

**Alma Mater Studiorum – Università di Bologna**

**DOTTORATO DI RICERCA IN**  
**Meccanica e Scienze Avanzate dell’Ingegneria**

**Ciclo XXXIII**

**Settore Concorsuale: 09/C1**

**Settore Scientifico Disciplinare: ING-IND/08**

**Development and testing of innovative methodologies for modelling and control of normal and knocking combustion and implementation of novel rapid prototyping solutions**

**Presentata da: Ing. Alessandro Brusa**

**Coordinatore Dottorato**

**Supervisore**

**Prof. Ing. Marco Carricato**

**Prof. Ing. Nicolò Cavina**

**Esame finale anno 2021**

# Abstract

The thesis work deals with topics that led to the development of innovative control-oriented models and control algorithms for modern gasoline engines. Knock in boosted spark ignition engines is the widest topic discussed in this document because it remains one of the most limiting factors for maximizing combustion efficiency in this kind of engine.

First chapter is thus focused on knock and a wide literature review is proposed to summarize the preliminary knowledge that even represents the background and the reference for discussed activities. Most relevant results achieved during PhD course in the field of knock modelling and control are then presented, describing every control-oriented model that led to the development of an adaptive model-based combustion control system. The complete controller has been developed in the context of the collaboration with Ferrari GT and it allowed to completely redefine the knock intensity evaluation as well as the combustion phase control. Results of the experimental validation process at the test bench are then shown. The last part of the first chapter deals with the development and calibration of an algorithm for the combustion phase estimation from the accelerometric signal. The second chapter is focused on the activity related to a prototyping Port Water Injection system that has been developed and tested on a turbocharged spark ignition engine, within the collaboration with Magneti Marelli. Such system and the effects of injected water on the combustion process were then modeled in a 1-D simulation environment (GT Power). A model-based control system has been developed and experimentally validated with a Rapid Control Prototyping system to manage both the injected water mass and the spark advance with the final aim to reach a target combustion phase and the maximum combustion efficiency.

Third chapter shows the development and validation of a control-oriented model for the real-time calculation of exhaust gas temperature that represents another important limitation to the performance increase in modern boosted engines. Indeed, modelling of exhaust gas temperature and thermocouple behavior are themes that play a key role in the optimization of combustion and catalyst efficiency.

The Appendix of this document describes the realization of a custom real time device. Such target machine has been built during the last two years of PhD course and it represents an extremely cheap solution to exploit high computational power of cooled CPU of desktop PC to test advanced control algorithm in real time. The device can be connected to the ECU and other systems of the engine test bench via CAN or with an analog and digital input/output board and some engine control parameters and signals can be directly controlled, logged, and analyzed through the execution of a custom RT application.

Topics described in these chapters may look substantially different from each other. Nevertheless, they are connected from a common thread that is the increase of combustion efficiency in modern SI engines. Indeed, presented activities are focused on the development of solutions for a more accurate management of combustion process in gasoline engines.

# Contents

Index of figures .....	7
1. Knock in Modern ICEs: Innovative Modelling and Control Techniques .....	20
1.1. Introduction.....	21
1.1.1. Source of knocking combustions.....	23
1.1.2. Factors that influence engine knock tendency .....	24
1.2. Literature Review .....	25
1.2.1. Knock Physics .....	25
1.2.1.1. Knock Mechanism .....	25
1.2.1.2. Knock Intensity Indexes.....	27
1.2.1.3. Knock, Pre-ignition and Super-knock.....	28
1.2.1.4. Pressure oscillation analysis.....	29
1.2.2. Knock Consequences.....	31
1.2.2.1. Knock-Induced Damage .....	31
1.2.2.2. Thermo-Mechanical Stress Estimation .....	33
1.2.2.3. Piston Damage Influent Parameters .....	34
1.2.2.4. Piston Temperature Measurement.....	35
1.2.2.5. Piston Damage Related Indexes .....	35
1.2.2.6. Conclusions About Presented Literature .....	36
1.2.3. Knock Modelling .....	36
1.2.3.1. Knock Process Characterization .....	37
1.2.3.2. Knock Modelling Approaches .....	38
1.2.3.3. 0-D Knock Simulation .....	41
1.2.4. Knock Control .....	41
1.2.4.1. Knock Control-Oriented Strategies .....	42
1.3. Control-Oriented Modelling .....	47
1.3.1. Control-Oriented Knock-Induced Piston Damage Model.....	49
1.3.2. Control-Oriented Analytical Knock Intensity Model .....	63
1.3.2.1. Knock Model Development and Calibration for the 4-cylinder engine .....	64
1.3.2.2. Knock Model Calibration for the 8-cylinder engine .....	74
1.3.2.3. Knock Model Validation and Its Implementation in GT Power Environment....	78
1.3.3. Analytical Modelling of Combustion Indexes.....	85
1.3.3.1. SA-to-MFB50 model .....	89
1.3.3.2. MFB50-to-PMAX Model .....	94

1.3.3.3.	MFB50-to-IMEP Model .....	97
1.3.3.4.	Knock Model .....	98
1.3.3.5.	Standard Deviation Model .....	99
1.3.3.6.	ANESIM .....	100
1.4.	Knock Control .....	106
1.4.1.	Adaptive piston damage-based combustion controller .....	106
1.4.1.1.	Experimental Setup .....	109
1.4.1.2.	Implemented Models .....	110
1.4.1.3.	Piston Damage Model.....	111
1.4.1.4.	Analytical Knock Model .....	114
1.4.1.5.	Analytical PMAX model.....	117
1.4.1.6.	Analytical MFB50 model .....	117
1.4.1.7.	Complete Combustion Controller .....	120
1.4.1.8.	Combustion Controller Validation.....	121
1.4.1.9.	Results and Discussion .....	121
1.4.1.10.	Conclusions and Proposals .....	128
1.4.2.	Innovative Solutions for The On-Board Knock Intensity and Combustion Phase Estimation.....	129
1.4.2.1.	MFB50 Estimation by Accelerometric Signal.....	129
1.4.2.2.	Signal analysis .....	130
1.4.2.3.	Signal processing algorithm and calibration .....	132
1.4.2.4.	Conclusions .....	134
	References .....	135
2.	Water Injection Modelling and Control .....	140
2.1.	Introduction and Motivations.....	141
2.2.	Port Water Injection System 1-D Modelling .....	143
2.2.1.	Water Injection and Knock Model.....	143
2.2.1.1.	Model Layout .....	143
2.2.1.2.	Water Injection Modelling Parameters.....	144
2.2.1.3.	GT Power Knock Model .....	147
2.2.1.4.	Modelling Results .....	150
2.2.2.	Experimental Tests and 1-D Model Validation .....	153
2.2.2.1.	Experimental Setup .....	153
2.2.3.	Conclusions .....	157

2.3.	Development and Software-in-the-Loop Validation of a Model-based Controller for a Port Water Injection System .....	159
2.3.1.	Experimental Campaign.....	159
2.3.2.	WI Combustion Model.....	161
2.3.2.1.	Polynomial Method.....	161
2.3.2.2.	Effect Separation Method 1-D.....	163
2.3.2.3.	Effect Separation Method 2-D.....	165
2.3.2.4.	Comparison by Correlation Coefficient.....	166
2.3.3.	WI-based Combustion Control.....	166
2.3.3.1.	CL on MAPO98.....	166
2.3.3.2.	CL on MAPO98 and MFB50.....	170
2.3.4.	Software in the Loop Control Validation .....	171
2.3.4.1.	GT Power Fast Running Model .....	171
2.3.4.2.	Simulation Results .....	172
2.3.5.	Conclusions .....	178
2.4.	Experimental Validation of Water Injection Control System.....	180
2.4.1.	Experimental Setup .....	180
2.4.1.1.	Prototypal WI engine Setup .....	180
2.4.1.2.	Rapid Control Prototyping Environment .....	180
2.4.2.	Controller Overview .....	181
2.4.2.1.	WI Combustion Model.....	182
2.4.2.2.	r/SA Correction Management.....	183
2.4.3.	Calibration Campaign.....	185
2.4.4.	Performance of WI Combustion Control System.....	186
2.4.4.1.	Open Loop on MFB50 .....	186
2.4.4.2.	Closed Loop on MFB50.....	186
2.4.4.3.	Closed Loop on MFB50 and MAPO98.....	187
2.4.4.4.	Controller Response Under Transient Conditions.....	190
2.4.4.5.	Fuel Consumption Implications .....	191
2.4.5.	Conclusions .....	192
	References.....	193
3.	Exhaust Gas Temperature Modelling.....	194
3.1.	Introduction and Literature Review .....	195
3.2.	Experimental Campaign.....	198

3.3.	Analytical Model .....	199
3.3.1.	Reference Temperature Model .....	200
3.3.2.	MFB50 Dependency Model.....	202
3.3.3.	Lambda Dependency Model.....	206
3.4.	Model Validation Under Steady-State Conditions .....	211
3.4.1.	TC Dynamics Model.....	211
3.5.	Model Validation under Transient Conditions .....	214
3.5.1.	Results and Discussion .....	214
3.6.	Conclusions .....	218
	References .....	219
4.	Conclusions .....	221
5.	Appendix: Simulink Real Time Target Machine Development.....	223
5.1.	Introduction and Motivations.....	224
5.2.	xPC Driver Authoring Tool.....	227
5.3.	Real-Time Software .....	231
5.3.1.	Real-Time Kernel .....	231
5.3.1.1.	Target Boot Disk .....	231
5.3.1.2.	Target PC BIOS .....	231
5.3.1.3.	Real-Time Kernel Features .....	232
5.3.1.4.	Target PC Heap.....	232
5.3.2.	Real-Time Application .....	232
5.3.3.	Signal Acquisition.....	233
5.3.4.	Parameter Tuning.....	233
5.4.	Real-Time Hardware.....	235
5.4.1.	Host and Target PCs.....	235
5.4.2.	Host-Target Connection.....	235
5.4.3.	I/O Driver Support .....	236
5.5.	Rapid Control Prototyping Process .....	238
5.6.	Embedded Application Development.....	240
5.7.	User Remote Interaction Options.....	242
5.7.1.	xPC Target Remote Control Tool .....	242
5.7.2.	MATLAB Command-Line Interface .....	243
5.7.3.	Simulink External Mode .....	243
5.7.4.	Simulink with xPC Target Scope Blocks .....	244

5.7.5.	Target PC Command-Line Interface .....	244
5.7.6.	Web Browser Interface.....	244
5.7.7.	Custom GUI.....	245
5.7.7.1.	Custom GUI with Simulink.....	245
5.7.7.2.	Custom GUI with xPC Target Application Programming Interface (API) .....	245
5.7.8.	XCP Communication Protocol.....	245
5.7.8.1.	Universal Measurement and Calibration Protocol (XCP).....	245
5.7.8.2.	Configuring the model for A2L file generation.....	246
5.8.	Custom Device Configuration .....	251
5.9.	Conclusions .....	255
	References .....	256

## Index of figures

Figure 1.1: Scheme of efficiency chain for spark ignition engines (picture taken from [1.1]).....	21
Figure 1.2: Specific fuel consumption trend on the engine operating field (picture taken from [1.1]) .....	22
Figure 1.3: Mixture auto-ignition as knock source (picture taken from [1.1]) .....	23
Figure 1.4: $\epsilon$ - $\xi$ peninsula. Circles indicate area of the field in which knock occurs and higher black fill indicates increasing severity of knock, from no knock to super-knock (picture taken from [1.15])......	26
Figure 1.5: Main combustion indexes for a knocking event (picture from [1.19]).....	27
Figure 1.6: Normal combustion, conventional knock and super knock pressure curves (picture from [1.19]). .....	28
Figure 1.7: Effect of engine load, for fixed intake air temperature (20°C) and different spark timings (adapted from [1.19]).....	29
Figure 1.8: Shapes of first 5 modes of a cylindrical cavity (adapted from [1.19]).....	30
Figure 1.9: Maximum net energy for knocking combustions. Blue points represent the experimental data, while the red dashed line is the fitting function (not knocking events are excluded).....	31
Figure 1.10: Variation of knock intensity PDF, moving the spark timing (from [1.38]). .....	37
Figure 1.11: Binomial distribution of knocking events observed in 1000 cycles on an engine point characterized by a knock probability of 0.01 (from [1.38]). .....	38
Figure 1.12: (a) Response of a standard controller for different initial conditions and (b) related spark advance distribution (adapted from [1.38]). .....	43
Figure 1.13: (a) Response of a Likelihood controller for different initial conditions and (b) related spark advance distribution (adapted from [1.38]). .....	44
Figure 1.14: Knock probability w.r.t. spark advance (from [1.38]). .....	45
Figure 1.15: (a) Knock probability with two different knock threshold and (b) response of a standard controller with an optimized threshold (adapted from [1.38]). .....	45
Figure 1.16: Three engine cycles with the same thermodynamic conditions at knock onset and different knock intensity (from [1.75]).....	47
Figure 1.17: (a) Knock intensities displayed with corresponding unburned mass fraction at the knock onset and (b) PDFs for different bin width of unburned mass fraction (adapted from [1.75]). .....	48
Figure 1.18: Hardness-temperature-time (HTT) curves for the aluminum alloy .....	49
Figure 1.19: Functional layout of communication network.....	50
Figure 1.20: Description of piston damage model with logic blocks. ....	51
Figure 1.21: (a) Location of micro-Vickers hardness tests on pistons crown: measurements 1-5 have been acquired in the intake side, while measurements 6-10 in the exhaust side. (b) Sites of interest for roughness measurements, indicated by red and black ellipses in the exhaust valve reliefs. ....	51

Figure 1.22: (a) Normalized knock level (average MAPO 99.5%) and (b) maximum pressure level (average Pmax 90%) depending on the average applied $\Delta SA$ , w.r.t. the mapped value. Numbers indicate the cylinder. ....	52
Figure 1.23: (a) Temperature values (normalized w.r.t. the maximum value) for pistons belonging to both set 1 (#1-4) and set 2 (#5-8). Temperatures have been calculated starting from the average residual hardness values and the HTT curves characteristic of this specific piston alloy. (b) Mean piston temperature (normalized w.r.t. the maximum value) depending on in-cylinder condition in terms of knock level and maximum pressure level. ....	53
Figure 1.24: (a) Comparison between roughness parameter RzD and erosion level, for all tested pistons. (b) Erosion damage level depending on exhaust side piston temperature and knock level. For each piston, assessed damage level is reported in the square, which size and color is coherent with the damage (low damage level $\rightarrow$ small blue square, high damage level $\rightarrow$ big red square). Dashed lines represent constant damage levels. ....	54
Figure 1.25: Measured hardness field within the intake and the exhaust side for 3 pistons. ....	55
Figure 1.26: Hardness measurements symmetry, used to average correspondent values. ....	55
Figure 1.27: Qualitative oil jets position and estimation of the measurement point distance from the oil-impinged surface. ....	56
Figure 1.28: Schematic representation of the heat transfer problem through thermal resistances series. ....	58
Figure 1.29: Convective coefficient estimations considering 4 different temperature measurements, for different engine speeds. ....	59
Figure 1.30: Heat transfer enhancer depending on knock intensity. ....	60
Figure 1.31: Calibration process for the piston temperature model. ....	60
Figure 1.32: Material damage simplified model. ....	61
Figure 1.33: Damage model calibration process. ....	61
Figure 1.34: (a) MAPO threshold as a function of normalized piston temperature. (b) The corresponding cumulated modelled damage is compared with the erosion level, for all the pistons. ....	62
Figure 1.35: MAPO sorted values for fixed operating conditions. Each cylinder records different MAPO values. ....	65
Figure 1.36: Mean MAPO PFs for different SA. ....	65
Figure 1.37: Normalized mean MAPO PFs with respect to 95th MAPO percentile (MAPO95) and the mode probability (mode-prob). ....	66
Figure 1.38: MAPO98 and MAPO50 for some tested engine points. Referring to Table 1.4, the legend provides the number of the engine point and the lambda value (fuel RON=98). ....	66
Figure 1.39: MAPO98 as a function of maximum in-cylinder pressure for some tested engine points. ....	67
Figure 1.40: MAPO98 as a function of CA50 for some tested engine points. ....	68
Figure 1.41: MAPO98 as a function of CA1090 for some tested engine points. ....	68

Figure 1.42: Correlation between scaled MAPO98 (M98) and scaled MAPO50 (M50) with ESm equal to $8.3e-5$ .....	70
Figure 1.43: Experimental values of MAPO98 curves reduced of $RPM * ESm$ .....	71
Figure 1.44: m98 map compared with experimental data (M98, red circles).....	71
Figure 1.45: MAPO98 absolute errors for engine points collected in the m98 map.....	71
Figure 1.46: m50 map compared with experimental points (M50, green circles).....	72
Figure 1.47: MAPO50 errors for engine points collected in the m50 map.....	73
Figure 1.48: M98 curves for fixed operating conditions and for different lambda values.....	73
Figure 1.49: Experimental (M98) and calculated (m98) MAPO98 for three different engine points and lambda values.....	74
Figure 1.50: Correlation between scaled MAPO98 (M98) and scaled MAPO50 (M50) with ESm equal to $9.4e-5$ .....	75
Figure 1.51: (a) m98 and (b) m50 function compared with experimental points (red and green circles, respectively). .....	76
Figure 1.52: Error between calculated and modelled MAPO98 (a) and MAPO95 (b). .....	76
Figure 1.53: m98 function compared with experimental value (yellow points) for RON95 fuel. ....	77
Figure 1.54: Knock tendency for 8-cylinder engine with different intake air temperature for fixed operating conditions (3000 RPM, 1300 mbar, lambda 1 and RON 95 gasoline). .....	77
Figure 1.55: Superposed calculated and experimental cumulative MAPO distributions for two different engine points.....	80
Figure 1.56: 1-D engine model used for the TPA. Red circles highlight the End-Environment template in which intake and exhaust pressure vector have been uploaded. Green circles highlight Switch template that increases the runner friction to damp undesirable pressure oscillations when the intake and exhaust valves are closed. ....	81
Figure 1.57: GT Power model used for combustion parameters optimization.....	81
Figure 1.58: Layout of User Knock Index (UKI) template. In mapo98 and mapo50 blocks, Equations 1.34 and 1.35 have been implemented, while Equations 1.50 and 1.51 have been introduced in $\mu$ and $\sigma$ blocks respectively.....	82
Figure 1.59: Engine model with UKI template, in which Equations 1.34, 1.35, 1.50 and 1.51 have been implemented. Sensors provide STAM and Pmax, while lambda and RPM are set as inputs in the template properties window. ....	82
Figure 1.60: Comparison between experimental, calculated with the Knock Model (the inputs are the experimental data) and calculated with the UKI Template (the inputs are the GT model outputs) MAPO cumulative probability distribution, for engine point 2250 RPM, 1210. ....	83
Figure 1.61: Comparison between experimental, calculated with the Knock Model (the inputs are the experimental data) and calculated with the UKI Template (the inputs are the GT model outputs) MAPO cumulative probability distribution, for engine point 3500 RPM, 1200. ....	84

Figure 1.62: Comparison between experimental, calculated with the Knock Model (the inputs are the experimental data) and calculated with the UKI Template (the inputs are the GT model outputs) MAPO cumulative probability distribution, for engine point 3500 RPM, 1810. ....	84
Figure 1.63: Comparison between experimental, calculated with the Knock Model (the inputs are the experimental data) and calculated with the UKI Template (the inputs are the GT model outputs) MAPO cumulative probability distribution, for engine point 4500 RPM, 1110. ....	84
Figure 1.64: Comparison between experimental, calculated with the Knock Model (the inputs are the experimental data) and calculated with the UKI Template (the inputs are the GT model outputs) MAPO cumulative probability distribution, for engine point 4500 RPM, 1850. ....	85
Figure 1.65: A total of 60 tested engine points on the Speed-Torque plane. For each point a spark sweep was performed. The parameters are normalized with respect to their maximum value. ....	87
Figure 1.66: ANESIM layout. Load and speed (RPM) are the two inputs of the simulator. The orange boxes are the SA and lambda maps, the yellow one represents the implementation of the MFB50 STD model, and the violet box identifies the calculation of the Normal PDF and the extrapolation of a related random value. ....	88
Figure 1.67: Experimental MFB50-SA points recorded for an engine speed equal to 40%. Each parabolic function corresponds to a different load, which increases from 20% (on the right) up to 80% (on the left), with steps of 20%. The blue points represent the experimental data, the red lines display the parabolic fitting. The SA and the MFB50 are expressed as normalized 'STEP' and 'SPACE' respectively. ....	89
Figure 1.68: Resulting surface for parameter A. Red markers show the experimental data used for the fitting of the surface. ....	92
Figure 1.69: Resulting surface for parameter B. Green markers show the experimental data used for the fitting of the surface. ....	92
Figure 1.70: Resulting surface for parameter C. Yellow markers show the experimental data used for the fitting of the surface. ....	93
Figure 1.71: R2 correlation coefficient corresponding to fittings with a different number of experimental points used for model calibration. The percentage values indicate the number of points used for fitting respect to the entire data set. 100% corresponds to 60 engine points (i.e. 60 spark sweeps), 20% to 12 engine points. ....	93
Figure 1.72: Error of the MFB50 model when it is calibrated with 20% of tested engine points (i.e. spark sweeps). The black dashed lines include 92% of data. ....	94
Figure 1.73: Pmax model. The surface represents the graphical trend of the analytical function and the red circles correspond to the experimental data. The MFB50 is expressed as normalized 'SPACE' ....	96
Figure 1.74: Correlation coefficient corresponding to fittings with a different number of experimental points used for the Pmax model calibration. The percentage values indicate the percentage of engine points included in the optimization of the model coefficients. 100% corresponds to 60 engine points (i.e. 60 spark sweeps), 20% to 12 engine points. ....	96
Figure 1.75: Errors of the Pmax model when it is calibrated with 20% of tested engine points (i.e. spark sweeps). The black dashed lines contain 92% of data. ....	96

Figure 1.76: IMEP model. The surface represents the graphical trend of the analytical function and the red circles correspond to the experimental data. The MFB50 is expressed as normalized 'SPACE' .....	97
Figure 1.77: Correlation coefficient corresponding to fittings with a different number of experimental points used for the IMEP model calibration. The percentage values indicate the percentage of engine points included in the optimization of the model coefficients. 100% corresponds to 60 engine points (i.e. 60 spark sweeps), 20% to 12 engine points.....	98
Figure 1.78: Error of the IMEP model when it is calibrated with 20% of tested engine points (i.e. spark sweeps). The black dashed lines include 92% of data. ....	98
Figure 1.79: Correlation coefficient corresponding to fittings with a different number of experimental points used for the knock model calibration. The percentage values indicate the percentage of engine points included in the optimization of the model coefficient. 100% corresponds to 60 points (i.e. 60 spark sweeps), 20% to 12 points. MAPO95 indicates the 95th MAPO percentile. ....	99
Figure 1.80: Error of the knock model when it is calibrated with 20% of tested engine points (i.e. spark sweeps). The black dashed lines include 92% of data. ....	99
Figure 1.81: MFB50 STD model. The surface represents the graphical trend of the analytical function (saturated at the minimum experimental value) and the blue circles are the experimental data. The MFB50 is expressed as normalized 'SPACE' .....	100
Figure 1.82: Calibration process step scheme. The functional layout of the Simulink model is displayed in Figure 1.66.....	101
Figure 1.83: Speed and load profiles for validation.....	102
Figure 1.84: Engine and ANESIM combustion indexes when the same speed and load profiles are used as inputs. This test was performed with the first SA map. Only 4 cylinders are included for clarity. The MFB50 is expressed as normalized 'SPACE' .....	103
Figure 1.85: Error made by ANESIM during the simulation with the first SA map. Each point represents the difference between the experimental moving average of the mean cylinder and the calculated mean value of the given combustion index, which is updated after each combustion. For MAPO95 a buffer of 400 elements was used to calculate the 95th percentile.....	103
Figure 1.86: Engine and ANESIM combustion indexes when the same RPM and load profiles are used as inputs. This test was performed with the second SA map. Only 4 cylinders are included for clarity. The MFB50 is expressed as normalized 'SPACE' .....	104
Figure 1.87: Errors made by ANESIM during the simulation with the second SA map. Each point represents the difference between the experimental moving average of the mean cylinder and the calculated mean value of the given combustion index, which is updated after each combustion. For the MAPO95 a buffer of 400 elements was used to calculate the 95th percentile. ....	104
Figure 1.88: Functional layout of communication loop.....	110
Figure 1.89: Controller layout, including the calculation of target MAPO% and the reversed analytical knock model. ....	112
Figure 1.90: MAPO threshold as a function of piston temperature.....	112

Figure 1.91: Damage per cycle associated to the MAPO distribution for the given MAPO threshold. ....	113
Figure 1.92: Block scheme of the algorithm used to find the MAPO PDF that guarantees the target cyclic piston damage. ....	113
Figure 1.93: Normalized and offset MAPO98% curves for fixed operating conditions and for two fuel RON values. ....	115
Figure 1.94: Controller layout, including the self-learning algorithm of the fuel RON value. ....	117
Figure 1.95: Controller layout, including the reversed P <sub>MAX</sub> model. ....	117
Figure 1.96: Controller layout, including the MFB50 model. ....	118
Figure 1.97: Experimental $\Delta$ SA-MFB50 points recorded for fixed engine speed and load. The MFB50 is normalized w.r.t. the maximum value. ....	118
Figure 1.98: Controller layout, including the adaptive algorithm for the $\Delta$ SA-MFB50 analytical equation updating. ....	120
Figure 1.99: Complete controller scheme. Text in red indicates the combustion indexes estimated from the system for the on-board in-cylinder pressure trace sensing, i.e. MAPO and MFB50. ....	121
Figure 1.100: Normalized engine speed and load profiles related to results presented below. ....	122
Figure 1.101: Target and experimental values of MFB50, P <sub>MAX</sub> , MAPO and damage speed for cylinder 1. Third graph represents the spark timing correction of the closed loop chain and the final output of the controller. ....	123
Figure 1.102: Error between the target and the experimental MFB50 and P <sub>MAX</sub> indexes, for cylinder 1. ....	123
Figure 1.103: Results for cylinder 2, 3 and 4. ....	124
Figure 1.104: Error between the target and the experimental MFB50 and P <sub>MAX</sub> indexes, for cylinder 2, 3 and 4. ....	124
Figure 1.105: Calculated fuel RON. ....	124
Figure 1.106: Normalized engine speed and load profiles for the validation test of MFB50 model adaptive algorithm. ....	125
Figure 1.107: MFB50 and $\Delta$ SA for cylinder 1. Dotted line indicates the moment in which an offset on the SA map is applied. ....	125
Figure 1.108: MFB50 and $\Delta$ SA for cylinder 2, 3 and 4. Dotted line indicates the moment in which an offset on the SA map is applied. ....	126
Figure 1.109: Calculated fuel RON for validation test of fuel RON adaptive strategy. ....	126
Figure 1.110: Target and experimental values of MFB50, P <sub>MAX</sub> , MAPO and damage speed for cylinder 1. Third graph represents the spark timing correction of the closed loop chain and the final output of the controller, for fuel RON adaptive strategy validation test. ....	127
Figure 1.111: Error between the target and the experimental MFB50 and P <sub>MAX</sub> indexes recorded during fuel RON adaptive strategy validation test, for cylinder 1. ....	127
Figure 1.112: Results for cylinder 2, 3 and 4, for fuel RON adaptive strategy validation test. ....	128

Figure 1.113: Error between the target and the experimental MFB50 and P <sub>MAX</sub> indexes, for cylinder 2, 3 and 4, for fuel RON adaptive strategy validation test. ....	128
Figure 1.114: On top, derivative of in-cylinder pressure signal. In the middle and in the bottom, filtered accelerometric signal from the accelerometers (low-pass 1 kHz), 1500 RPM and 1.2 bar of intake manifold pressure. ....	130
Figure 1.115: Correlation between accelerometric signal peak and MFB50 measured with in-cylinder pressure sensor for the engine point at 2500 RPM and 1.1 bar of intake manifold pressure. Red and blue dots represent the correlation obtained without angular compensation (red=Acc12; blue Acc34). Green dots represent the correlation with angular delay compensation (MFB50 estimated). ....	131
Figure 1.116: Block diagram of accelerometric signal processing algorithm. ....	132
Figure 1.117: RMSE on the Cut-off frequency for engine point at 2500 RPM and 1.4 bar of intake manifold pressure. ....	133
Figure 2.1: Gt-Power model layout. Intake, cylinder, and exhaust sub-models are shown from left to right. Sensors that measure water mass toward valves and within cylinder are marked in violet. ....	144
Figure 2.2: Effect of $F_d$ variation for fixed water quantity. Top graph represents such effects on the in-cylinder pressure profile, while in the bottom graph the corresponding effect on the in-cylinder temperature is shown. ....	145
Figure 2.3: Simulated in-cylinder temperature during the compression stroke (25°CA BTDC) depending on ToE parameter. As in-cylinder evaporation duration (ToE) is longer, the charge temperature is lower. ....	146
Figure 2.4: Percentage increase of water consumption, for fixed maximum in-cylinder temperature, depending on the evaporation split between intake ducts ( $F_i$ ) and combustion chamber ( $F_d$ ). ....	146
Figure 2.5: Percentage increase of water consumption, for fixed maximum in-cylinder temperature, depending on the imposed duration of the in-cylinder evaporation. As the in-cylinder evaporation is longer (ToE), a lower value of $r$ is required. ....	147
Figure 2.6: Comparison between experimental MAPO and predicted User Knock Index, for SA=15°CA BTDC. ....	150
Figure 2.7: Comparison between experimental MAPO and predicted User Knock Index, for SA=21°CA BTDC. ....	150
Figure 2.8: CA50%MFB as a function of applied SA, for different water mass fractions. ....	151
Figure 2.9: Simulated IMEP curves for different injected water quantities. Big circles highlight maximum IMEP points. ....	152
Figure 2.10: Modelled exhaust temperature reduction as a function of CA50%MFB and water ratio. ....	152
Figure 2.11: Modelled knock intensity as a function of Spark Advance and water ratio. Water injection heavily reduces knock intensity. Cross markers highlight maximum IMEP spark advance. ....	153
Figure 2.12: Detail of the experimental setup: modified intake manifold (red arrow) and water rail (green arrow) are visible. Water injectors are highlighted with blue circles. ....	153

Figure 2.13: Modelled and measured in-cylinder pressure average cycle, for the same spark advance and different water quantities.....	154
Figure 2.14: Modelled and measured ROHR average cycle, for the same spark advance and different water quantities. ....	154
Figure 2.15: Effect of water injection on combustion phasing. Experimental results are superimposable to model predictions (Figure 2.8): water injection reduces combustion speed and the same CA50%MFB is reached with higher Spark Advance values, as water ratio increases.....	155
Figure 2.16: Measured IMEP as a function of CA50%MFB, for different water quantities. Larger markers identify maximum IMEP (and efficiency) points. ....	155
Figure 2.17: Measured exhaust temperature reduction, depending on water ratio. Bigger markers correspond to MBT. A reduction of about 50°C is achieved with 60% of water ratio. ....	156
Figure 2.18: Comparison between measured and modelled knock intensity. Bigger markers correspond to MBT (crosses represent GT ones). Modelled and experimental knock intensities are very superimposable.....	156
Figure 2.19: Measured Brake Specific Fuel Consumption trends for different water ratio values. Water injection does not affect maximum achievable efficiency: minimum BSFC value is independent on the water ratio. ....	157
Figure 2.20: Parabolic fitting of the SA trend with respect to MFB50, for the engine point characterized by NL=1.2 and RPM=2500, and for different water-to-fuel mass ratios. It is an example of the parabolic trend that links the SA to the MFB50 for fixed operating conditions. ....	161
Figure 2.21: Fitting surfaces of the parabolic coefficients in (RPM, NL) domain, for r=0.....	162
Figure 2.22: Fitting of surface coefficients by linear function.....	163
Figure 2.23: Combustion Model with Polynomial Method.....	163
Figure 2.24: The calculated surface of parameter a, for r=0.5.....	164
Figure 2.25: Normalized curve Ka.....	164
Figure 2.26: Combustion model with Separation Effect 1-D.....	165
Figure 2.27: Average surface Ka on NL-r domain (left) and on RPM-r domain (right).....	165
Figure 2.28: Combustion model with Separation Effect 2-D.....	165
Figure 2.29: SA Maximum Brake Torque (MBT) determination procedure.....	167
Figure 2.30: R map determination and related SA MBT.....	168
Figure 2.31: SA determination, when r is saturated to 0.8.....	169
Figure 2.32: R map for the explored operative field.....	169
Figure 2.33: First version WICC layout.....	170
Figure 2.34: Second version WICC layout.....	171
Figure 2.35: FRM layout. The red circle highlights the Direct Water Injector, the violet one highlights the fuel one.....	172
Figure 2.36: Steady state simulation results for engine point 2500 RPM, NL 1.43. Five subplots show (from top to bottom): RPM-NL, cycle-to-cycle recorded MAPO, MAPO98 and the MAPO98	

threshold, cycle-to-cycle recorded MFB50, mean MFB50 and the corresponding target, the actuated  $r$  and the  $r$  correction calculated by CL chain, the actuated SA and the SA correction calculated by the CL chain..... 173

Figure 2.37: Steady state simulation results for engine point 3500 RPM, NL 1.37. Five subplots show (from top to bottom): RPM-NL cycle-to-cycle recorded MAPO, MAPO98 and the MAPO98 threshold, cycle-to-cycle recorded MFB50, mean MFB50 and the corresponding target, the actuated  $r$  and the  $r$  correction calculated by CL chain, the actuated SA and the SA correction calculated by the CL chain..... 173

Figure 2.38: Steady state simulation results for engine point 4500 RPM, NL 1.33. Five subplots show (from top to bottom): RPM-NL, cycle-to-cycle recorded MAPO, MAPO98 and the MAPO98 threshold, cycle-to-cycle recorded MFB50, mean MFB50 and the corresponding target, the actuated  $r$  and the  $r$  correction calculated by CL chain, the actuated SA and the SA correction calculated by the CL chain..... 174

Figure 2.39: Steady state simulation results for engine point 4500 RPM, NL 1.54. Five subplots show (from top to bottom): RPM-NL cycle-to-cycle recorded MAPO, MAPO98 and the MAPO98 threshold, cycle-to-cycle recorded MFB50, mean MFB50 and the corresponding target, the actuated  $r$  and the  $r$  correction calculated by CL chain, the actuated SA and the SA correction calculated by the CL chain..... 174

Figure 2.40: Transient simulation results. Five subplots show (from top to bottom): RPM-NL, cycle-to-cycle recorded MAPO, MAPO98 and the MAPO98 threshold, cycle-to-cycle recorded MFB50, mean MFB50 and the corresponding target, the actuated  $r$  and the  $r$  correction calculated by CL chain, the actuated SA and the SA correction calculated by the CL chain..... 175

Figure 2.41: Steady state simulation results for engine point 1500 RPM, NL 0.95 with the CL on MFB50. Five subplots show (from top to bottom): cycle-to-cycle recorded MAPO, MAPO98 and the MAPO98 threshold, cycle-to-cycle recorded MFB50, mean MFB50 and the corresponding target, the actuated  $r$  and the  $r$  correction calculated by CL chain, the actuated SA and the SA correction calculated by the CL chain. .... 176

Figure 2.42: Steady state simulation results for engine point 2500 RPM, NL 1.43 with the CL on MFB50. Five subplots show (from top to bottom): cycle-to-cycle recorded MAPO, MAPO98 and the MAPO98 threshold, cycle-to-cycle recorded MFB50, mean MFB50 and the corresponding target, the actuated  $r$  and the  $r$  correction calculated by CL chain, the actuated SA and the SA correction calculated by the CL chain. .... 176

Figure 2.43: Steady state simulation results for engine point 3500 RPM, NL 1.62 with the CL on MFB50. Five subplots show (from top to bottom): RPM-NL, cycle-to-cycle recorded MAPO, MAPO98 and the MAPO98 threshold, cycle-to-cycle recorded MFB50, mean MFB50 and the corresponding target, the actuated  $r$  and the  $r$  correction calculated by CL chain, the actuated SA and the SA correction calculated by the CL chain. .... 177

Figure 2.44: Steady state simulation results for engine point 4500 RPM, NL 1.3 with the CL on MFB50. Five subplots show (from top to bottom): RPM-NL cycle-to-cycle recorded MAPO, MAPO98 and the MAPO98 threshold, cycle-to-cycle recorded MFB50, mean MFB50 and the corresponding target, the actuated  $r$  and the  $r$  correction calculated by CL chain, the actuated SA and the SA correction calculated by the CL chain. .... 177

Figure 2.45: Transient simulation results with the CL on MFB50. Five subplots show (from top to bottom): RPM-NL, cycle-to-cycle recorded MAPO, MAPO98 and the MAPO98 threshold, cycle-to-

cycle recorded MFB50, mean MFB50 and the corresponding target, the actuated r and the r correction calculated by CL chain, the actuated SA and the SA correction calculated by the CL chain. ....	178
Figure 2.46: Port Water Injection System .....	180
Figure 2.47: Layout of Rapid Control Prototyping .....	181
Figure 2.48: D/O management for water injectors .....	181
Figure 2.49: Water Injection Combustion Controller lay-out. ....	182
Figure 2.50: Combustion Model with Polynomial Method. ....	183
Figure 2.51: r/SA management: Closed Loop on MAPO98. ....	184
Figure 2.52: r/SA management: Closed Loop on MAPO98 and Closed loop on MFB50 .....	184
Figure 2.53: Grid of calibration points. ....	185
Figure 2.54: Steady state engine point 3300 RPM, 1.25 NL, Cylinder 2. On top has been displayed the match between MFB50 measured and MFB50 target, in the middle the value of r actuated, at the bottom the SA as model output to achieve the MFB50 target. ....	186
Figure 2.55: Steady state engine point 2800 RPM, 1.25 NL, Cylinder 2. At time 0 controller starts in open loop (as seen in Figure 9). After second 13.5 the controller switches in closed loop mode and the PI correction dSA <sub>mfb</sub> compensates the error on MFB50. SA <sub>act</sub> represents the sum of SA <sub>byModel</sub> and dSA <sub>mfb</sub> .....	187
Figure 2.56: Steady state engine point 2700 RPM, 1.22 NL, Cylinder 1. At second 18 the controller switches from open loop to closed loop: dSA <sub>mfb</sub> starts to compensate SA model (as seen in Figure 2.55) and correction dR PID compensates R map (mapped value of r) to keep Mapo98 (knock Index) close to threshold (Mapo98 Thr). ....	188
Figure 2.57: Engine point 2700 RPM, 1.22 NL, Cylinder 4. The controller is in closed loop. After second 13 MFB50 target is moved to an advanced value to stimulate high knock tendency. After 2 seconds dR PID, attempting to compensate R <sub>act</sub> reaches saturation value (dR PID sat). After saturation, dSA <sub>mfb</sub> is limited to negative values and correction dSA <sub>knock</sub> starts to apply correction on SA <sub>act</sub> to delay MFB50.....	189
Figure 2.58: Engine point 3200 RPM, 1.3 NL, Cylinder 4. At time 0, the controller is in closed loop only for MFB50 and knock index exceeds the threshold value. After second 38 even loop on Mapo98 is closed.....	190
Figure 2.59: Transient test speed and load profiles. ....	190
Figure 2.60: Transient test controller performance, Cylinder 1.....	191
Figure 3.1: A total of 60 engine points on the Speed-Load range were tested for the calibration process. For each point a spark sweep was performed, and for 14 of these, a lambda sweep was carried out.....	198
Figure 3.2: Block-layout of the exhaust gas temperature model. The reference temperature is multiplied by two factors that depend on MFB50 and lambda values respectively. The MFB50 model from [3.11] allows to calculate the combustion phase without the need for indicating signals. ....	200

Figure 3.3: Trend of the reference temperatures ( $\lambda = \text{map}$ , $\text{MFB}_{50} = 2$ ) on the normalized RPM-Load range. The blue dots represent values corresponding to the $\text{MFB}_{50} = 2$ calculated through the interpolation of the experimental data, while the surface is the poly31 fitting function. ....	200
Figure 3.4: R2 index calculated with a different number of experimental points used for the model calibration. The percentage values indicate the amount of engine points used for fitting procedure with respect to the entire database. 100% corresponds to 60 engine points (i.e. 60 spark sweeps), 20% to 12. ....	202
Figure 3.5: Exhaust manifold gas temperature as a function of combustion phase ( $\text{MFB}_{50}$ ), for fixed engine speed and load. The blue dots and the red line represent the experimental values logged for different values of the SA and the fitting polynomial function respectively.....	203
Figure 3.6: Exhaust gas temperature is normalized with respect to the value in correspondence with the reference $\text{MFB}_{50}$ . Thus, the red line represents the correction function of the temperature for a given operating condition. ....	204
Figure 3.7: Numerical values of coefficient $p0_{\text{MFB}}$ identified for each spark-sweep. The black dashed line indicates the mean value, while the two red dashed lines represent the upper and lower limit of the rounded dispersion of data, which lie within $\pm 3\%$ .....	205
Figure 3.8: Numerical values of the coefficient $p1_{\text{MFB}}$ for each spark-sweep. The black dashed line indicates the mean value, while the two red dashed lines represent the upper and lower limit of the rounded dispersion of data, which lie within $\pm 30\%$ . ....	205
Figure 3.9: Absolute error for all the spark sweeps, given by the approximation of coefficients $p0$ and $p1$ with their corresponding mean value. The mean value of the error is equal to 0.43% and this result confirms the possibility of simplifying each fitting function with the average polynomial coefficients. ....	206
Figure 3.10: Red dots indicate the 14 engine points in which $\lambda$ sweeps were performed. The blue points represent the operating conditions for which a spark-sweep was carried out. ....	207
Figure 3.11: Exhaust gas temperatures were normalized with respect to the reference value. Blue dots indicate the normalized gas temperature values while the red line represents the corresponding fitting function.....	208
Figure 3.12: Numerical values of the coefficient $p0_{\text{LAMBDA}}$ for each $\lambda$ sweep. The black dashed line indicates the mean value, while the two red dashed lines represent the rounded limits of the data dispersion, which lie within $\pm 14\%$ . ....	209
Figure 3.13: Numerical values of the coefficient $p1_{\text{LAMBDA}}$ for each $\lambda$ sweep. The black dashed line indicates the mean value, while the two red dashed lines represent the rounded limits of the data dispersion, which lie within $\pm 10\%$ . ....	209
Figure 3.14: Absolute error for each $\lambda$ sweep given by approximation of coefficients $p0_{\text{LAMBDA}}$ and $p1_{\text{LAMBDA}}$ with the corresponding mean value. The mean error is equal to 1.11%: this result confirms the possibility to simplify each fitting function with the average polynomial.....	210
Figure 3.15: Error between measured and modelled steady-state exhaust gas temperatures. ....	211
Figure 3.16: Normalized TC temperature trend during a fast transient condition. ....	212
Figure 3.17: Simulink block scheme used to implement moving averages.....	213

Figure 3.18: Simulink blocks used for the sum of moving the average outputs. ....	213
Figure 3.19: The exhaust manifold temperature model was coupled with the TC dynamic model in order to reproduce the TC signal also under transient conditions.....	214
Figure 3.20: Normalized Engine speed, load, SA, and lambda profiles for test 1. ....	215
Figure 3.21: Performance of the complete analytical model under transient conditions (test 1). ...	215
Figure 3.22: Normalized Engine speed, load, SA, and lambda profiles for test 2. ....	216
Figure 3.23: Performance of the complete analytical model under transient conditions (test 2). ...	216
Figure 5.1: External mPCIe Ethernet board with Realtek 8125 controller. ....	227
Figure 5.2: Startup of xpcdrivertool. ....	227
Figure 5.3: Input Ports tab for R8125 ethernet driver. ....	228
Figure 5.4: Work Variables tab for R8125 ethernet driver. ....	228
Figure 5.5: Output Ports tab for R8125 ethernet driver. ....	229
Figure 5.6: Main tab for the C file template generation. ....	229
Figure 5.7: C file template for R8125 driver. ....	230
Figure 5.8: SRT Explorer with the R8125 ethernet driver available.....	230
Figure 5.9: Screen of launched RT kernel. ....	232
Figure 5.10: Example of instrumented panel developed with SRT Explorer.....	234
Figure 5.11: Visualizing root directories and loaded application on target PC with SRT Explorer. ....	234
Figure 5.12: TCP/IP network connection scheme between host and target PC. ....	236
Figure 5.13: Concurrent Execution panel. ....	238
Figure 5.14: Target boot mode set as Stand-Alone.....	240
Figure 5.15: The target PC is specified in model Configuration Parameters. ....	241
Figure 5.16: Target screen output for Stand-Alone mode.....	241
Figure 5.17: Master-Slave principle (from [4.4]).....	246
Figure 5.18: Simulink Signal configured for A2L file generation. ....	247
Figure 5.19: Simulink Parameter configured for A2L file generation. ....	248
Figure 5.20: Generation of A2L file when RT application is built. ....	248
Figure 5.21: INCA workspace with SRT device added via XCP.....	249
Figure 5.22: Memory pages for XCP connection. ....	249
Figure 5.23: INCA Experiment with SRT device signals and parameters.....	250
Figure 5.24: ASUS M3N78-VM motherboard. ....	251
Figure 5.25: Target device case: exploded view (a), upper case (b), lower case (c).....	252
Figure 5.26: (a) Softing CAN AC2 PCI board and (b) Contec AD12-16E. ....	252
Figure 5.27: Rear view of RT machine. ....	253

Figure 5.28: Top view of RT machine. .... 253  
Figure 5.29: Side view of RT machine..... 254  
Figure 5.30: 37pin DSUB break-out board..... 254  
Figure 5.31: CAN networks layout and connected devices..... 255

# 1. Knock in Modern ICEs: Innovative Modelling and Control Techniques

## 1.1. Introduction

Passenger vehicles will inevitably evolve over the next few decades. Engines and vehicle bodies will change because of increasingly stringent fuel consumption and anti-pollution regulations. New design choices, new types of fuel and new materials will be used to meet the more and more severe limits imposed by the new regulations and the requirements of the market. By the end of the current decade, the new internal combustion engines will be more efficient, with lower pollutant emissions, lighter, more reliable but also more expensive compared to those of nowadays. Spark-Ignition (SI) engines will be the predominant type of motorization that will equip both purely thermal propulsion and hybrid vehicles, due to the lower pollutant emissions that give them an advantage over diesel engines. As it is well known, spark-ignition engines are characterized by intrinsically inefficient load management, as this is achieved by throttling the air passage area to the intake manifold. This solution is primarily responsible for the high pumping losses that also determine the significant gap between petrol and diesel engines in terms of efficiency. Figure 1.1 summarizes the efficiency chain for a spark ignition engine, assuming the use of a stoichiometric mixture.

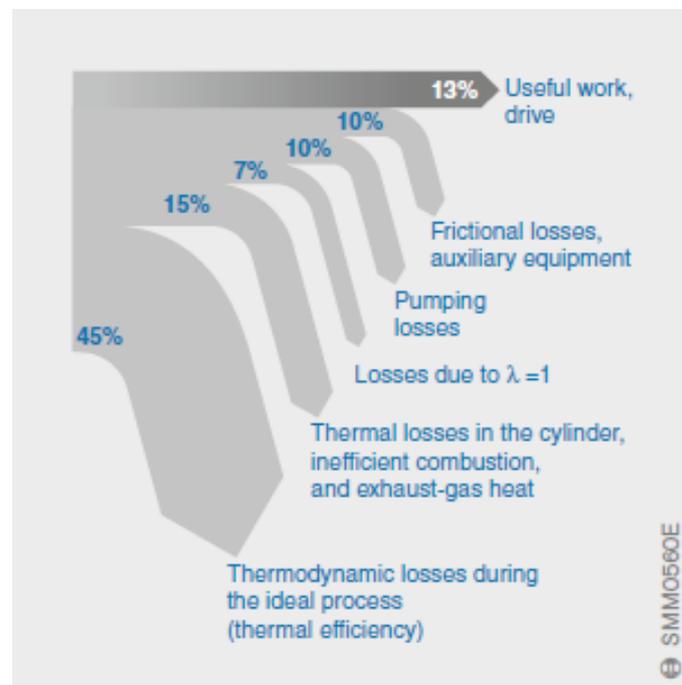


Figure 1.1: Scheme of efficiency chain for spark ignition engines (picture taken from [1.1])

The technological thrust of these days is mainly addressed to the use of new technologies and materials that allow to significantly increase the net obtainable work from the thermodynamic cycle (e.g. reducing wall heat losses, pumping and compression work, especially at medium-low loads) and above all by reducing the specific fuel consumption (defined as the grams of fuel required to produce the unit of work), by enabling the use of stoichiometric mixture in increasingly wider areas of the engine operating range, through the use of strategies for the application of stratified-charge and optimal ignition timing. The latter is certainly the most concrete control lever with which it is possible to manage combustion efficiency quickly and precisely. In fact, recent design choices aimed at reducing fuel consumption have pushed internal combustion engine manufacturers to reduce the displacement of all engine sizes and to adopt strategies that implement or suggest to the driver (based on the use of automatic or manual transmission respectively) the use of higher gears to move towards conditions characterized by the same power but lower specific fuel consumption (Figure 1.2).

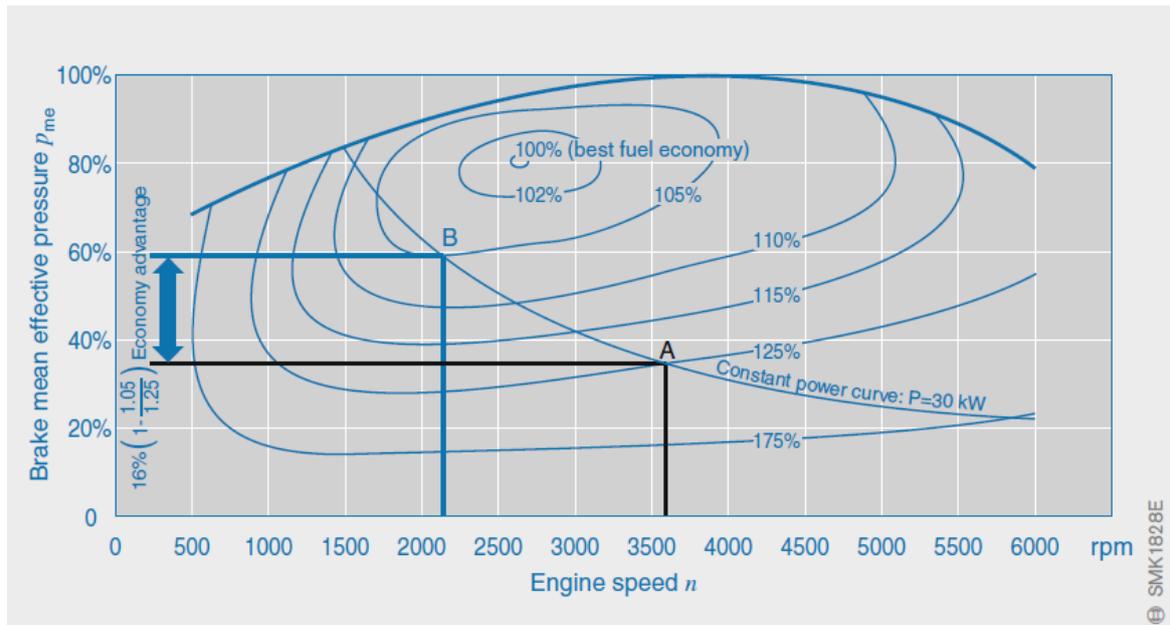


Figure 1.2: Specific fuel consumption trend on the engine operating field (picture taken from [1.1])

The solutions described above are called downsizing and down speeding respectively [1.1] and have made the development of turbocharged engines indispensable, in order to continue to guarantee products that are satisfactory also for users' requirements. These strategies have led to a significant increase in the average pressures in the combustion chamber per unit of displacement, increasing heat losses through the walls of the combustion chamber, thermal and mechanical stress on internal components and exhaust gas temperatures. These temperatures are a key parameter in the control systems of modern downsized turbocharged spark ignition engines, since exceeding the maximum inlet gas temperature of the turbine can permanently damage the turbocharger components. The most critical operating conditions in terms of high exhaust temperatures are those characterized by high engine speed and load. Under these conditions, the most common technique to reduce exhaust temperatures is to enrich the mixture to exploit the latent vaporization heat of the fraction that does not participate in combustion process to cool the exhaust gases. At the same time, the production of hot gases is of fundamental importance in the heating phase of the 3-way catalytic converter in order to reach the maximum conversion efficiency of the main pollutant categories as quickly as possible. The management of the ignition advance and therefore of the combustion efficiency allows to control the portion of energy that is converted into useful work, rather than lost by raising the temperature of the exhaust gases. This means that a precise and accurate control of the spark advance can have a double effect on the reduction of the specific consumption by maximizing the work that can be extracted from the thermodynamic cycle and reducing the exhaust temperatures in the meantime, thus avoiding an enrichment of the mixture. The ability to avoid mixture enrichment by limiting exhaust gas temperatures at high engine speed and load also creates an optimal operating environment for the catalytic converter over a wider area of the engine operating range. Ignition timing thus remains one of the control levers most impacting on overall performance for an automotive propulsion system. Among the factors that greatly restrict the possibility to apply optimal ignition timing over the entire engine operating range are pre-ignition and spontaneous knocking. This type of phenomenon is highly undesirable and is frequently accompanied by easily audible metallic noise, which is why it is identified as knock. Knocking combustion occurs when a portion of the mixture not yet reached by the flame front (characteristic of combustion in spark-ignition engines) ignites spontaneously. During this kind of combustion there is a very high peak in the heat release and consequently also very intense pressure oscillations that violently stress the

piston, the components installed on the head, the head gasket and the joints between piston and rod. Clearly, long periods of exposition to knock lead to erosion of the main elements of the combustion chamber until the engine is seized. It is therefore clear that a thorough knowledge of the physical mechanisms that govern knock, its harmful effects on the components of the combustion chamber, the ability to model it to accurately calculate its intensity and finally the possibility to effectively control it are key issues in research on modern spark-ignition engines. The research developed during the PhD course on these topics is the subject of this first chapter.

### 1.1.1. Source of knocking combustions

The Spark Advance (SA) is so called because it identifies the Crank Angle (CA) at which the spark is fired, and it is expressed in degrees before (or in *advance* w.r.t.) the Top Dead Center Firing (TDCF). The SA should typically take place before TDCF because it elapses several milliseconds before the mixture can ignite. This ignition lag depends mainly on the turbulence in the combustion chamber and therefore on engine speed. The flame front therefore extends with an ideally semi-spherical surface from the spark plug and expands into the entire combustion chamber. The mixture at this stage has already been compressed and the start of combustion further compresses and heats the mixture not yet reached by the flame front. As a result, some portions of the air/fuel mixture can reach temperatures high enough to induce spontaneous self-ignition (Figure 1.3).

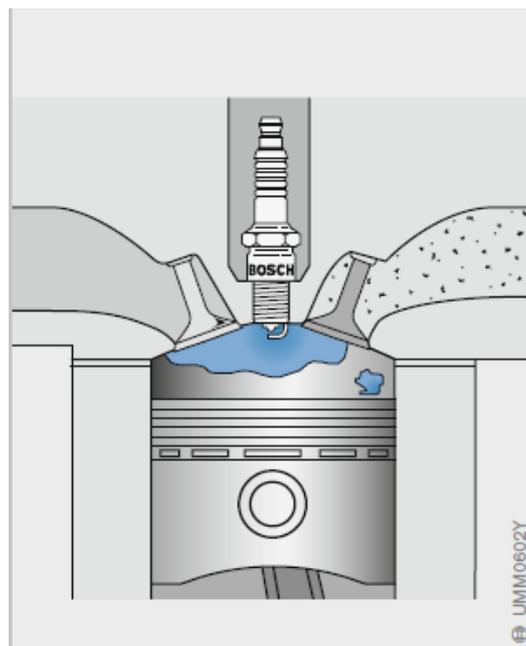


Figure 1.3: Mixture auto-ignition as knock source (picture taken from [1.1])

Typically, when this type of phenomenon occurs, it produces an additional flame front that starts from the ignition point with a propagation rate that is 10 to 100 times higher than that propagated by the spark plug. These very high frequency pressure oscillations propagate according to circular patterns from the core of the process to the outside. When these oscillations impact the walls of the combustion chamber (piston, head, cylinder) they generate the characteristic metallic noise mentioned above. According to the same process other self-igniting mixture cores can occur from the peripheral areas of the combustion chamber. Among the potential sources of knocking combustion is also the spark plug which, if subjected to several knocking cycles, can heat up abnormally, becoming a hot spot that triggers combustion. This is the case with pre-ignition, which

are particularly dangerous because, once ignited, it is no longer possible to control combustion phase by actuating the spark plug. The combustion therefore starts during the compression phase, generating very high pressure and temperature peaks (from 2 to 3 times higher than those achievable during controlled combustion) that quickly damage the chamber components but also the entire frame. In these conditions the engine can resist just a few cycles before breaking.

### 1.1.2. *Factors that influence engine knock tendency*

Among the control parameters that uniquely influence the detonating behavior of the engine is certainly the ignition timing. Increasing the *advance* with which combustion is triggered results in higher and higher pressure and temperature increases. The engine load, which inevitably increases when the torque demand rises, leads to high pressures and temperatures in the combustion chamber. It is also necessary to mention the engine speed: the higher the speed the faster the flame front, which therefore takes less time to reach all the areas of the combustion chamber. Fuel with a low percentage of isoctane provides limited knock resistance and therefore fuel quality is also a factor to consider. The design parameters include the compression ratio (CR), defined as the ratio between the displacement increased by the volume of the combustion chamber and the latter volume. Clearly a high compression ratio causes an increase in temperatures and pressures at the end of the compression stroke. Carbon deposits that accumulate on the pistons during life of the engine may slightly change this ratio over the years. From this point of view, engine ageing can also lead to a worsening of the piston seals on the cylinders, increasing oil leakage from the crankcase. Oil significantly lowers the antiknock power of the fuel and this is certainly one of the elements to consider when dealing with engine ageing. Similarly, even a poorly efficient cooling system leads to mixtures with a higher average temperature. The engine cooling system as well as the efficiency and geometrical characteristics of the oil-water exchangers play a key role in counteracting the excessively high temperatures of the intake mixture. Finally, the geometry of the combustion chamber, runners and intake manifold can positively or negatively influence knock. The reduction of turbulence in the chamber worsens the tendency of the engine to knock. Similarly, even an optimized intake geometry to increase turbulent and swirl motions may not be enough when coupled with a particularly complex and concave piston geometry. A piston with a surface made with the valve reliefs or a head with a particular geometry to privilege the compression ratio and the valve crossing for the optimization of the filling performance, can create obstructions to the flame front, favoring knock occurrence. The long experience gained during numerous tests at the engine test cell allowed to collect a lot of data about piston erosion induced by knock and about its location on the piston surface. Such erosion, as visible in this document, takes place close to the most pronounced concavities.

## 1.2. Literature Review

Knock and consequences of knocking combustions have been studied since the early development of spark ignition engines and the first documents date back to the last decades of the 1800s [1.2]. Numerous studies have been carried out to explain the phenomenology of knock [1.3-1.6], but still many questions remain open, especially on the details of knock-induced erosion and on the predictive modelling of its intensity.

In this section the aim is that to carry out a brief analysis of the existing literature in the field of knocking combustions. The goal consists in collecting some of the most innovative and significant studies in the field of knock physics, modelling and control of events' intensity and consequences on the main combustion chamber components, highlighting how there are still no relevant studies focused on the modelling of damage caused by knock. Moreover, the most common techniques for knocking events control and mitigation are described, opening up to model-based knock control strategies that are highly innovative and oriented to directly manage the piston erosion. In other words, the final purpose of the literature review is that to highlight the important and innovative results achieved during PhD course.

The chapter is divided into 4 paragraphs dealing with the following topics: knock physics, damage induced by detonating burns, modelling of knock intensity and control techniques.

### 1.2.1. *Knock Physics*

The paragraph deals with the mechanism that generates knock and the most common indices for the calculation of the intensity are resumed. The focus moves on the main types of knock and specifies the different characteristics that distinguish traditional knocking combustion from pre-ignition and super-knock. The characteristic signs of knocking combustion are finally analyzed.

#### 1.2.1.1. **Knock Mechanism**

Maly in [1.7] highlights how historically two main theories about the knock mechanism have been developed: the auto-ignition theory and the detonation one. The first one is widely accepted and argues that knock originates from the spontaneous ignition of the mixture not yet reached by the flame that reaches the critical pressure and temperature conditions, without any kind of dependence on the flame front. The second states that the flame front accelerates suddenly due to an increase in turbulence, which leads to the knocking phenomenon. Although the first theory is the most accredited, it is not enough to explain the absence of correlation between the amplitude of pressure oscillations and the estimated amount of mixture that self-ignites [1.8]. Moreover, some experiments [1.9, 1.10, 1.11, 1.12] show that during knocking combustions shock waves develop and propagate at speeds of the order of 2000 m/s. These further observations allow to assert that the knock is rather the result of the coupling of gas dynamics and reactivity phenomena.

First studies that collect the main categories in which it is possible to classify self-ignition processes are those of Zeldovich [1.13] of the second half of the 20th century, which through one-dimensional modeling of the physic-chemical processes that occur during the combustion of defined mixtures, show how there are different ways in which knock occurs: thermal explosion without shock waves, supersonic detonation, subsonic deflagration as well as gradual flame propagation. Bradley and co-authors [1.14, 1.15] expand such research by introducing two dimensional parameters that have been

widely used in the last two decades to classify the types of auto-ignition:  $\varepsilon$  defined as the ratio between the time of exposure to acoustic waves and the time of excitation in which most of the energy (from 5 to 100%) is released, and  $\xi$  the ratio between the speed of sound and the initial speed of flame propagation. Applying the calculation to the hot-spots from which the auto-ignition flames start, it is possible to plot the data in the so-called  $\varepsilon$ - $\xi$  peninsula (Figure 1.4) by differentiating the type of combustion according to the area of the plane where the points fall.

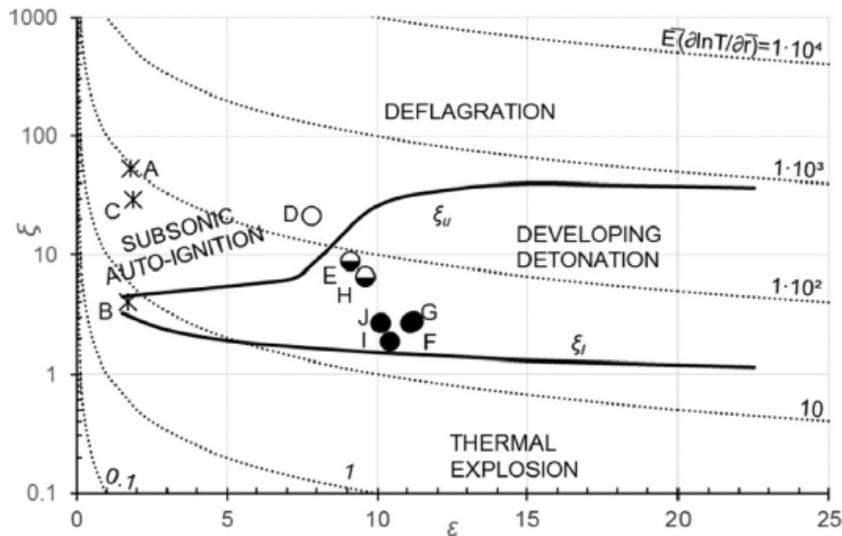


Figure 1.4:  $\varepsilon$ - $\xi$  peninsula. Circles indicate area of the field in which knock occurs and higher black fill indicates increasing severity of knock, from no knock to super-knock (picture taken from [1.15]).

In addition, Rudloff [1.16] and Chen [1.17] extended such approach to the negative temperature coefficient zone (NTC, the ignition delay is not monotonous with temperature), to apply the classification of auto-ignition modes also in NTC area.

Additional contribution in knock investigation come from studies of Pan [1.18] and Wang [1.19]. As pointed out above, auto-ignition therefore starts mainly from the areas where it is possible to reach critical conditions of pressure and temperature. Observing knock from a macroscopic point of view, in [1.19] it is highlighted how knocking combustion is characterized by two macro-phases: the propagation of the flame front due to the positive ignition of the mixture and the auto-ignition of the end-gas. Analyzing Figure 1.5 the flame front propagation starts a few instants after the spark plug is actuated (due to the ignition delay characteristic of the mixture) and ends at the crank angle corresponding to the knock onset ( $CA_{k0}$ ). In this first phase the heat release as well as the temperature and the pressure in the chamber grow in a stable way without high frequency oscillations. In the self-ignition phase (which starts instead from the  $CA_{k0}$ ) the pressure increases rapidly until it reaches the peak from which it then oscillates with decreasing amplitude. The high frequency oscillations on the measured signal derive from the pressure waves that propagate inside the combustion chamber. The measurement of the pressure signal and the oscillations induced by knock as well as the calculation of the end gas temperature and the Heat Release (HR) is very important for the characterization of the knocking event.

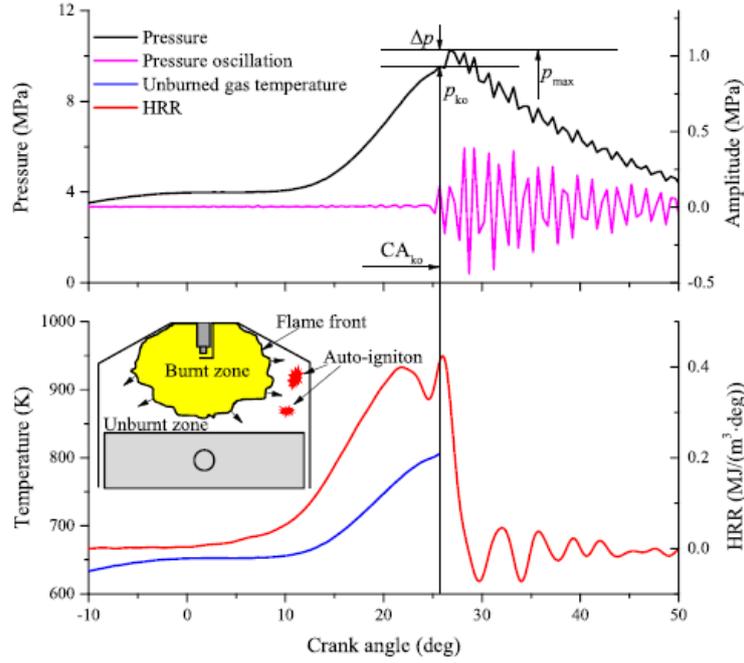


Figure 1.5: Main combustion indexes for a knocking event (picture from [1.19]).

### 1.2.1.2. Knock Intensity Indexes

There are several knock intensity indexes [1.20] that are based on different analyses of the filtered and angularly windowed pressure signal. High-pass filtering is commonly used, but frequently calculations can also be made by applying a band-pass filter. The values of the cut-off frequencies as well as the angular window extremes are often related to the experience and sensitivity of the technicians, as well as the values of the thresholds with which one is used to perform the SA calibration activities. The choice of these values is, however, relevant since, with the same index, it is possible to obtain very different ranges of values depending on the frequencies and the angular window chosen for the signal treatment. Defining as  $p_f$  the filtered pressure signal (high-pass or band-pass),  $Q$  the heat release,  $\theta$  the crank angle degrees and  $N$  the number of samples of the pressure vector, Table 1.1 collects the most common knock intensity (KI) indexes.

Table 1.1: Knock intensity indexes.

