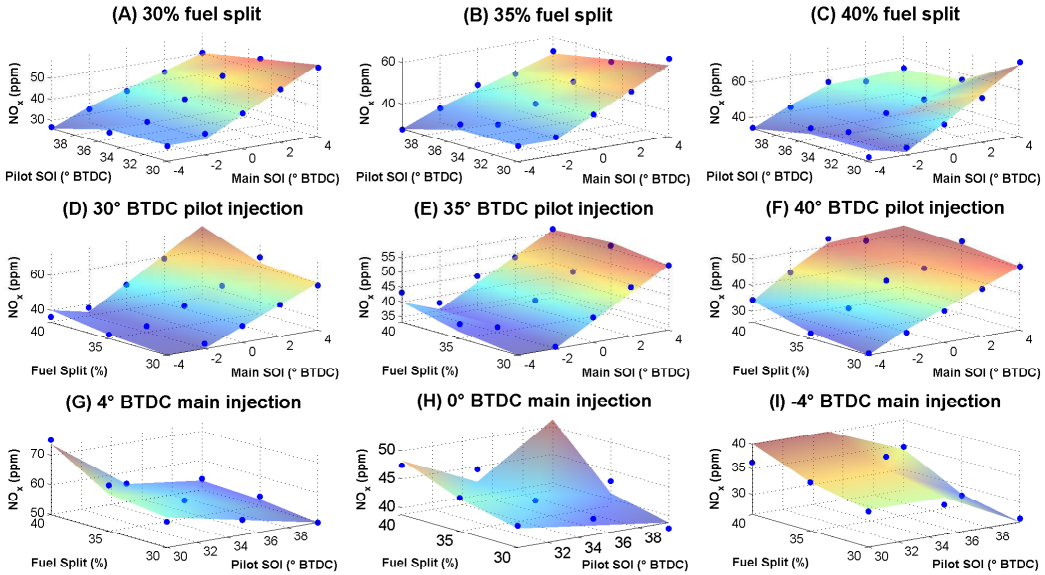


Figure 26: NOx emission from FACE 5 fuel as a function of pilot and main SOI on the first row, fuel split and main SOI on the second, fuel split and pilot SOI on the last row

Surface plots reveal how retarding the main SOI is beneficial in terms of NOx reduction, due to the lower temperature reached in the combustion chamber. Varying the pilot injection we observe a less significant (compared to Main SOI) but more complex effect on NOx emissions. The key for analyzing this parameter is to look at the relative position of pilot and main injection. When the two are close the burning pilot fuel quantity will overlap the main injection event, leading to higher temperatures and consequently higher NOx emissions. A substantial separation between the pilot and main injection event may quench the pilot quantity ignition. Minami et al. [13] explained this phenomenon comparing the effect of pilot injection to an internal EGR, especially when the quantity of fuel injected during the pilot event is higher than 10% of the total fuel. The burned gas produced from the combustion of the pilot injection will dilute the concentration of oxygen inside the combustion chamber for the main injection, acting basically like an EGR system. The fuel split among pilot and main injection works as a scale effect for the considerations that have just been made, this can be easily observed by looking at the different planes in Figure 26.

Figure 27 shows the regression surface in regards to the PM emissions. The complex phenomena that bring to PM production and oxidation during combustion make the correlation with the injection strategy not as predictable as other emission characteristics, making the regression of this emission characteristic is the most challenging for the model.

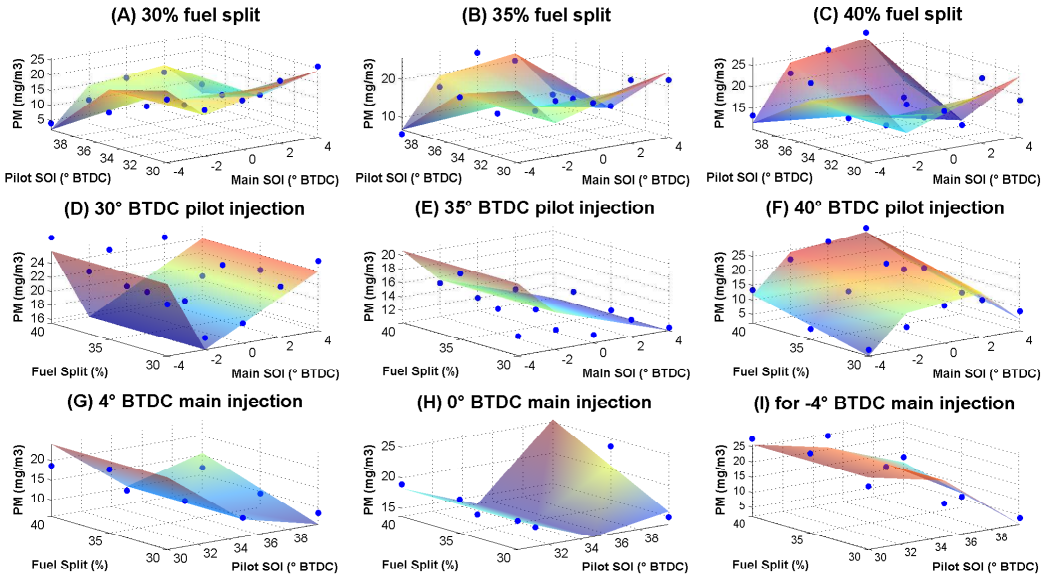


Figure 27: PM emission from FACE 5 fuel as a function of pilot and main SOI on the first row, fuel split and main SOI on the second, fuel split and pilot SOI on the last row

The regression surfaces show the advantage in using pilot injection, which allows to retard the main injection further than it would be possible with a single injection event. This is related to the reduction in ignition delay consequent to a higher temperature in the combustion chamber at the moment of the main injection. An important parameter in this sense is the time interval between the pilot and main injection event. When this time interval is too long the products of pilot injection mix sufficiently with the ambient gas in the cylinder, consequently no difference is produced with respect to the single injection case. On the other hand, when this interval is shorter a local high temperature zone is created near the fuel nozzle with a consequent shorter ignition delay [13]. It is still important to maintain a sufficient separation between pilot and main injection to ensure the main injection occurs after the injection delay of the pilot event.

Looking at the plots in Figure 27, especially at those where pilot SOI is varied (first and last row), it can be noticed that the highest values of soot concentration are produced when the pilot injection is started at 30° BTDC. One way to interpret this result is by examining the mechanism of soot formation by fuel rich regions of burning diesel jets. Having the pilot injection close to the main injection event bring part of the fuel injected to be still unburned when the main injection starts. The fuel-rich pockets which do not have time to mix and burn prior to exhaust valve opening are among the main cause of soot generation. Another key parameter to consider, strongly related to injection timing, is the temperature evolution during combustion. While high temperatures at the end of the combustion enhance the oxidation of soot, having high temperatures at the time of injection reduce air entrainment and increase soot formation.

The importance of the pilot injection in PM production is further underlined by considering the fuel split; this is indeed a parameter that represents the intensity of the pilot injection. Looking at the second and third rows in Figure 27 we notice that larger amount of fuel delivered during the pilot injection brings to higher PM production. This can be related to an increase in temperature and decrease of oxygen concentration before the main injection event starts. These conditions usually lead to higher PM production. Generally an increase in temperature inside the combustion chamber leads to a reduction in lift-off length, which is well known to increase soot emissions [48].

Figures 29 and 28 show the behavior of HC and CO represented in the same manner as for the previous plots. These two characteristics show a similar trend with respect to injection timing and in both cases low temperature combustion represents a penalty. The low cylinder temperature is useful in reducing NOx and soot production, but, since it involves slow oxidation reactions that cause misfiring and quenching, will result in higher HC and CO emissions compared with conventional diesel combustion.

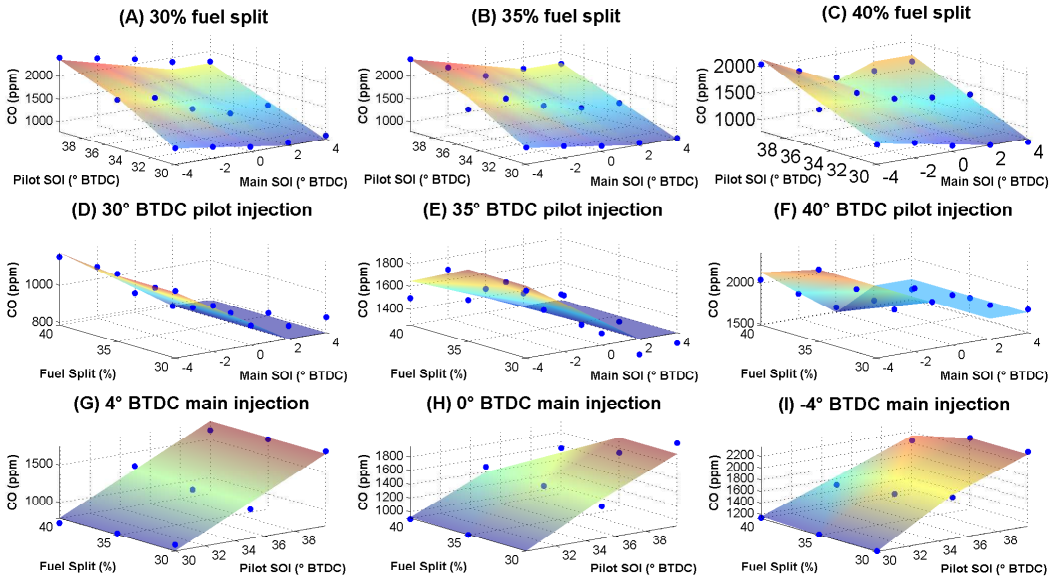


Figure 28: CO emission from FACE 5 fuel as a function of pilot and main SOI on the first row, fuel split and main SOI on the second, fuel split and pilot SOI on the last row

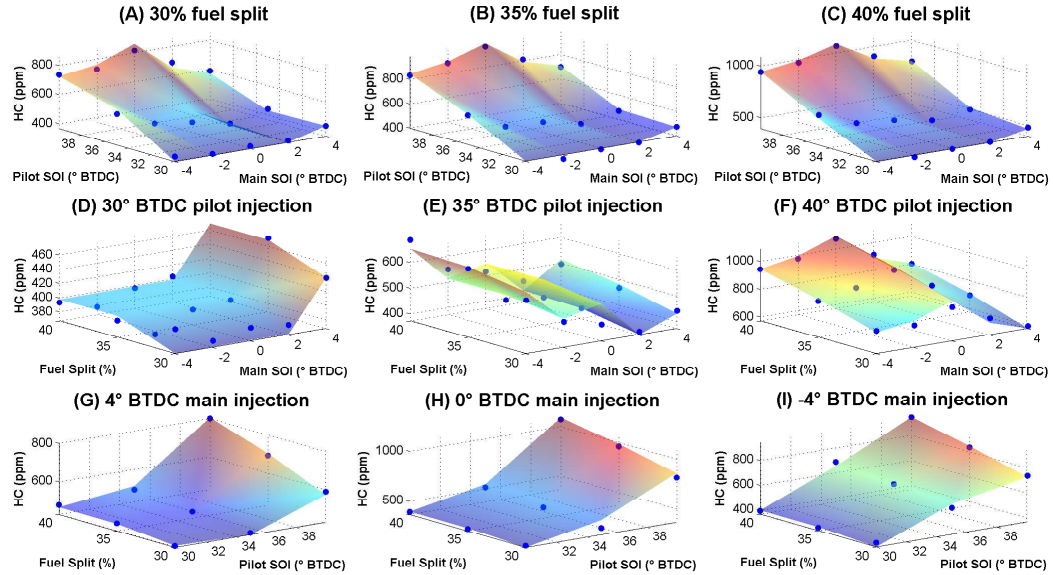


Figure 29: HC emission from FACE 5 fuel as a function of pilot and main SOI on the first row, fuel split and main SOI on the second, fuel split and pilot SOI on the last row

Figure 29 and 28 show the negative effect of advancing Pilot injection in terms of CO and HC production. The early fuel injection causes the spray interaction with the cylinder liners or piston walls; the result is wall quenching with resulting fuel problems and high concentrations of HC in the exhaust. Furthermore low temperature combustion is a major cause for high CO emissions.

The low temperature of the exhaust is also to be considered as an issue for the after-treatment of HC and CO. In general the factors that bring to higher production of CO and HC can be listed as follows [5]:

- The long spray penetrations which characterize LTC operation bring fuel to accumulate on the combustion chamber surfaces.
- Fuel lean zones inside the combustion chamber are more common in LTC operations rather than in conventional diesel. Fuel in these areas is more prone to escape combustion leading to the production of unburned HC.
- Since the local peak temperatures are lower than conventional diesel, the fuel near the wall does not burn.

The plots in Figure 30 show the trend of Brake Thermal Efficiency (BTE) with respect to the emission characteristics. The points analyzed in this study have a range of efficiency in the interval $29\% - 31\%$, which makes the regression on this characteristic challenging to perform.

The pattern obtained using the regression surfaces regarding the BTE resemble the one obtained for NOx. This is physically meaningful considering that both NOx production and thermal efficiency increase with temperature, hence will reach higher value in correspondence of an injection strategy that guarantees high temperature.

When analyzing the best trade off in terms of emissions and performance it must be considered what is the threshold value to associate at each emission characteristics. Under this point of view the most critical characteristics among those that have been presented is the soot concentration. In this sense it is advantageous to inject a small quantity of fuel (30% of the total) at 40° BTDC as depicted from Figure 26. Advancing the start of the pilot injection event is beneficial in terms of PM but will produce an increase in HC and CO production for the reasons previously discussed.

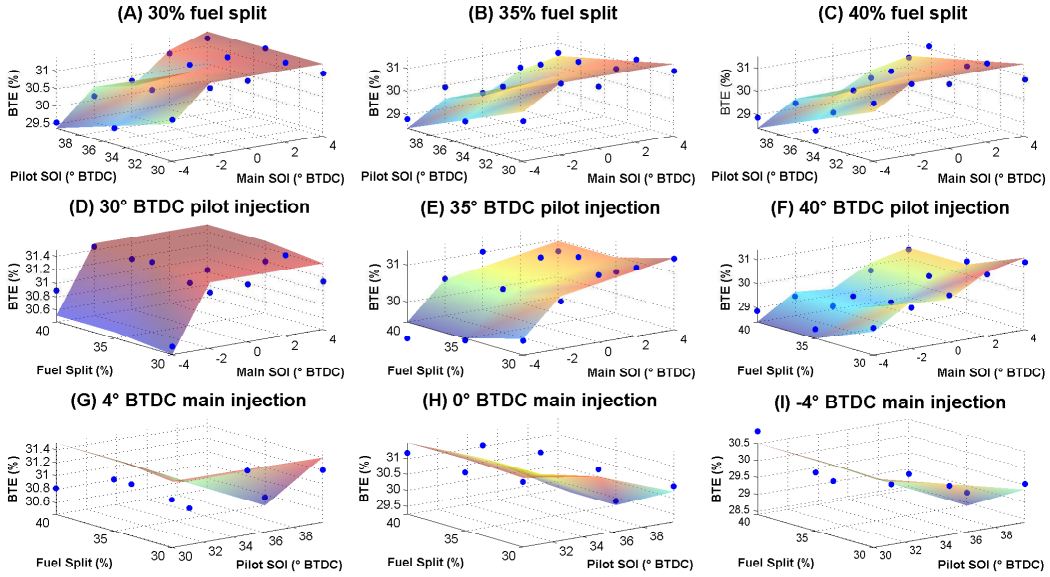


Figure 30: BTE from FACE 5 fuel as a function of pilot and main SOI on the first row, fuel split and main SOI on the second, fuel split and pilot SOI on the last row

Those characteristics still fall in the acceptable range. Regarding the start of the main injection event two choices are possible, both allowing for acceptable range of emission. It can be located at 4° BTDC or -2° BTDC, in the first case the engine shows better efficiency and lower PM emissions, while the latter case is more performing in terms of NOx reduction.

Experiment Reduction

In this section the potential use of MARS algorithm for optimizing the number of test necessary to characterize a given phenomenon is analyzed. The main idea is to use the regression capability of MARS within the classical Taguchi methodology.

Taguchi Standard Taguchi method applied to engine testing contemplates the selection of the levels of variation for the engine parameters. Three levels are the least number that can be used in order to study nonlinear responses of the parameter effects. Then a so called orthogonal matrix of tests is determined.

The orthogonal matrix testing makes use of an orthogonal array design to isolate the main effects of a single parameter on the response function. Orthogonality is intended in a combinatoric sense: for any pair of columns- in the array- all combinations of factors occur an equal number of times [50]. It is based on the assumption of no interactions between parameters, which in other terms implies that models used to fit the data collected do not include cross product terms between the parameters. However, the validity of this assumption needs to be proved, and this was done by comparing the model predictions with the measured points. The first step to estimate the factor effect is to calculate the overall means, which in the case of NO_x for a three level and three factors is:

$$NOx_{exp} = 1/9 \sum_{i=1}^{9} NOx_i \tag{23}$$

The factor level response for each level is calculated using analysis of means (ANOM) as follows:

$$NOx_{MainSOI} = 1/3 \sum_{i=1}^{3} NOx_i; NOx_{PilotSOI} = 1/3 \sum_{i=4}^{6} NOx_i; NOx_{MainSOI} = 1/3 \sum_{i=1}^{3} NOx_i \tag{24}$$

Then the actual effect of the factor level is determined by its deviation from the overall mean, for instance:

$$\Delta NOx_{MainSOI} = NOx_{MainSOI} - NOx_{exp} \tag{25}$$

The predictive model is derived for the data collected according to the orthogonal array design. This model consists of the simple sum of the individual factor effects and does not include cross-terms. The empirical model produced by this technique is compared to the results of a validation test. The verification test is performed to confirm the reliability of the prediction model. If the validation test confirms the model prediction then the set of engine tests selected is descriptive of the engine behavior.

MARS The approach suggested in this work makes use of the MARS algorithm to produce the predictive model. In this case the assumption of no interactions between the input parameters can be eliminated. The flexibility of the regression algorithm allows for use of scattered data for determining the prediction models. In selecting the test the entire range of variation of each parameter should be used, and like for the Taguchi method, at least three levels for each parameter

should be considered. The approach suggested in this section is displayed by using an example based on the data presented in this work. The control parameters that we are considering are the three injection parameters (Main SOI, Pilot SOI, and Fuel Split). A subset of data points is considered, those are distributed according to a central composite design. This subset includes just half of the total data available, distributed according to the prescriptions described above. Figure 31 shows the distribution of the tests in a three dimensional space where each of the axis represent one of the input parameters, and the intensity of the output is described by the dimension and color intensity of the spheres.

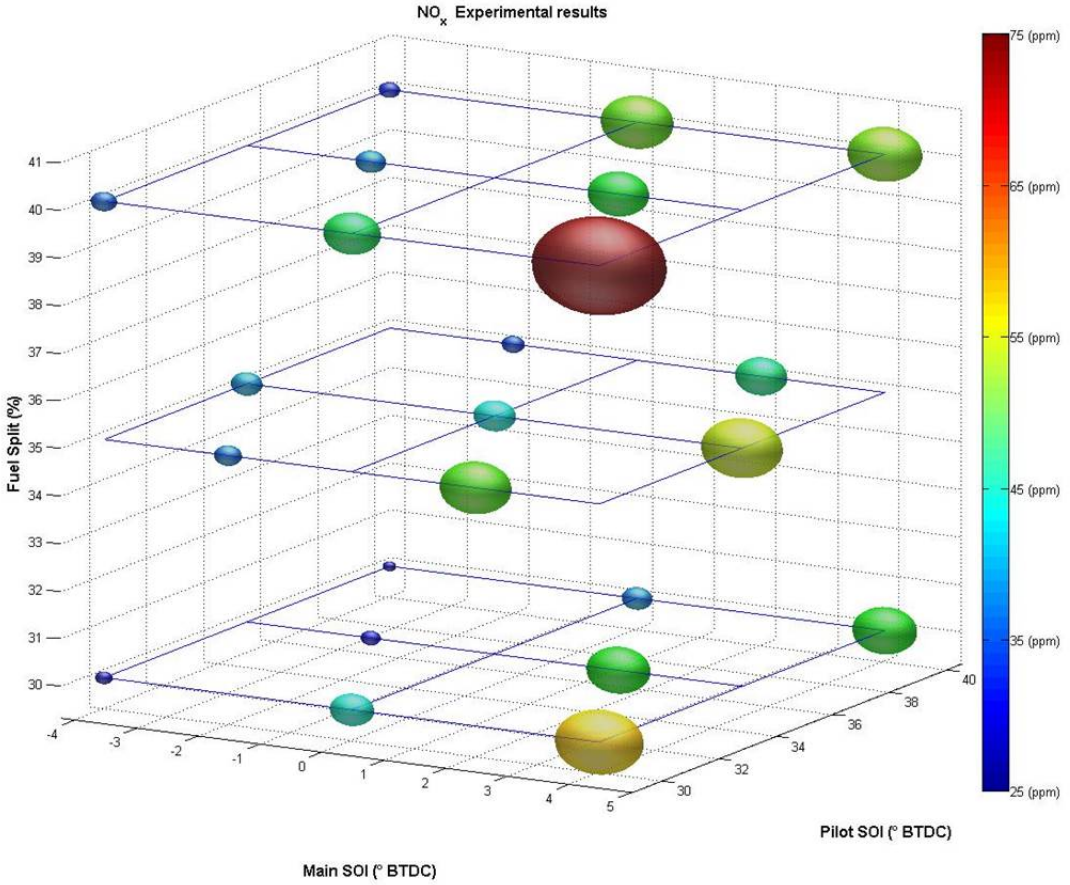


Figure 31: NOx data subset

Using this smaller data set an emulator model is generated using MARS to predict the NO_x emissions. At this point the equation obtained is interrogated in all the points corresponding to the original dataset. The results are compared with the experimental values in the original complete

dataset. Figure 32 shows the comparison result.

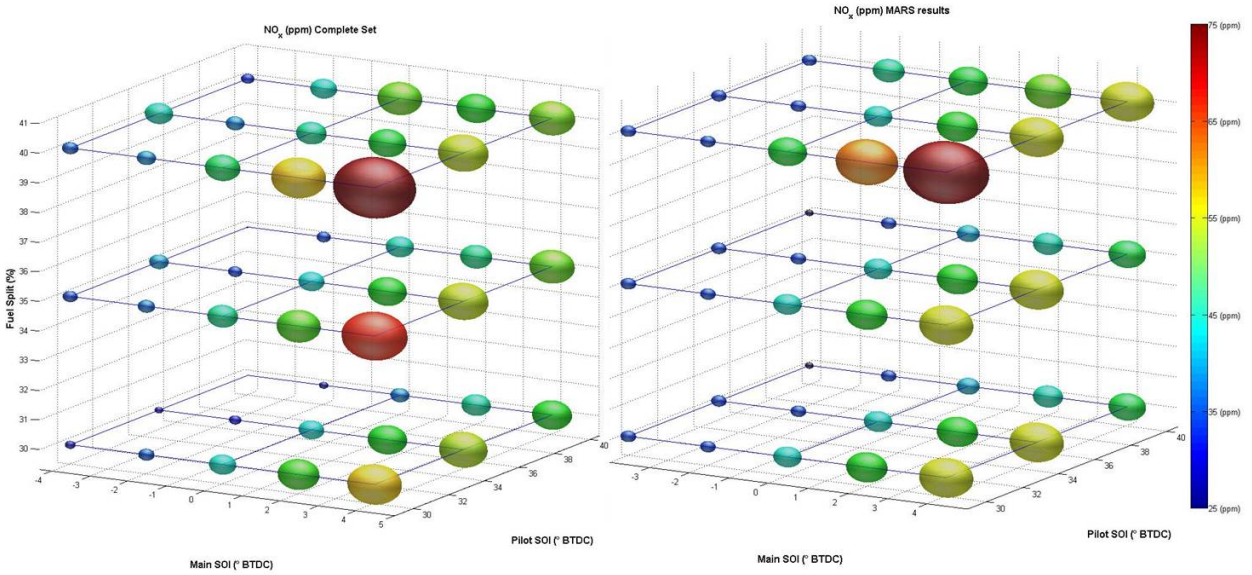


Figure 32: NOx comparison between MARS results and experimental values

The difference between the emulator model and the actual values can be quantified by the residual sum of squares, which in this case is $R^2 = 0.93$. The same procedure is applied to soot concentration, CO, HC, and BTE. The results are summarized in table 5, while the rest of the describing plots are collected in Appendix . Table 5 also presents the results obtained using the classical Taguchi methodology.

Characteristic	R^2 MARS	R^2 Taguchi
NOx	0.93	0.86
PM	0.69	0.29
CO	0.92	0.67
HC	0.93	0.92
BTE	0.73	0.69

Table 5: Residual sum of squares for each characteristic predicted

The results in table 5 show a good prediction quality for NOx, CO, and HC. It is weaker for PM, and BTE which are also the characteristics with lower regression quality. This analysis proves

that with half of the tests a similar characterization of the phenomenon would have been possible.

Results

To achieve LTC the most important requirement is to generate homogeneous mixture formation prior to spontaneous ignition. This requires a longer ignition delay, which can be achieved or by proper selection of injection strategies or fuel selection strategy. The boiling ranges of the fuel and its volatility will strongly affect the outcome of a certain injection strategy, making the fuel characteristics a key parameter in the combustion process. Also the use of a split injection strategy is very important, because it will reduce the PRR and hence limit the soot production. The split injection strategy has to be developed accordingly to the CN category of the fuel under consideration.

The fuel injection parameters (namely main SOI timing and pilot SOI timing) comprising the split injection test matrices were then applied according to CN category. In the shown data, 50 percent mass fraction burned (CA50) was not held constant and varied among all fuels tested. A more advanced range of main SOI timing for the low CN fuels (30) (when compared to medium (40) and high (50) CN fuels) was established to limit misfire at retarded main SOI timing resulting from the longer ignition delay provided by these fuels. The shorter ignition delay exhibited by the medium and higher CN fuels required that the range of main SOI timing be retarded (from that of low CN fuels) . Pilot SOI timings were also adjusted for each CN category to limit heat release before the main injection event; low CN fuels tolerated a more advanced range of pilot SOI timing, while the range of pilot SOI timing for medium and high CN fuels had to be retarded.

The approach used in this study to investigate these correlations is the one described in the Regression Analysis chapter. The performance parameters are then analyzed one at the time, in order to show how the variables of interests combine. The physical meaning of each performance

parameters can be very different from another, hence only a separate study can lead to an interpretability of the results. In a second stage a strategy to combine all the underlined outputs will be defined, in order to define an optimal set of variables for the multivariate optimization.

Independently from the parameters analyzed a preliminary manual screening of the training data is necessary in order to not spoil the model with unrealistic data. The experimental points where improper combustion was identified will not be used as training data. Only those tests where reliable and consistent data are produced will be fed in the model. Both the independent and dependent variables are adimensionalized based on the maximum and minimum value obtained in the entire range. The adimensionalization has to be carried out in order to allow model generation, and to simplify the optimization process.

The rest of this section will be organized as follows; one performance parameter at the time will be analyzed. For each characteristic a report on the quality of the regression will be provided, both in graphical and quantitative way, in order to evaluate how much we can relay on the regression results. Then an ANOVA analysis will be performed over the regressed equations, in order scale the input variables based on their effect on the specific characteristic under consideration, and to the determine the key interactions between the given parameters. At this point a graphical evidence of those interactions will be produced; the response surface will be used to understand the correlation between the variables used to perform the study and the output under analysis.

After analyzing the single output one at the time, a global study will be performed. The objective is to perform a multivariate optimization study in order to determine the combination of engine and fuel parameters able to produce the best trade-off relatively to the parameters analyzed (namely NOx, PM, CO, HC, BTE). Before combining all the engine output analyzed it is necessary to determine which set of parameters should compose the computation domain. The input parameters are screened in order to make sure the combustion produced falls within the scope of this work. Hence all those combinations of fuel characteristics and injection timings that produce either misfire or a combustion clearly not definable as low temperature. To perform this kind of analysis a modification of the core of the MARS algorithm is needed. The regression in this case must be operated over categorical variables instead of numerical, to represent the nature of the problem

where the combustion is either acceptable or not. The methodology leading to this study is reported in the optimization section.

NO_x

Following the procedure described above the first characteristic to be analyzed is the NOx .Figure 33 compares the experimental points and the regression results obtained using the MARS algorithm. The plot wants to show the accuracy of the model to fit the training data. It does not lead to any physical interpretation, and the division in areas corresponding to different fuel characteristics is only produced to motivate the strong discontinuities in the data.

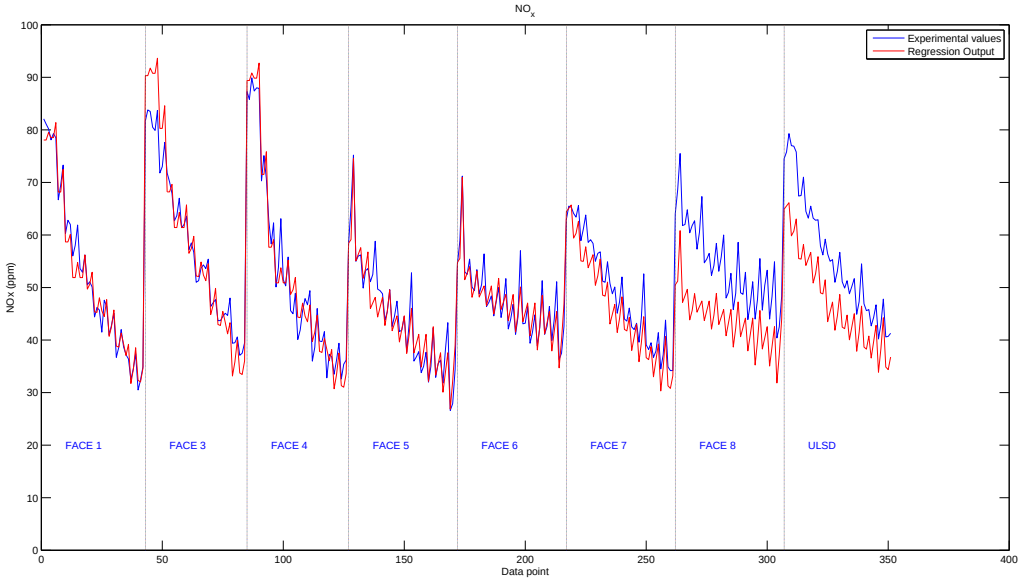


Figure 33: Comparison between experimental data and regression results

Once that it is established an acceptable level in the regression performance the equations obtained (in appendix A the explicit form is reported) are studied. The analysis method is based on ANOVA analysis, and the results are shown in table 6. The final equation is composed of 9 basis functions. These can be organized in the following table to underline the importance of the single variable on the final outcome.

Function	STD	GCV	# of basis	Variables
1	5.1	56.7	2	1
2	2.5	48.5	1	3
3	9.2	75.6	3	4
4	10.3	76.3	2	1 4
5	1.4	16.5	1	3 5
6	5.2	44.2	2	4 5
7	1.1	4.2	1	4 6
8	4.6	14.6	1	5 6
9	5.2	39.7	1	1 3 4
10	10.3	64.8	1	1 4 5
11	0.5	2.1	1	2 3 4
12	0.3	1.5	1	3 4 5

Table 6: ANOVA Decomposition NOx

The columns represent summary quantities for each one. The first column lists the function number. The second gives the standard deviation of the function. This gives one indication of its (relative) importance to the overall model and can be interpreted in a manner similar to a standardized regression coefficient in a linear model. The third column gives the associated value of GCV while the fourth column provides the number of basis functions comprising the ANOVA function. The last column gives the particular predictor variables associated with the ANOVA function (1: Main SOI, 2: Pilot SOI, 3: Fuel split, 4: Cetane number, 5: Aromatic content, 6: Distillation temperature).

The ANOVA table shows that the interaction effects among the independent variables have a stronger effect on the output than the same variables alone. These values are obtained analyzing the regression equation using the methodology described in the ANOVA section of the regression methods chapter. In particular entry 4, 5, and 8 indicate an important effect given by the joined contribution of fuel characteristics and engine parameters. The following plots give a graphical representation of the results presented in table 6.

Figure 34 represents the effect of Cetane number on NOx emissions, corresponding to entry 1 in Table 6. The plot includes three curves corresponding to three different values of Main SOI, this is done to account for the interaction effect between those variables depicted by the ANOVA analysis.

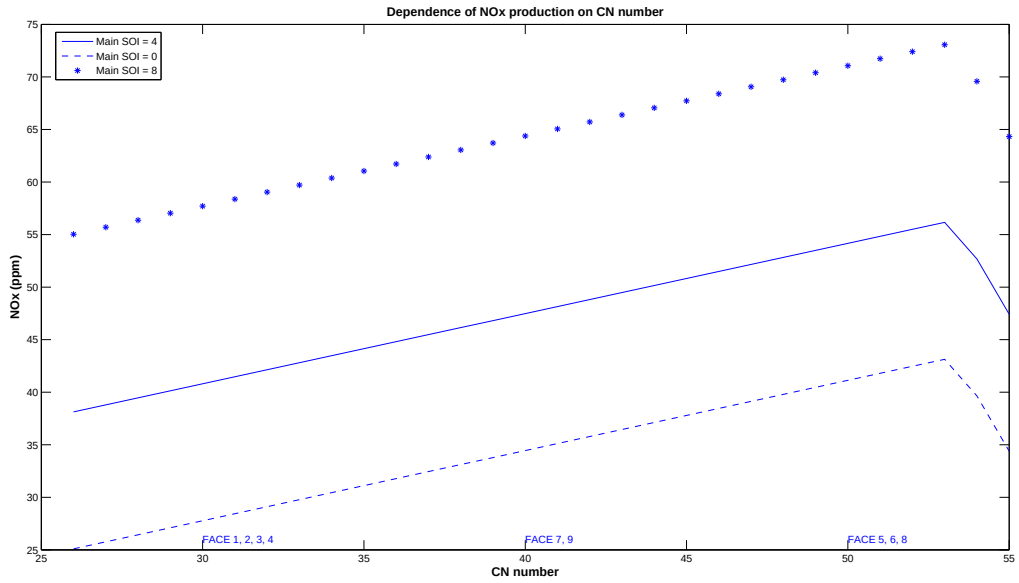


Figure 34: Dependence of NOx production on Cetane Number

From Figure 34 we see how there is a general increase in NOx production with CN. In order to understand the sharp change in slope of the regression plots in correspondence with high CN the characteristics of the regression algorithm should be considered. As a global method MARS tries to determine a general behavior of the objective function inside the domain. Where this global trend does not fit the data an extra basis function is added in order to reduce the lack of fit. In this case the extra basis function is activated at high CN to make up for the lack of fit that would be produced by a monotone function.

When analyzing the plots from figure 34 it must be underlined that dependently on the Cetane number a different injection strategy musty be selected, making the effect of these two characteristics possibly confounded. Low CN fuels require a more advanced range of main SOI timing, in order to limit the misfire resulting from a longer ignition delay. The shorter ignition delay exhibited by the medium and higher CN fuels required that the range of main SOI timing be retarded. Accordingly

to the main SOI also the pilot injection was adjusted, in order to avoid quenching phenomena between the two events. With respect to these considerations the results shown in Figure 34 gain even more importance. From what we have seen in FigureNOx , advancing injection does not have beneficial effect on NOx reduction. Since lower CN fuels show lower NOx production despite of a more advanced range of SOI timing, we can infer that low CN is beneficial for NOx production.This can be physically motivated assuming that, at equal conditions, higher ignition delays lead to lower combustion temperature, and consequently less NOx.

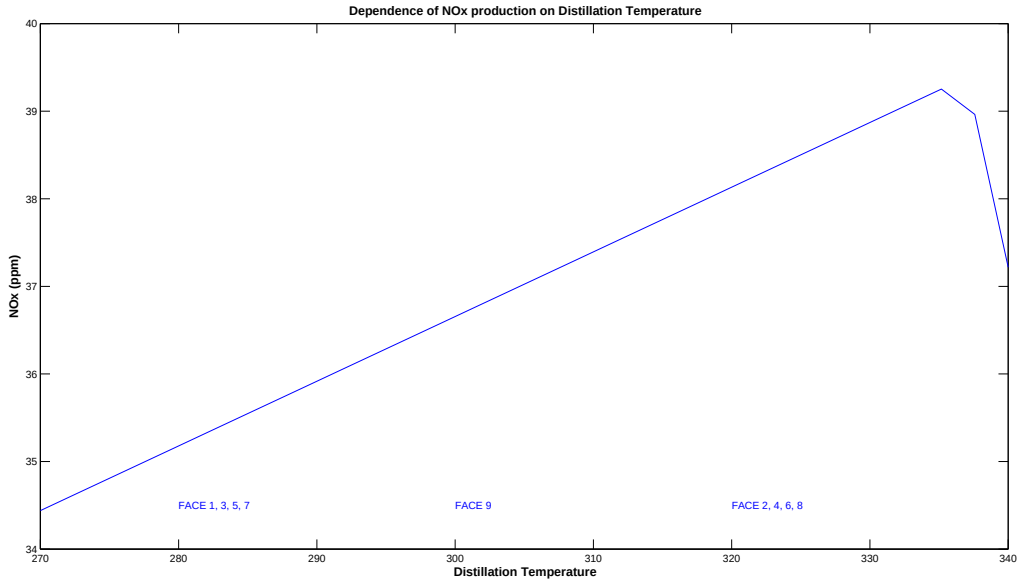


Figure 35: Dependence of NOx production on Distillation Temperature

Figure 35 represents the NOx production as a function of distillation temperature. The plot shows clearly an increase in NOx production proportional to T90. A possible explication for this phenomenon is that higher distillation temperature involves a higher volatility of the fuels, which hence will lead to higher temperature during the combustion.

The third single variable with the strongest influence on NOx emission is the main SOI which has been broadly discussed in the chapter relative to the injection parameters .

In terms of interaction the ANOVA table suggests that the first two variables to variables to analyze are CN and main SOI. Figure 36 shows the surface plot relative to the interaction

between these two parameters in terms of NOx production. The regression surface obtained through regression is compared with the experimental data represented by the scatter points. The blue points are relative to the same set of parameters which characterize the plotted surface; the green points instead are relative to a different set of parameters. These extra points have been added in order to justify the trend of the regression surface.

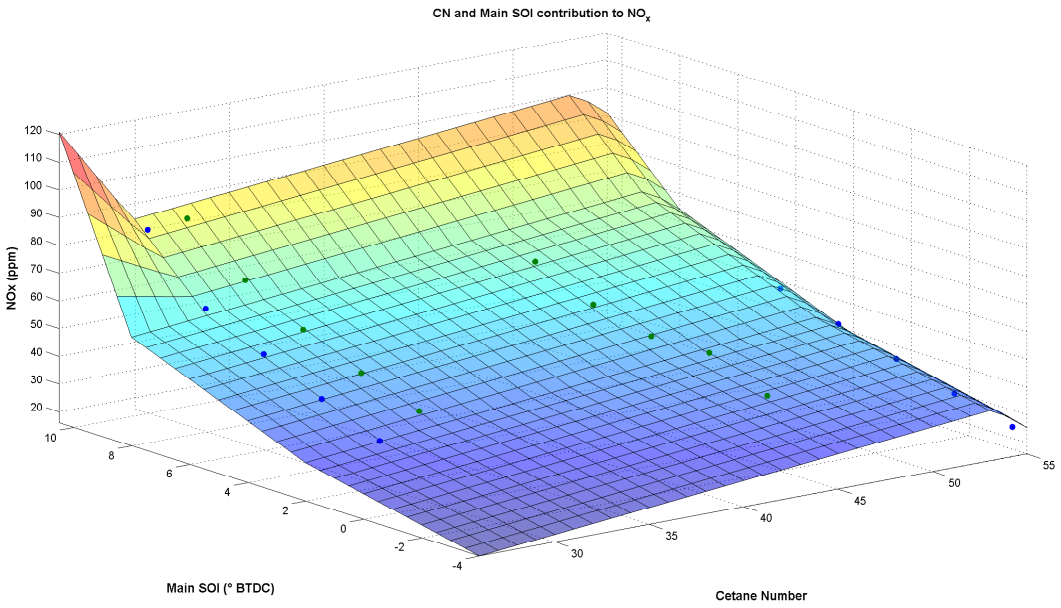


Figure 36: Dependence of NOx production on CN and Main SOI

The graphical representation of the interaction between CN and Main SOI shows that the second variable has a stronger impact on NOx emissions. Except for a peak corresponding to the most advanced Main SOI timing with the lowest CN fuel the trend of NOx emissions is rather monotone, showing a quadratic increase proportional to the distance from low CN and Main SOI. The surface in Figure 36 represents the regression equation obtained through MARS, where the variables that are not included in the plot assume a constant value. This value has been selected as the mean of the independent variable over its range of variation, and then these parameters are varied one at the time to check that they do not influence the morphology of the regression surface. In this case there is no relevant change in the shape of the surface but there is a change in the NOx scale. For this reason the surfaces corresponding to other set of parameters are not presented in this report, and

the same presenting strategy is adopted for those circumstances were the parameterized variables do not influence the shape of the surface.

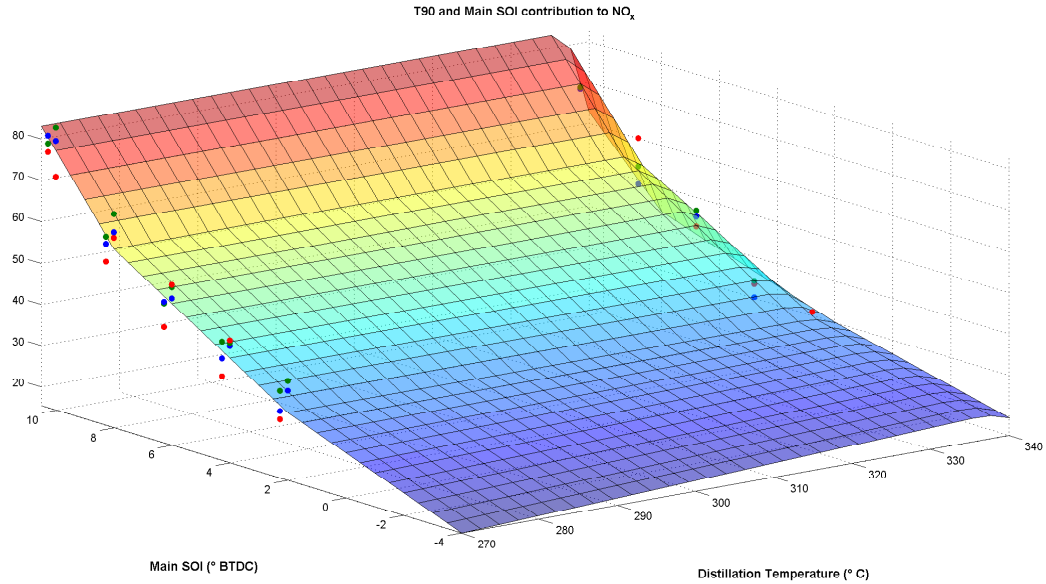


Figure 37: Dependence of NOx production on T90 and Main SOI

Figure 37 shows the contribution of distillation temperature and main SOI to NOx emissions. The blue scatter point represent the experimental data which share the same set of parameters used to produce the surface, green and red points are instead data points which characteristics are similar to the one in the equation but not equal. The surface obtained shows once again the dominant effect of Main SOI; the effect of distillation temperature is a light increase in NOx production.

The next two level interactions all refer to injection strategy parameters, which results are redundant with those shown in the preliminary results. The most important three level interactions according to the ANOVA table 6 is the one correlating the two fuel characteristics CN, and T90 with the main SOI timing. Figure 38 shows a three dimensional plot of the NO_x emissions, where the three axis represent the independent variables, namely T90, Main SOI, and Cetane Number. The intensity of NO_x production is represented by the color scale, and in order to make it visible inside the domain, it is sliced over five different planes judged to be representative of the overall behavior.

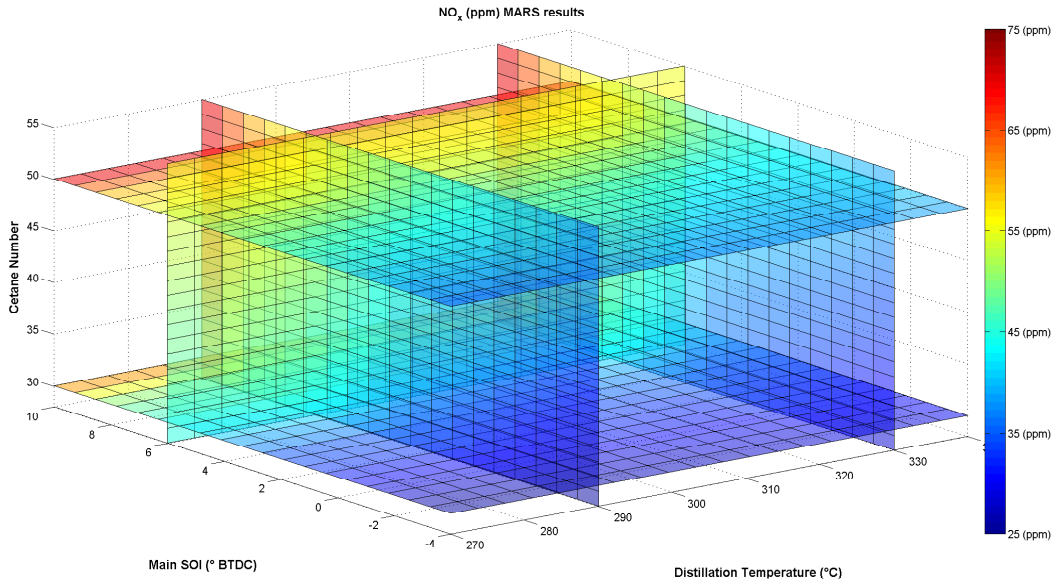


Figure 38: Dependence of NOx production on CN, T90, and Main SOI

The results presented in Figure 38 show that among the variable presented Main SOI is the strongest, and can only slightly be tempered by fuel characteristics as Cetane Number and distillation temperature. This plot also shows how an increase in Cetane Number leads to higher NO_x production. This statement is in contradiction with the conclusions drawn by recent studies [18]. The effect of CN on emissions has been misinterpreted as a consequence of the confounding effect of Main SOI. As we explained before the use of higher CN fuels imply the necessity of using more advanced injection timings. When the raw data are compared the effect of Main SOI can mask the other factors influence. The second strongest interaction depicted by the ANOVA table is between CN, main SOI and Pilot SOI. This three variable joint interaction is shown in Figure 39.

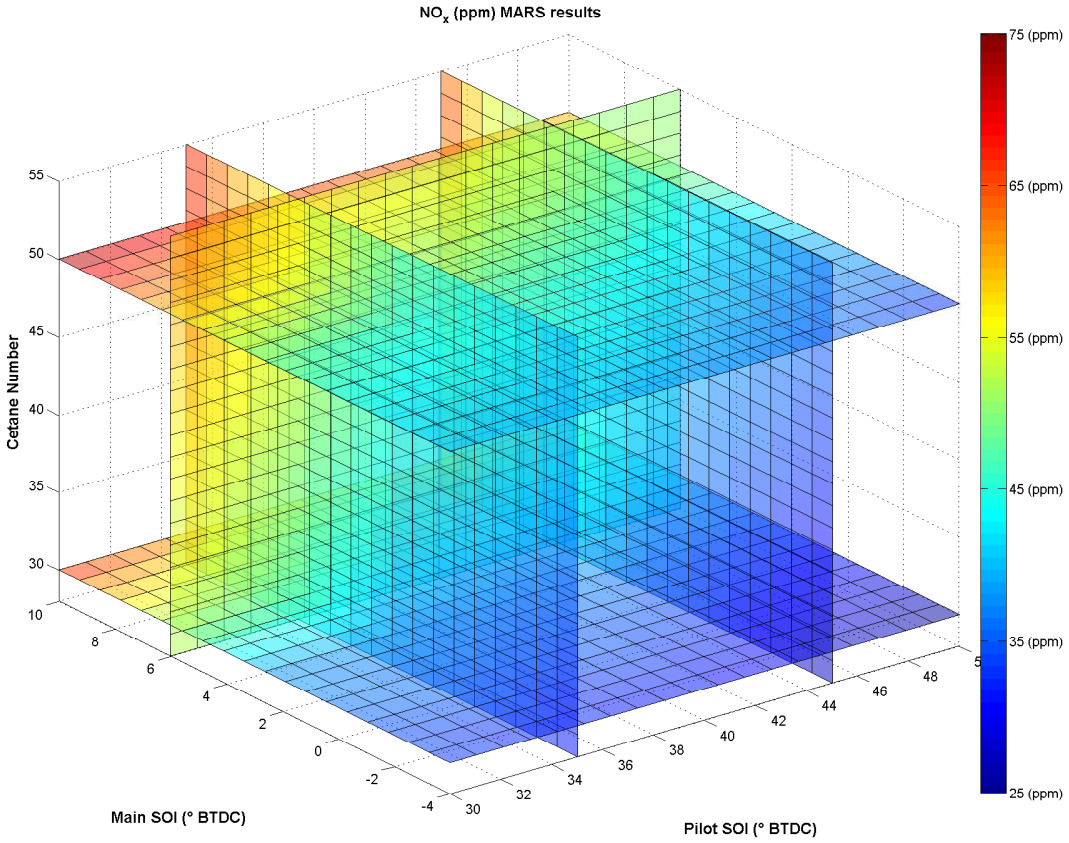


Figure 39: Dependence of NOx production on CN, Main SOI, and Pilot SOI

The response displayed in Figure 39 is the same of Figure 38 except for the pilot SOI instead of distillation temperature. The influence of pilot SOI on NO_x production is similar to the one of main SOI. The other interactions are either not very significant or already displayed in the section relative to injection characteristics, the plots are reported in appendix .

Soot Concentration

In this section the results relative to PM analysis will be displayed. Following the same structure of the previous section the first step is to analyze the quality of the regression. Figure 40 shows the experimental results compared with the emulator model. The first thing to notice is the high variability in the data, the analysis of the training data showed a standard deviation over 18%,

which clearly represent a challenge for the regression algorithm. The R^2 value obtained is 75% which is still in the range of acceptability.

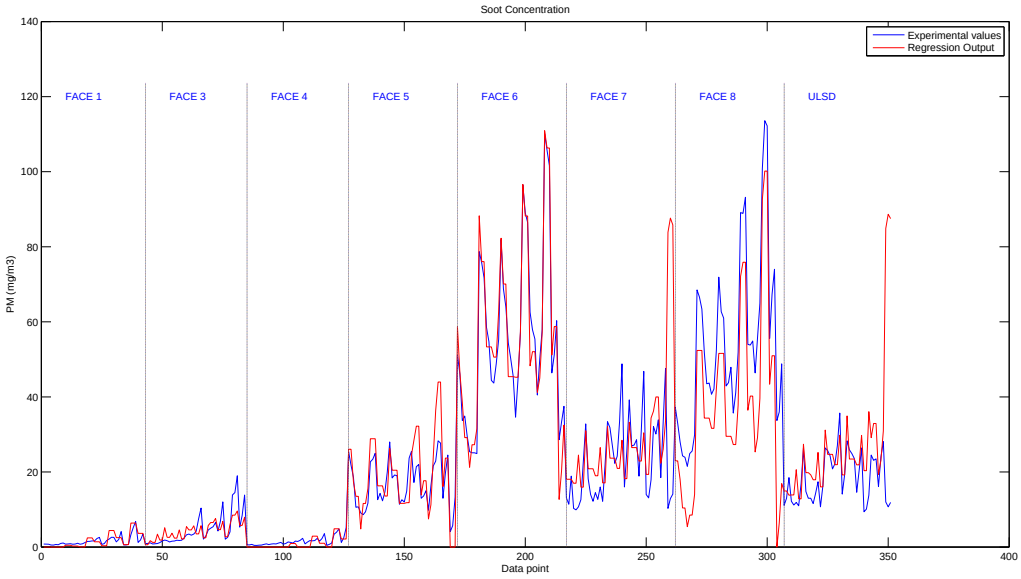


Figure 40: Comparison between experimental data and regression results

Figure 40 shows how widely the soot emissions vary among the fuels tested, with FACE 8 and FACE 6 being the ones producing by far more soot. The emulator model is then analyzed using the ANOVA procedure; the results of such analysis are summarized in the following table 7.

Function	**STD**	**GCV**	**# of basis**	**Variables**
1	8.4	87.3	2	1
2	7.5	78.5	2	4
3	5.6	57.6	2	5
4	1.3	20.9	1	3 4
5	6.2	44.5	2	1 2
6	5.2	76.5	1	2 5
7	1.1	18.2	1	4 5
8	0.6	14.6	1	4 6
9	5.2	39.7	1	1 2 4
10	1.3	4.4	1	1 4 5
11	0.5	2.1	1	1 4 6

Table 7: ANOVA Decomposition PM

The two single variables with the strongest impact on Soot emissions are Cetane number and main SOI as expect. Figure 41 and figure 42 show the contribution of those variables to the total PM emitted; in both cases three different trends have been produced.

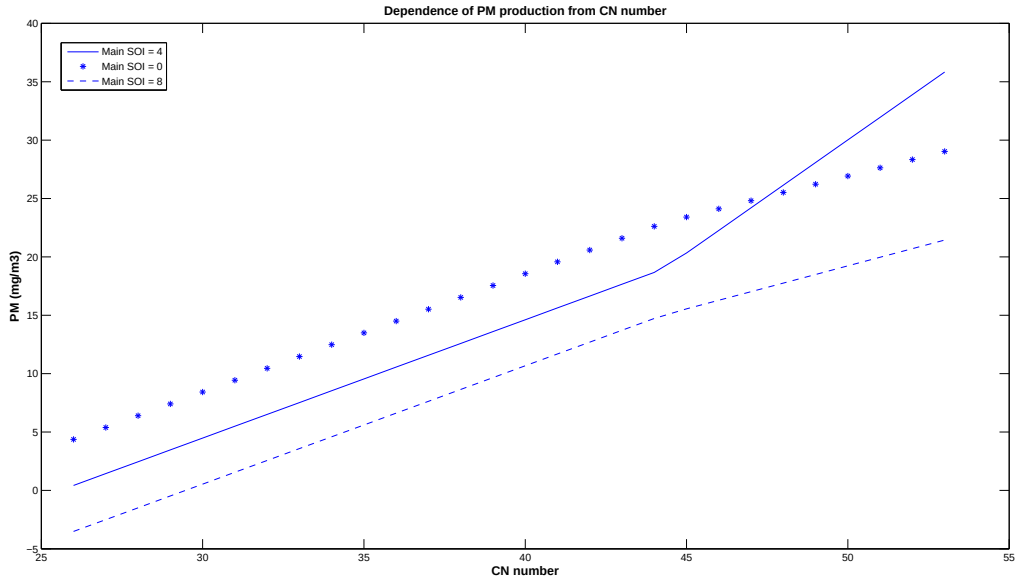


Figure 41: Dependence of soot production on Cetane number

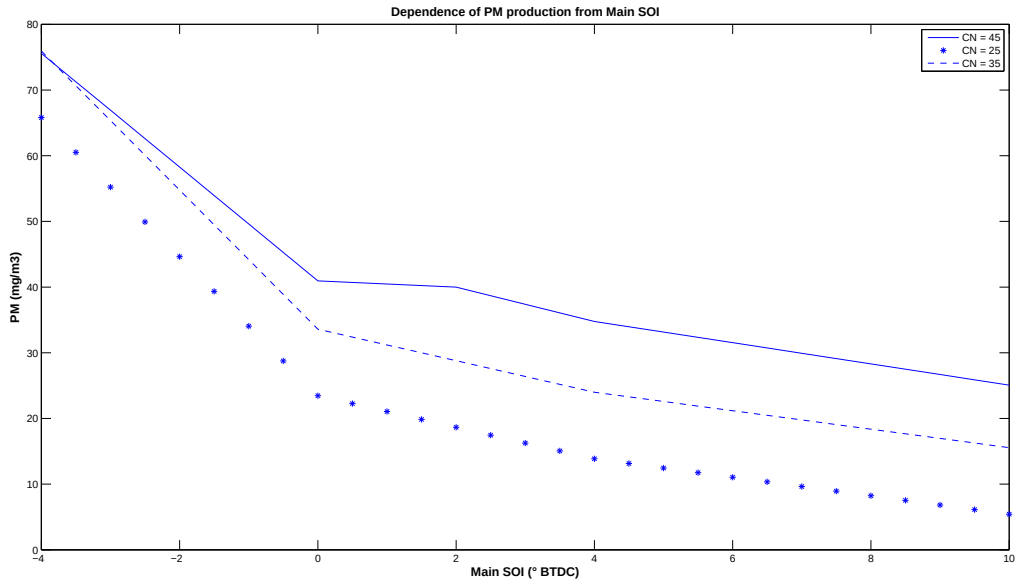


Figure 42: Dependence of soot production on main SOI

The soot emissions increase with Cetane number, as depicted from figure 41. A noticeable

interaction appears at high Cetane number with the Main SOI. Figure 42 shows the dependence of soot production from main SOI, it shows that the lowest values of PM are achieved at the most advanced injection timing. The combination low Cetane number and advanced injection produce the minimal soot value. When analyzing join interaction between variables the one to show the strongest effect is between CN and T90. Several studies confirmed the observation that T90 plays a significant role in soot formation([44] [45] [46]); its increase leads to higher soot emissions. Figure43 shows the interaction between T90 and Main SOI.

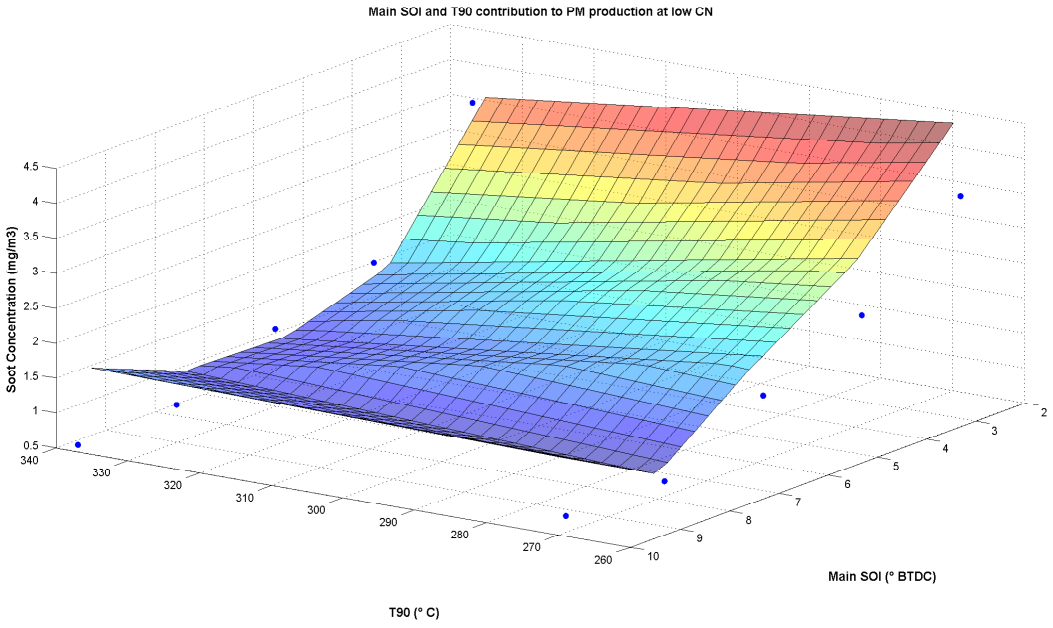


Figure 43: Dependence of soot production on distillation temperature and main SOI

The second strongest interaction is between Cetane number and aromatic content. Figure 44 shows that the effect of CN is very marked, and it also strongly affects the way other parameters act on soot formation.