Maximum Amplitude of Pressure Oscillation:	$MAPO = \max( p_f )$
Integer-Derivative:	$ID = \int \left  \frac{dp_f}{d\theta} \right  d\theta$
First Derivative:	$FST = \max\left(\left  \frac{dp_f}{d\theta} \right \right)$
Third Derivative:	$THD = \max\left(\left  \frac{d^3 p_f}{d\theta^3} \right \right)$
Mean Square Value:	$MSV = \frac{1}{N} \sqrt{\sum p_f^2}$
Maximum Amplitude Heat Release Oscillation:	$MAHRO = \max\left(\left  \frac{dQ}{d\theta} \right \right)$

MAPO is definitely the most commonly used index and it is frequently utilized in this document when dealing with knocking intensity.

### 1.2.1.3. Knock, Pre-ignition and Super-knock

The increasingly widespread use of downsized and high boosted engines has led to handle ever higher power densities. In this type of engine when working at high loads an abnormal detonation called super-knock may occur. These phenomena are much more dangerous than traditional detonating events and can cause the motor failure in few cycles. Wang in [1.21] highlights how the super-knock has been called pre-ignition in the past even though the two events are actually different. What emerges, however, from numerous studies [1.22-1.25] is that super-knock originates from pre-ignition. Achieving the conditions that cause this type of event not only depends on the load or engine revs, but also on other conditions that can lead to overheating of some points of the combustion chamber, i.e. the spark plug electrodes. When the engine runs with high levels of pressure and temperature or with high knock intensity values, locally very high temperatures can be reached that cause the mixture to spontaneously ignite before the spark is triggered. It is therefore easy to observe how super-knock events as well as pre-ignition events have slight relationships with the main engine control parameters, such as SA, or with indicating indexes (e.g. MFB50 and P<sub>MAX</sub>) also because in some cases they can occur randomly. The traditional strategies for knock control, first of all the SA delay, are not suitable for super-knock or pre-ignition mitigation and this makes such phenomena particularly dangerous. When on-board knock intensity monitoring systems detect particularly high values associated with abnormal events, typically drastic load reduction is used, for example by fully opening the WG valve, inhibiting the fuel injection to one or more cylinders or even varying the intake and exhaust valve overlap. Figure 1.6 shows the superposed pressure signals of a cycle without detonation, with high detonation and with super-knock.

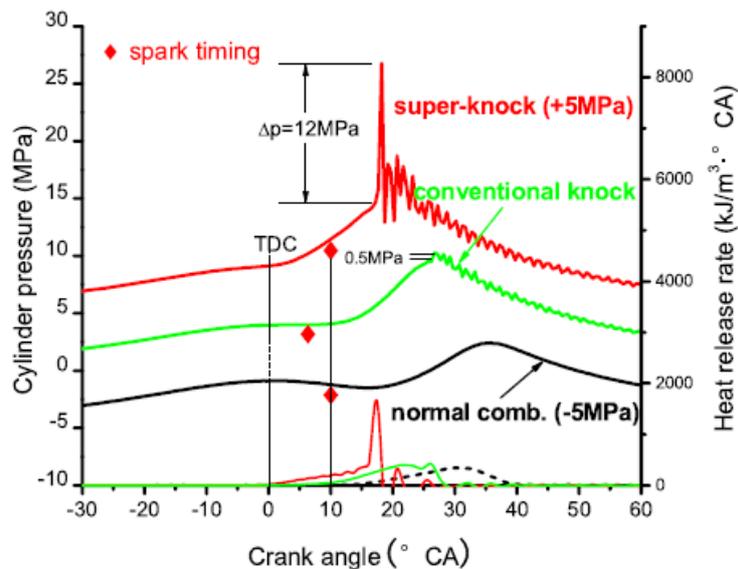


Figure 1.6: Normal combustion, conventional knock and super knock pressure curves (picture from [1.19]).

In order to clarify and to be able to distinguish the pre-ignition from the super-knock it can be stated that the first one happens when the combustion occurs due to a hot spot and not the actuated ignition, while the second one can occur in a random way when, following a pre-ignition, the combustion controlled by the spark plug starts and the two flame fronts bring the remaining hot

spots to the auto-ignition conditions. In other words, the super knock can be described as a detonation phenomenon that occurs when a part of the mixture pre-ignites before the trigger of the spark plug. In this case, the knock intensity is clearly much higher, as also highlighted by the name given to such phenomenon ("super"). In the past, these events have been defined as specific for operating conditions characterized by low engine speed and high loads. Based on this assumption, a criterion for identifying pre-ignition has been provided in [1.26] and establishes that if the peak-to-peak pressure oscillation is greater than 8.3 times the allowable knock amplitude then it deals with pre-ignition. In [1.27] it is determined that if the maximum pressure is 105 bar it is a pre-ignition cycle. Clearly these criteria are applicable for low RPM, where extremely high boost levels cannot be achieved. Nevertheless, it is difficult to identify constant threshold which can be applied generally.

Control variables that influence the pre-ignition tendency of the engine are mainly the load, the temperature, and the ignition timing. Higher intake pressure increases the detonation intensity and super-knock frequency. The temperature at the end of compression also influences knock intensity. Figure 1.7 shows the effect of pressure on knock intensity and temperature, obtained by setting the intake pressure and increasing the compression ratio (CR) respectively. However, SA is the lever with the most marked influence on the type of knock. By performing a very wide spark advance from -8 to 50 °CA BTDC, in [1.19] it is shown how, when the advance increases, it switches from normal combustion, to standard knock, to pre-ignition with super-knock, to return to knock and finally to normal combustion. This trend occurs under operating conditions that are characterized by high boost values and the same behavior is not achievable under naturally aspirated operating conditions.

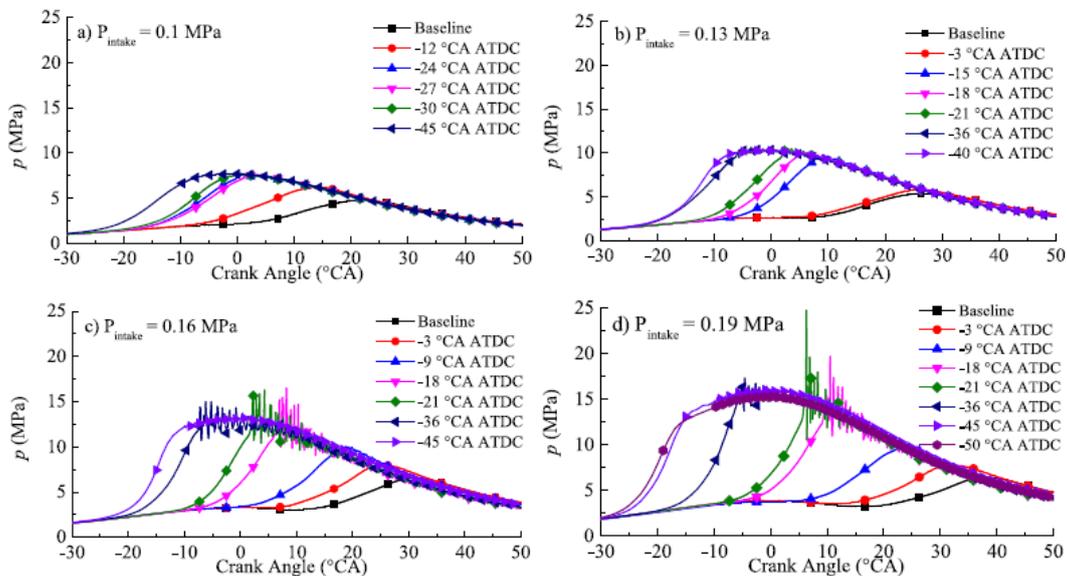


Figure 1.7: Effect of engine load, for fixed intake air temperature (20°C) and different spark timings (adapted from [1.19]).

#### 1.2.1.4. Pressure oscillation analysis

The pressure oscillations caused by knock are characterized by frequencies high enough to excite the combustion chamber vibrational modes. To capture the oscillation patterns an innovative approach has been developed using the light emissions of Carbon-Hydrogen molecules [1.19]. The shapes of the first vibration modes of a cylindrical cavity can be rebuilt using a high-speed camera

and then reconstructing the images by replacing the pixels with three colors of basic intensity based on the detected amplitudes of the light spectrum. Figure 1.8 collects the first 5 vibration modes of the camera and the respective frequencies of a cylindrical cavity for a given diameter. Such image shows captures of the high-speed camera, and, even if reported frequencies are slightly far from those of a modern combustion chamber, the indicated shapes are still representative of the mode patterns of a cylindrical cavity. According to the vibrational theory, the highest oscillation amplitudes of the pressure signal are recorded at these frequencies. The study of the chamber vibration modes makes it possible to avoid the installation of the instrumented spark plugs or piezoelectric pressure sensors of the most common indicating systems in correspondence with the nodes of the main frequencies of the chamber, which would not allow the detection of the main oscillations.

$(\alpha, \beta)$	$\rho(\alpha, \beta)$	$f_{theory}$ (kHz)	$f_{im}$ (kHz)	$T_{im}$ (K)	Mode Shapes (theoretical)	Mode Shapes (image)
(1,0)	1.841	11.51	11.25	2731		
(2,0)	3.054	19.1	18.28	2621		
(0,1)	3.831	23.96	23.91	2849		
(3,0)	4.021	26.27	26.02	2806		
(4,0)	5.318	33.26	33.05	2825		

Figure 1.8: Shapes of first 5 modes of a cylindrical cavity (adapted from [1.19])

Knock and the resulting pressure fluctuations also affect heat exchange through the combustion chamber walls [1.28]. Heat exchange by convection increases under heavy knocking conditions, up to 40% of the total energy introduced into the chamber. This observation makes it possible to clearly explain how strong knocking cycles correspond to a low value of the thermal efficiency, despite very high-pressure peaks. In fact, the exponential increase of the convective exchange coefficient causes a large dispersion of heat through the walls, reducing the portion of energy convertible into useful work. As highlighted above, the pressure waves generated during knock force the end gases near the walls to ignite. The last phase of knocking combustion that corresponds to the auto-ignition of the peripheral end gases follows the propagation of the spots positioned in front of the combustion flame and it causes the widest pressure oscillations. These waves, impacting on the walls, can break the boundary layer. This process significantly modifies the convective exchange coefficient, causing the described increase in wall exchanges. The low combustion efficiency associated with knocking cycles is clearly visible in Figure 1.9, where the maximum value of the cumulative net heat release is reported on knock intensity (i.e. MAPO): the higher the knock intensity, the lower the energy that is available in the combustion chamber to be converted into useful work.

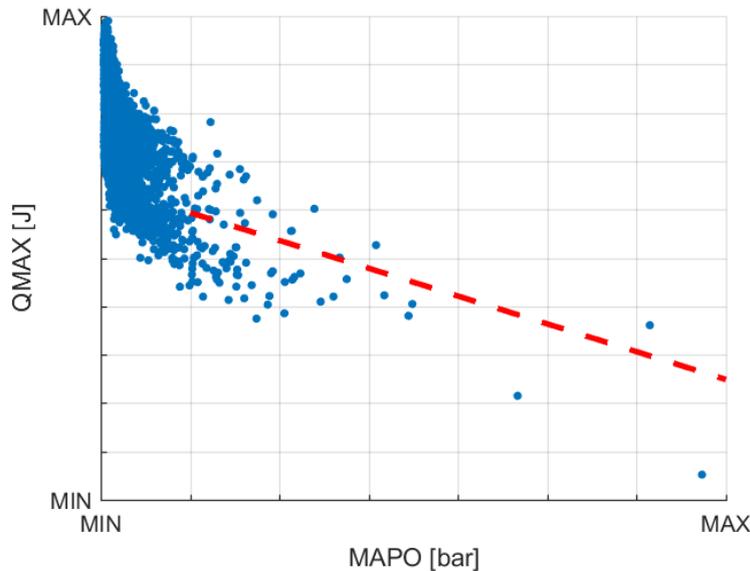


Figure 1.9: Maximum net energy for knocking combustions. Blue points represent the experimental data, while the red dashed line is the fitting function (not knocking events are excluded).

### 1.2.2. Knock Consequences

Knock effects on components of the combustion chamber are aspects that were first identified as a knock specific feature, because they cause a kind of damage that evolves quickly, it is easily recognizable, and it often leads to engine failure. The knock-induced damage was therefore one of the first distinctive features of this phenomenon and consequently it was also among the first features that have been investigated. While the types of damage caused by knock have been widely discussed, there are limited works concerning the study of the mechanism by which the knock compromises the components of the combustion chamber and the relationship between knock intensity and the magnitude of the damage. Moreover, as will be further discussed in the following paragraphs of this chapter, there is a relevant lack of studies on both physical and empirical modelling to calculate and control the damage induced on the combustion chamber components. Some significant results of the research activity concern these latter topics.

In the 1990s, Nates is certainly among the authors who studied the types of damage induced by the knock and analyzed the mechanisms by which damage occurs [1.29]. Earlier, Maly and Zeigler [1.7] discussed the effects of end-gas geometry on induced damage and in [1.8] researchers developed an experimental simulator to reproduce surface erosion of a detonation chamber component under controlled knocking conditions. From the analysis of the main works of these authors it is possible to describe the most significant advances in the field of damage induced by knocking combustions.

#### 1.2.2.1. Knock-Induced Damage

The first element that must be highlighted is that knock for given operating conditions and within certain levels of intensity does not produce tangible consequences on the components within the combustion chamber. In other cases, its presence is harmful and causes significant damage with two main mechanism:

- localized thermo-mechanical stresses

- heat flux steep increase

The first causes erosion of the material through what appears to be a fatigue mechanism. In fact, it takes thousands or tens of thousands of cycles to cause erosion, while a few large or very large episodes do not leave this type of trace. Indeed, in the author experience most of the catastrophic failure that are traditionally attributed to knock (such as piston seizure, formation of blow-by channels, ring gap closure, lands fracture...) are not merely caused by knock itself, but are rather the result of repeated, persistent knocking combustions, which grow in intensity, finally turning into mega-knock or preignition [1.30]. Nates [1.31] assumes a fatigue cycle that considers both mechanical and thermal effect. The shock wave, the mechanical action, is the fastest and therefore the one that first meets the walls of the combustion chamber. The stress is compression, perpendicular to the surface. The thermal wave, which arrives later, induces compression stress (at least in the outer layers) but parallel to the surface of the material. The time offset and perpendicularity of the stresses causes an alternate fatigue shear cycle. The thermal flow is due to the destruction of the thermal boundary layer and therefore to a sharp increase in the convective exchange coefficient. This phenomenon has effects on temperature, size and therefore on the tribology of the components, as well as causing a decay of the mechanical properties of the materials. The match between mechanical and thermal action therefore produces the damage that mainly occurs on aluminum components and on the piston. The types of damage are therefore summarized below.

- **Erosion:** it basically concerns aluminum components; steel components rarely show traces of this phenomenon. The surface of the affected area looks pitted, and it is like the cavitation. The affected areas of the piston are the surface, the top land and the first ring groove. A long exposure to knock can lead to the formation of a blow-by channel around the first ring, where melting temperatures are reached. The cylinder head and gasket may also suffer erosion, particularly in the squish area, even if it begins to occur after the piston is severely compromised. According to Maly [1.7], the surfaces, initially glossy, begin to become opaque, then take on the appearance of a sandblasted surface, and finally the typical cavitation damage is achieved.
- **Piston Seizure:** residual debris from erosion may interpose itself between the piston and the cylinder or disturb the correct functioning of the elastic rings. The increase in temperature due to the high heat flow can dilate the first ring, which, once the gap is completely closed, breaks the hydrodynamic film by meeting the surface of the cylinder. The heat generated by friction keeps or exacerbates this condition. Similarly, the piston seizure can occur between the piston and the cylinder when the heat flux is higher on the piston [1.32].
- **Pound-out:** another consequence of the knocking is the "pound-out" of the first ring groove. Excited by the oscillations induced by knock, the ring violently impacts on the two sides of the groove. Such groove expands, the material plasticizes and can also be pushed towards the surface of the cylinder that removes it. It has been shown that small variations in knock intensity can cause such effect. In addition, this mode of damage may also occur after a few severe knocking cycles and without the typical knock induced erosion [1.33].
- **Solid Deposits:** while on certain surfaces the action of the knock has a polishing effect because of the erosive action, in more protected areas but still exposed to high temperatures there may be deterioration (cocking) of the oil and the formation of carbon residues. Their formation in the groove of the first elastic ring locks the movements of the ring itself, producing a dangerous blow-by increment. Indeed, the ring rotation is essential because it allows the gap to change position. In this way the blow-by does not disturb the hydrodynamic film and does not locally heat the piston and the cylinder always in the same angular position. Moreover, these residues act as thermal insulation and cause an increase in piston temperatures [1.33].

- **First Ring Locking:** because of the first ring expansion, it can be circumferentially compressed. The ring increases its thickness, until it remains locked inside the groove. A micro-structural conversion of the grey cast iron happens when it exceeds 350° (which involves the transition of the progressive spheroidal graphite into lamellar graphite), which corresponds to a volumetric increase of about 10% [1.34].
- **Piston Top Land Break:** as a result of mechanisms number 2 and 5, the mechanical stress in the ring groove increase, cracks are created and finally a portion of the piston top land detaches. Few episodes of super-knock can cause the same consequences [1.32].

Some authors [1.8] highlight that the observed erosive surface destruction, very much resembles the damage caused by cavitation on the blades of water pumps and turbines. The cause of erosion seems mainly related to the direct shock pressure waves, while minor attention is focused on the effect of thermal stresses. However, the destruction of the thermal boundary layer leads to a sharp increase in the convective heat transfer coefficient, thus causing high heat flux to the walls. This influences the temperature profiles and, therefore, the size and tribology of the components; moreover, it can lead to a significant loss in material strength (mainly in the case of aluminum pistons) favoring a consequent mechanical damage.

### 1.2.2.2. Thermo-Mechanical Stress Estimation

The majority of literature focuses on the knock-induced damage and heat flow increase caused by the boundary layer break. There are just a few works about erosion level modelling and practically no studies about knock intensity control based on damage estimation. In the work of Maly and Nates, the level of thermal and mechanical stress to which the combustion chamber components (e.g. piston, head, and valves) are exposed is estimated. Nevertheless, such models found in literature are not suitable to be implemented in a real-time control unit.

Maly [1.8] measures experimentally, through photographic techniques and a detonation chamber, the speed of the wave-front, using lean mixtures of ethylene and oxygen. For  $\lambda = 2$  the speed is between 1500 and 2000 m/s, while for a stoichiometric mixture a value of 2200 m/s is estimated. The author, however, believes that in case of a non-planar wave-front the local speeds can vary considerably and thus reach even higher values (a +30% is estimated). Through two mathematical models (Chapman-Jouguet and von Neumann) he calculates the values of pressure in the free wave-front and when it is reflected. Depending on the used model the values deviate from each other. Moreover, the type and thermal and mechanical stress values are calculated. When the mechanical stress is above the fatigue limit of the material the author considers that erosion is happening.

Nates [1.31] calculates thermal stresses using the temperature and heat flow measurements performed by Yates (1988) on the surface of the cylinder head facing the combustion chamber. The stresses are calculated on a plate model, in which he sets the minimum temperature recorded within the cycle as temperature of the cold side, while imposing on the other surface a steep rise in temperature up to the maximum value recorded. The time the temperature profile is photographed is the instant in which the corresponding thermal flow is equal to the experimentally measured one. Known the temperature profile and the material properties the stress resulting from the theory of elasticity (Timoshenko) is calculated. For a plate with a thickness of 10 mm the maximum strain that can be obtained is around 80 Mpa in compression and 20 Mpa in traction, that are values greater than those calculated by Maly. However, the hypothesis about cold side is inaccurate and leads to underestimate the stress. Moreover, Nates supposes that in the piston top-land the temperature gradient may be even higher. In fact, under normal combustion conditions, the area between the top-land and the cylinder is a relatively "cold" region, where the flame cannot penetrate. When

knock occurs, this volume also contributes to the energy release, so the temperatures suddenly increase. This would explain the preferential erosion of the top-land.

### 1.2.2.3. Piston Damage Influential Parameters

Both Maly and Nates analyze the main engine parameters that influence the piston damage, especially the surface erosion. Such parameters are listed and described below.

- **Knock Pressure:** the pressure resulting from the reflection of the shock wave represents the mechanical action on the material and, in general, the damage is directly related to it
- **Temperature:** the temperature gradient in the component determines the amount of thermal stress and it directly determines the mechanical properties of aluminum. The piston temperature field depends on several elements, such as the engine operating conditions (engine speed, load, but also lambda and the SA), the mass flow of the lubrication oil, the position of the intake and exhaust valves, the cooling system layout and the position of each cylinder with respect to the others. Moreover, knock has a significant effect on the piston temperature. Indeed, when the knock does not occur, temperature linearly increases with respect to the SA, while if the knock occurs, the temperature increases faster [1.33]. According to Nates, the magnitude of the cyclic temperature variations of the combustion chamber surface triples in the presence of knock compared to normal combustion conditions [1.31]. Texture produced by the erosion on the piston surface increases the convective heat exchange coefficient, leading to a localized, and then global, temperature increment (from microscopic analysis on some pistons it can be seen fused metal structures).
- **Combustion Chamber Geometry:** Maly and Zeigler [1.7] carry out a specific study about the effect of the end-gas geometry with respect to the location of knock induced damage. The authors perform endurance tests on aluminum specimens with a detonation chamber confirming that damage occurs preferentially in geometry-constrained areas (corresponding to the area between the cylinder and the top-land of the piston). Within a limited volume where more surfaces are faced to each other, more reflections and therefore more impacts occur, even if with decreasing intensity. Tests on a SI engine, under constant knocking conditions, have been performed, applying several variations to a standard configuration of the piston. The most beneficial change in terms of piston erosion is given by the elimination of the chamfer of the upper edge of the first ring groove. Positive effects have also been obtained reducing the clearance between piston and cylinder. The adoption of both these modifications causes a reduction in damage with respect to the reference geometry, but still less than the two measures adopted separately. The presence of a groove along the circumference of the top-land halves the damage.
- **Air-to-Fuel Ratio:** the tests carried out with a detonation chamber demonstrate the piston erosion is higher with a rich mixture.
- **Piston Surface Coating:** a protective layer of Nickel (15  $\mu\text{m}$ ) on the aluminum piston in a research engine eliminates the appearance of erosion. The presence of Nickel limits the formation of carbon deposits (responsible for creating "hot spots"), reducing the engine's sensitivity to knock. Positive effects also on wear and friction coefficients have been observed [1.35].

#### 1.2.2.4. Piston Temperature Measurement

As discussed above, the temperature of combustion chamber component is a key variable for the calculation of the knock-induced piston damage. The main technologies for the measurement of such temperature are listed below:

- **Fusible Plugs:** plugs with fusible tips for given temperature
- **Tem-plugs:** tensile specimens for which the mechanical properties are measured before and after their installation in the piston. Through the property decay and the exposition time it is possible to estimate the mean temperature of the piston
- **Resistance Temperature Detector:** such tool measures the resistance variation of the material with respect to the temperature. Signal is transmitted through an inductive coupling with the engine block
- **Thermocouple:** the NiCr-Ni thermocouple can be installed on the piston [1.53] and a wire with a system that follows the piston alternative motion is required.
- **Hardness test:** the mean piston temperature can be estimated when the piston hardness before and after the test is measured

The first four methods imply the installation of an extra component on the piston or on the engine and, in some cases, it can become too intrusive. On the contrary, the last solution does not need any piston or engine case machining, and this makes such techniques preferable when the piston geometry cannot be modified.

#### 1.2.2.5. Piston Damage Related Indexes

In the analyzed works, Nates [1.29] tries to identify the indexes that show a reliable correlation with the main knock induced damaging effects. A significant effort has been applied to find the optimal indicators for the trend of the heat flux and the surface erosion. Results lead to consider the average value of the maximum pressures of the knocking cycles as a good index of the heat flux. It also suggested such index can be calculated considering all the cycles or only the knocking ones. The evaluation of the thermal flow can be experimentally quantified on a research engine through the expansion of the first ring. Erosion, on the other hand, is well correlated with the pressure value in the combustion chamber, recorded just before the knock starts. The sensitivity analysis carried out on the aluminum specimens within a research engine shows that the knock pressure is directly related to the initial one. It also defines a threshold for this parameter of 33 bar, below which, when knock occurs, there is no erosion. The benefit of this index is the simplicity and accuracy with which it can be calculated since it is not affected by the high frequency signal. Instead, there is no good correlation with MAPO type of indexes. Mahle in [1.33] proposes a different knock intensity index that can be correlated with the cumulative erosion during endurance tests. Such index is calculated with Equation 1.1:

$$KI = N \frac{\sum_{i=0}^M n_i f_i}{C} \quad (1.1)$$

Where:

- $K$  defines the pressure interval, with which the maximum knock pressure can be classified
- $M$  is the total number of intervals
- $n_i$  and  $f_i$  define the number of cycles within each interval and the weight given to each interval respectively

- $C$  is the total number of cycles
- $N$  is a constant

Such index includes some calibratable coefficients that are the weights applied to each interval and the constant  $N$ . Moreover, Mahle indicates the maximum positive amplitude of pressure oscillation as a well-correlated index with respect to the piston damage. Further tries have been carried out by Renault [1.36] and Essig [1.37].

More generally, several authors are rather wary about data measured with a piezoelectric pressure sensor faced to the combustion chamber. In fact, it is believed that its whole architecture (dynamics of the sensor, filter, etc.) is not able to correctly sense the pressure trace induced by the impact of the shock wave, which is the most interesting signal, but rather detects the chaotic and local response of the system. The first measured peak, which should correspond to the wave front of the detonation, should be greater in amplitude than the following ones, related to the system response. Often this is not verified. However, any measurement by the sensor leads to an underestimation due to the lack of orthogonality between wave direction and sensor surface and the non-uniform distribution of pressure. This means the force detected is then averaged over the total surface. For these reasons Nates and other authors believe that indexes derived from the pressure signal are scarcely correlated to the damage.

#### **1.2.2.6. Conclusions About Presented Literature**

Discussed works do not include an objective measure of the knock-induced piston damage. However, it has been demonstrated that the definition of a reliable method to evaluate the piston surface erosion is needed in order to develop a robust piston damage model. Temperature of combustion chamber components is one of the factors that strongly affect damage level. It is then necessary having an accurate piston temperature model which must be sensitive to the engine operating conditions (engine speed, load, intake air temperature, lambda and lubricant oil temperature). The choice of fuel also deserves attention, determining both the knock frequency and its intensity.

More generally there are a few authors that analyzed in detail the knock-induced mechanisms on main combustion chamber components, as observable by this analysis. For this reason, it can be argued that such phenomenon is not currently fully understood. Moreover, the development of a knock index directly related to the damage induced on engine components, and the corresponding methodology to identify proper diagnostic threshold values, is still an open and crucial challenge.

#### *1.2.3. Knock Modelling*

In this section of the document the main methods for the characterization of the knock process are presented. The main modelling works are then analyzed with particular attention to 1-D and 0-D approaches. In this way the aim is also that to highlight there are no control-oriented empirical models that allow the real-time calculation of knock intensity index that takes into account all the main functional variables that influence the knock event (i.e. engine speed, load, SA, lambda, air temperature in the intake manifold and fuel quality)

### 1.2.3.1. Knock Process Characterization

The most important preliminary remark to approach the knock modelling is definitely the stochastic nature of such phenomenon, which makes the process critical both for the characterization and for the control. This means that by fixing the operating conditions of the engine, each cycle can generate a knock process with very different intensity from the previous one. Clearly, these variations are linked to the Cycle-to-Cycle Variation (CCV) that is particularly accentuated in SI engines, but even with the same main synthetic combustion indexes, pressure oscillations with very different amplitudes can be achieved. This characteristic is intrinsically linked to the different fluid-dynamic conditions present in the combustion chamber due to the chamber filling and fresh charge mixing processes as well as spark ignition and flame front propagation, which are never repeatable in the same way.

As mentioned by Spelina, James and Jones in [1.38] knock process cannot therefore be characterized by the traditional statistical approaches used for common combustion indexes (i.e. average value for given operating conditions). The most reliable approach is based on the statistical analysis of knock intensity for given operating conditions and the definition of its Probability Density Function (PDF) or Cumulative Distribution Function (CDF). Such distribution can be easily obtained by considering the frequency with which each knock intensity interval occurs and by calculating the consequent probability with which each value is recorded [1.39]. The PDF characteristic of the knock intensity is asymmetric and the log-normal, Weibull or Gamma distribution are the kind of functions that are most used to describe this trend [1.40]. The main difference with the probability distribution of the combustion indexes is therefore related to the different effect that the CCV produces on the values of such indexes. In fact, the probability distribution for fixed engine operating conditions of combustion indexes such as MFB50, PMAX and IMEP is Gaussian and the average value can be used as a characteristic value of the engine point. The shape of the probability distribution of knock intensity changes as a function of the engine operating parameters. For example, if, for a given engine point, the SA increases or decreases, the mode of the distribution pattern moves towards higher or lower knock index values, respectively. The likelihood of heavy detonating cycles also increases or decreases (Figure 1.10).

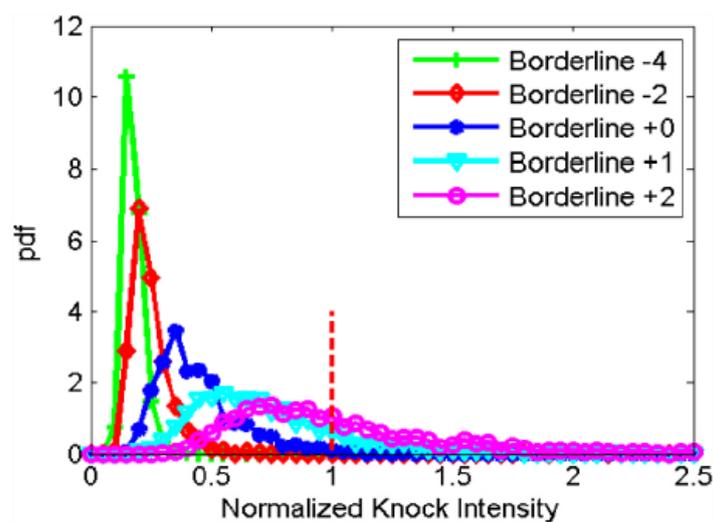


Figure 1.10: Variation of knock intensity PDF, moving the spark timing (from [1.38]).

The possibility to describe the characteristic PDF of each operating condition with a known function (i.e. log-normal, Weibull and so on) ensures that the engine's knocking behavior is described in a

very concise manner. The parametric approach also allows to calculate other characteristic indexes analytically, which can then be used to feed a possible control system [1.41]. Although this approach would allow to rebuild the whole probability distribution for given operating conditions, the identification of the PDF coefficients should consider the effect of many variables such as engine speed, load, SA, lambda, and the fuel quality and also a map-based approach would involve the definition of matrices with many dimensions, which would be difficult to use. For this reason, modelling the rate of knocking events remains the classic method to control knock. Most control systems therefore apply thresholds directly to the measured intensity levels and then reconstruct the probability with which cycles over such threshold occur. If the knock events rate is higher or lower than a certain value the SA is delayed or increased respectively, although the widest part of production control systems does not implement positive advance corrections but work only for protection purposes. The classification of knock intensity into knocking and non-knocking events is useful because the probability of events above the threshold alone is characterized by a binomial distribution (Figure 1.11). This observation further simplifies the characterization of the probability of the intensity of knock events. By comparing the probability distribution estimated by the online measurement with the modelled one, it is possible to evaluate the need to correct the SA.

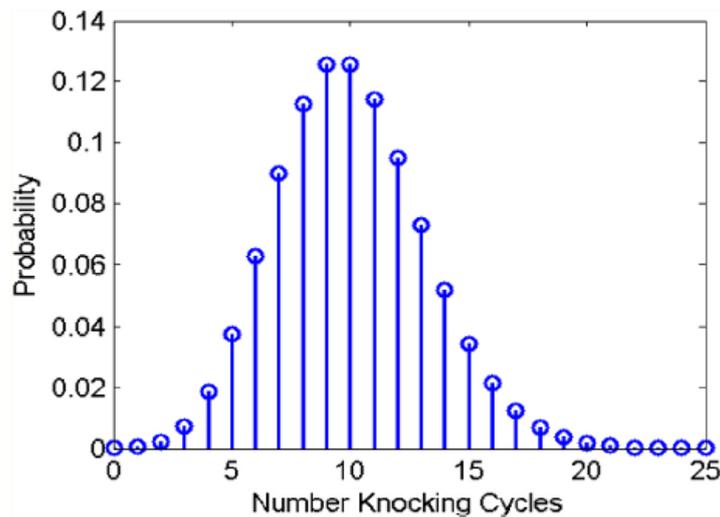


Figure 1.11: Binomial distribution of knocking events observed in 1000 cycles on an engine point characterized by a knock probability of 0.01 (from [1.38]).

### 1.2.3.2. Knock Modelling Approaches

As highlighted above, a statistical approach to knock phenomenon description has opened new opportunities to develop and improve the modelling activity.

For a fixed operating condition, knock intensity is strongly influenced by CCV. Several studies [1.42-1.46] demonstrate that a strong relationship exists between MAPO (Maximum Amplitude of Pressure Oscillations) knock intensity index and the charge turbulent motion near the spark plug or the in-chamber temperature distribution. Physical knock models that derive from these studies can be efficiently calibrated to determine the knock onset, the Knock Limited Spark Advance (KLSA) or, more recently, to predict the trend of a MAPO percentile with respect to SA. Extending the analysis, knock models can be generally classified in three main groups:

- Knock models based on detailed chemical kinetic mechanisms
- Knock models based on reduced chemical kinetic mechanisms

- Knock models based on Arrhenius equation

The first and the second ones simulate pre-flame reactions and consider every sub-reaction that involves several intermediate chemical species, during the combustion process. This kind of models are extremely time consuming and also require the rate coefficients for most (first category) or some (second category) chemical reactions, that need complex analysis [1.47-1.49]. Such models are then limited to 3D Computational Fluid Dynamic (CFD) simulations in which it is possible to compute the pre-flame reactions [1.50, 1.51]. The third category does not need such kind of data for detailed chemical reactions and for this reason they are more suitable for 1-D fluid -dynamics engine simulation environments, in which the two-zones method [1.44] for the SI combustion process modelling is commonly implemented (the combustion chamber after the spark plug actuation is divided into one burned and one unburned zones). This kind of models refers to the induction time of the mixture and the knock onset, that can be evaluated by implementing the Arrhenius equation:

$$\tau = Ap^{-n} \exp\left(\frac{B}{T_u}\right) \varphi^x \quad (1.2)$$

Where  $p$ ,  $T_u$  and  $\varphi$  are pressure, temperature, and equivalence ratio of unburned gas zone, while  $A$ ,  $n$ ,  $B$  and  $x$  are constants characteristic of the temperature region for which the Equation 1.2 is applied and these latter can be obtained from a fitting procedure of the values of a kinetic tabulated map [1.52, 1.53]. This kind of models establishes the knock onset when the induction time integral (ITI) becomes equal to 1. The formulation of this approach developed by Livengood and Wu in [1.44] is the following:

$$ITI = \int_{t=0}^{t_{knock}} \frac{1}{\tau} d\tau = 1 \quad (1.3)$$

Where  $\tau$  is the induction time for the air-fuel mixture and  $t_{knock}$  is the elapsed time between the start of compression stroke and the hot spots auto-ignition instant. Several models are based on such methods for the determination of knock occurrence and knock onset but the most extensively adopted is that proposed by Douaud and Eyzat [1.45]. A further development of such modelling approach is that proposed by Gamma Technologies [1.54], called Kinetics-Fit (KF) which allows to avoid some weakness of the standard Arrhenius-based methods. In particular the Negative Temperature Coefficient behavior of some fuels and the induction time dependency from AFR and residual gas fraction have been introduced. The KF model implements three induction times to capture in a more accurate way the hot spots auto-ignition over a wide range of temperature. The resulting auto-ignition delay time is defined by Equation 1.4 and the induction time for the low, medium and high temperature region are considered:

$$\frac{1}{\tau} = \frac{1}{\tau_1 + \tau_2} + \frac{1}{\tau_3} \quad (1.4)$$

Where  $\tau_1$ ,  $\tau_2$  e  $\tau_3$  are the induction time for low, intermediate and high temperature areas, respectively. The induction time is calculated through the Equation 1.5:

$$\tau_i = M_1 a_i \left(\frac{ON}{100}\right)^{b_i} Fuel^{c_i} O_2^{d_i} Diluent^{e_i} \exp\left(\frac{f_i}{M_2 T}\right) \quad (1.5)$$

Where:

- $ON$  is the fuel octane number

- *Fuel* is the concentration of fuel with respect to the mixture volume [mol/m<sup>3</sup>]
- *O<sub>2</sub>* is the concentration of oxygen molecules [mol/m<sup>3</sup>]
- *Diluent* is the concentration of the sum of N<sub>2</sub>, CO<sub>2</sub> and H<sub>2</sub>O molecules [mol/m<sup>3</sup>]
- *a<sub>i</sub>...f<sub>i</sub>* are model constants
- *M<sub>1</sub>* and *M<sub>2</sub>* are the Induction Time Multiplier (ITM) and the Activation Energy Multiplier (AEM) respectively

The ITM and AEM are the calibratable coefficients of the model that are tuned to minimize the error between the experimental and the simulated values. As observable from Equation 1.5, the AEM changes the relationship between the ignition delay time and the temperature, as well as the ITM allows to scale the overall induction time. The resulting mean induction time integral evaluated between all temperature regions allows to calculate the GT Knock Intensity (KI) index, that assumes values greater than 0 only when the knock occurs. Such index also linearly depends on the unburned mass fraction. Equation 1.6 represents the GT KI formulation:

$$KI = 10000Mu_b \frac{V_{TDC}}{V} \exp\left(\frac{-6000}{T_u}\right) \max(0, 1 - (1 - \varphi)^2 I_{ave}) \quad (1.6)$$

Where:

- *u<sub>b</sub>* is the unburned mass fraction
- *V<sub>TDC</sub>* is the cylinder volume at Top Dead Center
- *V* is the cylinder volume at knock onset
- *T<sub>u</sub>* is the unburned gas temperature
- *Φ* is the unburned zone equivalence ratio
- *I<sub>ave</sub>* is the average induction time integral, between all considered temperature regions

This means that when the CCV GT model is calibrated it is possible to mimic the cyclic dispersion of the knock intensity values. With the described model only the cycles that generate a KI over a fixed threshold have been modelled as knock event and, as discussed above, the PDF of such cycles is characterized by a binomial distribution. From a numerical point of view, the KI index cyclic trend derives mainly from the unburned mass fraction behavior, that has a binomial PDF due to the influence of the calibrated CCV model. In fact, the GT CCV model reproduces the gaussian PDF of the main synthetic combustion indexes. In this way not the entire log-normal PDF can be defined, but only the binomial distribution for the cycles over the threshold. In this context the knock threshold definition is a key parameter for the calibration of this kind of model, and if it is calibrated with two different threshold values the resulting KI indexes are not comparable.

Such models try to define the physics of the knock process and request lower computational efforts than those that implement chemical reactions. However, as described in the mentioned literature, they need to simulate (and therefore calibrate) the CCV and typically the Arrhenius formula is applied to three different temperature regions. Consequently, their calibration requires extensive experimental data and a big identification effort, due to the large number of calibration parameters. Moreover such model is not able to reproduce the complete PDF for a given engine operating condition and count the effect of an influent variable (for instance the fuel quality) on the characteristics of such distribution.

### 1.2.3.3. 0-D Knock Simulation

The development of modern knock control system usually needs an extensive use of simulation, to test novel algorithms via Software-in-the-Loop approach. Empirical knock simulators are the most suitable for this kind of application, due to the low computational power requested to run models and the low execution time that allows to test the control systems under conditions close to those faced at the test bench and on-board application. The 3-D CFD approach would be able to simulate the spatially distributed process of knocking combustion [1.55, 1.56] and reproduce the vector quantity of the knock signal. Nevertheless, knock control strategies work with cyclic knock intensity indexes and this imposes a calculation of such index from the vector quantity. To integrate all these processes into a single knock simulator a wide computational power is required and for this reason 0-D modelling is normally applied to simplify such procedure.

However, to reproduce the cyclic behavior of a knock intensity index, such as MAPO, a stochastic approach is required, due to the random nature of the knocking process. Physical models are typically deterministic and for this reason they are not suitable to excite a knock controller. A possible method is that to playback experimentally logged data, for fixed operating conditions [1.57]. On the other hand, the knock intensity PDF can be characterized and modelled using lookup tables [1.58, 1.59]. When the PDF coefficients are known, the cyclic knock intensity values can be randomly extracted, according to the probability distribution. Such methods are particularly suitable when exploring the effect of different threshold levels on the controller behavior. If the threshold is fixed and only the “knocking” events can be simulated, the binomial PDF has to be considered as highlighted above, simplifying the simulator implementation.

However, the described approach is suitable for knock intensity simulation for few engine points and for steady-state conditions. Modelling of the knock intensity PDF coefficients through a lookup table as a function of the main engine parameters that affects knock tendency (engine speed, load, SA, lambda, intake air temperature, fuel RON) would mean having an n-dimension lookup table that has to be calibrated with an extremely huge calibration database. Moreover, the approach based on the binomial PDF needs to be recalibrated when a different knock threshold is defined.

In the second part of this chapter an innovative analytical approach for the knock intensity cyclic simulation is described. Such model allows to simulate the cyclic MAPO values for any operating conditions, and it includes all the engine variables that influence engine knock in a single simple analytical equation. In this way such function can be also rearranged to target any MAPO threshold, converting the model in a controller of one (or more than one) of the included engine parameters.

### 1.2.4. Knock Control

Knock control is probably the main issue in the SI engine research field due to both the not complete knowledge of the phenomenon and the possibility of wide improvements in terms of engine efficiency and emissions levels with a more aware and accurate management of knocking combustions.

Strategies to mitigate engine knock tendency can be generally subdivided into two main categories:

1. Engine design-oriented methods
2. Combustion control-oriented strategies

The first group includes solutions that can be adopted in the engine design phase and are suitable to suppress and prevent knock, while the second group is about methods to mitigate knock intensity when the engine runs. In other words, the former category deals with available methods related to the engine design phase and the latter is useful when the engine has already been produced. Port and direct water injection can be considered as an example of a solution that falls within this group. It has become a key technology for increasing combustion efficiency in modern GDI turbocharged engines. In fact, the addition of water significantly mitigates the occurrence of knock, reduces exhaust gas temperatures, and opens the possibility to reach optimum heat release, phasing even at high load. Water injection is not a novel technology, especially for aeroengines, but during these last years it is becoming a suitable solution also for passenger cars. A prototyping Port Water Injection system has been developed by the author research group and a 1-D GT Power combustion model and a WI-based combustion controller has been developed and validated during the PhD course. This topic will be faced in the chapter 2 of this document.

As mentioned in [1.19], conventional knock is a “race” between the flame front and the thermal auto-ignitions of unburned end gas. The purpose of the methods to prevent knocking combustions is that to increase the induction time of the unburned mixture to make the time of the combustion flame propagation lower than the end gas auto-ignition one. In this section literature works about second category are discussed because it is the field in which research activity of the author focused.

#### **1.2.4.1. Knock Control-Oriented Strategies**

The most effective and common strategy for the online mitigation and control of conventional knock is for sure the spark timing retarding. The fuel quality improvement (RON) and the mixture enrichment are the other feasible control-oriented solutions for knock suppression. On one hand, the fuel RON is not an operating parameter that can be controlled or chosen by the engine manufacturer and, on the other hand, the mixture enrichment causes problems of fuel economy and emissions. For such reasons, the spark timing delay is commonly adopted as the “official” solution for knock mitigation, and the fuel enrichment is typically used for lowering the exhaust gas temperature. At the same time, as well known, a spark advance correction leads to significant variations of the exhaust gas temperature, that is a key variable in modern TC engines. Indeed, the SA control strategies cannot ignore the gas temperature at the turbine inlet, that is typically estimated by ECU with semi-physical models. An analytical approach to exhaust gas temperature modelling compatible with the real-time execution is developed by the author and it is proposed in the third chapter of this work.

As discussed in the Knock Modelling section, knock intensity can be approximated as a random process, that can be well described through a statistical approach. This means that it is not possible to control the outcome of the forthcoming cycle based on the observation of the current cyclic knock intensity. Thus, the standard control systems for the management of deterministic dynamic systems where the recorded error for the current cycle can be used as a feedback to apply a correction to the next timestep cannot be applied to control knock intensity.

As mentioned in [1.38], knock control refers to control of some statistical quantity, such as the knock event rate or a knock statistical index, rather than the cyclic knock intensity. The majority of knock control algorithm are thus stochastic. Some studies investigated the possibility to control statistical values of knock intensity (e.g. mean or standard deviation value), even if the most common strategies control knock event rate. The purpose is not that to avoid knock events entirely. In fact, this result, could be achieved by retarding the spark significantly, but would also result in an

unacceptable loss of energy efficiency and an excessive increment of exhaust gas temperature. The final aim is therefore to regulate the knock rate to some low (typically 1%) rate. In other words, a percentile of knock intensity (e.g. the 99<sup>th</sup>) is maintained under a certain threshold. It is clear how the definition of the percentile and the threshold are key parameters of such controllers. The higher the controlled percentile, the safer (and less efficient) the control behavior.

These algorithms can be classified according to the actuation rate and the magnitude of the control actuation. Most production knock control strategies use a standard algorithm where the controller actuates a correction every timestep. The magnitude of these control moves is also fixed, advancing the spark by a small amount ( $K_{adv}$ ) when the knock does not occur, or retarding by a much larger amount ( $K_{ret}$ ) if knocking events have been detected. The standard knock control strategy can be resumed by Equation 1.7:

$$\theta(n) = \begin{cases} \theta(n-1) - K_{ret} & \text{if knocking} \\ \theta(n-1) + K_{adv} & \text{otherwise} \end{cases} \quad (1.7)$$

In some cases, the strategy is that to actuate only retarded spark advance when knock occurs, without reacting if knock intensity is low. Equation 1.8 describes such algorithm:

$$\theta(n) = \begin{cases} \theta(n-1) - K_{ret} & \text{if knocking} \\ \theta(n-1) & \end{cases} \quad (1.8)$$

The main issue for this kind of controller is that there is a wide dispersion in applied spark angle, due to the actuation of a different spark advance every timestep. The controller described by Equation 1.7 are called “over-active” controllers. Figure 1.12 (a) shows the typical response of an “over-active” controller for an overly retarded and overly advanced initial condition. In both cases, the controller never reaches a steady-state condition. This is highlighted also in Figure 1.12 (b), where the probability distribution of the actuated spark advances is shown.

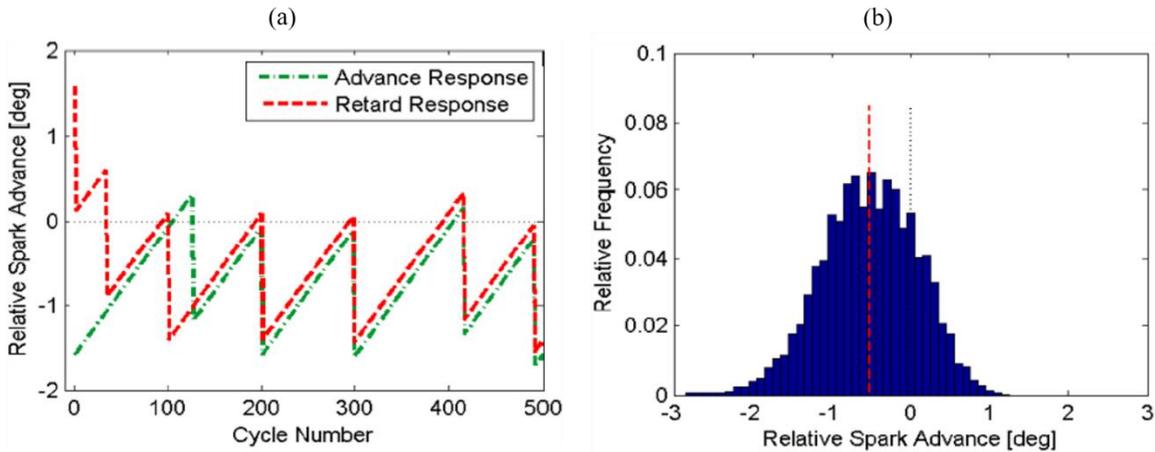


Figure 1.12: (a) Response of a standard controller for different initial conditions and (b) related spark advance distribution (adapted from [1.38]).

Alternative knock rate control strategies have been proposed in [1.71-1.74]. Cum-sum and Likelihood-based strategies are described. Such algorithms make no control moves unless there is a statistical evidence that there is a need to do so. Cum-sum-based controller [1.71, 1.72] maintains a cumulative summation of the number of knock events since the last control move and only adjust the spark angle is this Cum-sum exceeds or undershoots the statistical expectation by more than an

upper or lower threshold amount. Likelihood-based strategies [1.73, 1.74] also avoid unnecessary control actuations and only reacts when the knock rate target is not met. The observed knock event rate is compared to the target ( $p$ ) through the likelihood ratio, defined by Equation 1.9:

$$L_n(k) = \frac{p^k(1-p)^{(n-k)}}{p_{max}^k(1-p_{max})^{(n-k)}} \quad (1.9)$$

Where  $p_{max}=k/n$  is the most likely estimate of the knock rate, given by  $k$  events in  $n$  cycles. If the likelihood ratio is close to 1, it means that the observations are coherent with the target, and no corrections are needed. On the contrary, if such parameter falls under a specified threshold, it suggests that the data does not conform to the target rate, and the spark advance needs to be adjusted. Likelihood-based control strategies are more reactive than the Cum-sum method, because of the higher weight that a knock event that occurs in first few cycles has on the likelihood ratio. This could cause the fall of such index under the fixed threshold, while more than one knock event is required to exceed the Cum-sum upper threshold. However, the influence of a series of knocking events on the likelihood ratio diminishes as the number of cycles,  $n$ , increases. Typically, the magnitude of the spark angle correction is not fixed, but it can be the output of a lookup table, as well as proportional to the difference between the likelihood ratio and the threshold (the same can be applied also to Cum-sum strategies).

In Figure 1.13 the response of a Likelihood controller for both the steady-state and the transient operating condition is shown. As described above, the steady-state response is much less active than the traditional controller. In fact, also the PDF of the actuated spark advances is characterized by a much lower standard deviation value than that of a standard controller. Nevertheless, the algorithm is quite more complex than the Equation 1.7 or 1.8, and typically a preliminary Software-in-the-Loop calibration process is needed. The transient response from an advanced condition is still quite fast but is unacceptably slow the transient response when the test starts from a retarded spark advance.

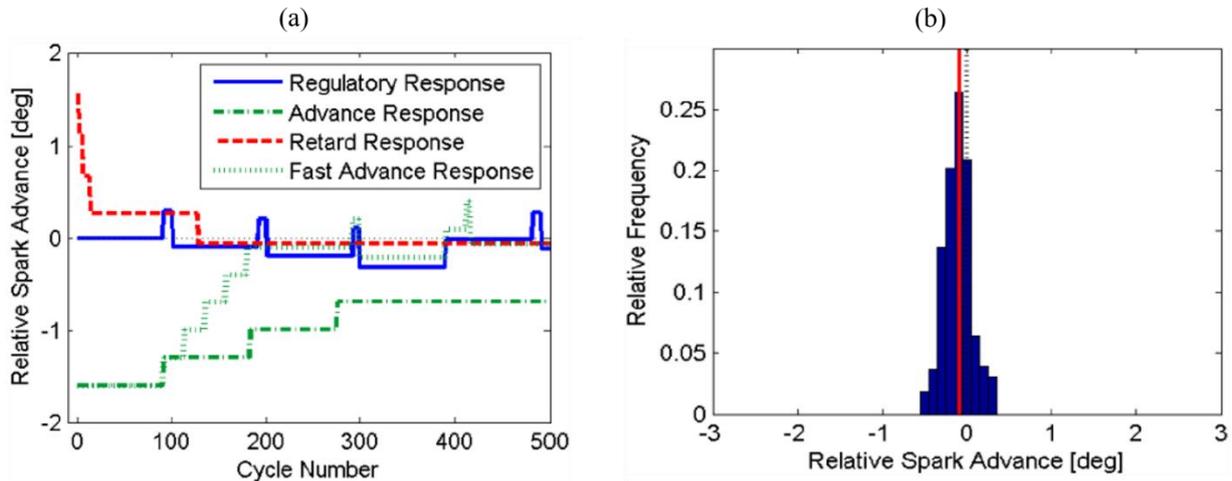


Figure 1.13: (a) Response of a Likelihood controller for different initial conditions and (b) related spark advance distribution (adapted from [1.38]).

The solution for this problem is the knock threshold optimization. The knock threshold used to define knocking events is typically set at a relatively high level based on the implicit assumption that to control potentially damaging knock events, it is necessary to use threshold suitable for the isolation of the highest intensity events. Dealing with this kind of controllers, this approach is partially incorrect, especially for pure closed-loop control strategy. In fact, fixing a high threshold (e.g. 99<sup>th</sup> percentile), the controller is forced to work with a rather low percentage of cycle (1%) and

this cause a shallow gradient of the knock probability on the SA (Figure 1.14). It is particularly complex to detect when the knock rate falls below the 1% target. Moreover, due to the low sensitivity of the knock probability to the spark advance, the controller behavior is mainly driven by the calibration setting: the higher the weight of the error between the knock ratio and the target, the more nervous the controller. The lower the weight of the error, the slower the controller in transient conditions.

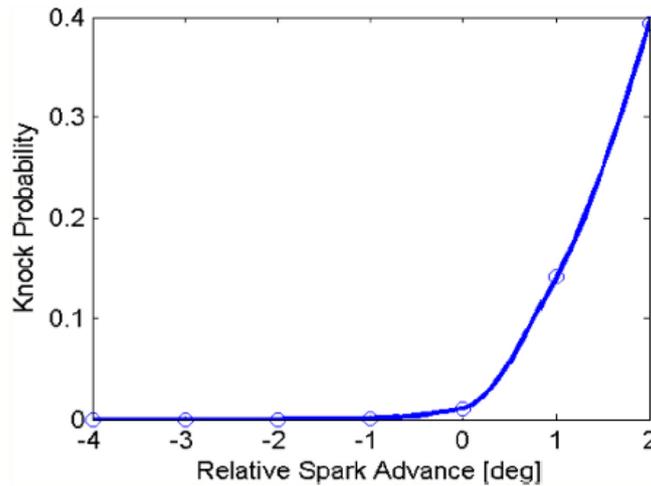


Figure 1.14: Knock probability w.r.t. spark advance (from [1.38]).

Lowering the threshold causes an apparent increasing knock rate, giving rise to the steeper knock probability characteristic shown in Figure 1.15 (a), though it should also be mentioned that these events represent “weak knock” episodes. It is possible to highlight how, with this approach, the knock probability significantly increases with respect to the SA. Since weak knock events occur much more frequently, the target knock rate has been raised to a higher value (qualitatively to the 75<sup>th</sup> percentile). The results are significantly different from the previous one. The controller response in transient condition is faster, as observable in Figure 1.15 (b) and the steady-state response does not exhibit the cycling behavior traditionally associated with this type of controller. Moreover, the applied spark angle is narrowly distributed, and the hard knock rate evaluated offline is close to the desired 1% target.

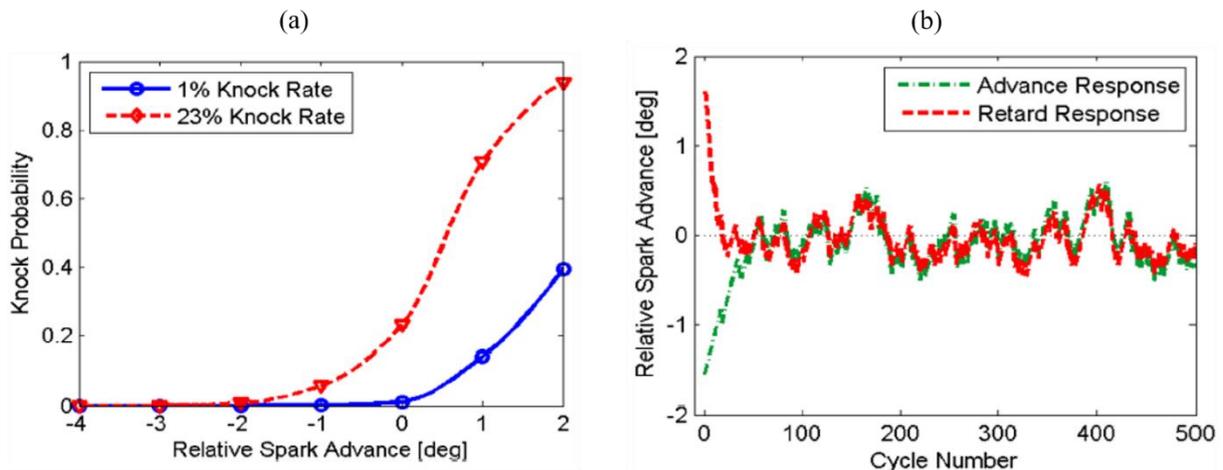


Figure 1.15: (a) Knock probability with two different knock threshold and (b) response of a standard controller with an optimized threshold (adapted from [1.38]).

The presented solution allows to partially mitigate the intrinsic lack of a purely closed loop knock control system. However, the limitations related to the calibration setting are not negligible and only a reliable and accurate open loop-based controller can suppress such issues. In the second part of this document an innovative open loop adaptive model-based controller is presented and validated.

### 1.3. Control-Oriented Modelling

In the previous chapters, it has been explained that the auto-ignition of the mixture is governed by the chemical kinetic rates, which are strongly affected by the time history of the end gas thermodynamic conditions. These latter are then influenced by the CCV that arise mostly due to the differences in the trapped charge quantity, composition, and the interaction of turbulence with the nascent flame kernel. It has been also found that knock onset can be predicted with a reasonably accuracy using a relatively simple calculation of the induction time integral [1.45, 1.54].

As observed by Gandhi in [1.75], in the control-oriented knock modelling field there are several physical and semi-physical approaches, mainly based on chemical kinetic rates, volume expansion rates, the unburned energy at knock onset, CO oxidation rate, and temperature gradient in the end gas zone. Nevertheless, several parameters and quantities related to the end gas condition are very difficult to measure or predict and most of these works do not achieve satisfactory results. Despite the Arrhenius formula is commonly implemented for the induction time calculation, the author demonstrates that the thermo-chemical history of the end gas is often not enough to entirely define the knock event intensity. In fact, Figure 1.16 shows three individual cycles for the same operating condition that are characterized by the same knock onset and the same combustion phase and rate, but with three different MAPO values.

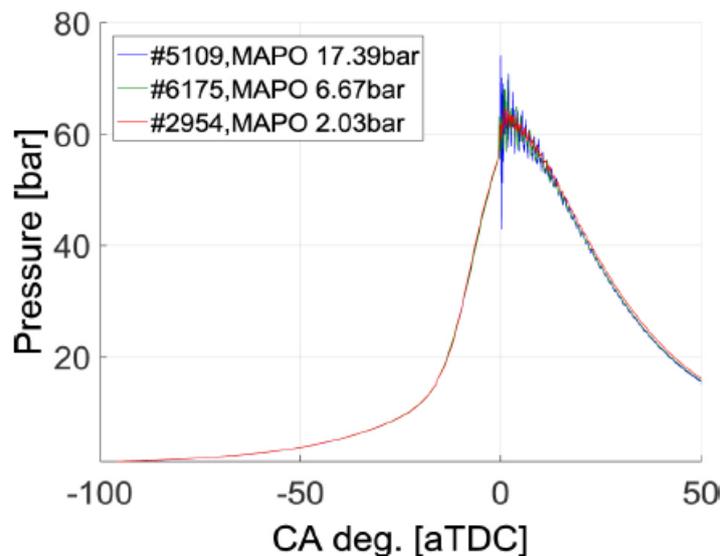


Figure 1.16: Three engine cycles with the same thermodynamic conditions at knock onset and different knock intensity (from [1.75]).

This means that the in-cylinder thermodynamic conditions at the time of knock onset are the same, i.e. the unburned mass fractions, the temperature and pressure history of the end gas are the same for the three cycles. However, knock intensity varies by more than a factor of eight. Gandhi suggests possible causes for this difference, such as an incorrect or a slightly different measurement of in-cylinder conditions, a heterogeneity in the end gas composition and temperature, even if the exact cause cannot be fully understood. In this study, the author then proposes a revision of a method based on the analysis of the knock intensity (evaluated through the MAPO index) with respect to the unburned mass fraction at the time of knock onset ( $X_{u@KO}$ ). In this case, the 95<sup>th</sup> percentile of the MAPO PDF has been calculated for a different interval of  $X_{u@KO}$  values, defined to include the same number of cycles. Figure 1.17 (a) shows a sample bin width, while Figure 1.17 (b) displays the PDFs calculated for each  $X_{u@KO}$  bin. A numerical model to calculate the 95<sup>th</sup> percentile

of these probability distributions is finally proposed. However, the direct comparison between the predicted and the measured knock intensity (MAPO 95%) shows a significant magnitude spread (the calculated values are about a factor of six too large) and no trend with engine speed, load or other functional parameter was therefore identified. This means that the parameters of such model cannot be reliably described as a function of the main engine parameters.

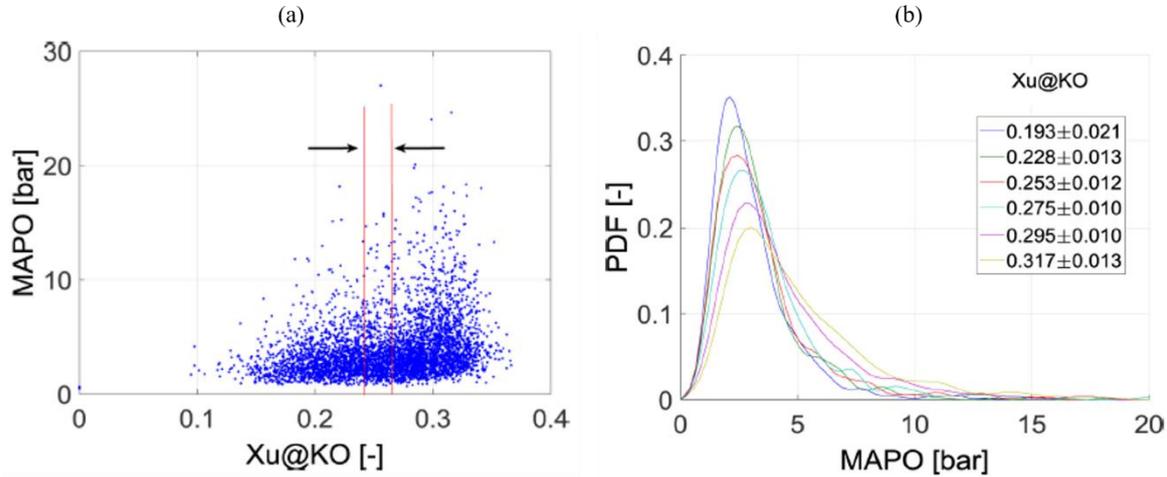


Figure 1.17: (a) Knock intensities displayed with corresponding unburned mass fraction at the knock onset and (b) PDFs for different bin width of unburned mass fraction (adapted from [1.75]).

Such analysis indicates unburned mass fraction is not a good indicator of knock intensity. At low values of  $X_{u@KO}$  there is just a low probability of high knock intensity and at high values of  $X_{u@KO}$  there is a possibility of a wide range of knock intensity, including high values.

As stated before, detailed literature review shows that mechanisms of damage induced by knock on the combustion chamber components are not currently fully understood and there is not a clear and robust methodology for knock intensity modelling that counts the effects of all influent engine parameters on the entire engine operating field and that is compatible with real time simulation. Described investigation is particularly useful to highlight novel advances achieved by the author research group during PhD course.

In this chapter, author describes innovative methodologies for knock intensity evaluation and modelling, providing two extremely powerful tools that are the key components of the combustion control system presented in the following chapter (1.4). Such models completely redefine the way with which knock intensity is evaluated and modelled, introducing, respectively, a control-oriented real-time-compatible piston damage model and a very simple analytical formula for the knock intensity calculation. The simple analytical approach allows therefore to manipulate the equation in order to reverse the model and directly determine the SA needed to target a specific value of a MAPO percentile. Some combustion indexes are needed, and this means that an indicating system is required for a possible on-board application. Nevertheless, the use of piezoelectric cylinder pressure sensor is very common at the engine test bench and some combustion indexes are therefore becoming available for on-board applications. Indeed, these days there are solutions suitable for on-vehicle applications and several manufacturers are proposing sensors that couple the low cost with the reliability that make the product compatible with production. Moreover, the recent trend of research in the combustion control system field is leading to develop sophisticated algorithms that can directly manage the combustion phase, using a measure as feedback, in order to better control the combustion efficiency for all operating conditions and during the entire life of the vehicle.

The analytical approach is finally extended for the main combustion indexes modelling, such as the crank angle for which the 50% of fuel mass is burnt (MFB50), the maximum in-cylinder pressure (P<sub>MAX</sub>), indicated mean effective pressure (IMEP) and maximum amplitude of pressure oscillation (MAPO) induced by knock. In the end, an analytical engine simulator (ANESIM) is discussed and implemented in Simulink environment. Such tool guarantees an extremely accurate feedback about these combustion indexes and provides the sensitivity to the SA and, more generally, to the combustion phase. It is then proposed as a valid engine simulator when a novel combustion control system must be developed and validated. In fact, considering that such kind of control system is executed by the ECU once per cycle, the extremely low computational power needed to run such simulator allows to test the controller with an execution frequency close or higher than the combustion one. This means that the controller components that have to be calibrated according to system time-based response (such as the closed-loop chain) can be tuned with values that have to be just slightly adjusted during the final online calibration phase.

However, these models have been individually validated and then implemented in a control strategy that is able to convert a target piston damage into a target combustion phase, though a robust open-loop chain.

First section (1.3.1) deals with the brief description of a control-oriented knock induced piston damage model. Such model was developed by the colleague Nahuel Rojo and described in his PhD thesis [1.76]. Nevertheless, topic is just introduced and adapted for this document because it is implemented in the combustion controller described in chapter 1.4.

### 1.3.1. Control-Oriented Knock-Induced Piston Damage Model

Piston damage has to be evaluated counting the effect of the piston temperature, that change the resistance of the piston alloy to be attacked by the knock induced mechanical stress. Figure 1.18 displays the decay of aluminum alloy hardness as a function of exposure time at different mean temperatures.

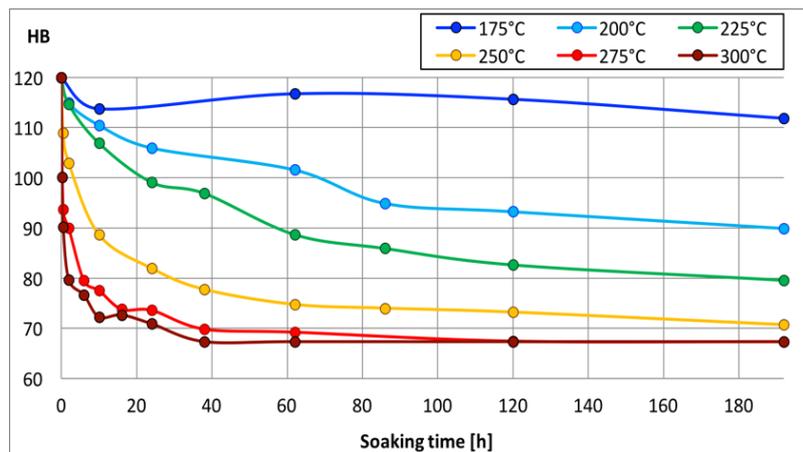


Figure 1.18: Hardness-temperature-time (HTT) curves for the aluminum alloy

Thus, the development and calibration of a semi-physical piston temperature model for aluminum pistons (mounted in the 8-cylinder engine used for these tests) precedes the discussion of the damage model.