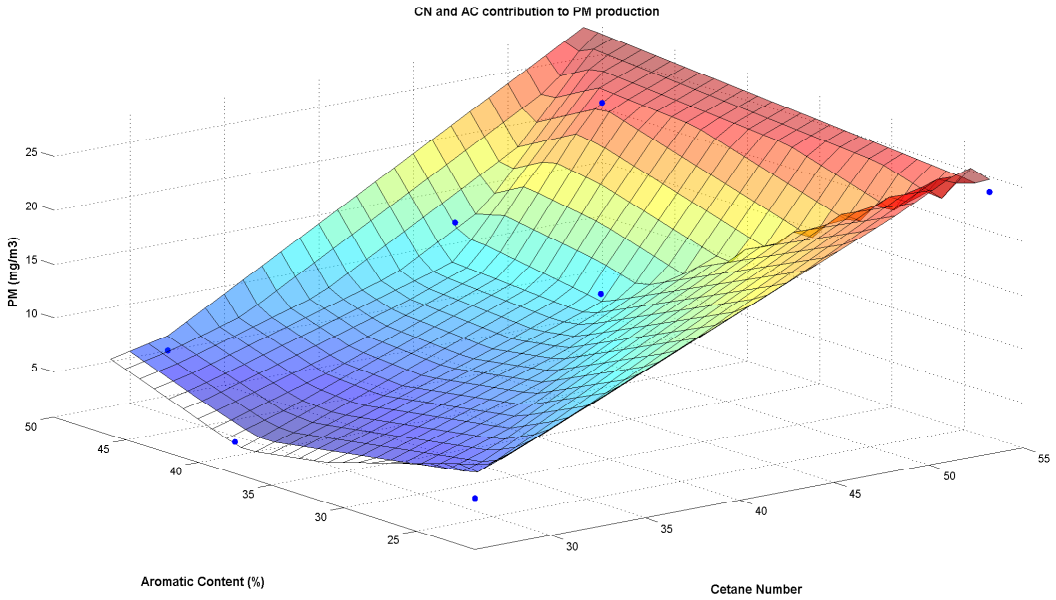


Figure 44: Dependence of soot production on Cetane number and aromatic content

The following two figures (45 46) show the behavior with respect to pilot SOI and aromatic content. A distinction has been made between results produced setting a high Cetane number (50) and low Cetane number (30). This is done because, as it is noticeable from figures 45 46, the interactions between those variables change significantly with different levels of CN. In both cases an advanced pilot injection is beneficial, while in general higher aromatic content lead to higher PM emissions. The influence of the aromatic content is stronger and less linear in the low CN number, and becomes more important for retarded pilot injections.

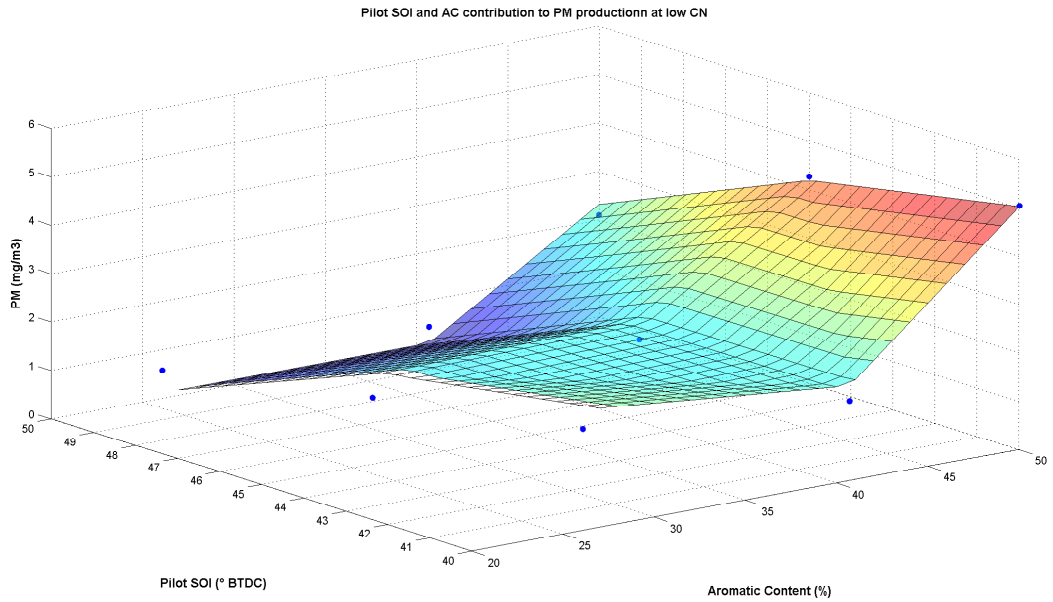


Figure 45: Dependence of soot production on aromatic content and pilot SOI at low Cetane number (CN=30)

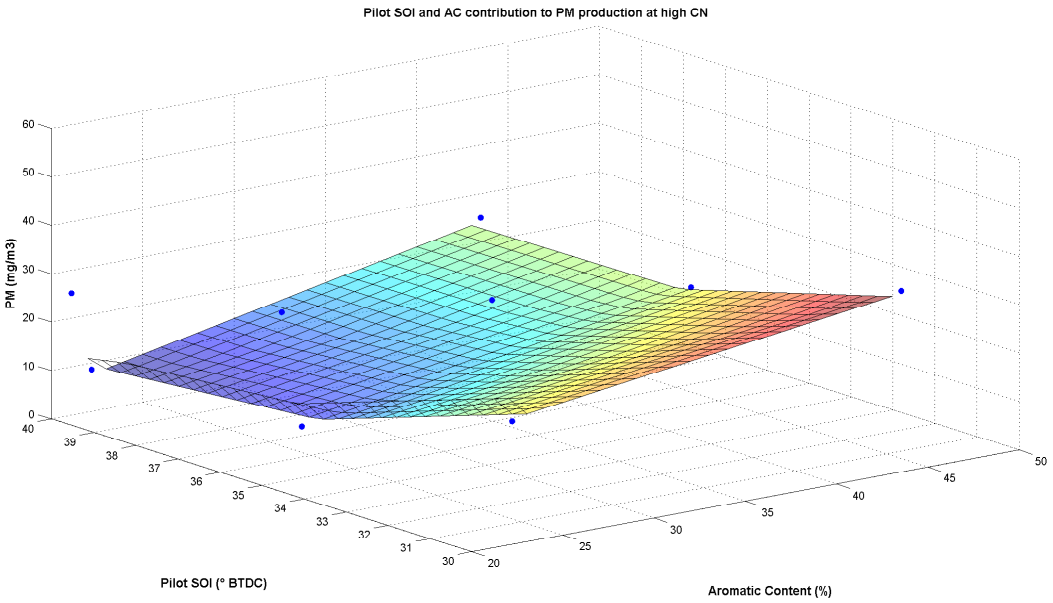


Figure 46: Dependence of soot production on aromatic content and pilot SOI at high Cetane number (CN=50)

The figures shown in this section are the one where more significant aspects of PM production are underlined, the rest of the plots are collected in the appendix .

Carbon Oxide

The quality of the fitting regarding the CO emissions is displayed in figure 47. The R^2 value obtained is of 71%, but as we can see from figure 47 the regression is optimal everywhere except for FACE 5 and FACE 6 fuels. Those fuels are characterized by a combination of high Cetane number and low aromatic content which generates a disagreement between the model results and the experimental data. Another source of uncertainty is non repeatability of the experimental data as depicted in section .

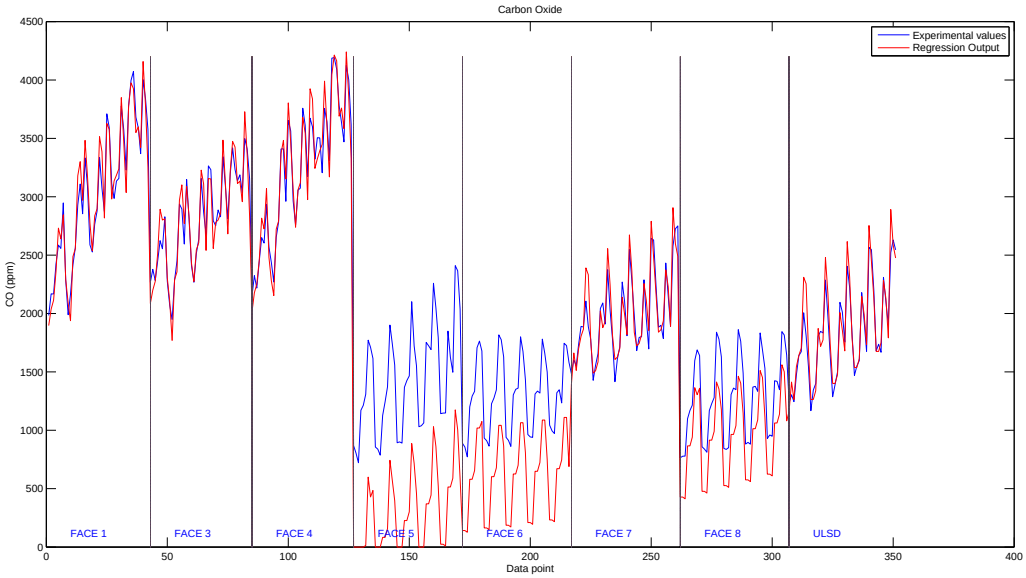


Figure 47: Comparison between experimental data and regression results

Table 8 summarize the results of ANOVA analysis regarding the CO emulator model.

Function	STD	GCV	# of basis	Variables
1	4.3	54.4	2	1
2	0.5	5.8	1	4
3	2.6	2.6	2	5
4	1.3	20.9	1	1 4
5	6.2	24.5	2	4 5
6	5.2	36.7	1	4 6
7	1.1	18.2	1	1 4 5
8	0.6	14.6	1	1 5 6
9	5.2	39.7	1	4 5 6

Table 8: ANOVA Decomposition CO

The fuel characteristic with the strongest impact on CO emissions appears to be CN. The following plot (figure 48) shows the response in terms of CO to different levels of CN. Lower CN lead to higher CO production, confirming the observations of other studies [3].

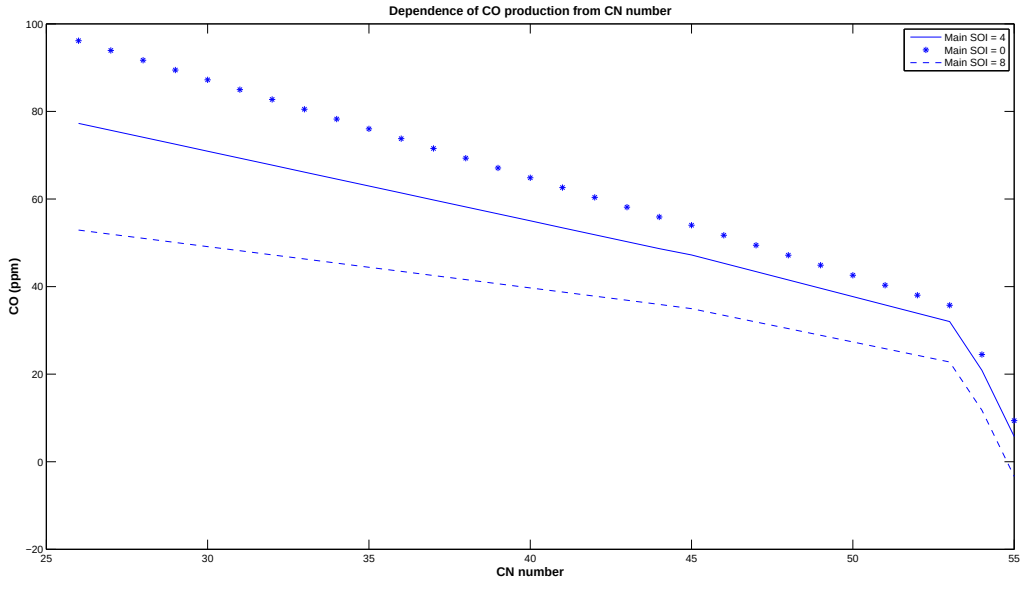


Figure 48: Dependence of CO from Cetane number

The other single variables to have an impact on CO emissions are all injection characteristics which have already been covered in the section . The two variable with the most relevant interaction are CN and main SOI, the surface plot is displayed in Figure 49

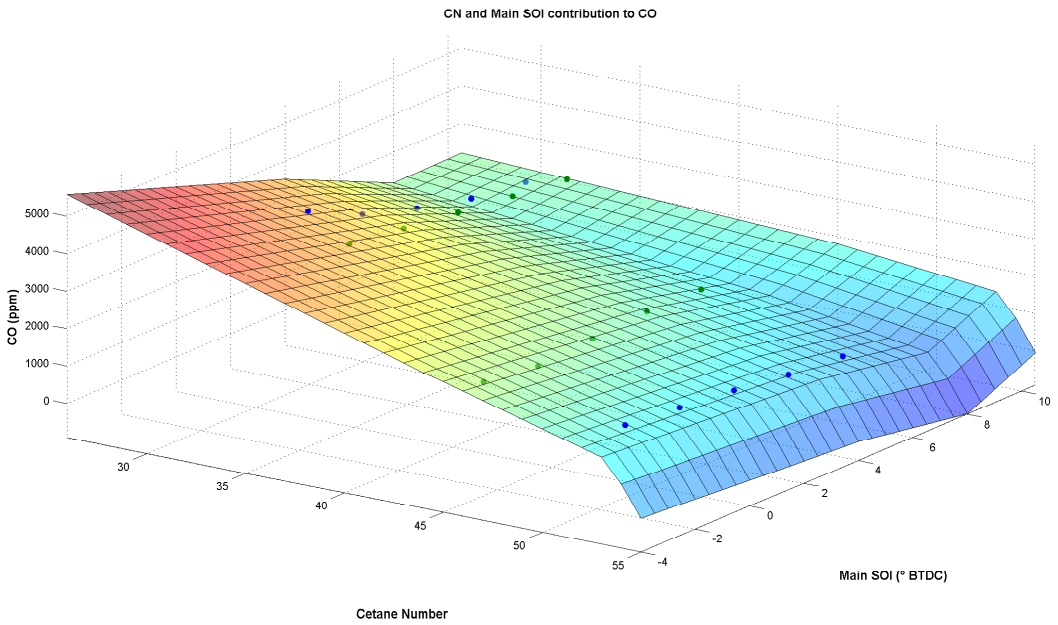


Figure 49: Dependence of CO on Cetane number and main SOI

The same interaction including also pilot injection is visualized in figure 50. It is interesting to notice that pilot SOI has an impact which is stronger than main SOI.

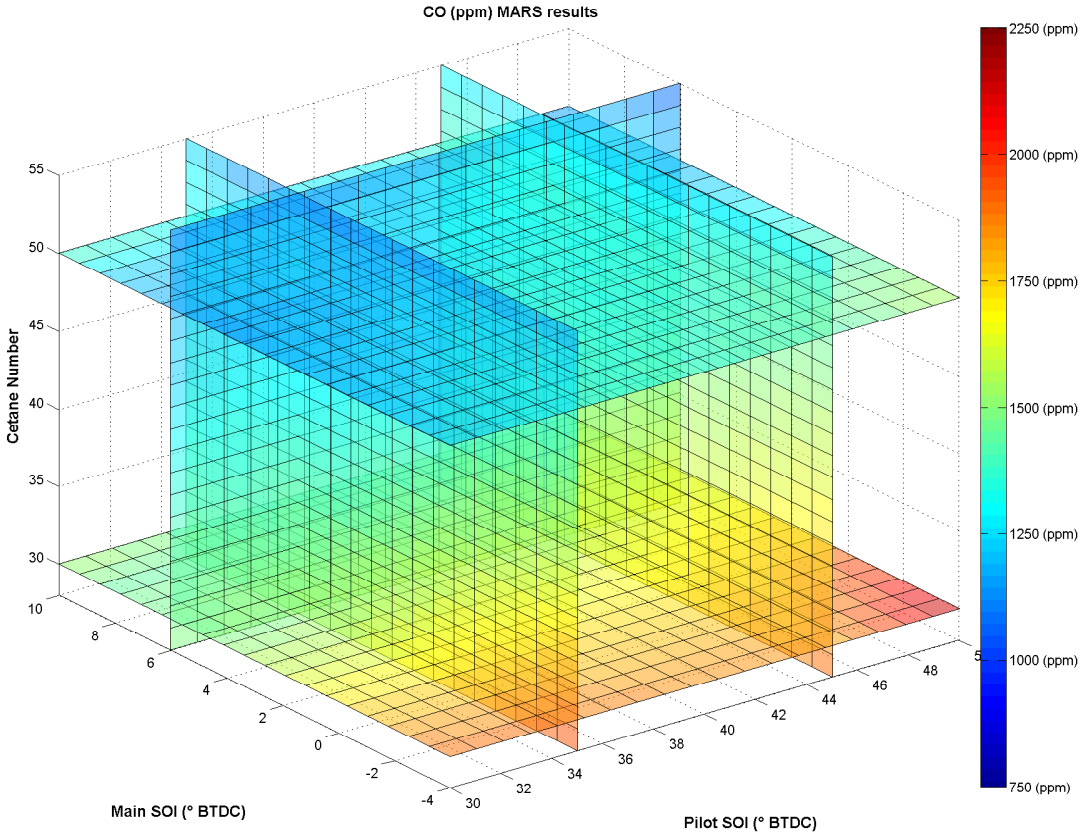


Figure 50: Dependence of CO from Cetane number, pilot SOI, and main SOI

Hydrocarbon

This section describes the modeling of hydrocarbons. The regression quality for this variable is very accurate as depicted from figure 51, and also by a R^2 value of 90%.

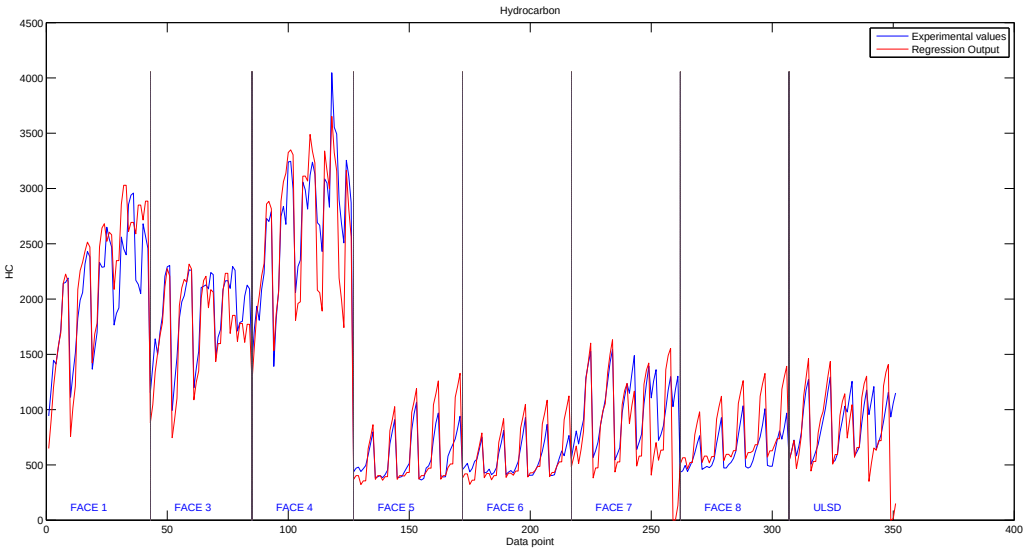


Figure 51: Comparison between experimental data and regression results

Function	**STD**	**GCV**	**# of basis**	**Variables**
1	6.4	74.3	2	1
2	2.5	28.5	2	2
3	5.6	12.6	3	4
4	1.3	2.9	1	5
5	0.7	1.5	1	6
6	4.2	36.7	1	1 4
7	1.1	18.2	1	2 4
8	5.6	14.6	2	2 3 4
9	4.5	39.7	1	1 4 5

Table 9: ANOVA Decomposition HC

The first characteristic listed in table 9 is the Cetane number, which is displayed in figure 52 for three different values. The plot shows a peak in HC at low CN numbers, which rapidly decreases.

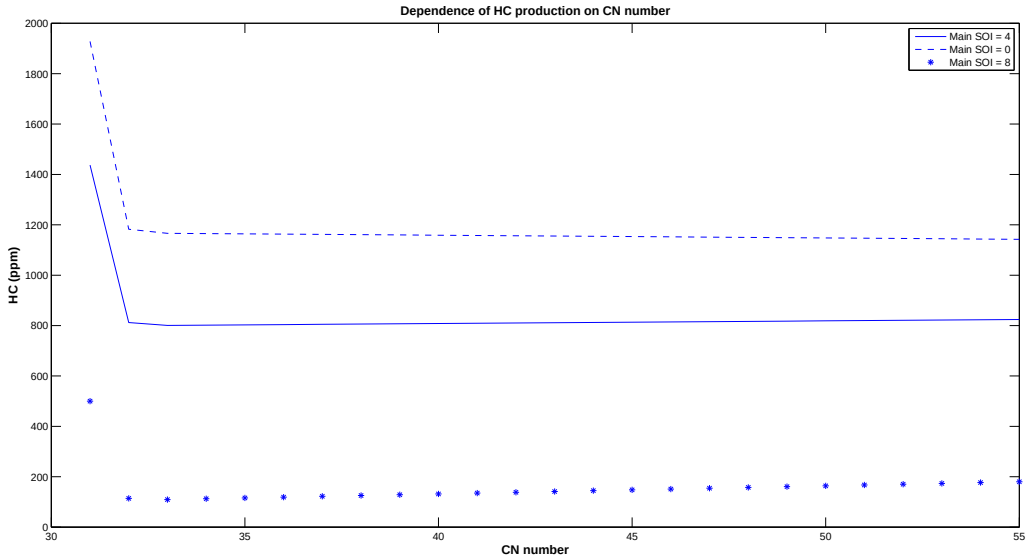


Figure 52: Dependence of HC on Cetane number

The curves in figure 52 are representative of three values of main SOI, and it shows the impact of this factor on the response is nonlinear. According to the ANOVA analysis the second most influential parameter is the aromatic content. From figure 53 we notice the negative impact of high AC on HC. Considering that AC represent the specific amount of energy in the fuel and that HC are leftover of unburned fuel, this behavior can be considered to be as an incapacity of the combustion to utilize all the energy available in the fuel.

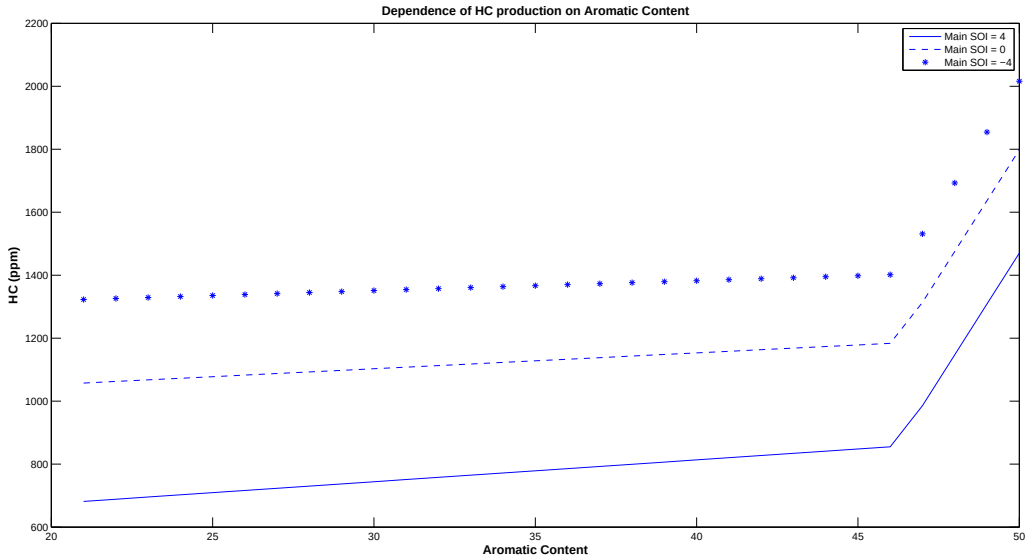


Figure 53: Dependence of HC on aromatic content

The deduction of a nonlinear effect of the main SOI is confirmed by plotting the HC response with respect to this factor (figure 54).

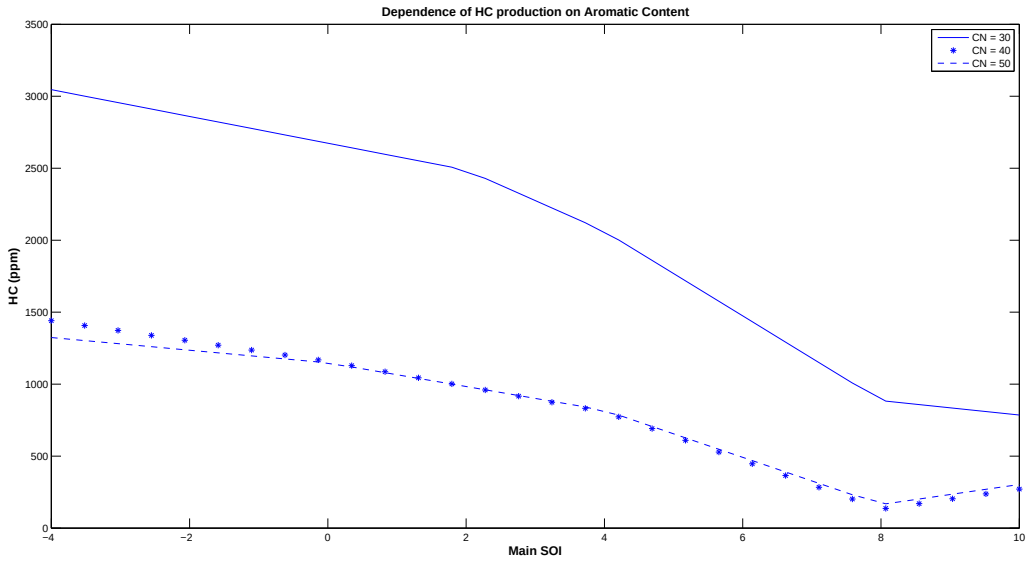


Figure 54: Dependence of HC on main SOI

It is now interesting to look at the join interaction between those two factors. Figure 55 shows how higher CN leads to lower emissions of HC and also to a less significant impact of the main SOI.

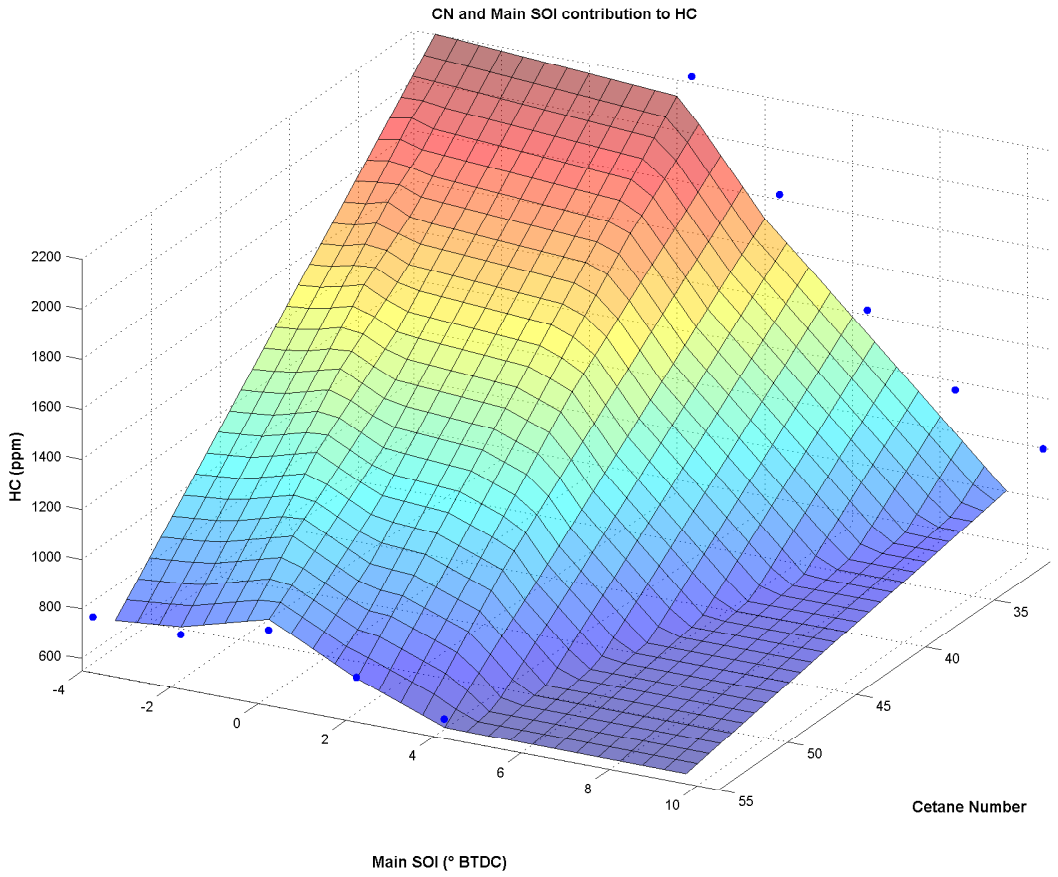


Figure 55: Dependence of HC on main SOI, and cetane number

From a three way prospective the most interesting results are obtained looking at the interaction between main SOI, distillation temperature and aromatic content as depicted in figure 56.

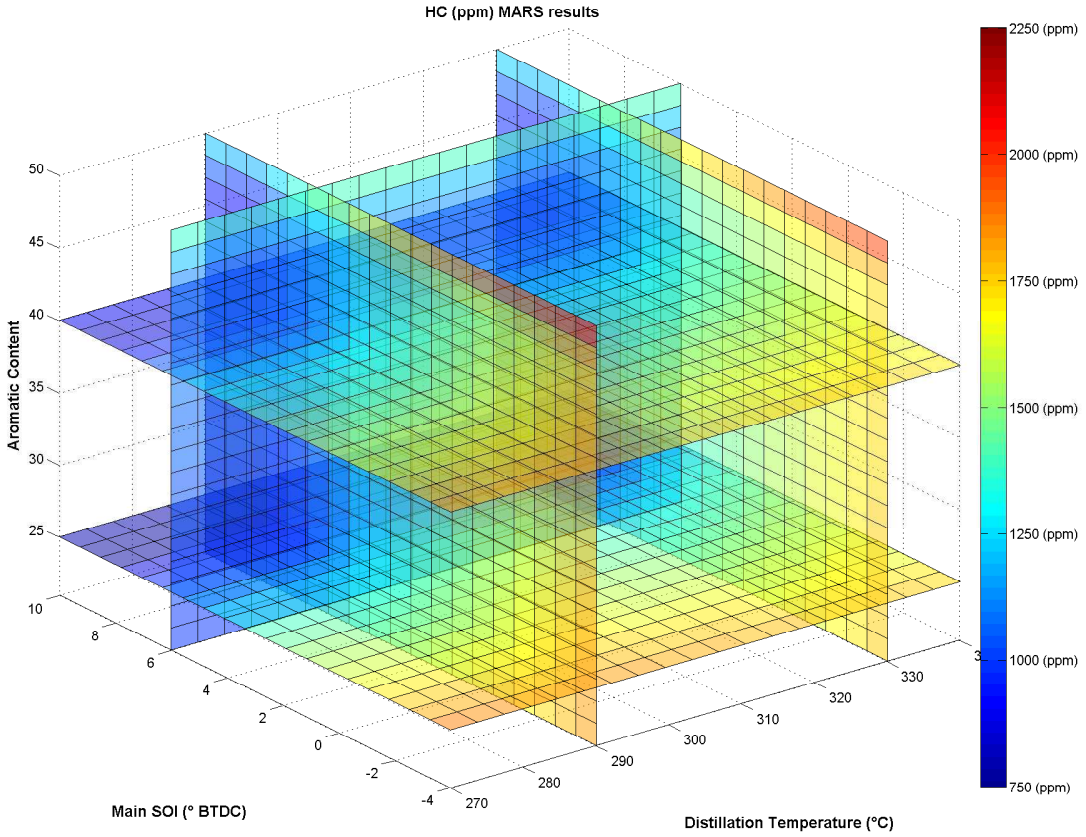


Figure 56: Dependence of HC on main SOI, distillation temperature, and aromatic content

Verification Study

The model built to analyze the full interaction between fuel characteristics and injection strategy has been analyzed in comparison with the experimental data used to train the model itself. A rigorous validation of the model would require the comparison with experimental points not included in the training set. This kind of validation is aimed to judge the prediction capability of the model. Since the model is generated through regression of the data, it is necessary to determine if the behavior of the emission characteristics can be extrapolated or it is completely unrelated to the data surrounding it.

In this section the validation of the model is performed using an unconventional approach. Instead of building the model using less than all the data available, and then using the remaining for validation, here the model presented was trained using all the information available. A mockup model is then generated just for validation purposes excluding some of the available data, and in this section the prediction capability of this secondary model is investigated. The justification for this approach is related to the pursuit of the most refined model to describe the LTC process, while the exclusion of some of the data from the training process would have led to a lost in accuracy.

The mockup model was generated using all the data except for seven points that will be used for validation. The validation subset is determined by randomly selecting seven points belonging to the original dataset. The only constrain applied to the data selection is that each point should pertain to a different fuel in order to guarantee the generality of the validation, and to make sure that the regression is effective on each fuel. Table 10 lists the points selected for validation, and their respective factor levels, in the rest of this section each of this points will be referred to using the fuel name.

CN	**AC**	**T90**	**Main SOI**	**Pilot SOI**	**Fuel Split**	**Fuel name**
29.93	26.1	269	8	50	40	**FACE 1**
32.02	50	270	6	50	40	**FACE 3**
28.44	40.7	337	4	45	35	**FACE 4**
54.2	22.2	279	-4	30	30	**FACE 5**
53.3	21.1	341	-4	30	40	**FACE 6**
50	43.5	342	0	30	30	**FACE 8**
44.95	37	321	2	235	40	**ULSD**

Table 10: Data points used for calibration

Figure 57 shows the comparison between the experimental points selected for the verification, and the values obtained using the mockup model. The response in figure 57 is NOx, and the plot clearly shows a good fitting between the predicted values and the experimental points.

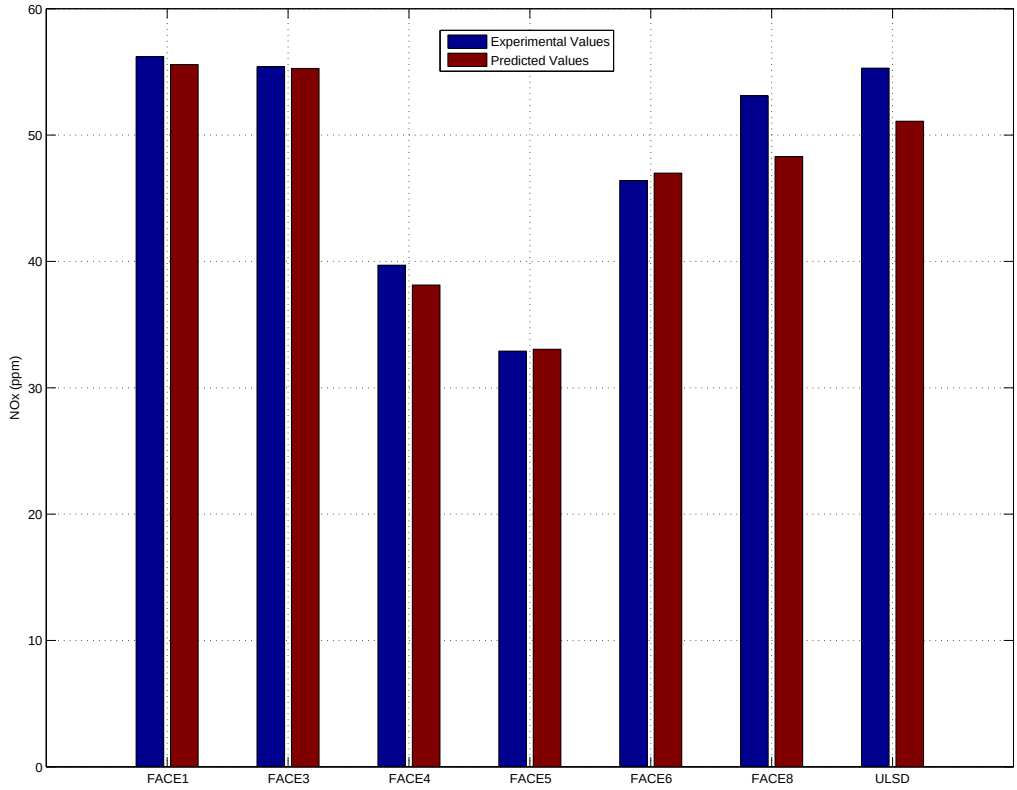


Figure 57: Comparison between NOx experimental values and predicted results

Figure 58 displays the prediction accuracy for the other four responses.

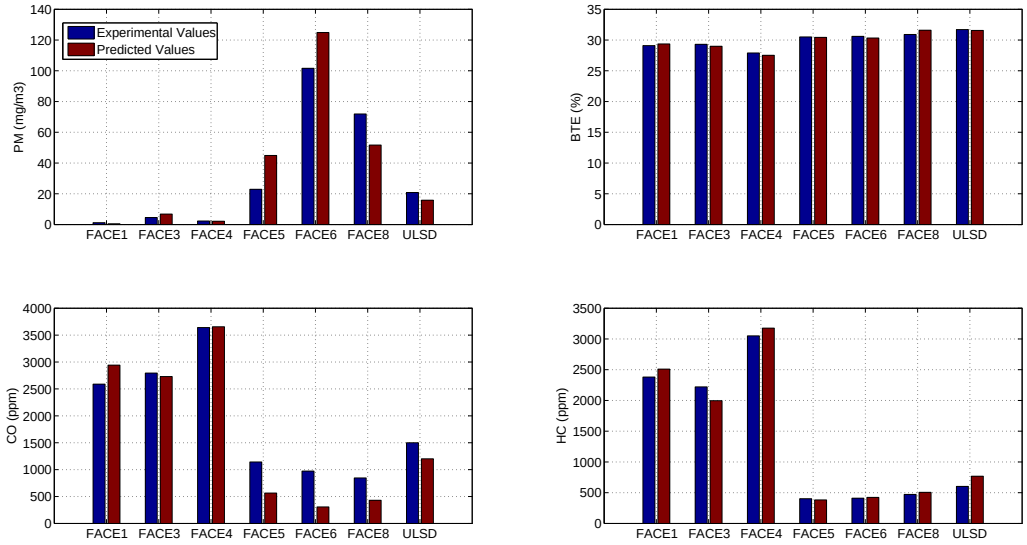


Figure 58: Comparison between experimental values and predicted results for CO, HC, PM, and BTE

Table 11 summarize the prediction performance with reference to the results displayed above. For each characteristic the maximum difference between the predicted value and the experimental value is displayed. Max err. % is the ratio between the error and the range of variation of the response. R^2 is conventionally defined as the ratio between the residual sum of squares and the total sum of square. These two parameters are usually proportional one to the other. The last column collects the values determined during the repeatability study.

Response	**Max err**	**Max err.** %	R^2	**Repeatability** %
NOx (ppm)	4.8	7.6	0.91	3
PM (mg/m3)	23.3	20.6	0.81	19
BTE (%)	0.69	11.9	0.91	1
CO (ppm)	666.3	19.2	0.83	8
HC (ppm)	225.12	6.1	0.98	3

Table 11: Prediction model errors

Comparing the results of this verification study with the threshold values defined during the repeatability study, we notice that there is a strong correlation between the quality of the model and the repeatability of the data. The repeatability indicates a variation in the output which is not controllable in the experimental apparatus used to collect the data. The regression model is generated starting from these same data; hence its prediction capability could never overcome the data quality.

Optimization

Once the emulator model is obtained by surface fit, it can be used to predict responses inside the factor space. It should be noted that the regression model should not be used for extrapolation outside the factor range. Most importantly, the model can be used to conduct optimization to search for the optima located on the response surface. In the optimization, the stationary point refers to the point of factor settings corresponding to zero partial derivatives of the response with respect to all the factors.

Multi-Objective Optimization Engine optimization is a multi-objective problem (MOP), as several objective functions are of interest. Hence a multi-objective solution always represents a trade-off in the MOP. In this family of problem the notion of optimum is referred as the Pareto optimum [47].

A visual interpretation of the Pareto optima can be given by considering just two variables, so to be able to plot it on a plane. The data collected are plotted in Pareto chart (figure 59), the two axes represent the quantity to minimize, i.e. NOx and Soot concentration. This representation of the output is useful to have a rapid perception of the optima points, those will be the closest to the origin.

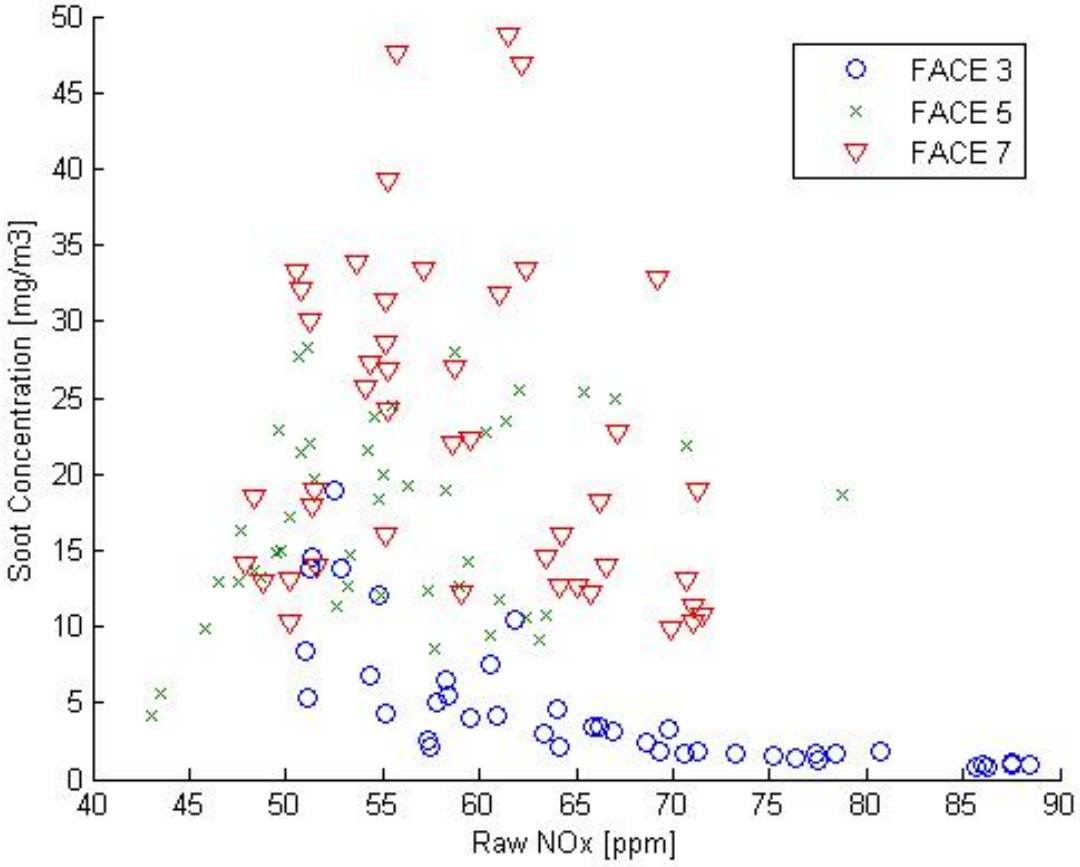


Figure 59: Distribution of the experimental points on a PM-NOx Pareto chart

Weighted sum method A traditional method for multi-objective optimization is the weighted sum method, which seeks the Pareto optimal solution by combining several objective functions into one. We can formulate a general MOP in the following form:

$$\begin{cases} minY(x,\beta) \\ g(x,\beta) < 0 \end{cases} \tag{26}$$

Where $Y = [y_1, ..., y_z]^T$ is an objective function vector, x is the factors vector, β is the regression

coefficients vector, and g is an inequality constraint vector. The weighted sum method consists in reducing the objective function vector to a scalar of the form:

$$\hat{Y} = \sum_{i=1}^{z} \lambda_i / s_i f_i y_i \tag{27}$$

Where s, f_i, and λ_i are the scale factors and the of the i-th objective, respectively. Typically, weights are chosen in such a way that their norm is one, and that none is negative.

The main drawback of this relatively simple method is that an even distribution of weights among objective functions does not always result in an even distribution of solutions on the Pareto front. This issue is particularly relevant in the problem studied in this work since the objective functions are expressed in different unit. In order to overcome this problem the weights have been chosen proportionally to the magnitude of the response for the given objective function, i.e.:

$$\lambda_i = (\sum_{j=1}^{z} max(y_j) - max(y_i))/(\sum_{j=1}^{z} max(y_j)) \tag{28}$$

With this weight selection the two objective functions assume equal importance in $\hat{Y}$.

The parameter s instead is chosen based on the results obtained from the compatibility study. A categorical function is associated with every set of independent variables in order to account for improper combustion. Its value is 1 in case of proper combustion and 0 when the combustion process is not achievable. By inserting this value at the denominator of the cumulative objective function it is guaranteed that the minimum will correspond to an allowable combustion process.

The weight factor $f_i$is used to determine the relative importance of each emulator function in the global model. For each output the factor f_i is determined by evaluating the density of each response above a certain threshold. For example the f_i relative to the NOx response corresponds to the percentage of experimental data above the value of 50 ppm, which is determined to be the threshold for acceptable NOx emissions. This parameter is then used to express how critical is each response to the overall optimization, because those characteristics with a larger distribution of points above the allowable level will be considered more important.

Once that we defined a scalar function that accounts for both objective the optimum can be determined using standard multivariate analysis. The global optima is determined by the following factors:

Cetane Number	30
Aromatic Content	26
Distillation temperature (°C)	276
Main SOI (°BTDC)	2
Pilot SOI (°BTDC)	45
Fuel Split	30

Table 12: Set of parameters producing a global unconstrained optima in emission characteristics

The corresponding emission and performance responses are summarized in table 13. The % difference from the target represent the ratio between the result and the range of values that the given characteristic assume among the available data.

Characteristc	**MOP values**	**% difference from target**
NOx (ppm)	30.5	6.1
Soot Concentration (mg/m3)	1.2	0.7
CO (ppm)	4000	0.7
HC (ppm)	2650	62
BTE %	29.1	63.8

Table 13: Responses corresponding to the global unconstrained optima

The values obtained represent the optima when no constrain is applied to the minimization method. This means that the characteristics which contribute the most to the global function have a stronger impact, namely NOx and PM, while the other are out of the range of acceptability. When describing low temperature combustion it is often accepted to produce high CO and HC, especially considering that those characteristics are easily retrofitted with a diesel oxidation catalyst (DOC). It is important to underline that the optimal point is refered to the load conditions reported in the

experimental setup section.

Constrained Optimization In the attempt to obtain an optima which will represent a better compromise among all the characteristics under examination a constrained optimization was performed. This means that the candidate set of factors for optimization must produce responses that lay within a range of acceptability. The threshold for the acceptability are reported in table 15. The set of factors corresponding to this optimization process is displayed in table 14. The CN in this case is much higher than the one obtained by unconstrained optimization, which according to the results displayed in this study leads to leaner HC and CO emissions. To compensate the increase in NOx and PM related to the selection of this fuel chemistry the Main SOI is retarded considerably.

Cetane Number	54
Aromatic Content	22
Distillation temperature (°C)	270
Main SOI (°BTDC)	-4
Pilot SOI (°BTDC)	40
Fuel Split	30

Table 14: Set of parameters producing a global constrained optima in emission characteristics

Characteristc	**Threshold**	**MOP values**	**% difference from target**
NOx (ppm)	50	26.6	0.5
Soot Concentration (mg/m3)	10	4.1	3.5
CO (ppm)	3000	2410	48.6
HC (ppm)	1000	735	10.1
BTE %	29	29.5	67.2

Table 15: Responses corresponding to the global constrained optima

Table 15 shows the results of applying constrained optimization. Except for PM production all

the other characteristics lay closer to the target value. The soot formation is still largely below the threshold, making the result of constrained optimization a better candidate for this scope.

Further studies based on MARS

The code described in this work has been developed with the specific objective to help the emission analyst in studying a given problem in the field of engines and emissions. In this chapter two studies where MARS has been successfully applied are reported.

The first one consists in the development of a simulation tool for engine testing, aimed to uncover engine map behavior dependently on the operation regime. This scenario is not drastically different from the one addressed in this work, merely from a regression point of view. The output corresponding to a set of tests is analyzed and based on the outcome some deduction can be made.

The second study instead does not relate to engine lab, instead it is aimed to the description of plume emitted from heavy duty diesel truck. Data are collected in a wind tunnel specifically designed to conduct experiments on plume evolution in the atmosphere. Several points behind a mockup heavy-duty truck were sampled in a three dimensional domain. MARS is used to produce a map of particulate matter plume starting from experimental points. Once the discrete data set is converted into continuous functions a better visualization and analysis of the plume evolution is possible. The main focus of the analysis is on the influence that dilution ratio and cooling velocity have on the particle size and distribution.

Diesel Engine Modeling Development for ICCT Heavy-Duty Vehicle Simulation Tool

This study is aimed to the characterization of the fuel consumption in a MY 2005 Mercedes depending on operation parameters. Dynamometer testing was performed over four test cycles; namely

Federal Test Procedure (FTP) test cycle, European Stationary Cycle (ESC), and two space filling designs matrix. The space filling design matrixes were generated using respectively Latin Hypercube and Gaussian process. Figure 60 shows the tested points under the lug curve. Engine was instrumented for coolant temp, oil temp, oil pressure, EGR circuit temperature, coolant flow, in-cylinder pressure, turbo enthalpy drop. Figure 61 shows the laboratory setup.

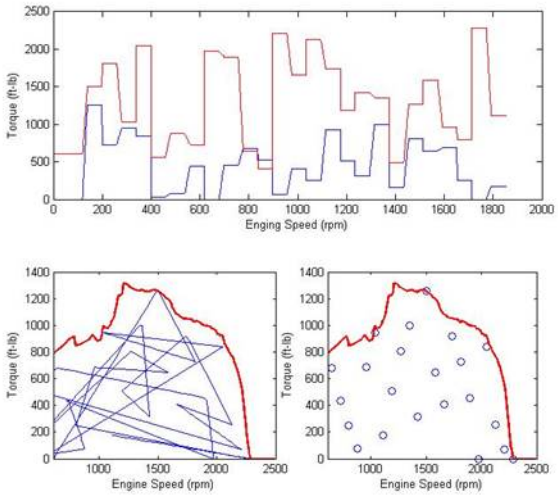


Figure 60: Distribution of the experimental points on a load-speed chart

Figure 61: MY 2005 Mercedes engine

Using the data relative only to the FTP cycle an emulator equation was obtained using the MARS code. The equation was then interrogated over the points tested over all of the four test cycles. Figure 62 shows the comparison between the model results and the ESC data. The emulator equation fits 100% of the data, both those used to generate the model and the validation data.

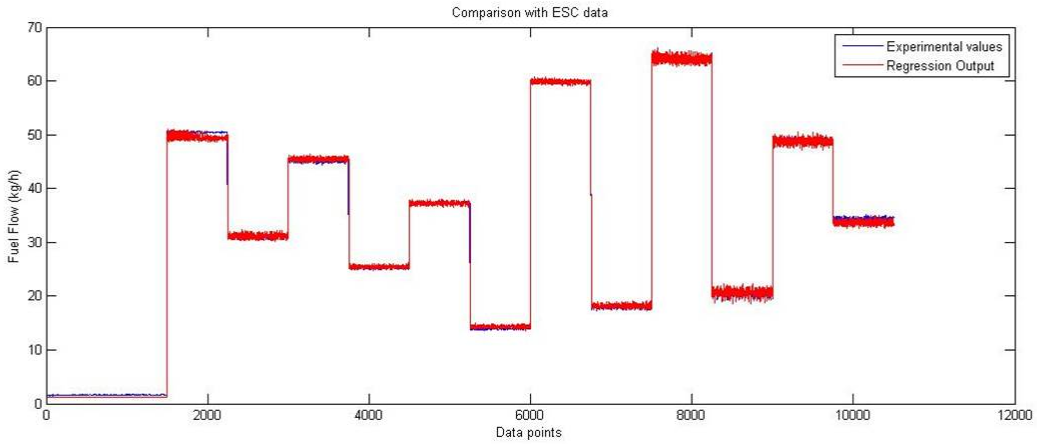


Figure 62: Comparison between MARS results and ESC data

Figure 63 shows the behavior of fuel consumption with respect to engine load and speed. The blue scattered points represent the experimental data. The quality of the fitting suggests that MARS could be used to generate engine maps, by substituting lookup table with the emulator function.

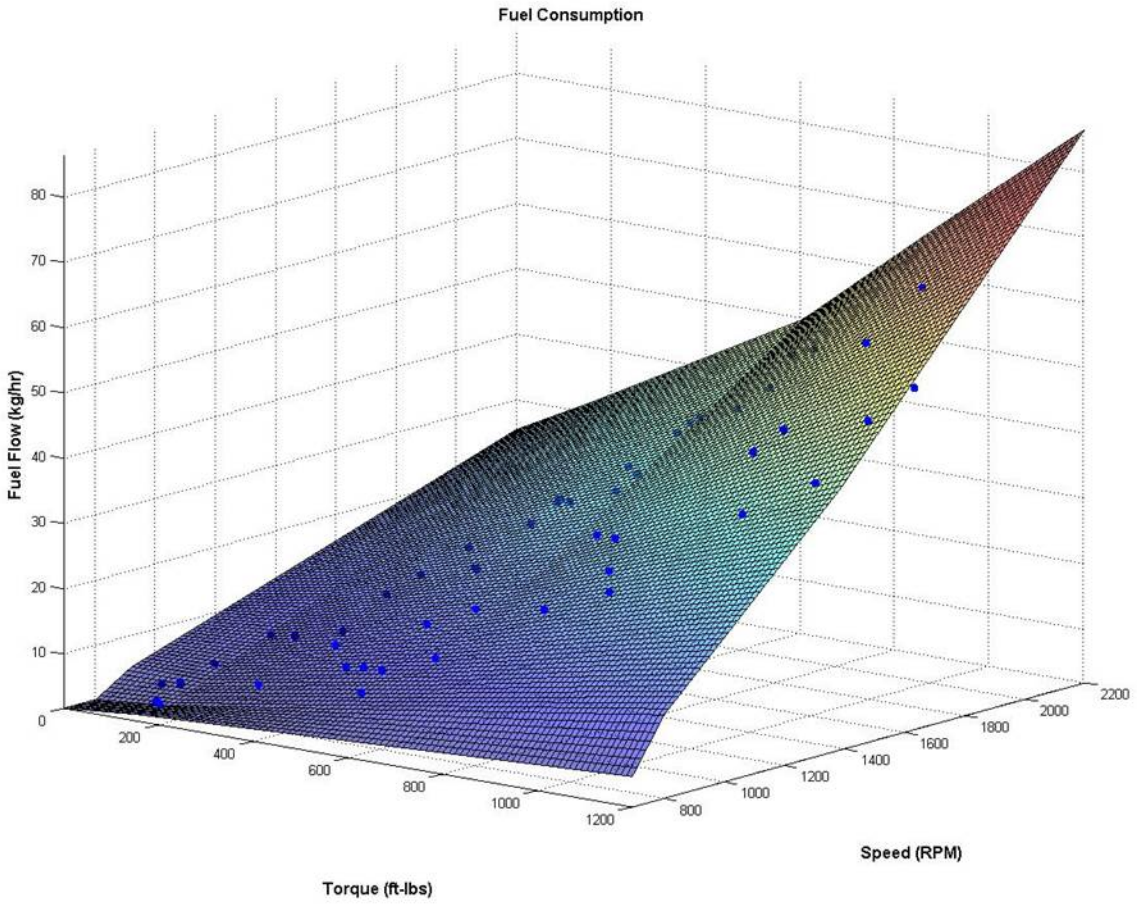


Figure 63: Fuel consumption surface plot

The next step of this study will be to compare the MARS results obtained using stationary test points with transient results. The aim is to have a reverse engineering tool able to extrapolate the engine map used by the data. This phase of the project is still in progress.

Analyze Dispersing Plume from Heavy Duty Diesel Trucks

The objective of this work is to investigate evolution of particle number distributions in the tailpipe plume of a vehicle using experimental data collected at WVU‘s wind tunnel (WT) facility coupled with the regression analysis method. Diesel PM remains in a state of continuous transformation (unstable) for some time after it is emitted into the atmosphere. The fate of these condensable organics/inorganics is significantly affected by the dilution and atmospheric aging of the exhaust

stream. A number of processes occur during atmospheric aging that can alter the size distribution of an aerosol, including homogeneous nucleation, binary homogeneous nucleation and coagulation.

Experimental Setup The WVU wind tunnel is a full-scale, open-circuit, suction tunnel. The open-circuit configuration is the key feature that guarantees fresh ambient background air flow over the vehicle and therefore, approximately constant dilution air composition and conditions for the entire test duration. It is capable to accommodate a full scale class 8 heavy-duty diesel truck, matching as closely as possible real world conditions. The test section is 16ft (5m) high, 16ft (5m) wide and 115ft (35m) long.

The instrumentation required to characterize the exhaust plume could not be placed within the flow field without strongly influencing it. However a single sampling probe could extract a localized fraction of the plume and redirect the sample to the instruments. In order to minimize the length of the transfer line connecting sampling probe and instrumentation and hence, the particulate matter losses, a novel solution has been adopted by implementing a cart carrying the entire suite of instruments and moving longitudinally within the ceiling of the tunnel.

In order to supply a continuous stream of exhaust for the exhaust plume interrogation within the tunnel, a vehicle is installed on a heavy-duty chassis-dynamometer, located outside the wind tunnel, and operated at a pre-defined vehicle speed. Inside the WVU wind tunnel a mock up cabin is used to generate the truck aerodynamics. Figure 64 shows the layout of the experimental setup.

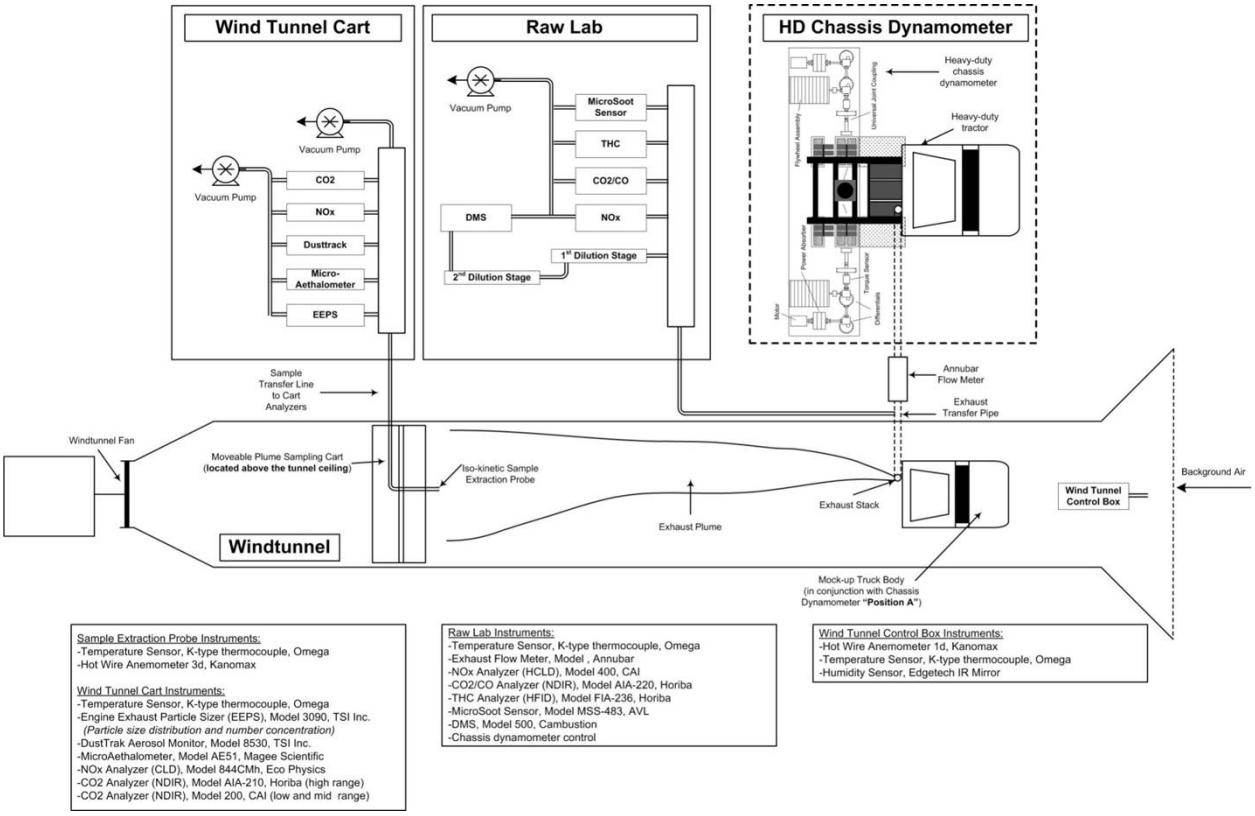


Figure 64: Lay out of the wind tunnel experimental setup

The WVU wind tunnel is capable to capture the first seconds of plume formation and evolution, from a single vehicle with parameterized input, thus give the capability of discern the weight on real world emission of different aftertreatment technologies. To achieve a detailed plume investigation 130 sampling points divided into 10 planes were used to capture PM characteristics. Each plane presents the same sampling pattern, with different vertical dimension and position, to better focusing on the plume shape and dimension evolution. The pattern is composed of 2 nested hexagons, rotated by 90, and an additional center point (see Figure 65).