Experimental tests are carried out on an 8-cylinder, 3.9 liter GDI Turbo-Charged engine and with two piston sets of 4 cylinders, that have been damaged and analyzed to collect data needed for the model calibration. It is characterized by high specific power, and main features are collated in Table 1.2.

Table 1.2: 8-cylinder engine characteristics

Displaced volume	3.9 L (8 cylinder)
Stroke	82 mm
Bore	86.5 mm
Connecting Rod	143 mm
Compression ratio	9.45:1
Number of Valves per Cylinder	4

Main combustion indexes were calculated from the in-cylinder pressure signals: CHAMPION charge amplifier and OBI-M2 indicating system by Alma Automotive were used for signal conditioning and acquisition, respectively. Indeed, each cylinder was equipped with a piezoelectric transducer, and the signal was sampled with a frequency of 200 kHz. The calculation of MFB50, IMEP and Pmax uses the low pass filtered pressure trace. Instead, the same signal is band-pass filtered to calculate the MAPO knock index, defined by Equation (1.10):

$$MAPO = \max(|p_{filt}|) \quad (1.10)$$

Where  $p_{filt}$  is the band-pass filtered in-cylinder pressure signal, applied on an appropriate angular window. In general, MAPO can be calculated using a high-pass or a band-pass filter and frequencies under 3 kHz are typically excluded. Band-pass filters are applied to neglect the combustion noise and usually the higher threshold is close to 30 kHz. The cut-off frequencies used in this case are within this range (3-30 kHz), but precise values cannot be disclosed for confidentiality reasons. Other combustion indexes, such as MFB50 and PMAX, are calculated from the low-pass filtered pressure trace, with a cut-off frequency of 3kHz.

Engine is operated under high engine speed and load conditions, targeting a high fixed knock level to damage pistons with a constant knock intensity (evaluated as a MAPO percentile). Only 4 cylinders at a time are operated and knock intensity is controlled independently for each cylinder, managing the SA with a closed loop controller (that uses the feedback of the indicating system as input) implemented in RCP system that communicates with ECU and indicating system via Controller Area Network (CAN) communication protocol. For each cylinder RT machine transmits SA corrections with respect to the mapped value that are read by ECU and receives main engine parameters (engine speed, load and lambda) and combustion indexes from the engine control unit and OBI-M2 system respectively. Figure 1.19 shows the functional layout of the communication loop that is used to test and validate combustion controller.

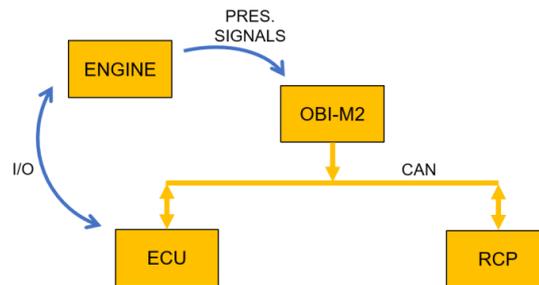


Figure 1.19: Functional layout of communication network.

Tests are performed on both engine banks using fuels with different RON value. Both the analysis of in-cylinder pressure history and piston damage allow to develop a predictive piston damage model that is compatible with RT execution. Inputs are the engine speed, P<sub>MAX</sub> and MAPO, while output is the instantaneous damage speed, calculated as the cyclic damage induced by knocking event, divided for the time of the engine cycle. Layout of damage model is shown in Figure 1.20 where logic blocks are reported.

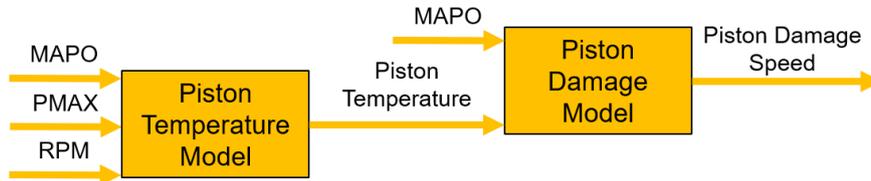


Figure 1.20: Description of piston damage model with logic blocks.

On one hand, the analysis of the in-cylinder pressure history consists of evaluating the 90<sup>th</sup> and the 99.5<sup>th</sup> percentile of cyclic P<sub>MAX</sub> and MAPO, respectively to collect data needed for the model development and calibration. On the other hand, piston analysis has been carried out performing micro-Vickers hardness tests on the piston crown, considering 5 points of measurements in the intake side and 5 in the exhaust side, as shown in Figure 1.21 (a). Deep visual investigations were conducted with a digital microscope to determine the damage morphology in the exhaust valve reliefs, that is particularly sensitive to knocking erosion damage. Finally, roughness tests were carried out to quantify the visible damage in the exhaust valve reliefs, as displayed in Figure 1.21 (b).

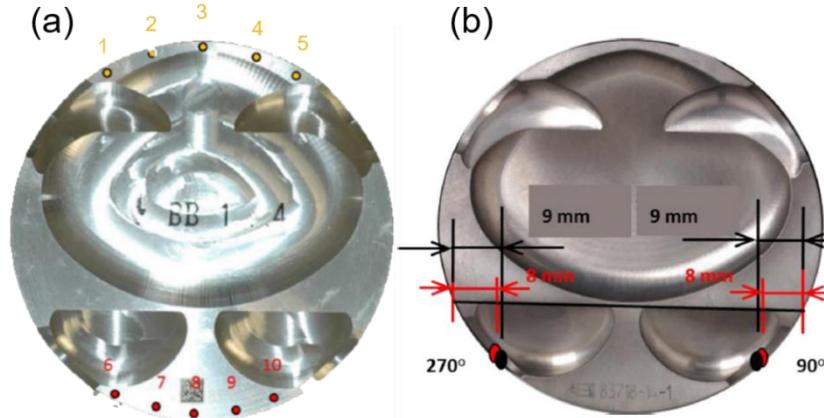


Figure 1.21: (a) Location of micro-Vickers hardness tests on pistons crown: measurements 1-5 have been acquired in the intake side, while measurements 6-10 in the exhaust side. (b) Sites of interest for roughness measurements, indicated by red and black ellipses in the exhaust valve reliefs.

Performing tests with different fuels (RON 91 and RON 95 gasoline) it is possible to separate the contributions of the heat transfer induced by combustion process and the additional amount due to knock. This evaluation is important to accurately determine piston surface temperature and then, the piston erosion. Indeed, if two cylinders are controlled to have different knocking levels, they will experience different wall thermal load (as well as P<sub>MAX</sub>), regardless of knock intensity. Consequently, differences on measured thermal effects on pistons (residual hardness) are not fully attributable to knock, because they depend on both different average P<sub>MAX</sub> and knock intensity. Thus, the combustion-related and knock-related heat transfer rate cannot be separated moving the spark timing only. The use of fuels with different RON number, allows for example to obtain the

same P<sub>MAX</sub> level by applying the same SA with different knock intensity, and vice versa. Moreover, the proposed method is needed also to quantify and calibrate the temperature role in the damage mechanism by controlling, for instance, different piston temperatures but the same knock intensity. If, for example, pistons subjected to the same knock intensity show different induced damage levels, the piston temperature role in the damage mechanism is confirmed and it can be understood and modelled. Such effect is displayed in Figure 1.22. Figure 1.22 (b) and (a) reports the 90<sup>th</sup> percentile of maximum pressure and knock intensity (estimated in terms of MAPO 99.5<sup>th</sup> percentile) as a function of the actuated spark advance variations from calibrated value. As well known, knock intensity is strongly dependent also on the adopted fuel RON number. Figure 1.22 (b) shows the linear dependence of the mean value of maximum pressure with the spark advance. As expected, the maximum pressure level dependence on the spark advance is not affected by the adoption of different fuels. However, maximum pressure levels change significantly between the two tests.

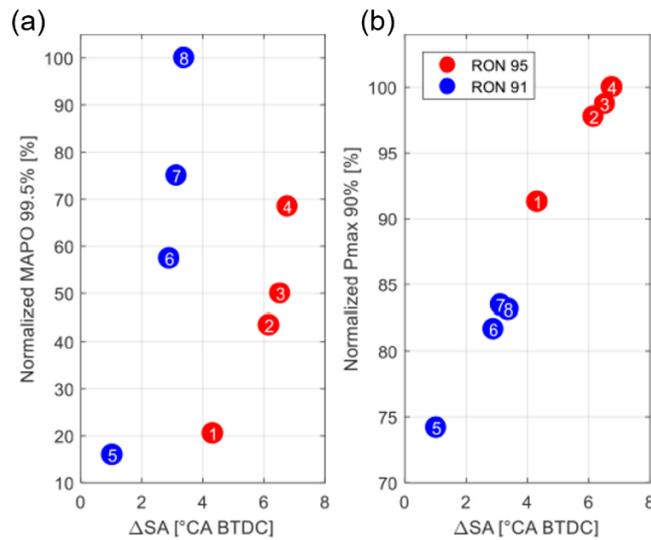


Figure 1.22: (a) Normalized knock level (average MAPO 99.5%) and (b) maximum pressure level (average P<sub>max</sub> 90%) depending on the average applied  $\Delta SA$ , w.r.t. the mapped value. Numbers indicate the cylinder.

Both sets of tests performed with RON 95 (cylinders 1-4) and RON 91 (cylinders 5-8) gasoline have been conducted at the engine point characterized by engine speed equal to 6000 RPM and load of 2300 mbar of air pressure, measured within the intake manifold. Target levels of knock intensity are different for each cylinder and for both engine banks a cylinder has been controlled to operate without knock, to have a reference condition. Hence, Figure 1.22 allows to collect normalized knock intensity and maximum in-cylinder pressure for each test, w.r.t to the applied  $\Delta SA$ .

Mean piston surface temperature can be estimated by measuring residual hardness after tests and using the hardness-temperature-time curves. Normalized values are shown in Figure 1.23 (a) and it is observable that the intake side is generally hotter than the exhaust one. This phenomenon is related to piston geometry and cooling conditions. Moreover, pistons 1 and 5, which operated in knock-free conditions, describe the mean nominal temperature difference between the two sides. In the other cases, the temperature difference between intake and exhaust tends to be smaller with knock intensity, even if such trend is not strictly monotonic. Instead, Figure 1.23 (b) reports mean temperatures in the P<sub>MAX</sub>-MAPO normalized domain, where the effects of knock on piston temperature clearly emerge. Piston 7 and 8, for instance, have the same maximum pressure but the temperature difference is coherent with different knocking levels. On the contrary, pistons 3 and 4 have different maximum pressure and knock intensities but practically the same temperature. Such evaluations about mean piston temperatures have been obtained averaging 5 measurements for each

piston side (intake and exhaust) and each area is characterized by a certain spatial temperature non-uniformity. This introduces a certain degree of uncertainty, and at the same time it demonstrates the relatively low knock intensity effects on average piston temperatures, also due to the low frequency of knocking events (the knock intensity is associated to one combustion event every 200, i.e. 99.5<sup>th</sup> percentile).

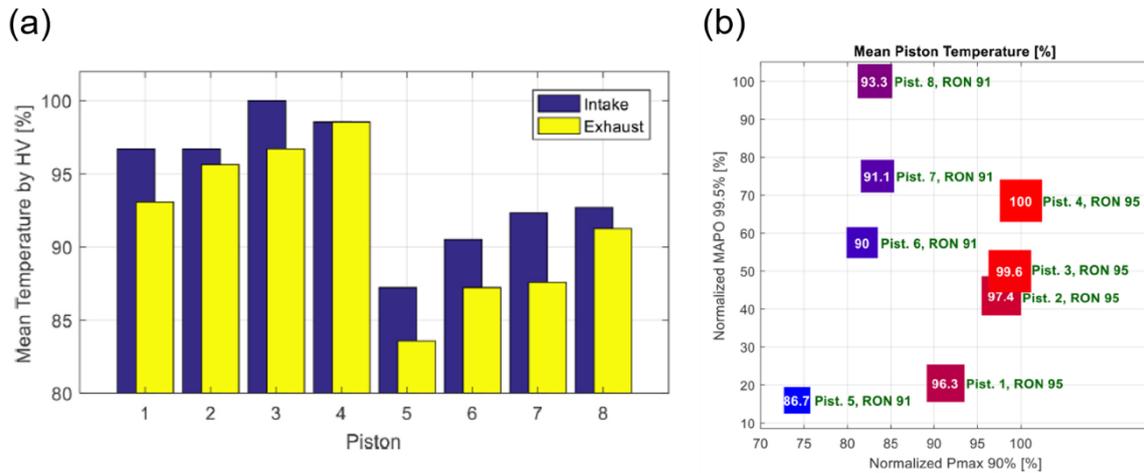


Figure 1.23: (a) Temperature values (normalized w.r.t. the maximum value) for pistons belonging to both set 1 (#1-4) and set 2 (#5-8). Temperatures have been calculated starting from the average residual hardness values and the HTT curves characteristic of this specific piston alloy. (b) Mean piston temperature (normalized w.r.t. the maximum value) depending on in-cylinder condition in terms of knock level and maximum pressure level.

Surface piston erosion has been evaluated with observations through the naked eye, 3D digital micrographs of the exhaust valve reliefs. On the basis of visual investigations, the most damaged valve relief was selected for each piston, and a qualitative erosion index was then assigned to the corresponding piston. However, roughness measurements are needed to quantify the observed level of erosion and they have been carried out on the most damaged piston valve reliefs. The mean roughness depth RzD was selected to quantitatively evaluate the erosion damage and according to DIN-4768 standard, RzD index is in fact calculated by determining the average maximum peak-to-valley distance within five successive sampling lengths. Results are collected in Figure 1.24 (a). Under the threshold of  $RzD = 4\mu\text{m}$ , no erosion has been observed by visual examination, while a significant growth in roughness values is clearly associated to the most severely damaged pistons #4 and #8. It can be therefore inferred that the roughness measurements are well related to the perceivable level of erosion. Figure 1.24 (b) reports erosion levels in the normalized piston temperature - knock level domain. Exhaust side temperature has been considered since exhaust side is the only one reporting damage. As expected, erosion damage increases with knock intensity level, and such damage seems to have a non-linear dependence with knocking level: below a certain threshold little or no damage is measured (pistons 1, 2, 3, 5 and 6), while a wide increase in erosion damage corresponds to a relatively small increase in knocking level (comparison between pistons 7 and 8, and between 3 and 4), notwithstanding that the adopted scale is subjective and that the statistical sample (number of pistons) is small. Thus, piston surface temperature plays a significant role in producing piston damage and this is evident considering pistons 4 and 7, which operated at the same knocking intensity but recorded different erosion levels, that is due to different piston temperature. To be more accurate, reported erosion values should be normalized w.r.t. the operation time (different for the two tests), assuming the hypothesis that damage cumulation is linearly time dependent. For this reason, in Figure 1.24 (b), pistons 5-8 damage corresponding to piston 1-4 operation time has been reported in brackets (e.g. if piston 8 were operated for 10 hours, instead of

7.5 hours, its damage level would have been 13.3 instead of 10). Even with this assumption, but with weaker evidence, piston temperature appears to be the only responsible for different erosion levels encountered at equal knocking intensity. Consequently, the higher the specific power is (and so the piston temperature), the lower the knock level to be considered “safe” for the engine components. This is an important element, to be considered in modelling knock damage, and in general for engine calibration. Usually, knock threshold used for SA calibration increase with engine speed, because of the increasing high-frequency content in the regular, non-knocking in-cylinder pressure waveform, due to faster combustion. On the other hand, piston temperature generally increases with engine speed, because of an unbalance between the thermal power exchanged with the walls (and the piston), and the power removed from the piston by the cooling system. Finally, “real” knock-safe thresholds should mainly depend on piston temperature.

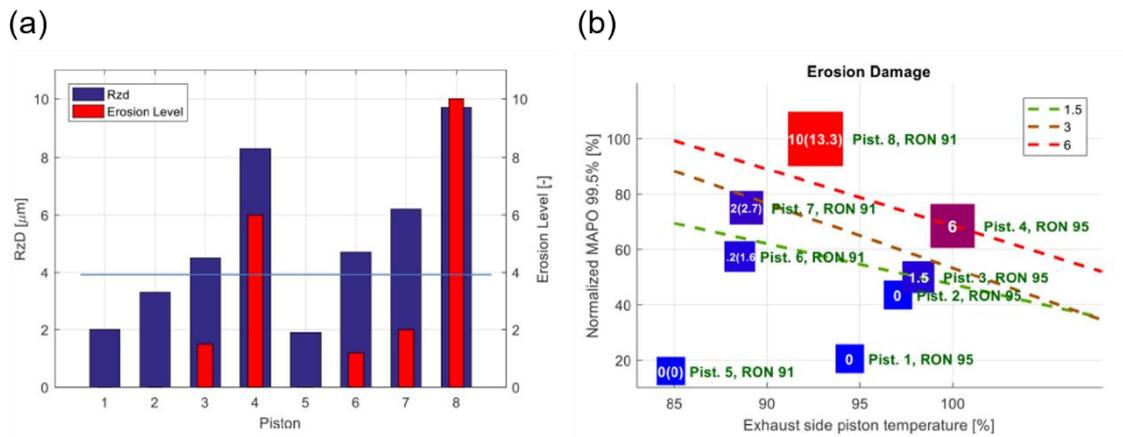


Figure 1.24: (a) Comparison between roughness parameter RzD and erosion level, for all tested pistons. (b) Erosion damage level depending on exhaust side piston temperature and knock level. For each piston, assessed damage level is reported in the square, which size and color is coherent with the damage (low damage level → small blue square, high damage level → big red square). Dashed lines represent constant damage levels.

With this approach, i.e. by using different fuels, by conducting stationary tests, by measuring residual hardness and induced knock damage, it is theoretically possible to calibrate both the piston temperature and the damage models, with statistically relevant data. It is important to mention that results reported in this chapter are just a portion of the complete database collected to have a reliable statistical base for the calibration process and that several sets of pistons have been operated and analyzed with this approach.

As highlighted above, post-operation hardness measurements are available for every piston, 5 for each side. In Figure 1.25, the measured (normalized) hardness values are reported for 3 pistons, showing non-uniformity of the temperature field, both on the intake and on the exhaust side. Location #3 corresponds, for both sides, to the measurement made on the thrust-antithrust axis. As it can be observed, the temperature range is relatively wide but, at the same time, within every side, 5 measurements are symmetrical w.r.t. the thrust-antithrust axis, as shown in Figure 1.26. The symmetry can be used to average the measurements. This way, as described on the right side of Figure 1.26, for every piston it is possible to consider 6 resulting hardness values, of which 4 are averaged, while the remaining 2 are on the thrust-antithrust axis. This approach allows to significantly simplify the piston temperature model, without excluding the spatial temperature range on the piston crown.

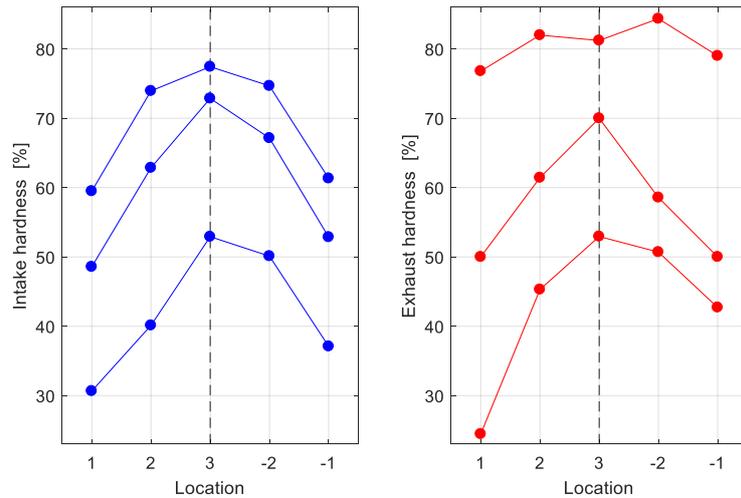


Figure 1.25: Measured hardness field within the intake and the exhaust side for 3 pistons.

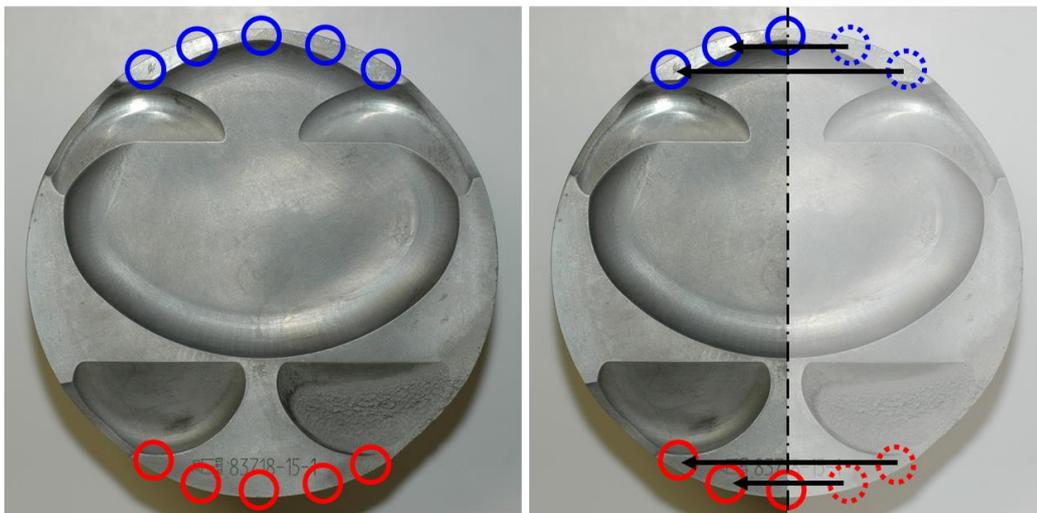


Figure 1.26: Hardness measurements symmetry, used to average correspondent values.

The temperature model has been developed according to the following hypothesis:

1. Even if the heat transfer rate on both sides of piston (the combustion chamber and the oil side) changes order of magnitude within the cycle, the skin piston temperature can be supposed almost constant during the cycle, due to the high conductivity and the specific heat of the material. Considering such observation and the control-oriented purpose of this approach, the model is executed once per cycle.
2. The exchange condition between gas and the piston crown is supposed uniform on the entire surface. The specific heat flux is imposed on the piston surface and it is empirically defined as a function of engine speed, P<sub>MAX</sub> and M<sub>APO</sub>, that are also the inputs of the model (Figure 1.20). This means that the heat transfer is not depending on the surface temperature, which is not rigorous, but it should be negligible because the piston temperature is within a relatively small range compared to the gas space-averaged temperature.
3. The oil jet cooling is the only mechanism considered (it represents the 50-60 % according to [1.77]), so that the heat exchange through the ring belt and the skirt is neglected. These contributions are not negligible, but their calibration would require, for instance, to

conduct tests with different coolant temperature (and to make the corresponding hardness measurements). The conductive heat transfer is very complex and difficult to investigate with specific experimental tests and for this reason it is simplified supposing the piston bottom inner surface is involved in the heat exchange process and the most of such process takes place in the surface directly touched by the oil jets. Indeed, as demonstrated by the hardness measurement, the higher the distance between the considered area and the oil jets, the higher the temperature. Figure 1.27 shows the described method for the estimation of the distance between different points on the piston surface and the area touched by the oil jets.

4. The ratio between the gas-exposed and the oil-impinged areas is unitary, which means that under stationary conditions the specific heat flux across the thickness is uniform, and the heat flux is mono-dimensional. Substantially, it is equivalent to a flat plate, but with differential thermal conductivity (because of the varying distance between the two sides in the real piston).

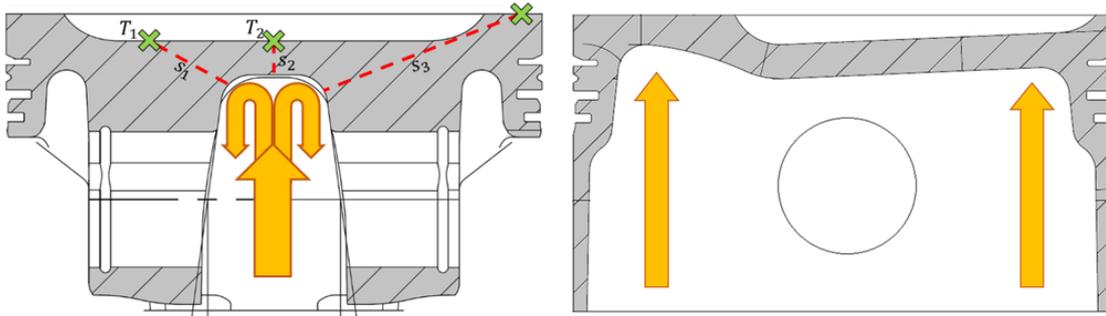


Figure 1.27: Qualitative oil jets position and estimation of the measurement point distance from the oil-impinged surface.

The result is a non-stationary finite-elements model, where the nodes are distributed along the thickness of the piston and an equivalent thickness is defining the location. The boundary conditions are the same for every location (thickness) considered:

- imposed heat flux on the gas side (Neumann condition)
- conductive heat transfer on the oil side (Robbins condition)

The temperature in the nodes  $T_t$  at time  $t$  are calculated by:

$$T_t = MM \setminus FF \quad (1.11)$$

With  $MM$  and  $FF$  are defined by Equation (1.12) and (1.13):

$$MM = \frac{MC}{ts} + \alpha (MK + MH) \quad (1.12)$$

$$FF = \left[ \frac{MC}{ts} - (1 - \alpha)(MK + MH) \right] T_{t-1} + F \quad (1.13)$$

Where:

$\alpha$  defines the method: 0, 1, 0.5 for explicit, implicit and Crank-Nicholson, respectively

$ts$  is the time-step

$T_{t-1}$  is the temperature vector at the previous time-step  $t - 1$

$MC$  is the capacity matrix

$$MC = \frac{\rho c l}{2} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.14)$$

Where  $\rho$ ,  $c$  and  $l$  are the density, the specific heat and the elements length, respectively.  $MK$  is the conductivity matrix:

$$MK = \frac{\lambda}{l} \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix} \quad (1.15)$$

Where  $\lambda$  is the material conductivity.  $MH$  is the convective matrix:

$$MH = H_{oil} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.16)$$

Where  $H_{oil}$  is the convective coefficient at the oil-piston boundary.

$F$  is the vector of the loads:

$$F = \begin{pmatrix} q_{pist_t} \alpha + q_{pist_{t-1}} (1 - \alpha) \\ 0 \\ 0 \\ 0 \\ H_{oil} T_{oil} \end{pmatrix} \quad (1.17)$$

$q_{pist_t}$  is the specific heat flux at the gas-piston boundary at time  $t$ ;

$q_{pist_{t-1}}$  is the specific heat flux at the gas-piston boundary at time  $t-1$ ;

$T_{oil}$  is the oil temperature.

$H_{oil}$  is an unknown model parameter, such as  $q_{pist}$ . The heat transfer problem is represented in Figure 1.28 and it needs to impose  $H_{oil}$  or  $q_{pist}$  to be solved. In the figure, piston and the oil-piston interface are two thermal resistance connected in series.

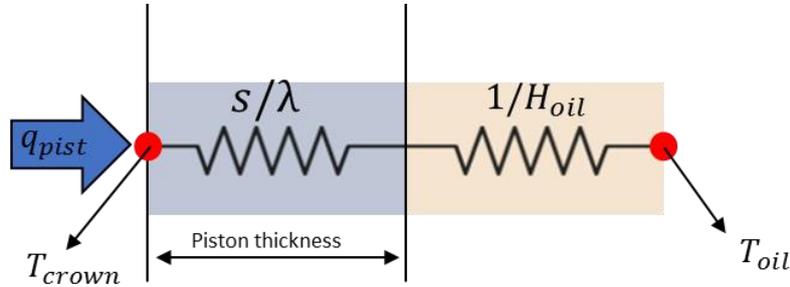


Figure 1.28: Schematic representation of the heat transfer problem through thermal resistances series.

In stationary conditions,  $H_{oil}$  can be calculated through Equation (1.18) from piston temperature, piston thickness  $s$  (evaluated by analyzing the piston geometry) and specific power  $q_{pist}$  (defined below):

$$H_{oil} = \frac{1}{\frac{T_{crown} - T_{oil}}{q_{pist}} - \frac{s}{\lambda}} \quad (1.18)$$

By analyzing data recorded with NTC (Negative Temperature Coefficient) technology, for the same engine, an estimation of  $H_{oil}$  has been calculated. In particular, the data used contains for different engine speeds:

- 4 piston crown temperature measurements
- Total power transferred to the water  $Q_{wall}$ , calculated on inlet-outlet water temperature difference and flow
- Oil temperature

By assuming the value of  $q_{pist}/q_{wall}$ , (i.e. the fraction between the heat transferred from the gas to the piston and the total power transferred to the water, including friction losses) and  $\lambda$ , and by estimating the values of  $s$ , 4 independent trends of  $H_{oil}$  can be calculated as a function of the engine speed, as described in Figure 1.29. The fraction between  $q_{pist}$  and  $q_{wall}$  has been chosen equal to 0.2. This hypothesis is not verified, but it is necessary to have an estimation of the heat transfer rate. An error on this parameter will affect all the following results, but not the applied approach.

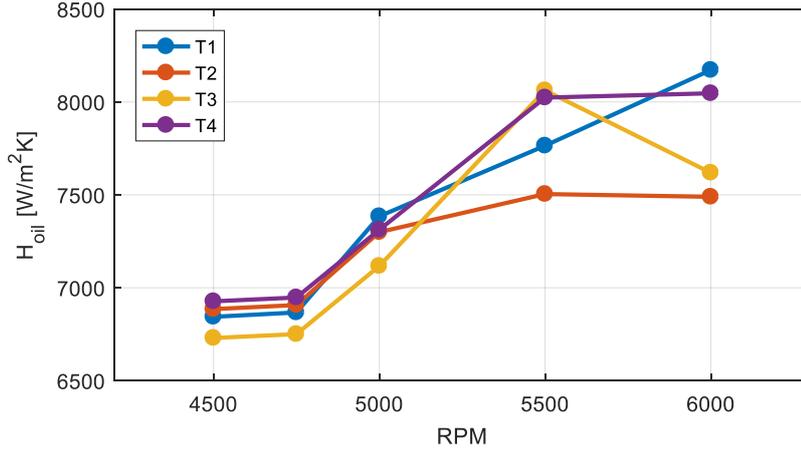


Figure 1.29: Convective coefficient estimations considering 4 different temperature measurements, for different engine speeds.

The 4 estimations are very similar, both in terms of values and trends, confirming that the 1-D approach is feasible. The apparent dependence of the 4 convective coefficients on the engine speed visible in Figure 1.29, can be a consequence of the simplified approach, while the ratio  $q_{pist}/q_{wall}$  is probably varying with the engine speed as well. Because of this,  $H_{oil}$  has been supposed independent on engine speed and equal to  $7320 \text{ W/m}^2\text{k}$  (referred to the piston crown surface, being 1 the ratio between top and bottom surfaces).

As already mentioned, the mean value of the specific heat flux within the combustion cycle,  $q_{pist}$ , is imposed on the gas side and it is estimated by a linear correlation, to be calibrated, with the maximum pressure of the cycle. This is supposed to be valid in non-knocking combustions, while an additional contribution is given by knock, but through a localized effect. For this engine and for the investigated operating conditions, the damage was observed in the exhaust side, so it is assumed this is the location where knock occurs. Consequently, the knock-related heat transfer contribution is to be considered only in the exhaust side. Thus, the specific power model,  $q_{pist}$  ( $P_{max}$ , RPM, MAPO), is defined as a linear function of  $P_{max}$  and RPM; moreover, for the exhaust side, it is multiplied by a term linearly increasing with MAPO:

$$q_{pist} = (a P_{max} + b \text{RPM} + c)K_{MAPO} \quad (1.19)$$

Where:

$$\begin{aligned} K_{MAPO} &= 1 && \text{in the intake side} \\ K_{MAPO} &= \max(\text{MAPO} - 4, 0)d && \text{in the exhaust side} \end{aligned} \quad (1.20)$$

And  $a, b, c, d$  have to be identified.

$K_{MAPO}$  definition is thought as a line with positive slope, passing through  $\text{MAPO} = 4$  and saturating at 0. The meaning is that cycles with a MAPO value lower than 4 (an arbitrary threshold) are not increasing the nominal  $q_{pist}$  value, while higher values of MAPO are amplifying such amount. Figure 1.30 gives a qualitative idea of the resulting function.

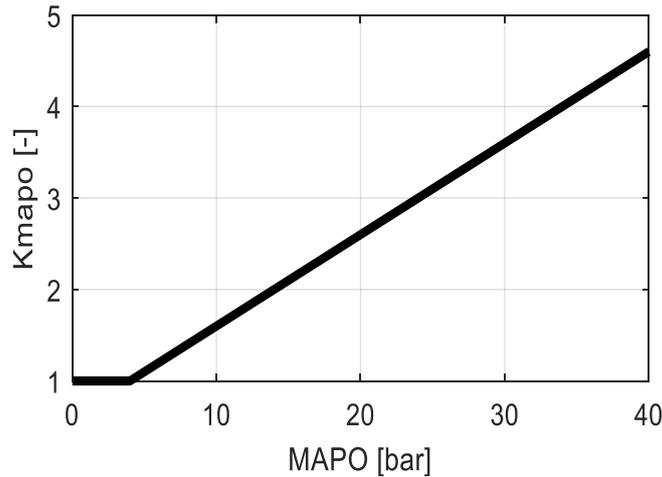


Figure 1.30: Heat transfer enhancer depending on knock intensity.

In Figure 1.31 the calibration process is summarized. The combustion indexes (Pmax and MAPO) recorded during tests at the bench are vectors describing the whole pressure history (one value for every combustion cycle) experienced by the piston. The piston temperature model estimates the temperature history (one value for every cycle) for every measurement point. Then, the material hardness decay model (the HTT curves seen above) estimates the hardness history for every measurement point, using data about cumulative exposure time and mean temperature, and finally the residual hardness values are compared to the measured ones. This applies in parallel to every tested piston.

To minimize the cost function, which is defined as the sum of all the absolute errors, for every piston, for every measure point, `fminsearch` function from the MATLAB optimization toolbox [1.78] has been used.

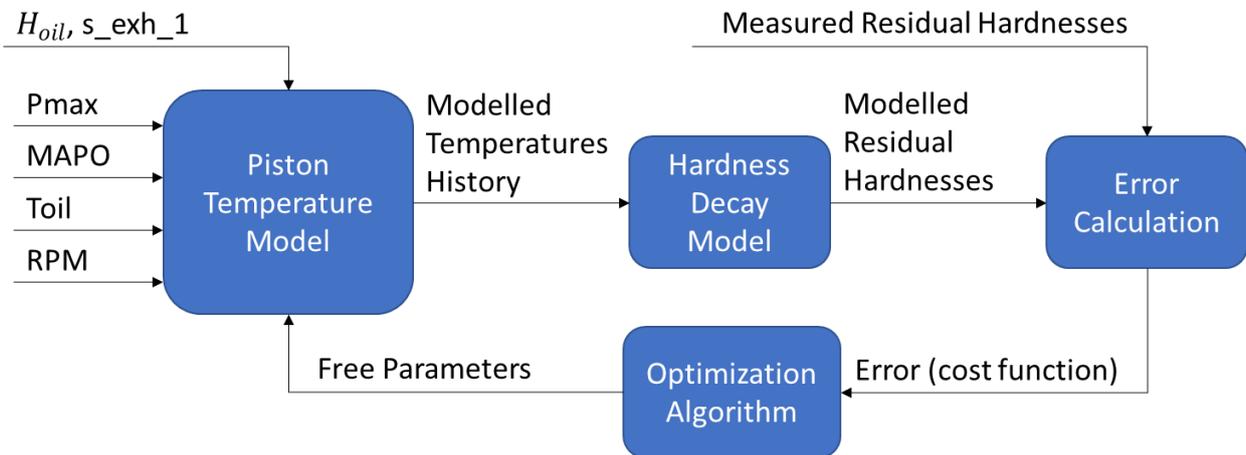


Figure 1.31: Calibration process for the piston temperature model.

As showed in Figure 1.20, knock damage model inputs are MAPO, representing the mechanical stress acting on piston surface, and piston surface temperature, from which piston alloy resistance strongly depends. It can be reasonably assumed a linear dependence between measured knock intensity and some representative stress value since it is very difficult to accurately estimate mechanical forces on piston surface due to knock. As better explained in [1.77], knock model assumes that there is a knock threshold, definable as a function of piston surface temperature, beyond which the material deformation falls in the alloy plastic field. Such model practically consists

in defining and calibrating mentioned threshold. The real meaning of such knock intensity threshold can be better explained through Figure 1.32, where the y-axis (stress) has been replaced by normalized knock intensity index values, and the x-axis (where permanent deformation can be achieved) is a damage scale, whose values are intentionally hidden as they have no physical meaning.

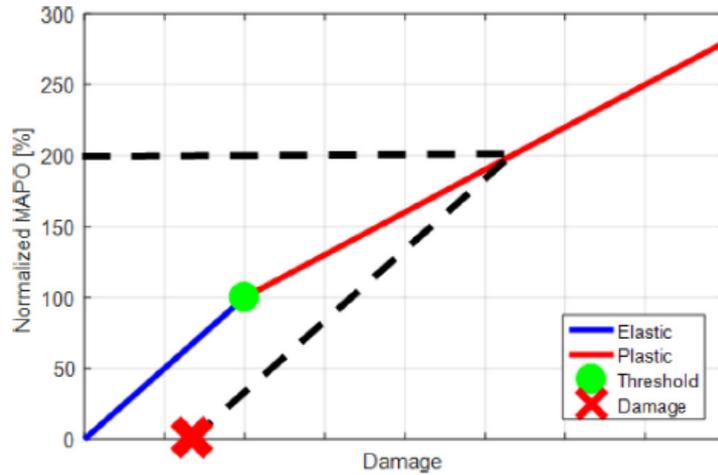


Figure 1.32: Material damage simplified model.

Temperature dependence is obtained by moving plastic field boundary (i.e. knock index threshold), while the plastic slope is assumed to be constant. The total modelled damage is given by the summation of the corresponding damage of every cycle. The model calibration consists in the identification of 2 parameters (the gain and the offset of linear MAPO threshold) and this means the problem is undetermined and potentially only the negative slope of the function can be determined. Nevertheless, since the linear dependence between the induced damage and the piston temperature was already partly identifiable by considerations made on Figure 1.24 (b), a reasonable initial solution can be identified. The calibration algorithm is schematized in Figure 1.33 and it is very similar to the one adopted for the previous model. Model inputs are the measured MAPO and the piston temperature, updated with engine cycle frequency. The cost function (CF) is defined to maximize Pearson’s correlation,  $\rho$ , between the measured and the modelled damage  $D$ :

$$CF = 1 - \rho(D_{meas}, D_{mod}) \quad (1.21)$$

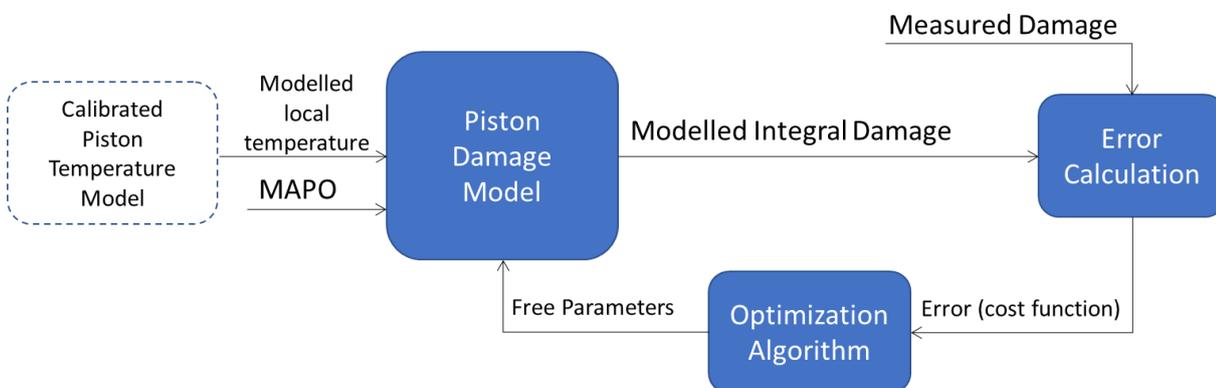


Figure 1.33: Damage model calibration process.

The best configuration found for the knock index threshold dependence on temperature is reported in Figure 1.34 (a). The adopted threshold map is shown in 1.34 (a), while the comparison between modelled index (whose absolute value has no physical meaning) and measured erosion is reported in 1.34 (b).

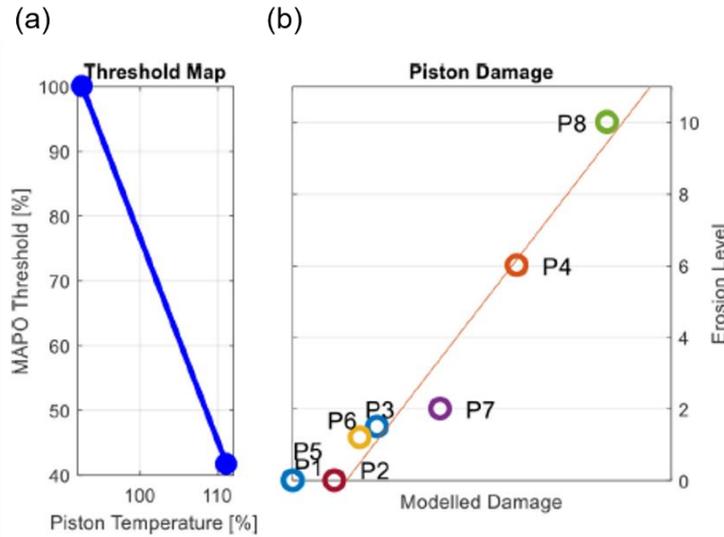


Figure 1.34: (a) MAPO threshold as a function of normalized piston temperature. (b) The corresponding cumulated modelled damage is compared with the erosion level, for all the pistons.

By comparing the MAPO value and the calibrated threshold for given piston temperature the excess over the threshold can be calculated for every cycle, estimating the induced damage in the piston exhaust valve reliefs. The calculated damage can then be expressed in *bar*.

At this point, supposing to implement such models in a SA control strategy to actively improve combustion efficiency by managing knock intensity, it is clear that the targeted damage, the corresponding knock intensity and engine performance has to be spread throughout engine life. Indeed, a condensed damage generation and the corresponding performance increase within a short time would be a very inefficient way of using this model. Thus, the control strategy target should be a constant damage speed, which finally generates the desired damage level upon completion of the design life cycle. Of course, the acceptable piston erosion has to be experimentally established by the engine manufacturer, verifying the maximum damage that does not compromise any engine functional parameter. Supposing a predefined engine life,  $Life_{tgt}$  (expressed in seconds) and a target modelled damage,  $Damage_{tgt}$  (expressed in *bar*), a damage speed target  $DS_{tgt}$  can be defined, as described in Equation (1.22):

$$DS_{tgt} = \frac{Damage_{tgt}}{Life_{tgt}} \left[ \frac{bar}{s} \right] \quad (1.22)$$

Similarly, the estimated instantaneous damage speed can be defined as:

$$DS_{est} = \frac{Damage_{est|P}}{P} \left[ \frac{bar}{s} \right] \quad (1.23)$$

Where  $P$  is the considered period and  $Damage_{est|P}$  is the cumulated damage within the period  $P$  estimated by the model. For the model implementation in the open-loop based combustion controller the period  $P$  is the engine cycle time, that for a 4-stroke engine can be calculated with Equation (1.24):

$$P = \frac{2 * 60}{RPM} \left[ \frac{s}{cycle} \right] \quad (1.24)$$

Hence, such model (piston temperature coupled with the MAPO threshold) can be used to directly measure the damage speed during the engine operation and to control knock intensity, dosing the desired final damage, which can be even null, throughout the defined lifetime of the engine. The variables involved in the model, can suggest how the targeted knock level and consequently the resulting engine efficiency, can be qualitatively influenced by the piston cooling conditions, piston geometry and high temperature material properties. However, the physical approach used to include the effects of such variables on the calculated piston temperature allows to calibrate the model by updating equations constant with values related, for instance, to the different piston geometry or alloy. Piston damage model is then used coupled with other tools described below to develop an innovative model-based adaptive combustion controller.

### 1.3.2. *Control-Oriented Analytical Knock Intensity Model*

As it is well-known, knock is a non-deterministic phenomenon and, for this reason, a statistical approach is needed to describe the phenomenon. MAPO probability function (PF) is characterized by a non-symmetrical distribution and it can be well-approximated with log-normal, Weibull or Gamma distributions. The first one is typically preferred, due to its few-parameters formulation and because its analytical equation is easily deducible from the Gaussian one, from which inherits numerous properties.

In this section, a statistical analysis of knock intensity is carried out and an innovative 0-D predictive knock model is defined. A part of results presented in this section have been already published by the authors in [1.79] and it represents one of the key goals of PhD course. The proposed model is based on a new empirical approach to directly calculate a MAPO percentile (which is the typical approach to evaluate experimental knock intensity) or even its whole log-normal probability distribution, without the need of simulating CCV, and by calibrating few parameters. In fact, such model uses only two mean combustion parameters to predict knock behavior.

Beyond the calibration effort, accuracy and predictivity, physical models require the calculation of many combustion cycles to extract a single synthetic statistical index (such as a percentile value), which is usually the desired output. Instead, the proposed approach totally bypasses the need of a statistical basis, focusing and calculating directly the desired statistical index as a function of static operating parameters. This aspect makes this model very attractive for its real-time control-oriented implementation.

The experiments have been performed on two different GDI TC engines. The first one is a 4-cylinder engine, while the second one is a high-performance 8-cylinder engine. Using the experimental database collected with the 4-cylinder engine a statistical analysis of knock intensity has been carried out. Several distributions are normalized and self-compared, and a crucial correlation between two MAPO percentiles is found. The knock model is described and the 98th MAPO percentile is calculated through a power function of the Trapped Air Mass (TAM) and maximum in-cylinder pressure ( $P_{max}$ , calculated on the low pass filtered in-cylinder pressure signal), and some observations about 50th percentile are carried out. Moreover, some parameters that influence such function are identified and the effects of engine speed and lambda variations are discussed. The knock model is then calibrated and validated for the 8-cylinder engine and the effects of the fuel quality have been introduced in the function. Finally, an application example of the knock model in

a 1-D simulation environment is described. The proposed empirical model is implemented in a user defined template and tested in a GT Power combustion model of the 4-cylinder engine.

### 1.3.2.1. Knock Model Development and Calibration for the 4-cylinder engine

The 4-cylinder engine characteristics are reported in Table 1.3. In-cylinder pressure signals were sampled at 200 kHz and knock intensity was measured using MAPO as index, filtering the pressure signal with a Butterworth high-pass filter with a 5 kHz cutoff frequency.

Table 1.3: 4-cylinder engine characteristics.

Displaced volume	1389.9 cc (4 cylinder)
Stroke	75.6 mm
Bore	76.5 mm
Connecting Rod	144 mm
Compression ratio	10:1
Number of Valves	4
Exhaust Valve Open	580° BTDC @ 0.1 mm lift
Exhaust Valve Close	356° BTDC @ 0.1 mm lift
Inlet Valve Open	358° BTDC @ 0.1 mm lift
Inlet Valve Close	132° BTDC @ 0.1 mm lift

The experimental tests consist of several SA sweeps, for given operating conditions. Engine speed and intake manifold pressure were controlled at constant values for each tested point, and the spark advance was increased (typically starting from the calibration value) until knocking conditions were achieved. Some SA sweeps have also been performed for different values of lambda to analyze its effects on knock tendency. Table 1.4 collects engine speed, intake manifold pressure, lambda, and fuel RON values for the tested engine points. For each point, a SA sweep has been performed.

Table 1.4: Tested engine points. For fixed RPM, intake manifold pressure, lambda ad fuel a spark sweep has been performed. The nomenclature  $x1:dx:x2$  means that the lambda value has been varied from  $x1$  to  $x2$  with  $dx$  increment.

Engine point	Speed [RPM]	Load [mbar]	Lambda	Fuel RON
1	2000	1130	1	98
2	2500	1500	1	98
3	2500	1600	1	98
4	3000	980	1	98
5	3000	1300	1	98
6	3000	1500	1	98
7	3000	1700	0.85:0.1:1.15	98
8	3000	2000	1	98
9	4000	1430	0.85:0.1:1.15	98
10	4500	1630	1	98
11	5000	1120	0.85:0.1:1.15	98
12	5000	1490	1	98

As already stated, MAPO values are characterized by a non-symmetrical distribution and log-normal, Weibull and Gamma parametric functions can well describe the statistical MAPO trend. The log-normal PF has the benefit that it is defined by a simple analytical function, which can be easily deduced from the Gaussian one. In other words, several properties of normal PF can also be used for manipulating and calculating parameters of MAPO curves. This approach gives the possibility to analytically describe the phenomenon in a very concise manner and, therefore, to adapt the function to different cases, by varying few parameters. In this way, a complete characterization of

knock intensity can be obtained for a given engine point, and percentiles values can be easily evaluated.

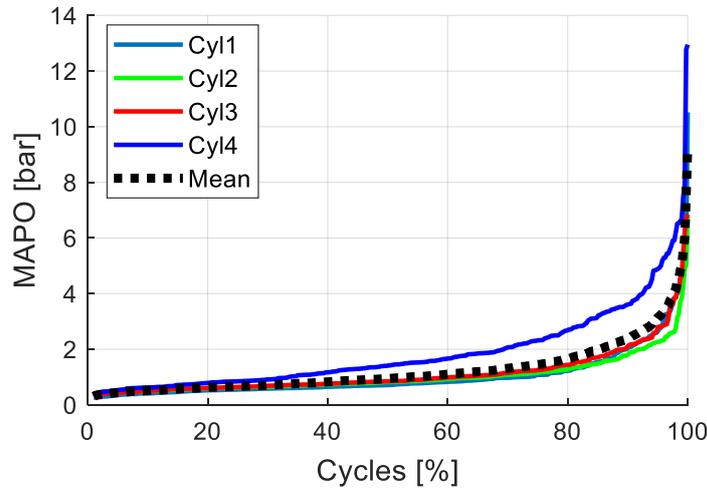


Figure 1.35: MAPO sorted values for fixed operating conditions. Each cylinder records different MAPO values.

In order to simplify the analysis of MAPO values, the experimental distributions of each cylinder have been compared, and the mean curves (corresponding to the mean cylinder) have been evaluated for each tested engine point. The MAPO probability distributions are typically distinct for each cylinder (Figure 1.35 shows an example of such non-uniformity), probably due to different filling levels, air-to-fuel ratios, temperatures, inlet turbulence conditions, and so on. Figure 1.36 shows mean MAPO PFs for fixed operating conditions and for different spark advance angles. For increasing SA, the mean value, and the standard deviation of the MAPO log-normal distribution significantly increase. An introduced above, some works analyze knock intensity with respect to the unburned mass fraction at knock onset and Gandhi in [1.75] explain the similarity of MAPO PFs when they are evaluated for fixed intervals of  $X_{u@KO}$ , after their normalization with respect to a large-enough percentile. In the mentioned literature the 95<sup>th</sup> percentile is taken as reference, but the same result is achievable also with other high-enough percentiles. But, for a wide range of SA variations, normalized MAPO curves are typically too different to identify a unique probability function (Figure 1.37).

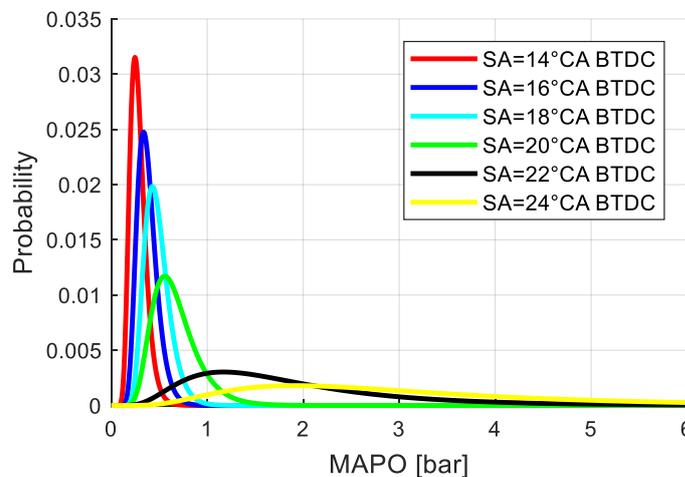


Figure 1.36: Mean MAPO PFs for different SA.

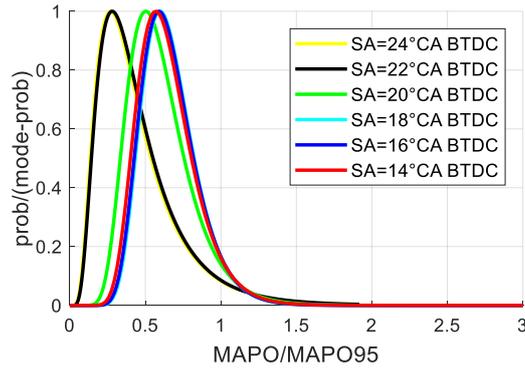


Figure 1.37: Normalized mean MAPO PFs with respect to 95th MAPO percentile (MAPO95) and the mode probability (mode-prob).

Given such preliminary remarks, the novel approach proposed in this activity leads to determine the MAPO PF by identifying two percentiles (one relatively high that can be arbitrarily chosen and the 50<sup>th</sup> percentile), and to use them to calculate the mean value and the standard deviation of the log-normal distribution for each engine point. For the higher one the choice fell on 98<sup>th</sup> (MAPO98) and 50<sup>th</sup> (MAPO50) MAPO percentiles, and a correlation between these values and some mean combustion parameters has been investigated. In particular, the 50<sup>th</sup> and the 98<sup>th</sup> percentiles are well related to each other, as shown in Figure 1.38. However, the same results are obtainable with other percentiles because this choice is somewhat arbitrary, even if, for the higher percentile, a larger value does not work well due to its higher variability. For different operating conditions (different speed, load and lambda) the 50<sup>th</sup> MAPO percentile can be calculated from the 98<sup>th</sup> one (supposing it is known) through a linear equation. In this way, knowing two percentiles, the PF mean value and the standard deviation can be identified and so the whole distribution. Figure 1.38 shows that also for very small values of the 98<sup>th</sup> percentile, the 50<sup>th</sup> one does not fall under 0.2-0.15, due to noise contribution. At the same time, it is possible inferring that when the 98<sup>th</sup> is zero (at the limit) also the 50<sup>th</sup> is null. Therefore, the correlation can be approximated through a linear function that passes through 0, with a small error increase. However, in the next part of this section the background noise will be related to the engine speed.

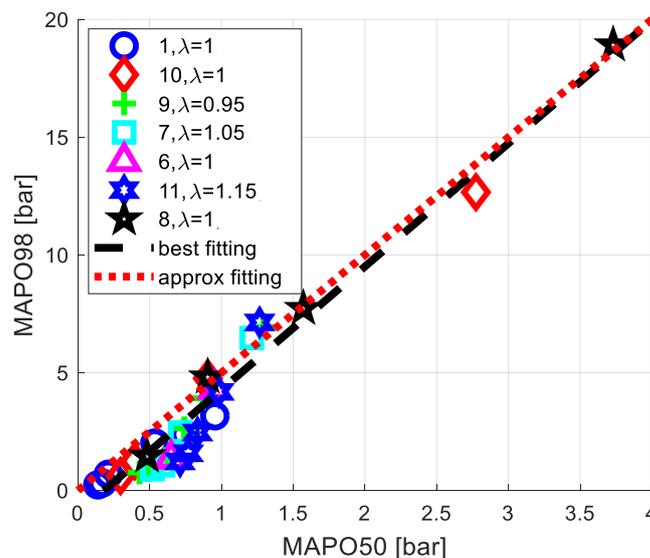


Figure 1.38: MAPO98 and MAPO50 for some tested engine points. Referring to Table 1.4, the legend provides the number of the engine point and the lambda value (fuel RON=98).

The proposed methodology involves that MAPO98 is calculated as a parametric function of two mean combustion indexes. An important objective is to use the lowest possible number of parameters to completely define the mentioned function. Some synthetic data, characteristic of engine operating point, are analyzed, to identify two independent variables that are well representative of the available chemical energy inside the combustion chamber and how such energy is converted into heat (since load and spark advance are the main control parameters responsible for knock occurrence). As first independent variable the stoichiometric trapped air mass (STAM) has been selected, which is equal to the TAM when the mixture is stoichiometric or rich, and to the ratio between TAM and lambda when the mixture is lean. Equivalently, it could have been considered the stoichiometric fuel mass. Such index is more representative of the effective engine load than intake manifold pressure and it does not derive from a calculation (like the cumulative heat release within the cycle, CHRnet). For this reason, it is a good choice also for a possible real-time implementation. The second independent variable has been chosen between the in-cylinder maximum pressure (Pmax), the crank angle for which the 50% of fuel mass is burned (CA50) and the difference between CA90 and CA10 (CA1090). The 98<sup>th</sup> MAPO percentile has been evaluated as a power function of all these parameters, in order to select the most influent on knock intensity. In Figures 1.39, 1.40 and 1.41, MAPO98 is evaluated with respect to the investigated combustion indexes for different engine speeds, loads, lambda and injected water masses. In Figure 1.39 it is possible to see that the variation of these parameters impacts on the values of maximum pressure for which a given MAPO98 is recorded, but it does not influence the shape of the relationship, and this is then the feature that is explored for a wide range of operating conditions. A similar behavior can be observed in Figure 1.40, where MAPO98 is reported as a function of CA50. On the contrary, Figure 1.41 shows that MAPO98 cannot be described as a power function of CA1090.

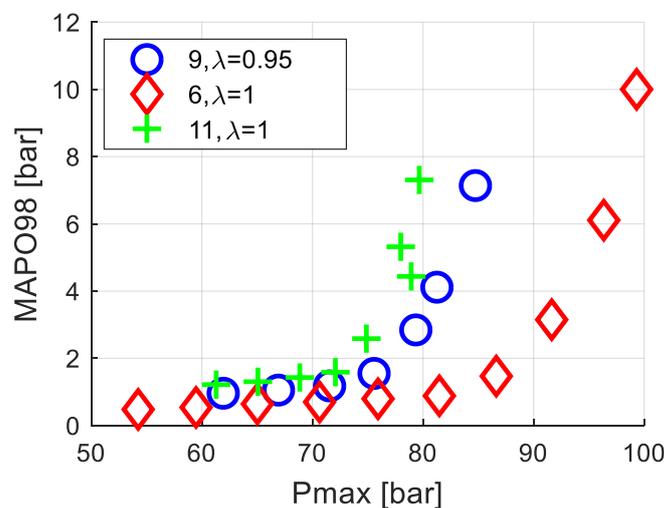


Figure 1.39: MAPO98 as a function of maximum in-cylinder pressure for some tested engine points.

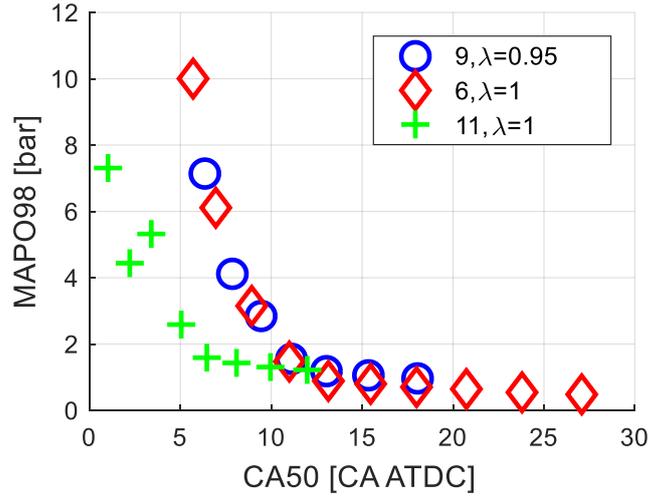


Figure 1.40: MAPO98 as a function of CA50 for some tested engine points.

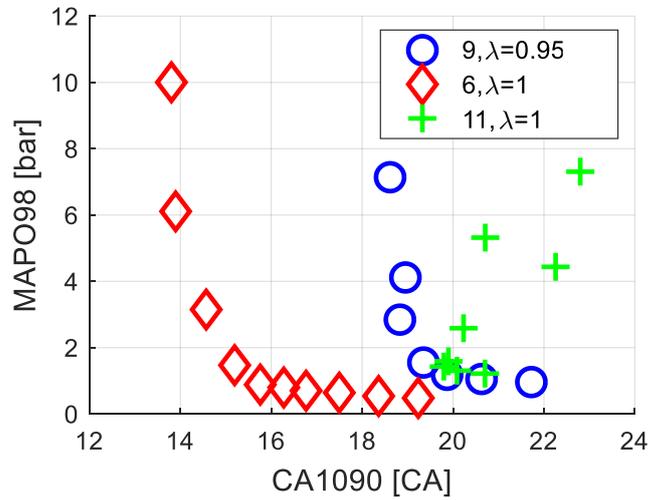


Figure 1.41: MAPO98 as a function of CA1090 for some tested engine points.

Trends represented in Figures 1.39 and 1.40 can be interpolated with a power function, like the one defined by Equation 1.25:

$$y = a x^b + c \quad (1.25)$$

Where  $y$  is the 98<sup>th</sup> MAPO percentile and  $x$  the maximum in-cylinder pressure or the CA50. To identify the optimum couple to fit the 98<sup>th</sup> percentile trend, different power functions of STAM-Pmax and STAM-CA50 are investigated, and the mean percentage error (one for each SA sweep) between experimental and calculated MAPO98 is evaluated. Equation 1.26 resulted as the five-parameters function that better fits experimental data:

$$mapo98 = (a x^b) y^{(c x+d)} + e \quad (1.26)$$

Where  $x$  is STAM and  $y$  is Pmax or CA50. It should be observed that Equations 1.25 and 1.26 do not represent physical relationships between the selected parameters. They have been chosen since they mathematically represent the best fit between MAPO98, STAM and CA50 or Pmax. Table 1.5 shows

the mean percentage relative errors between experimental MAPO98 and values calculated with Equation 1.27. The error is calculated with the following equation:

$$err = \left| \frac{MAPO98 - mapo98}{MAPO98} \right| 100 \quad (1.27)$$

Where  $MAPO98$  is the experimental 98<sup>th</sup> MAPO percentile and  $mapo98$  is the one calculated with Equation 1.26. To evaluate the errors, the optimization of five parameters is carried out for each considered variable couple. Table 1.5 reports both the mean relative error (%) and the mean absolute error (bar) between the calculated and the experimental knock level, both considering as independent variables STAM-Pmax and STAM-CA50. The table shows that best results are achieved by considering STAM-Pmax, even if the two parameters are not independent.

Table 1.5: Absolute and relative mean MAPO98 errors calculated using Pmax (2nd and 3rd column) and CA50 (4th and 5th column) as y variable in Equation 1.26. For each column, the minimum error is highlighted in blue, the maximum one in orange.

Engine point ( $\lambda=1, r=0$ )	Error with STAM-Pmax [%]	Absolute error with STAM-Pmax [bar]	Error with STAM-CA50 [%]	Absolute error with STAM-CA50 [bar]
1	11.2	0.14	31.4	0.43
4	8.7	0.03	15.3	0.05
7	3.8	0.08	45.5	1.02
8	5.5	0.24	68.8	3.11
9	7.9	0.16	20.7	0.43
10	11.6	0.26	20.5	0.46
11	11.5	0.29	23.9	0.60
12	4.4	0.11	72.6	1.96

To close the gap between MAPO98-Pmax curves, caused by different values of lambda some calibration parameters are added to the relationship defined by Equation 1.26. In this way, such curves are shifted and collapse in a unique one, and the effects of the mentioned variables are directly captured by the model.

Background noise contribution on MAPO98 is visible for non-knocking conditions and it is dependent on engine speed. Non-zero values of MAPO for non-knocking conditions are caused by combustion roughness (i.e. combustion speed). Consequently, noise increases with engine speed (the higher the engine speed, the higher the turbulence and so the time-based combustion speed). This effect can be observed for example in Figures 1.39 and 1.40: the MAPO98 value corresponding to the sharp derivative variation of the curve, is increasing from test 6 to 9 to 11 (corresponding respectively to 3000, 4000 and 5000 RPM). For the same reason, in non-knocking conditions, MAPO98 slightly increases with spark advance. Engine speed contribution can be described by the value of the parameter  $c$  of Equation 1.25. Such parameter is supposed to have a linear dependence with engine speed, which can be directly transferred to the parameter  $e$  in Equation 1.26:

$$e = RPM \ ESm \quad (1.28)$$

Where  $ESm$  is the Engine Speed Multiplier and its values (obtained by fitting independently every spark-sweep for all the tested operating conditions) resulted all very close to  $1e-4$ . Therefore, if MAPO98 and MAPO50 are reduced of  $RPM/1e4$ , all curves start approximately from the same value. At the same time, the  $ESm$  value can be optimized to maximize the linear dependence between scaled MAPO98 and MAPO50, and its identification can be carried out through the minimization of the sum of squared errors calculated with the following equation:

$$se = \sum (g(MAPO98 - RPM ESm) - (MAPO50 - RPM ESm))^2 \quad (1.29)$$

In Equation 1.29,  $se$  is the sum of squared errors and  $g$  is the inverse of the slope of the red dashed line shown in Figure 1.42. The optimum  $ESm$  value resulted in  $8.3e-5$ , and it is used for the following considerations. Such value confirms also that  $1e-4$  is a good compromise between describing the engine speed effect on the MAPO distribution and maximizing the correlation between the two percentiles. In this way, Equation 1.26 can be rewritten as follows:

$$m98 = (a STAM^b) Pmax^{(c STAM+d)} \quad (1.30)$$

$$m98 = mapo98 - RPM ESm \quad (1.31)$$

Where  $a, b, c,$  and  $d$  are the four calibration parameters and  $m98$  is the calculated MAPO98, scaled of the engine-speed dependent noise contribution. Equation 1.30 has been used to fit experimental data and to map the scaled MAPO98 trend. The optimum calibration parameters set has been defined by minimizing the root-mean-square of the errors (RMSEs) between experimental percentiles and calculated ones, and it is reported below:

- $ESm = 8.3e-5$  [bar/rpm]
- $a = 1.07e-27$  [kg<sup>-1</sup>]
- $b = -11.53$  []
- $c = 4.40$  [kg<sup>-1</sup>]
- $d = 10.41$  []

It is then convenient to define these quantities to better understand the following figures:

$$M50 = MAPO50 - RPM ESm \quad (1.32)$$

$$M98 = MAPO98 - RPM ESm \quad (1.33)$$

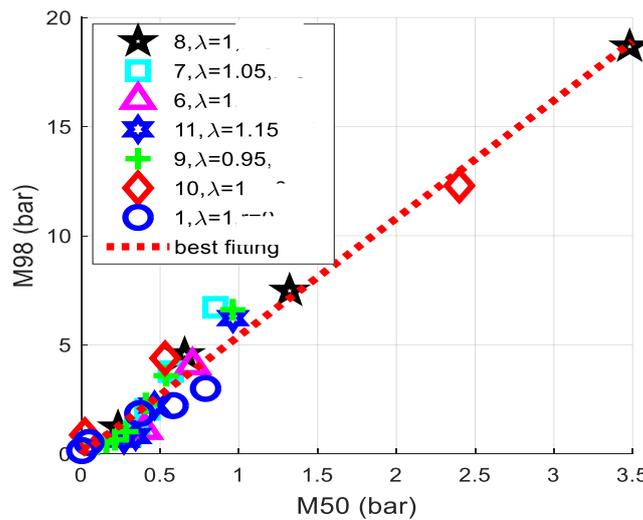


Figure 1.42: Correlation between scaled MAPO98 ( $M98$ ) and scaled MAPO50 ( $M50$ ) with  $ESm$  equal to  $8.3e-5$ .

Figure 1.43 reports  $M98$  values for three different operating points. Instead, Figure 1.44 represents the  $m98$  map, expressed as a function of STAM and Pmax according to the Equation 1.30, and the experimental  $M98$  values, for different SA sweeps. This figure is also useful to highlight the reduced

number of engine points requested to calibrate the knock model (this element can be deduced also from the low number of calibration parameters of Equation 1.30, which defines a smooth surface in the STAM-Pmax plane). Moreover, in Figure 1.44 a part of the map has been neglected because, for each STAM value, it exists a maximum theoretical Pmax value. Figure 1.45 illustrates the absolute error of the data shown in Figure 1.44 (calculated and measured MAPO98, for different operating conditions and different SA). The maximum error is generally related to the brusque change in the curves slope.

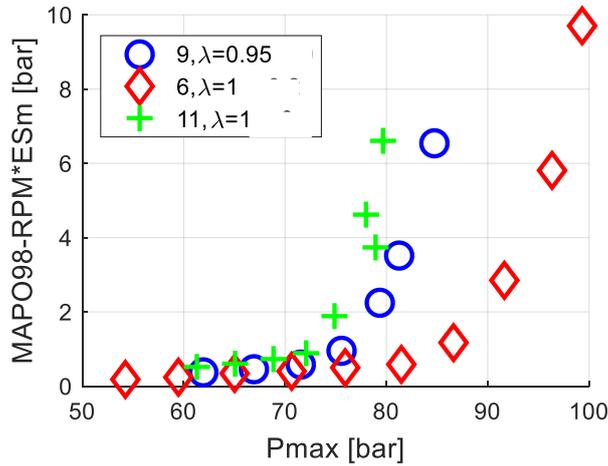


Figure 1.43: Experimental values of MAPO98 curves reduced of RPM\*ESm.

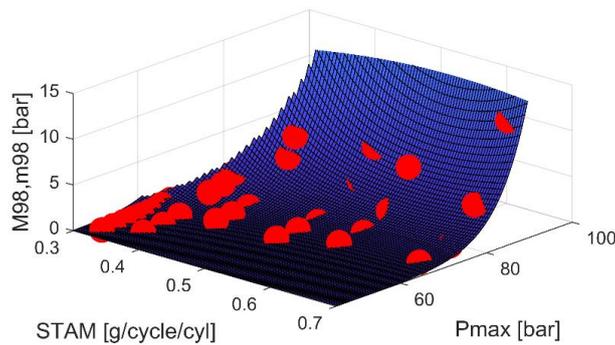


Figure 1.44: m98 map compared with experimental data (M98, red circles).

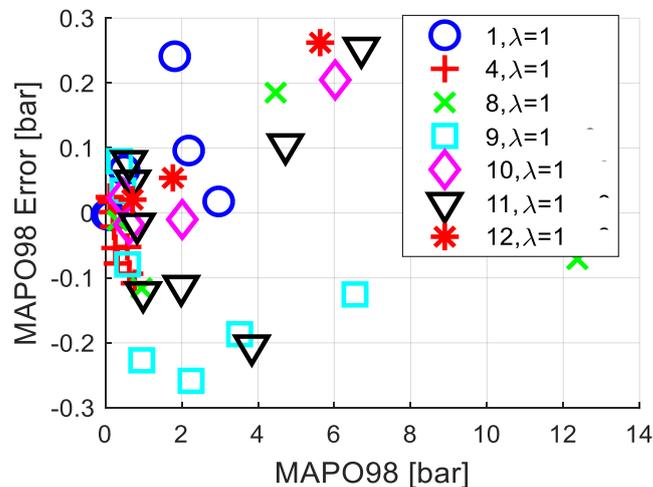


Figure 1.45: MAPO98 absolute errors for engine points collected in the m98 map.

As mentioned above, the calculation of the scaled 50<sup>th</sup> MAPO percentile can be carried out starting from Equation 1.30, through a linear correlation. The  $m50$  calculation can be carried out by introducing only one new parameter in Equation 1.30. The expression of  $m50$  can therefore be defined by Equation 1.34:

$$m50 = g m98 \quad (1.34)$$

In this case the optimal value for parameter  $g$  is 0.17, which is the inverse of the slope of the fitting line found in Figure 1.42.

The absolute and the relative mean error between experimental and calculated 50<sup>th</sup> percentiles have then been evaluated, and the results are collected in Table 1.6. Even if errors are slightly greater than those of the evaluation of MAPO98 (because of the non-perfect correlation between the two percentiles, see Figure 1.42), they remain widely acceptable. Figure 1.46 represents the  $m50$  map obtained by using Equation 1.34 and Figure 1.47 collects the absolute errors for SA sweeps represented in the  $m50$  map. As for Figure 1.44, a part of the map has been neglected to exclude regions that are out of the operating range. The low number of parameters is the main cause of such errors. Moreover, MAPO50 and MAPO98 curves are approximately linear in the lower range, and therefore the model exponential shape tends to overestimate the values corresponding to incipient knock conditions (this is clearly visible in the following Figures 1.48 and 1.49). However, these errors are absolutely acceptable considering their limited magnitude and that they mainly affect the smaller MAPO98 and MAPO50 values.

Table 1.6: Absolute and relative mean error between experimental and calculated MAPO50. The minimum mean error is highlighted in blue, the maximum in orange.

Engine point ( $\lambda=1, r=0$ )	Relative error [%]	Absolute error [bar]
1	11.7	0.03
4	14.8	0.01
7	9.0	0.08
8	15.4	0.12
9	7.3	0.02
10	10.3	0.03
11	7.5	0.03
12	8.1	0.02

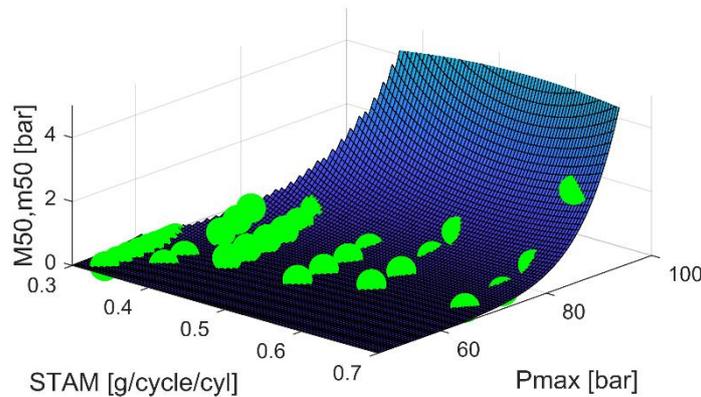


Figure 1.46:  $m50$  map compared with experimental points ( $M50$ , green circles).

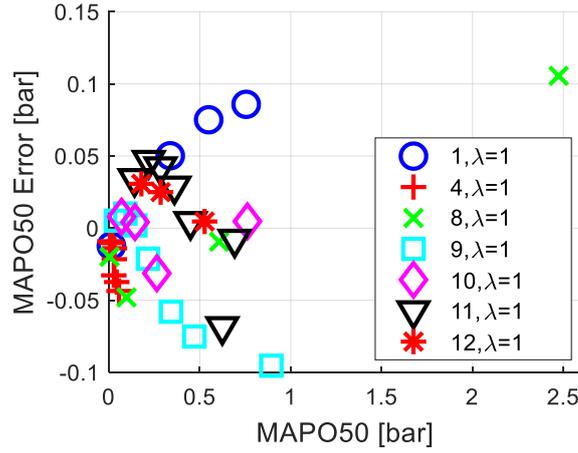


Figure 1.47: MAPO50 errors for engine points collected in the m50 map.

The mixture enrichment (generally used to lower exhaust gas temperature) can be used to suppress knock (on the contrary, mixture leaning is generally a knock enhancer). The effect of the mixture composition can be seen as a mere translation of M98 and M50 curves along the Pmax axis. Such phenomenon is visible in Figure 1.48.

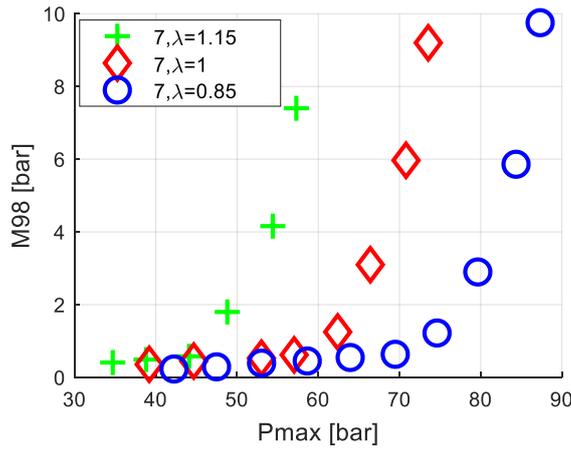


Figure 1.48: M98 curves for fixed operating conditions and for different lambda values.

To adapt the  $m98$  model, a corrective term depending on lambda has been added to Pmax, as shown in Equation 1.35. For three engine points, the SA sweeps are carried out for different values of lambda and these data have been used to calibrate the multiplier that models these effects on MAPO percentiles. Equation 1.35 represents the correlation that was identified to model such dependencies ( $m50$  is calculated with the Equation 1.34):

$$m98 = (a STAM^b) (Pmax + ((\lambda - 1)lm))^{(c STAM+d)} \quad (1.35)$$

Where:

- $\lambda$  is lambda
- $lm$  is the lambda multiplier and it is the calibration parameter. Its optimized value is 18.43 [].

In Figure 1.49, three experimental sets are compared with the curves calculated with Equation 1.35, for three different engine points and lambda values. Table 1.7 shows the mean percentage errors

evaluated between experimental and calculated MAPO98 and MAPO50 for each lambda value. Even in this case, encouraging results have been obtained.

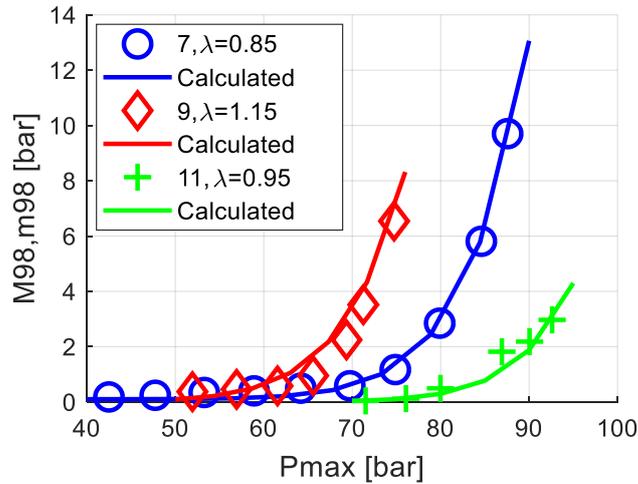


Figure 1.49: Experimental ( $M_{98}$ ) and calculated ( $m_{98}$ ) MAPO98 for three different engine points and lambda values.

Table 1.7: Mean percentage errors between experimental and calculated MAPO98-50, for different lambda values.

Engine point	lambda	MAPO98 error [%]	MAPO50 error [%]
7	0.85	10.7	10.0
	0.95	11.5	16.7
	1.05	12.0	9.9
	1.15	6.3	18.1
9	0.85	11.1	11.2
	0.95	10.8	9.3
	1.05	4.9	10.0
	1.15	4.4	14.4
11	0.85	8.9	12.6
	0.95	14.5	11.2
	1.05	7.0	16.98
	1.15	9.8	14.0

### 1.3.2.2. Knock Model Calibration for the 8-cylinder engine

The described analytical knock model is calibrated also for a 8-cylinder high performance engine. The engine characteristics are listed in Table 1.8.

Table 1.8: 8-cylinder engine characteristics

Displaced volume	3855
Stroke	82 mm
Bore	86.5 mm
Connecting Rod	143 mm
Compression ratio	9.4:1
Number of Valves	4
Exhaust Valve Open	540° BTDC @ 0.1 mm lift
Exhaust Valve Close	360° BTDC @ 0.1 mm lift
Inlet Valve Open	360° BTDC @ 0.1 mm lift
Inlet Valve Close	180° BTDC @ 0.1 mm lift

Six spark sweeps have been carried out, using two types of fuel characterized by different knock resistance levels (RON). In this case, knock model formulation is already defined by the activity carried out on 4-cylinder engine. For this reason, the number of calibration points is significantly lower than those collated in Table 1.4, because model does not need to be identified but just calibrated. Table 1.9 shows the engine speed, intake manifold pressure, lambda and fuel RON for which a spark sweep is performed.

Table 1.9: Tested engine points. For fixed RPM, intake manifold pressure, lambda and fuel a spark sweep has been performed. The nomenclature  $x1-x2$  means that the tested values are  $x1$  and  $x2$ , for both lambda and fuel RON.

Engine point	Speed [RPM]	Load [mbar]	Lambda	Fuel RON
1	2000	1200	0.8-1	95-98
2	2000	1500	0.8-0.98	98
3	2500	1100	1	95-98
4	3000	1300	0.8-1	95-98
5	4000	1500	0.8-0.97	95-98
6	4500	2100	0.8-0.97	95-98

The Equation 1.34 and 1.35 coefficients and the Engine Speed Multiplier have been calibrated to calculate the  $mapo98$  and  $mapo50$  for this kind of engine and the optimized values are the following:

- $g=4.07$  []
- $ESm = 9.4e-5$  [bar/rpm]
- $a = 1.07e-27$  [ $kg^{-1}$ ]
- $b = -8.4$  []
- $c = 1.8$  [ $kg^{-1}$ ]
- $d = 10.8$  []
- $lm = 27.25$  []

The optimal values are very close to those calculated for the 4-cylinder engine. Despite the significant gap in terms of knock tendency and specific power between these engines the values of knock model coefficients are just slightly different, and this is a further confirmation of the approach robustness and its general validity. Figure 1.50, 1.51 and 1.52 display the linear relationship found between  $mapo98$  and  $mapo50$ , the  $m98$  and  $m50$  surfaces compared with the experimental values and the error calculated as the difference between modelled and experimental MAPO percentiles, respectively.

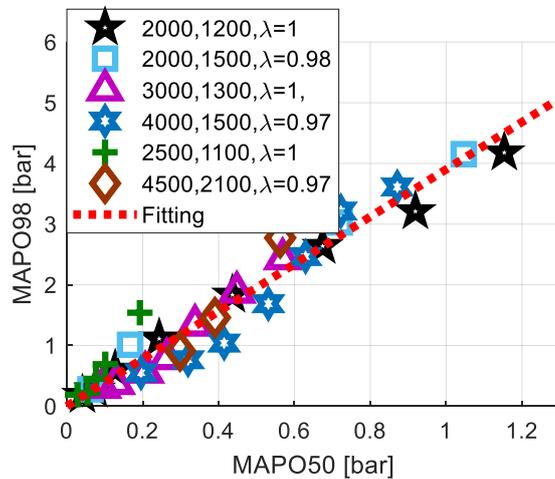


Figure 1.50: Correlation between scaled MAPO98 ( $M98$ ) and scaled MAPO50 ( $M50$ ) with  $ESm$  equal to  $9.4e-5$ .

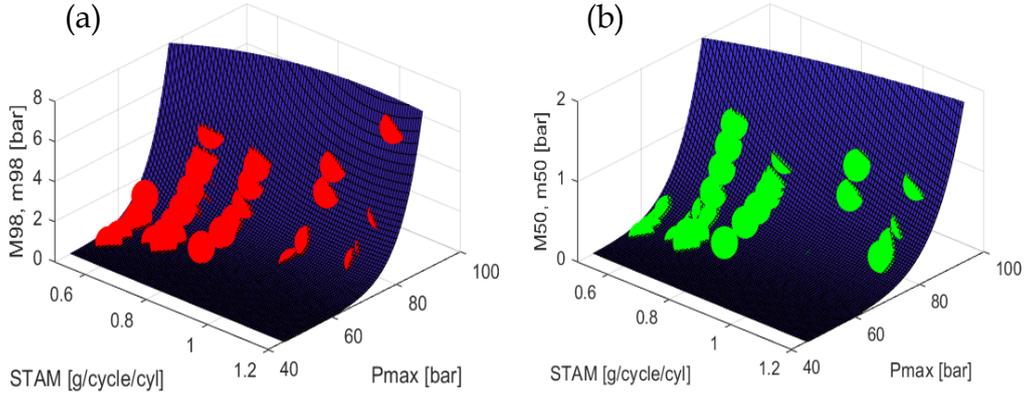


Figure 1.51: (a)  $m_{98}$  and (b)  $m_{50}$  function compared with experimental points (red and green circles, respectively).

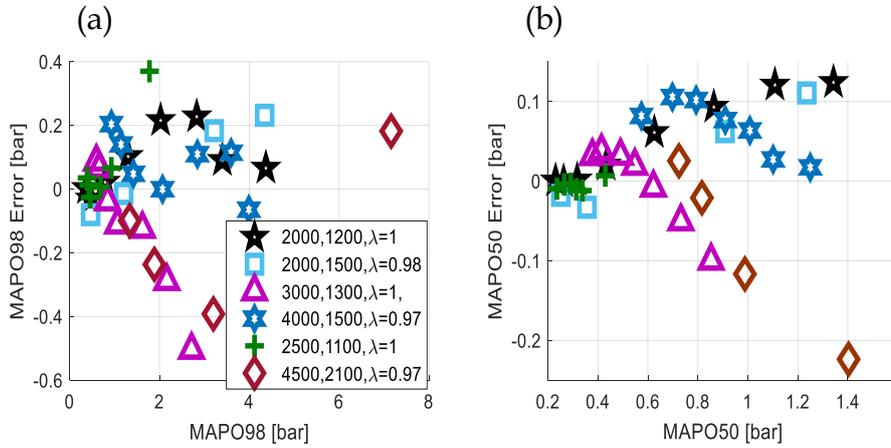


Figure 1.52: Error between calculated and modelled MAPO98 (a) and MAPO95 (b).

The same method used to introduce the influence of a lambda variation in the knock model equation, is applied to compensate the effect caused by fuel with a different RON value and another calibration parameter is added to Equation 1.35, which evolves in Equation 1.36 ( $m_{50}$  is still calculated with the Equation 1.34):

$$m_{98} = (a STAM^b)(Pmax + (\lambda - 1)lm + RONm(RON_{ref} - RON_{fuel}))^{(c STAM + d)} \quad (1.36)$$

Where:

- $RON_{ref}$  is the reference fuel RON value, fixed to 98
- $RON_{fuel}$  is the RON value of the burned fuel
- $RONm$  is the RON multiplier. Its optimized value resulted in 1.63 [].

Figure 1.53 shows the comparison between experimental and calculated  $M_{98}$  for RON95 fuel. The accuracy of the fitting process is demonstrated by the resulting R2 correlation index and the Root Mean Square Error that are equal to 0.96 and 0.5, respectively.

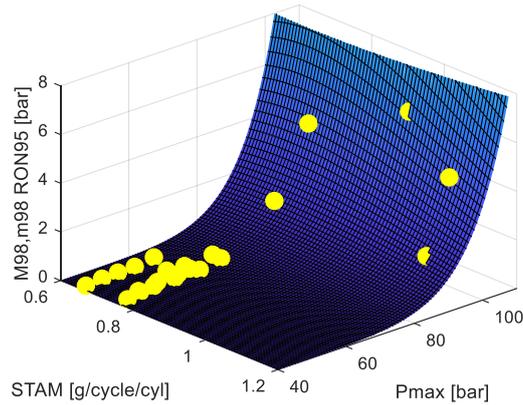


Figure 1.53:  $m_{98}$  function compared with experimental value (yellow points) for RON95 fuel.

The same approach could be applied to include the effect of other influent parameters on the engine knock tendency, such as the intake air temperature, the variable valve timing (VVT), cold high/low pressure EGR or water injection (as demonstrated in [1.79]). It is important to mention that this modelling approach is suitable to calculate or control knock intensity for a calibrated engine. This means that the effects of the calibrated ECU strategies that manages VVT, EGR or other technologies that affect the combustion process are already included in the parameters  $a$ ,  $b$ ,  $c$  and  $d$  of  $map_{098}$  and  $map_{050}$  equations. For this reason, the model includes only variable that are commonly changed by the ECU when the engine is working like in the final on-vehicle application (the SA and lambda), and the fuel quality. The air temperature in the intake manifold might be also included in the model, but as well known, such variable has an opposite effect on the fresh charge. The higher the intake air temperature, the higher the charge temperature, but even the lower the air density. Thus, this double effect does not result in a clear increase or decrease in the knock tendency of the engine. Figure 1.54 shows  $MAPO_{98}$  on mean  $P_{MAX}$  recorded performing two spark sweeps under the same operating conditions (3000 RPM, 1300 mbar of intake manifold pressure, lambda equal to 1 and RON 95 gasoline), with different air temperature in the intake manifold. Reported trends demonstrate how, for the considered engine, the knocking behavior is practically the same. This observation allows to confirm that the effect of the intake air temperature can be excluded from the knock model calibration for this 8-cylinder engine. Nevertheless, the influence of the air temperature for another engine cannot be deduced a priori, but, however, it can be easily introduced in the knock model equation with the same approach used for fuel RON and lambda.

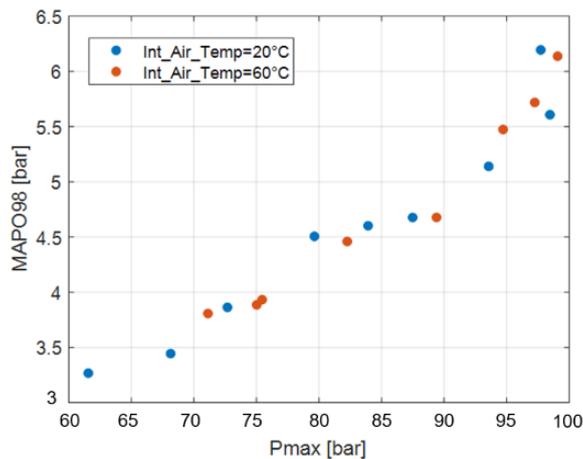


Figure 1.54: Knock tendency for 8-cylinder engine with different intake air temperature for fixed operating conditions (3000 RPM, 1300 mbar, lambda 1 and RON 95 gasoline).

### 1.3.2.3. Knock Model Validation and Its Implementation in GT Power Environment

When two percentiles are known, characteristic parameters of a log-normal probability function can be analytically determined. With this procedure, the entire MAPO distribution can be rebuilt, for a fixed operating condition. The mean value ( $\mu$ ) and the standard deviation ( $\sigma$ ) are easily obtainable from two generic quantiles (such as 50<sup>th</sup> and 98<sup>th</sup> percentiles), using properties of Gaussian distributions. In fact, a log-normal probability density function can be easily obtained from the associated Gaussian distribution:

$$Y(\mu, \sigma) = e^{X(\mu, \sigma)} \quad (1.37)$$

Where:

- $Y$  is the log-normal distribution with mean value  $\mu$  and standard deviation  $\sigma$
- $X$  is the associated normal distribution with mean value  $\mu$  and standard deviation  $\sigma$

All normal distributions can be therefore calculated starting from the standardized normal distribution  $Z$  with mean value 0 and standard deviation 1:

$$X(\mu, \sigma) = \sigma Z(0,1) + \mu \quad (1.38)$$

Replacing the Equation 1.38 in 1.37:

$$Y(\mu, \sigma) = e^{(\sigma Z + \mu)} \quad (1.39)$$

For its properties, the log-normal distribution is typically used to describe MAPO values and it is preferred to other more complex expressions. These properties can be used to calculate the mean value and the standard deviation of a log-normal distribution from two percentiles. If two percentiles of normal distribution  $X$  are given, it is possible to write:

$$P(X < x_1) = p_1 \quad (1.40)$$

$$P(X < x_2) = p_2 \quad (1.41)$$

In which:

- $P$  indicates the probability
- $x_i$  is the value of the percentile
- $p_i$  is the value of the percentage of the percentile

Replacing Equation 1.38 in 1.40 and 1.41:

$$P(\sigma Z + \mu < x_i) = p_i \quad (1.42)$$

Rearranging the above equation, it is possible to write:

$$P\left(Z < \frac{x_i - \mu}{\sigma}\right) = p_i \quad (1.43)$$

$$p_i = \Phi\left(\frac{x_i - \mu}{\sigma}\right) \quad (1.44)$$

In which  $\Phi$  is the CDF of standardized normal distribution  $Z$ . By reversing Equation 1.44, a system of two equations in two variables is defined:

$$\Phi^{-1}(p_i) = \frac{x_i - \mu}{\sigma} \quad (1.45)$$

$$\begin{cases} \Phi^{-1}(p_1)\sigma + \mu = x_1 \\ \Phi^{-1}(p_2)\sigma + \mu = x_2 \end{cases} \quad (1.46)$$

Solving the first equation for mean value, replacing it in the second one, and recalling Equation 1.37:

$$\begin{cases} \mu = \frac{x_1 \Phi^{-1}(p_2) - x_2 \Phi^{-1}(p_1)}{\Phi^{-1}(p_2) - \Phi^{-1}(p_1)} \\ \sigma = \frac{x_2 - x_1}{\Phi^{-1}(p_2) - \Phi^{-1}(p_1)} \end{cases} \quad (1.47)$$

For a normal PDF

$$\begin{cases} \mu = \frac{\ln(x_1) \Phi^{-1}(p_2) - \ln(x_2) \Phi^{-1}(p_1)}{\Phi^{-1}(p_2) - \Phi^{-1}(p_1)} \\ \sigma = \frac{\ln(x_2) - \ln(x_1)}{\Phi^{-1}(p_2) - \Phi^{-1}(p_1)} \end{cases} \quad (1.48)$$

For a log-normal PDF

In particular, if  $x_1$  is the 50<sup>th</sup> percentile and  $x_2$  is the 98<sup>th</sup>,  $\Phi^{-1}(p_1) = 0$  and  $\Phi^{-1}(p_2) = 2.0057$ , and the final relationship is defined by:

$$\begin{cases} \mu = \ln(x_1) \\ \sigma = \frac{\ln(x_2) - \ln(x_1)}{2.0057} \end{cases} \quad (1.49)$$

The final equations applied to the knock model are then the following:

$$\mu = \ln(m50 + RPM ESm) \quad (1.50)$$

$$\sigma = \frac{\ln(m98 + RPM ESm) - \ln(m50 + RPM ESm)}{2.0057} \quad (1.51)$$

Where:

- $\mu$  is the mean value of MAPO log-normal distribution
- $\sigma$  is the standard deviation of MAPO log-normal distribution

Once  $\mu$  and  $\sigma$  are known, the MAPO log-normal distribution is defined. At this point, the distribution can be used, for example, to randomly generate knock intensities (if the stochastic behavior simulation is needed), associated with the current engine operation, in a 0/1-D model, as

described below. Moreover, it is also possible to calculate any desired percentile, other than the 50<sup>th</sup> or the 98<sup>th</sup>. Finally, the whole distribution (or its cumulative curve) can be compared to the experimental one to validate the calculation.

Some extra engine points have then been tested with both engines, to collect experimental data to verify the model predictivity. Such tests have been performed under steady state conditions (fixed engine speed, load, and SA) and Figure 1.55 (a) and (b) display results for 4-cylinder engine but valid even for 8-cylinder one.

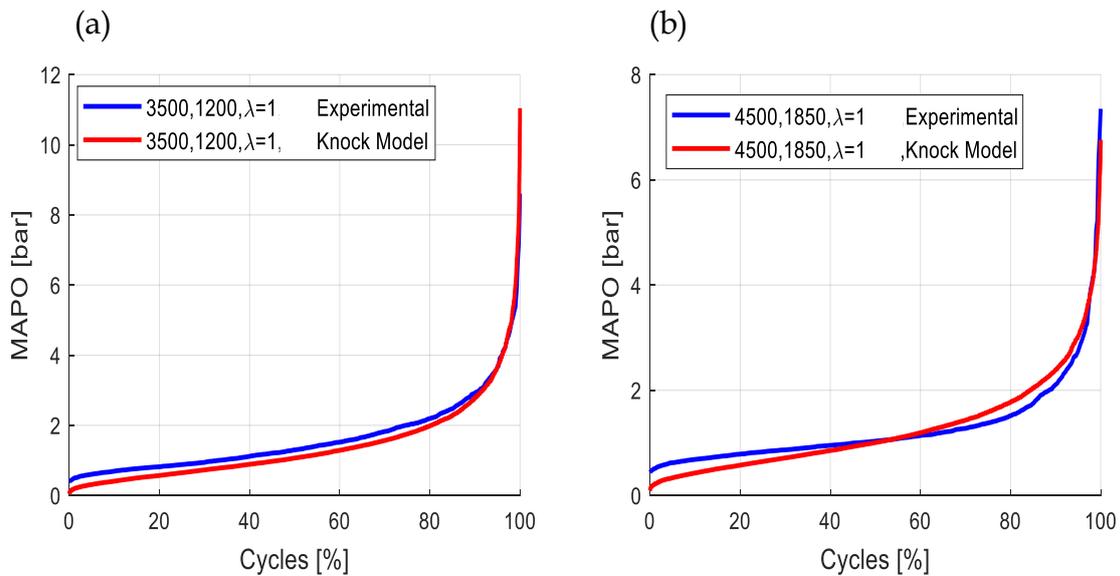


Figure 1.55: Superposed calculated and experimental cumulative MAPO distributions for two different engine points.

The described knock model can be applied in several ways. For example, it could be implemented in a 0-D simulation environment for the cyclic generation of a MAPO-like index, or for real-time, control-oriented applications, or it could be coupled with a predictive combustion model. In the following example, it has been coupled with a GT Power predictive combustion model [1.80] of the 4-cylinder engine presented above, that was already calibrated via Three Pressure Analysis (TPA). The aim of this application is the calculation of two MAPO percentiles and the distribution parameters, using the values of STAM and Pmax provided by the 1-D engine model. Thus, Equations 1.34, 1.35, 1.50 and 1.51 have been implemented in a user-defined template.

The GT Power engine model has one-cylinder layout, and it simulates the mean cylinder behavior. Such model includes the components between intercooler outlet and turbine inlet. Combustion simulations are governed by a predictive model (CombSITurb GT Power template, [1.80]) that had been calibrated via TPA. The selected approach involves two macro-steps: during the first one, the combustion model is not needed, and the analysis of intake, in-cylinder and exhaust pressure curves recorded under steady-state conditions is carried out. The combustion model is introduced during the second step and its parameters are calibrated in that phase. During the first stage, the solver determines the in-cylinder conditions at the start of the cycle (SOC, which coincides with Intake Valve Closing event, IVC) using experimental pressure and other thermal and geometric data of the combustion chamber. For the first cycle, a dummy burn rate is used, and no pressure analysis is performed. For the second cycle and beyond, the forward run simulation will pause at the start of each cycle and calculate the apparent burn rate using the trapped conditions in the cylinder at that point (IVC) along with the measured pressure profile. The injection profile (if an injector is present) and the heat transfer rate are taken from the previous cycle results. The forward simulation

continues, and the apparent burn rate calculated in the previous step is imposed during the cycle. Cycles repeat until steady state convergence is reached. The model used for this first stage excludes the combustion model and it includes only components that are between the intake and the exhaust pressure measuring points (Figure 1.56). The second step needs the introduction of the combustion model, and the solver optimizes combustion parameters to find the set which better reproduces the burn rates found in the previous phase. During these simulations, the model includes only the injector, the cylinder (with the combustion model template), and the crankshaft templates (Figure 1.57). Such procedure is repeated for each engine point available for the calibration process.

Results of described calibration process are not showed in this chapter, because the focus is the validation of the analytical knock model in 1-D simulation environment. Anyway, the accuracy of the GT Power predictive combustion model is demonstrated in chapter 2, where experimental and modelled pressure curves are compared, also including the Port Water Injection (PWI) system.

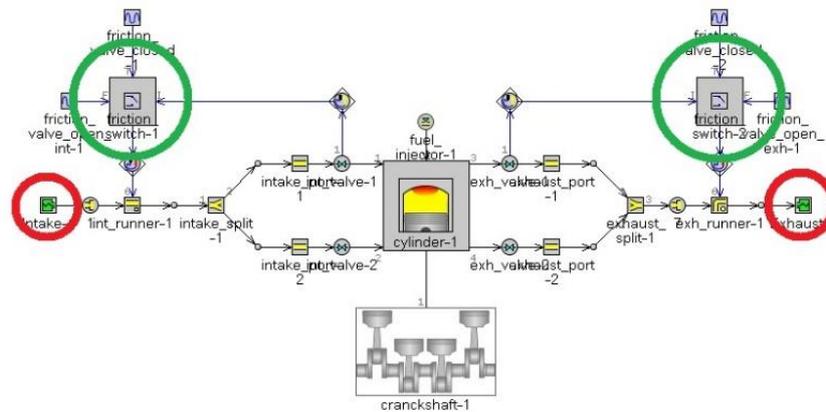


Figure 1.56: 1-D engine model used for the TPA. Red circles highlight the End-Environment template in which intake and exhaust pressure vector have been uploaded. Green circles highlight Switch template that increases the runner friction to damp undesirable pressure oscillations when the intake and exhaust valves are closed.

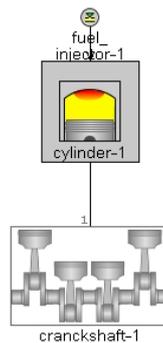


Figure 1.57: GT Power model used for combustion parameters optimization.

Knock model has been easily implemented in GT Power simulation environment, due to the simple analytical formulation. Equations and parameters involved by the model have been introduced in a user defined template (Figure 1.58) that has been then added to the main engine model (Figure 1.59). Its inputs are STAM (calculated at IVC), maximum in-cylinder pressure, engine speed and lambda.

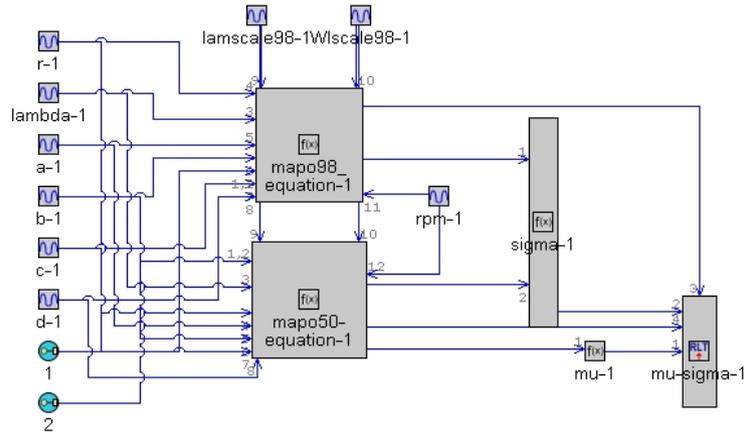


Figure 1.58: Layout of User Knock Index (UKI) template. In *mapo98* and *mapo50* blocks, Equations 1.34 and 1.35 have been implemented, while Equations 1.50 and 1.51 have been introduced in  $\mu$  and  $\sigma$  blocks respectively.

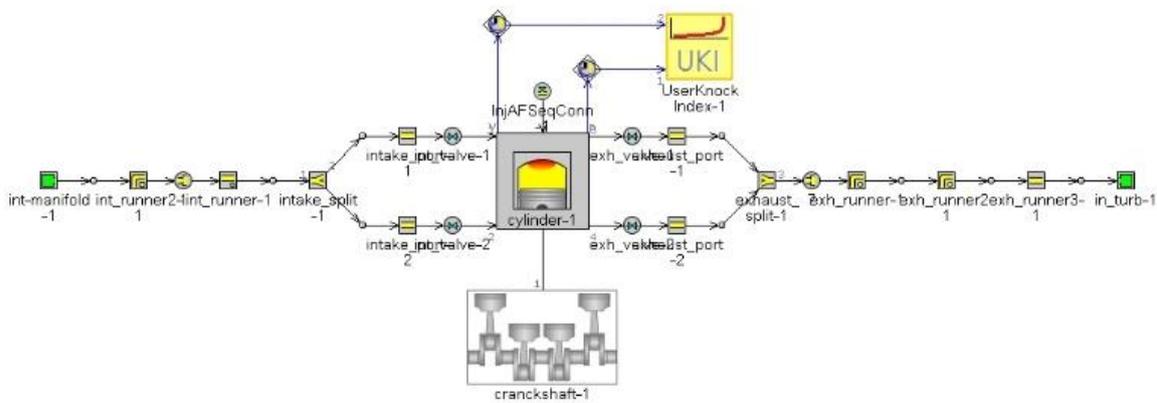


Figure 1.59: Engine model with UKI template, in which Equations 1.34, 1.35, 1.50 and 1.51 have been implemented. Sensors provide STAM and  $P_{max}$ , while  $\lambda$  and RPM are set as inputs in the template properties window.

The 1-D model capabilities have been verified reproducing all the experimental engine points used for calibrating and validating knock model. Such engine points are different than those applied for the TPA calibration process. Therefore, the comparison between experimental and GT MAPO98 and MAPO50 is also a validation of the combustion model. For this reason, the 1-D model has been validated also on engine points used for the calibration of knock model. Table 1.10 shows the percentage errors between experimental and simulated STAM,  $P_{max}$ , MAPO98 and MAPO50 for all tested operating conditions. For engine points with  $\lambda$  variations, the table shows the mean error for all tested  $\lambda$  values. Table 1.10 shows that small errors on STAM and  $P_{max}$  can produce very high errors on MAPO98 and MAPO50, but this is not always true. Moreover, the model has been run on validation points and in the following figures (1.60, 1.61, 1.62, 1.63, 1.64) the experimental MAPO cumulative distributions have been compared with those calculated by the knock model and by UKI template. In such figures, the red lines show the results obtained with the knock model, in which experimental STAM and  $P_{max}$  are used as inputs, while the green lines represent distributions calculated by the knock model implemented in UKI template, in which STAM and  $P_{max}$  calculated by GT engine model are used as inputs. It can be observed that the knock model implementation within the 1-D environment does not introduce significant additional errors, except at intermediate engine speeds, where its sensitivity to STAM and  $P_{max}$  errors appears to be higher.

Table 1.10: Comparison between experimental, calculated with knock model (KM, with experimental data as inputs) and calculated with the UKI template (UKI, with GT outputs as inputs) performance parameters.

Engine point [RPM – mbar]	Error TAM Exp-GT [%]	Error Pmax Exp-GT [%]	Error MAPO98 Exp-KM [%]	Error MAPO98 Exp-UKI[%]	Error MAPO50 Exp-KM [%]	Error MAPO50 Exp-UKI [%]
2000-1130	11.6	3.54	13.0	16.5	20.9	26.2
2500-1500	3.1	12.9	9.6	13.9	23.0	16.8
2500-1600	5.9	4.0	10.6	13.7	21.9	23.6
3000-980	8.7	12.9	11.6	7.8	8.7	8.8
3000-1300	10.7	15.7	13.0	17.9	19.0	29.7
3000-1500	9.5	16.9	10.3	12.2	19.1	24.5
3000-1700	11.2	13.3	14.9	34.2	21.2	27.9
3000-2000	1.7	17.9	7.5	17.3	13.9	30.6
4000-1430	5.8	4.8	20.3	16.6	17.0	14.9
4500-1630	7.9	10.0	4.4	5.1	19.1	21.5
5000-1120	9.3	7.6	13.5	11.8	21.1	17.4
5000-1490	6.7	12.3	8.2	6.7	10.5	13.7
2250-1210	3.6	3.7	8.3	1.9	14.0	15.8
3500-1200	10.1	5.9	2.0	45.3	17.8	48.2
3500-1810	2.2	14.8	4.2	30.0	18.3	38.7
4500-1110	7.1	11.9	1.6	15.3	9.3	5.2
45001850	8.8	9.4	0.9	25.6	1.7	18.3

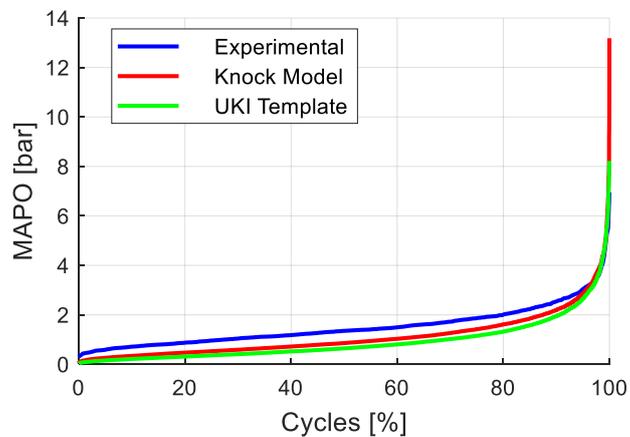


Figure 1.60: Comparison between experimental, calculated with the Knock Model (the inputs are the experimental data) and calculated with the UKI Template (the inputs are the GT model outputs) MAPO cumulative probability distribution, for engine point 2250 RPM, 1210.

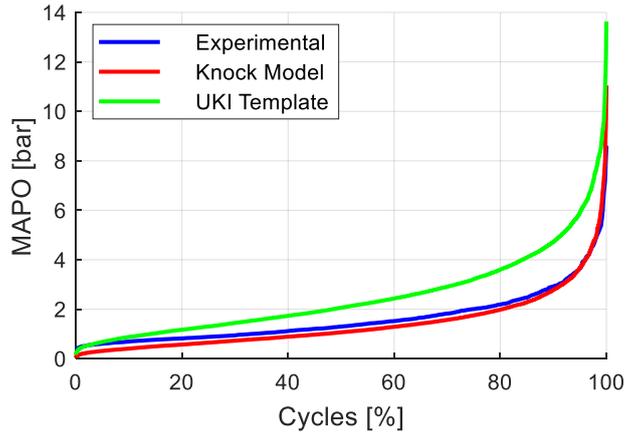


Figure 1.61: Comparison between experimental, calculated with the Knock Model (the inputs are the experimental data) and calculated with the UKI Template (the inputs are the GT model outputs) MAPO cumulative probability distribution, for engine point 3500 RPM, 1200.

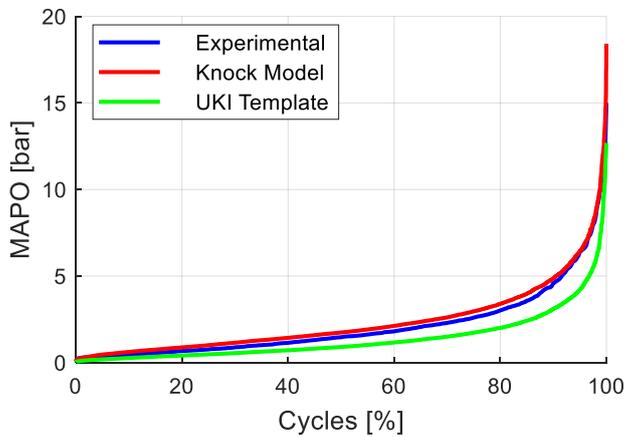


Figure 1.62: Comparison between experimental, calculated with the Knock Model (the inputs are the experimental data) and calculated with the UKI Template (the inputs are the GT model outputs) MAPO cumulative probability distribution, for engine point 3500 RPM, 1810.

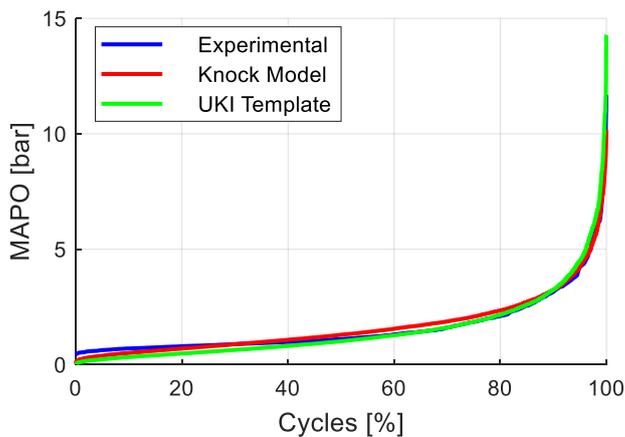


Figure 1.63: Comparison between experimental, calculated with the Knock Model (the inputs are the experimental data) and calculated with the UKI Template (the inputs are the GT model outputs) MAPO cumulative probability distribution, for engine point 4500 RPM, 1110.

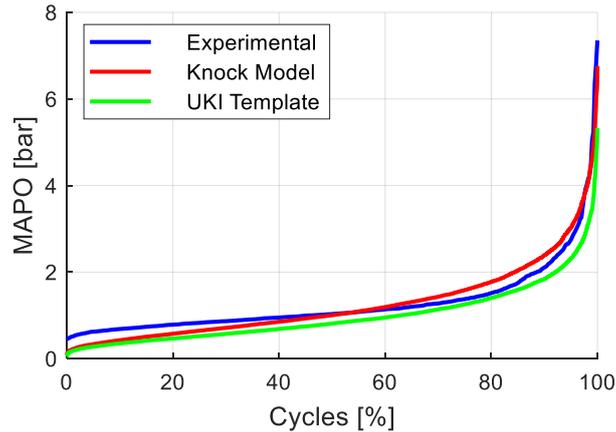


Figure 1.64: Comparison between experimental, calculated with the Knock Model (the inputs are the experimental data) and calculated with the UKI Template (the inputs are the GT model outputs) MAPO cumulative probability distribution, for engine point 4500 RPM, 1850.

These results are a further demonstration of high accuracy and reliability of proposed approach that is based on the analysis of trends of considered index w.r.t. influent engine variables. Observed trend are then described with analytical functions, that can be easily manipulated and are compatible with the real time execution, for instance, in a knock control strategy.

The analytical methodology is extended to model main combustion indexes, such as the MFB50, PMAX and IMEP. An analytical engine simulator has been developed with the aim to have an accurate combustion indexes generator sensitive to the combustion phase variation. Approaches and results presented in the following session have been already published in [1.81].

### 1.3.3. Analytical Modelling of Combustion Indexes

This section is not strictly related with knock modelling. Nevertheless, it describes an extension of analytical approach proposed above that led to the development of some reliable models that have been implemented in an innovative knock control strategy discussed in the next chapter. Thus, some control-oriented models are shown below, and they allow a better understanding of the following topic. As introduced above, in this section a control-oriented, empirical engine simulator is described, calibrated, and validated. It is composed of some analytical models that can calculate the main combustion indexes, such as MFB50, the maximum in-cylinder pressure, IMEP (including Cycle-to-Cycle Variation, CCV) and a pre-defined percentile of a knock index statistical distribution, implementing the analytical knock model discussed in the previous section. Each model is obtained by fitting the experimental data, which can be described by relevant statistical quantities. The so-found equations define the relationship between the independent variables (for example engine speed and load) and the dependent ones (i.e. the modelled quantities). The idea is that to estimate the mean value and the standard deviation (STD) for every considered combustion index apart from knock intensity (MFB50, Pmax, and IMEP); then, by utilizing the statistical definition of the Gaussian Probability Density Function (PDF), the cyclic index is calculated according to the related distribution. In fact, as described in [1.80], the cyclic values of such indexes can be efficiently described with a statistical approach, through a normal PDF. When the mean value and the STD are known, the entire distribution is completely defined, and the cyclic value can be randomly extracted from such PDF. All these empirical models are then implemented in Simulink environment, realizing an analytical simulator capable to calculate these cyclic indexes, on the whole engine

operating field (i.e. speed, load, and Spark Advance). The proposed simulator has been developed with the aim to generate combustion indexes with a high fidelity with respect to the real engine, when the same speed, load, SA, and the same calibration maps (such as target lambda values) are applied. This tool is useful to debug or validate a new combustion controller, which needs accurate combustion data. In fact, due to the analytical approach, it can easily run in RT simulations, reproducing conditions which are very close to those faced during experimental tests.

Different 0-D approaches are those based on polynomial models, Gaussian Process (GP) regression models and Artificial Neural Network (ANN), which allow to mathematically calculate engine indexes (i.e. combustion parameters, engine torque, fuel consumption, exhaust gas composition, etc.) as a function of some influent variables [1.81]. Such kind of model needs to be trained with an experimental dataset to identify the best set of calibration parameters. The GP regression models are particularly sensitive to the number of inputs and the simulation time significantly increases with the number of independent variables. The ANNs implementation issues are mainly related to the engine variables identified to become the network inputs and the possibility to reverse the resulting functions. Polynomial models are generally calibrated with stepwise regression methods, which involve the function degree is progressively increased until the error between the experimental values (or those of the detailed model) and the calculated ones goes under a fixed threshold. The proposed approach is based, instead, on simple analytical functions that depend on few independent variables and this feature makes the models easy to reverse. The described mean value combustion index models are comparable with MVMs in terms of computational effort, but the aim of the final simulator is that to generate cyclic synthetic combustion data, including the CCV. For this reason, an innovative approach to the cyclic variability modeling has been developed, and in this way cyclic combustion indexes (including MAPO) are calculated with high accuracy, through the evaluation of few simple equations. Moreover, the particular feature of the described polynomial method is that an analysis of the physical trends of the studied variables is requested to define the best polynomial order. In this way there is a more aware approach to the modelling, and it allows to minimize the number of model parameters. Moreover, the models capture some physical trends which are usually hidden to standard polynomial ones or ANN.

The proposed simulator is thought to be coupled with a combustion control system and to be used like a generator of indexes with a strong fidelity to the real engine. In other words, ANESIM does not replicate the intake/exhaust flow and the combustion physics, but it works like a virtual engine with a very high accuracy on estimating combustion indexes, which serve as an input during the controller development phase (for example a combustion phase control strategy). The innovative contribution of this work is then the development of a simulator that can be calibrated with a very limited amount of experimental data, allowing to calculate the main combustion indexes with extremely low errors, on the entire engine operating field. In fact, the required data to calibrate ANESIM are the main combustion indexes during SA variations (i.e. spark sweeps) for different engine points. Moreover, the described models and the proposed approach are completely new, and the accuracy is comparable or even higher than that achievable with physics-based 1-D and 0-D engine models, for the considered quantities. ANESIM is defined by analytical equations and this involves that the required computational effort is extremely low. In this way, the simulator can run in Real-Time on a common laptop. ANESIM is then particularly suitable for Software-in-the-Loop (SiL) simulations when the tested controller requires these combustion indexes. Therefore, the analytical formulation of the described models allows the models inversion and implementation in a control strategy, to estimate indicated quantities or actuation targets.

The experimental setup is described, and the campaigns used to calibrate the models and validate the simulator are explained. In this work the mean cylinder was considered but the proposed simulator can be calibrated individually for all cylinders. The calibration process is described for each modelled index, and its performance is evaluated through the R-squared correlation coefficient (R<sup>2</sup>). The choice of the polynomial degree used to fit the experimental data is driven by the obtained accuracy. Moreover, a sensitivity analysis to the number of points used for the model calibration was carried out. Then the implementation of the analytical models in a zero-dimensional simulation environment is carried out. In this way the only three inputs required by ANESIM are the engine speed, load, and the SA strategy, and it can generate the cyclic MFB50, Pmax, IMEP and MAPO. Finally, the validation of the zero-dimensional simulator is presented. The trend of the combustion indexes registered at the test bench are compared with those calculated by the ANESIM, when it is supplied with the same engine speed and load profiles. The Root Mean Squared Error (RMSE) between the indexes mean values is then calculated.

The experimental campaign was carried out on the 8-cylinder GDI TC engine used for the calibration and validation of knock model. Thus, the main characteristics are collated in Table 1.8. Data used for the development and the calibration of the models were recorded performing some spark sweeps on the entire operating field and the mean cylinder was considered. For each engine point, the mapped lambda was applied, and the recordings were performed under steady state conditions, including about 200 cycles for every engine point. The database used for the model calibration is not that used for the knock model calibration and it comes from an engine characterization experimental campaign, useful to calibrate some ECU control strategies. In other words, such kind of combustion data have not been logged for this specific modelling activity but was already available and it does not include lambda or fuel variations. The spark sweeps have been carried out avoiding excessively high MAPO values and, on the other hand, too high exhaust gas temperature, especially for high load operating conditions. On the contrary, data used for ANESIM validation was logged during tests that include both steady state and transient conditions. Each cylinder is equipped with an in-cylinder pressure transducer, and the signal is sampled at 200 kHz. The calculation of MFB50, IMEP and Pmax uses the low pass filtered pressure trace. Instead, the same signal is band-pass filtered to calculate MAPO index. Cut-off frequencies are those defined in section 1.3.1 (page 48).

Figure 1.65 shows the engine points tested during the experimental campaign, where torque and speed are normalized with respect to the maximum value for confidentiality reasons (this choice is adopted also for other indexes and variables). Data are intentionally numerous, to test the simulator robustness and predictivity while varying the amount of data considered for the calibration process.

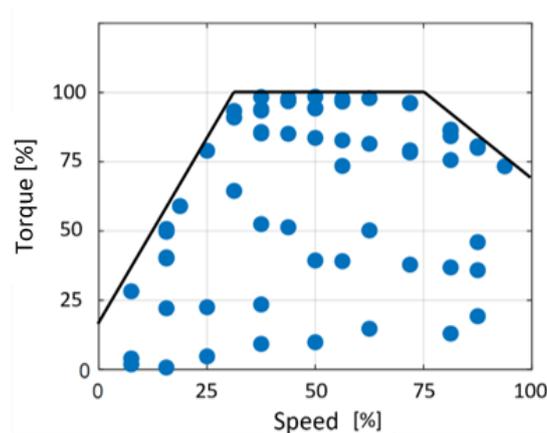


Figure 1.65: A total of 60 tested engine points on the Speed-Torque plane. For each point a spark sweep was performed. The parameters are normalized with respect to their maximum value.

The first model described in this work implements the relationship between the Spark Advance (or  $\Delta SA$ ) and the MFB50. The idea is to find an analytical equation to calculate the mean MFB50 for fixed working conditions (speed, load, and SA). The calculation of the mean MFB50 is the first step to be executed also within the resulting simulator, because the combustion phase information is required by the Pmax and IMEP models. In fact, such models define the relationship between MFB50, engine load, and the maximum in-cylinder pressure and the indicated mean effective pressure, respectively. The Pmax is then used by the knock model to estimate MAPO percentile. In other words, such models are designed to be cascaded one on top of the other. As introduced above, the calculated MFB50, Pmax and IMEP are the mean values characteristic of the combustion cycles generated by the engine in fixed conditions, and with fixed calibrations (which are the same used to supply the models). More specifically, these models do not include the effects of the actuations which are mapped and therefore constant for fixed operating conditions. For example, the air-to-fuel ratio, the injection timing, and the camshaft phasing are not considered as inputs, and their effects are implicitly considered. This is because such calibrations are supposed to be constant and perfectly actuated for the given engine operating point. The same method was followed for the calibration of knock model coefficient, as described above. It is important to highlight the empirical nature of the proposed models, which aim to be a copy of a calibrated engine in terms of synthetic combustion data. On the contrary, the knock model also considers the effects of lambda on the calculated MAPO values. For this reason, the lambda map is included in ANESIM. Figure 1.66 shows the layout of the resulting simulator which is implemented in a Simulink environment and highlights inputs and outputs of each model. Such scheme includes all the model discussed in this work and the final combustion indexes calculation chain. The MFB50 and the STD MFB50 blocks calculate the mean value and the standard deviation of the CA50 respectively, and these values are sent to the PDF block which calculates the corresponding cyclic ('cc') index. Such index is then used by IMEP and PMAX models to generate their respective cyclic values. The mean value of MFB50 is also sent to another PMAX model because the knock model needs this value. The detailed description of the simulator layout is proposed in the section ANESIM.

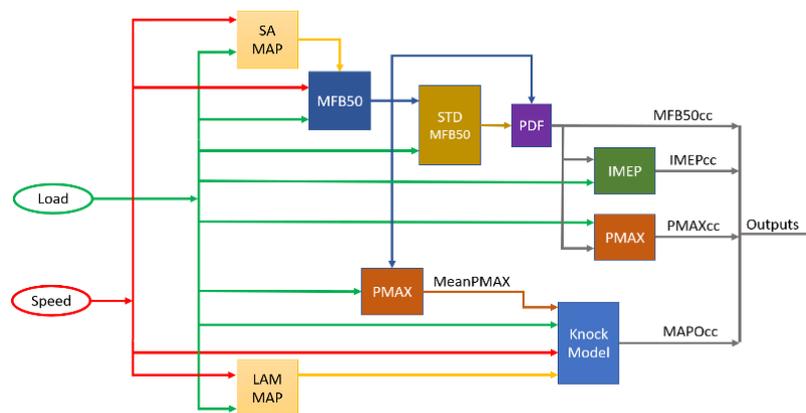


Figure 1.66: ANESIM layout. Load and speed (RPM) are the two inputs of the simulator. The orange boxes are the SA and lambda maps, the yellow one represents the implementation of the MFB50 STD model, and the violet box identifies the calculation of the Normal PDF and the extrapolation of a related random value.

For all the presented models the analytical formulation is described and a sensitivity analysis to the polynomial fitting order is shown. The model reliability is demonstrated also with a sensitivity analysis to the number of engine points (i.e. spark sweeps) used for the calibration. Such process is carried out with an increasing number of data and by calculating the errors with all those available. The errors of the proposed models, which were eventually calibrated with the lowest number of spark sweeps, are therefore collected, and shown for all the points (speed, load, and SA).

### 1.3.3.1. SA-to-MFB50 model

For each engine point (fixed speed and load), the relationship between Spark Advance and MFB50 is characterized by a parabolic trend, which can be described with a second order polynomial. Such trend derives from a variable MFB50 sensitivity to the SA: the higher the MFB50, the higher the curve slope. The curvature increases also with respect to the engine load. Figure 1.67 shows an example of what has been asserted, collecting the spark sweeps recorded for different loads at a fixed engine speed (equal to the 40% of the maximum speed). The red lines represent the parabolic fitting functions. In all figures of this section the SA and the MFB50 are normalized and respectively indicated as 'STEP' and 'SPACE'.

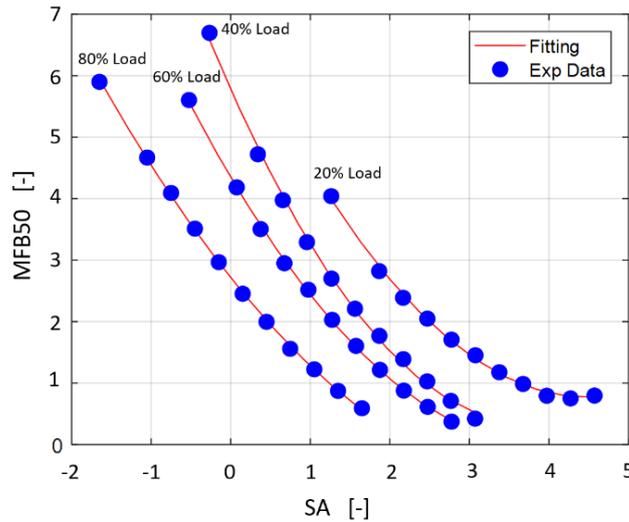


Figure 1.67: Experimental MFB50-SA points recorded for an engine speed equal to 40%. Each parabolic function corresponds to a different load, which increases from 20% (on the right) up to 80% (on the left), with steps of 20%. The blue points represent the experimental data, the red lines display the parabolic fitting. The SA and the MFB50 are expressed as normalized 'STEP' and 'SPACE' respectively.

For each engine point (defined by the couple speed and load) a second order polynomial can be calculated. Then the parameters  $A$ ,  $B$  and  $C$  of Equation 1.52 can be defined in the speed – torque domain. Equation 1.52 describes the kind of equation used to fit MFB50 with respect to SA, for each engine point:

$$MFB50 = A SA^2 + B SA + C \quad (1.52)$$

Where  $A$ ,  $B$  and  $C$  are the equation parameters.

The calibration process of such equation consists of fitting the  $A$ ,  $B$  and  $C$  parameters as a function of speed and load, through a polynomial with two independent variables. Different order polynomials were tested and compared in terms of correlation and RMSE between the experimental (i.e.  $A$ ,  $B$  and  $C$  calculated from the fitting of the MFB50-SA experimental data) and the fitted parameters. The choice of the polynomial order is carried out by weighting the fitting performance and the function complexity. If by increasing the polynomial order, the accuracy improvement is negligible, the last formulation that guaranteed a significant performance step is chosen. Tables 1.11, 1.12 and 1.13 collect the polynomial order,  $R^2$  and RMSE for parameters  $A$ ,  $B$  and  $C$ , respectively (the correlation and the errors are evaluated between the experimental and the fitted parameters). The nomenclature '*polyXY*' indicates that the polynomial is characterized by a degree  $X$  for the first

independent variable (engine speed) and a degree Y for the second one (engine load). All the possible order combinations were tested from *poly11* to *poly33*. Equations 1.53 and 1.54 define the R<sup>2</sup> and the RMSE analytical formulation respectively, while Equation 1.55 represents an example of the *poly33* equation:

$$R^2 = \left( \frac{\sum(c_i - \bar{c})(e_i - \bar{e})}{N(\sigma_c \sigma_e)} \right)^2 \quad (1.53)$$

$$RMSE = \sqrt{\frac{\sum(c_i - e_i)^2}{N}} \quad (1.54)$$

Where:

- $N$  is the number of samples
- $c_i/e_i$  is the generic calculated/experimental value
- $\bar{c}/\bar{e}$  is the calculated/experimental mean value
- $\sigma_c/\sigma_e$  is the calculated/experimental standard deviation

$$Z = p_{00} + p_{10}X + p_{01}Y + p_{20}X^2 + p_{02}Y^2 + p_{11}XY + p_{30}X^3 + p_{03}Y^3 + p_{21}X^2Y + p_{12}XY^2 \quad (1.55)$$

In which:

- $Z$  is the dependent variable (for example, it can be the parameter  $A$ ,  $B$  or  $C$ )
- $X$  and  $Y$  are the independent variables (engine speed and load)
- $p_{lv}$  is the coefficient for the element in which there is a degree  $l$  for  $X$  and a degree  $v$  for  $Y$ .

Table 1.2: Sensitivity analysis for the fitting polynomial order, for parameter A. The green row highlights the polynomial order chosen for the fitting of experimental data.

Polynomial	R2	RMSE [-]
poly11	0.4677	0.0105
poly12	0.7482	0.0101
poly21	0.6116	0.0104
poly13	0.8655	0.0084
poly31	0.7008	0.0099
poly22	0.9545	0.0012
poly23	0.9584	0.0008
poly32	0.9752	0.0005
poly33	0.9813	0.0004

Table 1.3: Sensitivity analysis for the fitting polynomial order, for parameter B. The green row highlights the polynomial order chosen for the fitting of experimental data.

Polynomial	R2	RMSE [-]
poly11	0.5897	0.3199
poly12	0.6736	0.2804
poly21	0.6463	0.3025
poly13	0.7509	0.2587
poly31	0.6877	0.2897
poly22	0.9177	0.1928
poly23	0.9793	0.1483
poly32	0.9709	0.1530
poly33	0.9794	0.1508

Table 1.4: Sensitivity analysis for the fitting polynomial order, for parameter C. The green row highlights the polynomial order chosen for the fitting of experimental data.

Polynomial	R2	RMSE [-]
poly11	0.6624	8.4845
poly12	0.8895	4.9428
poly21	0.7310	7.7122
poly13	0.9449	3.5565
poly31	0.7988	6.7972
poly22	0.9603	3.1408
poly23	0.9879	3.0719
poly32	0.9323	3.8225
poly33	0.9881	3.0008

Tables demonstrate that *poly22* is the best choice in terms of high accuracy and low implementation effort, for all parameters. Moreover, the over-fitting problem is also avoided due to the low fitting function order. The resulting MFB50 model is defined by Equation 1.56:

$$\begin{aligned}
 MFB50 = & (p_{00A} + p_{10A}ES + p_{01A}EL + p_{20A}ES^2 + p_{02A}EL^2 + p_{11A}ES EL) SA^2 \\
 & + (p_{00B} + p_{10B}ES + p_{01B}EL + p_{20B}ES^2 + p_{02B}EL^2 + p_{11B}ES EL) SA \\
 & + (p_{00C} + p_{10C}ES + p_{01C}EL + p_{20C}ES^2 + p_{02C}EL^2 + p_{11C}ES EL)
 \end{aligned}
 \tag{1.56}$$

Where:

- *ES* indicates the engine speed
- *EL* indicates the engine load
- $p_{lvX}$  represents the coefficient that multiplies the two independent variables of degree  $l$  and  $v$  respectively, for the parameter  $X$ .

At this point, the calibration of the parameters  $p_{lvX}$  in Equation 1.56 is carried out through the Ordinary Least Squares Method (OLSM). With this method, the optimized coefficient values are the solution of the linear system of the equations defined for each condition (fixed speed, load, and SA). In this way, it is not strictly required to perform a spark sweep for every considered engine operating point. Equation 1.57 defines the matrix formulation of such system:

$$\overline{MFB50}_{nx1} = \overline{C}_{nxm} \overline{P}_{mx1}
 \tag{1.57}$$

In which:

- $\overline{MFB50}_{nx1}$  is the vector of the experimental MFB50 and  $n$  is the number of recorded data (fixed RPM, load and SA)
- $\overline{C}_{nxm}$  is the matrix of the independent variables and  $m$  is the number of the coefficients  $p_{lvX}$
- $\overline{P}_{mx1}$  is the vector of the of the coefficients  $p_{lvX}$

The solution of Equation 1.57 is the following:

$$\overline{P}_{mx1} = (\overline{C}_{nxm}^T \overline{C}_{nxm})^{-1} \overline{C}_{nxm}^T \overline{MFB50}_{nx1}
 \tag{1.58}$$

Where:

- $C_{nxm}^T$  is the transposed matrix of independent variables
- The symbol -1 indicates the inverted operator

For each point (defined by speed, load, and the applied Spark Advance) an equation can then be written, and the system can be solved when the number of tested conditions is higher than the coefficient number. The described approach does not need the spark sweeps for different engine points. The OLSM allows optimizing all calibration coefficients through the minimization of the error between the calculated and the experimental MFB50, for fixed speed, engine load, and SA. The analytical formulation of the model is still derived from the fitting of MFB50 on the SA axes and then describing the  $A$ ,  $B$  and  $C$  parameters trend on the speed - load plane with a polynomial equation. Even if the calculation of the coefficient is carried out with the OLSM, an important check of the calibration process robustness is the comparison between the calculated surfaces and the experimental parameters of the parabolic relationship between MFB50 and SA (i.e.  $A$ ,  $B$  and  $C$ ). Figures 1.68, 1.69 and 1.70 show the surfaces for such parameters, deduced from the coefficients optimized with the OLSM.

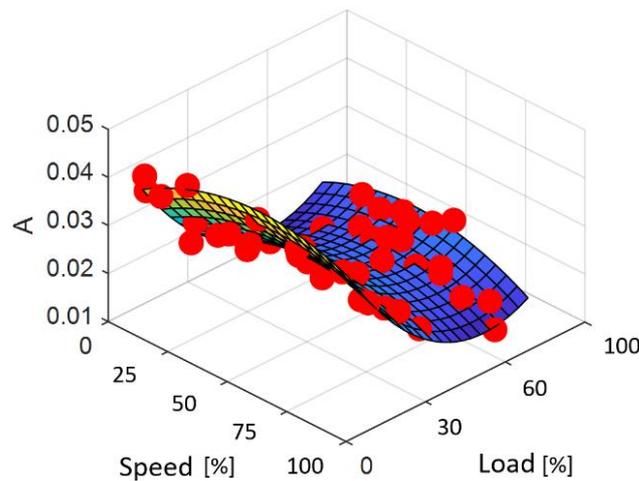


Figure 1.68: Resulting surface for parameter  $A$ . Red markers show the experimental data used for the fitting of the surface.