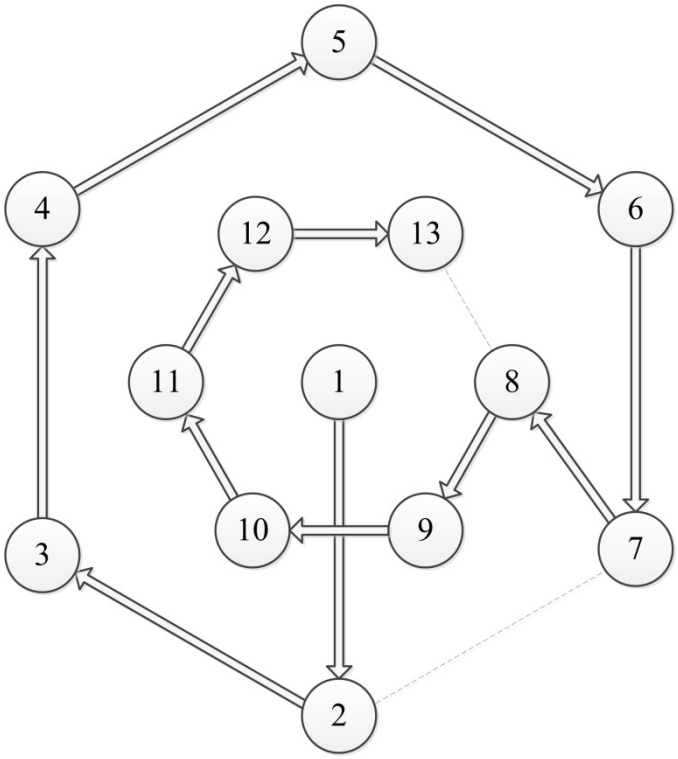


Figure 65: Sampling pattern

The experimental data were analyzed using a modified version of the MARS code presented in this work. The input variables in this case were the three Cartesian coordinates representing a point inside the wind tunnel. Several outputs were modeled, ranging from temperature to particle size distribution. A total of 48 different outputs were considered, 32 of them being the channels of the EEPS. Figure 66 shows the R^2 value of each variable.

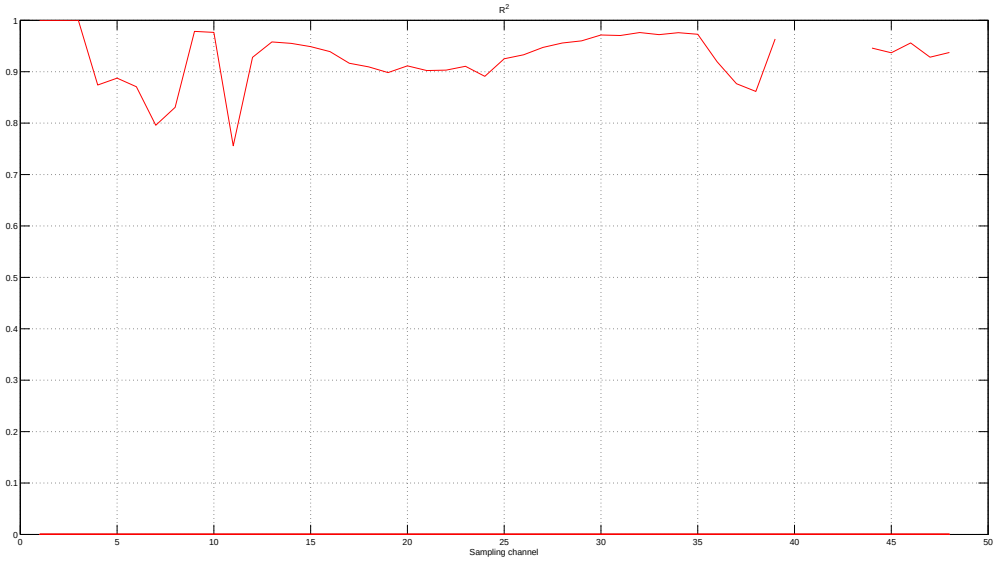


Figure 66: Residual sum of square for each samling channel

Figure 67 shows a comparison between two contour plots representing NOx values at 5 cm from the exhaust stack. The one on the left was obtained using MARS, the one on the right is obtained from the original data.

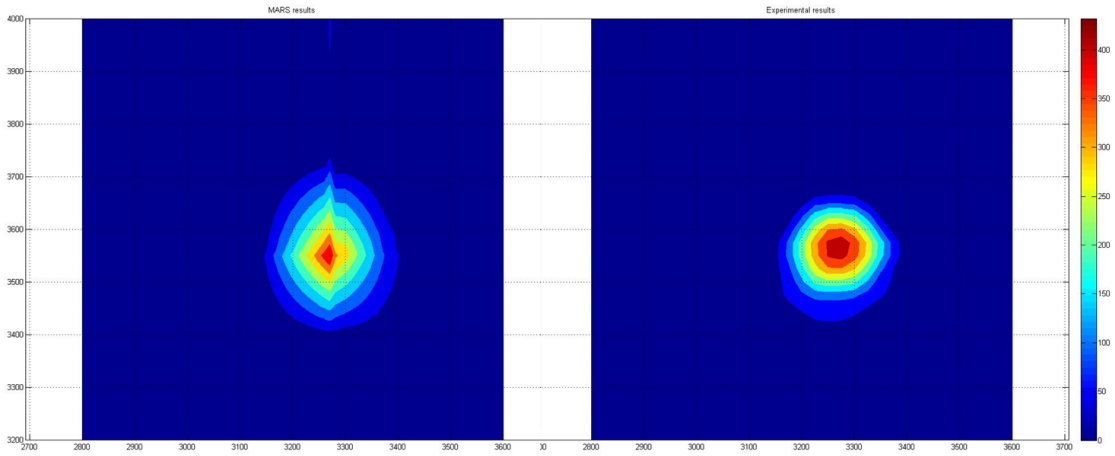


Figure 67: Comparison between MARS and experimental values

The goal of this analysis is to produce curves representing isovalues of nucleation PM inside this tunnel. The same typology of curves will be produced for those characteristics that are suspected to influence PM nucleation, such as temperature or turbulence intensity. In the opinion of the author this way to display the problem will help in determining the physical phenomena leading to nucleation.

Conclusions

The study presented in this work consisted in an analysis tool to support engine testing. Based on the study of regression modeling techniques presented in the regression analysis chapter, the MARS algorithm was reputed a viable basis for emission analysis. The features that led to the selection of MARS as the base algorithm can be listed as:

- Accuracy: When dealing with non-linearity in the training data, MARS is more accurate than polynomial regression. A comparison between these two methodologies is presented in the experiment reduction section.
- Transparency: The contribution of each input factor, and the interaction between them can be evaluated. The ANOVA tables are a synthetic way to display it.
- Robustness: The MARS model has been tested over different problem types and sampling sizes, proving a fair accuracy.
- Efficiency: The MARS algorithm does not require high computation effort to generate a model.

The main drawbacks of MARS compared to other regression algorithms are the conceptual complexity, and the constrain in the sample size. Compared to other regression strategies MARS require more time to be implemented, a version of the code produced is presented in Appendix C. The user of this analysis tool will not have to change the algorithm but just to recall the functions in Appendix C in a Matlab environment. The sample used to train the model must include at least 30 data points in order for MARS to operate correctly.

The analysis tool was developed and tested on data collected during the project AVFL-16 [3]. These experimental data describe the response of an engine to changes in the injection strategy and

fuel properties. To investigate the effect of fuel properties on advanced combustion a systematic multivariate analysis based on the MARS algorithm was carried on. The varying fuel characteristics were CN, aromatic content, and T90 coupled with a split injection control strategy on a GM Z19DTH light-duty compression-ignition engine.

A single engine operating condition consisting of a fixed engine speed of 2100 rpm and 3.5 bar BMEP was utilized. The split injection control strategy involved varying the start of the pilot injection, start of the main injection, and fuel split. A repeatability study was performed to develop a standard by which emissions and performance changes among the fuels could be attributed to fuel property differences and not to the variability associated with the equipment or control strategy. The analysis performed lead to the following conclusions regarding low temperature combustion:

- A quadratic correlation exists between NOx and main SOI. As expected, NOX decreased as the main SOI was retarded.
- Increasing the Cetane number per se leads to higher NOx emissions, despite the effect can be masked as a consequence of Main SOI.
- An increase in CN also leads to higher soot concentrations in the exhaust.
- Higher values of CN are beneficial in reducing CO and HC.
- Regarding CO emissions, CN has a stronger impact than injection strategies.
- Aromatic content and distillation temperature lead to less soot emissions. Their effect is less noticeable compared to CN and injection strategies.
- AC has a strong impact on hydrocarbons production.
- Efficiency is increased by advancing the injection timing, and higher CN.

Using the equation obtained through the regression analysis a simple optimization procedure was performed. Based on this study the fuel characteristics that best suit low temperature combustion are; very low CN, low aromatic content, and low distillation temperature. This combination of characteristics is such to inhibit the combustion to happen too fast, giving the mixture enough time to homogenize completely.

Regarding the methodology itself we can draw the following conclusions. Since the MARS model is not a physical model it strongly relies on the data used to educate it. The regression equations quality depends strongly on the accuracy of the data used to generate them. Overall, the technique applied to develop the illustrative conceptual model was useful in screening the data, in determining the optimal solution, and the minimum number of tests necessary to characterize the phenomenon. The use of MARS to generate regression equations can lead to much wider use of RSM in the field of engine testing, both for analysis purposes and procedure development. Below are some recommendations for additional testing and further analysis of the existing data.

- It would be beneficial to include engine modifications in a further study. Certain fuels lend themselves to different engine conditions. Engine conditions and hardware of special interest include intake temperature, intake pressure, intake oxygen, and compression ratio, EGR flow, and EGR temperature.

- Investigation of additional engine operating conditions would assist in better understanding the effects of fuel properties on engine emissions and performance while operating in advanced combustion regimes.

APPENDIX A

$$NO_x = 0.27472 * heaviside(0.9650 - 1.0 * x1) * (x1 - 0.9650) - 0.7595 * heaviside(x3 - 0.9333) * (x3 - 0.9333) - 2.1436 * heaviside(x1 - 0.9650) * (x1 - 0.9650) + 2.5809 * heaviside(0.42857 - 1.0 * x4) * (x4 - 0.4285) - 2.4005 * heaviside(0.85714 - 1.0 * x4) * (x4 - 0.8571) + 2.9457 * heaviside(x4 - 0.4285) * (x4 - 0.4285) + 0.78919 * heaviside(x5 - 0.5) * heaviside(0.85714 - 1.0 * x4) * (x4 - 0.8571) * (x5 - 0.5) + 0.1814 * heaviside(x5 - 0.25) * heaviside(x6 - 0.5) * (x5 - 0.25) * (x6 - 0.5) - 11.9976 * heaviside(x3 - 0.9333) * heaviside(x4 - 0.4285) * (x3 - 0.9333) * (x4 - 0.4285) - 0.24074 * heaviside(0.42857 - 1.0 * x4) * heaviside(0.98667 - 1.0 * x3) * (x3 - 0.98667) * (x4 - 0.4285) - 0.1974 * heaviside(1.0 - 1.0 * x6) * heaviside(0.85714 - 1.0 * x4) * (x4 - 0.8571) * (x6 - 1.0) - 23.5635 * heaviside(x4 - 0.8571) * heaviside(0.057842 - 1.0 * x1) * (x1 - 0.0578) * (x4 - 0.8571) - 1.0005 * heaviside(0.75 - 1.0 * x5) * heaviside(x4 - 0.4285) * (x4 - 0.4285) * (x5 - 0.75) + 4.2008 * heaviside(0.75 - 1.0 * x5) * heaviside(x4 - 0.8571) * (x4 - 0.8571) * (x5 - 0.75) - 8.0998 * heaviside(x4 - 0.8571) * heaviside(x5 - 0.75) * (x4 - 0.85713) * (x5 - 0.75) - 0.2378 * heaviside(0.5 - 1.0 * x5) * heaviside(1.0 - 1.0 * x6) * heaviside(0.85714 - 1.0 * x4) * (x4 - 0.8571) * (x5 - 0.5) * (x6 - 1.0) - 2.548 * heaviside(x2 - 0.8685) * heaviside(x4 - 0.4285) * heaviside(0.93333 - 1.0 * x3) * (x3 - 0.9333) * (x2 - 0.8685) * (x4 - 0.4285) - 3.8785 * heaviside(x4 - 0.8571) * heaviside(0.93333 - 1.0 * x3) * heaviside(x5 - 0.75) * (x3 - 0.93332) * (x4 - 0.8571) * (x5 - 0.75) + 0.87591 * heaviside(x5 - 0.5) * heaviside(1.0 - 1.0 * x6) * heaviside(0.85714 - 1.0 * x4) * (x4 - 0.8571) * (x5 - 0.5) * (x6 - 1.0) - 13.0159 * heaviside(x6 - 0.5) * heaviside(0.25 - 1.0 * x5) * heaviside(x4 - 0.4285) * (x4 - 0.4285) * (x5 - 0.25) * (x6 - 0.5) - 175.6666 * heaviside(x1 - 0.96505) * heaviside(x3 - 0.9333) * heaviside(x4 - 0.4285) * (x1 - 0.96505) * (x3 - 0.9333) * (x4 - 0.4285) - 154.7285 * heaviside(0.5 - 1.0 * x5) * heaviside(x1 - 0.05784) * heaviside(x4 - 0.8571) * (x1 - 0.05784) * (x4 - 0.8571) * (x5 - 0.5) + 6.9722 * heaviside(x6 - 0.5) * heaviside(0.25 - 1.0 * x5) * heaviside(x4 -$$

$0.4285) * heaviside(x2 - 0.0380) * (x4 - 0.42856) * (x2 - 0.03806) * (x5 - 0.25) * (x6 - 0.5) - 0.6137$

(29)

$$PM = 0.2306 * heaviside(0.96506 - 1.0 * x1) * (x1 - 0.9650) - 8.563 * heaviside(x1 - 0.9650) * (x1 - 0.9650) - 0.2494 * heaviside(0.28571 - 1.0 * x4) * (x4 - 0.2857) - 0.4434 * heaviside(0.25 - 1.0 * x5) * (x5 - 0.25) - 0.1739 * heaviside(x4 - 0.2857) * (x4 - 0.2857) - 0.1560 * heaviside(x5 - 0.25) * (x5 - 0.25) + 4.5442 * heaviside(x3 - 0.9333) * heaviside(x4 - 0.2857) * (x3 - 0.9333) * (x4 - 0.2857) - 0.5418 * heaviside(0.77509 - 1.0 * x2) * heaviside(x1 - 0.6156) * (x2 - 0.7750) * (x1 - 0.6156) + 9.4347 * heaviside(x2 - 0.7750) * heaviside(x1 - 0.6156) * (x2 - 0.7750) * (x1 - 0.6156) + 6.2743 * heaviside(x5 - 0.25) * heaviside(0.14286 - 1.0 * x4) * (x4 - 0.1428) * (x5 - 0.25) - 16.3151 * heaviside(x5 - 0.5) * heaviside(0.28571 - 1.0 * x4) * (x4 - 0.2857) * (x5 - 0.5) + 0.20789 * heaviside(x5 - 0.25) * heaviside(x4 - 0.1428) * (x4 - 0.1428) * (x5 - 0.25) + 4.4402 * heaviside(0.25 - 1.0 * x5) * heaviside(0.038062 - 1.0 * x2) * (x2 - 0.038) * (x5 - 0.25) + 2.3423 * heaviside(0.5 - 1.0 * x5) * heaviside(0.28571 - 1.0 * x4) * (x4 - 0.2857) * (x5 - 0.5) - 0.4680 * heaviside(0.5 - 1.0 * x6) * heaviside(0.28571 - 1.0 * x4) * (x4 - 0.2857) * (x6 - 0.5) + 66.319 * heaviside(0.25 - 1.0 * x5) * heaviside(0.57143 - 1.0 * x4) * heaviside(0.12 - 1.0 * x3) * (x3 - 0.1199) * (x4 - 0.5714) * (x5 - 0.25) - 22.6751 * heaviside(0.25 - 1.0 * x5) * heaviside(0.5 - 1.0 * x6) * heaviside(0.038062 - 1.0 * x2) * (x2 - 0.0380) * (x5 - 0.25) * (x6 - 0.5) - 19.8129 * heaviside(0.75 - 1.0 * x5) * heaviside(0.42857 - 1.0 * x4) * heaviside(x1 - 0.6156) * (x4 - 0.4285) * (x5 - 0.75) * (x1 - 0.6156) + 117.61 * heaviside(0.038062 - 1.0 * x2) * heaviside(x4 - 0.4285) * heaviside(x1 - 0.6156) * (x4 - 0.4285) * (x1 - 0.6156) * (x2 - 0.0380) - 197.8254 * heaviside(x5 - 0.5) * heaviside(0.28571 - 1.0 * x4) * heaviside(0.61568 - 1.0 * x1) * (x1 - 0.6156) * (x4 - 0.2857) * (x5 - 0.5) + 605.14 * heaviside(0.42857 - 1.0 * x4) * heaviside(x5 - 0.75) * heaviside(x1 - 0.61567) * (x4 - 0.4285) * (x5 - 0.75) * (x1 - 0.6156) - 9.8699 * heaviside(x4 - 0.4285) * heaviside(x1 - 0.6156) * heaviside(x2 - 0.0380) * (x4 - 0.4285) * (x1 - 0.6156) * (x2 - 0.0380) - 2.7928 * heaviside(x6 - 0.5) * heaviside(x4 - 0.4285) * heaviside(x1 - 0.6156) * (x4 - 0.4285) * (x1 - 0.6156) * (x6 - 0.5) + 0.95225 * heaviside(x5 - 0.25) * heaviside(x6 - 0.5) * heaviside(x1 - 0.0578) * (x1 - 0.0578) * (x5 - 0.25) * (x6 - 0.5) + 8.3351 * heaviside(x5 - 0.5) * heaviside(x6 - 0.5) * heaviside(0.28571 - 1.0 * x4) * (x4 -$$

$0.2857) * (x5 - 0.5) * (x6 - 0.5) + 1.5384 * heaviside(0.25 - 1.0 * x5) * heaviside(0.57143 - 1.0 * x4) * heaviside(x3 - 0.1199) * (x3 - 0.1199) * (x4 - 0.5714) * (x5 - 0.25) - 29.0101 * heaviside(0.72 - 1.0 * x3) * heaviside(0.75 - 1.0 * x5) * heaviside(0.42857 - 1.0 * x4) * heaviside(x1 - 0.61567) * (x4 - 0.4285) * (x5 - 0.75) * (x3 - 0.7199) * (x1 - 0.6156) + 65.23 * heaviside(0.75 - 1.0 * x5) * heaviside(0.42857 - 1.0 * x4) * heaviside(x3 - 0.7199) * heaviside(x1 - 0.6156) * (x4 - 0.42856) * (x5 - 0.75) * (x3 - 0.7199) * (x1 - 0.6156) + 188.45 * heaviside(x4 - 0.4285) * heaviside(x5 - 0.75) * heaviside(x1 - 0.61567) * heaviside(x2 - 0.0380) * (x4 - 0.4285) * (x5 - 0.75) * (x1 - 0.6156) * (x2 - 0.0380) + 0.31094$
(30)

$$CO = 0.5714 * heaviside(0.75 - 1.0 * x5) * (x5 - 0.75) - 0.46230 * heaviside(0.96506 - 1.0 * x1) * (x1 - 0.9650) - 0.2902 * heaviside(0.57143 - 1.0 * x4) * (x4 - 0.5714) - 4.4475 * heaviside(x1 - 0.9650) * (x1 - 0.9650) - 0.3894 * heaviside(x5 - 0.75) * (x5 - 0.75) - 0.1014 * heaviside(x6 - 0.5) * (x6 - 0.5) - 0.0528 * heaviside(0.96506 - 1.0 * x1) * heaviside(x3 - 0.0399) * (x1 - 0.9650) * (x3 - 0.0399) - 2.0822 * x1 * heaviside(x5 - 0.75) * heaviside(x1) * (x5 - 0.75) - 1.8265 * heaviside(0.5 - 1.0 * x5) * heaviside(x4 - 0.5714) * (x4 - 0.5714) * (x5 - 0.5) + 1.0074 * heaviside(0.5 - 1.0 * x6) * heaviside(x4 - 0.5714) * (x4 - 0.5714) * (x6 - 0.5) + 0.78575 * heaviside(x6 - 0.5) * heaviside(0.25 - 1.0 * x5) * (x5 - 0.25) * (x6 - 0.5) - 0.5787 * heaviside(x6 - 0.5) * heaviside(0.5 - 1.0 * x5) * (x5 - 0.5) * (x6 - 0.5) + 2.0172 * heaviside(x6 - 0.5) * heaviside(x5 - 0.75) * (x5 - 0.75) * (x6 - 0.5) + 1.2819 * heaviside(x5 - 0.5) * heaviside(x4 - 0.5714) * (x4 - 0.5714) * (x5 - 0.5) - 0.9081 * heaviside(x6 - 0.5) * heaviside(x4 - 0.5714) * (x4 - 0.5714) * (x6 - 0.5) - 2.5193 * heaviside(x5 - 0.5) * heaviside(x6 - 0.5) * (x5 - 0.5) * (x6 - 0.5) - 1994.7080 * heaviside(0.038062 - 1.0 * x2) * heaviside(x1 - 0.9650) * (x1 - 0.9650) * (x2 - 0.0380) + 0.28194 * heaviside(0.57143 - 1.0 * x4) * heaviside(x3 - 0.1199) * (x3 - 0.1199) * (x4 - 0.5714) + 1.6451 * heaviside(0.04 - 1.0 * x3) * heaviside(0.96506 - 1.0 * x1) * (x1 - 0.9650) * (x3 - 0.0399) + 0.5423 * heaviside(0.85714 - 1.0 * x4) * heaviside(0.96506 - 1.0 * x1) * (x1 - 0.9650) * (x4 - 0.8571) - 1.7001 * heaviside(0.57143 - 1.0 * x4) * heaviside(0.12 - 1.0 * x3) * (x3 - 0.1199) * (x4 - 0.5714) - 0.4711 * heaviside(0.5 - 1.0 * x6) * heaviside(0.64092 - 1.0 * x1) * (x1 - 0.6409) * (x6 - 0.5) + 5.5008 * heaviside(0.75 - 1.0 * x5) * heaviside(0.026667 - 1.0 * x3) * (x3 - 0.0266) * (x5 - 0.75) + 0.89697 *$$

$$heaviside(x4 - 0.8571) * heaviside(0.96506 - 1.0 * x1) * (x1 - 0.9650) * (x4 - 0.8571) - 2.25205 * heaviside(0.75-1.0*x5)*heaviside(x4-0.8571)*(x4-0.8571)*(x5-0.75)-0.0680*heaviside(0.75-1.0*x5)*heaviside(x3-0.0266)*(x3-0.0266)*(x5-0.75)-3.8196*heaviside(0.85714-1.0*x4)*heaviside(x5-0.75)*(x4-0.8571)*(x5-0.75)-3.8293*heaviside(x4-0.8571)*heaviside(x5-0.75)*(x4-0.8571)*(x5-0.75)-31.9873*heaviside(x1-0.9650)*heaviside(x2-0.0380)*(x1-0.9650)*(x2-0.0380)-1.5163*heaviside(0.61568-1.0*x1)*heaviside(0.75-1.0*x5)*heaviside(0.85714-1.0*x4)*(x4-0.85713)*(x5-0.75)*(x1-0.6156)-54.16799*heaviside(0.04-1.0*x3)*heaviside(x4-0.8571)*heaviside(0.96506-1.0*x1)*(x1-0.96505)*(x4-0.8571)*(x3-0.0399)+10.595*heaviside(0.5-1.0*x6)*heaviside(0.75-1.0*x5)*heaviside(x4-0.85713)*(x4-0.8571)*(x5-0.75)*(x6-0.5)-8.1382*heaviside(0.5-1.0*x6)*heaviside(0.85714-1.0*x4)*heaviside(x5-0.75)*(x4-0.85713)*(x5-0.75)*(x6-0.5)-8.0255*heaviside(0.5-1.0*x6)*heaviside(x4-0.85713)*heaviside(x5-0.75)*(x4-0.8571)*(x5-0.75)*(x6-0.5)-0.1639*heaviside(0.85714-1.0*x4)*heaviside(0.96506-1.0*x1)*heaviside(x2-0.1730)*(x1-0.9650)*(x4-0.8571)*(x2-0.1730)+0.92503*heaviside(x4-0.8571)*heaviside(0.96506-1.0*x1)*heaviside(x3-0.0399)*(x1-0.9650)*(x4-0.8571)*(x3-0.0399)-11.2077*heaviside(x5-0.25)*heaviside(0.5-1.0*x6)*heaviside(x1-0.9650)*(x1-0.9650)*(x5-0.25)*(x6-0.5)+1.0864*heaviside(x5-0.25)*heaviside(0.5-1.0*x6)*heaviside(0.96506-1.0*x1)*(x1-0.9650)*(x5-0.25)*(x6-0.5)+1.3151*heaviside(x5-0.25)*heaviside(x6-0.5)*heaviside(0.42857-1.0*x4)*(x4-0.4285)*(x5-0.25)*(x6-0.5)-5.9924*heaviside(x5-0.5)*heaviside(x6-0.5)*heaviside(0.71429-1.0*x4)*(x4-0.7142)*(x5-0.5)*(x6-0.5)+3.735*heaviside(x5-0.25)*heaviside(x6-0.5)*heaviside(x4-0.4285)*(x4-0.4285)*(x5-0.25)*(x6-0.5)-4.3952*heaviside(x6-0.5)*heaviside(x5-0.75)*heaviside(x4-0.5714)*(x5-0.75)*(x4-0.5714)*(x6-0.5)+0.2122 \quad (31)$$

$$HC = 0.0693 * heaviside(0.8685 - 1.0 * x2) * (x2 - 0.8685) - 5.3652 * heaviside(0.13898 - 1.0 * x1) * (x1 - 0.1389) + 5.9501 * heaviside(0.57143 - 1.0 * x4) * (x4 - 0.5714) - 0.5384 * heaviside(0.25 - 1.0 * x5) * (x5 - 0.25) + 7.4159 * heaviside(0.75 - 1.0 * x5) * (x5 - 0.75) + 0.08863 * heaviside(0.5 - 1.0 *$$

$x6) * (x6 - 0.5) + 1.2671 * heaviside(x2 - 0.8685) * (x2 - 0.8685) - 1.0368 * heaviside(x4 - 0.8571) * (x4 - 0.8571) - 0.6098 * heaviside(x5 - 0.75) * (x5 - 0.75) + 5.6406 * heaviside(x4 - 0.5714) * (x4 - 0.5714) + 0.075051 * heaviside(x1 - 0.138) * (x1 - 0.1389) - 27.1994 * heaviside(0.5 - 1.0 * x5) * heaviside(x4 - 0.5714) * (x4 - 0.5714) * (x5 - 0.5) - 0.0822 * heaviside(0.5 - 1.0 * x6) * heaviside(x1 - 0.1389) * (x1 - 0.1389) * (x6 - 0.5) - 0.5774 * heaviside(x5 - 0.25) * heaviside(0.13898 - 1.0 * x1) * (x1 - 0.13898) * (x5 - 0.25) - 0.3734 * heaviside(x5 - 0.25) * heaviside(0.85714 - 1.0 * x4) * (x4 - 0.8571) * (x5 - 0.25) - 19.8611 * heaviside(x5 - 0.5) * heaviside(0.5714 - 1.0 * x4) * (x4 - 0.5714) * (x5 - 0.5) + 24.629 * heaviside(x5 - 0.75) * heaviside(x4 - 0.5714) * (x5 - 0.75) * (x4 - 0.5714) - 21.6816 * heaviside(x5 - 0.5) * heaviside(x4 - 0.5714) * (x4 - 0.5714) * (x5 - 0.5) + 0.37334 * heaviside(x6 - 0.5) * heaviside(x4 - 0.5714) * (x4 - 0.5714) * (x6 - 0.5) - 0.2914 * heaviside(x5 - 0.25) * heaviside(x1 - 0.1389) * (x1 - 0.1389) * (x5 - 0.25) + 1.13 * heaviside(0.75 - 1.0 * x5) * heaviside(x3 - 0.9333) * (x3 - 0.9333) * (x5 - 0.75) + 0.1953 * heaviside(0.8571 - 1.0 * x4) * heaviside(0.8685 - 1.0 * x2) * (x2 - 0.8685) * (x4 - 0.8571) - 1.2463 * heaviside(0.5 - 1.0 * x6) * heaviside(0.13898 - 1.0 * x1) * (x1 - 0.1389) * (x6 - 0.5) - 0.0869 * heaviside(0.75 - 1.0 * x5) * heaviside(0.9333 - 1.0 * x3) * (x3 - 0.9333) * (x5 - 0.75) - 24.7899 * heaviside(0.5 - 1.0 * x5) * heaviside(0.57143 - 1.0 * x4) * (x4 - 0.5714) * (x5 - 0.5) + 24.499 * heaviside(0.75 - 1.0 * x5) * heaviside(0.85714 - 1.0 * x4) * (x4 - 0.8571) * (x5 - 0.75) + 5.8 * heaviside(0.13898 - 1.0 * x1) * heaviside(x4 - 0.4285) * (x4 - 0.4285) * (x1 - 0.1389) + 17.358 * heaviside(0.75 - 1.0 * x5) * heaviside(x4 - 0.8571) * (x4 - 0.8571) * (x5 - 0.75) + 112.32 * heaviside(0.85714 - 1.0 * x4) * heaviside(0.86851 - 1.0 * x2) * heaviside(0.057842 - 1.0 * x1) * (x1 - 0.0578) * (x2 - 0.8685) * (x4 - 0.8571) - 0.9164 * heaviside(x5 - 0.5) * heaviside(x6 - 0.5) * heaviside(x4 - 0.5714) * (x4 - 0.5714) * (x5 - 0.5) * (x6 - 0.5) + 0.2957 * heaviside(x5 - 0.25) * heaviside(x6 - 0.5) * heaviside(x1 - 0.1389) * (x1 - 0.1389) * (x5 - 0.25) * (x6 - 0.5) + 2.3854 * heaviside(x5 - 0.25) * heaviside(0.5 - 1.0 * x6) * heaviside(0.1389 - 1.0 * x1) * (x1 - 0.1389) * (x5 - 0.25) * (x6 - 0.5) + 0.3531 * heaviside(x5 - 0.25) * heaviside(0.8571 - 1.0 * x4) * heaviside(0.933 - 1.0 * x3) * (x3 - 0.933) * (x4 - 0.8571) * (x5 - 0.25) - 1.3942 * heaviside(x5 - 0.5) * heaviside(0.5714 - 1.0 * x4) * heaviside(0.72 - 1.0 * x3) * (x3 - 0.7199) * (x4 - 0.5714) * (x5 - 0.5) + 0.8340 * heaviside(x5 - 0.25) * heaviside(0.2857 - 1.0 * x4) * heaviside(x1 - 0.1389) * (x4 - 0.2857) * (x1 - 0.1389) * (x5 - 0.25) + 0.1671 * heaviside(x5 - 0.25) * heaviside(0.5 - 1.0 * x6) * heaviside(x1 - 0.1389) * (x1 - 0.1389) * (x5 - 0.25) * (x6 - 0.5) - 0.2448 * heaviside(x1 - 0.0578) * heaviside(0.8571 - 1.0 * x4) * heaviside(0.8685 - 1.0 * x2) * (x1 - 0.0578) * (x2 - 0.8685) * (x4 - 0.8571) + 1440.3 * heaviside(0.5 -$