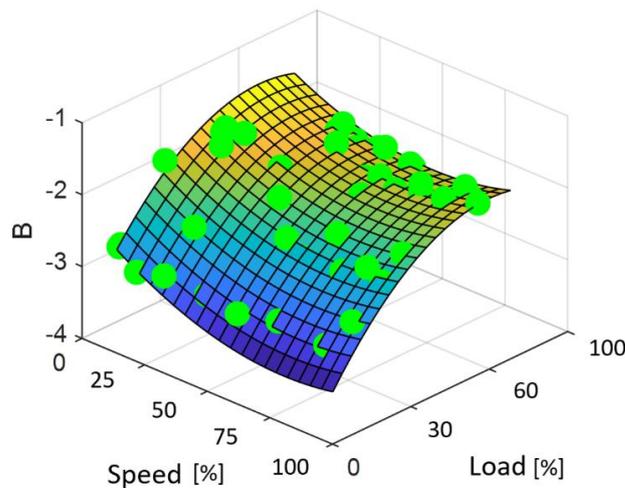


Figure 1.69: Resulting surface for parameter  $B$ . Green markers show the experimental data used for the fitting of the surface.

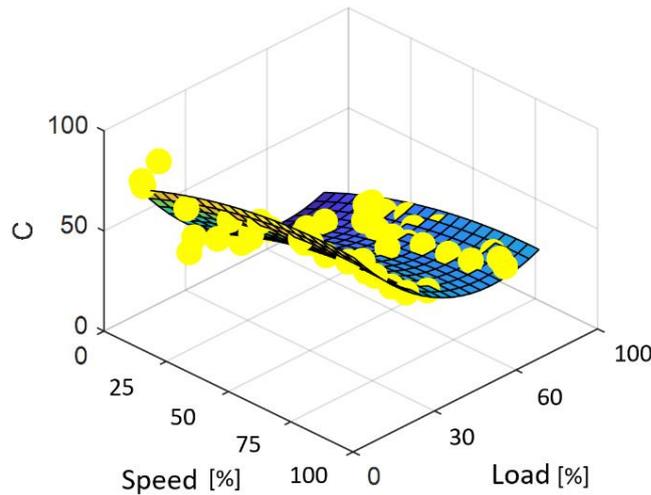


Figure 1.70: Resulting surface for parameter C. Yellow markers show the experimental data used for the fitting of the surface.

The analytical description of such coefficients allows to easily adjust values with some corrective gain (or function) to introduce the sensitivity to other engine variables, such as control parameters that can be changed w.r.t. the mapped values. In chapter 2 MFB50 model is further developed to include the sensitivity of coefficients A, B and C to the injected water mass, to make it suitable to be applied to an engine with a prototyping port water injection system.

The robustness of the MFB50 model was tested by reducing the number of engine points used for the calibration process and verifying the performance on the total amount of experimental data. Figure 1.71 shows the R2 values for fittings with a different number of used data points: it is possible to highlight that the accuracy of the model does not significantly change when a low number of Spark Sweeps are used for calibration. By reducing the calibration data, the engine points are still selected in order to be broad enough to cover the engine operating range. Figure 1.72 shows the errors between the calculated and the experimental MFB50, when the model is calibrated with 20% of available data. The dashed lines include the 92% of data. Such threshold has been arbitrarily chosen to highlight the high model accuracy.

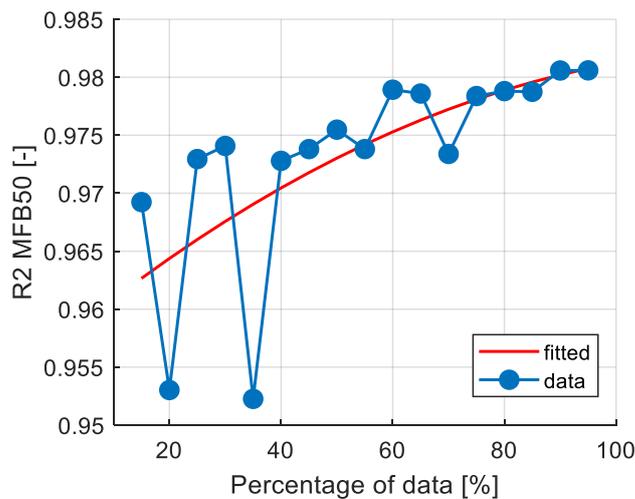


Figure 1.71: R2 correlation coefficient corresponding to fittings with a different number of experimental points used for model calibration. The percentage values indicate the number of points used for fitting respect to the entire data set. 100% corresponds to 60 engine points (i.e. 60 spark sweeps), 20% to 12 engine points.

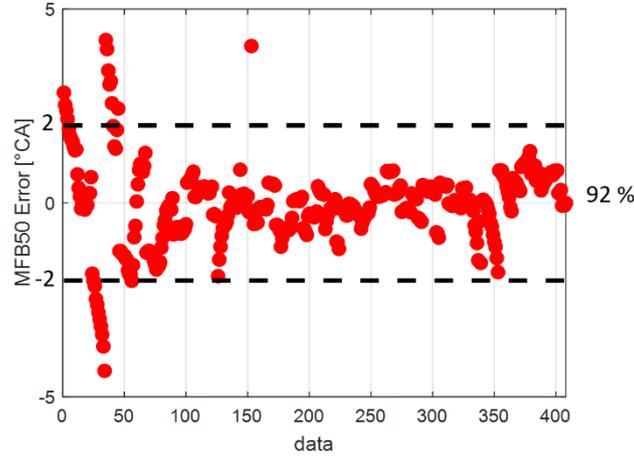


Figure 1.72: Error of the MFB50 model when it is calibrated with 20% of tested engine points (i.e. spark sweeps). The black dashed lines include 92% of data.

### 1.3.3.2. MFB50-to-PMAX Model

The proposed Pmax empirical model is based on the analytical description of the trend of the maximum in-cylinder pressure with respect to MFB50 and engine load.

The method followed to fit the experimental data is the same applied for the parameters  $A$ ,  $B$  and  $C$  of the MFB50 model. In fact, such data has been fitted with a polynomial function, in which MFB50 and load are independent variables. This because the trend was observed more dependent on engine load, instead of speed. As it is well known, when combustion is shifted towards the exhaust phase (i.e. high MFB50 values), the maximum pressure value becomes independent on MFB50 and it corresponds to the maximum pressure in motoring condition. This consideration can be used both when choosing the data for the fitting, and when estimating the Pmax value through the model. For this reason, all registrations with MFB50 greater than 50 °CA ATDC were discarded for the model calibration. Then, the choice of the polynomial order is carried out by evaluating the model performance in terms of R2 and RMSE. As for the MFB50 model, polynomials from  $poly11$  to  $poly33$  were tested. Table 1.14 collects the results of this comparison.

Table 1.5: Sensitivity analysis of the fitting polynomial order, for the Pmax model. The green row highlights the polynomial order chosen for the fitting of experimental data.

Polynomial	R2	RMSE [bar]
poly11	0.9093	6.9864
poly12	0.9717	3.5930
poly21	0.9986	1.8993
poly13	0.9765	5.1045
poly31	0.9994	1.802
poly22	0.9994	1.8977
poly23	0.9994	1.7174
poly32	0.9995	1.6696
poly33	0.9995	1.6681

The selected kind of equation is the  $poly21$  which guarantees high accuracy with a low implementation and computational effort. The resulting formulation of the Pmax model is the following:

$$P_{max} = p_{00P} + p_{10P}MFB50 + p_{01P}EL + p_{20P}MFB50^2 + p_{11P}MFB50 EL \quad (1.59)$$

Where:

- $p_{lvP}$  is the coefficient that multiplies the first independent variable elevated to the degree  $l$  and the second one to the degree  $v$ , for the Pmax model
- $EL$  is again the engine load

For the Pmax model, the polynomial coefficients are optimized by minimizing the error between experimental and calculated data. To avoid the calculation of unphysical values by the Pmax model when the MFB50 value is particularly high (which corresponds to very retarded combustion or misfires) the Pmax value is assigned and equal to the motoring value, which can be estimated through the perfect gas law:

$$Pmax_{|motoring} = \frac{m R T_{max}}{V_0} = \frac{m R T_{inlet} CR^{n-1}}{V_0} \quad (1.60)$$

Where:

- $m$  is the trapped air (or air and fuel) mass, which is directly related to the load, which is supposed to be known
- $R$  is the gas constant
- $T_{max}$  is the end-of-compression temperature
- $T_{inlet}$  is the intake or the start of compression temperature
- $CR$  is the geometric compression ratio
- $n$  is the compression polytropic coefficient, typically chosen equal to 1.34
- $V_0$  is the combustion chamber volume

$T_{max}$  can be taken equal to 450 °C (calculated with the polytropic function and the ideal gas law), which gives reasonable values of Pmax in motoring conditions, for every load condition. The Pmax estimation can be summed up as:

$$\begin{cases} \overline{Pmax} & \text{for } \overline{Pmax} > Pmax_{|motoring} \text{ or MFB50} < 5 \\ Pmax_{|motoring} & \text{otherwise} \end{cases} \quad (1.61)$$

Where  $\overline{Pmax}$  is the estimation through Equation 1.59.

Figure 1.73 shows the experimental Pmax on the MFB50-load plane and the surface derived from Equation 1.61 with the saturation corresponding to motoring conditions. For fixed engine load the model is able to capture the parabolic trend of Pmax with respect to MFB50 when the combustion occurs and its value when the mixture is compressed only.

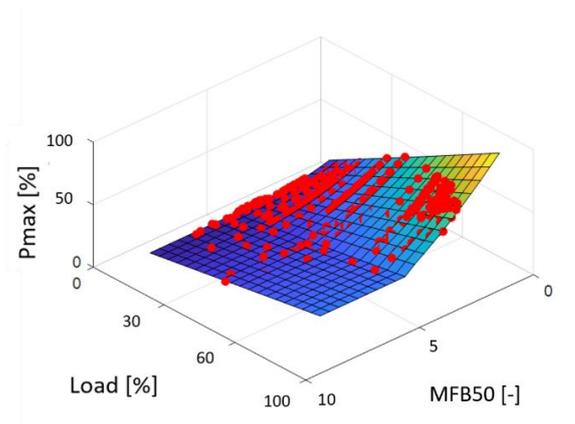


Figure 1.73: Pmax model. The surface represents the graphical trend of the analytical function and the red circles correspond to the experimental data. The MFB50 is expressed as normalized 'SPACE'.

The robustness of the Pmax model was verified with the same method used for the MFB50 model. The number of spark sweeps used for the calibration process was varied and the R2 coefficient was calculated on the total amount of available data. Using 20% of data, the errors performed by the model were then calculated and collected in Figure 1.75. This model is characterized by a high accuracy and very small errors, also when few recordings are considered.

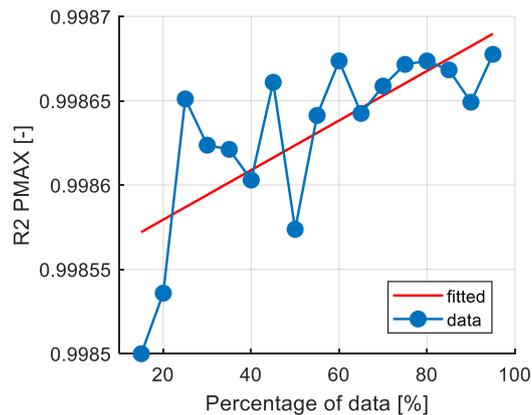


Figure 1.74: Correlation coefficient corresponding to fittings with a different number of experimental points used for the Pmax model calibration. The percentage values indicate the percentage of engine points included in the optimization of the model coefficients. 100% corresponds to 60 engine points (i.e. 60 spark sweeps), 20% to 12 engine points.

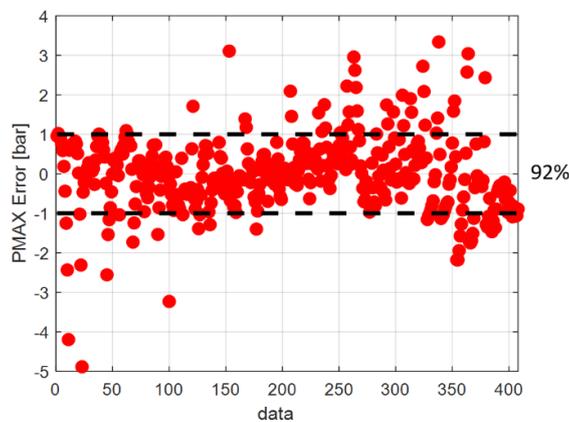


Figure 1.75: Errors of the Pmax model when it is calibrated with 20% of tested engine points (i.e. spark sweeps). The black dashed lines contain 92% of data.

### 1.3.3.3. MFB50-to-IMEP Model

The method to define and calibrate the IMEP model is similar to the one implemented for the Pmax. This new function defines the relationship between the MFB50, engine load and IMEP. In other words, the equation considers the same independent variables used in the Pmax model.

Also for the IMEP model, the experimental data were fitted with a polynomial function. Different polynomial orders were tested, and their accuracy was evaluated in terms of R2 and RMSE. The results of this comparison are collected in Table 1.15.

Table 1.6: Sensitivity analysis to the fitting polynomial order, for the IMEP model. The green row highlights the polynomial order chosen for the fitting of experimental data.

Polynomial	R2	RMSE [bar]
poly11	0.9989	1.1418
poly12	0.9989	1.0023
poly21	0.9992	0.5140
poly13	0.9990	0.9341
poly31	0.9999	0.4781
poly22	0.9999	0.4943
poly23	0.9999	0.4545
poly32	0.9999	0.4361
poly33	0.9999	0.4365

The IMEP trend on the MFB50 – load plane can be captured with a simple polynomial function as well. In fact, the *poly21* function allows to obtain accurate results with a very low computational and implementation effort and for this reason it was chosen for the fitting of the experimental data. Moreover, the parabolic behavior along the MFB50 was expected, because of the well-known bell-shaped efficiency curve. In other words, for a given value engine load the IMEP model describes the parabolic trend of combustion efficiency. Equation 1.62 represents the IMEP analytical model:

$$IMEP = p_{00l} + p_{10l}MFB50 + p_{01l}EL + p_{20l}MFB50^2 + p_{11l}MFB50 EL \quad (1.62)$$

Where:

- $p_{lv}$  is the coefficient which multiplies the first independent variable elevated to the degree  $l$  and the second one to the degree  $v$ , for the IMEP model
- $EL$  is the engine load

The optimization of the equation coefficients was carried out by minimizing the error between the experimental data and the analytical surface.

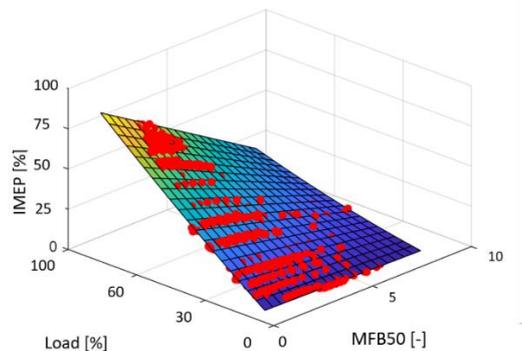


Figure 1.76: IMEP model. The surface represents the graphical trend of the analytical function and the red circles correspond to the experimental data. The MFB50 is expressed as normalized 'SPACE'.

The accuracy and the robustness of the IMEP model was verified by decreasing the number of data used for the calibration process and by evaluating R<sup>2</sup> for each configuration. The model was then calibrated with 20% of available data and the error of the obtained estimations is displayed in Figure 1.78.

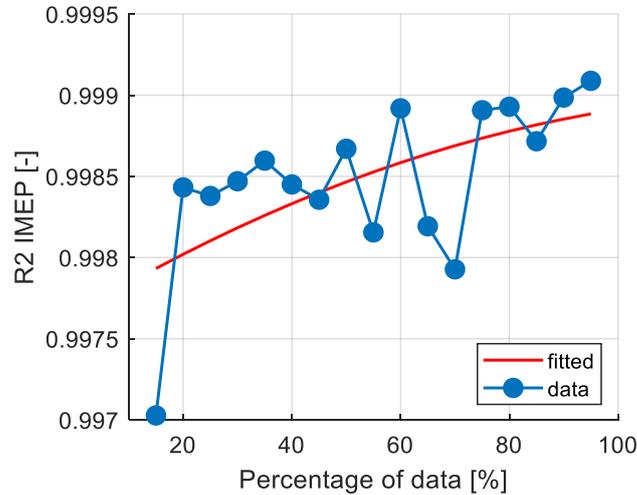


Figure 1.77: Correlation coefficient corresponding to fittings with a different number of experimental points used for the IMEP model calibration. The percentage values indicate the percentage of engine points included in the optimization of the model coefficients. 100% corresponds to 60 engine points (i.e. 60 spark sweeps), 20% to 12 engine points.

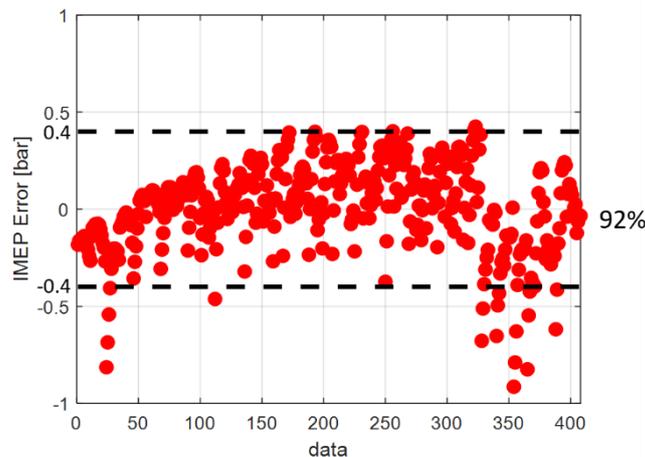


Figure 1.78: Error of the IMEP model when it is calibrated with 20% of tested engine points (i.e. spark sweeps). The black dashed lines include 92% of data.

### 1.3.3.4. Knock Model

The previously described knock model has been calibrated with the new experimental database. The sensitivity analysis of the model performance depending on the number of calibration data is carried out. In other words, starting from the results of the previous chapter, the R<sup>2</sup> coefficient is evaluated while varying the number of data used for the calibration. As for the previous models, 20% of available data are used to calibrate the model coefficients. Figure 1.79 collects the results of the sensitivity analysis. In such figure, R<sup>2</sup> is calculated between the experimental and the calculated 95<sup>th</sup> MAPO percentile (MAPO95). Figure 1.80 shows the model error.

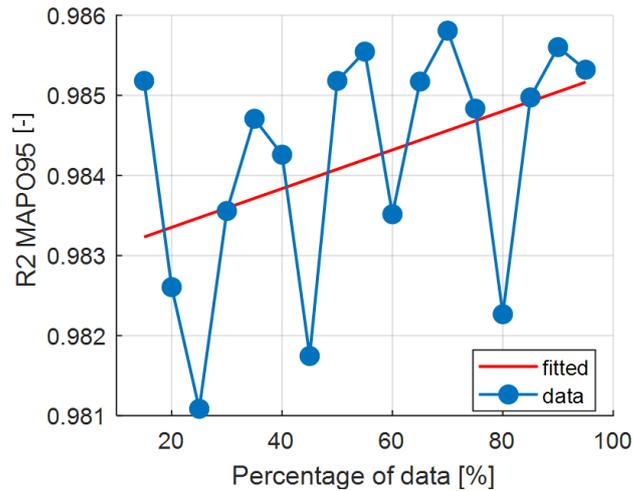


Figure 1.79: Correlation coefficient corresponding to fittings with a different number of experimental points used for the knock model calibration. The percentage values indicate the percentage of engine points included in the optimization of the model coefficient. 100% corresponds to 60 points (i.e. 60 spark sweeps), 20% to 12 points. MAPO95 indicates the 95th MAPO percentile.

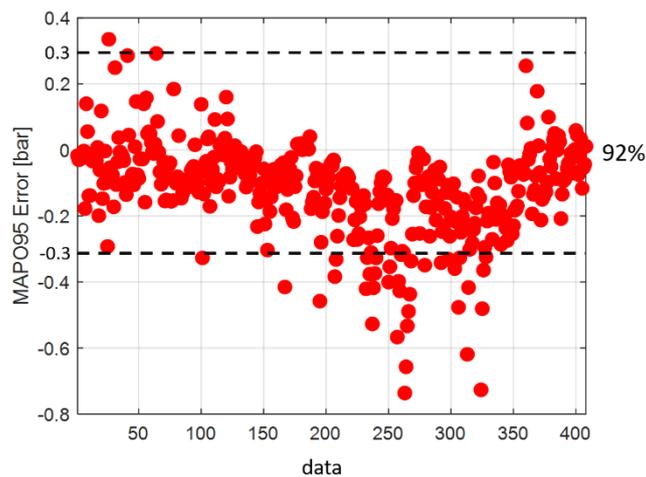


Figure 1.80: Error of the knock model when it is calibrated with 20% of tested engine points (i.e. spark sweeps). The black dashed lines include 92% of data.

### 1.3.3.5. Standard Deviation Model

With the presented models, the mean values of MFB50, Pmax, IMEP and MAPO log-normal distribution are calculated for fixed operating conditions. The proposed knock model is able to generate the cyclic MAPO values, while the MFB50, Pmax and IMEP models are free from the cycle-to-cycle variation (CCV) up to now. The correct way to model the cyclic variability is using a statistical approach. In this way, the cyclic variation of the considered indexes can be described with a Gaussian Probability Density Function [1.81]. In other words, when the standard deviation for each index is defined, the characteristic Normal PDF can be calculated for given speed and load, and the CCV can be generated randomly choosing a value from such distribution. A similar approach is described in the previous section to generate MAPO values when the characteristic log-normal PDF is calculated. One option is to model and predict the STDs for MFB50, IMEP and Pmax, depending on the operating conditions. This requires studying these quantities and to find the proper fitting functions.

Another option is to consider the MFB50 as the only combustion index intrinsically affected by variability; then, the variability of IMEP and Pmax can be totally attributed to the variation of the MFB50, which is one input of the two models. This is not strictly true: for the same MFB50 value, IMEP and Pmax assume different values, attributable to cyclic differences in terms of filling, AFR, internal EGR, etc. Anyway, this second source of noise is less significant, and it will be neglected in this work. Therefore, the STD of the MFB50 needs to be estimated. It was analyzed on the MFB50-load plane and fitted with a polynomial function, as for the previous models. The sensitivity of the results to the polynomial order and the related model performance are presented in Table 1.16, and Figure 1.81 shows the resulting surface. If for the mean value models a high accuracy and robustness is the focus, for the STDs the aim is mostly that to recreate a probable behavior of the simulated indexes. The control-oriented ANESIM is thought to be accurate on the mean values because these are usually the quantities recorded and managed by a control system, while the CCV works like a disturbance for the external controller. For this reason, the activity on STD model is mainly focused on the definition of the fitting polynomial with the lowest computational effort, at the expense of some accuracy. The data collected in Table 1.16 demonstrate that the model performance does not increase significantly with the polynomial order. For this reason, the highest degree equations (i.e. *poly23*, *poly32* and *poly33*) are not considered in such table. The model outputs are saturated at the minimum experimental value to avoid the calculation of negative values.

Table 1.7: Sensitivity analysis to the fitting polynomial order, for the MFB50 STD model. The green row highlights the polynomial order chosen for the fitting of experimental data.

Polynomial	R2	RMSE [°CA]
poly11	0.8363	0.1834
poly12	0.8624	0.1734
poly21	0.8685	0.1708
poly13	0.8673	0.1723
poly31	0.8697	0.1713
poly22	0.8685	0.1713

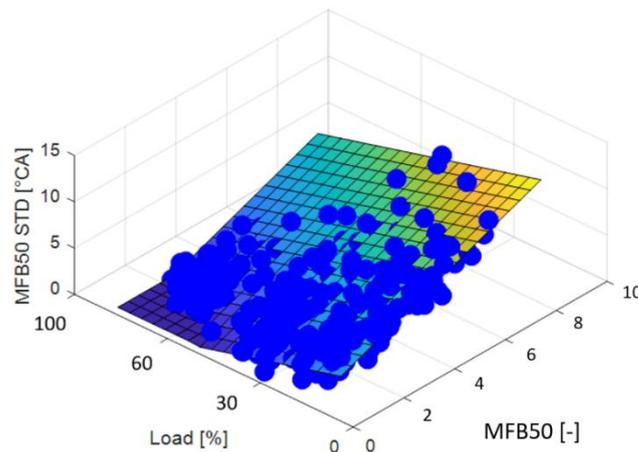


Figure 1.81: MFB50 STD model. The surface represents the graphical trend of the analytical function (saturated at the minimum experimental value) and the blue circles are the experimental data. The MFB50 is expressed as normalized 'SPACE'.

### 1.3.3.6. ANESIM

All the presented models have been calibrated simply through the observation of the dependent variables physical trend with respect to the chosen independent ones, carrying out their respective

polynomial fitting. In other words, the parameters optimization has been performed minimizing the RMSE between experimental and calculated data. The calibration procedure is summarized by the following scheme.

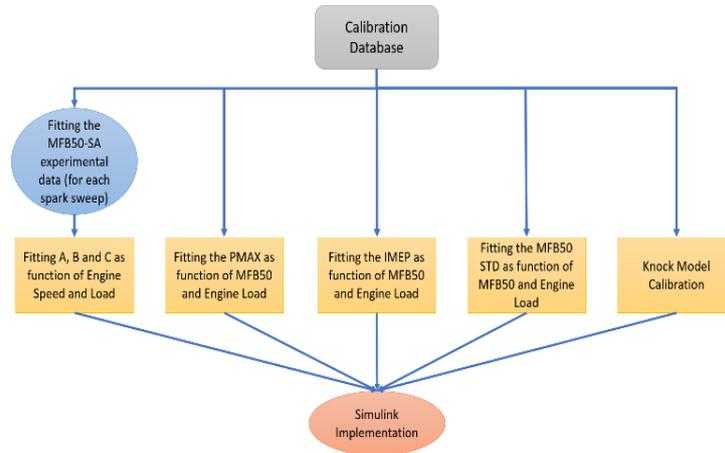


Figure 1.82: Calibration process step scheme. The functional layout of the Simulink model is displayed in Figure 1.66.

The presented models were then implemented in a Simulink environment. The described equations were concatenated to each other, to finally provide all the combustion quantities. For fixed values of engine speed, load, and SA, the MFB50 model estimates the mean value and the STD, which are used to generate the cyclic index by using the PDF definition. Such value is then used by the Pmax and IMEP models, which calculate the corresponding cyclic indexes. The same Pmax model is used to calculate the mean Pmax value, associated to the mean MFB50 (i.e. without the cycle-to-cycle variability) to feed the knock model. This is because the knock model intrinsically considers the CCV for the MAPO calculation, extracting a value from the characteristic log-normal PDF of the operating conditions. Moreover, if the cyclic Pmax value is used as input for the knock model, a combination of two probability functions would occur (this aspect will be further investigated in future studies). When the speed and load profiles are given and used as inputs, ANESIM calculates the cyclic combustion indexes. The SA and lambda maps are introduced in the simulator and the lambda map has the same values used during the spark sweeps performed to produce the database used for the calibration process. As introduced above, in this work the dependency of MFB50, Pmax and IMEP on lambda or VVT phasing has been considered as negligible. Instead, the knock model needs lambda as input.

To verify and validate the resulting performance of ANESIM, the outputs calculated by the simulator are compared with those generated by the reference engine, when the same speed and load profiles are used as inputs for both systems. Even if the performance of each model is already evaluated and presented above, the validation of the resulting simulator allows to quantify the effects of error propagation from one model to the other. The error on the mean combustion indexes is calculated, while that between the modelled and the experimental STDs are neglected in this phase. The modelling of standard deviations has in fact the aim of introducing a realistic cyclic variability of the mean values, which is an important aspect during the development of a novel combustion control strategy.

Figure 1.83 shows the speed and load profiles used to drive the real engine and ANESIM. The described profiles were executed at the test bench and with the analytical simulator, and were repeated twice, with very similar speed-load profiles and different SA maps. The tested profile consists of a portion of a real vehicle maneuver and it includes very steep transients but also semi-

steady state conditions (in the first and in the last part of the test). In this way the simulator is stressed with a realistic profile and the results may demonstrate the high reliability of the proposed tool. During these tests, the simulator is forced to calculate the modelled combustion indexes also for engine points outside the calibration database and this demonstrates the good interpolation and extrapolation capabilities of the models. Each profile was tested with two different SA maps, which are labelled as “first” and “second” SA map. In this way, the sensitivity of the model to SA was further validated. Such maps were defined to carry out the validation tests only (they are not standard calibration maps) with the aim to demonstrate that the simulator is able to correctly capture the influence of the spark advance on the considered combustion indexes. At the same time the validation tests prove it is not necessary to include the neglected variables (like lambda and VVT) for this kind of application, because their mapped values are not frequently modified by the ECU control strategies during a real driving cycle. Of course, these dependencies become relevant if the mapped values are often adjusted by ECU.

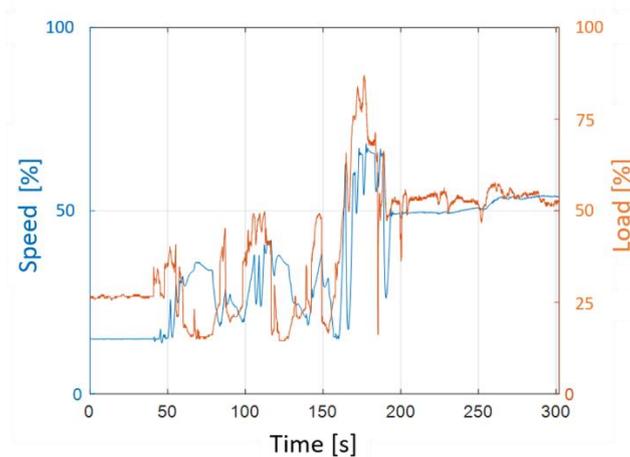


Figure 1.83: Speed and load profiles for validation.

The cyclic values of the calculated and experimental combustion indexes were synchronized and overlapped. In Figures 1.84 and 1.86 the recorded and the simulated data are shown. The green and the blue points represent the cyclic combustion indexes calculated by ANESIM for MFB50, Pmax, IMEP, and MAPO respectively. In such figures a very good agreement between the simulated and the experimental data is observed. In particular, it is possible to highlight that the calculated mean values and the CCV are strongly coherent with the same quantities produced by the real engine (data of 4 cylinders are only reported for clarity). In this way, when ANESIM is coupled with a combustion controller which has to be developed or validated, it guarantees a feedback that is very close to that of the real engine on the test bench. For each test, the difference between the on-line registered mean combustion indexes and those simulated by ANESIM are calculated and displayed in Figures 1.85 and 1.87. The experimental cyclic MFB50, Pmax and IMEP trends were filtered through a moving average on a buffer of 100 elements and the error was evaluated subtracting the mean values calculated by each model (the blue, brown, and green boxes of Figure 1.66 respectively). For the 95<sup>th</sup> percentile of MAPO, the error was calculated as the difference between the experimental and the simulated data (the yellow and the brown lines in the fifth subplot respectively), both calculated on a buffer of 400 elements. Such figures demonstrate that the simulator is able to produce results with a mean absolute error on MFB50 around 1°CA, 5 bar on Pmax, 2 bar on IMEP and 0.4 bar on MAPO95. The results confirm the high accuracy and robustness of the proposed analytical models and, more generally, of the empirical approach.

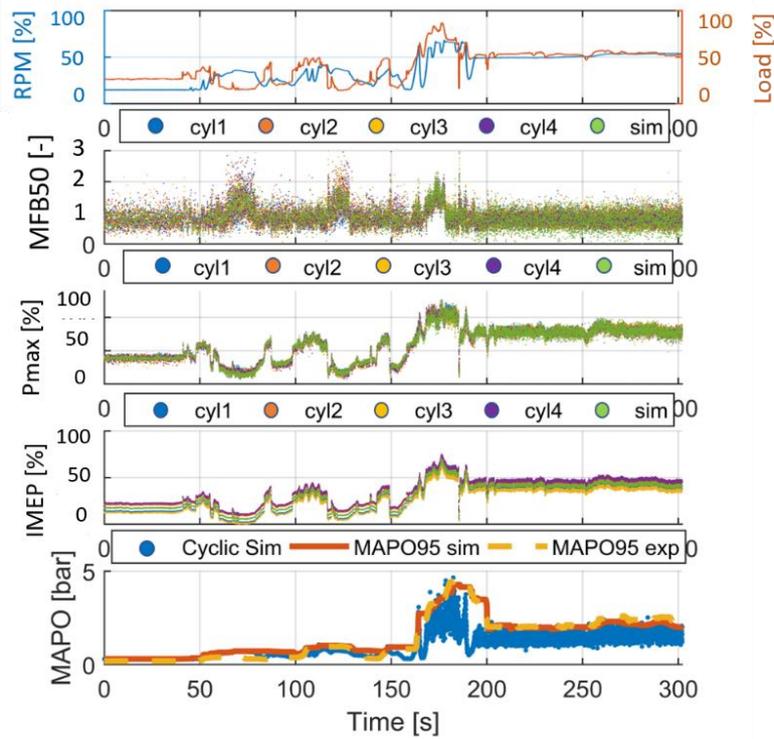


Figure 1.84: Engine and ANESIM combustion indexes when the same speed and load profiles are used as inputs. This test was performed with the first SA map. Only 4 cylinders are included for clarity. The MFB50 is expressed as normalized 'SPACE'.

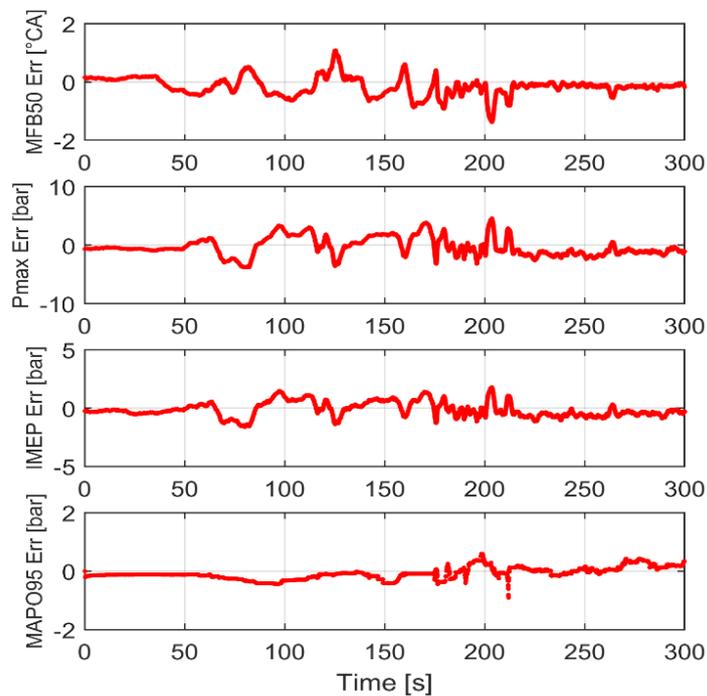


Figure 1.85: Error made by ANESIM during the simulation with the first SA map. Each point represents the difference between the experimental moving average of the mean cylinder and the calculated mean value of the given combustion index, which is updated after each combustion. For MAPO95 a buffer of 400 elements was used to calculate the 95th percentile.

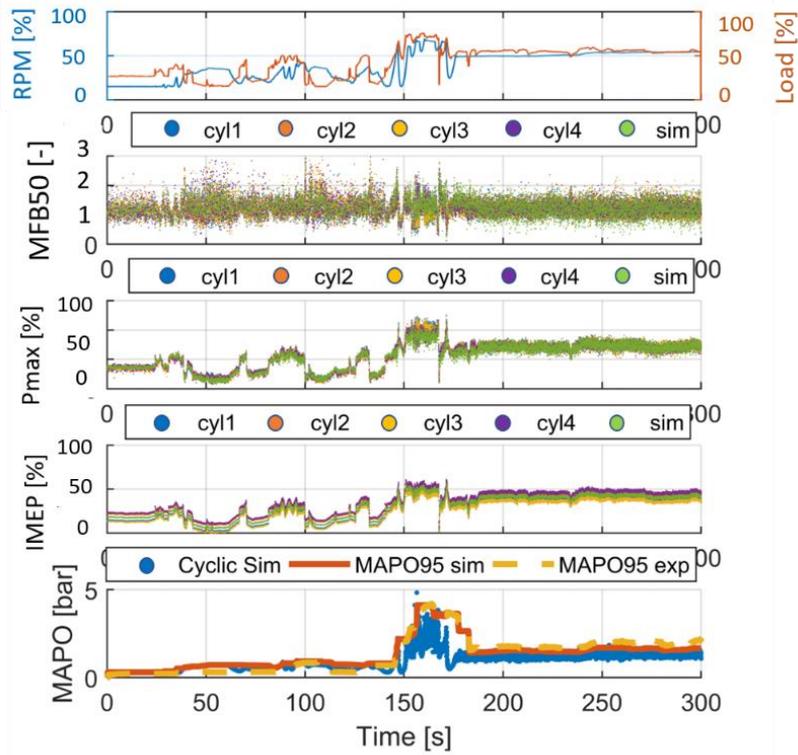


Figure 1.86: Engine and ANESIM combustion indexes when the same RPM and load profiles are used as inputs. This test was performed with the second SA map. Only 4 cylinders are included for clarity. The MFB50 is expressed as normalized 'SPACE'.

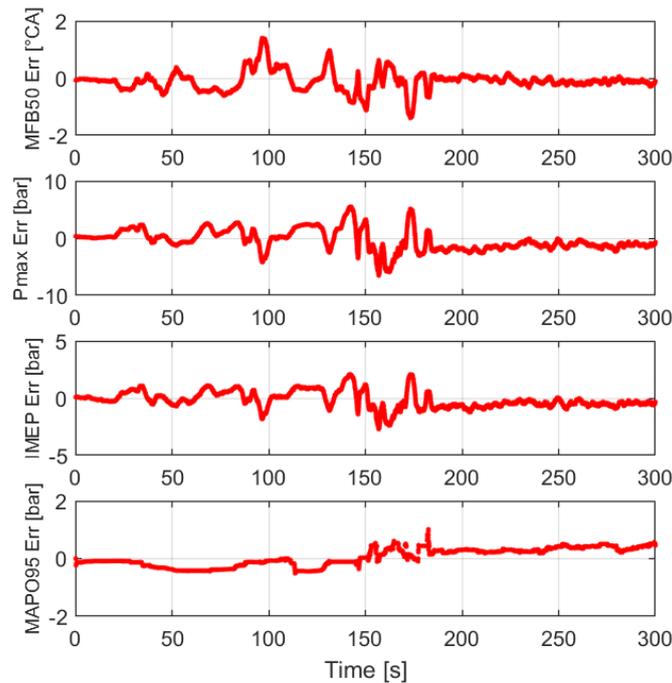


Figure 1.87: Errors made by ANESIM during the simulation with the second SA map. Each point represents the difference between the experimental moving average of the mean cylinder and the calculated mean value of the given combustion index, which is updated after each combustion. For the MAPO95 a buffer of 400 elements was used to calculate the 95th percentile.

Results confirm the high accuracy and robustness of the proposed models and the resulting simulator. ANESIM is a valid tool to simulate the combustion indexes trends when few combustion data are available for the calibration process. This limited amount of experimental data is often available because they are necessary for the engine benchmarking and for base engine maps calibration. This is another point in favor of the proposed approach. Moreover, ANESIM offers high accuracy with an extremely low computational effort, thanks to its simple analytical formulation. This allows to test a combustion control strategy via Software-in-the-Loop, and easily implement an extremely reliable engine simulator for Hardware-in-the-Loop applications, also when high calculation capabilities are not available.

More generally, it has been demonstrated that this kind of modelling methodology allows to define synthetic combustion indexes with an extremely high accuracy, including the engine parameters that are effectively changed with respect to the maps when the engine runs and that concretely affect combustion process. These models are based on the effect separation method that guarantees reliable results. In this way combustion indexes can be calculated also when they are not measurable, due to the absence of the in-cylinder pressure sensing or when the on-board combustion diagnosis system cannot provide a complete in-cylinder pressure trace. Such models can be executed at the combustion frequency also when implemented in a real-time machine, such as the ECU.

These observations make the models proposed in this chapter particularly suitable to be implemented in an innovative combustion control system because they can provide statistical values of the main combustion indexes (that are the values typically used to manage the combustion, to avoid the effects of the CCV) when the on-board indicating system is not available. Otherwise, they can integrate the data that come from the indicating system when the complete in-cylinder pressure trace is not available. This can happen when the on-board technology cannot provide indexes that came from the integral pressure curve. During PhD course a novel combustion control strategy based on these models has been developed, and it is described in the next chapter.

## 1.4. Knock Control

As widely clarified in previous chapters of this work, knock control is a key topic in the research and development field of modern SI engines. The previous chapter deals with knock modelling and some innovative contributions for knock-induced damage and knock intensity estimation are thoroughly described. Knock intensity evaluation can be completely redefined through a concrete and more meaningful index that directly indicates the induced erosion on the piston surface, as a function of piston temperature. A completely new way for knock intensity evaluation and the related control-oriented piston damage model are therefore discussed and an extremely simple analytical knock model is defined, allowing to directly calculate a statistical knock index (e.g. a percentile of MAPO) when the maximum in-cylinder pressure ( $P_{max}$ ) is known, for fixed engine operating conditions, i.e. engine speed, load,  $\lambda$  and fuel quality. The analytical formulation makes such model highly powerful because the mathematical function can be easily manipulated and reversed to have the knock index as input and another variable included in the equation as output, such as the P<sub>MAX</sub>. In this way, it is possible, for instance, to convert a target value of a statistical knock index into the corresponding mean value of maximum in-cylinder pressure. Such flexibility is the most important feature of the analytical approach because the reversed knock model can be directly implemented in a combustion control strategy to target a given value of the knock intensity managing a mean combustion index, which can be achieved with high accuracy. More generally, the development of a control algorithm based on the piston damage model coupled with the analytical knock model can be considered as a fundamental step to innovate the standard closed loop-based approach commonly used to control knock intensity. The description and the experimental validation of such model-based controller is the focus of this chapter and main experimental results are shown. In the second part of this chapter, a novel and viable solution for the on-board combustion phase estimation is finally described.

### 1.4.1. *Adaptive piston damage-based combustion controller*

Knock presents a completely different challenge with respect to that which is commonly encountered when a standard dynamic system has to be controlled. Main reasons for this deep difference between knock events and other combustion phenomenon can be synthesized in two wide issues:

1. The lack of a robust relationship between values of knock intensity indexes and the associated damage on the combustion chamber components
2. Knock stochastic nature

On one hand, as it has been observed in the second chapter of this work, numerous indexes for the knock intensity estimation have been defined from the high frequency content of the measured or estimated in-cylinder pressure trace [1.19], from both the high and low frequency range of such signal [1.82], or from the high-frequency vibrations induced by abnormal combustions on the engine head [1.83]. Regardless of the source of such signal and the mathematical formulation of the intensity index calculation, the totality of such indexes does not provide a direct indication about the consequences that knocking event has on the combustion chamber, such as the erosion on the piston surface. On the other hand, the non-deterministic nature of knock forced the control systems developers to define some statistical quantities, i.e. some percentiles values of a certain knock index (typically higher than the 90<sup>th</sup>) to manage the event rate over a given threshold [1.38], rather than the cyclic knock intensity, that closely approximates an independent random process. Both these

points have led over the years to the diffused consolidation of a partially or totally unaware approach to knock control, with respect to the damage induced by abnormal combustions especially on the aluminum components. Knock intensity thresholds and criteria used for the SA calibration process are typically the result of the individual experience and, in most cases, they are defined to avoid knocking combustions at all. The highest percentiles (such as the 98<sup>th</sup> or the 99<sup>th</sup>) or even the cyclic intensity values are controlled to avoid the complete exceedance of an extremely precautionary limit. Moreover, such thresholds have not been updated over the years and they did not follow the evolution of the materials of the combustion chamber components. Indeed, the improvement of the alloys mechanical characteristics has led a significant increase of mean specific power, but it has not been accompanied by a corresponding redefinition of knock intensity thresholds. This is certainly caused by the lack of a robust quantitative relationship between the calculated knock index and the corresponding induced damage.

Standard knock control strategies typically use the signal that comes from an on-board compatible system for knock intensity sensing and they move the spark advance at every time step. The spark timing is advanced whenever the intensity is lower than the threshold, while it is retarded when knock occurs, applying fixed corrections. Nevertheless, the majority of on-board algorithm are designed to have only the capability to reduce the SA, with the purpose to prevent any kind of abnormal combustion. In fact, the common approach is that to calibrate the SA maps that allows to maintain the knock intensity under the threshold for nominal operating conditions, using the knock controller just to retard SA when the on-board system senses “high” intensity knocking events according with spark timing correction maps. The SA retarding is then gradually reduced, reaching again the mapped value, and this kind of controller is the so called “sawtooth”. In other words, on-board control algorithms are designed to reduce knock intensity, but they cannot target an event rate value. In some rare cases, standard sawtooth controller is typically based on closed loop approach where the feedback sent by knock intensity sensing system about the current cycle is used to define the magnitude of the control move for the following one. Nevertheless, as demonstrated in [1.39, 1.40], this method leads to have a really active control behavior that is often unacceptable. The extremely reactive response is also due to the controlled high percentile (or the low event rate) that is affected by a strong variability caused by knock stochastic nature. Even if some techniques to reduce the variance of the actuated SA corrections exist [1.73, 1.74], the closed loop-based controller is intrinsically limited, because its actuation is always determined by the feedback of the system [1.84]. For example, PID controller determine the correction magnitude w.r.t. the error between the current input value and the target. The transfer function of a standard PID controller is defined by the Equation (1.63):

$$u(t) = K_p \left[ e(t) + \frac{1}{T_i} \int e(t) + T_d \frac{de(t)}{dt} \right] \quad (1.63)$$

Where  $K_p$  is the proportional gain,  $T_i$  and  $T_d$  are the time constant of the integral and derivative action, respectively. As well known, on one hand, the PID controller is particularly robust, but, on the other hand, it is particularly suitable to manage systems that commonly work mainly under steady-state conditions. The controller performance under transient conditions strongly depends on values of calibration constants but, however, it cannot maintain an engine parameter on target when engine runs under operating conditions equal or like those faced during the final on-vehicle application. For this reason, in most cases, the PID controllers used to manage knock intensity are calibrated with a precautionary approach and this leads to a further combustion efficiency loss. In summary, the two points listed above led to two controlling issues that can be resumed in the following way:

1. The definition of a threshold for statistical knock indexes that are defined independently from the damage induced by knocking events
2. The implementation of closed loop controllers that are unsuitable to control an extremely variable engine parameter, such as a high knock intensity percentile, especially when engine runs under fast transient conditions.

The knock control algorithm proposed in this section is a revolutionary solution to solve both these issues and the two key tools that allow to face in a completely innovative manner the management of knock intensity are the control-oriented piston damage model [1.77] and the analytical knock model [1.79], described in chapter 1.3. The first one is a model for the real time estimation of the piston surface erosion, that is calculated as a function of the piston temperature and knock intensity (evaluated through MAPO index), while the second one is a knock intensity model that analytically defines the relationship between a MAPO percentile and the mean Pmax, for fixed operating conditions (i.e. engine speed, load, lambda and fuel quality). As mentioned above, the analytical formulation allows to easily reverse the equation and to define the MAPO percentile as input and the mean Pmax as the output. In this way a target value of a certain statistical MAPO index can be directly converted in a target value for the average Pmax. These two models can be considered as one of the most significant recent contribution of the author research group, especially because they allowed to develop the complex open loop control chain described below that has the capability to convert a target piston damage in a combustion phase value. Moreover, the 50% of Mass Fraction Burned (MFB50) analytical model described in section 1.3.3.1 is reversed to use the target MFB50 as input and directly calculate the corresponding spark timing that allows to reach such combustion phase value, for given operating conditions. The reversed analytical knock model and the MFB50 model are further developed implementing adaptive strategies to automatically learn fuel RON of vehicle tank and the analytical parabolic relationship between MFB50 and SA, respectively. Indeed, specific self-learning algorithms have been developed by the authors to guarantee an always up to date fuel RON value and MFB50-SA functions. Nevertheless, such algorithms need the cyclic indexes of knock intensity (MAPO) and combustion phase (MFB50). This means that proposed control strategy needs an on-board system for knock intensity and combustion phase estimation, because such combustion indexes are used as inputs for the self-learning algorithms.

Nowadays, the need of an on-board indicating system or a solution for the calculation of both knock intensity and combustion phase indexes has not to be considered as a stringent limitation. In fact, recent anti-pollution policies as well as the latest applications of Low Temperature Combustion (LTC) that are affected by poor combustion stability and the risk of knock or misfire are forcing automotive companies to develop on-board compatible sensors for in-cylinder pressure sensing. Several solutions are available in literature, and some of these have been patented [1.85-1.89]. For instance, accelerometer is considered as a viable and robust solution, not only for knock intensity estimation, but also for MFB50 calculation [1.83]. Ionization current is a valid technology for the in-cylinder pressure signal estimation [1.90]. A third viable method is that based on the acoustic emissions that can be measured with microphones [1.91]. Finally, some low-cost force transducers have been developed for in-cylinder pressure sensing. Using piezoelectric materials, the forces acting on the engine head may be partially related to the pressure of the gases inside the cylinder. Thus, instrumenting the head stud bolt [1.92] or using a piezoelectric spark plug washer and acquiring at high frequency the electric output signal, the in-cylinder pressure variation can be estimated. Alma Automotive recently developed a piezoelectric spark plug washer that achieved very promising results [1.93]. Following section deals with a novel algorithm for the accurate estimation of combustion phase from accelerometric signal that can be considered as a viable solution for MFB50 estimation.

These considerations about in-cylinder pressure sensing techniques for on-board applications allow to legitimate the development of accurate and reliable combustion control strategies based on the availability of some combustion indexes, that until a few years ago were available only at the test bench.

With the described knock control strategy, the authors are thus proposing a solution for both issues listed above:

1. The target value of MAPO percentile derives from the admissible piston surface erosion in a certain amount of time (i.e. a target damage speed) and this means it is reasonably defined according to the relationship between knock intensity and the induced damage. Pmax and MFB50 models then convert that value into a target combustion angular phase
2. The open loop chain allows to achieve the target MFB50 calculating the necessary SA just before each cycle. The controller is thought to be cycle-solved, that makes such algorithm suitable to accurately manage the combustion phase even under transient conditions.

Outside the engine operating field in which the SA is limited due to knock, the MFB50 model continues to calculate the SA to maintain the combustion phase close to the target value. This condition can be easily identified by calibrating a saturation value equal to the combustion phase of Maximum Brake Torque (MBT). When the target MFB50 (the value that guarantees the target piston damage) is lower than the calibrated value, the MBT combustion phase becomes the target. For this reason, such controller efficiently integrates knock control and combustion phase management, and it can be considered as a combustion controller. Moreover, the adaptive strategies update the coefficients of knock and MFB50 models, using sensed combustion indexes. Results confirm the robustness of these algorithms that make the closed loop chain movements very limited. In fact, it is implemented to react only for protective actions, when knock intensity exceeds an extremely high threshold.

In the following section, the experimental setup and the database used for the calibration and the validation of models is described. Each model is then briefly introduced, retrieving parts of the previous chapter in which such models have been described and validated. Variations needed for their implementation in the combustion controller are presented. The complete combustion controller is described in detail, even showing the mathematical reformulation of some models needed for their implementation in open loop chain and their coupling with self-learning strategies. The controller is then implemented in a real time machine and tested online at the test bench via Rapid Control Prototyping (RCP). Even the RT device used for experimental validation of combustion controller via RCP is developed during PhD course and its development is described in the Appendix. Tests are carried out by reproducing at the engine test cell the speed and load profiles recorded during real vehicle maneuvers performed on track. Results are finally collected and discussed in the last part of this chapter.

#### **1.4.1.1. Experimental Setup**

The experimental tests for controller validation are carried out on the 8-cylinder engine used for the calibration of piston damage and knock intensity model presented in chapter 1.3. Engine characteristics and the layout of network used to test custom controller are presented in the previous Table 1.8 and Figure 1.19, respectively. Likewise, the indicating system is composed by CHAMPION charge amplifier and OBI-M2 indicating system by Alma Automotive, that are used for signal conditioning and acquisition. The calculation of MFB50, IMEP and Pmax combustion indexes uses the low-pass filtered pressure trace while the same signal is band-pass filtered to calculate the MAPO

knock index, as defined by Equation 1.10. Data used for the controller validation was logged during tests performed reproducing different engine speed and load profiles logged during real vehicle maneuvers. However, engine characteristics and communication network layout are reported in Table 1.17 and Figure 1.88 for reasons of clarity.

Table 1.17: 8-cylinder engine characteristics

Displaced volume	3855
Stroke	82 mm
Bore	86.5 mm
Connecting Rod	143 mm
Compression ratio	9.4:1
Number of Valves	4
Exhaust Valve Open	540° BTDC @ 0.1 mm lift
Exhaust Valve Close	360° BTDC @ 0.1 mm lift
Inlet Valve Open	360° BTDC @ 0.1 mm lift
Inlet Valve Close	180° BTDC @ 0.1 mm lift

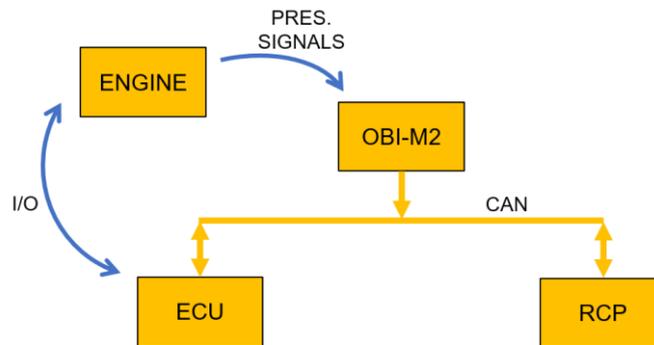


Figure 1.88: Functional layout of communication loop.

As mentioned above, combustion controller is tested through its deployment in the RT machine for RCP application that communicates with ECU and indicating system via Controller Area Network protocol. For each cylinder controller calculates a SA correction that is communicated to the ECU via CAN and it receives main engine parameters (engine speed, load, lambda, etc.) and combustion indexes from the engine control unit and OBI-M2 system, respectively.

#### 1.4.1.2. Implemented Models

Differently from the previous chapter in which knock intensity, Pmax and MFB50 model have been calibrated to reproduce the mean cylinder behavior and have been individually validated, in this case each model has been specialized for a single cylinder. This approach is necessary to accurately control knock intensity and combustion phase including the cylindrical variability. Implemented self-learning algorithms ensure an adaptation of knock and MFB50 one parameters to the specific cylinder even when they are calibrated for the mean one. Nevertheless, the purpose of these adaptive strategies is that to count only the effects of variables that can change during the vehicle life and that cannot be predicted during the calibration process. Fuel quality and the relationship between the engine operating conditions and the corresponding combustion phase are elements that can vary and that strongly influence the engine knock tendency and the SA needed to reach a given target value of MFB50. Indeed, as it will be thoroughly described below, stored data allow to update the current fuel RON value used by knock model and the parabolic relationship between SA and MFB50

as a function of engine speed and load. The analytical formulation of such models is the key feature that allows to directly update model parameters solving a simple equation, deduced by reversing model function. In this way, adaptive strategies have been easily implemented in the control algorithm.

Knock, P<sub>MAX</sub> and MFB<sub>50</sub> models have been developed to analytically describe the effects of engine operating conditions and other influent parameters, such as lambda and fuel RON (especially in knock model), on a certain MAPO percentile, on average maximum in-cylinder pressure and combustion phase, respectively. They are developed by applying the separation effects method and only the engine parameters that significantly change when engine runs and that effectively affect such combustion indexes are included in these models. In other words, as stated in the previous chapter, analytical models have been thought and developed for a calibrated production engine and this means that not all the variables that can influence the combustion process have been included in the modelling process. Valve timing, for instance, is an engine parameter that has a strong impact on the combustion, but the Variable Valve Timing (VVT) map is one of those calibrations that rarely changes, and it can be considered frozen after the ECU calibration procedure. Spark timing and lambda are varied by specific strategies that react to manage, for instance, the catalyst heating, the turbine inlet gas temperature as well as the traction of the vehicle and so on. On the contrary, fuel quality and intake air temperature are uncontrollable variables that undoubtedly change during the vehicle use. Such observations allow to state that the values of main combustion indexes are the result of the ECU calibration. Hence, as described in the previous chapter, proposed approach is focalized to include only the parameters that can be significantly changed during the engine operation. Such method for the prediction of the effect of operating condition variation on combustion indexes can be considered robust and reliable, especially when the final purpose of modelling activity is that to develop control-oriented algorithms for a production engine.

In this section, the piston damage model, the analytical knock intensity, P<sub>max</sub> and MFB<sub>50</sub> ones are briefly described, highlighting those features that have to be considered to better understand the development of the combustion controller. Results presented below are only a portion of those achieved with the specific activity carried out to develop each model that have been described in the previous chapter. Key features of each model are reported to clarify the controller development. All the variables and parameters have been normalized or partially offset, for confidentiality reasons.

#### **1.4.1.3. Piston Damage Model**

Starting from equations 1.22, 1.23 and 1.24, cyclic piston damage speed is calculated by piston temperature model coupled with the knock intensity threshold, using the engine speed, P<sub>max</sub> and MAPO as inputs, as shown in Figure 1.20 of previous chapter. These models are the solution proposed by the authors for the first issue discussed in the introductory part of this section. Indeed, they represents a validated and reliable tool for the RT conversion of the cyclic MAPO value into the corresponding piston surface erosion. Thus, such approach translates a no practical knock intensity index in a more significant and tangible damage index. However, MAPO remains a knock intensity index that can be directly measured through the in-cylinder pressure sensing system, while the piston surface erosion can be only calculated through a model. Considering the final purpose of the feed-forward combustion controller, that consists in converting the target damage speed into a target combustion phase, the translation of the target piston surface erosion into combustion index that is strongly correlated with the MFB<sub>50</sub> is needed. Analytical knock model previously discussed becomes, thus, a useful tool to perform mentioned conversion (target statistical MAPO percentile can be converted into the corresponding mean P<sub>max</sub> value). In fact, the simple analytical

formulation makes it particularly suitable to be used for this application. Starting from Equation 1.36, MAPO percentile becomes the input as well as P<sub>MAX</sub> becomes the output, simply by reversing the model equation, as deepened below. Nevertheless, this means the target damage speed has to be converted into a target MAPO percentile (that can be arbitrarily chosen) to supply the reversed analytical knock model. Moreover, as mentioned in the introduction, an adaptive strategy for the update of knock model parameters related to the fuel RON is developed and implemented in the final controller. Thus, independently of the used knock model, the RON fuel self-learning algorithm must be based on a measurable quantity. This means that knock model must have a pressure-based knock index as input. Even if a more detailed description of the implemented knock model follows below, the schematic layout of the proposed controller is described in Figure 1.89, to clarify the calculation step that follows the estimation of the instantaneous piston damage.

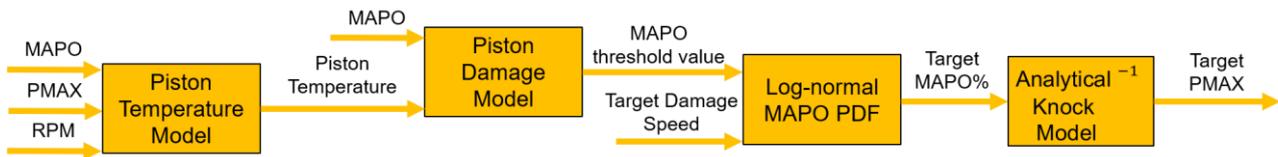


Figure 1.89: Controller layout, including the calculation of target MAPO% and the reversed analytical knock model.

Depending on both the operating conditions and knock intensity, the piston temperature model estimates the temperature of the piston location usually interested by damage (exhaust valve reliefs, in this case) and based on this, MAPO threshold between safe and damaging intensities is assigned. It is reported in Figure 1.90 that is deduced from 1.34 (a).

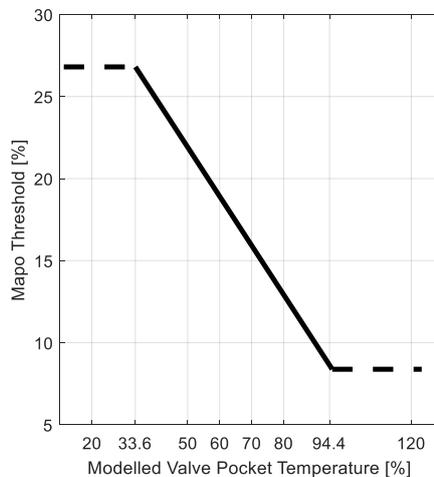


Figure 1.90: MAPO threshold as a function of piston temperature.

Thus, for given MAPO threshold and for given knock intensity log-normal probability density function (PDF), it is possible to calculate the cyclic knock damage as the integral of MAPO values over the threshold. Consequently, given the instantaneous estimated piston temperature and the targeted damage speed, the corresponding MAPO PDF (within the  $\infty^1$  possible distributions, depending on the spark timing, for the given operating condition) can be identified and converted into the value of a given MAPO%. Such value is finally converted into the corresponding mean (considering the CCV) P<sub>MAX</sub>, by reversing the knock model.

The reversed analytical knock model needs that the target damage speed is converted into a target value of the fixed MAPO percentile, that is used as input of the model. For a certain MAPO PDF, the related average damage per cycle is graphically represented in Figure 1.91 and it can be calculated

through Equation 1.64, where  $\Delta MAPO$  is the interval used to discretize the MAPO domain and PF is the probability function.

$$Damage\ per\ cycle\ [bar] = \int_{THR}^{+\infty} (PDF\ \Delta MAPO) dMAPO = \int_{THR}^{+\infty} PF\ dMAPO \quad (1.64)$$

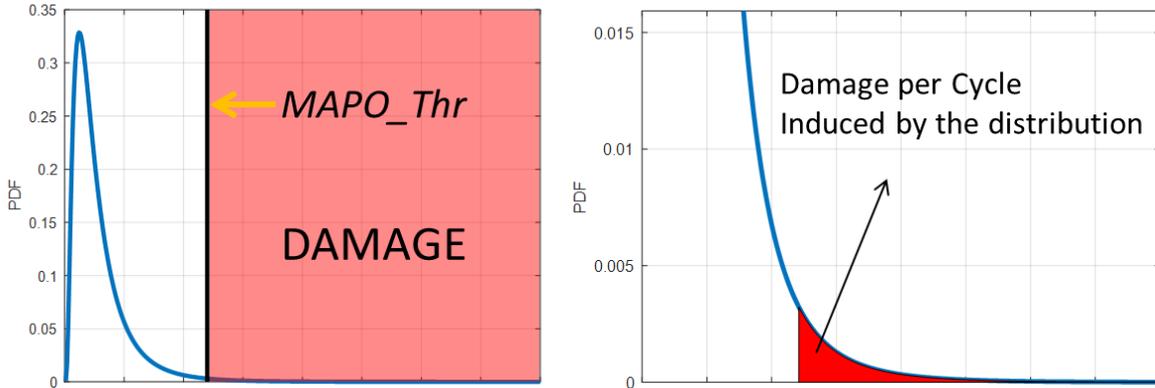


Figure 1.91: Damage per cycle associated to the MAPO distribution for the given MAPO threshold.

An iterative calculation of the target PDF (and corresponding MAPO percentile) has been implemented in the control strategy. When the log-normal PDF that fits the integral value calculated through Equation 1.64 is identified, the corresponding value of MAPO percentile can be used by the knock model as input. Such algorithm uses the linear function introduced above between two percentiles of a log-normal distribution to halve the independent variables and the PDF properties to calculate the mean value ( $\mu$ ) and the standard deviation ( $\sigma$ ) when two percentiles are known (see section 1.3.2.3). The linear function found between MAPO50 and MAPO98 is reported in the Equation (1.65):

$$MAPO50 = g\ MAPO98 \quad (1.65)$$

Where  $g$  is the calibratable coefficient. In other words, the procedure varies the highest percentile (in that case the 98<sup>th</sup>, but it can be arbitrarily chosen) and the 50<sup>th</sup> percentile (directly deductible with Equation 1.65) to find the MAPO PDF for which the integral value of knock index values over the threshold (defined by damage model) coincide with the cyclic target damage (calculated multiplying the target damage speed for the cycle duration, that coincide with the execution time-step). Figure 1.92 displays the algorithm scheme.

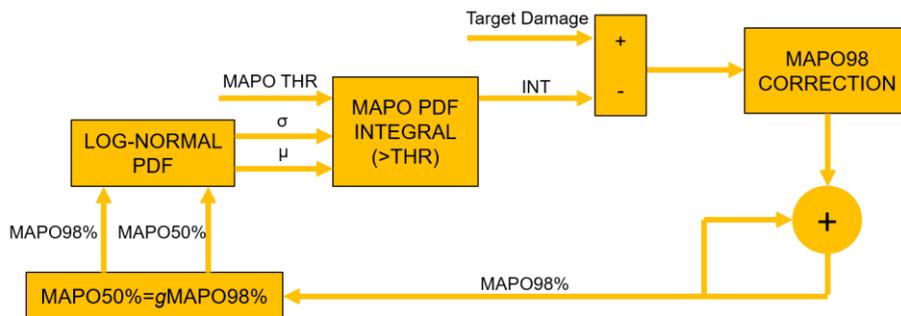


Figure 1.92: Block scheme of the algorithm used to find the MAPO PDF that guarantees the target cyclic piston damage.

However, the iterative procedure is not compatible with the RT execution of the complete controller because it needs multiple iterations that must be completed between two consecutive engine cycles. Thus, the procedure has been executed offline and a 3-D lookup table is calibrated. The target damage speed, the MAPO threshold and the engine speed are the inputs, while the target value of MAPO98% (98<sup>th</sup> percentile of MAPO index) is the output.

#### 1.4.1.4. Analytical Knock Model

The analytical knock model is used to convert the target value of MAPO98% into the corresponding mean Pmax. As explained in section 1.3.2.3, such model was initially developed and calibrated for a 4-cylinder GDI TC engine [1.79] and then it has been calibrated for the 8-cylinder engine used for this activity [1.81]. Equation (1.66) shows the analytical formulation of the model for a standard GDI TC engine:

$$MAPO98\% = (a STAM^b) \left( PMAX + \left( (\lambda - \lambda_{ref}) \lambda m \right) \right)^{(c STAM+d)} + RPM ESm \quad (1.66)$$

Where:

- *STAM* is the Stoichiometric Trapped Air Mass, which is equal to the trapped air mass (TAM) per cycle per cylinder when the mixture is stoichiometric or rich, and to the ratio between TAM and lambda when the mixture is lean
- *PMAX* is the maximum in-cylinder pressure
- *ESm* is the Engine Speed Multiplier
- *a*, *b*, *c*, and *d* are the four calibration parameters of the model
- $\lambda_{ref}$  represents the reference lambda value, with which model coefficients (*a*, *b*, *c*, and *d*) have been calibrated. In this case it is equal to 1
- $\lambda m$  is the lambda multiplier, the calibratable coefficient the converts the lambda numerical difference in a  $\Delta Pmax$

As observable in Equation (1.66), the effects of the mixture enrichment on knock intensity is modelled as an additive contribution to the Pmax value. The effect of the mixture composition can be seen as a mere translation of MAPO98% curve along the PMAX axis. In other words, for fixed engine speed and load, when the mixture is enriched the same knock intensity is achieved with a higher Pmax. Analytical knock model has been further improved, including the effects on knock intensity caused by a fuel quality (RON) variation, especially for 8-cylinder engine. Indeed, the same approach used to model the influence on statistical knock index of lambda is applied to fuel RON changes. Indeed, as described in previous section and as shown in Figure 1.93, a fuel RON increase acts like a mere translation of MAPO98% curves along Pmax axis, showing the same qualitative trend seen for mixture enrichment. Equation (1.66) is then updated, and it evolves in Equation (1.67):

$$MAPO98\% = (a STAM^b) \left( PMAX + \left( (\lambda - \lambda_{ref}) \lambda m \right) + \left( RONm(RON_{ref} - RON_{fuel}) \right) \right)^{(c STAM+d)} + RPM ESm \quad (1.67)$$

Where:

- *RONref* is the reference fuel RON value, used during tests that generates the database with which the parameters *a*, *b*, *c* and *d* are identified

- $RON_{fuel}$  is the RON number of the current fuel
- $RON_m$  is the fuel RON multiplier, the calibratable coefficient the converts the RON numerical difference in a  $\Delta P_{MAX}$ , as well as the  $\lambda m$ .