$1.0 * x5) * heaviside(0.8571 - 1.0 * x4) * heaviside(0.8685 - 1.0 * x2) * heaviside(0.0578 - 1.0 * x1) * (x1 - 0.0578) * (x2 - 0.8685) * (x4 - 0.8571) * (x5 - 0.5) + 60.853 * heaviside(0.5 - 1.0 * x6) * heaviside(0.85714 - 1.0 * x4) * heaviside(0.8685 - 1.0 * x2) * heaviside(0.0578 - 1.0 * x1) * (x1 - 0.0578) * (x2 - 0.8685) * (x4 - 0.8571) * (x6 - 0.5) - 190.0494 * heaviside(x5 - 0.5) * heaviside(0.85714 - 1.0 * x4) * heaviside(0.8685 - 1.0 * x2) * heaviside(0.0578 - 1.0 * x1) * (x1 - 0.0578) * (x2 - 0.8685) * (x4 - 0.8571) * (x5 - 0.5) + 28.671 * heaviside(x6 - 0.5) * heaviside(0.8571 - 1.0 * x4) * heaviside(0.8685 - 1.0 * x2) * heaviside(0.0578 - 1.0 * x1) * (x1 - 0.0578) * (x2 - 0.8685) * (x4 - 0.8571) * (x6 - 0.5) + 0.2426$ (32)

$BTE = 0.20364 * heaviside(x1 - 0.0578) * (x1 - 0.0578) + 6.282 * heaviside(0.0578 - 1.0 * x1) * (x1 - 0.0578) + 0.0708 * heaviside(0.7750 - 1.0 * x2) * (x2 - 0.7750) + 1.0573 * heaviside(0.4285 - 1.0 * x4) * (x4 - 0.4285) - 0.5192 * heaviside(0.75 - 1.0 * x5) * (x5 - 0.75) - 0.1017 * heaviside(0.5 - 1.0 * x6) * (x6 - 0.5) - 1.309 * heaviside(x2 - 0.7750) * (x2 - 0.7750) - 0.3092 * heaviside(x6 - 0.5) * (x6 - 0.5) - 0.4131 * heaviside(x6 - 0.5) * heaviside(0.8369 - 1.0 * x1) * (x1 - 0.8369) * (x6 - 0.5) - 0.8066 * heaviside(x6 - 0.5) * heaviside(0.5 - 1.0 * x5) * (x5 - 0.5) * (x6 - 0.5) - 0.4730 * heaviside(x5 - 0.25) * heaviside(x4 - 0.4285) * (x4 - 0.4285) * (x5 - 0.25) + 0.8492 * heaviside(x3 - 0.1199) * heaviside(x4 - 0.4285) * (x3 - 0.1199) * (x4 - 0.4285) + 0.57063 * heaviside(0.7750 - 1.0 * x2) * heaviside(0.4285 - 1.0 * x4) * (x4 - 0.4285) * (x2 - 0.7750) - 0.5143 * heaviside(0.25 - 1.0 * x5) * heaviside(0.7750 - 1.0 * x2) * (x2 - 0.7750) * (x5 - 0.25) - 5.3919 * heaviside(x4 - 0.4285) * heaviside(0.12 - 1.0 * x3) * (x3 - 0.1199) * (x4 - 0.4285) + 4.8437 * heaviside(0.25 - 1.0 * x5) * heaviside(x4 - 0.4285) * (x4 - 0.4285) * (x5 - 0.25) - 5.4912 * heaviside(0.42857 - 1.0 * x4) * heaviside(x2 - 0.7750) * (x4 - 0.4285) * (x2 - 0.7750) - 16.1963 * heaviside(0.4285 - 1.0 * x4) * heaviside(x5 - 0.75) * (x4 - 0.4285) * (x5 - 0.75) + 11.712 * heaviside(x6 - 0.5) * heaviside(0.5 - 1.0 * x5) * heaviside(x4 - 0.5714) * (x4 - 0.5714) * (x5 - 0.5) * (x6 - 0.5) + 0.3896$ (33)

APPENDIX B

NOx

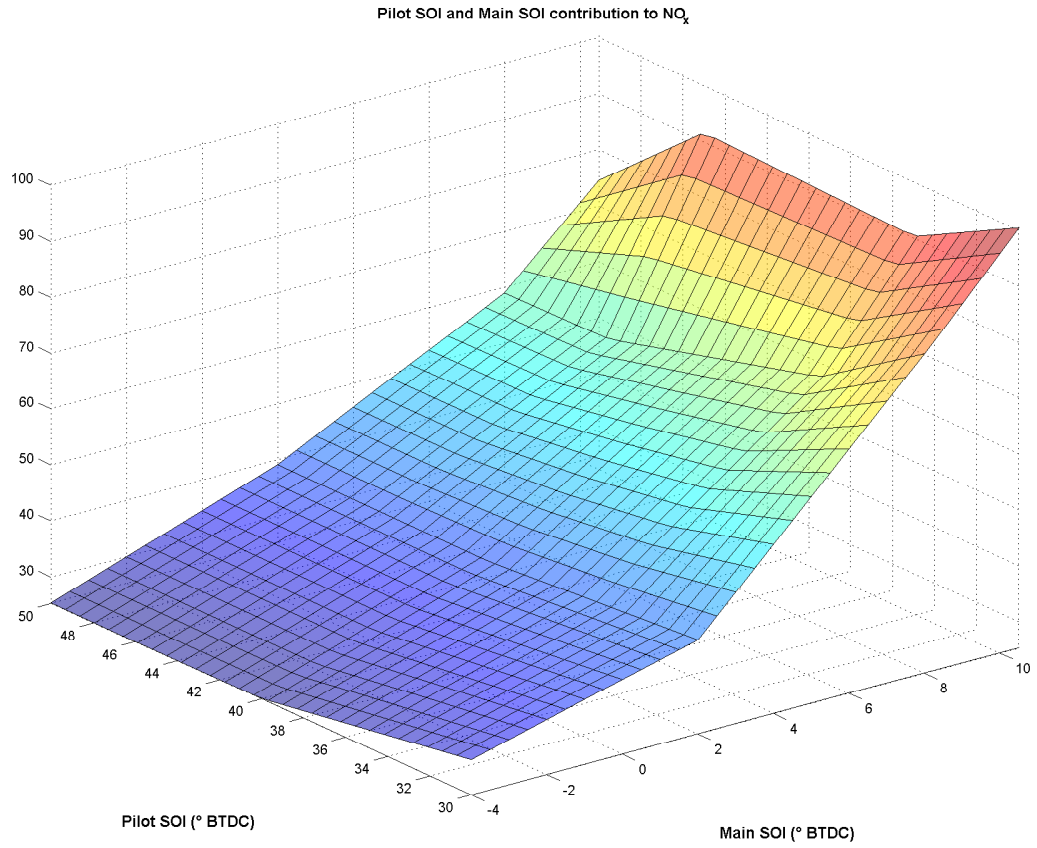


Figure 68: Dependence of NOx from main SOI, and pilot SOI

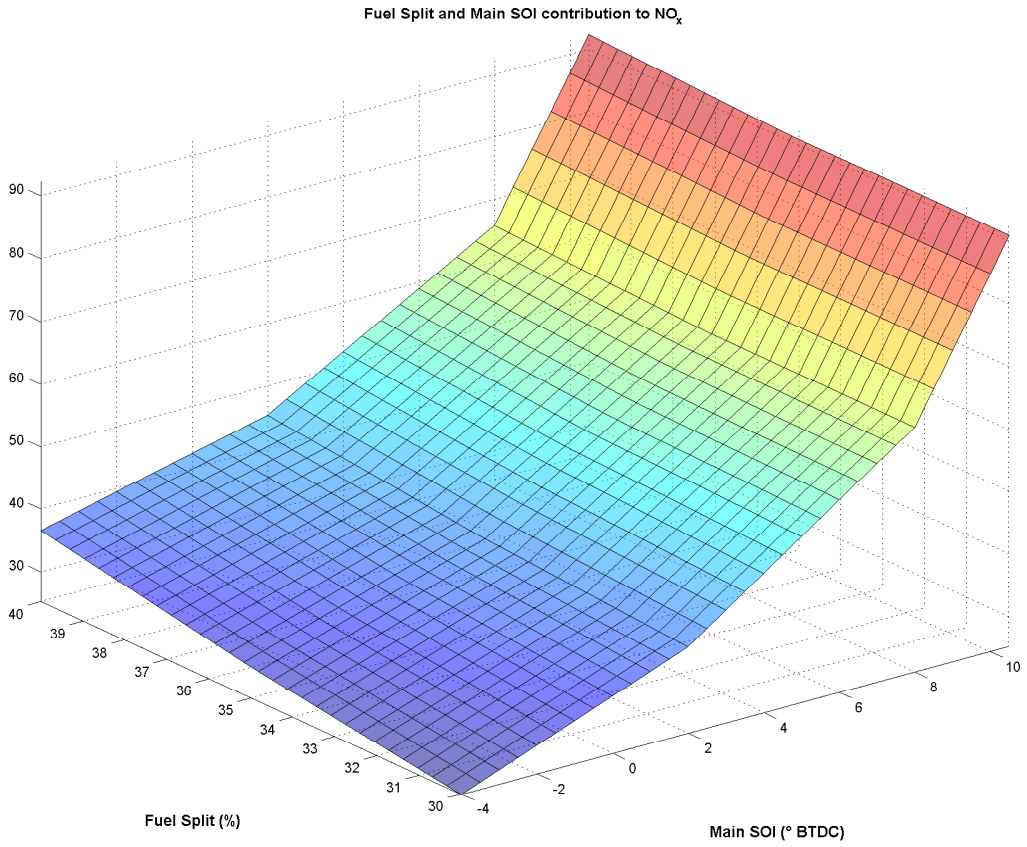


Figure 69: Dependence of NOx from main SOI, and fuel split

Soot Concentration

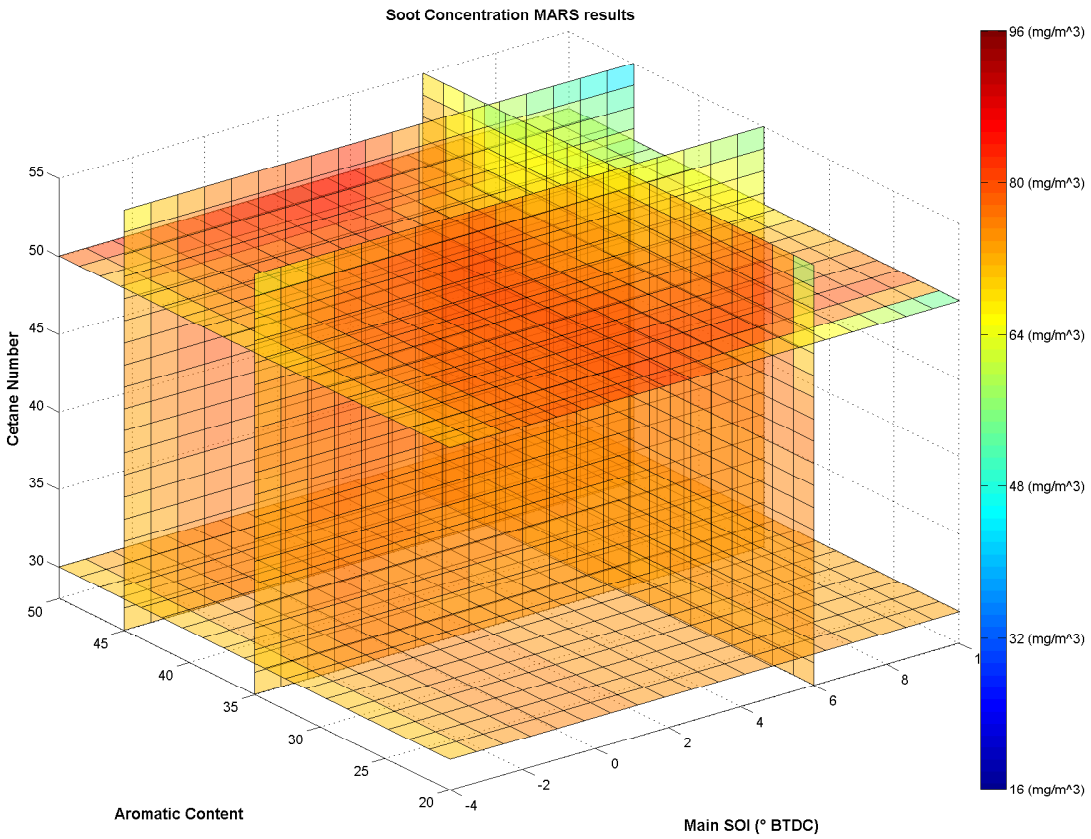


Figure 70: Dependence of PM from cetane number, aromatic content, and main SOI

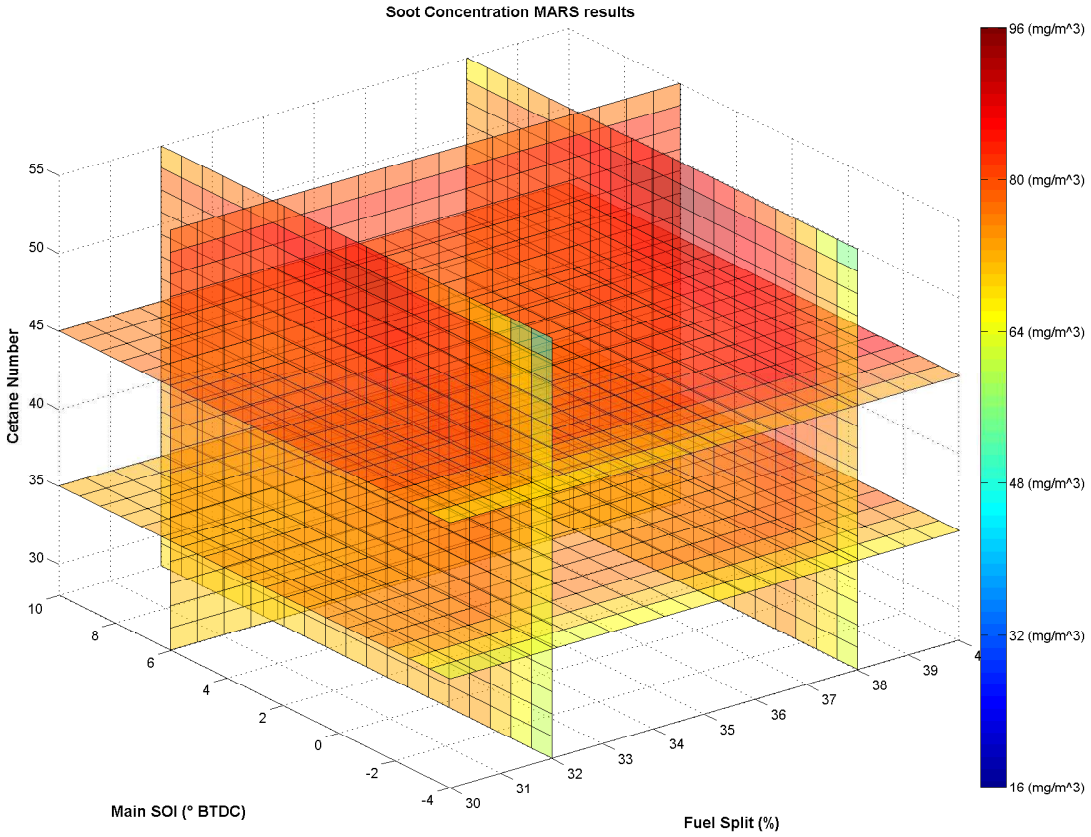


Figure 71: Dependence of PM from cetane number, fuel split, and main SOI

CO

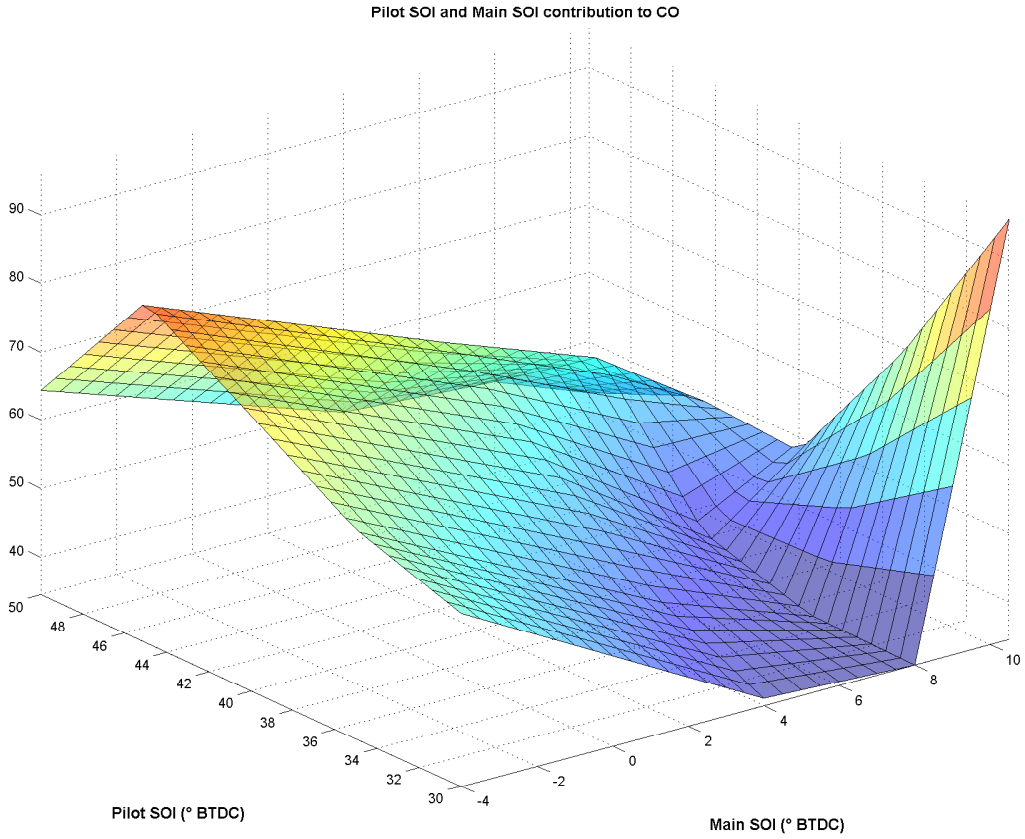


Figure 72: Dependence of CO from pilot SOI, and main SOI

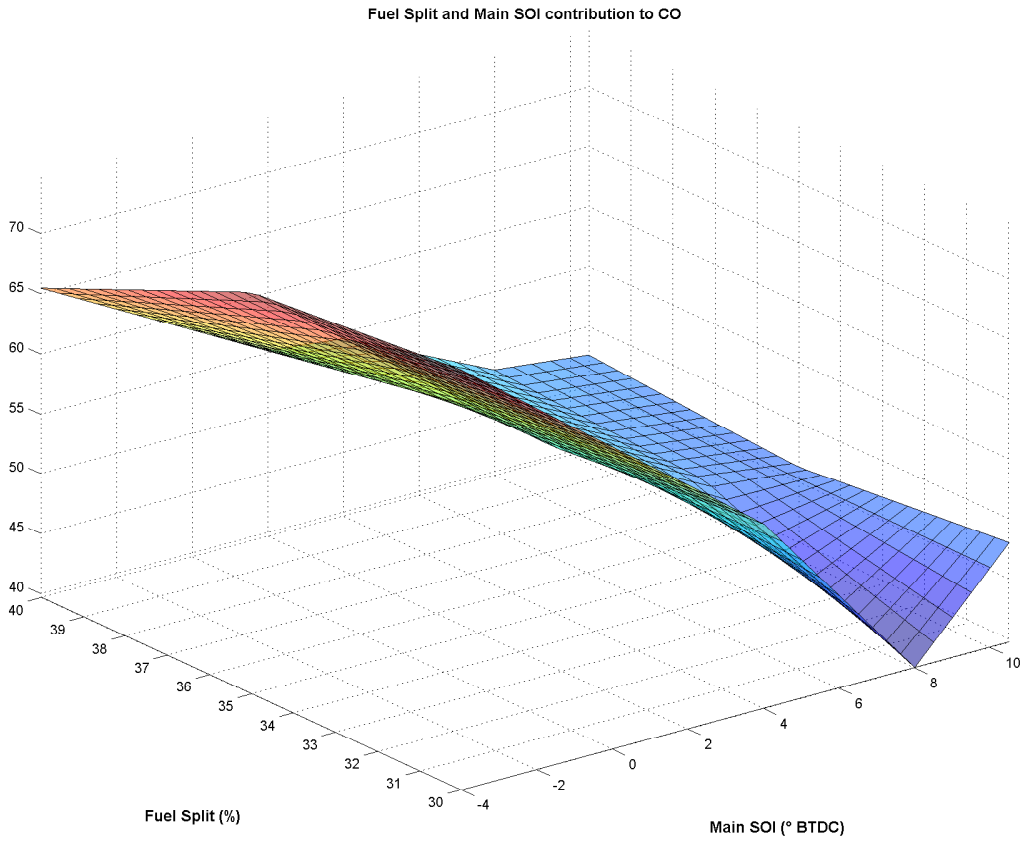


Figure 73: Dependence of CO from fuel split, and main SOI

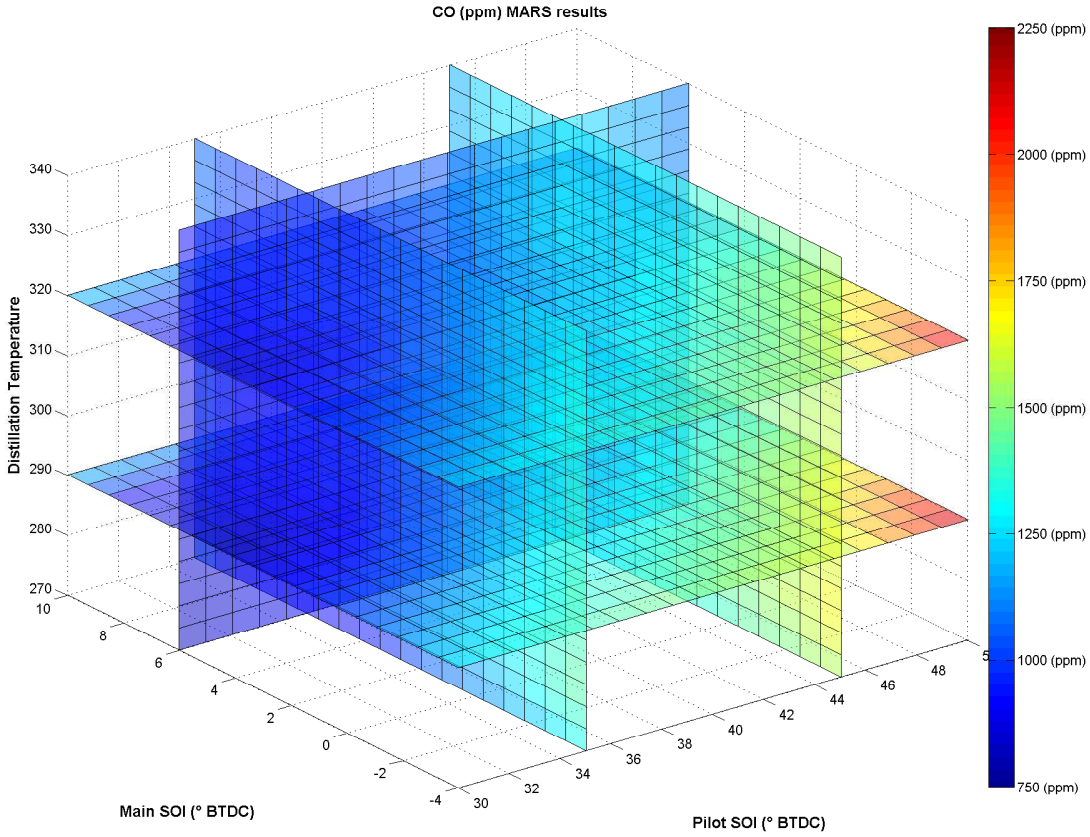


Figure 74: Dependence of CO from distillation temperature, pilot SOI, and main SOI

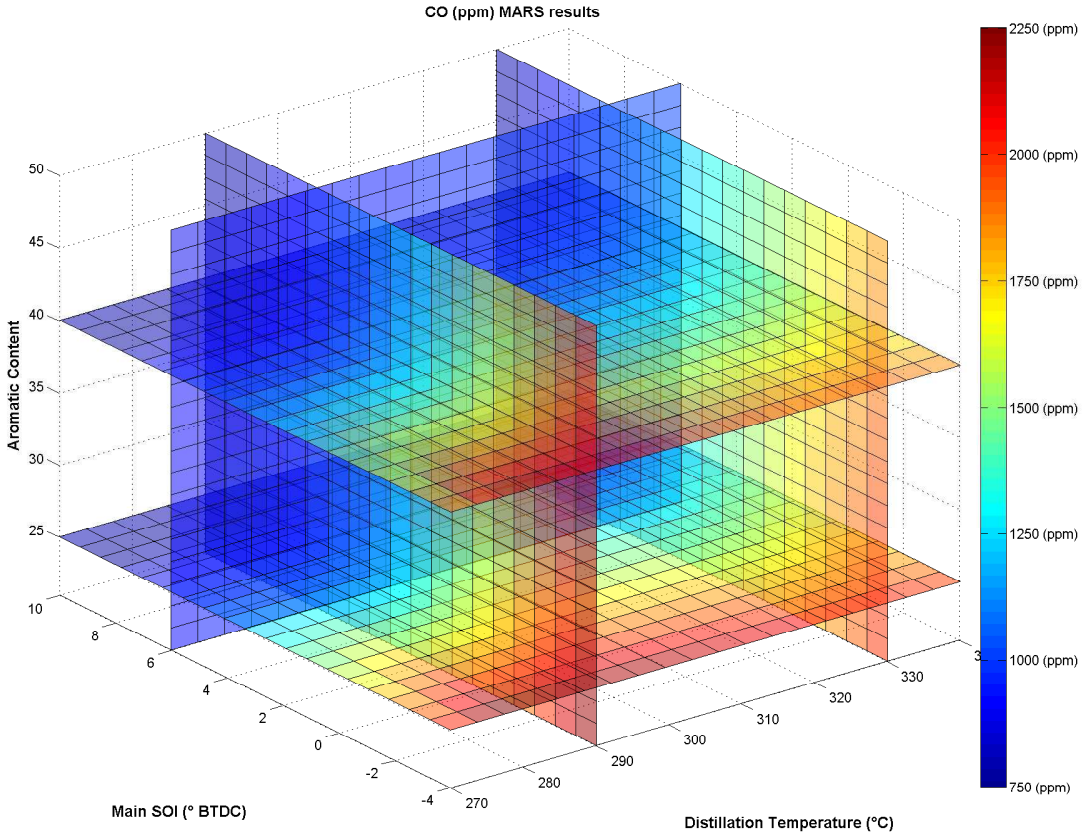


Figure 75: Dependence of CO from aromatic content, distillation temperature, and main SOI

HC

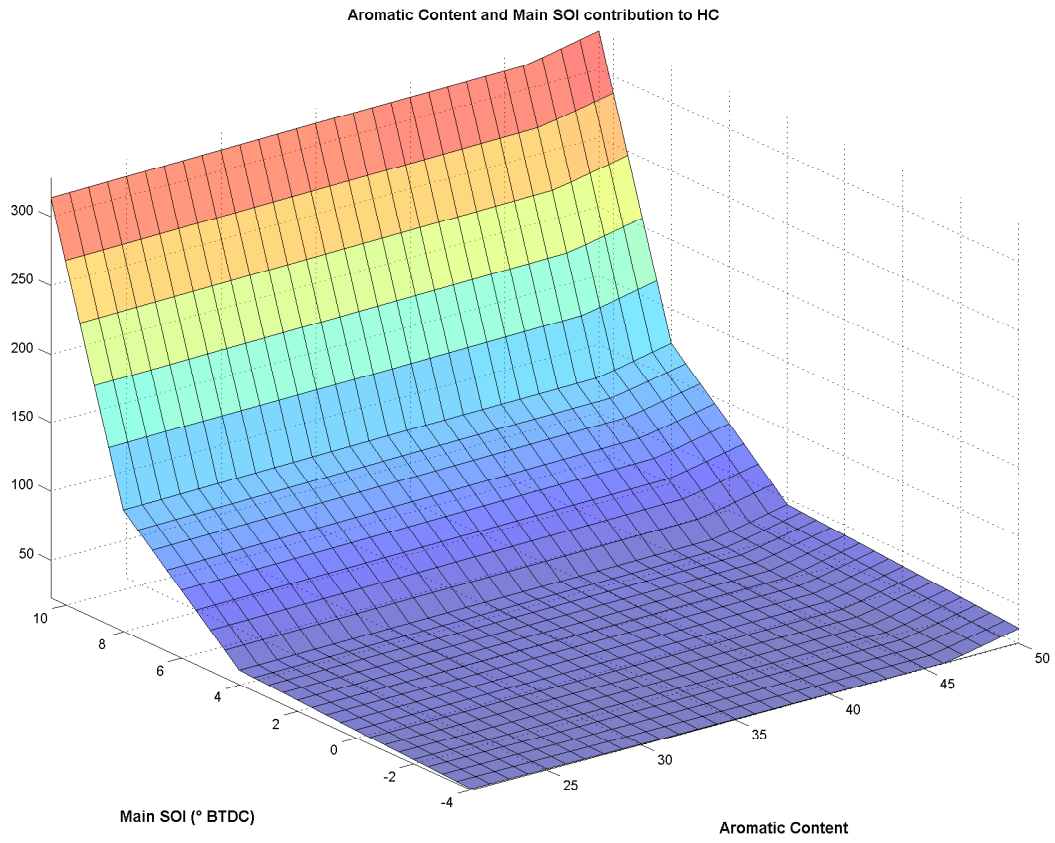


Figure 76: Dependence of HC from aromatic content, and main SOI

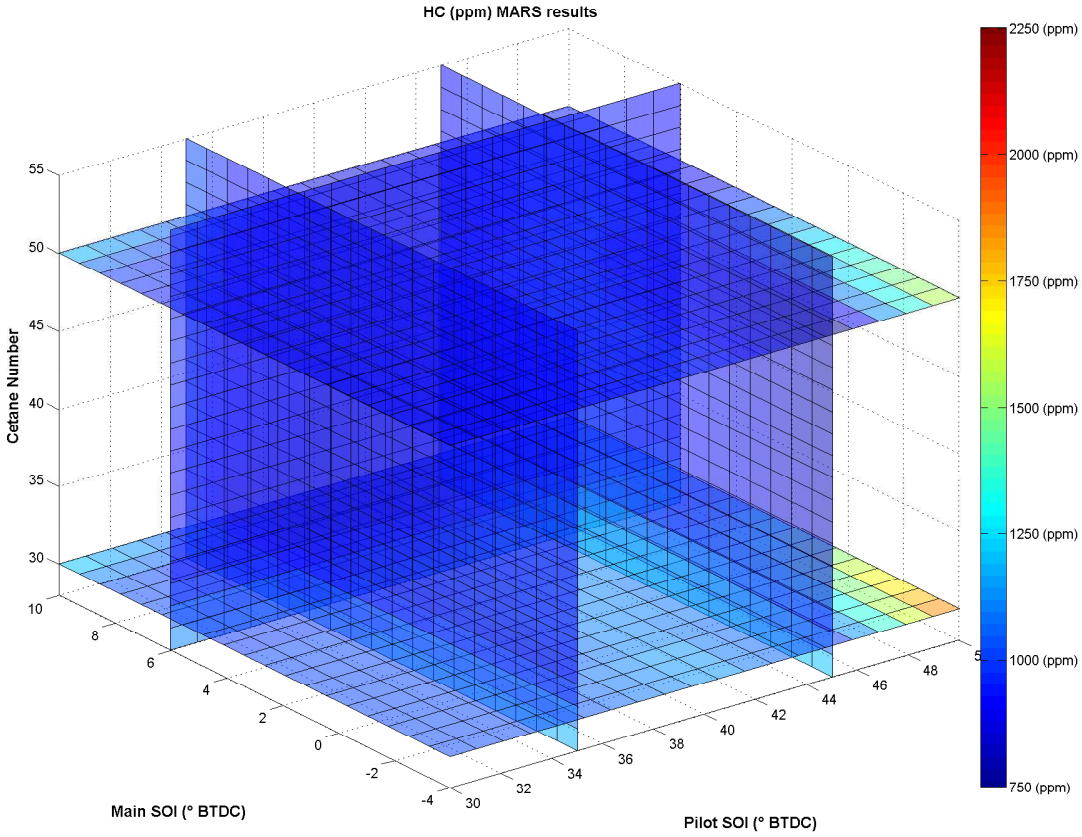


Figure 77: Dependence of HC from cetane number, pilot SOI, and main SOI

APPENDIX C

```
clc
close all
clear all


load FuelComp1.csv    %Studio Caratteristiche combustibile e parametri
    di igniezione
x1=FuelComp1(:,1);   %Cetane Number
x2=FuelComp1(:,2);   %Aromatic Content
x3=FuelComp1(:,3);   %T90
x4=FuelComp1(:,4);   %Main SOI
x5=FuelComp1(:,5);   %Pilot SOI
x6=FuelComp1(:,6);   %Fuel Split
y1=FuelComp1(:,7); %NOx
y2=FuelComp1(:,8); %Soot Concentration
y3=FuelComp1(:,9); % BTE (%)
y4=FuelComp1(:,10); %CO (ppm)
y5=FuelComp1(:,11); %HC (ppm)

%Store the uncoded variable
Xp = [x1 x2 x3 x4 x5 x6];
Yp=y5;                                    %!!!!!!!!!!!!!!!!!!!!
```

```
22
23 %Coded Variables
24 coded=struct('x1l',{min(x1)},'x1h',{max(x1)},'x2l',{min(x2)},'x2h',{max
       (x2)},'x3l',{min(x3)},'x3h',{max(x3)},...
25       'x4l',{min(x4)},'x4h',{max(x4)},'x5l',{min(x5)},'x5h',{max(x5)},'
             x6l',{min(x6)},'x6h',{max(x6)},...
26       'y1l',{min(y1)},'y1h',{max(y1)},'y2l',{min(y2)},'y2h',{max(y2)},'
             y3l',{min(y3)},'y3h',max(y3),'y4l',{min(y4)},...
27       'y4h',{max(y4)},'y5l',{min(y5)},'y5h',{max(y5)});
28
29 x1=(x1-coded.x1l)./(coded.x1h-coded.x1l);
30 x2=(x2-coded.x2l)./(coded.x2h-coded.x2l);
31 x3=(x3-coded.x3l)./(coded.x3h-coded.x3l);
32 x4=(x4-coded.x4l)./(coded.x4h-coded.x4l);
33 x5=(x5-coded.x5l)./(coded.x5h-coded.x5l);
34 x6=(x6-coded.x6l)./(coded.x6h-coded.x6l);
35 %Coded Output
36
37 y1=(y1-coded.y1l)./(coded.y1h-coded.y1l);
38 y2=(y2-coded.y2l)./(coded.y2h-coded.y2l);
39 y3=(y3-coded.y3l)./(coded.y3h-coded.y3l);
40 y4=(y4-coded.y4l)./(coded.y4h-coded.y4l);
41 y5=(y5-coded.y5l)./(coded.y5h-coded.y5l);
42
43
44 X = [x1 x2 x3 x4 x5 x6];
45 Y=y5;
46
47 %Use MARS model
48
```

```
49 %Set the parameters
50 params = params(56, [], [], 0, [], 4);
51
52 %Build the model
53 model = build(X, Y, params)
54
55
56 % %Evaluate the equations
57 %  X=allunga(X);
58
59 yp=PROVAequationBis(X,model);
60
61 %UNCODE THE VARIABLES
62 X(:,1)=X(:,1).*(coded.x1h-coded.x1l)+coded.x1l;
63 X(:,2)=X(:,2).*(coded.x2h-coded.x2l)+coded.x2l;
64 X(:,3)=X(:,3).*(coded.x3h-coded.x3l)+coded.x3l;
65 X(:,4)=X(:,4).*(coded.x4h-coded.x4l)+coded.x4l;
66 X(:,5)=X(:,5).*(coded.x5h-coded.x5l)+coded.x5l;
67 X(:,6)=X(:,6).*(coded.x6h-coded.x6l)+coded.x6l;
68
69 yp(:)=yp(:).*(coded.y5h-coded.y5l)+coded.y5l; %THE UNCODING PARAMETERS
       HAVE TO CORRESPOND TO THE VARIABLE OF INTEREST
70
71 % %Plot pair-wise contribution of variables
72 % Graphics(X,yp,Xp,Yp)
73
74 save('FuelComp1HC')
75
76 anova(model,X,Yp)
77
```

```
78
79 %CHeck
80 plot(Yp, 'DisplayName', 'Yp', 'YDataSource', 'Yp'); figure(gcf)
81 hold on
82 plot(yp, 'r', 'DisplayName', 'yp', 'YDataSource', 'yp'); figure(gcf)
83
84
85
86 function yp=PROVAequationBis(X, model)
87 % Create the equation using the coefficients and the knot determined by
       the
88 % MARS algorithm
89 %
90  yp=zeros(length(X),length(model.coefs));
91 yp(:,1)=model.coefs(1);
92 i=1;
93 papa=[0;model.parents(:)];
94
95 for i=1:length(model.coefs)-1    % I coefficienti sono uno in piu
      rispetto alle BF
96
97      %Scrive la ultima BF, che non e' mai imparentata
98      x = X(:,model.knotdims{i}(length(model.knotdims{i})));
99      BF(:,i)= max(0, (x - model.knotsites{i}(length(model.knotsites{i}))
            )*model.knotdirs{i}(length(model.knotdirs{i})));
100
101
102     %Stdudia le parentele
103
104     fifi(i)=length(model.knotsites{i});
```

```
105     pipi(i)=length(model.knotsites{i})';
106
107     if fifi(i)>1
108
109
110     if model.parents(i)>0
111         x = X(:,model.knotdims{i}(length(model.knotdims{i})));
112         BF(:,i)=BF(:,papa(i+1)).*max(0, (x - model.knotsites{i}(fifi(i)
                ))*model.knotdirs{i}(fifi(i)));
113             %Se due BF sono imparentatati il BF parente si prende tale e
                    quale senza cambiare neanche la direzione
114             pipi(i)= pipi(i)-fifi(papa(i+1));
115             %papa(i+1)=0;
116     end
117
118
119     if pipi(i)>1
120
121
122             if  papa(model.parents(i)+1)==0 %Studia il caso in cui non
                    c'e' parantela ma ci sono piu variabili
123                 for j=1:pipi(i)-1;   %Ci possono essere piu variabili
                        dentro la BF(i)
124
125                 x2 = X(:,model.knotdims{i}(j));
126                  BF(:,i)=BF(:,i).*max(0,(x2 - model.knotsites{i}(j))*
                        model.knotdirs{i}(j));
127                 %BF(:,i)= max(0,(x - model.knotsites{i}(fifi(i)))*
                        model.knotdirs{i}(fifi(i))).*max(0,(x2 - model.
                        knotsites{i}(1))*model.knotdirs{i}(1));
```

```
128                     end
129                 end
130             end
131     end
132
133     yp(:,i+1)=yp(:,i)+ model.coefs(i+1).*BF(:,i);
134 end
135       yp=yp(:,length(model.knotdims)+1);
136
137 return
138
139
140
141 function trainParams = params(maxFuncs, c, cubic, cubicFastLevel, ...
142 selfInteractions, maxInteractions, threshold, prune, useMinSpan, ...
143 useEndSpan, maxFinalFuncs)
144
145 if (nargin < 1) || (isempty(maxFuncs))
146     trainParams.maxFuncs = 21;
147 else
148     trainParams.maxFuncs = maxFuncs;
149 end
150
151 if (nargin < 2) || (isempty(c))
152     trainParams.c = 3;
153 else
154     trainParams.c = c;
155 end
156
157 if (nargin < 3) || (isempty(cubic))
```

```
158     trainParams.cubic = true;
159 else
160     trainParams.cubic = cubic;
161 end
162
163 if (nargin < 4) || (isempty(cubicFastLevel))
164     trainParams.cubicFastLevel = 2;
165 else
166     trainParams.cubicFastLevel = cubicFastLevel;
167 end
168
169 if (nargin < 5) || (isempty(selfInteractions))
170     trainParams.selfInteractions = 1;
171 else
172     trainParams.selfInteractions = selfInteractions;
173 end
174 if (trainParams.cubic) && (trainParams.selfInteractions > 1)
175     trainParams.selfInteractions = 1;
176 end
177
178 if (nargin < 6) || (isempty(maxInteractions))
179     trainParams.maxInteractions = 1; % applicable maximum is d *
          trainParams.selfInteractions
180 else
181     trainParams.maxInteractions = maxInteractions;
182 end
183
184 if (nargin < 7) || (isempty(threshold))
185     trainParams.threshold = 1e-4;
186 else
```

```
187     trainParams.threshold = threshold;
188 end
189
190 if (nargin < 8) || (isempty(prune))
191     trainParams.prune = true;
192 else
193     trainParams.prune = prune;
194 end
195
196 if (nargin < 9) || (isempty(useMinSpan))
197     trainParams.useMinSpan = -1; % default = -1 = automatic
198 else
199     if useMinSpan == 0
200         trainParams.useMinSpan = 1; % 1 and 0 is the same here (no
              endspan)
201     else
202         trainParams.useMinSpan = useMinSpan;
203     end
204 end
205
206 if (nargin < 10) || (isempty(useEndSpan))
207     trainParams.useEndSpan = -1; % default = -1 = automatic
208 else
209     if useEndSpan == 0
210         trainParams.useEndSpan = 1; % 1 and 0 is the same here (no
              endspan)
211     else
212         trainParams.useEndSpan = useEndSpan;
213     end
214 end
```

```
215
216 if (nargin < 11) || (isempty(maxFinalFuncs))
217     trainParams.maxFinalFuncs = Inf;
218 else
219     trainParams.maxFinalFuncs = maxFinalFuncs;
220 end
221
222 return
223
224
225 function [model, time] = build(Xtr, Ytr, trainParams, weights)
226
227
228 if trainParams.maxInteractions >= 2
229     trainParams_actual_c = trainParams.c;
230 else
231     trainParams_actual_c = 2*trainParams.c/3; % penalty coefficient for
          additive modelling
232 end
233
234
235 if trainParams.useMinSpan == 0
236     trainParams.useMinSpan = 1; % 1 and 0 is the same here (no endspan)
237 end
238 if trainParams.useEndSpan == 0
239     trainParams.useEndSpan = 1; % 1 and 0 is the same here (no endspan)
240 end
241 if (nargin < 4)
242     weights = [];
243 else
```

```
244
245 end
246 wd = diag(weights);
247 if nargin < 5
248     modelOld = [];
249 end
250
251
252  fprintf('Building model...\n');
253 ws = warning('off');
254 tic;
255
256 maxIters = floor(trainParams.maxFuncs / 2); % because basis functions
      are added two at a time
257 YtrMean = mean(Ytr);
258 YtrSS = sum((Ytr - YtrMean) .^ 2);
259 minX = min(Xtr);
260 maxX = max(Xtr);
261
262 if trainParams.useEndSpan < 0
263     endSpan = getEndSpan(d);
264 else
265     endSpan = trainParams.useEndSpan;
266 end
267
268 if isempty(modelOld)
269     X = ones(n,1);
270     err = 1; % errore normalizzato al livello costante
271     model.coefs = YtrMean;
272     model.knotdims = {};
```

```
273     model.knotsites = {};
274     model.knotdirs = {};
275     model.parents = [];
276     model.trainParams = [];
277     model.MSE = Inf;
278     model.GCV = Inf;
279 else
280     model = modelOld; % Usa modelOld come modello iniziale
281 end
282
283 if endSpan*2 >= n
284     if isempty(modelOld)
285         model.MSE = YtrSS / n;
286         model.GCV = gcv(model, model.MSE, n, trainParams_actual_c);
287         if trainParams.cubic
288             model.t1 = [];
289             model.t2 = [];
290         end
291     end
292 else
293
294     % FORWARD PHASE
295
296     if isempty(modelOld) % no forward phase when modelOld is used
297
298         fprintf('Forward phase   .');
299
300         % crea una lista ordinata di elementi
301         [ordinatiXtr ordinatiXtrInd] = sort(Xtr);
302         if trainParams.useEndSpan ~= 1
```

```
303             % elimina i dati iniziali e finali
304             ordinatiXtr = ordinatiXtr(endSpan:end-(endSpan-1),:);
305             ordinatiXtrInd = ordinatiXtrInd(endSpan:end-(endSpan-1),:);
306         end
307
308         if trainParams.cubic
309             tmp_t1 = [];
310             tmp_t2 = [];
311         end
312         basisFunctionList = []; % basis functions candidate
313         BasisAggiunte = 0; % basis functions aggiunte all ultima
                iterazione
314
315         % the main loop of the forward phase
316         for depth = 1 : maxIters
317             basisFunctionList = createList(basisFunctionList, Xtr,
                    ordinatiXtr, ordinatiXtrInd, ...
318                                             n, d, model, BasisAggiunte,
                                                trainParams, endSpan);
319
320             % ferma la forward phase se non ci sono candidati
321             if isempty(basisFunctionList)
322                 if trainParams.cubic
323                     t1 = tmp_t1;
324                     t2 = tmp_t2;
325                 end
326                 break;
327             end
328
329             tmpErr = inf(1,size(basisFunctionList,2));
```

```
330             tmpCoefs = inf(length(model.coefs)+2, size(
                    basisFunctionList,2));
331             Xtmp = zeros(n,size(X,2)+2);
332             if ~trainParams.cubic
333                 Xtmp(:,1:end-2) = X;
334             end
335
336             % prova tutte le accoppiate di basis functions
337             for i = 1 : size(basisFunctionList,2)
338                 if trainParams.cubic
339                     [t1 t2 dif] = KnotsEstremi(model, basisFunctionList
                            {1,i}, basisFunctionList{2,i}, ...
340                                 d, minX, maxX, tmp_t1, tmp_t2);
341                     Xtmp(:,1:end-2) = X;
342                     % update basis functions with the updated side
                            knots
343                     for j = 1 : length(model.knotdims)
344                         if dif(j)
345                             Xtmp(:,j+1) = createbasisfunction(Xtr, Xtmp
                                    , model.knotdims{j}, model.knotsites{j},
                                    ...
346                                         model.knotdirs{j}, model.
                                            parents(j), minX, maxX, t1
                                            (j,:), t2(j,:));
347                         end
348                     end
349                     % New basis function
350                     dirs = basisFunctionList{3,i};
351                     Xtmp(:,end-1) = createbasisfunction(Xtr, Xtmp,
                            basisFunctionList{1,i}, basisFunctionList{2,i},
```

```
                ...
352                                 dirs, basisFunctionList{4,i}, minX,
                                        maxX, t1(end,:), t2(end,:));
353             if isnan(Xtmp(1,end-1)), Xtmp(:,end-1) = []; end
354             % Reflected partner
355             dirs(end) = -dirs(end);
356             Xtmp(:,end) = createbasisfunction(Xtr, Xtmp,
                    basisFunctionList{1,i}, basisFunctionList{2,i},
                    ...
357                                 dirs, basisFunctionList{4,i}, minX,
                                        maxX, t1(end,:), t2(end,:));
358             if isnan(Xtmp(1,end)), Xtmp(:,end) = []; end
359         else
360             % New basis function
361             dirs = basisFunctionList{3,i};
362             Xtmp(:,end-1) = createbasisfunction(Xtr, Xtmp,
                    basisFunctionList{1,i}, ...
363                                 basisFunctionList{2,i}, dirs,
                                        basisFunctionList{4,i}, minX,
                                        maxX);
364             if isnan(Xtmp(1,end-1)), Xtmp(:,end-1) = []; end
365             % Reflected partner
366             dirs(end) = -dirs(end);
367             Xtmp(:,end) = createbasisfunction(Xtr, Xtmp,
                    basisFunctionList{1,i}, ...
368                                 basisFunctionList{2,i}, dirs,
                                        basisFunctionList{4,i}, minX, maxX
                                        );
369             if isnan(Xtmp(1,end)), Xtmp(:,end) = []; end
370         end
```

```
371                 if size(Xtmp,2) == size(X,2)+2 % crea una coppia di
                        basis functions
372                     [coefs tmpErr(i)] = lreg(Xtmp, Ytr, weights, wd);
373                     tmpErr(i) = tmpErr(i) / YtrSS;
374                     tmpCoefs(:,i) = coefs;
375                 elseif size(Xtmp,2) == size(X,2)+1 % crea una sola
                        basis function
376                     [coefs tmpErr(i)] = lreg(Xtmp, Ytr, weights, wd);
377                     tmpErr(i) = tmpErr(i) / YtrSS;
378                     tmpCoefs(:,i) = [coefs; NaN];
379                     Xtmp = [Xtmp zeros(n,1)];
380                 else % no basis function created (size(Xtmp,2) == size(
                        X,2))
381                     tmpErr(i) = Inf;
382                     Xtmp = [Xtmp zeros(n,2)];
383                 end
384             end
385
386             [newErr, ind] = min(tmpErr); % analizza il contributo della
                    base aggiunta
387
388 %Ferma la forward phase se non sta dando contributo
389             if (isnan(newErr)) || (err(end) - newErr < trainParams.
                    threshold)
390                 if trainParams.cubic
391                     t1 = tmp_t1;
392                     t2 = tmp_t2;
393                 end
394                 break;
395             end
```

```
396 %Update the model with new basis function
397
398             if trainParams.cubic
399                 [t1 t2 dif] = KnotsEstremi(model, basisFunctionList{1,
                        ind}, basisFunctionList{2,ind}, ...
400                                 d, minX, maxX, tmp_t1, tmp_t2);
401                 % update basis functions with the updated side knots
402                 for j = 1 : length(model.knotdims)
403                     if dif(j)
404                         X(:,j+1) = createbasisfunction(Xtr, X, model.
                                knotdims{j}, model.knotsites{j}, ...
405                                         model.knotdirs{j}, model.parents(j),
                                            minX, maxX, t1(j,:), t2(j,:));
406                     end
407                 end
408                 % Add the new basis function
409                 dirs = basisFunctionList{3, ind};
410                 Xn = createbasisfunction(Xtr, X, basisFunctionList{1,
                        ind}, basisFunctionList{2,ind}, ...
411                     dirs, basisFunctionList{4,ind}, minX, maxX, t1(end
                            ,:), t2(end,:));
412                 if isnan(Xn(1)), Xn = []; end
413                 % aggiunge una coppia di basis functions
414                 dirs(end) = -dirs(end);
415                 Xn2 = createbasisfunction(Xtr, X, basisFunctionList{1,
                        ind}, basisFunctionList{2,ind}, ...
416                     dirs, basisFunctionList{4,ind}, minX, maxX, t1(
                            end,:), t2(end,:));
417                 if isnan(Xn2(1)), Xn2 = []; end
418                 X = [X Xn Xn2];
```

```
419             if ~isempty(Xn) && ~isempty(Xn2) % aggiunge una sola
                    basis function
420                 t1(end+1,:) = t1(end,:);
421                 t2(end+1,:) = t2(end,:);
422             end
423         else
424             dirs = basisFunctionList{3, ind};
425             % Add the new basis function
426             Xn = createbasisfunction(Xtr, X, basisFunctionList{1,
                    ind}, ...
427                   basisFunctionList{2,ind}, dirs, basisFunctionList
                          {4,ind}, minX, maxX);
428             if isnan(Xn(1)), Xn = []; end
429             % Add the reflected partner
430             dirs(end) = -dirs(end);
431             Xn2 = createbasisfunction(Xtr, X, basisFunctionList{1,
                    ind}, ...
432                   basisFunctionList{2,ind}, dirs, basisFunctionList
                          {4,ind}, minX, maxX);
433             if isnan(Xn2(1)), Xn2 = []; end
434             X = [X Xn Xn2];
435         end
436
437         model.coefs = tmpCoefs(:,ind);
438
439         % add the basis functions to the model
440         BasisAggiunte = 0;
441         dirs = basisFunctionList{3, ind};
442         if ~isempty(Xn)
443             model.knotdims{end+1,1} = basisFunctionList{1, ind};
```

```
444                 model.knotsites{end+1,1} = basisFunctionList{2, ind};
445                 model.knotdirs{end+1,1} = dirs;
446                 model.parents(end+1,1) = basisFunctionList{4, ind};
447                 BasisAggiunte = BasisAggiunte + 1;
448             else
449                 model.coefs(end) = [];
450             end
451             if ~isempty(Xn2)
452                 dirs(end) = -dirs(end);
453                 model.knotdims{end+1,1} = basisFunctionList{1, ind};
454                 model.knotsites{end+1,1} = basisFunctionList{2, ind};
455                 model.knotdirs{end+1,1} = dirs;
456                 model.parents(end+1,1) = basisFunctionList{4, ind};
457                 BasisAggiunte = BasisAggiunte + 1;
458             else
459                 model.coefs(end) = [];
460             end
461
462
463
464
465             err(end+1) = newErr;
466             if (newErr < trainParams.threshold) || ...
467                (length(model.coefs) + 2 > n)
468                 break;
469             end
470
471             if trainParams.cubic
472                 tmp_t1 = t1;
473                 tmp_t2 = t2;
```

```
474             end
475             basisFunctionList(:,ind) = [];
476         end % end of the main loop
477
478         if verbose, fprintf('\n'); end
479
480     end % end of "isempty(modelOld)"
481
482
483
484     % BACKWARD PHASE
485
486     if trainParams.prune
487
488         fprintf('Backward phase .');
489
490         if ~isempty(modelOld) % Se non c'e' un modello gia scremato
                rianalizza tutte le basi
491             if (doCubicFastLevel == -1) || (doCubicFastLevel >= 2)
492
493                 X = ones(n,length(model.knotdims)+1);
494                 for i = 1 : length(model.knotdims)
495                     X(:,i+1) = createbasisfunction(Xtr, X, model.
                            knotdims{i}, model.knotsites{i}, ...
496                                 model.knotdirs{i}, model.parents(i),
                                        minX, maxX);
497                 end
498                 [model.coefs model.MSE] = lreg(X, Ytr, weights, wd);
499                 model.MSE = model.MSE / n;
```

```
500                 model.GCV = gcv(model, model.MSE, n,
                        trainParams_actual_c);
501             else
502                 % create all the basis functions (cubic) from scratch
503                 t1 = model.t1;
504                 t2 = model.t2;
505                 X = ones(n,length(model.knotdims)+1);
506                 for i = 1 : length(model.knotdims)
507                     X(:,i+1) = createbasisfunction(Xtr, X, model.
                            knotdims{i}, model.knotsites{i}, ...
508                                 model.knotdirs{i}, model.parents(i),
                                        minX, maxX, t1(i,:), t2(i,:));
509                 end
510             end
511         end
512
513         models = {model};
514         mses = model.MSE;
515         gcvs = model.GCV;
516
517         % the main loop of the backward phase
518         for j = 1 : length(model.knotdims)
519             tmpErr = inf(1, length(model.knotdims));
520             tmpCoefs = inf(length(model.coefs)-1, length(model.knotdims
                    ));
521
522             % cancella le basis functions una alla volta
523
524             for k = 1 : length(model.knotdims)
525                 Xtmp = X;
```

```
526                 Xtmp(:,k+1) = [];
527                 if trainParams.cubic
528                     % create temporary t1, t2, and model with a deleted
                            basis function
529                     tmp_t1 = t1;
530                     tmp_t1(k,:) = [];
531                     tmp_t2 = t2;
532                     tmp_t2(k,:) = [];
533                     tmp_model.knotdims = model.knotdims;
534                     tmp_model.knotdims(k) = [];
535                     tmp_model.knotsites = model.knotsites;
536                     tmp_model.knotsites(k) = [];
537                     tmp_model.knotdirs = model.knotdirs;
538                     tmp_model.knotdirs(k) = [];
539                     tmp_model.parents = model.parents;
540                     tmp_model.parents(k) = [];
541                     tmp_model.parents = updateParents(tmp_model.parents
                            , k);
542                     [tmp_t1 tmp_t2 dif] = KnotsEstremi(tmp_model, [],
                            [], d, minX, maxX, tmp_t1, tmp_t2);
543                     % update basis functions with the updated side
                            knots
544                     for i = 1 : length(tmp_model.knotdims)
545                         if dif(i)
546                             Xtmp(:,i+1) = createbasisfunction(Xtr, Xtmp
                                    , tmp_model.knotdims{i}, tmp_model.
                                    knotsites{i}, ...
547                                             tmp_model.knotdirs{i},
                                                tmp_model.parents(i), minX
                                                , maxX, tmp_t1(i,:),
```

```
                                                    tmp_t2(i,:));
548                         end
549                     end
550                 end
551                 [coefs tmpErr(k)] = lreg(Xtmp, Ytr, weights, wd);
552                 tmpCoefs(:,k) = coefs;
553             end
554
555             [dummy, ind] = min(tmpErr); % find out the best
                    modification
556             X(:,ind+1) = [];
557             model.coefs = tmpCoefs(:,ind);
558             model.knotdims(ind) = [];
559             model.knotsites(ind) = [];
560             model.knotdirs(ind) = [];
561             model.parents(ind) = [];
562             model.parents = updateParents(model.parents, ind);
563
564             if trainParams.cubic
565                 t1(ind,:) = [];
566                 t2(ind,:) = [];
567                 [t1 t2 dif] = KnotsEstremi(model, [], [], d, minX, maxX
                        , t1, t2);
568                 % update basis functions with the updated side knots
569                 for i = 1 : length(model.knotdims)
570                     if dif(i)
571                         X(:,i+1) = createbasisfunction(Xtr, X, model.
                                knotdims{i}, model.knotsites{i}, ...
572                                         model.knotdirs{i}, model.parents(i),
                                             minX, maxX, t1(i,:), t2(i,:));
```

```
573                     end
574                 end
575                 model.t1 = t1;
576                 model.t2 = t2;
577             end
578
579             models{end+1} = model;
580             mses(end+1) = tmpErr(ind) / n;
581             gcvs(end+1) = gcv(model, mses(end), n, trainParams_actual_c
                    );
582
583             % end of the main loop
584
585         if trainParams.maxFinalFuncs >= length(models{1}.coefs)
586             [g, ind] = min(gcvs);
587         elseif trainParams.maxFinalFuncs > 1
588             [g, ind] = min(gcvs(end-trainParams.maxFinalFuncs+1:end));
589             ind = ind + length(gcvs) - trainParams.maxFinalFuncs;
590         else
591             g = gcvs(end);
592             ind = length(gcvs);
593         end
594         model = models{ind};
595
596         if doCubicFastLevel >= 2
597             % turn the cubic modelling on
598             trainParams.cubic = true;
599             [t1 t2] = KnotsEstremi(model, [], [], d, minX, maxX, [],
                        []);
600             % update all the basis functions
```

```
601             X = ones(n,length(model.coefs));
602             for i = 1 : length(model.knotdims)
603                 X(:,i+1) = createbasisfunction(Xtr, X, model.knotdims{i
                        }, model.knotsites{i}, ...
604                                 model.knotdirs{i}, model.parents(i), minX,
                                        maxX, t1(i,:), t2(i,:));
605             end
606             model.t1 = t1;
607             model.t2 = t2;
608             [model.coefs model.MSE] = lreg(X, Ytr, weights, wd);
609             model.MSE = model.MSE / n;
610             model.GCV = gcv(model, model.MSE, n, trainParams_actual_c);
611         else
612             model.MSE = mses(ind);
613             model.GCV = g;
614         end
615
616         if verbose, fprintf('\n'); end
617
618     end % end of "trainParams.prune"
619
620 end % end of "if endSpan*2 >= n"
621
622 model.trainParams = trainParams;
623 model.minX = minX;
624 model.maxX = maxX;
625 model.endSpan = endSpan;
626
627 time = toc;
628 if verbose
```

```
629     fprintf('Number of basis functions in the final model: %d\n',
            length(model.coefs));
630     fprintf('Total effective number of parameters: %0.1f\n', ...
631             length(model.coefs) + model.trainParams.c * length(model.
                    knotdims) / 2);
632     maxDeg = 0;
633     if length(model.knotdims) > 0
634         for i = 1 : length(model.knotdims)
635             if length(model.knotdims{i}) > maxDeg
636                 maxDeg = length(model.knotdims{i});
637             end
638         end
639     end
640
641 end
642 warning(ws);
643 return
644
645
646 %%%%%%%%%%%%%%%%%  Auxiliary functions
647
648 function g = gcv(model, MSE, n, c)
649 % Calculates GCV
650
651 enp = length(model.coefs) + c * length(model.knotdims) / 2; % parametri
        nel modello
652 if enp >= n
653     g = Inf;
654 else
655     p = 1 - enp / n;
```

```
656       g = MSE / (p * p);
657 end
658 return
659
660 function parents = updateParents(parents, deletedInd)
661 % Updates direct parent indexes after deletion of a basis function.
662 parents(parents == deletedInd) = 0;
663 tmp = parents > deletedInd;
664 parents(tmp) = parents(tmp) - 1;
665 return
666
667 function basisFunctionList = createList(basisFunctionList_old, Xtr, ...
668 ordinatiXtr, ordinatiXtrInd, n, d, model, BasisAggiunte, trainParams,
      endSpan)
669 % Takes the old list of basis functions and adds new ones according to
      the
670 % current model. If the old list is empty, adds only linear basis
671 % functions. If it is non-empty, adds only basis functions with
672 % interactions which result from the last BasisAggiunte basis functions
673
674
675
676 % Create linear basis functions
677
678 if (isempty(basisFunctionList_old)) && (BasisAggiunte == 0)
679       basisFunctionList = cell(3,0);
680       counter = 1;
681       if trainParams.useMinSpan ~= 1
682           % get the list of knot sites allowed due to minSpan
```

```
683         massimi = mod(1:n, getMinSpan(d, n, trainParams.useMinSpan)) ==
                0;
684     end
685     for di = 1 : d %for each dimension
686         if trainParams.useMinSpan ~= 1
687             massimiPos = ordinatiXtr(:,di);
688         else
689             massimiPos = unique(ordinatiXtr(:,di));
690         end
691         lastknot = Inf;
692 %aggiunge le basis functions alla lista
693         for i = 1 : size(massimiPos,1)
694             if (trainParams.useMinSpan == 1) || ...
695                ((massimi(i + endSpan-1)) && (lastknot ~= massimiPos(i))
                    )
696                 lastknot = massimiPos(i);
697                 basisFunctionList{1, counter} = di;
698                 basisFunctionList{2, counter} = lastknot;
699                 basisFunctionList{3, counter} = 1;
700                 basisFunctionList{4, counter} = 0;
701                 counter = counter + 1;
702             end
703         end
704     end
705     return
706 end
707
708 if (trainParams.maxInteractions < 2) || (BasisAggiunte < 1)
709     basisFunctionList = basisFunctionList_old;
710     return
```

```
711 end
712
713 % Create basis functions with interactions
714
715 basisFunctionList = basisFunctionList_old;
716 counter = size(basisFunctionList_old,2) + 1;
717 start = length(model.knotdims)-(BasisAggiunte-1);
718
719 % loop through all the basis functions already in the model
720 for j = start : length(model.knotdims)
721     if length(model.knotdims{j}) < trainParams.maxInteractions
722         alloweddims = 1 : d;
723         if trainParams.selfInteractions <= 1
724             % will not consider the already used dimensions
725             alloweddims = setdiff(alloweddims, model.knotdims{j});
726         else
727             for i = 1 : d
728                 if length(find(model.knotdims{j} == i)) >= trainParams.
                        selfInteractions
729                     alloweddims = setdiff(alloweddims, i);
730                 end
731             end
732         end
733         if isempty(alloweddims)
734             continue
735         end
736
737         if trainParams.useMinSpan ~= 1
738
```

```
739             nonzero = listNonZero(Xtr, model.knotdims{j}, model.
                    knotsites{j}, model.knotdirs{j});
740             minSpan = getMinSpan(d, length(find(nonzero)), trainParams.
                    useMinSpan);
741             if ~isfinite(minSpan)
742                 continue
743             end
744             massimi = mod(1:n, minSpan) == 0;
745             for di = alloweddims %for each dimension
746                 lastknot = Inf;
747
748                 ind = ordinatiXtrInd(nonzero(ordinatiXtrInd(:,di)),di);
749                 ind = ind(massimi(ind))';
750                 for i = ind
751                     if lastknot ~= Xtr(i,di)
752                         lastknot = Xtr(i,di);
753                         basisFunctionList{1, counter} = [model.knotdims
                                {j} di];
754                         basisFunctionList{2, counter} = [model.
                                knotsites{j} lastknot];
755                         basisFunctionList{3, counter} = [model.knotdirs
                                {j} 1];
756                         basisFunctionList{4, counter} = j;
757                         counter = counter + 1;
758                     end
759                 end
760             end
761
762         else
763
```

```
764             nonzero = listNonZero(Xtr, model.knotdims{j}, model.
                    knotsites{j}, model.knotdirs{j});
765             for di = alloweddims %aggiunge una base per ogni dimensione
766
767                 massimiPos = unique(ordinatiXtr(nonzero(ordinatiXtrInd
                        (:,di)),di));
768                 for i = 1 : size(massimiPos,1)
769                     basisFunctionList{1, counter} = [model.knotdims{j}
                            di];
770                     basisFunctionList{2, counter} = [model.knotsites{j}
                             massimiPos(i)];
771                     basisFunctionList{3, counter} = [model.knotdirs{j}
                            1];
772                     basisFunctionList{4, counter} = j;
773                     counter = counter + 1;
774                 end
775             end
776
777         end
778
779     end
780 end
781 return
782
783 function nonzero = listNonZero(Xtr, knotdims, knotsites, knotdirs)
784 % Lists nonzero (according to the parent basis function) sites where
        knots
785 % may be placed.
786 nonzero = true(size(Xtr,1),1);
787 for j = 1 : size(Xtr,1)
```

```
788     for i = 1 : length(knotdims)
789         z = Xtr(j,knotdims(i)) - knotsites(i);
790         if ((z >= 0) && (knotdirs(i) < 0)) || ...
791            ((z <= 0) && (knotdirs(i) > 0))
792             nonzero(j) = false;
793             break;
794         end
795     end
796 end
797 return
798
799 function s = getEndSpan(d)
800 % Calculation of endSpan so that potential knot sites that are too
      close to
801 % the ends of data intervals are not considered.
802 %s = floor(3 - log2(0.05/d));
803 s = floor(7.32193 + log(d) / 0.69315); % precomputed version
804 if s < 1, s = 1; end
805 return
806
807 function s = getMinSpan(d, nz, param)
808 %Determina la posizione per i nuovi knot
809 if nz == 0
810     s = Inf;
811 else
812     if param < 0 % automatic
813         %s = floor(-log2(-log(1-0.05)/(d*nz)) / 2.5);
814         s = floor((2.9702 + log(d*nz)) / 1.7329);
815     else
816         s = param;
```

```
817             if s > nz
818                 s = Inf;
819             end
820         end
821         if s < 1, s = 1; end
822 end
823 return
824
825 function [coefs err] = lreg(x, y, w, wd)
826 % Linear regression (unweighted and weighted)
827 if isempty(wd)
828         coefs = (x' * x) \ (x' * y);
829         err = sum((y-x*coefs).^2);
830 else
831         x_wd = x' * wd;
832         coefs = (x_wd * x) \ (x_wd * y);
833         err = sum((y-x*coefs).^2.*w);
834 end
835 return
836
837
838
839 function graFICA(X,yp,Xp,Yp)
840 %%%%%Grafica
841
842
843 %Plot y as a function of X1 and X2 with X3 locked at 0.1
844 i=1:11:length(yp);
845 y=yp(i);
846 x1=X(1:121:length(yp),1);
```

```
847 x2=X(1:11:121,2);
848 YYY=reshape(y,11,11);
849
850 figure(1)
851 surfc(x1,x2,YYY)
852 hold on
853
854 j=1:3:length(Yp); %Takes only the 30%fuel split points
855 Xs=Xp(j,:);
856 Ys=Yp(j);
857
858 scatter3(Xs(:,1),Xs(:,2),Ys,'filled')
859 title('HC (ppm)for 30% fuel split');
860 xlabel('Main SOI ( BTDC)')
861 ylabel('Pilot SOI ( BTDC)')
862 zlabel('HC (ppm)')
863
864
865 % 35% fuel split
866 clear y x1 x2 YYY Xs Ys i j
867
868 i=6:11:length(yp);
869 y=yp(i);
870 x1=X(6:121:length(yp),1);
871 x2=X(6:11:121,2);
872 YYY=reshape(y,11,11);
873
874 figure(2)
875 surfc(x1,x2,YYY)
876 hold on
```

```
877
878 j=2:3:length(Yp); %Takes only the 35%fuel split points
879 Xs=Xp(j,:);
880 Ys=Yp(j);
881
882 scatter3(Xs(:,1),Xs(:,2),Ys,'filled')
883 title('HC (ppm)for 35% fuel split');
884 xlabel('Main SOI ( BTDC)')
885 ylabel('Pilot SOI ( BTDC)')
886 zlabel('HC (ppm)')
887
888 % 40% fuel split
889 clear y x1 x2 YYY Xs Ys i j
890
891 i=11:11:length(yp);
892 y=yp(i);
893 x1=X(11:121:length(yp),1);
894 x2=X(11:11:121,2);
895 YYY=reshape(y,11,11);
896
897 figure(3)
898 surfc(x1,x2,YYY)
899 hold on
900
901 j=3:3:length(Yp); %Takes only the 40%fuel split points
902 Xs=Xp(j,:);
903 Ys=Yp(j);
904
905 scatter3(Xs(:,1),Xs(:,2),Ys,'filled')
906 title('HC (ppm)for 40% fuel split');
```

```
907 xlabel('Main SOI (  BTDC)')
908 ylabel('Pilot SOI (  BTDC)')
909 zlabel('HC (ppm)')
910
911 %%%% MAIN SOI & FUEL SPLIT
912 clear y x1 x2 YYY Xs Ys i j
913 clear i k j l d A
914 %generate the index for selecting the apropriate points
915 i=12:121:length(yp);
916    i=i';
917    k=0:10;
918    k=k';
919    k=repmat(k,length(i),1);
920    l=1:length(k);
921    l=l';
922    j(l,1)=i(floor((l-1)/11+1))+k(l);
923
924    %select the points
925 y=yp(j);
926 x1=X(1:121:length(yp),1);
927 x3=X(1:11,3);
928 YYY=reshape(y,11,11);
929
930 figure(4)
931 surfc(x1,x3,YYY)
932 hold on
933
934 clear j
935
936 b=1:9:length(Yp); %Takes only the lower Pilot injection points
```

```
937 b=b';
938 v=0:2;
939 v=v';
940 v=repmat(v,length(b),1);
941 m=1:length(v);
942 m=m';
943 j(m)=b(floor((m-1)/3+1))+v(m);
944
945
946 Xs=Xp(j,:);
947 Ys=Yp(j);
948
949 scatter3(Xs(:,1),Xs(:,3),Ys,'filled')
950 title(' HC (ppm) for 40  BTDC pilot injection');
951 xlabel('Main SOI (  BTDC)')
952 ylabel('Fuel Split (%)')
953 zlabel('HC (ppm)')
954
955 %%%% 45deg BTDC Pilot Injection
956 clear y x1 x2 YYY Xs Ys i j
957 clear i k j l d A
958 %generate the index for selecting the apropriate points
959 i=56:121:length(yp);
960    i=i';
961    k=0:10;
962    k=k';
963    k=repmat(k,length(i),1);
964    l=1:length(k);
965    l=l';
966    j(l,1)=i(floor((l-1)/11+1))+k(l);
```

```
967
968   %select the points
969 y=yp(j);
970 x1=X(56:121:length(yp),1);
971 x3=X(1:11,3);
972 YYY=reshape(y,11,11);
973
974 figure(5)
975 surfc(x1,x3,YYY)
976 hold on
977
978 clear j
979
980 b=4:9:length(Yp); %Takes only the medium Pilot injection points
981 b=b';
982 v=0:2;
983 v=v';
984 v=repmat(v,length(b),1);
985 m=1:length(v);
986 m=m';
987 j(m)=b(floor((m-1)/3+1))+v(m);
988
989
990 Xs=Xp(j,:);
991 Ys=Yp(j);
992
993 scatter3(Xs(:,1),Xs(:,3),Ys,'filled')
994 title('HC (ppm) for 45 BTDC pilot injection');
995 xlabel('Main SOI ( BTDC)')
996 ylabel('Fuel Split (%)')
```

```
997 zlabel('HC (ppm)')
998
999 %50deg BTDC Pilot Injection
1000 clear y x1 x2 YYY Xs Ys i j
1001 clear i k j l d A
1002 %generate the index for selecting the apropriate points
1003 i=111:121:length(yp);
1004    i=i';
1005    k=0:10;
1006    k=k';
1007    k=repmat(k,length(i),1);
1008    l=1:length(k);
1009    l=l';
1010    j(l,1)=i(floor((l-1)/11+1))+k(l);
1011
1012    %select the points
1013 y=yp(j);
1014 x1=X(111:121:length(yp),1);
1015 x3=X(1:11,3);
1016 YYY=reshape(y,11,11);
1017
1018 figure(6)
1019 surfc(x1,x3,YYY)
1020 hold on
1021
1022 clear j
1023
1024 b=7:9:length(Yp); %Takes only the medium Pilot injection points
1025 b=b';
1026 v=0:2;
```

```
1027 v=v';
1028 v=repmat(v,length(b),1);
1029 m=1:length(v);
1030 m=m';
1031 j(m)=b(floor((m-1)/3+1))+v(m);
1032
1033
1034 Xs=Xp(j,:);
1035 Ys=Yp(j);
1036
1037 scatter3(Xs(:,1),Xs(:,3),Ys,'filled')
1038 title(' HC (ppm) for 50  BTDC pilot injection');
1039 xlabel('Main SOI (  BTDC)')
1040 ylabel('Fuel Split (%)')
1041 zlabel('HC (ppm)')
1042
1043 %%%% 2D Parametric plot
1044 clear y x1 x2 YYY Xs Ys i j
1045 clear i k j l d A
1046
1047 figure(7)
1048 i=11:121:length(yp);
1049 i=i';
1050 y1=yp(i);
1051 x=X(i,1);
1052 i2=56:121:length(yp);
1053 i2=i2';
1054 y2=yp(i2);
1055 i3=121:121:length(yp);
1056 i3=i3';
```

```
y3=yp(i3);
plot(x,y1,'-.g+');
title('Experimental vs Fitted data points');
xlabel('Main SOI')
ylabel('HC (ppm)')
hold on
plot(x,y2,'-.ro');
hold on
plot(x,y3,'-.ms');
legend('40 pilot SOI and 40% fuel split','45 pilot SOI and 40% fuel
    split','50 pilot SOI and 40% fuel split')
hold on
j=3:9:length(Yp);
Xs=Xp(j,1);
Ys=Yp(j);
scatter(Xs,Ys,100,'g+');
hold on
j=6:9:length(Yp);
Xs=Xp(j,1);
Ys=Yp(j);
scatter(Xs,Ys,100,'ro');
j=9:9:length(Yp);
Xs=Xp(j,1);
Ys=Yp(j);
scatter(Xs,Ys,100,'ms');
```