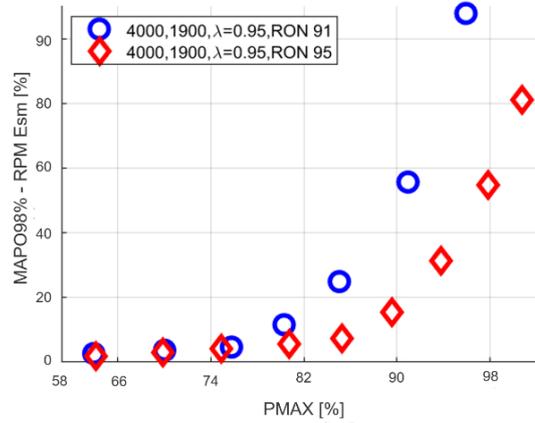


Figure 1.93: Normalized and offset MAPO98% curves for fixed operating conditions and for two fuel RON values.

The influence on engine knock tendency of air temperature in the intake manifold is also evaluated. As highlighted in the section 1.3.2, the intake air temperature does not produce a univocal effect on knock intensity, especially on tested 8-cylinder engine. Indeed, this experimental observation allows to confirm that the effect of the intake air temperature is negligible, and its contribution can be excluded from the knock intensity calculation for this specific engine. Innovative technologies, such as WI or Low-Pressure EGR are not available on this engine but the effects produced by this kind of systems (that introduces inert substances within the combustion chamber) can be easily counted by the model. Nevertheless, it is important to highlight that the implementation of this kind of contribution is needed only if such technology is actively applied to manage knock intensity or exhaust gas temperature and the controlled quantity (injected water mass or recirculated exhaust gas mass) changes w.r.t. the calibration map. On the contrary, the resulting effect would be already included by base model coefficients ( $a$ ,  $b$ ,  $c$  and  $d$ ).

Model equation is then reversed to convert a target MAPO98% into a Pmax value, as reported in Figure 1.89. Analytical formulation of the reversed model is shown by Equation 1.68:

$$P_{MAX_{tgt}} = \sqrt{\frac{(MAPO98\%_{tgt} - RPM ESm)}{(a STAM^b)}} + (\lambda_{ref} - \lambda)\lambda m + RON_m(RO_{N_{fuel}} - RO_{N_{ref}}) \quad (1.68)$$

In which  $MAPO98\%_{tgt}$  is the target value of MAPO98% determined converting the target piston damage and the MAPO threshold into a MAPO PDF and  $P_{MAX_{tgt}}$  is the corresponding PMAX value, for fixed operating conditions. The algebraic step that allows to achieve the desired model very quickly without requiring a new calibration process of model parameters ( $a$ ,  $b$ ,  $c$ ,  $d$ ,  $ESm$  and so on) is a demonstration of the possibilities that derive from the analytical approach proposed by the authors. The easy way with which the fuel RON self-learning algorithm can be implemented is a further confirmation of the methodology benefits. Indeed, when the main independent variables of the knock model are stored, the  $RO_{N_{fuel}}$  coefficient can be directly calculated by solving an equation similar to Equation 1.68.

The engine operating field is described through a speed-load grid and all the points are numbered. In this way, variables needed to univocally identify an engine operating condition are halved, due to the possibility to indicate the corresponding point number. It is important to mention that the fuel RON learning algorithm is not applied to all the engine points. Indeed, data are stored only for operating conditions characterized by highest load values (the threshold is defined in terms of Pmax), avoiding the risk to determine fuel RON by including the combustion noise. The self-learning algorithm can be summarized in the following steps:

- MAPO index offset by engine speed contribution ( $RPM * ES_m$ ) is stored in a 2-D matrix in which the row represents the enumerated engine point and contains the elements of a buffer that is updated every cycle, following a First-In-First-Out logic. For each matrix row (i.e. engine point) is then possible to calculate the MAPO 98<sup>th</sup> percentile net of engine speed effect. Thus,  $MAPO98\%_{net}$  indicates the vector of percentile values
- Pmax index is offset by the lambda contribution ( $(\lambda_{ref} - \lambda)\lambda_m$ ) and stored in a 1-D column vector, in which each row represents the engine point. In this case the mean value is considered, and each vector element is updated by computing the weighted average between the stored value and the new one and giving a higher weight to the old number (such as a first order low-pass filter). After each cycle, the vector is updated and it contains the mean Pmax value for each engine point net of lambda contribution, indicated as  $PMAX_{net}$
- STAM values are stored in a column vector by applying the same method used for Pmax to solve the equations with the actual values of engine load for which knock intensity is recorded and stored. The fuel  $RON_{fuel}$  parameter can be then calculated with Equation 1.69:

$$RON_{fuel} = \frac{\overline{PMAX}_{net} - \frac{(c \overline{STAM} + d) \sqrt{(\overline{MAPO98\%}_{net})}}{(a \overline{STAM}^b)}}{RON_m} + RON_{ref} \quad (1.69)$$

$\overline{PMAX}_{net}$ ,  $\overline{MAPO98\%}_{net}$  and  $\overline{STAM}$  are upper-marked to highlight that are vector quantities and the calculated fuel RON is the mean value which results from the solution of the equations system. Moreover, the mean fuel RON value is determined through a weighted average of the available engine points. The vector of history of touched engine points is used to define the corresponding weights vector: the current engine point has the highest weight, while the oldest operating condition is associated to the lowest value. Such algorithm updates the learned fuel RON only when the engine is operated on the points characterized by the highest load. Otherwise, the last value is fixed. The frequency with which the fuel RON learning algorithm is executed can be decided by the engine manufacturer:  $RON_{fuel}$  parameter can be updated every cycle, every  $n$  cycles, once per vehicle mission and so on.

The fuel RON learning algorithm is implemented to calculate the up to date  $RON_{fuel}$  coefficient that is then sent to the reversed knock model, which converts the target MAPO98% into the corresponding mean PMAX value, as mentioned above. Figure 1.89 evolves in the following Figure 1.94.

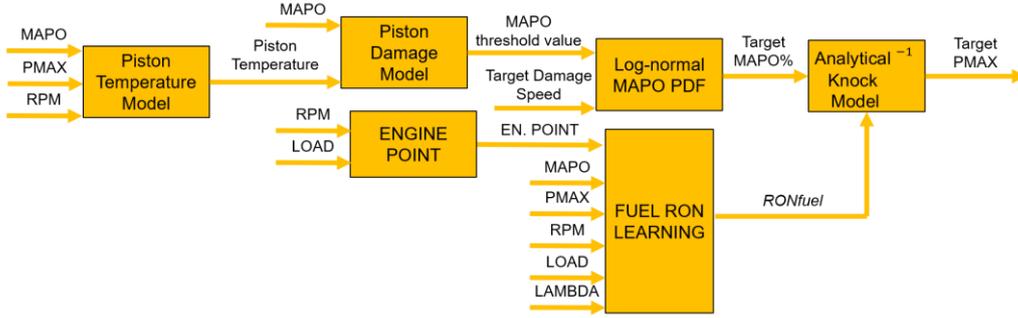


Figure 1.94: Controller layout, including the self-learning algorithm of the fuel RON value.

Another analytical model is then reversed to convert the target Pmax value into the related combustion phase, expressed as MFB50 index.

#### 1.4.1.5. Analytical PMAX model

The proposed Pmax model has been described and validated in the previous chapter of this work. The reversed analytical formulation of such model can be easily implemented in the combustion controller to translate a Pmax value into the corresponding MFB50. Equation (1.70) represents the standard model, and it is a second order polynomial, w.r.t. the MFB50 variable. This means that two possible solutions exist, but only the positive one allows to calculate the correct result. Thus, Equation (1.71) show the proper formulation of the reversed Pmax model.

$$P_{MAX} = p_{00P} + p_{10P}MFB50 + p_{01P}STAM + p_{20P}MFB50^2 + p_{11P}MFB50 STAM \quad (1.70)$$

$$MFB50 = \frac{-(p_{11P}STAM + p_{10P}) + \sqrt{(p_{11P}STAM + p_{10P})^2 - 4 p_{20P}(P_{MAX} + p_{00P} + p_{01P}STAM)}}{2 p_{20P}} \quad (1.71)$$

The implementation of the reversed Pmax model in the open loop chain of combustion controller allows to determine the target combustion phase, that is the value that guarantees the target damage speed. Figure 1.95 displays the controller layout at this point of the system development. The dotted line (Adapt. (RONfuel)) represents the algorithm that implements the fuel RON learning strategy.

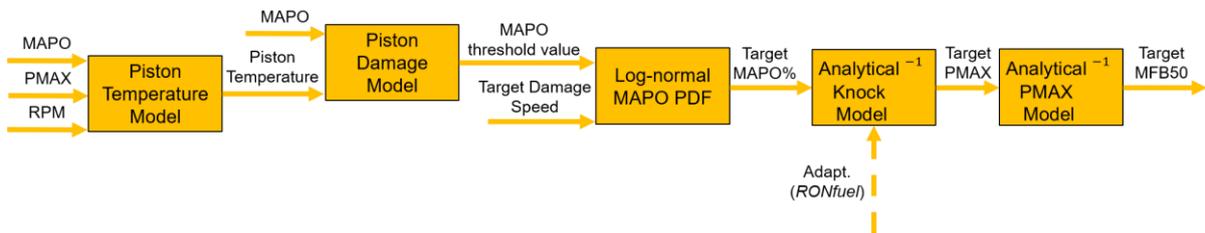


Figure 1.95: Controller layout, including the reversed PMAX model.

#### 1.4.1.6. Analytical MFB50 model

The final output of the open-loop chain is the  $\Delta SA$  (w.r.t. the calibrated value) that allows to achieve the target MFB50. The MFB50 model analytically defines the relationship between actuated spark timing and the combustion phase. Figure 1.96 represents the block scheme of the controller open-loop chain, including the MFB50 model.

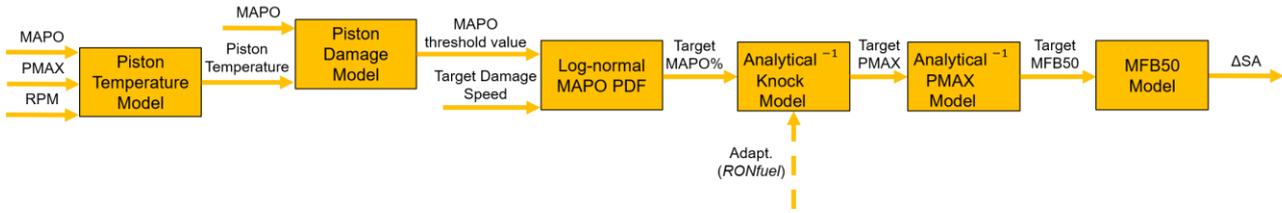


Figure 1.96: Controller layout, including the MFB50 model.

Such model is modified to make it compatible with the adaptive strategy that allows to update the parabolic relationship between the MFB50 and the corresponding spark timing, for fixed engine speed and load. For each engine point (fixed speed and load), the relationship between mean MFB50 and actuated SA is characterized by a parabolic trend, which can be described with a second order polynomial. Such trend derives from a variable MFB50 sensitivity to the SA: the higher the MFB50, the lower the curve slope. Figure 1.97 shows an example of what has been asserted, collecting data for a spark sweep recorded for given engine speed and load. SA is reported as a difference from the reference value and MFB50 is normalized w.r.t. the maximum value.

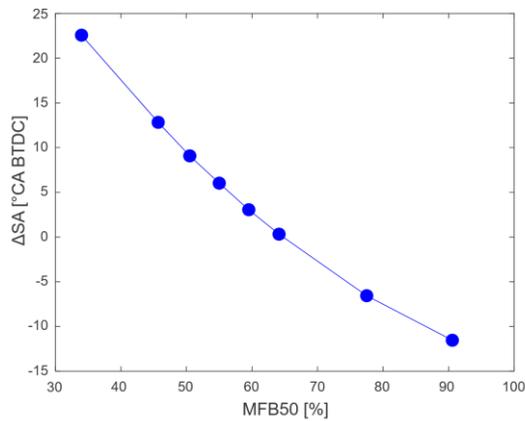


Figure 1.97: Experimental  $\Delta SA$ -MFB50 points recorded for fixed engine speed and load. The MFB50 is normalized w.r.t. the maximum value.

Thus, for each engine point (defined by the couple speed and load) a second order polynomial can be calculated. In the model introduced in the analytical engine simulator described in the section 1.3.3.6, parameters  $A$ ,  $B$  and  $C$  of Equation (1.52) are defined in the speed – load domain through a polynomial equation. For the model implementation in the combustion controller such further fitting step is not needed. Equation 1.72 defines the kind of equation used to fit  $\Delta SA$  on the MFB50 axis, for each engine point:

$$MFB50 = A \Delta SA^2 + B \Delta SA + C \quad (1.72)$$

Where  $A$ ,  $B$  and  $C$  are the equation parameters.

Mean MFB50 values are stored in a 2-D matrix, where rows represent the engine point and the columns the  $\Delta SA$  steps, equally distributed between positive and negative values. A unitary spark advance step corresponds to a certain number of crank angles (CA) degrees. Limitations on  $\Delta SA$  are typically imposed by the maximum and minimum spark timing that can be actuated by ECU. For instance, if the controller can communicate a  $\Delta SA$  step within the range equal to  $\pm 20$  (that corresponds to a certain range of CA degrees), 2-D matrix has 41 columns. Each value in the matrix (corresponding to a given engine point and applied  $\Delta SA$  step) is updated by computing the

weighted average between the stored value and the new one that works as a moving average, as for the Pmax vector in the fuel RON learning algorithm. Initial values of 2-D matrix are defined via Software-in-the-Loop (SiL) using the analytical engine simulator proposed in the previous chapter. For every execution time-step the stored MFB50 value for current operating condition is updated with the more recent recorded MFB50, though the moving average operator. For each matrix row a parabolic equation (such as that defined by Equation 1.72) can be determined by fitting stored values w.r.t. the fixed  $\Delta SA$  vector. The identification process of optimal values for parameter  $A$ ,  $B$  and  $C$  for the current engine point can be performed by applying the least squares method, for the case in which the equation degree is equal to 2 [1.94]. Considering the example described above, in which the admissible  $\Delta SA$  steps go from -20 to +20 and the 2-D matrix is then characterized by 41 columns, the associated system of  $m$  (in this case 41) equations of degree  $n$  (with  $m > n$ ) can be defined, and the least squared methods allows to directly determine the vector of  $n+1$  elements of coefficients that guarantees the optimal approximation of the available data. Equation 1.73 shows the matrix formulation of the associated squared system:

$$V^T V a = V^T y \quad (1.73)$$

Where  $V$  is the Vandermonde rectangular matrix ( $m+1 \times n+1$ ) that contains values  $V_{ij} = x_{i-1}^{j-1}$  of independent variable,  $a$  is the vector with  $n+1$  values of the optimal polynomial coefficients while  $y$  is the vector with  $m+1$  values of dependent variable. For this specific application, the Vandermonde matrix ( $42 \times 3$ ), vectors  $a$  ( $3 \times 1$ ) and  $y$  ( $42 \times 1$ ) are defined by Equation (1.74), (1.75) and (1.76), respectively:

$$V = \begin{pmatrix} 1 & \dots & 1 \\ x_0 & \dots & x_{41} \\ x_0^2 & \dots & x_{41}^2 \end{pmatrix} \quad (1.74)$$

$$a = \begin{pmatrix} C \\ B \\ A \end{pmatrix} \quad (1.75)$$

$$y = \begin{pmatrix} y_0 \\ \vdots \\ y_{41} \end{pmatrix} \quad (1.76)$$

If  $S = V^T V$  and  $c = V^T y$ , the vector  $a$  can be determined by solving the following squared ( $3 \times 3$ ) system:

$$a = S^{-1} c \quad (1.77)$$

The least squared method is implemented in the combustion controller directly defining matrices  $S$  and  $c$ . In this way some complex calculations are avoided, and the resulting procedure is compatible with RT execution. Moreover, this choice allows to carry out the matrix multiplication ( $V^T V$ ) by applying a weight vector to assign the highest importance to equations related to the  $\Delta SA$  close to the value that has to be actuated. In this way the found coefficients  $A$ ,  $B$  and  $C$  for the specific engine point (MFB50 matrix row) are calculated guaranteeing a higher accuracy within the range of commonly actuated spark timing corrections. For every engine cycle the matrix in which the filtered MFB50 are stored is updated including the logged MFB50 for current engine operating conditions

(speed, load and  $\Delta SA$ ) and the parabolic equation is directly calculated to determine with high accuracy the  $\Delta SA$  needed to achieve the target combustion phase value. Thus, the controller layout represented in Figure 1.98 includes data that the adaptive algorithm for the identification of polynomial equation coefficients needs as input.

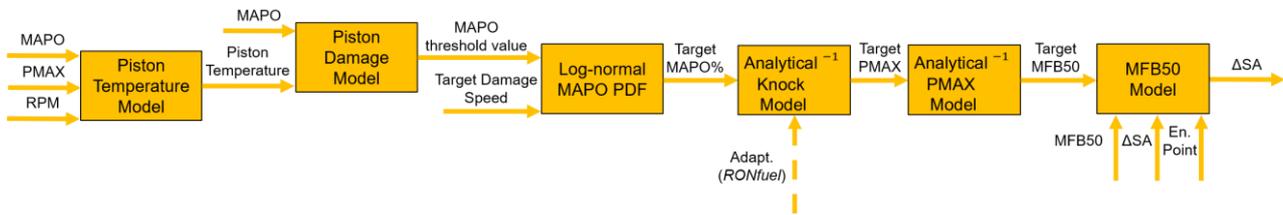


Figure 1.98: Controller layout, including the adaptive algorithm for the  $\Delta SA$ -MFB50 analytical equation updating.

It is important to highlight that vector and matrix of combustion index and the engine load (*STAM*) are updated according to the discretization of the speed-load engine operating field. On one hand, if the breakpoints of such grid are sufficiently dense, the algorithm can assign to a certain operating condition the closest engine point, rounding the speed and load w.r.t. the breakpoints resolution. On the other hand, when the breakpoints of the engine speed-load grid are not so dense, the weighted update of the four breakpoints between which the generic operating condition falls is implemented. In other words, indexes of the four breakpoints (two engine speed and two engine loads) that include the current operating conditions are updated distributing the logged index on each of such breakpoints, by defining a weight that is inversely proportional to the distance between current operating condition and each discretized engine point. Both strategies have been tested during the development process of described combustion controller, but the first solution (based on the updating of the closest breakpoint value, w.r.t. the current condition) has been finally adopted. Indeed, a sufficiently dense discretization of the engine speed-load operating field has been reached, due to a numerical optimization process of stored data for adaptive strategies that allowed to minimize the impact of matrices and vectors on the volatile memory of the RCP device. Such solution has been preferred for the lower computational power needed, due to the simpler algorithm and because the performance of both strategies was practically the same.

#### 1.4.1.7. Complete Combustion Controller

Knock intensity and combustion phase are indexes that can be estimated by using common systems for in-cylinder pressure trace sensing compatible with on-board applications, as described in the introductory part of this section. Instead, the controller scheme shown in Figure 1.98 also needs the cyclic maximum in-cylinder pressure as input of the piston temperature model. Thus, the control system is further developed, introducing the standard PMAx model defined by Equation 1.70, to convert the MFB50 index into PMAx. Moreover, a closed-loop chain based on a PI controller is implemented only for a protection purpose. Indeed, the PI controller moves only when the instantaneous damage speed exceeds a safety threshold that corresponds to a very high knock intensity level. This because the aim is that to control combustion phase entirely with the model-based open loop chain and leave that the PI controller only retards the spark timing. The closed loop chain has the capability to calculate only negative  $\Delta SA$ . Figure 1.99 shows the complete combustion controller scheme that is then implemented in a RT machine and validated at the test bench via RCP. Text in red highlights the inputs that come from the on-board in-cylinder pressure sensing system.

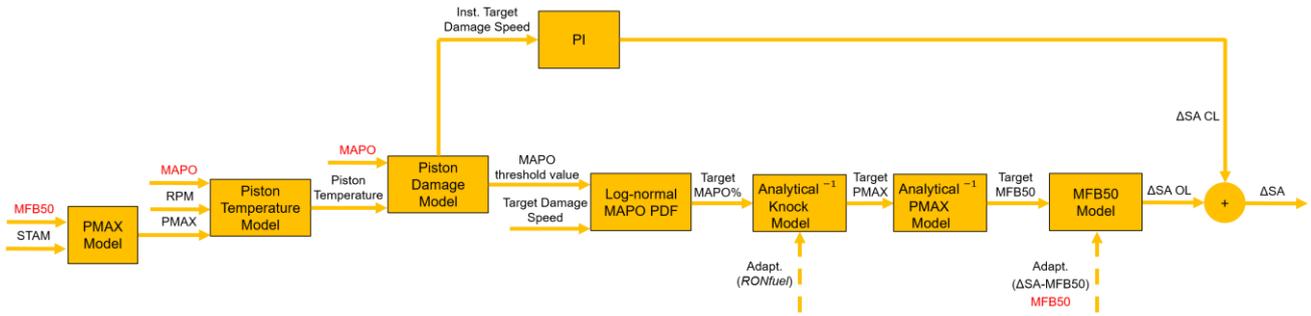


Figure 1.99: Complete controller scheme. Text in red indicates the combustion indexes estimated from the system for the on-board in-cylinder pressure trace sensing, i.e. MAPO and MFB50.

### 1.4.1.8. Combustion Controller Validation

The resulting combustion controller is finally implemented in an RCP system that communicates the calculated  $\Delta SA$  to the ECU via CAN. The indicating system (OBI-M2) is connected to the same network and provides MFB50 and MAPO indexes to the controller. Combustion indexes are published once per engine cycle, while the combustion controller code is executed at 1 kHz to have an execution frequency that is higher than the engine cycle one, for the entire engine operating field. The complete controller is implemented for each cylinder, except for the fuel RON learning algorithm: matrix of MAPO and PMAX vector are specialized for each cylinder, but the calculated fuel RON is defined by averaging the specific values (one for each cylinder), and it is valid for all the cylinders. While the open loop chain uses the current engine operating parameters (speed, load, lambda), the actuated  $\Delta SA$ , the MFB50, and MAPO recorded during the current cycle to calculate the following  $\Delta SA$ , the adaptive algorithms and the closed loop chain receive the same data after the combustion. In other words, the open loop calculations are performed before the engine cycle for which the target damage speed is defined, ensuring the as fast as possible response. On the contrary, the adaptive algorithms and the PI controller are executed after the engine cycle for which the  $\Delta SA$  is determined.

Engine at the test bench is operated with speed and load profiles recorded during vehicle maneuvers. The tested profiles consist of complete track laps and it includes very steep transients but also semi-steady state conditions. In this way controller is stressed with a realistic engine speed and load profiles and the results demonstrate the high reliability of the proposed tool. Moreover, some tests are carried out with an offset SA map, to validate the performance and the robustness of the adaptive algorithm that updates mean MFB50 values and  $\Delta SA$ -MFB50 functions. Even the fuel quality is changed during a portion of these tests, and the reliability of knock model is then demonstrated. Gear shifts have been removed from such profiles because the brake dynamics does not allow to reproduce that kind of transients. In the following section of the chapter, most significant results are thoroughly described.

### 1.4.1.9. Results and Discussion

Several hours of tests have been carried out with a single engine bank, due to the dynamometric brake limitations on the maximum power. Following figures show main results obtained during such tests. Figure 1.100 displays a portion of engine speed and load profiles tested during the validation procedure, normalized w.r.t. maximum values of the engine operating filed. Thus, the entire engine operating filed is covered with these tests.

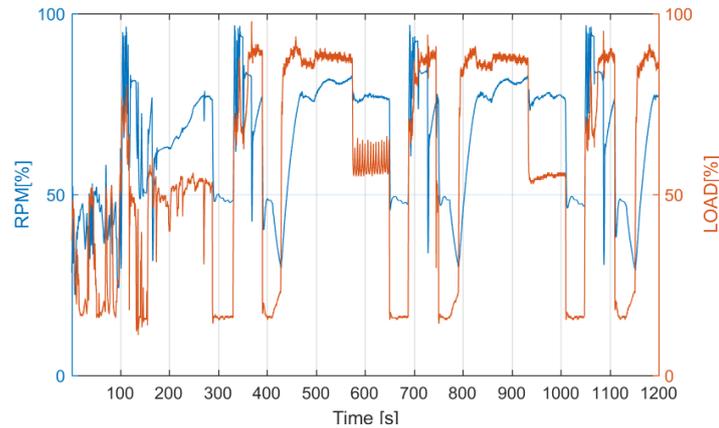


Figure 1.100: Normalized engine speed and load profiles related to results presented below.

Figure 1.101 reports MFB50, Pmax, final  $\Delta SA$  and the closed-loop contribution, MAPO and its 98<sup>th</sup> percentile and the damage speed normalized w.r.t. maximum values, for cylinder 1. For this test, the matrices and vectors included in the learning algorithms contain calibrated values. Logged MFB50 values are filtered by applying a moving average. At first it is important to mention that the filtered MFB50 (to exclude the cyclic variability) is always on target value, except when the PI controller reacts. This condition can be easily identified when the  $\Delta SA$  defined by the closed loop is lower than 0. The target MFB50 is defined as the value that determines a given damage speed value saturated at the maximum combustion efficiency value. This means that when the combustion phase value given by the conversion of the target damage speed is lower than the MBT value, the resulting target is equal to the angular value of maximum combustion efficiency. For this reason, Pmax and MAPO98 go on the corresponding target values only when the target MFB50 is higher than the MBT value. Indeed, during such portions of the test, Pmax and MAPO98 are very close to the target, and this allows to achieve the fixed damage speed target, as observable in the related graph. Results demonstrate that the model-based open loop chain allows to accurately convert the target damage speed into a corresponding combustion phase that realizes a certain surface piston erosion. The reliability of all described model is therefore confirmed. Even the adaptive MFB50 model is reliable and allows to directly calculate the  $\Delta SA$  needed to reach the target MFB50. As demonstrated in the third graph of the figure the PI controllers reacts only with small SA corrections when the instantaneous damage speed exceeds a certain threshold value. Moreover, the calculated cyclic Pmax is completely superposed to the experimental one and this result confirms the accuracy and the robustness of the proposed analytical model, as previously demonstrated in chapter 1.3. Thus, it is possible to state that the initial conversion of the MFB50 into the corresponding Pmax through the implementation of the analytical model is extremely accurate and it introduce negligible errors. This choice, as highlighted above, is particularly important because it allows to use a commercial in-cylinder pressure sensing system, for the estimation of the MFB50 and MAPO indexes. Figure 1.102 shows the percentage error between the filtered MFB50, PMAX and the corresponding target values for cylinder 1, for the analyzed test. On one hand, the error between the target and the filtered MFB50 has to be evaluated excluding engine cycles in which the PI controller reacts, because in that case the recorded index is given by the sum of both model and closed loop contributions. On the other hand, the proper manner to calculate the error between the target and the experimental PMAX is that to exclude all the engine cycles in which the target MFB50 is saturated to the MFB value, because the target value of Pmax cannot be achieved. For both indexes, error is mainly included within the range  $\pm 4\%$ . These values, together with the achievement of the damage speed target, is an extremely relevant result.

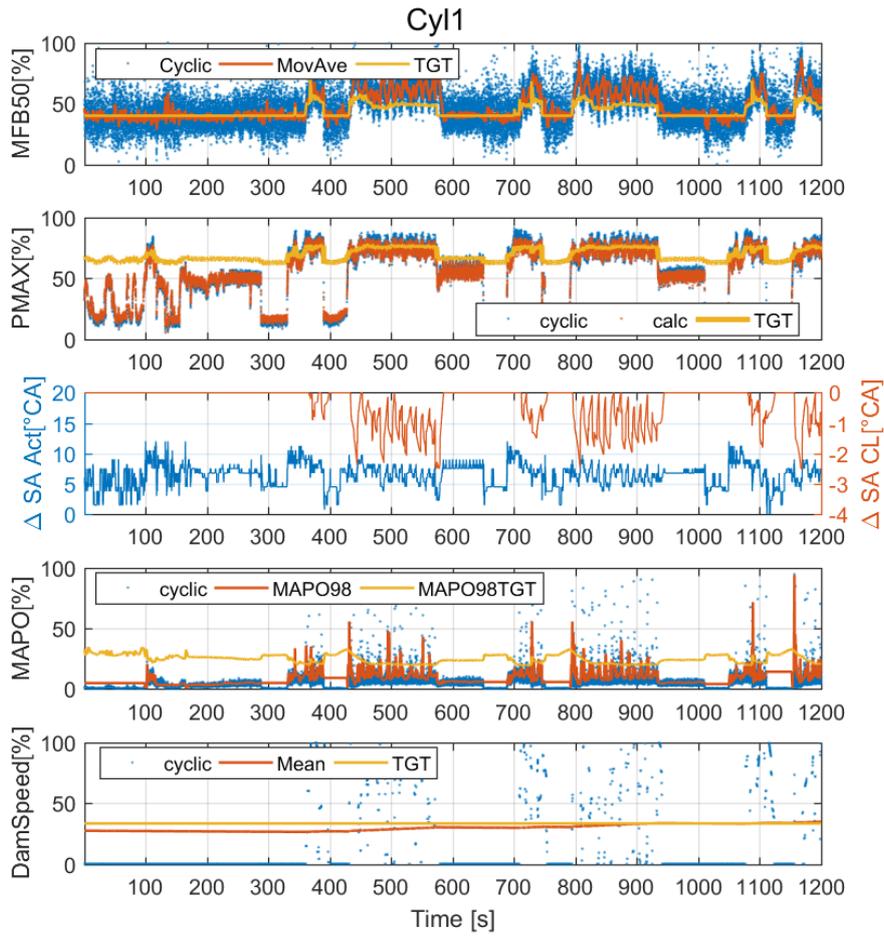


Figure 1.101: Target and experimental values of MFB50, PMAX, MAPO and damage speed for cylinder 1. Third graph represents the spark timing correction of the closed loop chain and the final output of the controller.

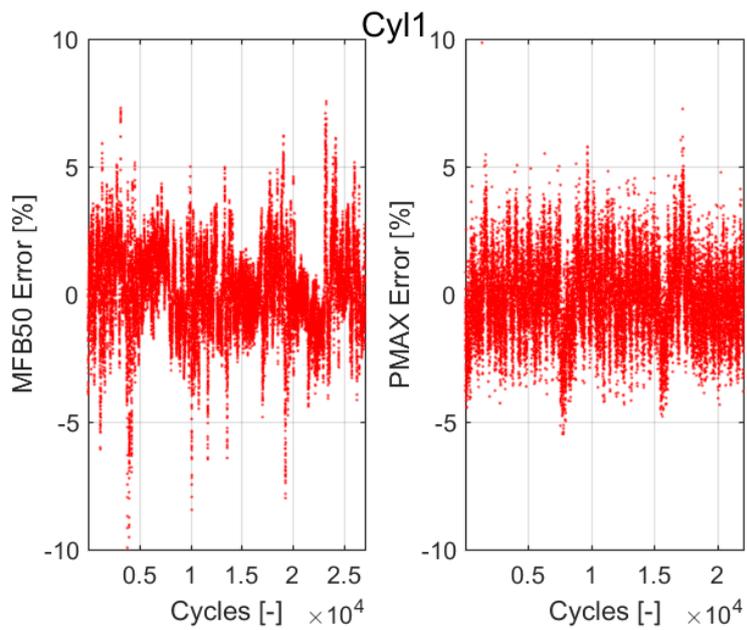


Figure 1.102: Error between the target and the experimental MFB50 and PMAX indexes, for cylinder 1.

Following figures show trends for other cylinders, for which the same results have been obtained, while Figure 1.105 shows calculated fuel RON value, that remains practically constant during the test. In this case a standard RON 95 fuel has been used. Nevertheless, fuel octane number is purely indicative, and the number 94.2 calculated by the reversed knock model is coherent with the nominal fuel quality. As discussed above, the fuel RON value is calculated only when the PMAX exceeds a given threshold, to ensure excluding engine point in which knock does not occur. For such reason, during the test fuel RON assumes constant values.

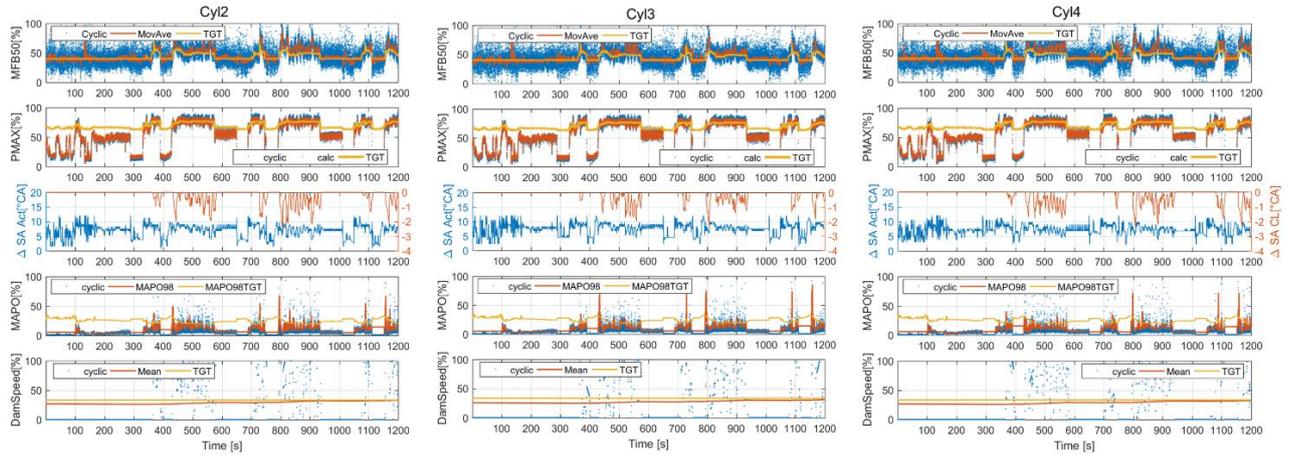


Figure 1.103: Results for cylinder 2, 3 and 4.

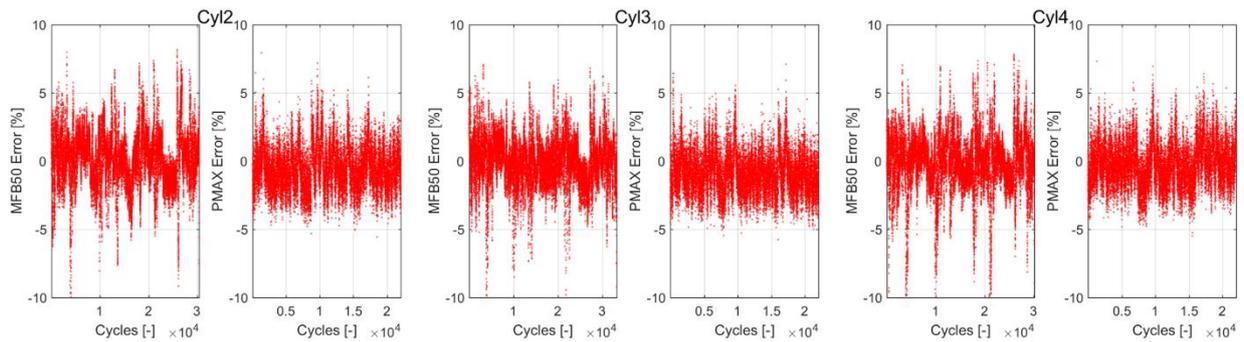


Figure 1.104: Error between the target and the experimental MFB50 and PMAX indexes, for cylinder 2, 3 and 4.

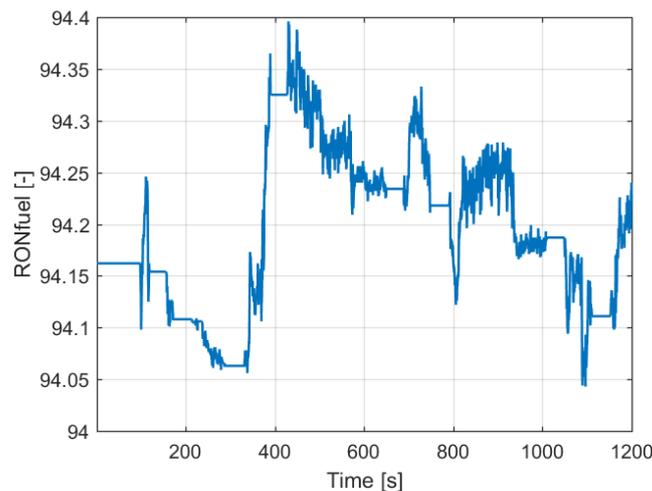


Figure 1.105: Calculated fuel RON.

Both the adaptive strategies are validated with specific tests. The algorithm for the  $\Delta SA$ -MFB50 relationship learning is stressed by applying an offset to the entire spark timing map. The parabolic function that defines the combustion phase w.r.t. the SA correction needs to be updated. The reliability of the algorithm implementation is then validated, and the time needed for the values update of the MFB50 matrix is evaluated. In this way the condition that leads to save incorrect MFB50 values (for instance due to a broken sensor) is reproduced. In this case test has been carried out by fixing engine speed and load and forcing a spark timing correction on the SA map. In this way it is possible to quantify the time spent by the adaptive MFB50 model to update the stored MFB50 values and calculate the correct parabolic function. Engine speed and load profiles are shown in Figure 1.106, while Figure 1.107 displays results in terms of MFB50 and  $\Delta SA$  for cylinder 1. Dashed line indicates the moment in which the offset is applied at the spark timing map. The adaptive strategy is able to recognize and calculate new  $\Delta SA$ -MFB50 parabolic function by updating values saved in the MFB50 matrix. Error returns close to 0 in 7 seconds and this kind of test demonstrates the robustness of such algorithm. It can be noted that the implementation of an eventual closed loop contribution to achieve the target value of combustion phase is not needed. Indeed, it would introduce an additional error, especially under transient operating conditions. Figure 1.108 reports the same results for other cylinders.

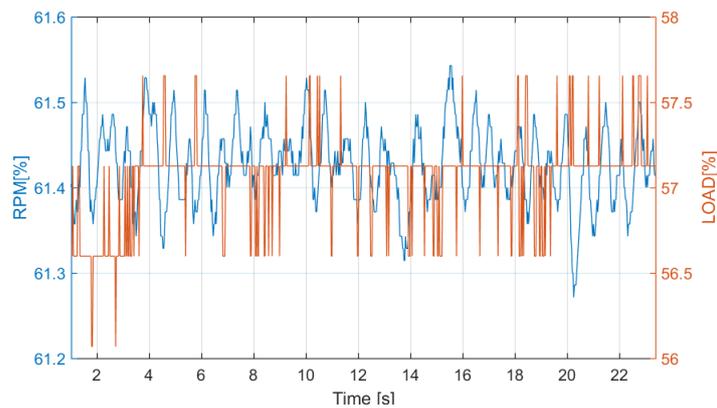


Figure 1.106: Normalized engine speed and load profiles for the validation test of MFB50 model adaptive algorithm.

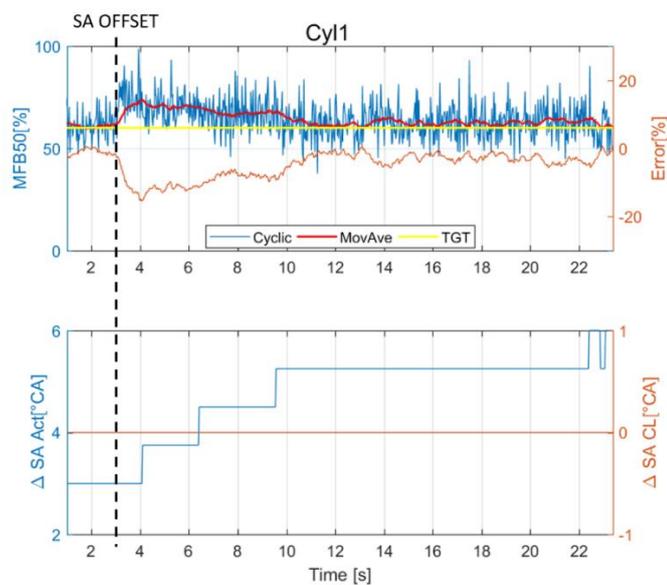


Figure 1.107: MFB50 and  $\Delta SA$  for cylinder 1. Dotted line indicates the moment in which an offset on the SA map is applied.

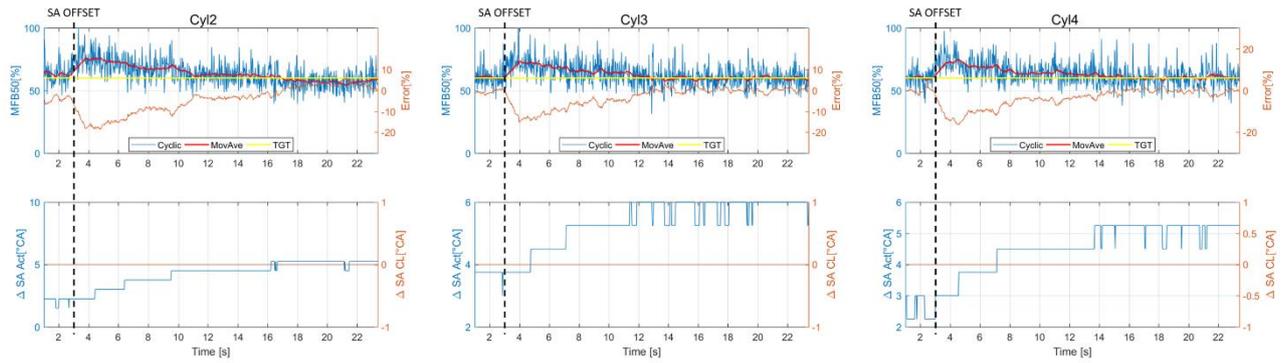


Figure 1.108: MFB50 and  $\Delta SA$  for cylinder 2, 3 and 4. Dotted line indicates the moment in which an offset on the SA map is applied.

Fuel RON learning algorithm is finally validated. A long test has been carried out reproducing at the bench engine speed and load profiles used for the first kind of test (Figure 1.100) and switching between two different types of fuel. This kind of validation has been performed with two tanks that supply the engine test bench fuel line. Every tank is filled with a different fuel quality: RON95 and RON100, respectively. Engine tank that supplies engine fuel line is changed during the test stressing fuel RON learning strategy. This allows therefore to further validate the implementation in the knock model of the fuel quality effects on engine knock tendency. Even in this case results demonstrate high reliability of proposed control strategy. Indeed, Figure 1.109 shows the trend of calculated fuel RON value during the test. Dashed lines indicate moments in which fuel quality is switched. As mentioned above, the  $RON_{fuel}$  coefficient updating is performed when P<sub>MAX</sub> exceeds a calibrated threshold and this is the reason why displayed value is constant for some portions of the test. Calculated RON clearly follows the nominal fuel octane number.

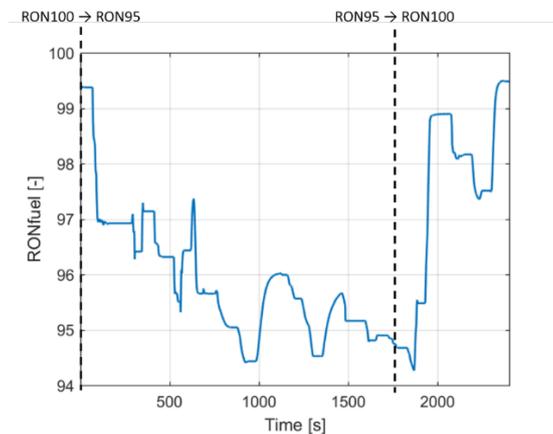


Figure 1.109: Calculated fuel RON for validation test of fuel RON adaptive strategy.

Figure 1.110 shows trend of main combustion indexes, spark timing corrections and damage speed for cylinder 1. The fuel RON learning algorithm allows to properly count the effects of fuel quality and calculate the maximum in-cylinder pressure that ensures the target MAPO98 value for the current kind of fuel, on the entire engine operating field. Average damage speed remains very close to the target, as well as the MFB50 and P<sub>max</sub>, when the closed loop contribution is null and the target combustion phase is not saturated to the MBT value, respectively. Figure 1.111 reports the percentage error between the target and the moving average of cyclic MFB50 and P<sub>MAX</sub>, while following figures show results for other cylinders.

Validation tests demonstrate the high accuracy of model-based open loop controller and the reliability of implemented adaptive strategies for real on-vehicle operating conditions. Indeed, complete combustion controller has been experimentally tested operating the engine with speed and load profiles recorded during on track vehicle maneuvers. In this way it faces the condition of the final on-board application. Moreover, self-learning algorithms for fuel RON value and MFB50 model have been validated with specific tests with which some difficult conditions have been reproduced at the engine bench, forcing fast variation of SA map, and switching instantaneously between two different types of fuel. During all these tests controller allows to properly convert the piston damage speed into the corresponding combustion phase and to achieve the target MFB50 with small errors.

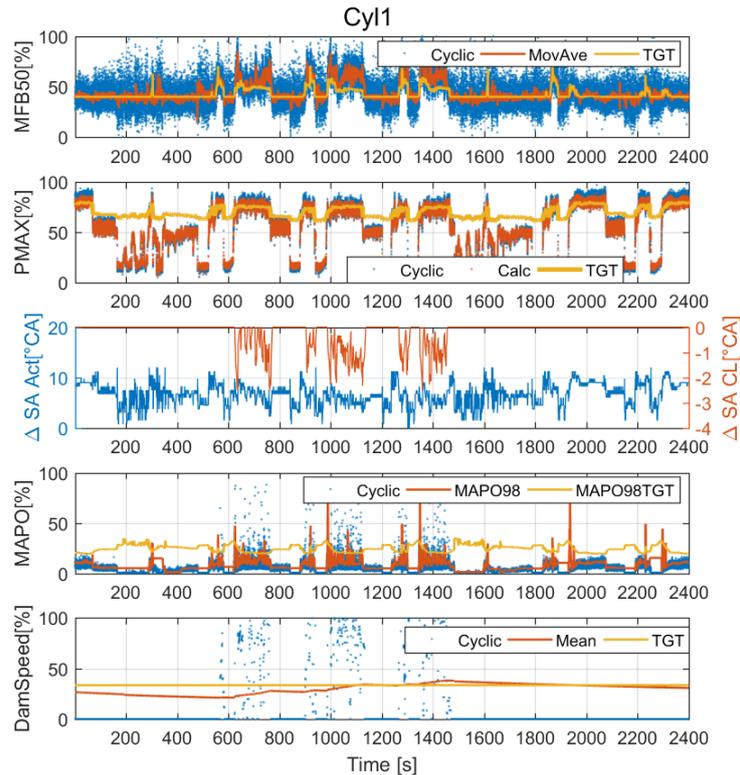


Figure 1.110: Target and experimental values of MFB50, PMAX, MAPO and damage speed for cylinder 1. Third graph represents the spark timing correction of the closed loop chain and the final output of the controller, for fuel RON adaptive strategy validation test.

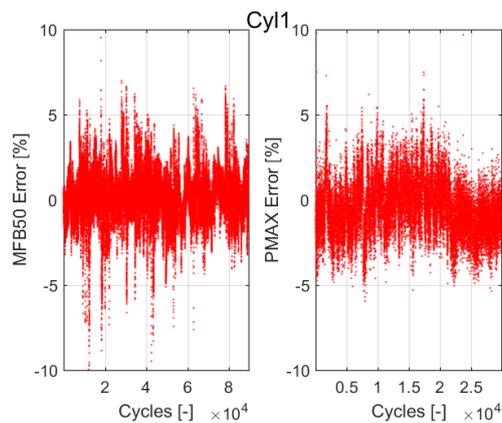


Figure 1.111: Error between the target and the experimental MFB50 and PMAX indexes recorded during fuel RON adaptive strategy validation test, for cylinder 1.

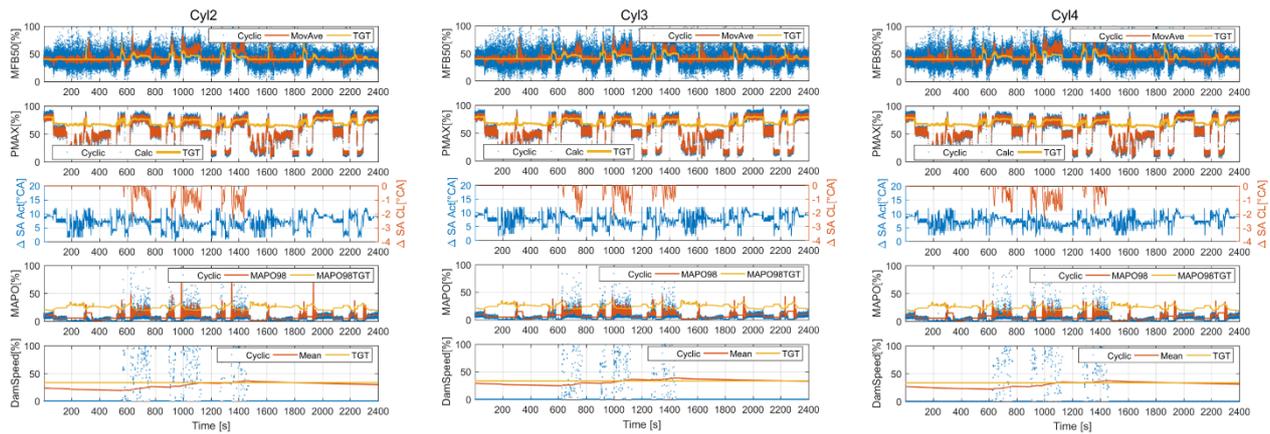


Figure 1.112: Results for cylinder 2, 3 and 4, for fuel RON adaptive strategy validation test.

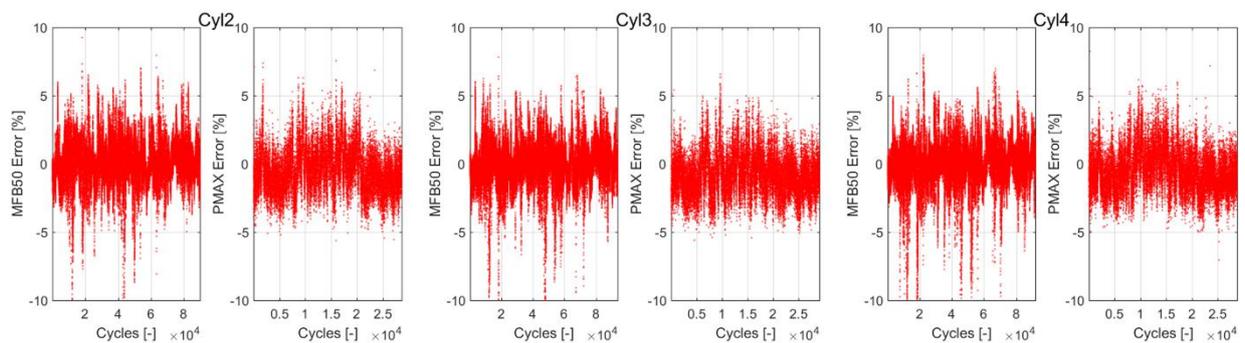


Figure 1.113: Error between the target and the experimental MFB50 and PMAX indexes, for cylinder 2, 3 and 4, for fuel RON adaptive strategy validation test.

#### 1.4.1.10. Conclusions and Proposals

A novel model-based combustion control system is described in this section. The proposed scheme completely redefines the standard knock control approach, providing a solution to efficiently integrate knock intensity management with the combustion phase control. This is possible by converting the target knock intensity value into the corresponding target MFB50 and this is realized through the implementation of a chain of models that describes the trend of main combustion indexes in a very accurate and simple way. Proposed system meets both unsolved issues of common knock control approach. Indeed, target knock intensity is defined as a piston erosion level, that represents the effective and tangible consequence of knocking combustion, and it is controlled through open loop chain that guarantees proper management performance even under fast transient conditions. Models are developed for a production engine. Indeed, the final controller is implemented in a RCP system that communicates the calculated SA correction via CAN to the ECU, that applies such correction to the mapped values. Proposed controller needs MAPO and MFB50 indexes, that can be obtained by standard on-board systems for the in-cylinder pressure signal estimation, such as accelerometers, ionization currents or piezoelectric components for engine head that can sense forces induced by combustion process. Nevertheless, tests shown in this section are carried out with an engine equipped with standard piezoelectric transducers and combustion indexes are calculated with a commercial indicating system.

Results of the validation process demonstrate both the high accuracy of complete open loop chain and the reliability of self-learning algorithms. Error between target and experimental indexes are

low and this further confirms that this novel approach to the combustion control can be considered as a viable and robust progress in this field.

Combustion controller will be further validated and coupled with an on-board compatible indicating system that provides MAPO and MFB50 indexes. Moreover, it will be finally implemented in a prototyping ECU for on-board tests.

### 1.4.2. *Innovative Solutions for The On-Board Knock Intensity and Combustion Phase Estimation*

Combustion controller described above needs knock intensity and combustion phase indexes as inputs. Such indexes can be estimated from an on-board compatible sensor that allows to robustly and accurately evaluate both features of combustion process. While knock intensity is commonly available on the final on-vehicle application (it is calculated from ION or accelerometric signal), estimation of MFB50 (or a well correlated phase index) is one of strategic topic of research activities in the automotive field.

As mentioned above, Alma Automotive has recently developed a piezoelectric spark plug washer and very promising results have been achieved [1.93]. However, during PhD course possibility of estimating MFB50 angle by accelerometric signal is investigated. This activity demonstrates the feasibility of obtaining a feedback on combustion phase using on-board available sensors. Through this solution, it is proven that there is a possibility of implementing the entire combustion controller proposed in the previous section of this chapter (as well as any other control system that needs knock intensity and combustion phase index) on a production scheme.

Results shown below have been already published in [1.95].

#### 1.4.2.1. **MFB50 Estimation by Accelerometric Signal**

The development of MFB50 estimation algorithm based on accelerometric signal has been developed on a different set-up. The engine used for this activity is an in-line 4 cylinder and its main features are summarized in Table 1.18. The engine is set up with two accelerometers as standard production equipment, one located between cylinders 1 and 2 (*Acc12*) and the other one located between cylinders 3 and 4 (*Acc34*). The accelerometric signal is acquired with a sampling frequency of 200 kHz.

*Table 1.8: Engine Features.*

Displaced volume	1995
Stroke	90
Bore	84
Connecting Rod	163
Compression ratio	10
Number of Valves	16
Exhaust Valve Open	10 Before BDC
Exhaust Valve Close	15 After TDC
Inlet Valve Open	50 Before TDC
Inlet Valve Close	75 After BDC

### 1.4.2.2. Signal analysis

The main inputs for the closed loop section of the Combustion Controller are MAPO and MFB50, being both measurements taken from the processing of the in-cylinder pressure signal. With an on-board application as a goal, the replacement of the in-cylinder pressure sensor with an on-board available alternative is necessary. Accelerometers are widely used to detect knock via a band-pass filtering of the signal. This opens the opportunity to evaluate the possibility of using this signal, processed differently, for combustion phasing detection as well. It is already known that there is a correlation between the peak angular position of in-cylinder pressure derivative and MFB50, reason why the first step of the analysis is to determine if the accelerometer can generate a signal that is able to identify this peak.

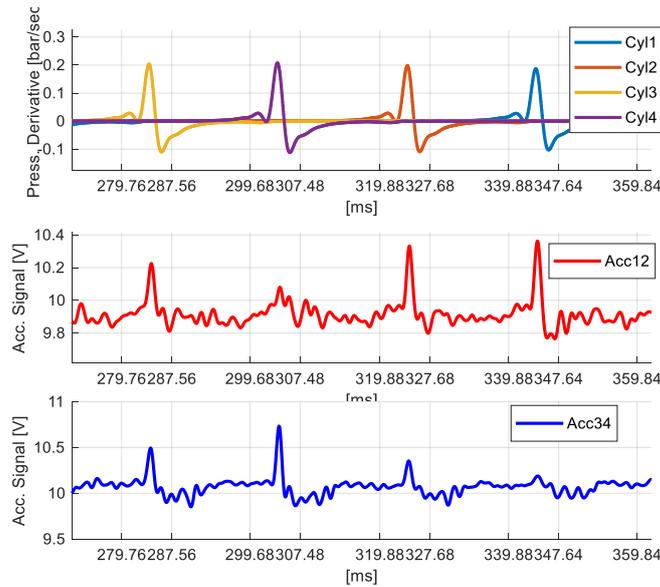


Figure 1.114: On top, derivative of in-cylinder pressure signal. In the middle and in the bottom, filtered accelerometric signal from the accelerometers (low-pass 1 kHz), 1500 RPM and 1.2 bar of intake manifold pressure.

A vast number of experimental tests at different engine points are carried out, ranging from 1500 RPM to 5500 RPM and values of load that are between 0.4 and 1.4 bar of intake manifold pressure. During these tests, both in-cylinder pressure sensors and accelerometers installed are registered.

A first processing of the accelerometric signal was carried out to understand the placement of the main frequencies related to combustion phasing. Contrary to knock, characterized by frequencies above 5 kHz, components of the accelerometric signal that can be used to identify combustion phasing are found in a frequency domain that can be as much as five times lower. A first analysis is made, using a 4<sup>th</sup> order Butterworth low pass filter with a cut-off frequency of 1 kHz. Figure 1.114 shows the results of this processing, from which it is immediately observed that near the peak of derivative of in-cylinder pressure related to a combustion event, there is a corresponding accelerometric signal peak as well. This implies that there is the possibility to directly correlate the position of the accelerometer peak and MFB50. As shown in Figure 1.114, the peak is more relevant for the cylinder located near to the sensor analyzed. To ensure a more precise peak detection, its identification is carried inside an angular window in which only combustion occurs, excluding all the rest of the cycle phases.

From this preliminary analysis, two observations can be made. The first, there is a time (and thus angular) delay between the peak of the pressure signal and the one from the accelerometric signal.

The physical explanation relies on the fact that the vibrations from the instant of maximum in-cylinder pressure must mechanically travel through the engine block before arriving to the accelerometer sensor. This process takes time and generates a delay between the physical event in the combustion chamber and its detection by the accelerometer. This delay, called angular delay, is not constant for all engine points and needs to be adequately identified. The second observation is that the cut-off frequency used on the low pass filtering process of the accelerometric signal impacts on the correlation between its peak and the location of MFB50. This implies that there is the need to generate a map with the cut-off frequency that maximizes correlation at each engine point. Angular delay can be calculated as:

$$Angular\ delay = \sum_{i=1}^N \frac{(Acc.Peak - MFB50meas)_i}{N} \quad (1.78)$$

Where *Acc.Peak* is the position of the maximum value of the filtered accelerometric signal inside the angular window and MFB50 measured with in-cylinder pressure sensor, both registered in same cycle. Therefore, the estimation of MFB50 by accelerometric signal (for each cycle) can be defined by Equation 1.79.

$$MFB50est = AccPeak - Angular\ delay \quad (1.79)$$

As shown in Figure 1.115, a correlation of over 90% can be obtained between accelerometric signal peak and measured MFB50 from indicating system. All the forthcoming results are obtained using the accelerometer positioned between cylinder 1 and 2 to observe those two cylinders, and in the same way the other sensor has been used for cylinder 3 and 4. General trends show that at mid and high loads, correlation remains above 90% while at low loads (load < 0.6 bar of intake manifold pressure), the accelerometric signal is not able to generate a distinctive peak due to a lack of combustion intensity, resulting in a correlation under 50%.

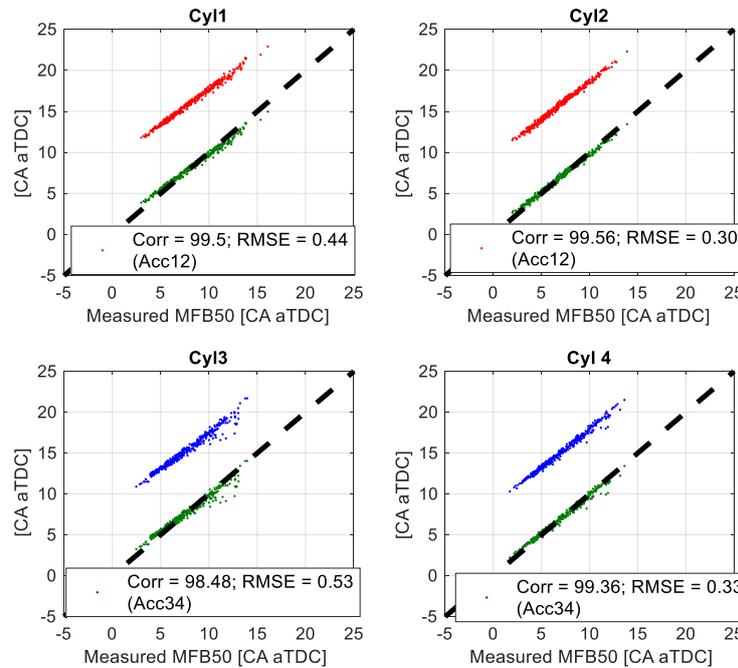


Figure 1.115: Correlation between accelerometric signal peak and MFB50 measured with in-cylinder pressure sensor for the engine point at 2500 RPM and 1.1 bar of intake manifold pressure. Red and blue dots represent the correlation obtained without angular compensation (red=Acc12; blue Acc34). Green dots represent the correlation with angular delay compensation (MFB50 estimated).

### 1.4.2.3. Signal processing algorithm and calibration

Once established that it is possible to estimate combustion phasing with accelerometric signal, an algorithm is designed to obtain the best results at each engine point. The strategy consists in signal windowing to focus the combustion phase, a low-pass filtering, peak recognition, and angular delay compensation (Figure 1.116). The algorithm has been calibrated by three steps:

- windowing of the accelerometric signal to be processed
- selection of the cut-off frequency for signal filtering
- angular delay identification.

Signal windowing has been imposed from 5 °CA until 70 °CA after spark event, interval in which accelerometric signal peak is generally located. Regarding the other two factors, optimum cut-off frequency and angular delay are identified and mapped for each of the tested engine points, process that is explained in the following paragraphs.

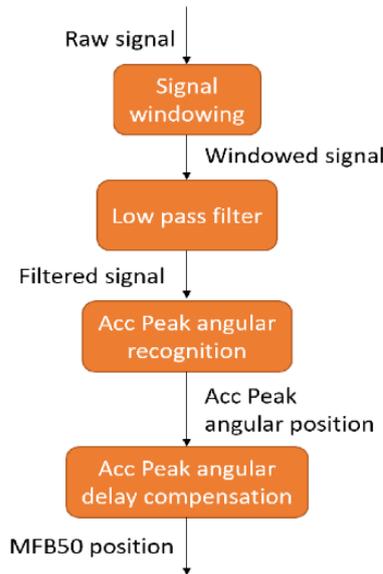


Figure 1.116: Block diagram of accelerometric signal processing algorithm.

To evaluate the performance of the algorithm, the Root Mean Square Error (RMSE) is calculated for each engine point in order to determine the angular error between the estimation of MFB50 by accelerometric signal and that calculated by the indicating system. These are calculated as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (MFB50_{est} - MFB50_{meas})_i^2}{N_{Cycles}}} \quad (1.81)$$

As a first procedure, a sweep of cut-off frequency is made at each engine point using a Design of Experiment (DOE), in order to find the frequency that minimizes the calibration RMSE. Figure 1.117 displays the results from the DOE for several cases of load at 2500 RPM as an example.

This optimal cut-off frequency is not equal for all four cylinders and depends on both engine speed and load (mainly the speed), and ranges from 400 Hz to 1500 Hz. After the DOE, a cut-off frequency as function of RPM has been selected, shown in Table 1.19. The dependence with load has been neglected in order to make the maps as smooth as possible.

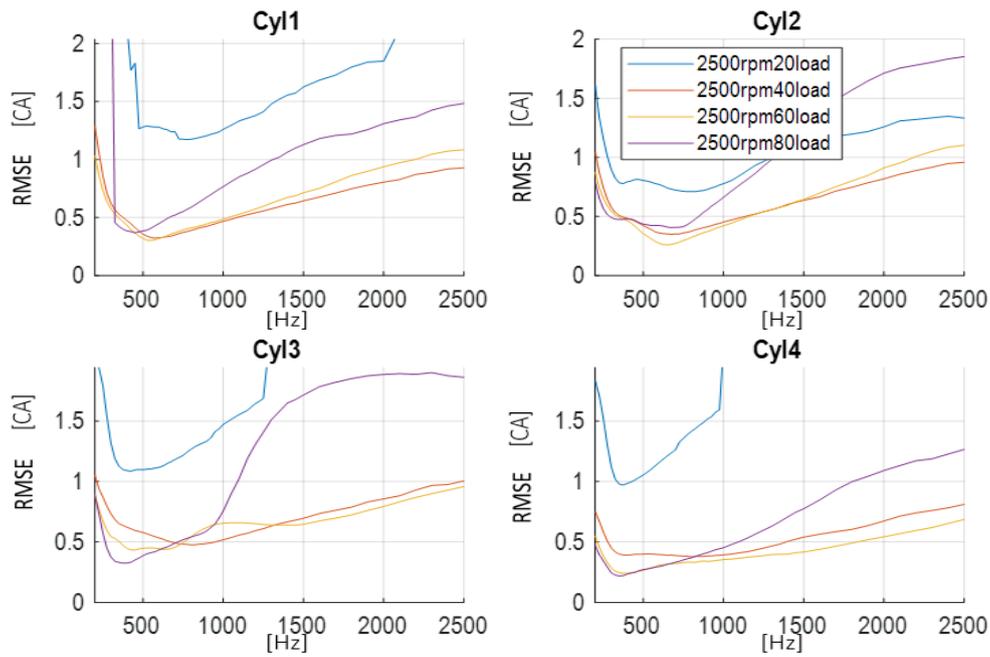


Figure 1.117: RMSE on the Cut-off frequency for engine point at 2500 RPM and 1.4 bar of intake manifold pressure.

Table 1.19: Cut-off frequency map [Hz].

Load[%]\RPM	1500	2500	3500	4500	5500
80		600	700	800	
60	500	600	700	800	900
40	500	600	700	800	900
20	500	600	700	800	900

The cut-off frequency for each engine point in the grid is identified and mapped, the angular delay associated to each cut-off frequency is mapped as well. The final scatter of points is displayed by the green points of Figure 1.115, which are shifted towards the black dashed line corresponding to the real MFB50 angle.

The tables below contain the results of the algorithm applied to each tested engine point, in just one cylinder as example. Correlation coefficients seen in Table 1.20 are above 95% in all cases except at 4400 RPM and 1.6 bar, point in which the accelerometric signal presented difficulties to generate a clean peak, even after filtering. Table 1.21 displays the RMSE for all tested engine points and denotes those out of the preestablished limit of 1 CA. A general trend has been observed in which over a certain limit of RPM, accelerometric signal starts to be affected by vibrations from other sources that finally impact in the precision of the MFB50 estimation. These vibration disturbances in the signal could not be eliminated with filtering because of being in the same frequency spectrum than the signal coming from cylinder combustion.

Table 1.20: Correlation percentage between estimated MFB50 from accelerometer and indicating system.

Load[bar]\RPM	2000	2400	3000	3600	4000	4400	5200
<b>0.6</b>	98.1	98.3	98.7	98.3	97.6	95.0	93.7
<b>0.8</b>	99.1	99.5	99.3	99.0	98.8	96.7	94.8
<b>1.0</b>	99.5	99.1	99.5	99.2	99.0	98.6	95.9
<b>1.2</b>	99.3	99.3	99.4	99.2	98.5	98.5	96.5
<b>1.6</b>	\	98.9	98.8	98.3	97.6	36.4	\

Table 1.21: RMSE in CA degrees for the estimation of MFB by accelerometric signal.

Load[bar]\RPM	2000	2400	3000	3600	4000	4400	5200
<b>0.6</b>	0.80	0.90	0.60	1.00	0.90	1.30	2.60
<b>0.8</b>	0.60	0.80	0.70	0.50	0.70	1.20	1.90
<b>1.0</b>	0.40	0.50	0.50	0.40	0.60	0.70	1.80
<b>1.2</b>	0.30	0.30	0.50	0.40	0.90	0.40	1.60
<b>1.6</b>	\	0.40	0.50	0.50	0.60	1.90	\

#### 1.4.2.4. Conclusions

MFB50 estimation by accelerometer signal is evaluated on a wide engine operating field. The proposed algorithm is capable of estimating MFB50 with a RMSE value below 1 CA with exception of a small area located at the highest speed. Proposed algorithm can be coupled with a standard production strategy for the knock intensity calculation and this makes accelerometric signal suitable for both knock and combustion phase index estimation. In this way it can be concretely used to supply the adaptive piston damage-based combustion controller presented above.

As anticipated in section 1.2.4 (page 40, 41), the development of a prototyping port water injection system and an innovative approach to the exhaust gas temperature modelling are other topics of the PhD course and they are discussed in the following chapters. In particular, chapter 2 deals with water injection modelling in 1-D environment (GT Power) and the development of a model-based controller that manages both the injected water mass and the spark advance. In fact, water modifies the combustion duration and then its phase, for fixed SA. The MFB50 analytical model presented in section 1.3.3.1 is thus further developed to include the sensitivity to the water and it is then implemented in an innovative combustion control strategy.

## References

- 1.1. Reif, K., "Gasoline Engine Management, Systems and Components", Springer Vieweg, 2015, DOI:10.1007/978-3-658-03964-6.
- 1.2. Curry, S., "Effects of Antiknocks on Flame Propagation in a Spark Ignition Engine", 9<sup>th</sup> Symposium on Combustion, The Combustion Institute, Pittsburgh 9, 1056-1068, 1963.
- 1.3. Downs, D., Wheeler, R. W., "Recent Development in Knock Research", Proc. Aut. Div. Inst. Of Mech. Eng., 88-99, 1951.
- 1.4. Kirsch, L.J., Quinn, C.P., "A fundamentally based Model of Knock in the Gasoline Engine", 16<sup>th</sup> Symposium on Combustion, The Combustion Institute, 16, 233-242, 1976.
- 1.5. Volkswagen AG, "Knocking of Combustion Engines", Volkswagen AG, Wolfsburg, 1981.
- 1.6. Lee, W., Schafer, H.J., "Analysis of Local Pressures Surface Temperatures and Engine Damage under Knock", SAE Paper, 830508, 1983.
- 1.7. Maly, R. and Ziegler, G., "Thermal Combustion Modeling – Theoretical and Experimental Investigation of the Knocking Process," SAE Technical Paper 820759, 1982, <https://doi.org/10.4271/820759>.
- 1.8. Maly, R., Klein, R., Peters, N., Konig, G., "Theoretical and Experimental Investigation of Knock Induced Surface Destruction", SAE Paper, 900025, 1990.
- 1.9. Miller, C.D., "Roles of Detonation Waves and Autoignition in S.I. Engine Knock", SAE Quarterly Transaction, 1, 1947.
- 1.10. Olsen, A.C., Miller, C.D., "The Interdependence of Various Types of Autoignition and Knock", NACA Report, 912, 1948.
- 1.11. Mayer, J.W., Oppenheim, A.K., "On the Shock Induced Ignition of Explosive Gases", 13<sup>th</sup> Symposium on Combustion, The Combustion Institute, Pittsburgh, 1971.
- 1.12. Majda, A.J., Roytbur, V., "Numerical Study of the Mechanism for Initiation of Reacting Shock Waves", SIAM J. Sci. Stat. Comp., 1989.
- 1.13. Y.B. Zeldovich, Regime classification of an exothermic reaction with nonuni- form initial conditions, Combust. Flame 39 (1980) 211–214.
- 1.14. D. Bradley, G.T. Kalghatgi , Influence of auto-ignition delay time characteristics of different fuels on pressure waves and knock in reciprocating engines, Com- bust. Flame 156 (2009) 2307–2318.
- 1.15. L. Bates, D. Bradley, G. Paczko, N. Peters, Engine hot spots: Modes of auto-ignition and reaction propagation, Combust. Flame 166 (2016) 80–85.
- 1.16. J. Rudloff, J.M. Zaccardi, S. Richard, J.M. Anderlohr, Proc. Combust. Inst. 34 (2013) 2959–2967.
- 1.17. Z. Chen , Studies on the initiation, propagation, and extinction of premixed flames Ph.D. Thesis, Princeton University, Princeton, NJ, USA, 2008 .
- 1.18. Pan, J., Wei, H., Shu, G., Chen, R., Effect of pressure wave disturbance on auto-ignition mode transition and knocking intensity under enclosed conditions, Combustion and Flame 185 (2017) 63–74.
- 1.19. Wang, Z., Liu, H., Reitz, R., Knocking combustion in spark-ignition engines, Progress in Energy and Combustion Science 61 (2017) 78-112.
- 1.20. Amann, M., Alger, T., Lubricant reactivity effects on gasoline spark ignition engine knock, SAE Int J Fuels Lubr, 2012;5:760–71.
- 1.21. Wang Z, Liu H, Song T, Qi Y, He X, Shuai S et al., Relationship between super-knock and pre-ignition, Int J Engine Res, 2015;16:166–80.
- 1.22. Kalghatgi GT., Developments in internal combustion engines and implications for combustion science and future transport fuels, Proc Combust Inst 2015;35:101–15.

- 1.23. Kalghatgi GT, Bradley D., Pre-ignition and super-knock in turbo-charged spark-ignition engines, *Int J Engine Res*, 2012;13:399–414.
- 1.24. Wang Z, Wang Y, Reitz RD., Pressure oscillation and chemical kinetics coupling during knock processes in gasoline engine combustion, *Energy Fuels*, 2012;26:7107–19.
- 1.25. Zaccardi J-M, Escudié D., Overview of the main mechanisms triggering low-speed pre-ignition in spark-ignition engines, *Int J Engine Res*, 2015;16:152–65.
- 1.26. Haenel, P., Seyfried, P., Kleeberg, H., and Tomazic, D., "Systematic Approach to Analyze and Characterize Pre-ignition Events in Turbocharged Direct-injected Gasoline Engines," SAE Technical Paper 2011-01-0343, 2011, <https://doi.org/10.4271/2011-01-0343>.
- 1.27. Zahdeh, A., Rothenberger, P., Nguyen, W., Anbarasu, M. et al., "Fundamental Approach to Investigate Pre-Ignition in Boosted SI Engines," *SAE Int. J. Engines* 4(1):246-273, 2011, <https://doi.org/10.4271/2011-01-0340>.
- 1.28. Grandin, B. and Denbratt, I., "The Effect of Knock on Heat Transfer in SI Engines," SAE Technical Paper 2002-01-0238, 2002, <https://doi.org/10.4271/2002-01-0238>.
- 1.29. R.J. Nates, "Knock Damage in Spark-Ignition Engines", University of Cape Town, 1995.
- 1.30. N. Cavina, N. Rojo, A. Businaro, L. Ceschini, E. Balducci, A. Cerofolini, "Analysis of pre-ignition combustions triggered by heavy knocking events in a turbocharged GDI engine", *Energy Procedia* 101 (2016) 893–900, <http://dx.doi.org/10.1016/j.egypro.2016.11.113>.
- 1.31. Nates, R., "Thermal Stresses Induced by Knocking Combustion in Spark-Ignition Engines," SAE Technical Paper 2000-01-1238, 2000, <https://doi.org/10.4271/2000-01-1238>.
- 1.32. Nates, R. and Yates, A., "Knock Damage Mechanisms in Spark-Ignition Engines," SAE Technical Paper 942064, 1994, <https://doi.org/10.4271/942064>.
- 1.33. Mahle GmbH - "Pistons and engine testing", 2012, ISBN 978-3-8348-8662-0.
- 1.34. J. Fitton and R. J. Nates, "Investigation into the relationship between knock intensity and piston seizure", 1992.
- 1.35. Boehm, G. and Harrer, J., "Nickel Coated Pistons for Improved Durability in Knock Control Engines," SAE Technical Paper 900453, 1990, <https://doi.org/10.4271/900453>.
- 1.36. Renault, F., "A New Technique to Detect and Control Knock Damage," SAE Technical Paper 820073, 1982, <https://doi.org/10.4271/820073>.
- 1.37. Essig, G., Schellmann, K., and Wacker, E., "Piston Loading Under Knocking Combustion (Detonation)," SAE Technical Paper 825011, 1982.
- 1.38. Spelina, J., Peyton Jones, J., and Frey, J., "Recent Advances in Knock Analysis, Simulation, and Control," *SAE Int. J. Engines* 7(2):947-955, 2014, <https://doi.org/10.4271/2014-01-1349>.
- 1.39. Spelina, J.M., Peyton Jones, J.C., and Frey, J. "Characterization of knock intensity distributions, Part I: statistical independence and scalar measures," *Proc IMechE, Part D: J. Auto Eng*, in print, doi:10.1177/0954407013496233.
- 1.40. Spelina, J.M., Peyton Jones, J.C., and Frey, J. "Characterization of knock intensity distributions, Part II: parametric models," *Proc IMechE, Part D: J. Auto Eng*, in print, doi:10.1177/0954407013496234.
- 1.41. Lonari, Y., Polonowski, C., Naber, J., and Chen, B., "Stochastic Knock Detection Model for Spark Ignited Engines," SAE Technical Paper 2011-01-1421, 2011, doi:10.4271/2011-01-1421.
- 1.42. Wenig, M., Grill, M. and Bargende, M., "A New Approach for Modeling Cycle-to-Cycle Variations within the Framework of a Real Working-Process Simulation," *SAE Int. J. Engines* 6(2):2013, doi:10.4271/2013-01-1315.
- 1.43. D'Adamo, A., Breda, S., Fontanesi, S., and Cantore, G., "A RANS-Based CFD Model to Predict the Statistical Occurrence of Knock in Spark-Ignition Engines," *SAE Int. J. Engines* 9(1):2016, doi:10.4271/2016-01-0581.

- 1.44. J.C. Livengood, P.C. Wu, "Correlation of auto-ignition Phenomena in Internal Combustion Engines and Rapid Compression Machines", 5th Symposium (International) on Combustion, pp.347-356, 1957.
- 1.45. A.M. Douaud and P.Eyzat, "Fou-Octane-Number Method for Predicting the Anti-Knock Behavior of Fuels and Engines", SAE Technical Paper Series, SAE 780080, 1978.
- 1.46. Worret, R., Bernhardt, S., Schwarz, F., and Spicher, U., "Application of Different Cylinder Pressure Based Knock Detection Methods in Spark Ignition Engines," SAE Technical Paper 2002-01-1668, 2002.
- 1.47. Naik, C., Pitz, W., Westbrook, C., Sjöberg, M. et al., "Detailed Chemical Kinetic Modeling of Surrogate Fuels for Gasoline and Application to an HCCI Engine," SAE Technical Paper 2005-01-3741, 2005, doi:10.4271/2005-01-3741.
- 1.48. Curran, H., Gaffuri, P., Pitz, W., Westbrook, C. et al., "Autoignition Chemistry of the Hexane Isomers: An Experimental and Kinetic Modeling Study," SAE Technical Paper 952406, 1995, doi:10.4271/952406.
- 1.49. Elmquist, C., Lindström, F., Ångström, H., Grandin, B. et al., "Optimizing Engine Concepts by Using a Simple Model for Knock Prediction," SAE Technical Paper 2003-01-3123, 2003, doi:10.4271/2003-01-3123.
- 1.50. Kleinschmidt W., "Zur Simulation des Betriebes von Ottomotoren an der Klopfgrenze", VDI Fortschrittsbericht, Reihe 12 422, VDI 2000
- 1.51. Jenkin, R., James, E., and Malalasekera, W., "Predicting the Onset of End-Gas Autoignition with a Quasi-Dimensional Spark Ignition Engine Model," SAE Technical Paper 972877, 1997, doi:10.4271/972877.
- 1.52. Corti, E., Forte, C., Cazzoli, G., Moro, D., Falfari, S., Ravaglioli, V., "Comparison of Knock Indexes Based on CFD Analysis", Energy Procedia, Volume 101, 2016, Pages 917-924, ISSN 1876-6102, <http://dx.doi.org/10.1016/j.egypro.2016.11.116>.
- 1.53. Corti, E., Forte, C., Bianchi, G., and Moro, D., "Relating Knocking Combustions Effects to Measurable Data," SAE Int. J. Engines 8(5):2015, doi:10.4271/2015-24-2429.
- 1.54. Millo, F., Rolando, L., Pautasso, E., and Servetto, E., "A Methodology to Mimic Cycle to Cycle Variations and to Predict Knock Occurrence through Numerical Simulation," SAE Technical Paper 2014-01-1070, 2014, doi:10.4271/2014-01-1070.
- 1.55. Eckert, P., Kong, S., and Reitz, R., "Modeling Autoignition and Engine Knock Under Spark Ignition Conditions," SAE Technical Paper 2003-01-0011, 2003, doi:10.4271/2003-01-0011.
- 1.56. Teraji A, Tsuda T, Noda, T., Kubo, M., et al., "Development of a three-dimensional knock simulation method incorporating a high accuracy flame propagation model," Int J. Engine Research, 6(1): 73-83, 2005.
- 1.57. Schmillen, K. and Rechs, M., "Different Methods of Knock Detection and Knock Control," SAE Technical Paper 910858, 1991, doi:10.4271/910858.
- 1.58. Peyton Jones, J. and Muske, K., "A Database-Driven In-Cycle Engine Simulator for Control, Calibration and Robustness Testing," SAE Int. J. Passeng. Cars - Electron. Electr. Syst. 1(1):425-431, 2008, doi:10.4271/2008-01-1002.
- 1.59. Peyton Jones, J., Spelina, J., and Frey, J., "Control-Oriented Knock Simulation," SAE Int. J. Engines 9(2):1201-1209, 2016, <https://doi.org/10.4271/2016-01-0821>.
- 1.60. Potteau, S., Lutz, P., Leroux, S., Moroz, S. et al., "Cooled EGR for a Turbo SI Engine to Reduce Knocking and Fuel Consumption," SAE Technical Paper 2007-01-3978, 2007, <https://doi.org/10.4271/2007-01-3978>.
- 1.61. Grandin, B., Ångström, H., Stålhammar, P., and Olofsson, E., "Knock Suppression in a Turbocharged SI Engine by Using Cooled EGR," SAE Technical Paper 982476, 1998, <https://doi.org/10.4271/982476>.

- 1.62. Alger, T., Mangold, B., Roberts, C., and Gingrich, J., "The Interaction of Fuel Anti-Knock Index and Cooled EGR on Engine Performance and Efficiency," SAE Int. J. Engines 5(3):1229-1241, 2012, <https://doi.org/10.4271/2012-01-1149>.
- 1.63. Y-l Bai, J-x Wang, Wang Z, Shuai S-j, Knocking suppression by stratified stoichiometric mixture with two-zone homogeneity in a DISI engine, J Eng Gas Turbines Power, 2012;135:012803.
- 1.64. Liang, L., Reitz, R., Iyer, C., and Yi, J., "Modeling Knock in Spark-Ignition Engines Using a G-equation Combustion Model Incorporating Detailed Chemical Kinetics," SAE Technical Paper 2007-01-0165, 2007, <https://doi.org/10.4271/2007-01-0165>.
- 1.65. Chintala V, Subramanian KA, Experimental investigations on effect of different compression ratios on enhancement of maximum hydrogen energy share in a compression ignition engine under dual-fuel mode, Energy 2015;87:448-62, <https://doi.org/10.1016/j.energy.2015.05.014>.
- 1.66. Kim N, Cho S, Min K, A study on the combustion and emission characteristics of an SI engine under full load conditions with ethanol port injection and gasoline direct injection, Fuel 2015;158:725-32, <https://doi.org/10.1016/j.fuel.2015.06.025>.
- 1.67. Liu H, Wang Z, Long Y, Wang J, Dual-fuel spark ignition (DFSI) combustion fuel led with different alcohols and gasoline for fuel efficiency, Fuel 2015;157:255-60, <https://doi.org/10.1016/j.fuel.2015.04.042>.
- 1.68. Peters, N., Kerschgens, B., and Paczko, G., "Super-Knock Prediction Using a Refined Theory of Turbulence," SAE Int. J. Engines 6(2):953-967, 2013, <https://doi.org/10.4271/2013-01-1109>.
- 1.69. Towers, J. and Hoekstra, R., "Engine Knock, A Renewed Concern In Motorsports - A Literature Review," SAE Technical Paper 983026, 1998, <https://doi.org/10.4271/983026>.
- 1.70. Luisi, S., Doria, V., Stroppiana, A., Millo, F. et al., "Experimental Investigation on Early and Late Intake Valve Closures for Knock Mitigation through Miller Cycle in a Downsized Turbocharged Engine," SAE Technical Paper 2015-01-0760, 2015, <https://doi.org/10.4271/2015-01-0760>.
- 1.71. Peyton Jones, J., Muske, K., Frey, J., and Scholl, D., "A Stochastic Knock Control Algorithm," SAE Technical Paper 2009-01-1017, 2009, doi:10.4271/2009-01-1017.
- 1.72. Peyton Jones, J., Frey, J., and Muske, K., "A Fast-Acting Stochastic Approach to Knock Control," IFAC Workshop on Engine Powertrain Control, Simulation and Modeling, 2009, doi:10.3182/20091130-3-FR-4008.00003.
- 1.73. Peyton Jones, J.C., Spelina, J.M., and Frey, J. "Likelihood-based Control of Engine Knock," IEEE Trans. on Control Sys. Tech. PP(99): 1, 2013, doi:10.1109/TCST.2012.2229280.
- 1.74. Thomasson, A., Eriksson, L., Lindell, T., Peyton Jones, J., et al., "Tuning and Experimental Evaluation of a Likelihood-Based Engine Knock Controller," to be presented at 2013 IEEE 52nd Conference on Decision & Control, Italy, December 10-13, 2013.
- 1.75. Ghandhi, J. and Kim, K., "A Statistical Description of Knock Intensity and Its Prediction," SAE Technical Paper 2017-01-0659, 2017, doi:10.4271/2017-01-0659.
- 1.76. Rojo, N., "Development and experimental validation of a knock induced damage model and real-time implementation of model-based strategies to control knock intensity in boosted gasoline engines", University of Bologna, 2019 (under the embargo period).
- 1.77. Cavina, N., Rojo, N., Ceschini, L., Balducci, E. et al., "Investigation of Knock Damage Mechanisms on a GDI TC Engine," SAE Technical Paper 2017-24-0060, 2017, doi:10.4271/2017-24-0060.
- 1.78. Mathworks, Matlab Documentation, R2017a, 2017.

- 1.79. Cavina, N., Brusa, A., Rojo, N., and Corti, E., "Statistical Analysis of Knock Intensity Probability Distribution and Development of 0-D Predictive Knock Model for a SI TC Engine," SAE Technical Paper 2018-01-0858, 2018, doi:10.4271/2018-01-0858.
- 1.80. Gamma Technology Inc., 2017. GT Gamma Technology.
- 1.81. Brusa, A., Cavina, N., Rojo, N., Cucchi, M. et al., "Development and Validation of a Control-Oriented Analytic Engine Simulator," SAE Technical Paper 2019-24-0002, 2019, doi:10.4271/2019-24-0002.
- 1.82. Corti, E. and Forte, C., "Statistical Analysis of Indicating Parameters for Knock Detection Purposes," SAE Technical Paper 2009-01-0237, 2009, <https://doi.org/10.4271/2009-01-0237>.
- 1.83. Businaro, A., Cavina, N., Corti, E., Mancini, G., Moro, D., Ponti, F., Ravaglioli, V., "Accelerometer Based Methodology for Combustion Parameters Estimation", Energy Procedia, Volume 81, December 2015, Pages 950-959, <https://doi.org/10.1016/j.egypro.2015.12.15>
- 1.84. Tibaldi, M., "Controlli automatici vol.2", Pitagora, 1990, ISBN: 8837105002.
- 1.85. Patent: NGK Spark Plug Co., Ltd (2003). EP 1 338 879 B1. 01998804.7.
- 1.86. Patent: NGK Spark Plug Co., Ltd., Nagoya, Japan (2019). EP 1 338 879 B1. 4686861.
- 1.87. Patent: Cessna Aircraft Company Wichita Kansas 67201, USA (1996), EP 0 715 160 B1.
- 1.88. Patent: Nissan Motor Co., Ltd, Yokohama, Japan (1986), US Patent 4,566,316.
- 1.89. Patent: NGK Spark Plug, Ltd, Nagoya, Japan (1987), US Patent 4,686,861.
- 1.90. Cavina, N., Rojo, N., Poggio, L., Calogero, L. et al., "Investigation on Pre-Ignition Combustion Events and Development of Diagnostic Solutions Based on Ion Current Signals," SAE Int. J. Engines 10(4):2017, doi:10.4271/2017-01-0784.
- 1.91. Cavina, N., Sgatti, S., Cavanna, F., and Bisanti, G., "Combustion Monitoring Based on Engine Acoustic Emission Signal Processing," SAE Technical Paper 2009-01-1024, 2009, <https://doi.org/10.4271/2009-01-1024>.
- 1.92. Romani, L., Lenzi, G., Ferrari, L., and Ferrara, G., "Indirect Estimation of In-Cylinder Pressure through the Stress Analysis of an Engine Stud", SAE Technical Paper 2016-01-0814, 2016, <https://doi.org/10.4271/2016-01-0814>.
- 1.93. Corti, E., Abbondanza, M., Ponti, F., and Raggini, L., "The Use of Piezoelectric Washers for Feedback Combustion Control," SAE Technical Paper 2020-01-1146, 2020, <https://doi.org/10.4271/2020-01-1146>.
- 1.94. Monegato, G., "Elementi di calcolo numerico", Levrotto & Bella, 1997, ISBN:8882180174.
- 1.95. Ranuzzi, F., Cavina, N., Scocozza, G., Brusa, A. et al., "Experimental Validation of a Model-Based Water Injection Combustion Control System for On-Board Application," SAE Technical Paper 2019-24-0015, 2019, doi:10.4271/2019-24-0015.

## 2. Water Injection Modelling and Control

## 2.1. Introduction and Motivations

The aim of author research activity is the development of innovative solutions and control strategies for the combustion efficiency increase in modern SI engines. As widely discussed in chapter 1, knock is the most limiting factor for the optimization of the combustion phase on the entire engine operating field. Considering the control-oriented solution for knock mitigation, the most common strategy to manage knock intensity in modern SI engines is to retard spark advance for critical load levels. The adaptive damage-based combustion controller presented above is the novel proposal of the author to face knock control issues in a completely innovative way. Nevertheless, the standard approach of spark retarding causes an increase of gas temperature at the turbine inlet, especially in downsized TC engines with specific powers, that go up to 150 HP/L. Therefore, as the SA decreases, the mixture tends to get rich, targeting lambda values that can go down to 0.8-0.7. In fact, the exceeding fuel, evaporating, removes heat from gases, lowering in-chamber and exhaust gas temperature. Nevertheless, this solution goes in a diametrically opposite direction to the reduction in consumption imposed by recent laws about CO<sub>2</sub> emissions. It is precisely in this context that the idea of Water Injection (WI) is introduced and developed to contain pressures and temperatures in the combustion chamber and effectively mitigate knock. WI has been investigated during PhD course as a viable solution for reducing the in-cylinder and exhaust gas temperature. The latter is thus another key topic in the field of combustion process management and a novel control-oriented approach for gas temperature modelling at the turbine inlet is presented in chapter 3. Water injection modelling and control is instead the focus of this chapter.

WI is certainly not a new solution. It was studied as early as 1930 by the British company Ricardo and has subsequently been taken up and re-evaluated on numerous occasions. Between 1970 and 1980 it was tested on some Chrysler and Saab turbo engines, although the first car equipped with this technology was the 1990 Toyota Celica. The common objective of all these first applications was to use the high latent heat of water vaporization to limit the temperatures in the chamber and exhaust, in particular to prevent the turbine failure. WI is currently used in motorsport (e.g. on the BMW M4 GTS) and aeronautic applications and there are several studies about benefits that could be obtained even in diesel engines. Indeed, reduction of temperatures during the combustion phase effectively limits the formation of the most problematic pollutants for CI engines, namely NO<sub>x</sub> and the particulate matter (PM). Other solutions are also being evaluated that envisage coupling WI with complex technologies such as VNT (Variable Nozzle Turbochargers), CEGR (Cooled Exhaust Gas Recirculation) and VCR (Variable Compression Ratio).

Main advantage resulting from the use of the WI, however, remains the important reduction of temperatures that allows to avoid the mixture enrichment at critical engine points and this element makes it certainly attractive for further investigations. It must be pointed out that, from the moment the Real Driving Emission has become a binding test for the homologation of passenger cars (i.e. the end of 2018), small engines will hardly be able to maintain fuel consumption and emissions within the legal limits. For sure the road test would lead to the use of engines at higher speed and load levels than those typical of homologation test, causing an increase in pollutant emissions.

It is no mystery that the biggest automotive groups such as General Motors, Volkswagen and the FCA group are working on larger engines than those currently in production. The increase in displacement is between 10 and 30% on average for gasoline engines under 1.2 L and diesel engines under 1.5 L. Clearly in this scenario, the containment of temperatures offered by water injection is an important element both in terms of consumption and emissions.

First phase of activity carried out during PhD course is focused on development and validation of a 1-D water injection model. A single cylinder GT Power model of Volkswagen 1.4 TSI engine installed at the test cell in Forlì has been developed and validated. Such model includes components between the intercooler outlet and the turbine inlet. An injector template is introduced on the intake manifold to model water injection system mounted on the real engine (i.e. a Port Water Injection) and even to take into account the dynamics of storage, transport and evaporation of the fluid inside ducts.

Model was then further modified by adding a second water injector template connected directly to the cylinder, to simulate and control that fraction of the total water mass that evaporates within the combustion chamber.

The second phase deals with development of a model-based control strategy able to manage injected water mass to obtain the best benefits in terms of combustion efficiency and knock control. At the same time, also the spark advance must be controlled to obtain the desired combustion phase and a compensation of spark timing is required due to the delay of combustion dynamics given by the introduction of an inert fluid like water. The value of injected water mass must be chosen as the smallest value able to guarantee the necessary mitigation of dangerous events to keep knock level under a defined threshold, while combustion phase (MFB50) is maintained at optimum angle. The choice of the optimization of injected water mass allows to make the controller compatible with an eventual production application. On the other hand, spark advance (SA) must be defined to achieve the optimum MFB50 angle, considering the delay induced by water addition. The MFB50 analytical model described in section 1.3.3.1 is particularly suitable for this application. It is further developed to include the sensitivity to the injected water mass, and it is then implemented in the combustion controller to ensure the highest combustion efficiency at high load using WI, by the determination of optimal SA for given operating conditions (defined for fixed engine speed, load, lambda and injected water mass). The analytical approach, as widely described in the previous chapter, makes the model executable with an extremely low computational effort. On the other hand, the necessity of a 4-dimensional lookup table is avoided (the SA should be mapped on a grid of different values of engine speed, load, injected water mass and MFB50). Hence, the most important result of this activity is the update of MFB50 analytical model, demonstrating its ability to maximize efficiency by accurately tracking target MFB50 also under knock-limited operation, while guaranteeing engine integrity and minimizing water consumption.

Resulting Water Injection Combustion Controller (WICC) is finally tested and validated with a Software in the Loop system, using the GT Power model as virtual engine, and then at the test bench by implementing the algorithm in the RT machine for Rapid Control Prototyping developed by the author and described in detail in the Appendix (used also for testing the adaptive damage-based combustion controller, as anticipated in section 1.4.1).

Activities and results proposed in this chapter have been already published in [2.1, 2.2, 2.3].

## 2.2. Port Water Injection System 1-D Modelling

In this section Port Water Injection (PWI) possibilities are investigated by modelling in 1-D environment (GT Power) the prototyping PWI system developed by the author research group and installed on a 4-cylinder VW TSI engine. Since this layout results in no modification or complication of the cylinder head it can be easily installed on a pre-existing engine. The simplicity of this layout may compensate extra benefits and operation flexibility obtainable with direct water injection. Experimental tests are conducted on mentioned engine at the test cell in Forlì to verify and validate the model. Even if experimental tests were limited to port water injection, possibilities given by a direct injector have also been investigated with the 1-D model.

### 2.2.1. Water Injection and Knock Model

To evaluate water injection effects on combustion features, a 1-D model of the 4-cylinder engine used for the development of analytical knock model (section 1.3.2.1) has been developed in Gt-Power [2.4] environment. In this first stage, a single-cylinder model has been set, to reduce computational efforts. Main characteristics of engine were reported in Table 1.3, but they are collected in Table 2.1 for clarity reasons.

Table 2.1: Engine specifications.

Displaced volume	1389.9 cc (4 cylinder)
Stroke	75.6 mm
Bore	76.5 mm
Connecting Rod	144 mm
Compression ratio	10:1
Number of Valves	4
Exhaust Valve Open	580° BTDC @ 0.1 mm lift
Exhaust Valve Close	356° BTDC @ 0.1 mm lift
Inlet Valve Open	358° BTDC @ 0.1 mm lift
Inlet Valve Close	132° BTDC @ 0.1 mm lift

#### 2.2.1.1. Model Layout

The developed engine model is based on a 1-D representation, and only one cylinder, in this first stage, is considered. Moreover, components upstream the intake and downstream the exhaust manifolds have been neglected. Model parameters identifications has been carried out through experimental data, without water injection, in several operating conditions, with production engine calibrations.

Combustion simulation is governed by a predictive three-zones model (implemented in CombSITurb GT Power template [2.4]), previously calibrated via Three Pressure Analysis, as described in section 1.3.2.3. Hence, in this chapter results of the calibration process of predictive combustion mode are reported. Such model should be sensitive to in-cylinder variations of the charge physical properties due to heat subtraction operated by water evaporation. Indeed, primary aim of this specific modelling activity is to simulate the effects on combustion process of a PWI system, which is also the experimental configuration available at the test bench.

To consider water film formation and evaporation dynamics in the intake ports, PipePort and FlowSplitPort templates (circled in green in Figure 2.1) have been used, which are designed to model

port water injection [2.4]. Water injection within intake runners implies that a fraction of liquid mass vaporizes within ducts, and remaining mass enters the combustion chamber in liquid form. But model intake ducts thermal conditions cause evaporation of entire injected water mass, so configuration with a port water injection with does not allow considering liquid fraction that evaporates within cylinder, due to the specific boundary conditions reached within intake runners. In the simulations, ducts wall temperature depends on working conditions (manifold pressure and engine speed). The water introduced with port injection (circled in blue in Figure 2.1) forms a fluid film against intake ducts walls, and it entirely evaporates within runners. In particular, the fluid film evaporates during intake stroke, when liquid it achieves critical pressure and temperature conditions.

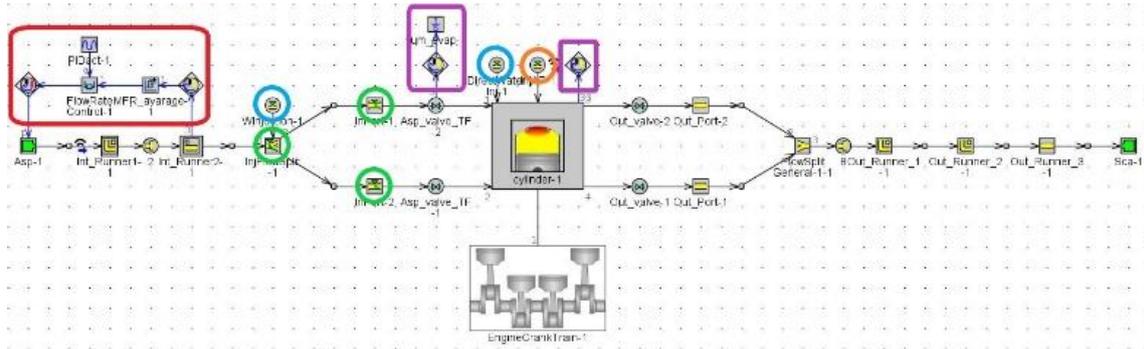


Figure 2.1: Gt-Power model layout. Intake, cylinder, and exhaust sub-models are shown from left to right. Sensors that measure water mass toward valves and within cylinder are marked in violet.

To control the amount of liquid water entering the cylinder, a fictitious direct injector was added in the model (blue circle upon cylinder in Figure 2.1). This injector is set in order to vaporize injected water instantaneously during injection and in this way evaporation rate for the liquid mass fraction that reaches the combustion chamber can be controlled (i.e., imposed) by the direct injection rate. Direct injection starts with IVO because it is assumed that water which is still in liquid form in the intake ducts starts entering the cylinder as soon as the intake valves open. A PID controller calibrated with Ziegler-Nichols method (marked in red in Figure 2.1) has been introduced to ensure the same air mass flow rate within intake runners, managing manifold pressure. Figure 1 shows a schematic of the developed 1-D model.

### 2.2.1.2. Water Injection Modelling Parameters

The total amount of injected water is described by  $r$ , which is the ratio between injected water quantity and fuel mass. This convenient normalization is proposed in [2.5]. The split between port-injected and (fictitious) directly injected water mass is described, and user-imposed, by  $F_i$  and  $F_d$ , which represent the intake-injected and directly injected water mass fractions, respectively. By definition, the sum of these two parameters is 1. The Start of Injection (SOI) for the port injector has practically no effects due to the model intake dynamic characteristics. In fact, as mentioned above, evaporation of entire fluid wall film takes place during intake stroke. This is one of the main limits of this model, which cannot represent tridimensional interactions between evaporating water spray and pressure waves in the intake manifolds. In-cylinder injection start is imposed at IVO (Intake Valve Opening), since it should model port-injected liquid water entering the combustion chamber during the intake stroke. This hypothesis is not necessarily verified: depending on the port-injection timing, injected water may not enter the cylinder during the first part of the intake stroke. But model

simulations reveal that this is the most unfavorable condition, and also the most probable one when port water injection is implemented.

Injection (and therefore evaporation) duration is imposed too and described by the parameter ToE: Time of Evaporation, which will be considered below in terms of equivalent crank angle interval (for now, evaporation rate, or equivalently direct injector mass flow, is supposed to be constant). The split between in-cylinder and in-runner water evaporation (i.e.  $F_d$  and  $F_i$ ) has to be identified and calibrated, as well as the parameter ToE, depending on injection phasing and engine operating conditions. Indeed, this modelling approach is relatively simple, but it is not predictive.

A first sensitivity analysis to these parameters variations has been conducted using the above-described model. In this way, parameters that effectively represent degrees of freedom of the model can be identified to fit experimental data. Variation of parameters  $F_d$ ,  $F_i$  and  $r$  affects intake air mass flow rate, so during these simulations PID controller has been set on. Other parameters, which do not significantly affect the model outputs, due to the grey/black-box nature of this part of the model, or because of expected very weak physical correlations, will be kept constant. Location of the water evaporation (where it takes place) has a severe impact on the useful effect of water injection or, equivalently, on water consumption. As shown in Figure 2.2, by varying the parameter  $F_i$  from 0 to 1 (corresponding to  $F_d$  variations from 1 to 0), the same water mass ( $r = 0.5$ , in this case) progressively produces minor impact on in-cylinder temperature. Figure 2.2 shows that temperature reduction achievable with direct injection is the highest. The engine operating point related to Figure 2.2 is defined by engine speed equal to 3000 rpm and intake manifold pressure equal to 1.5 bar, with a constant SA of  $13^\circ$  CA Before Top Dead Center (BTDC), but the results are qualitatively valid also for different operating conditions.

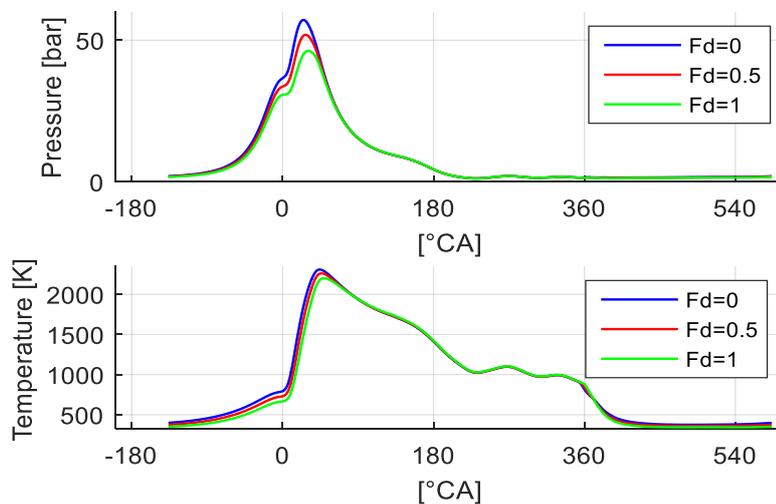


Figure 2.2: Effect of  $F_d$  variation for fixed water quantity. Top graph represents such effects on the in-cylinder pressure profile, while in the bottom graph the corresponding effect on the in-cylinder temperature is shown.

This means that direct injection is undoubtedly the best solution.

These evidences confirm results reported in [2.5] and suggest, supposing a port injection solution, an injector installation as close as possible to the intake valves, while a single-point configuration is not recommended. Figure 2.3 clearly shows that also the angular position of the in-cylinder evaporation (i.e. by varying in-cylinder injection duration, since the SOI is bonded to IVO event) affects temperature reduction, for fixed  $r$  and  $F_i/F_d$  values.

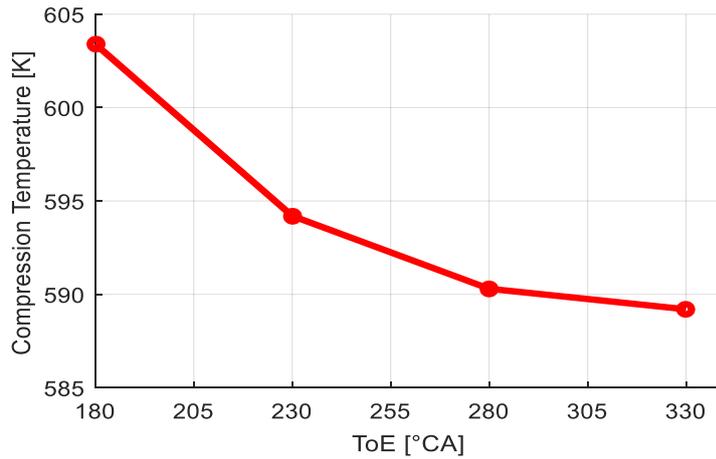


Figure 2.3: Simulated in-cylinder temperature during the compression stroke (25°CA BTDC) depending on ToE parameter. As in-cylinder evaporation duration (ToE) is longer, the charge temperature is lower.

Figure 2.3 is related to the same operating conditions of Figure 2.2, and the effect of ToE parameter is shown: the later (or longer) evaporation in the compression stroke produces higher temperature reduction. Even in this case, such considerations can be generalized, as verified by applying the very same model to different engine operating conditions.

Injected liquid water temperature does not significantly affect model response, as heat subtraction is given basically by its latent heat of evaporation. This means that, supposing an on-board application, water tank position under the bonnet is not particularly constrained.

As mentioned before, the model is not sensitive to SOI variations for the port injector, contrary to how the real process is expected to behave.

Similarly, water consumption dependence on these parameters can be analyzed, for fixed/given useful temperature reduction effect. In Figure 2.4, the increment of water consumption is evaluated as a function of  $F_d$  (i.e. the fraction of water that evaporates within the cylinder), compared to a direct injection ( $F_d = 1$ ). This analysis is done with fixed ToE, in this case equal to 330°CA, while  $r$  is determined to achieve the target effect, which is the same in-cylinder maximum temperature. The engine operating point is the same as the one shown in previous figures, and in the subsequent Figure 2.5.

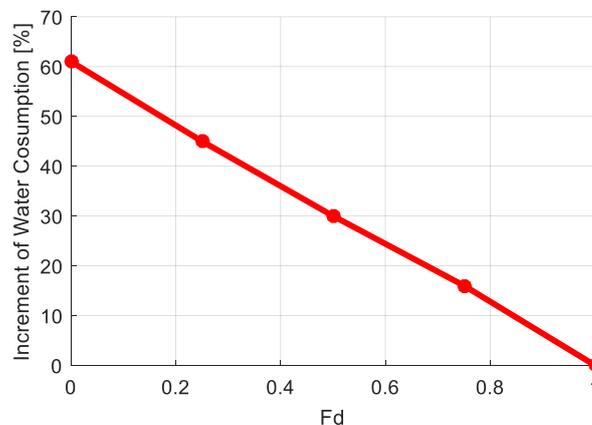


Figure 2.4: Percentage increase of water consumption, for fixed maximum in-cylinder temperature, depending on the evaporation split between intake ducts ( $F_i$ ) and combustion chamber ( $F_d$ ).

By transferring the evaporation process from the cylinder to the intake ducts, the increase of water consumption is very significant (up to 60%) to achieve the same effect on the in-cylinder conditions.

In Figure 2.5 the same kind of analysis is carried out with respect to ToE parameter. Reported values describe the increase of water consumption, compared to a full direct injection with ToE equal to 180°CA, as a function of parameter ToE. The longer evaporation duration makes lower water consumption to achieve the same target effect.

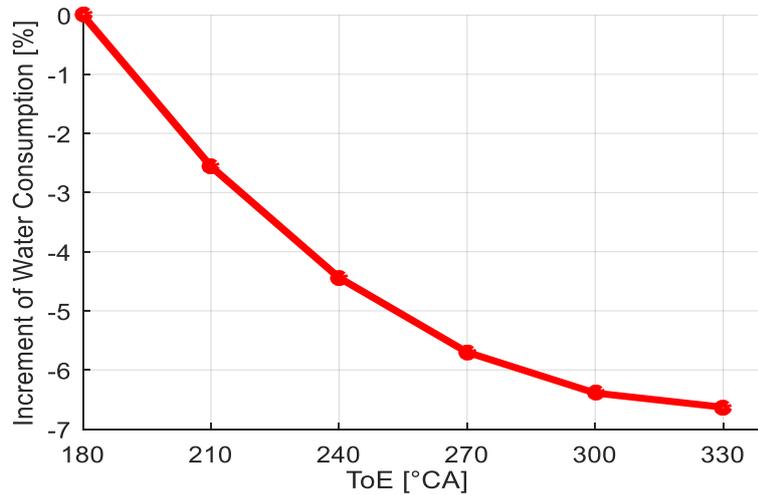


Figure 2.5: Percentage increase of water consumption, for fixed maximum in-cylinder temperature, depending on the imposed duration of the in-cylinder evaporation. As the in-cylinder evaporation is longer (ToE), a lower value of  $r$  is required.

Therefore, considering an external water injection layout, it is clear the importance of the injector location and installation, in order to achieve maximum benefits while minimizing water consumption. Going back to the analysis reported in Figure 2.4, it is now clear that a long duration of the in-cylinder evaporation process is desirable, so that the trend reported in the same figure is the most pessimistic, while if considering a shorter duration (i.e. lower ToE value) the increase of water consumption corresponding to a reduction of  $F_d$  is lower.

### 2.2.1.3. GT Power Knock Model

The approach available on GT Power to model knocking combustion requires the cycle-to-cycle variability (CCV) to be modelled, which can be obtained through stochastic variations of the main influencing parameters. In this way, the model generates a statistical distribution for each selected parameter, and every modeled combustion process is different from the previous. Knock model is then deterministically applied to every stochastic combustion, achieving a statistical distribution of the knock index. To model CCV, *EngCylinderCCV* template has been introduced in the model, and equations proposed in [2.6] have been implemented. In the following, the empirical relationship adopted between CCV and variation of laminar flame speed and kernel growth speed is described. Model equations define the standard deviation of these two quantities as follows:

$$S_F[\%] = \left( e^{\frac{0.9285 - \chi_{ZS}}{0.17}} B_m^{-\chi_{ZS}} NF(\chi_{ZS}) \right) \quad (2.1)$$

$$NF(\chi_{ZS}) = \left(\frac{BD1090}{1000}\right)^{\chi_{ZS}-1} BD0010^{0.5 \chi_{ZS}} \left(\frac{T_{u,max}}{T_{ref}}\right)^{2(\chi_{ZS}+1)} \quad (2.2)$$

$$\chi_{ZS} = \chi_{ZS,mod} + 0.008 \chi_n \left(\frac{n}{1000}\right)^{-2} \quad (2.3)$$

$$EP = \varphi_{ZS} + 2.6 \varphi_n \left(\frac{n}{1000}\right)^{-2} \quad (2.4)$$

Where:

- $S_F$  [%] is the percentage standard deviation of laminar flame speed
- $B_m$  is the maximum value of laminar flame speed
- $T_{u,max}$  is the maximum exhaust gas temperature
- $n$  is the engine speed
- BD1090 and BD0010 are the burn duration from 10 to 90 % and from 0 to 10 % of fuel mass fraction
- $\chi_{ZS,mod}$  is the *Laminar Flame Speed Variation Multiplier*
- $\chi_n$  is a non-dimensional quantity, normally set to 1
- $EP$  is standard deviation of kernel growth speed
- $\varphi_{ZS}$  is the *Flame Kernel Growth Variation Multiplier*
- $\varphi_n$  is a non-dimensional quantity, normally set to 1

Calibration of the CCV model has been carried out by varying *Laminar Flame Speed Variation Multiplier* and *Flame Kernel Growth Variation Multiplier*, to minimize the parameter  $f_{o,CCV}$ , defined by the following equation:

$$f_{o,CCV} = 2 \left( \frac{STD_{imep,GT} - STD_{imep,ex}}{STD_{imep,ex}} \right)^2 + \left( \frac{STD_{pmax,GT} - STD_{pmax,ex}}{STD_{pmax,ex}} \right)^2 + \left( \frac{STD_{CA50,GT} - STD_{CA50,ex}}{STD_{CA50,ex}} \right)^2 \quad (2.5)$$

In Equation (2.5),  $STD_{x,GT}$  is the standard deviation of the quantity “x” calculated by GT Power and  $STD_{x,ex}$  is the standard deviation of the same “x” quantity experimentally measured in the test cell. Then, the *Kinetics Fit* method [2.7] already introduced in section 1.2.3.2 has been applied, in order to model knock intensity. This model calculates a knock index, whose value or intensity depends on the unburned fuel mass when auto-ignition conditions occur in the combustion chamber. The adopted knock index has been defined as follows:

$$KI = 1000 M u_b \left(\frac{V_{TDC}}{V}\right) e^{\left(\frac{-6000}{T_u}\right)} \max(0.1 - (1 - \phi)^2 I_{ave}) \quad (2.6)$$

Where:

- $M$  is the *Knock Index Multiplier*
- $u_b$  is the unburned fraction of the mixture when auto-ignition occurs
- $V_{TDC}$  is the in-cylinder volume at TDC
- $V$  is the in-cylinder volume when auto-ignition occurs
- $T_u$  is the temperature of the unburned mass fraction
- $\phi$  is the equivalence ratio of the unburned zone
- $I_{ave}$  is the integral of the induction time

Induction time integral is defined by the following expression:

$$I(t) = \int_{SOC}^t \frac{1}{\tau} dt \quad (2.7)$$

Where  $SOC$  is the Start of Combustion and  $\tau$  is the Induction Time (autoignition delay) defined as follows:

$$\frac{1}{\tau} = \frac{1}{\tau_1 + \tau_2} + \frac{1}{\tau_3} \quad (2.8)$$

$$\tau_i = M_1 a_i \left( \frac{RON}{100} \right)^{b_i} [Fuel]^{c_i} [O_2]^{d_i} [Dilution]^{e_i} \exp\left( \frac{f_i}{M_2 T} \right) \quad \text{for } i = 1, 2, 3 \quad (2.9)$$

Where:

- $\tau_1, \tau_2$  and  $\tau_3$  are Induction Times for low, intermediate and high temperature regions
- $RON$  is the fuel Research Octane Number
- $[Fuel]$  is the fuel concentration (mol/m<sup>3</sup>)
- $[O_2]$  is the oxygen concentration (mol/m<sup>3</sup>)
- $[Dilution]$  is the sum of concentration of N<sub>2</sub>, CO<sub>2</sub> and H<sub>2</sub>O (mol/m<sup>3</sup>)
- $M_1$  is the Knock Induction Time Multiplier
- $M_2$  is the Activation Energy Multiplier
- $a_i, b_i, c_i, d_i, e_i$  and  $f_i$  are coefficients depending on the selected model (*Kinetics Fit*)

As reported in section 1.2.3.2, these expressions represent an empirical model developed by Gamma Technologies [2.4] that is used to predict knocking conditions: knock occurs when the integral defined by Equation (2.7) becomes equal to 1. Nevertheless, as widely described in chapter 1, such approach does not allow to rebuild the entire log-normal knock intensity (MAPO) distribution for given operating conditions. For this reason, the analytical knock model developed by the author and implemented in GT Power environment (through the User Knock Index template, introduced in section 1.3.2.3) is applied also for this investigation. Engine load defined as the Stoichiometric Trapped Air Mass (STAM) and the maximum in-cylinder pressure provided by the GT Power combustion model are used as inputs for the model. As a reference knock index, MAPO (Maximum Amplitude of Pressure Oscillation) has been chosen since it represents one of the most widely used,

and accepted, knock indicators [2.8, 2.9, 2.10]. Analytical knock model was previously calibrated and validated for 4-cylinder engine by using spark sweeps performed on engine points collected in Table 1.4 and 1.10, respectively.

Some validation results are reported in figures 2.6 and 2.7, showing a good agreement between modelled and experimental knock intensities. Figure 2.6 shows a light-medium knock operating condition, while Figure 2.7 reports conditions corresponding to higher spark advance angles, and therefore with higher knock intensity.

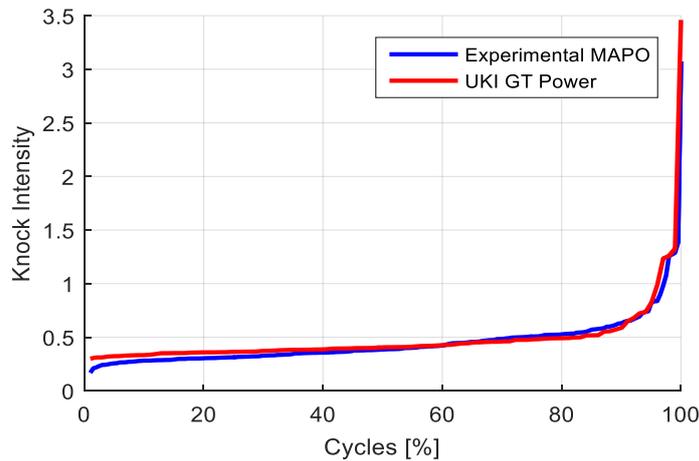


Figure 2.6: Comparison between experimental MAPO and predicted User Knock Index, for SA=15°CA BTDC.

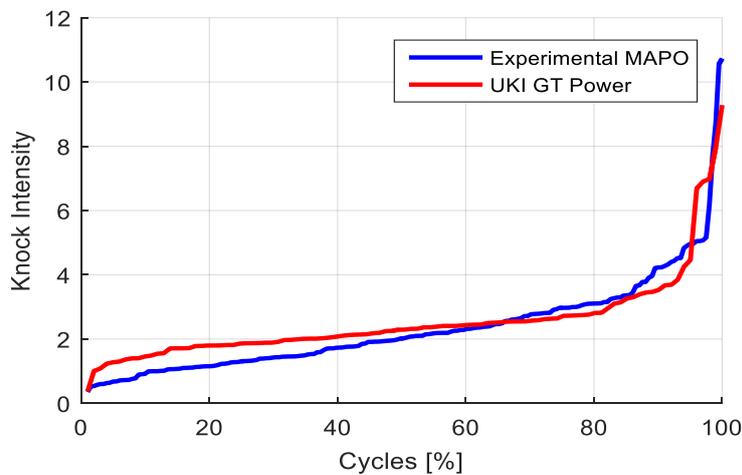


Figure 2.7: Comparison between experimental MAPO and predicted User Knock Index, for SA=21° CA BTDC.

Partly depending on the combustion model predictivity, this model should bring forward water injection potential on knock reduction.

#### 2.2.1.4. Modelling Results

Some simulations regarding SA and  $r$  sweeps have been run to analyze their combined effects on the combustion process, while keeping constant the following parameters:

- $F_d = 0.4$
- $ToE = 330^\circ CA$
- $wT = 80^\circ C$

This choice derives from a hypothesis made to consider values that well represent a possible experimental condition for water evaporation. During future activities, investigations about intake and in-cylinder evaporated water mass fraction will be carried out. For consistency, the operating point considered next is again 3000 rpm, 1.5 bar of intake manifold pressure, and stoichiometric mixture.

One of the aspects that should be considered to evaluate engine response to water injection, is the effect on combustion. As the mass of water increases, combustion should shift towards the exhaust stroke for the same spark advance, because of lower in-cylinder temperatures. Predicted values for CA50%MFB (angle corresponding to 50% of mass burnt) as a function of the applied Spark Advance (SA) are reported in Figure 2.8.

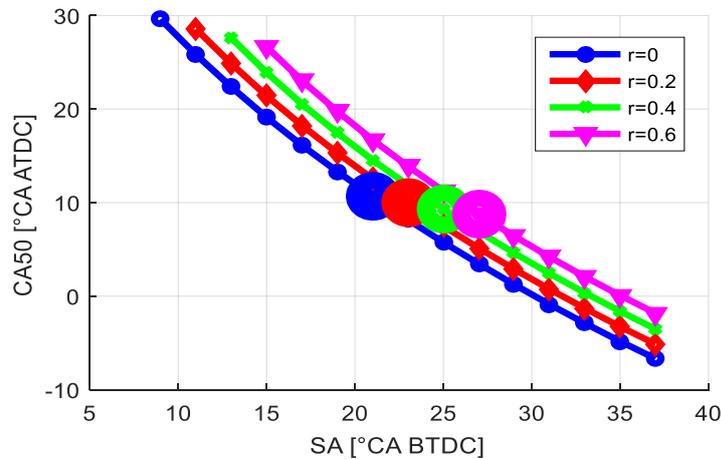


Figure 2.8: CA50%MFB as a function of applied SA, for different water mass fractions.

According to the model, as water mass increases, combustion slows down. This means that an additional effort is necessary for the spark advance calibration, or that an on-board closed-loop combustion control could be very convenient. Also, CA50%MFB corresponding to maximum efficiency condition is lower as water ratio increases (represented by large circular markers in Figure 2.8).

Water injection affects simulated IMEP curves. In particular, if injected water quantity increases, the IMEP peak decreases. This is caused by thermodynamic and indicated efficiencies reduction, associated with longer combustion durations, also considering that intake air mass flow is controlled at constant value. IMEP curves collected in Figure 2.9 show this phenomenon, and also the combustion slowdown discussed above.

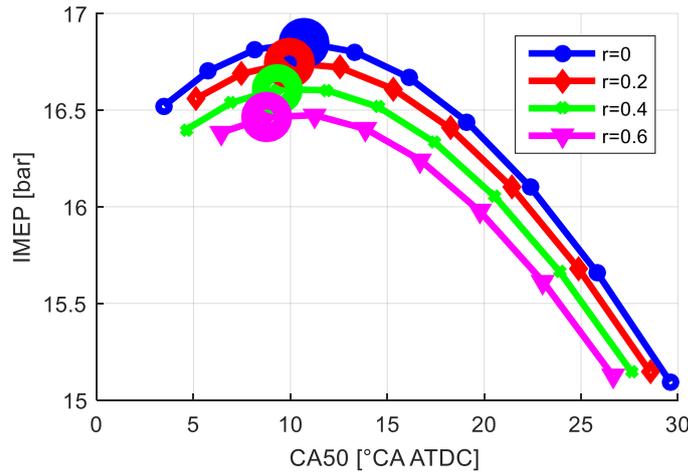


Figure 2.9: Simulated IMEP curves for different injected water quantities. Big circles highlight maximum IMEP points.

As described in the previous paragraph, water injection affects combustion phasing. Because of the exhaust temperature dependence on combustion phasing, water injection effect on such quantity should not be compared at fixed spark advance. Supposing the operating point is not knock limited, exhaust temperature difference/reduction should be compared under maximum efficiency conditions (which can be obtained with about the same CA50%MFB), for every water mass quantity, and consequently with different spark advance angles.

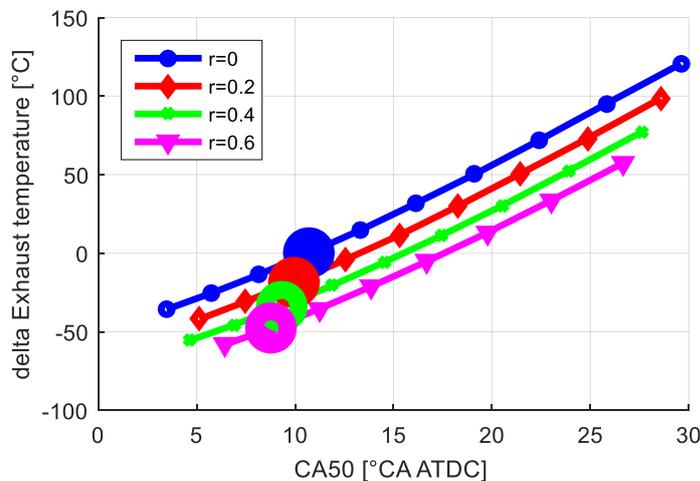


Figure 2.10: Modelled exhaust temperature reduction as a function of CA50%MFB and water ratio.

In Figure 2.10 modelled exhaust temperature reduction is reported as a function of CA50%MFB, for different values of  $r$ . The point considered as reference is the big blue dot in Figure 2.10, corresponding to the maximum efficiency spark advance for the case without injected water ( $r = 0$ ). For example, with 60% of injected water with respect to the stoichiometric fuel mass ( $r = 0.6$ ), a reduction of about 50°C is achievable when comparing maximum efficiency conditions.

Figure 2.11 reports UKI values calculated by the model, for different values of  $r$  and SA. The predicted knock reduction is substantial, even though, again, different curves should not be compared at fixed spark advance.

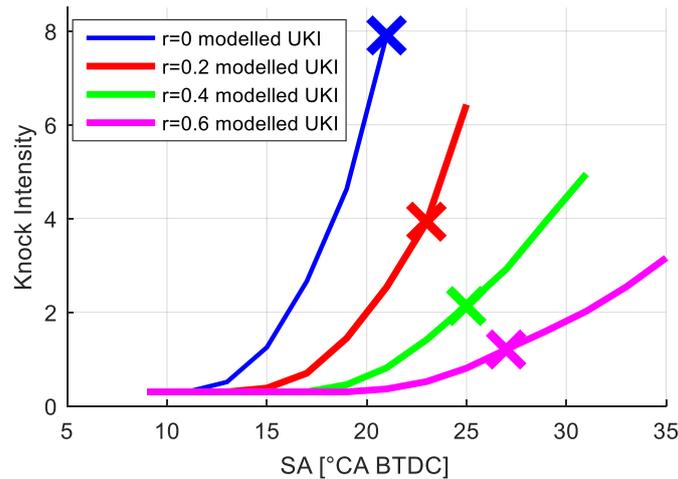


Figure 2.11: Modelled knock intensity as a function of Spark Advance and water ratio. Water injection heavily reduces knock intensity. Cross markers highlight maximum IMEP spark advance.

## 2.2.2. Experimental Tests and 1-D Model Validation

Experimental tests have been carried out on the 4-cylinder GDI TC engine (Table 2.1) whose intake system has been modified in order to install port water injectors and rail.

### 2.2.2.1. Experimental Setup

In Figure 2.12 the modified intake manifold water rail and injectors are visible. Water injectors location is as close as possible to the intake valves, according to the model indication about water evaporation dynamics. Prototype water injectors and pump were developed by Magneti Marelli and are controlled by a RCP (Rapid Control Prototyping) system developed by this research group in collaboration with Alma Automotive, controlling injection timing and rail pressure.



Figure 2.12: Detail of the experimental setup: modified intake manifold (red arrow) and water rail (green arrow) are visible. Water injectors are highlighted with blue circles.

Experimental data here presented have been obtained with constant injection pressure and timing, 10 bar and 360°CA ATDC (beginning of the intake stroke), respectively. Engine operating point is

the same: 3000 rpm, 1.5 bar of intake manifold pressure, and stoichiometric mixture. Additional information about operating conditions is collected in Table 2.2.

Table 2.2: Operating conditions.

r	In-Cylinder Air [mg/cycle/cylinder]	Fuel [mg/cycle/cylinder]	Water [mg/cycle/injector]
0	519.17	35.56	0
0.2	495.38	33.93	6.78
0.4	504.86	34.58	13.83
0.6	500.63	34.29	20.57

All cylinder pressure data have been acquired and analyzed, then the “mean” cylinder has been considered in all the figures below.

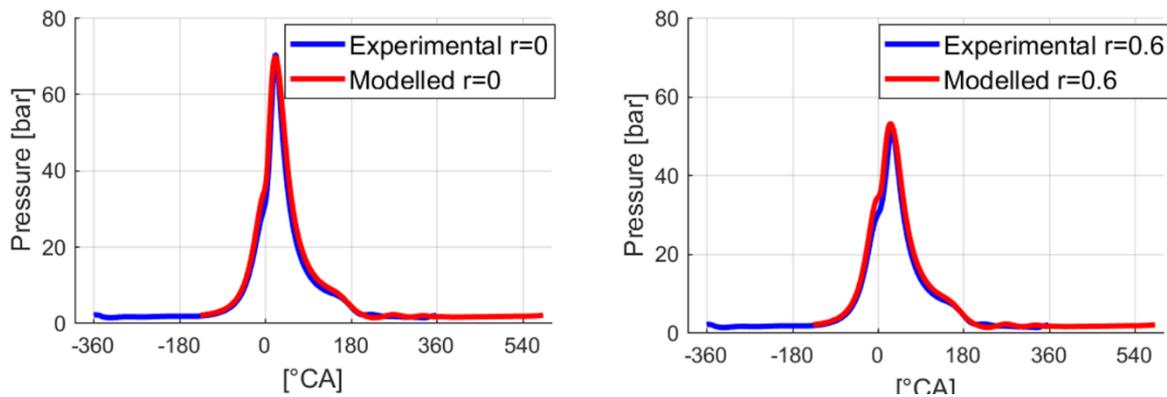


Figure 2.13: Modelled and measured in-cylinder pressure average cycle, for the same spark advance and different water quantities.

In Figure 2.13, modelled and experimental in-cylinder pressure profiles are reported, for two different quantities of injected water. The match between simulation and experimental data is very encouraging, in terms of water quantity variations effects on peak pressure value and position. Instead Figure 2.14 collects ROHR curves for the same two values of injected water.

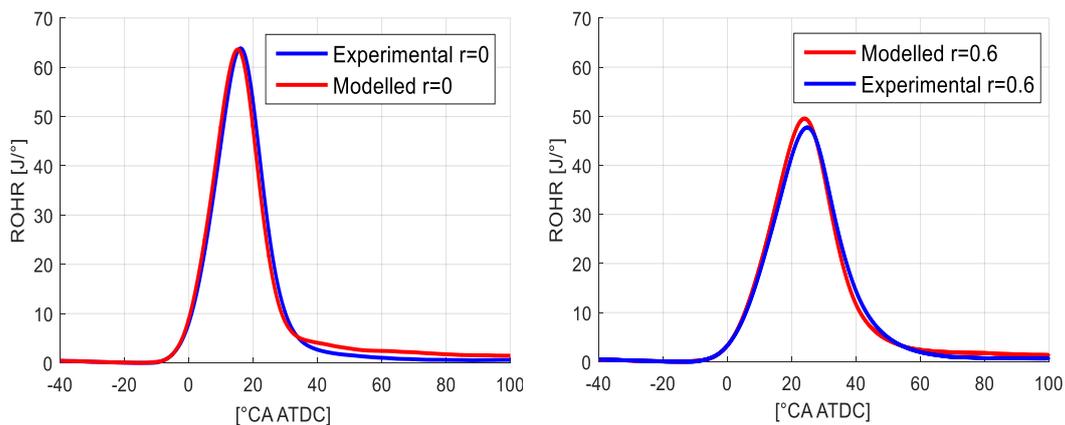


Figure 2.14: Modelled and measured ROHR average cycle, for the same spark advance and different water quantities.

Figure 2.15 confirms the significant water injection effects on combustion timing: results are very close to those reported in Figure 2.8, and again larger markers identify maximum efficiency points.

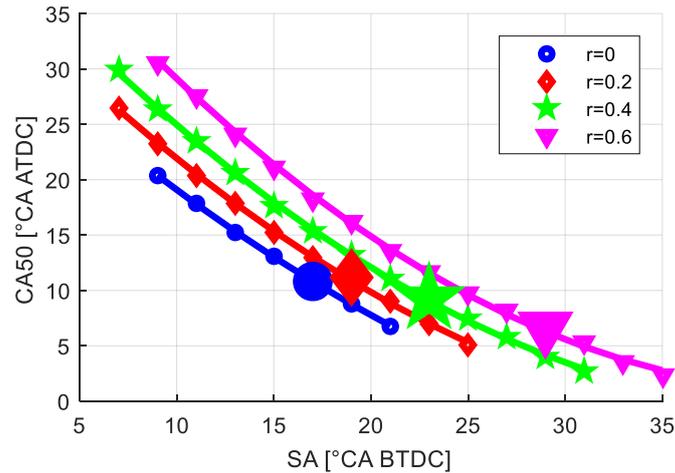


Figure 2.15: Effect of water injection on combustion phasing. Experimental results are superimposable to model predictions (Figure 2.8): water injection reduces combustion speed and the same CA50%MFB is reached with higher Spark Advance values, as water ratio increases

Another effect of the water injection, which can be seen also in the model results, is the shift towards lower values of CA50%MFB corresponding to maximum efficiency, as water quantity increases. In Figure 2.15 such value goes from about 11°CA ATDC for  $r = 0$  to about 6°CA ATDC for  $r = 0.6$ . There is some incongruence between Figure 2.15 and Figure 2.8: the model and the real engine return slightly different values of CA50%MFB for the same Spark Advance. Some further effort is required for the combustion model calibration.

Figure 2.16 shows IMEP (Indicated Mean Effective Pressure) values. It results a slight reduction in the maximum IMEP achievable as water is injected. This confirms the simulation results, even if the IMEP reduction is smaller and much less correlated to injected water quantity.