References

[1] IARC, *The carcinogenicity of outdoor air pollution.* in The Lancet Oncology , ed, 2013

[2] K. Straif, A. Cohen, J. Samet *Air Pollution and Cancer.* IARC Scientific Publication No. 161

[3] D. Carder, R. Ryskamp, J. Nuszkowski, H. Li, N. Clark, G. Thompson, M. Gautam, and S. Wayne, Center for Alternative Fuels, Engines, and Emissions West Virginia University *Fuels to Enable Light Duty Diesel Advanced Combustion Regimes.* CRC Report No. AVFL-16

[4] T.Gallant, J.Franz, S.Alnajjar, J.Storey, S.Lewis, C.Sluder, W.Cannella, C.Fairbridge, D.Hager, H.Dettman, J.Luecke, M.Ratcliff, B.Zigler, *Fuels for Advanced Combustion Engines Research Diesel Fuels: Analysis of Physical and Chemical Properties.* SAE International Journal of Fuels and Lubricant, 2009, Vol. 2, No. 2, pp. (262- 272)

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

[6] J.Friedman, Department of Statistics Standford University, *Multivariate adaptive regression splines (with discussion)*, Annals of statistics, 1990, pp. (1-141)

[7] S .Crino, D.E.Brown, *Global Optimization With Multivariate Adaptive Regression Splines.* IEEE Transaction on Systems, Man, and Cybernetics, 2007, pp. (333 - 340)

[8] G. Bression, D. Soleri, S. Savy, S. Dehoux, D. Azoulay, H. Hamouda, L. Doradoux, N. Guerrassi, N. Lawrence, *A study of methods to lower HC and CO emissions in diesel HCCI*, SAE Technical Paper 2008-01-0034

[9] P. Flynn, R. Durrett, G. Hunter, W. Akinyemi, L. Farrell, *Minimum Engine Flame Temperature Impacts on Diesel and Spark-Ignition Engine NOx Production*, SAE Technical Paper 2000-01-1177

[10] J. Green, *Novel Combustion Regimes for Higher Efficiency and Lower Emissions*, "Brown Bag" Luncheon Series, December 16, 2002

[11] D. Kawano, H. Suzuki, H.Ishii, Y.Goto, O. Matsuo, *Ignition and Combustion Control of Diesel HCCI*, SAE Technical Paper 2005-01-2132

[12] M. Ikegami, T. Kamimoto, *Conventional Diesel Combustion*, in Flow and Combustion in Reciprocating Engines, ed. C. Acroumanis and T, Kamimoto. Springer-Verlag, 2009, ISBN: 978-3-540-64142-1

[13] T.Minami,K.Takeuchi,N.Shimazaki, *Reduction of Diesel Engine NOx Using Pilot Injection.* SAE Technical Paper 950611, 1995.

[14] R. Hasegawa, H. Yanagihara, *HCCI Combustion in a DI Diesel Engine*, SAE Technical Paper 2003-01-0745

[15] H. Yanagihara, *Compression-ignition type engine*, US Patent 6,173,691

[16] Z. Peng, H. Zhao, N. Ladommatos, *Effects of Air/Fuel Ratios and EGR Rates on HCCI Combustion of n-heptane, a Diesel Type Fuel* SAE Technical Paper 2003-01-0747

[17] Sandia, 2005, *CFR Newsletter: Imaging of advanced low-temperature diesel combustion*, Combustion Research Facility News, Sandia National Laboratories, Livermore, CA, USA

[18] CRC, *Chemical and Physical Properties of the Fuels for Advanced Combustion Engines (FACE) Research Diesel Fuels* Coordinating Research Council, Inc. FACE-1, 2010

[19] G. Kalghatgi, P. Risberg, *Partially Pre-Mixed Auto-Ignition of Gasoline to Attain Low Smoke and Low NOx at High Load in a Compression Ignition Engine and Comparison with a Diesel Fuel*, SAE Technical Paper 2007-01-0006

[20] G. Shibata, T. Urushihara, *Auto-Ignition Characteristics of Hydrocarbons and Development of HCCI Fuel Index*, SAE Technical Paper 2007-01-0220

[21] B. Bunting, C. Wildman, J. Szybist, S. Lewis, J. Storey *Fuel chemistry and cetane effects on diesel homogeneous charge compression ignition performance, combustion, and emissions*, Int J Engine Res, 2007, Vol. 8, No. 21, pp. (15- 27)

[22] G. Shibata, K. Oyama, T. Urushihara, T. Nakano, *The Effect of Fuel Properties on Low and High Temperature Heat Release and Resulting Performance of an HCCI Engine*, SAE Technical Paper 2004 - 01-0553

[23] P. Risberg, *Describing the Auto-Ignition Quality of Fuels in HCCI Engines*, Doctoral thesis, School of Industrial Engineering and Management, Royal Institute of Technology, S-100 44 Stockholm

[24] S. Kobori, T. Kamimoto, A. Aradi, *A Study of Igntion Delay of Diesel Sprays* Int. J Engine Research, 2000, Vol 1, No. 1, pp. (29-39)

[25] J. Wang, G. Neely, T. Ryan, *On-Board Fuel Property Classifier for Fuel Property Adaptive Engine Control System*, SAE Technical Paper 2006-01-0054

[26] G. Martin, C. Muller, D. Milam, M. Radovanovic, C. Gehrke, *Early Direct-Injection, Low-Temperature Combustion of Diesel Fuel in an Optical Engine Utilizing a 15-Hole, Dual-Row, Narrow-Included-Angle Nozzle*, SAE Technical Paper 2008-01-2400

[27] A. Upatnieks, C. Muller, G. Martin, *The influence of charge-gas dilution and temperature on DI diesel combustion processes using a short-ignition-delay, oxygenated fuel*, SAE Technical Paper 2005-01-2008

[28] F. Tao, Y. Liu, B. Rempelewert, D. Foster, R. Reitz, D. Choi, P. Miles, *Modeling the effects of EGR and injection pressure on soot formation in a high-speed direct-injection (HSDI) diesel engine using a multi-step phenomenological soot model*, SAE Technical Paper 2005-01-0121

[29] M. Alriksson, S. Gjirja, I. Denbratt, *The Effect of Charge Air and Fuel Injection Parameters on Combustion with High Levels of EGR in a HDDI Single Cylinder Diesel Engine*, SAE Technical Paper 2007-01-0914

[30] A. Bhave, M. Kraft, L. Montorsi, F. Mauss, *Sources of CO emissions in an HCCI engine: A numerical analysis*, Combustion and Flames, 2006, Vol. 4, pp. (634-637)

[31] P.Bonsack, *Concentration and Size Distributions of Nanoparticle Emissions during Low Temperature Combustion using Fuels for Advanced Combustion Engines (FACE)*. Master Thesis, Department of Mechanical and Aerospace Engineering, 2012, WVU

[32] EPA. *Code of Federal Regulations*. Title 40, Part 1065, Engine-Testing Procedures.

[33] AVL, *AVL Micro Soot Sensor, AVL Exhaust Conditioning Unit, Operating Manual, Product Guide* 2008

[34] Horiba. *Automotive Emission Analysis System MEXA-7000, Series 7000 User's Guide*. 2007

[35] Horiba. *CO/CO_2 (HC,NO,N2O,SO2) Analyzer AIA-72X Series, Instruction Manual* 2007

[36] Horiba. *THC Analyzer FIA-725A, Instruction Manual* 2007

[37] D.S. Eddy, *Physical principles of the zirconia exhaust gas sensor* Vehicular Technology, IEEE Transactions, 1974, Vol. 23, pp. (125-128)

[38] Horiba *Non-Sampling Type NOx-A/F Analyzer MEXA-720NOx, User Manual* 2003

[39] R. Bellman *Adaptive Control Processes*, Princeton University Press, 1961

[40] J.Friedman, E. Grosse, W. Stuetzle, *Multidimensional additive spline approximation*, SIAM J. Sci. Stat. Comput. 1983, Vol. 4, No. 2, pp. (291301)

[41] AirVac, *TD Single Stage Vacuum Generators* 2012, Available: http://www.airvacpumps.com/TDpump.html

[42] G. Jekabsons, *ARESLab: Adaptive Regression Splines toolbox for Matlab/Octave*, 2011, available at http://www.cs.rtu.lv/jekabsons/

[43] T. Hastie, R. Tibshirani, *Generalized Additive Models*, Statist. Science, 1986, Vol. 1, No. 3 , pp. (297-318)

[44] K.Cho, M. Han,C. Sluder, R. Wagner,G. Lilik, *Experimental Investigation of the Effects of Fuel Characteristics on High Efficiency Clean Combustion in a Light-Duty Diesel Engine* SAE Technical Paper 2009-01-2669, 2009

[45] W. De Ojeda, T. Bulicz, X. Han, M. Zheng, F. Cornforth, *Impact of Fuel Properties on Diesel Low Temperature Combustion* SAE Technical Paper 2011-01-0329, 2011

[46] C. Dumitrescu, W. Neill, H. Guo, V. Hosseini, W. Chippior, *Fuel Property Effects on PCCI Combustion in a Heavy-Duty Diesel Engine* ASME 2010 ICE Conference ICEF2010-35194, San Antonio, TX, 2010

[47] D.C.Montgomery, *Design and analysis of experiment*. 7ed, New York, John Wiley & Sons, Inc., 2009

[48] B.S. Higgins, D.L. Siebers, *Measurement of the flame lift-off location in DI diesel sprays using OH chemiluminescence*. SAE Technical Paper 2001-01-0918, 2001.

[49] D. Andrews, A. Herzberg *A Collection of Problems from Many Fields for the Student and Research Worker* Springer-Verlag, New York, 1985

[50] L. Breiman, J.Friedman, R. Olshen, C. Stone, *Classification and Regression Trees* Wadsworth, Belmont, CA, 1984

[51] P. Craven, G. Whaba, *Smoothing noisy data with spline functions. Estimating the correct degree of smoothing by the method of generalized cross-validation*, Numerische Methematik 1979

[52] C. deBoor, *A Practical Guide to Splines*, Springer-Verlag, New York, 1978

[53] D. Donoho, I. Johnstone *Projection-based approximation, and a duality with kernel methods* Ann. Statist, 1989,Vol. 17, No. 1, pp. (58-106)

[54] B. Efron, *Estimating the error rate of a prediction rule: Improvement on cross-validation* J. Amer. Statist. Assoc 1983 , Vol. 78, No. 382., pp. (316-331)

[55] J. Friedman *A tree-structured approach to nonparametric multiple regression*, Smoothing Techniques for Curve Estimation, Springer-Verlag, New York, 1979

[56] J. Friedman *Classification and multiple response regression through projection pursuit* Department of Statistics, Stanford University, Technical report, 1985

[57] J. Friedman *Fitting functions to noisy data in high dimensions* Computer Science and Statistics. 1988

[58] B. Silverman, *Some aspects of the spline smoothing approach to non-parametric regression curve fitting*, J. Roy. Statist. Soc. 1985, Vol. 47, No. 1, pp. (1-52)

[59] C. Stone, *Consistent nonparametric regression* Ann. Statist. 1977, Vol. 5, No. 4, pp. (595-620)

[60] Z Win, R P Gakkhar, S C Jain, M Bhattacharya, *Investigation of diesel engine operating and injection system parameters for low noise, emissions, and fuel consumption using response surface methodology.*Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering 2005, Vol. 219, No. 10, pp. (1237-1251)

[61] K. Eppring, S. Aceves, R. Bechtold, J. Dec, *The Potential of HCCI Combustion for High Efficiency and Low Emissions*, SAE Technical Paper 2002-01-1923, 2002.

[62] M. Musculus, *On the Correlation between NOx Emissions and the Diesel Premixed Burn*, SAE Technical Paper 2004-01-1401, 2004

[63] P. Carlucci, A. Ficarella, D. Laforgia, *Effects of Pilot Injection Parameters on Combustion for Common Rail Diesel Engines*SAE Technical Paper 2003-01-0700, 2003

[64] M. Musculus, T. Lachaux, L. Pickett, C. Idicheria, *End-of-Injection Over-Mixing and Unburned Hydrocarbon Emissions in Low-Temperature-Combustion Diesel Engines* SAE Technical Paper 2007-01-0907, 2007

[65] A. Weall, N. Collings, *Highly Homogenous Compression Ignition in a Direct Injection Diesel Engine Fuelled with Diesel and Biodiesel* SAE Technical Paper 2007-01-2020, 2007

[66] U. Wagner, R. Anca, A. Velji, U. Spicher, *An Experimental Study of Homogeneous Charge Compression Ignition (HCCI) with Various Compression Ratios, Intake Air Temperatures and Fuels with Port and Direct Fuel Injection* SAE Technical Paper 2003-01-229, 2003

[67] R. Buchwald, M. Brauer, A. Blechstein, A. Sommer,J. Kahrstedt, *Adaption of Injection System Parameters to Homogeneous Diesel Combustion* SAE Technical Paper 2004-01-0936, 2004