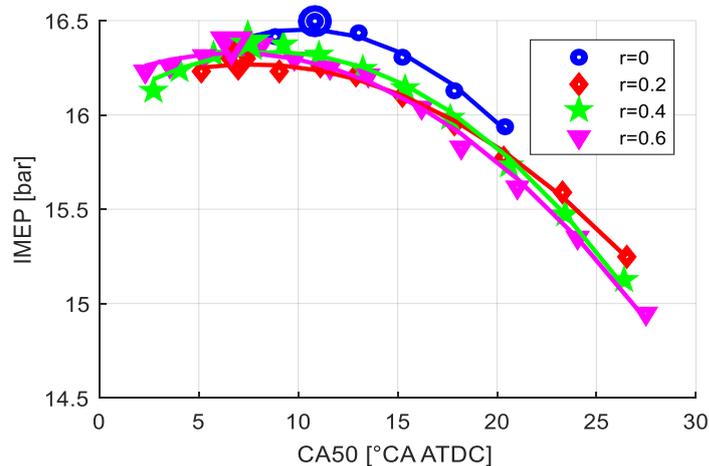


Figure 2.16: Measured IMEP as a function of CA50%MFB, for different water quantities. Larger markers identify maximum IMEP (and efficiency) points.

Exhaust temperature reduction, as underlined above, should be evaluated (at least in a first analysis) at maximum efficiency points (highlighted in Figure 2.17 by larger markers). In fact, because of water injection effect on combustion duration, exhaust temperature slightly increases as water mass increases, for the same spark advance angle.

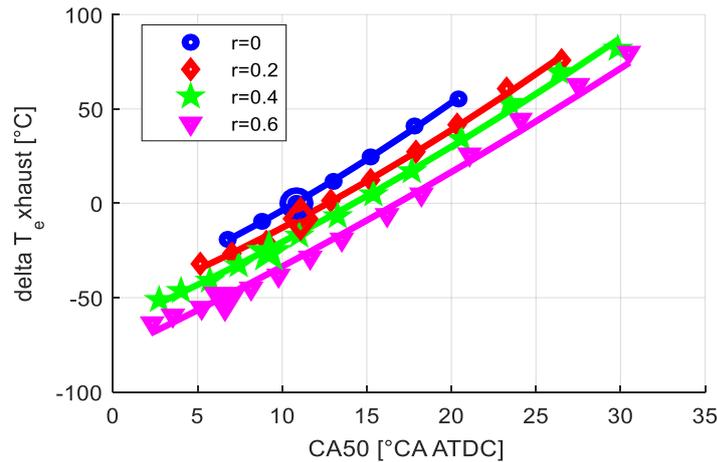


Figure 2.17: Measured exhaust temperature reduction, depending on water ratio. Bigger markers correspond to MBT. A reduction of about 50°C is achieved with 60% of water ratio.

Supposing the highlighted point for  $r = 0$  as knock limited (i.e., supposing knock intensity to be above the admissible threshold), the achievable gain in terms of exhaust temperature reduction is even greater than 50 °C.

Experimental and modelled knock intensities have been superimposed in Figure 2.18, for a direct comparison. What is particularly interesting is the knock intensity reduction for the maximum efficiency points (larger markers) as water mass increases. Inconsistently, for  $r = 0.2$ , knock intensity at maximum efficiency is greater than for  $r = 0.0$ . This is probably due to an ambiguous identification of the maximum efficiency spark advance value. The selected operating point (1500 rpm, 1,5 bar) is not considered to be knock limited but a slight reduction on knock intensity as injected water mass increases is however verified.

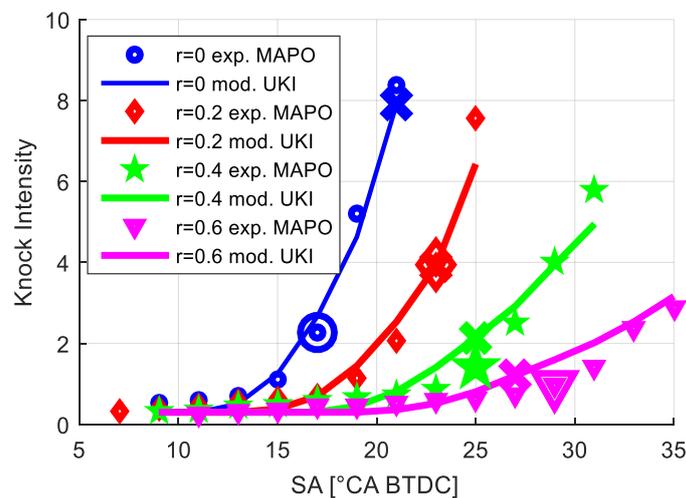


Figure 2.18: Comparison between measured and modelled knock intensity. Bigger markers correspond to MBT (crosses represent GT ones). Modelled and experimental knock intensities are very superimposable.

Last analysis is on Brake Specific Fuel Consumption (Figure 2.19). Basically, investigated water quantities do not affect engine efficiency, confirming that effectively water enables an extension of the engine operation, with no compromises.

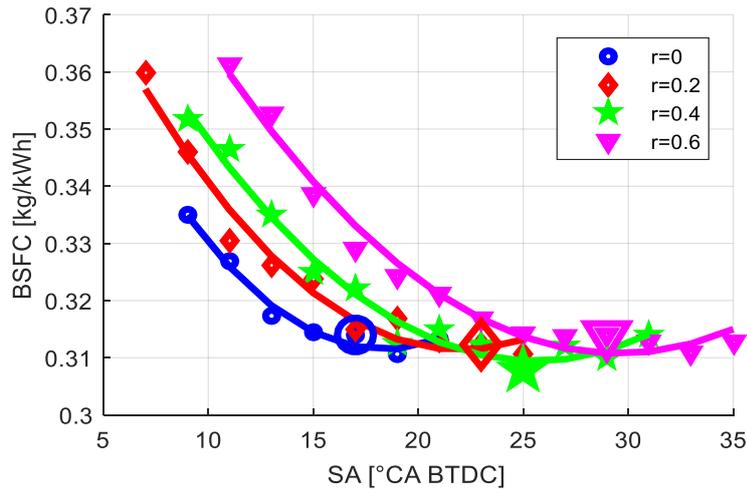


Figure 2.19: Measured Brake Specific Fuel Consumption trends for different water ratio values. Water injection does not affect maximum achievable efficiency: minimum BSFC value is independent on the water ratio.

### 2.2.3. Conclusions

This investigation, both based on model evaluation and by experimental observation, confirms the effectiveness of water injection as a solution to control/reduce exhaust gas temperature and knock tendency. Both these aspects comply with the aim of reducing fuel consumption under high load conditions: on one hand it is possible to reduce or avoid mixture enrichment to limit exhaust temperature, and at the same time water injection can enable operation with maximum efficiency Spark Advance angles, due to the shift of the KLSA (Knock Limited Spark Advance), thus further contributing to exhaust gas temperature reduction. The primary aim of this activity was confirming the feasibility of port water injection implementation as a non-native layout, and therefore with intrinsically limited control possibilities. First simulations show the influence of main model parameters on water injections benefits. In particular, effects of evaporation location and duration (by varying parameters  $F_i$ ,  $F_d$ ,  $ToE$ ) and water temperature have been investigated. Results confirm that major effects on knock tendency, and in-cylinder and exhaust temperature are obtainable with larger water mass fraction evaporating within the combustion chamber, and with longer evaporation duration. Instead, the model seems to be not sensitive to others water injection operating parameters (i.e. injection pressure and phasing), so their optimization cannot be executed on the model. The second part of this chapter is focused on experimental setup and WI test on bench. Matches between experimental and simulated results are presented and a good model predictivity is verified. The following tables report a numerical comparison between simulations (Table 2.3) and experimental tests (Table 2.4). The values of the most relevant parameters are compared during conditions corresponding to maximum efficiency Spark Advance. A relevant difference is tangible between simulated and experimental KI 98% values for the case without water injection. This is explainable observing in this case the knock intensity curve is very steep and a small difference on maximum efficiency SA involves relevant errors on KI 98% values. The same comparison made in terms of CA50 is less ambiguous. Therefore, this phenomenon becomes more marked due to SA step used for simulations and experimental tests, which is probably not small enough.

Table 2.3: Simulated characteristics quantities of maximum efficiency engine points.

r	SA	CA50	IMEP	Exhaust dT	KI 98%
0	21	11.25	16.84	0	7.90
0.2	23	9.95	16.74	-18.38	3.91
0.4	25	9.31	16.60	-33.97	2.14
0.6	27	8.78	16.47	-47.67	1.20

Table 2.4: Experimental characteristics quantities of maximum efficiency engine points.

r	SA	CA50	IMEP	Exhaust dT	KI 98%
0	17	10.84	16.49	0	2.26
0.2	19	10.85	16.29	-8.16	3.89
0.4	23	9.04	16.38	-25.66	1.41
0.6	29	6.62	16.38	-50.84	0.96

Exhaust temperature and knock intensity reduction achieved with 0.4 – 0.6 values of  $r$ , are probably obtainable with 0.2 – 0.4 mass fractions in an optimized condition and this is an extremely auspicious result. In fact, water ratio should be as low as possible, to reduce water tank capacity and, mainly, to reduce refill frequency.

## 2.3. Development and Software-in-the-Loop Validation of a Model-based Controller for a Port Water Injection System

A Water Injection based Combustion Controller (WICC) has been developed using a combined approach of experimental investigation and simulation environment:

- First, an experimental investigation has been carried out to investigate the effects of several values of  $r$  on combustion. Tests have been carried out over the most critical engine operating region in terms of knock tendency, applying SA sweeps for each  $r$  value
- Then, experimental data have been processed to define a WI Combustion Model (WICM) able to compute the optimal SA depending on MFB50 target and injected water mass. Three different approaches developed from analytical MFB50 model described in chapter 1 are presented and critically compared
- In the third stage, the WICM has been implemented within the Open Loop (OL) branch of an innovative WI Combustion Controller (WICC), which has been tested via Software in the Loop (SiL), coupling the algorithm to a one-dimensional engine simulator. The WICC has been proposed first with a Closed Loop on Knock Index, and then with an additional MFB50 Closed Loop contribution

The implementation of the two CL branches is motivated by the aim of exploring all the achievable benefits in presence of both Knock Index and MFB50 measurements. Both inputs have been obtained by processing in-cylinder pressure signals. The next step of this activity consists in substituting in-cylinder pressure trace with accelerometric signal treated as described in section 1.4.2 for the calculation of MAPO and MFB50 indexes.

### 2.3.1. *Experimental Campaign*

Tests have been again performed on the 4-cylinder GDI TC engine for which the GT Power model is developed and validated above.

The additional experimental campaign has been designed to explore the effects of WI in a wide operating engine field, focusing on the areas with high knock tendency. Therefore, a grid of operating points has been defined by 2 levels of loads and 4 engine speeds, as shown in Table 2.5. The load is expressed with the Net Load (NL) parameter defined below:

$$NL = pressure_{manifold} * \eta_{intake} \quad (2.10)$$

Where  $\eta_{intake}$  is a factor that takes into account pressure drop within intake runners and valves.

Table 2.5: Experimental grid operating points.

Engine point	Speed [RPM]	NL [bar]	Lambda	r
1	1500	0.86	1	0:0.2:0.8
2	1500	1.00	1	0:0.2:0.8
3	2500	1.20	1	0:0.2:0.8
4	2500	1.40	1	0:0.2:0.8
5	3500	1.45	1	0:0.2:0.8
6	3500	1.8	1	0:0.2:0.8
7	4500	1.30	1	0:0.2:0.8
8	4500	1.50	1	0:0.2:0.8

For each operating point, the same investigation methodology has been carried out. It consists in the execution of a specific SA sweep at different  $r$  values that are applied in ascending sequence from 0 to 0.8, with incremental steps of 0.2. The angular SA step used in every sweep depends on Knock Limited Spark Advance (KLSA), and near Knock Limit Spark Advance (nKLSA). Such indexes define the SA angle that causes a knock tendency close to the safety threshold (KLSA) and close to 60% of the same threshold (nKLSA). The knock intensity is quantified as the 98<sup>th</sup> percentile of MAPO. It is defined as the value below which 98 percent of MAPO index values are contained, in a sorted set of consecutive combustions. The threshold for this index is defined according to Equation 2.11:

$$MAPO_{98_{threshold}} = \frac{RPM}{1500} \quad (2.11)$$

The first element ( $SA_{first}$ ) of each SA sweep is defined by Equation 2.12, unless the limit on maximum exhaust gas temperature is exceeded:

$$SA_{first} = SAbase + \frac{3 * r}{0.2} \quad (2.12)$$

In which:

- $SAbase$  is the calibration value of SA
- The 3 CA offset has been arbitrarily assumed as the average value of combustion delay related to a step of 0.2  $r$  [2.1]

In conclusion the SA sweeps are carried out with the following specifications:

1. Angular steps of 3CA from  $SA_{first}$  to nKLSA
2. Angular steps of 1 CA from nKLSA to KLSA

This methodology has been conceived to obtain a wide vision of WI effects on MFB50 and, at the same time, to investigate with more accuracy all combustion indexes near the KLSA area.

### 2.3.2. WI Combustion Model

In this section, the analytical MFB50 model presented in 1.3.3.1 has been further developed to introduce the sensitivity to the injected water mass and three different methods for its implementation are proposed and compared. Experimental data has been processed, point by point, to analyze the relationship between SA and MFB50, for each tested  $r$  value. As demonstrated in chapter 1, for a single spark sweep, the most appropriate fitting function to analytically describe the relationship between SA and MFB50 is the quadratic polynomial, as clearly shown in Figure 2.13. Such figure displays some spark sweeps for different values of parameter  $r$ , highlighting the influence of such parameter on the trend that could be identified in the absence of water injection.

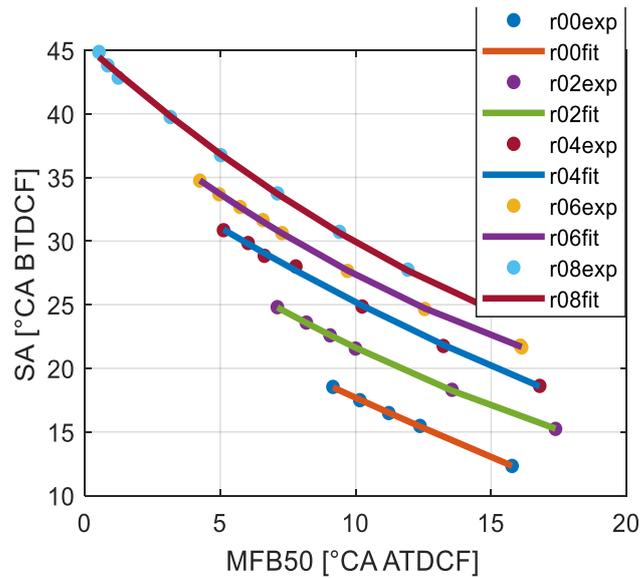


Figure 2.20: Parabolic fitting of the SA trend with respect to MFB50, for the engine point characterized by  $NL=1.2$  and  $RPM=2500$ , and for different water-to-fuel mass ratios. It is an example of the parabolic trend that links the SA to the MFB50 for fixed operating conditions.

The following equation defines the polynomial fitting of the SA on MFB50 domain.

$$SA = a MFB50^2 + b MFB50 + c \quad (2.13)$$

This analysis allows to conceive a WI Combustion Model that processes the RPM, NL,  $r$  and the target of MFB50 as inputs, to compute the corresponding SA to be applied. The base concept is to calculate the parameters  $a$ ,  $b$  and  $c$  of the parabolic function and investigate the analytical dependence of each parameter from the  $r$  value. Three different methods have been designed to build such model, and the respective performances have been evaluated by comparing experimental and modeled SA. The fitting quality has been quantified for each proposed method by evaluating the correlation coefficient.

#### 2.3.2.1. Polynomial Method

Through the parabolic fitting of MFB50 and SA data for each engine point and for each value of  $r$ , the three coefficients  $a$ ,  $b$ ,  $c$ , (called Parabolic Coefficients), have been identified. Each parameter has been fitted with a polynomial function of RPM and NL (Net Load), for each value of  $r$ . The resulting equation for  $a$ ,  $b$  and  $c$  is the following:

$$PCoeff = p_{00} + RPM * p_{10} + NL * p_{01} + + RPM * NL * p_{11} + RPM^2 * p_{20} \quad (2.14)$$

Where  $p_{00}, p_{10}, p_{01}, p_{11}, p_{20}$ , are called Surface Coefficients. In Figure 2.21 are shown all the resulting surfaces, for  $r = 0$ .

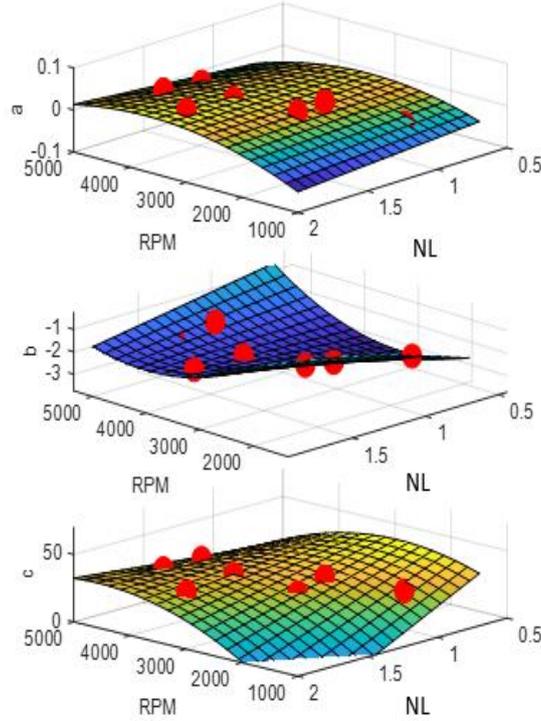


Figure 2.21: Fitting surfaces of the parabolic coefficients in (RPM, NL) domain, for  $r=0$ .

Every single surface is described by a set of 5 Surface Coefficients  $p_{xx}$  and each coefficient features a quite constant slope in  $r$  domain as displayed in Figure 2.22, so it can be fitted with a linear function, as shown below:

$$p_{xx} = o_{xx} + r * g_{xx} \quad (2.15)$$

Where  $o_{xx}$  is the constant term and  $g_{xx}$  the curve slope.

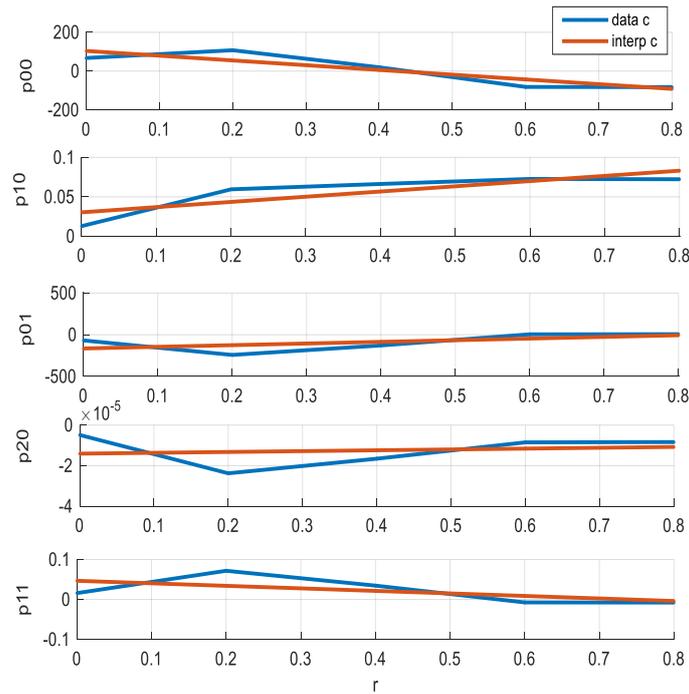


Figure 2.22: Fitting of surface coefficients by linear function.

Figure 2.23 shows a block diagram of the model based on polynomial fitting, relating the SA to the MFB50 target, for different combinations of  $r$ , RPM, and NL (i.e., the model inputs).

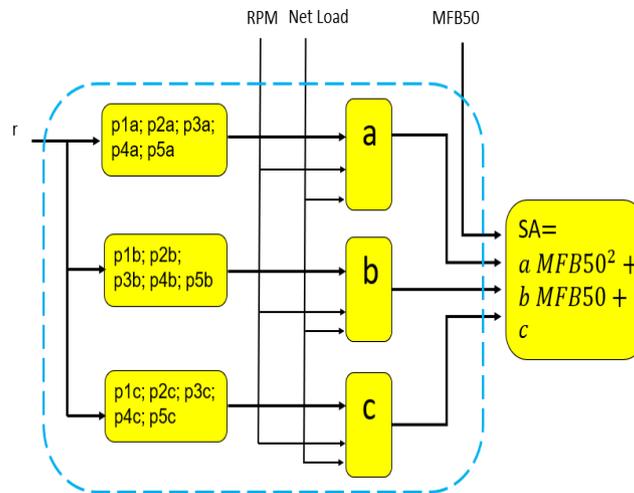


Figure 2.23: Combustion Model with Polynomial Method.

### 2.3.2.2. Effect Separation Method 1-D

To verify the possibility of reducing the computational load and the allocated memory, other two implementation methods of the MFB50-SA open-loop combustion model have been investigated. The first one is the simplest and fastest one, where the effects of water-to-fuel ratio, described by the parameter  $r$ , are considered to be independent of speed and load. With this method, the mean surface for  $r=0.5$  of parameters  $a$ ,  $b$  and  $c$  has been calculated and it has been defined as the reference between all surfaces obtained for the different  $r$  values (Figure 2.24). Every coefficient has then been

identified for all values of parameter  $r$ , for each engine point. Such values have been normalized with respect to the value which corresponds to  $r=0.5$ . The final trend is then described by the mean of the normalized values. In this way, such curve represents the gain of the related parameter which adapts the reference parameter value when  $r$  is different from 0.5. The three normalized curves are discretized as arrays called  $Ka$ ,  $Kb$ ,  $Kc$ , with  $r$  as input. In Figure 2.25 the trend of  $Ka$  for each engine point and the mean curve are shown.

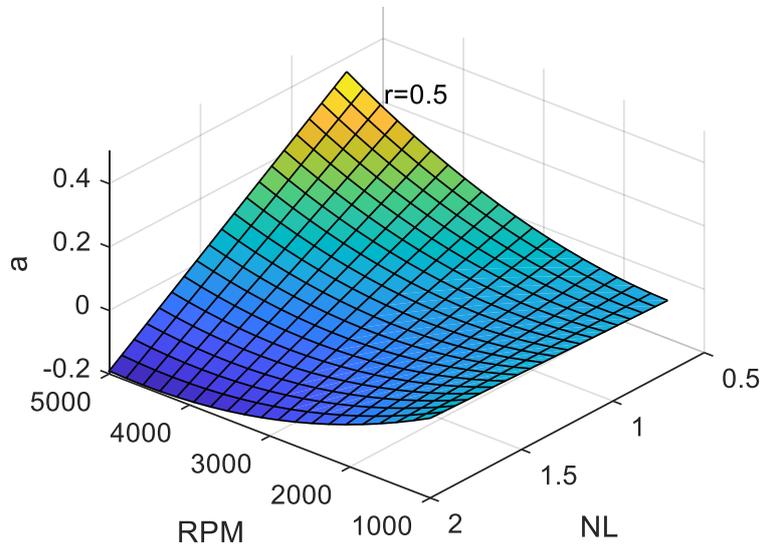


Figure 2.24: The calculated surface of parameter  $a$ , for  $r=0.5$ .

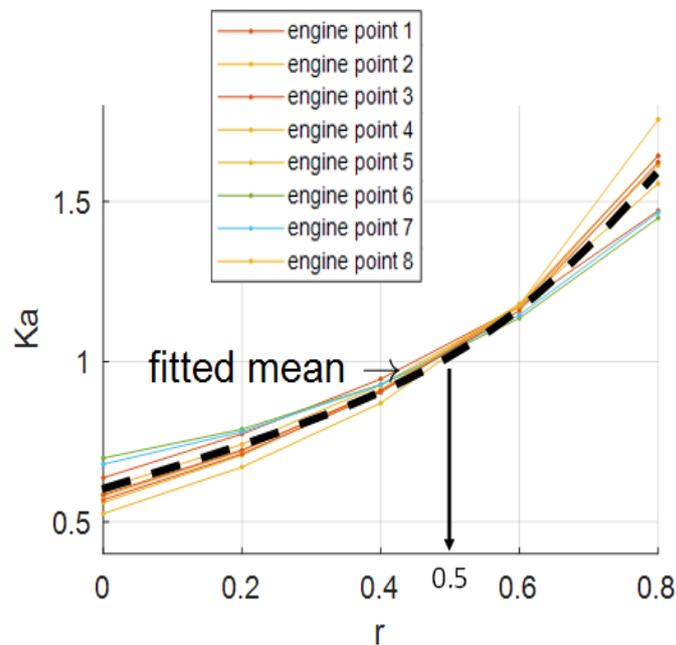


Figure 2.25: Normalized curve  $Ka$ .

In other words, the final approximated coefficient value can be recovered by multiplying the gain identified by the injected water mass  $r$  with the respective value of the same parameter at  $r=0.5$ , which depends on the engine point (RPM, NL). Figure 2.26 shows a block diagram of the model based on full mono-dimensional effects separation.

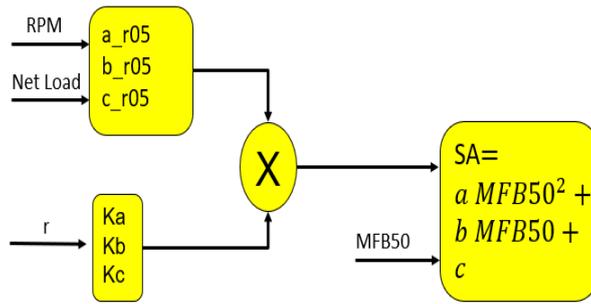


Figure 2.26: Combustion model with Separation Effect 1-D.

### 2.3.2.3. Effect Separation Method 2-D

In this case, a degree of complexity is added to the previous model, by mapping the “correction” factors  $Ka$ ,  $Kb$  and  $Kc$  as two-dimensional functions, depending on  $r$  and on another influent parameter, chosen between NL and RPM. To identify the most influencing factor, every coefficient has been displayed through two different representations, for each engine point. First, as surfaces on  $r$ -RPM domain, and second, as surfaces on  $r$ -NL domain. For both representations, a normalization over the 0.5  $r$  value has been investigated, and the mean of all normalized surfaces has been computed and displayed. The comparison between the two representations highlights that NL has more influence on surface gradient than RPM (Figure 2.27). Thus, in this method the gain has been replaced by a 2-D matrix with NL and  $r$  as inputs. Figure 2.28 shows a block diagram of the model based on partial effects separation.

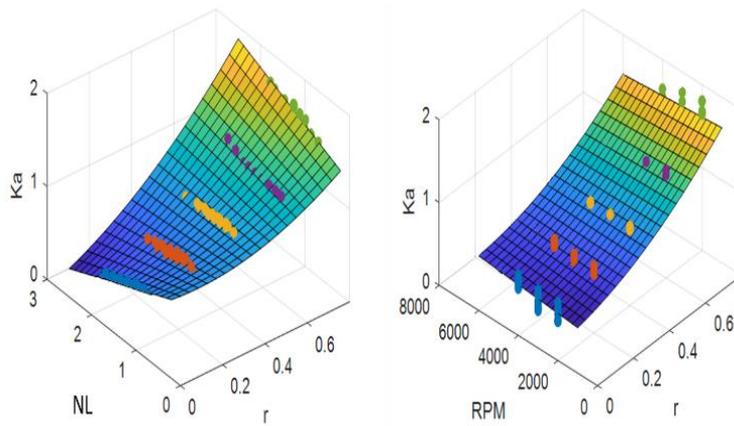


Figure 2.27: Average surface  $Ka$  on NL- $r$  domain (left) and on RPM- $r$  domain (right).

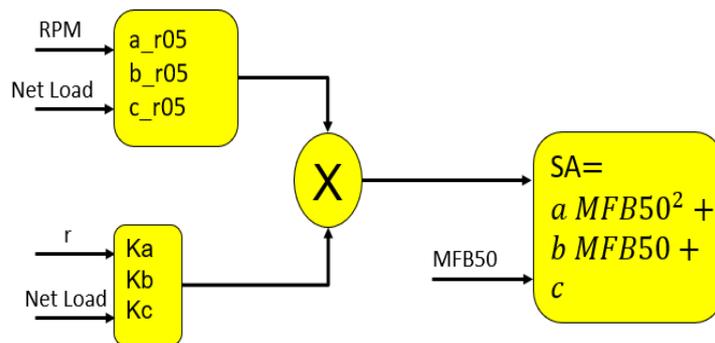


Figure 2.28: Combustion model with Separation Effect 2-D.

### 2.3.2.4. Comparison by Correlation Coefficient

To evaluate the accuracy of each method, the relative Correlation Coefficients  $\rho$  have been computed, as defined by Equation 2.16:

$$\rho(A, B) = \frac{1}{N-1} * \sum_{i=1}^N \left( \frac{A_i - \mu_A}{\sigma_A} \right) * \left( \frac{B_i - \mu_B}{\sigma_B} \right) \quad (2.16)$$

Where:

- A is the array of the computed SA values by the different methods
- B is the array of experimental SA values
- $\mu_A$  and  $\sigma_A$  are the mean and standard deviation of A
- $\mu_B$  and  $\sigma_B$  are the mean and standard deviation of B

As shown in Table 2.6, the polynomial method produces the highest  $\rho$  value. Thus, this approach has been selected for control implementation. Even because in this phase of the project the computational burden of the controller has not been considered as a hard constraint.

Table 2.6: Correlation Coefficient values for the three different methods used to define the combustion model. The green box highlights the best result and the corresponding method.

Method	Polynomial	Eff. Sep. 2-D	Eff. Sep. 1-D
$\rho$	0.99	0.97	0.91

### 2.3.3. WI-based Combustion Control

The definition of a combustion model able to compute the SA required to reach the MFB50 target at a specific  $r$ , allows designing a WI combustion controller oriented to knock mitigation at high loads, while keeping MFB50 at its optimum value. The WI-based Combustion Controller (WICC) has been designed with both Open Loop (OL) and Closed Loop (CL) branches. The OL requires as inputs the water-to-fuel mass ratio ( $r$ ) and the MFB50 target. For this reason, two look-up tables (one for  $r$  and one for MFB50) have been defined. The CL operates to maintain the measured MAPO98 close to the threshold, with a statistical approach (MAPO98 is the 98<sup>th</sup> percentile of the MAPO distribution considered in the given buffer of MAPO values). A second version of the CL controller has also been developed. Such algorithm is able to control also the MFB50 by closing the loop on its measurement, to overcome errors of the combustion model and to meet the target. The performance of the two algorithms is evaluated simulating several steady state engine points and several transient conditions. The results quality is then highlighted with the Root Mean Squared Error (RMSE) between the mean MFB50 and the corresponding target. The same parameter has been used for the MAPO98.

#### 2.3.3.1. CL on MAPO98

The developed OL branch requires a MFB50 target map and a  $r$  map, both based on RPM and NL (Net Load), to provide the necessary inputs for the combustion model. The first one provides the MFB50 target angle, and the second a  $r$  value, calibrated with the same methodology for all the investigated experimental points. The methodology consists of three steps. At first, the required SA

angles to target the optimum MFB50 (arbitrarily fixed to 8°CA ATDCF) are computed by processing the related parabolic function for all injected water masses, for each experimental point (Figure 2.29):

$$SA_{opt} = a MFB50_{opt}^2 + b MFB50_{opt} + c \quad (2.17)$$

Where:

- $a$ ,  $b$  and  $c$  are the coefficients of the parabolic function  $SA=f(CA)$ , for a  $r$  value and for a fixed engine point
- $SA_{opt}$  is the spark angle which guarantees the  $MFB50_{opt}$

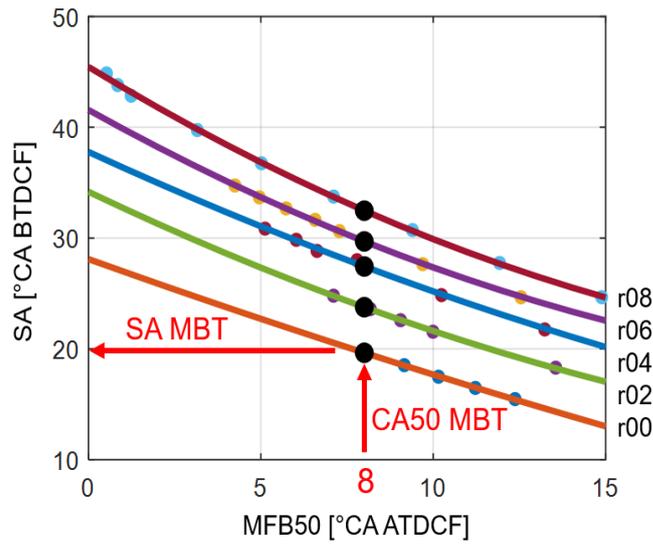


Figure 2.29: SA Maximum Brake Torque (MBT) determination procedure.

Then the MAPO98 values corresponding to optimum SA are determined by evaluating the following exponential function, one for each  $r$  value (colored curves in Figure 2.21):

$$MAPO98_{opt} = g SA_{opt}^f + k \quad (2.18)$$

Where:

- $g$ ,  $f$  and  $k$  are the parameters of the exponential function  $MAPO98=f(SA)$ , for a  $r$  value and for a fixed engine point
- $MAPO98_{opt}$  is the MAPO98 value obtained for the  $MFB50_{opt}$

At last, optimal MAPO98 values are fitted with another parabolic function on  $r$  domain (red curve in Figure 2.30), and through intersection with MAPO98 threshold (KL, in Figure 2.30), it is possible to evaluate the minimum  $r$  value that guarantees a permissible MAPO98 (equal to 0.2 in the example shown in Figure 2.30, where the corresponding SA MBT is also highlighted):

$$r_{map} = m MAPO98_{thr}^2 + n MAPO98_{thr} + l \quad (2.19)$$

Where:

- $m$ ,  $n$  and  $l$  are the parameters of the parabolic function  $r=f(MAPO98_{opt})$ , for each engine point
- $r_{map}$  is the minimum  $r$  value which allows respecting the  $MAPO98_{thr}$

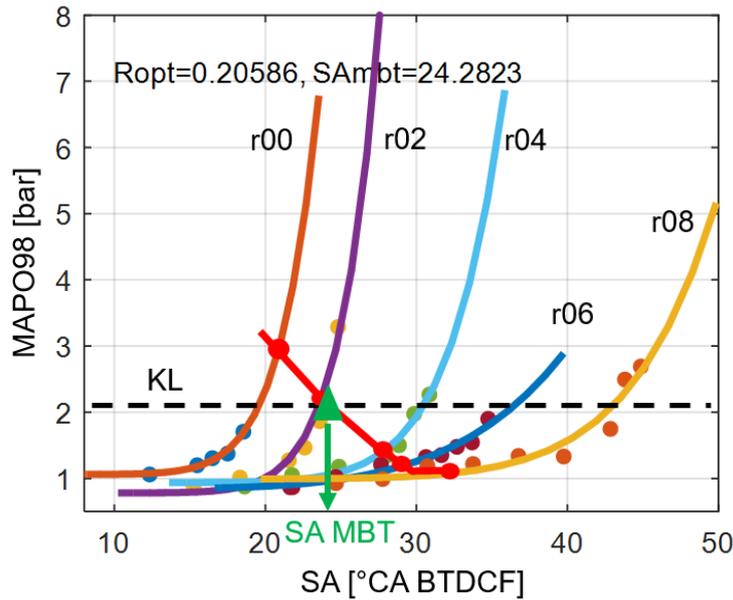


Figure 2.30: R map determination and related SA MBT.

When this value saturates at maximum  $r$  value allowed (arbitrarily fixed at 0.8), a spark retard is necessary to guarantee knock reduction. Thus, a new SA angle must be defined in the intersection between the exponential MAPO98 function calculated for  $r=0.8$  and knock threshold (Figure 2.31, where SA ACT does not correspond to SA MBT):

$$SA_{act} = e^{\frac{\log \frac{MAPO98_{thr} - k_{08}}{g_{08}}}{f_{08}}} \quad (2.20)$$

Where:

- $g_{08}$ ,  $f_{08}$  and  $k_{08}$  are the parameters of the exponential function  $MAPO98=f(SA)$  for  $r=0.8$ , for a specific engine point
- $SA_{act}$  is the spark advance actuated to obtain a MAPO98 equal to the threshold

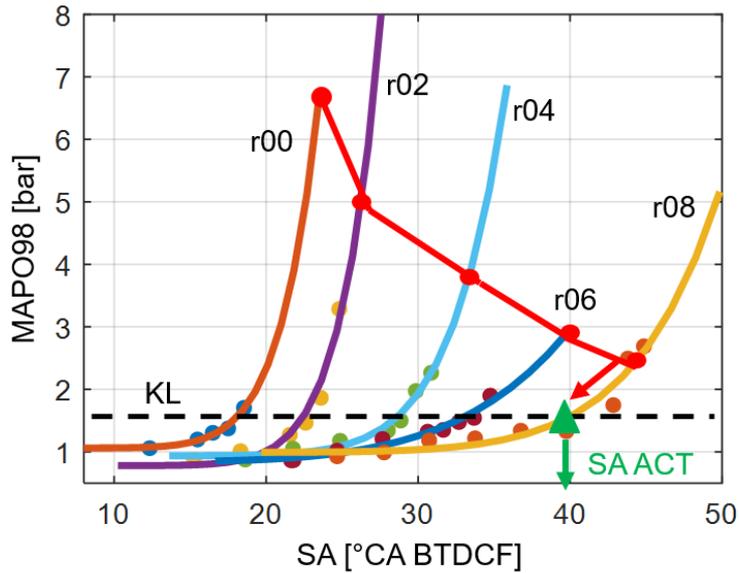


Figure 2.31: SA determination, when  $r$  is saturated to 0.8.

As a result, the MFB50 map provides the optimum value (equal to 8CA ATDCF) when  $r < 0.8$  and a delayed value when  $r = 0.8$ . The experimental data demonstrate that this circumstance is never verified, and the parameter  $r$  does not saturate to the maximum value. Consequently, the MFB50 target is equal to 8 on the entire operating field. Moreover, the experimental tests have been carried out at the highest load for the tested engine speeds. Thus, for the operating field characterized by lower load levels, the choice of a MFB50 target equal to 8 is legitimate. The resulting  $r$  map for the mean cylinder is shown in Figure 2.32. The consequent MFB50 target map within the explored range and for lower loads becomes a constant equal to 8. Outside the operative field explored during the experimental campaign, the maps trends have been linearly extrapolated.

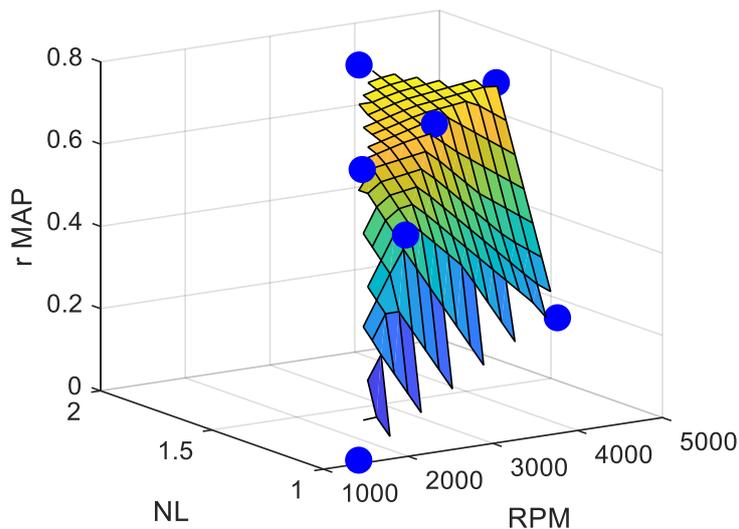


Figure 2.32: R map for the explored operative field.

The controller is also composed by a CL contribution (Figure 2.33). The first version of such closed-loop controller manages the MAPO98, computed from MAPO measured cycle by cycle with in-cylinder pressure signal, by applying corrections to  $r$  and SA outputs. Basically, the implemented

algorithm is based on a Proportional and Integral (PI) control system, which has been evolved to freely split the entire correction between the two levers on which the controller can act. The error between the measured MAPO98 and the corresponding threshold is translated into a proportional and integral correction through a gain scheduling PI structure. The resulting sum of two contributions is then converted in a percentage value which represents the total amount of the correction (Total Percentage Correction, TPC) which is required by the system for that specific operating condition. Such correction is then converted into  $r$  and SA variations, with a pre-defined logic. In this way, the TPC allows also to define a common reference to manage the two available actuators to control knocking events. When a  $r$  correction is applied, the OL chain compensates the final  $r$  value with the SA calculated by the Combustion Model, since the  $r$  correction is considered as input to such model (Figure 2.33). If the water-to-fuel ratio ( $r$ ) reaches its saturation value (arbitrarily defined), to further reduce knocking a SA correction is also applied by the CL controller. The controller management of  $r$  and SA correction depends only on calibrations. Moreover, a protection strategy reacts proportionally to single MAPO events on the value of the TPC. Complete controller scheme is shown in Figures 2.33.

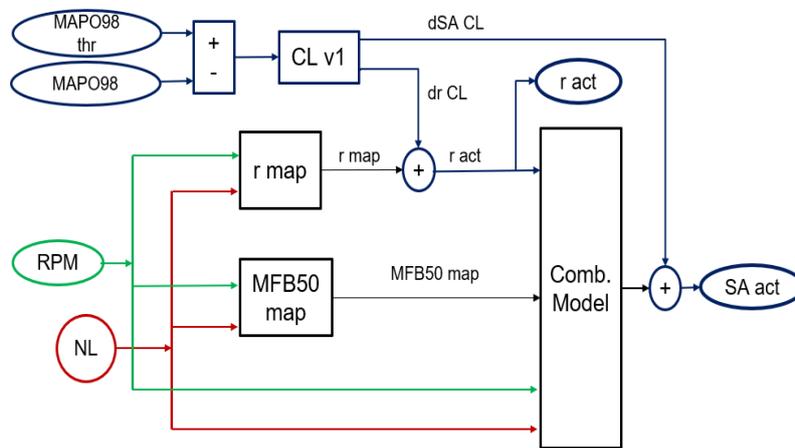


Figure 2.33: First version WICC layout.

### 2.3.3.2. CL on MAPO98 and MFB50

The CL chain has been further developed and it has been integrated with an algorithm which evaluates the SA corrections needed to respect the MFB50 target. In other words, the second version of the Closed Loop branch can calculate the SA variations also to reduce the error between the measured MFB50 (obtained from in-cylinder pressure signal) and the target one, which is the output of the map implemented within the OL chain. The MFB50 values are filtered with a moving average, to avoid the signal oscillations caused by the Cycle-to Cycle Variation (CCV) and to prevent a consequent unstable controller behavior. The idea is to create a second TPC related to MFB50 error (MFB50TPC), which defines the SA correction only when the engine works in not-knocking or light-knocking conditions. In particular, the MFB50TPC cannot be increased when the TPC related to the MAPO98 error is over the level which activates SA correction for knock mitigation. With such approach the final SA correction is always a sum of two contributions, and the definition of the actuation priority between the two CL controllers is not required. The new CL scheme allows to manage the SA variations, avoiding the overlap between opposite contributions which derive from the two different Total Percentage Corrections. When the TPC is increasingly close to the value which is fixed as the threshold which discriminates between acceptable and not-acceptable knock

level, the maximum MFB50TPC is progressively reduced. So, with an appropriate calibration of the vector which defines the SA correction with respect to MFB50TPC, it is then possible to have two controllers that act corrections on spark advance (the first based on MAPO98 error and the second on MFB50 error) without conflicts. The complete layout of the second version of the Water Injection based Combustion Control is shown in Figure 2.34.

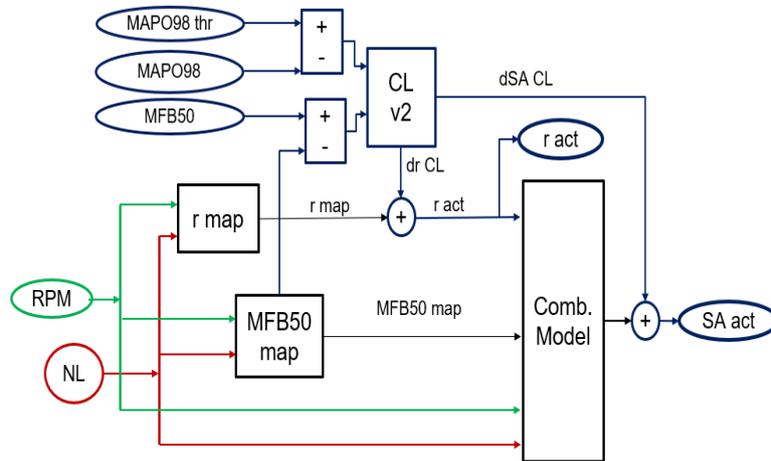


Figure 2.34: Second version WICC layout.

### 2.3.4. Software in the Loop Control Validation

The two versions of the WICC have been tested and validated via Software in the Loop (SiL) simulations. The controller has been coupled with a mono-dimensional mean-cylinder engine simulator which includes the WI, the CCV and a knock model. Such simulator consists in the GT Power model of the experimental setup developed and validated in the previous section of this chapter, in which the combustion model has been calibrated via Three Pressure Analysis (TPA). The GT model has been also converted in a Fast-Running Model (FRM), to test the controller in Real-Time (RT). Nine different engine points have been simulated, covering almost the entire boosted operating field. Of course, the lowest loads points have been neglected, because of the absence of knocking events and the consequent controller inactivity. The quality of the simulations results is defined through the R-squared index. Moreover, the simulations with the first version of the controller (that with the CL only on the MFB50 target), are also a validation of the reliability of the Combustion Model, because the accuracy on the MFB50 target is ensured only by the OL chain.

#### 2.3.4.1. GT Power Fast Running Model

In order to debug the controller algorithm, the WICC has been coupled with the GT Power model of the mean cylinder of the 4-cylinders. The predictive combustion model (CombSITurb) has been previously calibrated and validated and the analytical knock model developed by the author was introduced with a user defined template (UKI). For the validation of the WICC such GT model has been converted in an FRM. This procedure consists in the reduction of engine model complexity, mainly through the combination of single ducts in parts with larger volumes. On the contrary, the combustion model and the intake and exhaust valves and ports are not modified, and this allows to maintain the predictive capabilities of the detailed model. While the accuracy of some ducts signals

(like pressure and flow rate waves) is partially lost, the entire conversion process can be driven by the tolerance (imposed by the user) that has to be maintained for some combustion indexes (like maximum in-cylinder pressure, MFB50, etc.). In this way, the detailed model can be easily adapted for RT simulations [2.4]. Also, the WI modelling has been modified to be in accordance with the new ducts' configuration. As accurately described above, the WI system has been modelled with two injectors, where the first is a Port Water Injector (PWI) and the second is a Direct Water Injector (DWI). Through the calibration of parameters which define how the injected water mass is split between such injectors, the water vapor quantity, and the angular duration of in-cylinder water evaporation can be correctly reproduced. Due to the new engine model layout, the WI previous configuration has been replaced with a single DWI layout and the parameters values have been replaced with a new calibration set. Figure 2.35 shows the layout of the FRM.

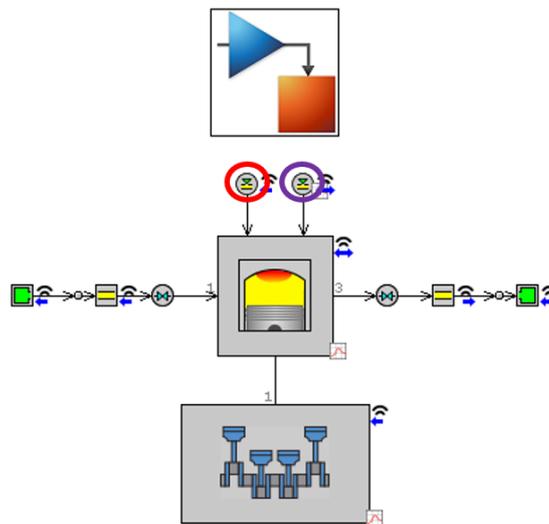


Figure 2.35: FRM layout. The red circle highlights the Direct Water Injector, the violet one highlights the fuel one.

### 2.3.4.2. Simulation Results

The FRM has been appropriately compiled and it has been consequently implemented in a 0-D co-simulation environment. The first simulations allow to identify the best calibration parameters set for the CL controller. They have been carried out for different engine points, under steady state and transient conditions. During such simulations, the PI controller applies  $r$  corrections which are then added to the map value, and CL contribution is not saturated above a specific value. This means the final  $r$  can assume too high values. Of course, this is not representative of a real application, in which the maximum  $r$  is certainly much lower, but such strategy allows studying the PI behavior on  $r$  and SA corrections also for high load conditions. It is important to accurately calibrate the CL parameters on the entire operating field.

In Figures 2.36, 2.37, 2.38 and 2.39 the results of simulations with the first version of the Water Injection based Combustion Control (characterized by the CL on MAPO98) are collected, for two different engine points. In Figure 2.37 and 2.38 it has been simulated a medium NL condition and the CL works only with  $r$  corrections. In Figure 2.39 a high load engine point has been simulated and during the firsts 50 cycles it is possible to highlight the protection action with a SA decrement, due to a high MAPO cycle. For small errors on MAPO98 the PI manages the water mass and for high errors also introduces SA variations. The  $r$  corrections can also have a negative sign, to reduce the mapped water mass when the recorded MAPO98 is lower than the threshold (Figure 2.36). The CL

parameters set has been chosen to guarantee a good correction stability and, at the same time, fast responses during transients. Figure 2.40 shows a transient simulation and during the rising ramp the CL can manage the knock intensity only with  $r$  corrections. In figures 2.37 through 2.40 the error between the mean MFB50 and the map value (the target) is due to inaccuracies of the Water Injection based Combustion Model used in the open-loop controller, which are quantified with the Root Mean Squared Error. Table 2.7 collects the RMSE committed by the controller on the MFB50 and MAPO98 targets for each simulation, because they represent the indexes which allow to quantify the controller robustness. The errors are evaluated excluding the firsts cycles, due to the MAPO98 and MFB50 buffers which are filling and do not produce coherent values.

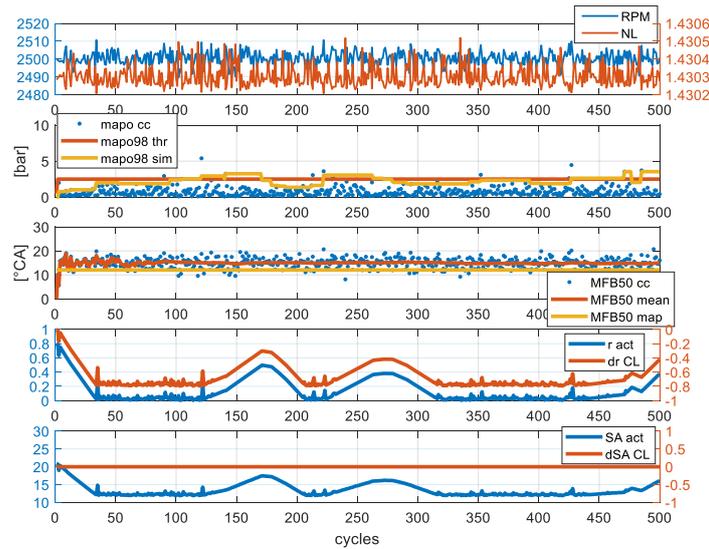


Figure 2.36: Steady state simulation results for engine point 2500 RPM, NL 1.43. Five subplots show (from top to bottom): RPM-NL, cycle-to-cycle recorded MAPO, MAPO98 and the MAPO98 threshold, cycle-to-cycle recorded MFB50, mean MFB50 and the corresponding target, the actuated  $r$  and the  $r$  correction calculated by CL chain, the actuated SA and the SA correction calculated by the CL chain.

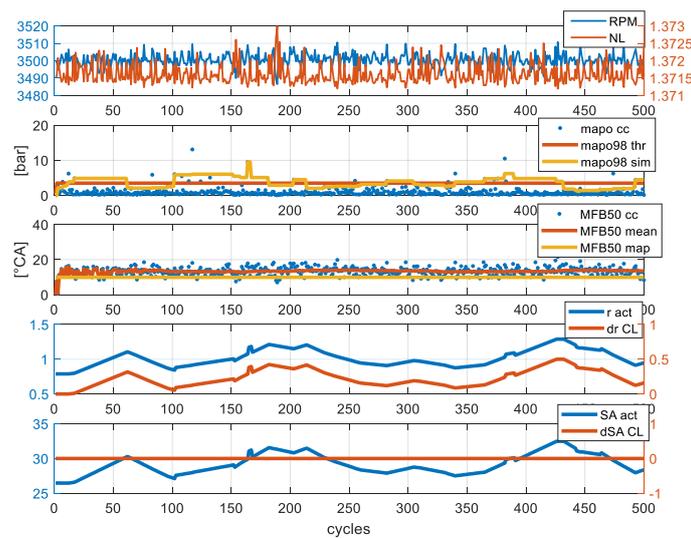


Figure 2.37: Steady state simulation results for engine point 3500 RPM, NL 1.37. Five subplots show (from top to bottom): RPM-NL cycle-to-cycle recorded MAPO, MAPO98 and the MAPO98 threshold, cycle-to-cycle recorded MFB50, mean MFB50 and the corresponding target, the actuated  $r$  and the  $r$  correction calculated by CL chain, the actuated SA and the SA correction calculated by the CL chain.

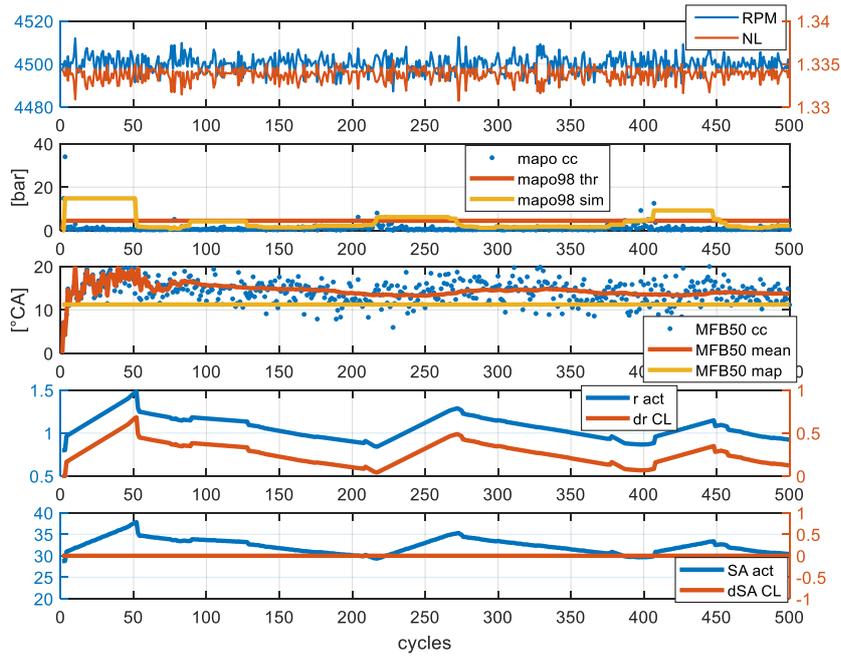


Figure 2.38: Steady state simulation results for engine point 4500 RPM, NL 1.33. Five subplots show (from top to bottom): RPM-NL, cycle-to-cycle recorded MAPO, MAPO98 and the MAPO98 threshold, cycle-to-cycle recorded MFB50, mean MFB50 and the corresponding target, the actuated  $r$  and the  $r$  correction calculated by CL chain, the actuated SA and the SA correction calculated by the CL chain.

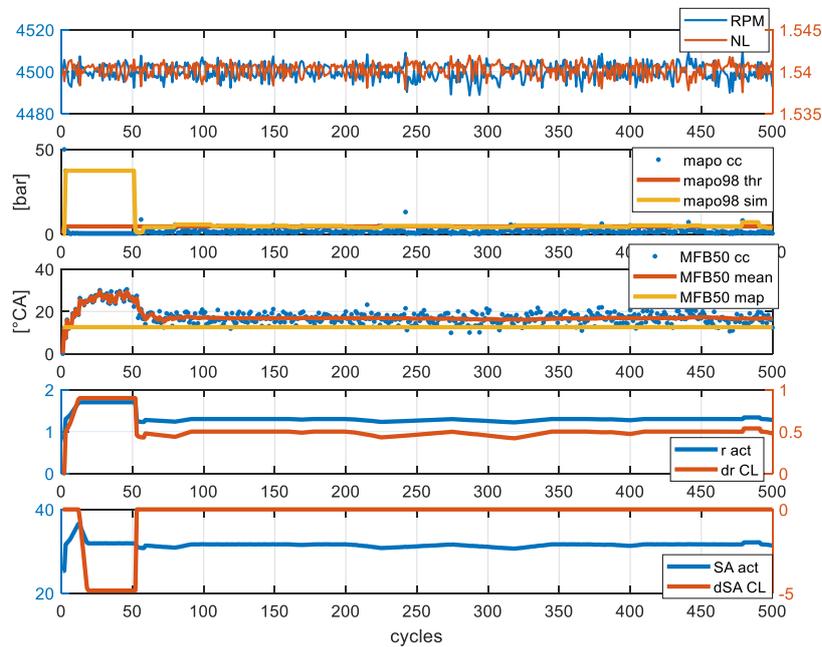


Figure 2.39: Steady state simulation results for engine point 4500 RPM, NL 1.54. Five subplots show (from top to bottom): RPM-NL cycle-to-cycle recorded MAPO, MAPO98 and the MAPO98 threshold, cycle-to-cycle recorded MFB50, mean MFB50 and the corresponding target, the actuated  $r$  and the  $r$  correction calculated by CL chain, the actuated SA and the SA correction calculated by the CL chain.

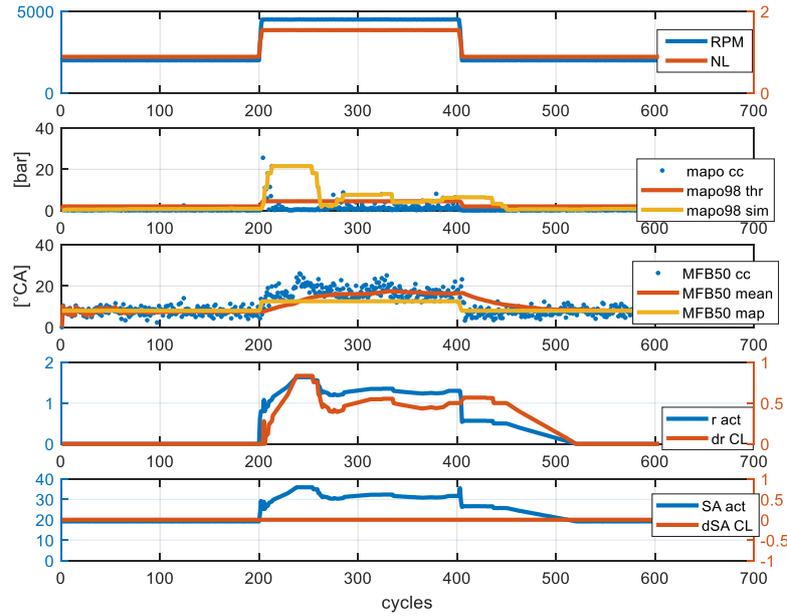


Figure 2.40: Transient simulation results. Five subplots show (from top to bottom): RPM-NL, cycle-to-cycle recorded MAPO, MAPO98 and the MAPO98 threshold, cycle-to-cycle recorded MFB50, mean MFB50 and the corresponding target, the actuated  $r$  and the  $r$  correction calculated by CL chain, the actuated SA and the SA correction calculated by the CL chain.

Table 2.7: Root Mean Squared Errors for the validation simulations with the first version of the WI based Combustion Control. The mean value of the RMSE on mean (by moving average) MFB50 represents the accuracy of the Combustion Model.

Engine Point	RMSE MFB50 [°CA]	RMSE MAPO98 [bar]
RPM 2500 NL 1.43	1.81	0.55
RPM 3500 NL 1.37	2.55	1.24
RPM 4500 NL 1.33	2.10	0.63
RPM 4500 NL 1.54	1.61	0.81
RPM 2000-4500-2000 NL 0.8-1.7-0.8	1.75	4.02
Mean	1.96	1.45

Results of simulations with the second version of WICC (characterized by CL on MAPO98 and MFB50 target) are shown in Figures 2.41, 2.42, 2.43 and 2.44. The CL on MAPO98 parameters values does not change, but during these simulations the final SA correction is the sum of two contributions (from MAPO98 and MFB50 CL controllers). The positive SA corrections are accepted only when the TPC is under a predetermined value. In Figure 2.43 it is possible to highlight negative and quick SA corrections in correspondence with a cycle characterized by a high MAPO level. In fact, in such simulation, the high MAPO value produces a high lower saturation of TPC, which is translated in an upper saturation of MFB50TPC to manage measured knock levels. This system avoids also on-off CL responses. In both figures it can be clearly seen that the MFB50 target is reached by the control system, thanks to the closed loop corrections evaluated by the MFB50 CL controller. The Figure 2.45 shows the transient simulation result during which the controller is able to maintain the mean

MFB50 on the corresponding target. The CL calibration parameters have been set on values which guarantees fast responses of the controller to have quite fast responses also in few simulated cycles, at the expense of the best stability. The Table 2.8 collects the RMSE for all simulations. The values highlight the reduction of the error on the MFB50 target.

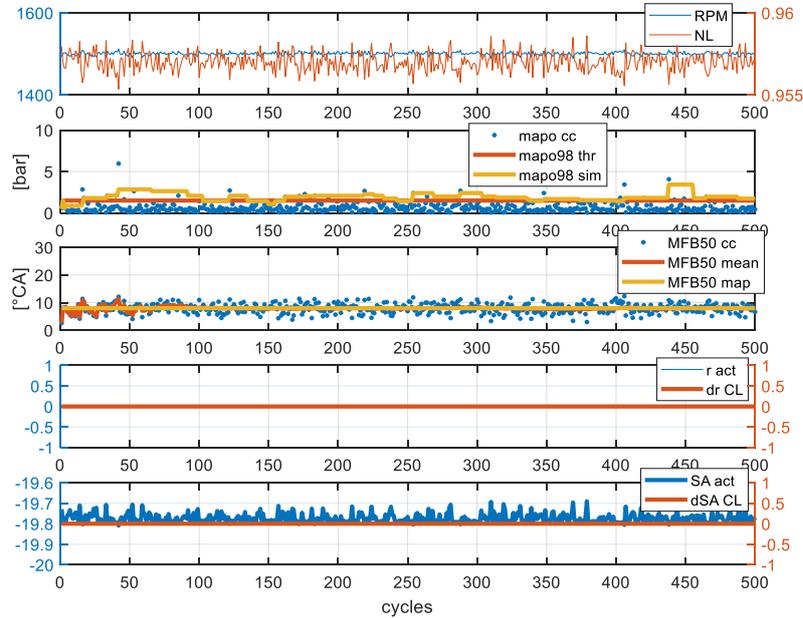


Figure 2.41: Steady state simulation results for engine point 1500 RPM, NL 0.95 with the CL on MFB50. Five subplots show (from top to bottom): cycle-to-cycle recorded MAPO, MAPO98 and the MAPO98 threshold, cycle-to-cycle recorded MFB50, mean MFB50 and the corresponding target, the actuated  $r$  and the  $r$  correction calculated by CL chain, the actuated SA and the SA correction calculated by the CL chain.

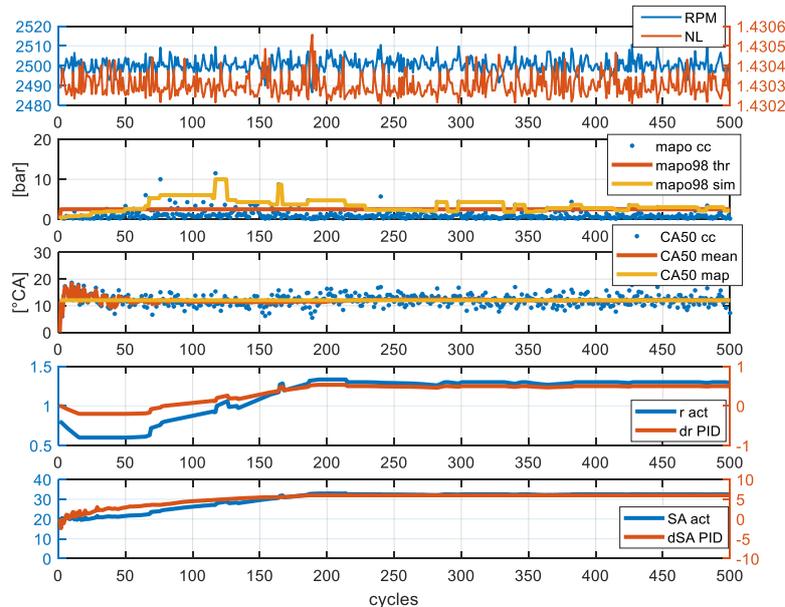


Figure 2.42: Steady state simulation results for engine point 2500 RPM, NL 1.43 with the CL on MFB50. Five subplots show (from top to bottom): cycle-to-cycle recorded MAPO, MAPO98 and the MAPO98 threshold, cycle-to-cycle recorded MFB50, mean MFB50 and the corresponding target, the actuated  $r$  and the  $r$  correction calculated by CL chain, the actuated SA and the SA correction calculated by the CL chain.

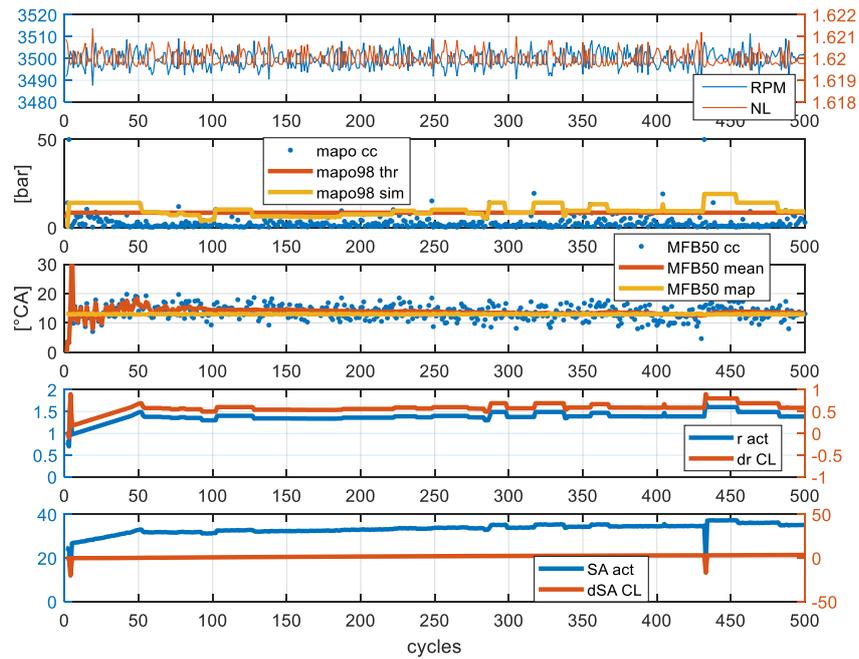


Figure 2.43: Steady state simulation results for engine point 3500 RPM, NL 1.62 with the CL on MFB50. Five subplots show (from top to bottom): RPM-NL, cycle-to-cycle recorded MAPO, MAPO98 and the MAPO98 threshold, cycle-to-cycle recorded MFB50, mean MFB50 and the corresponding target, the actuated  $r$  and the  $r$  correction calculated by CL chain, the actuated SA and the SA correction calculated by the CL chain.

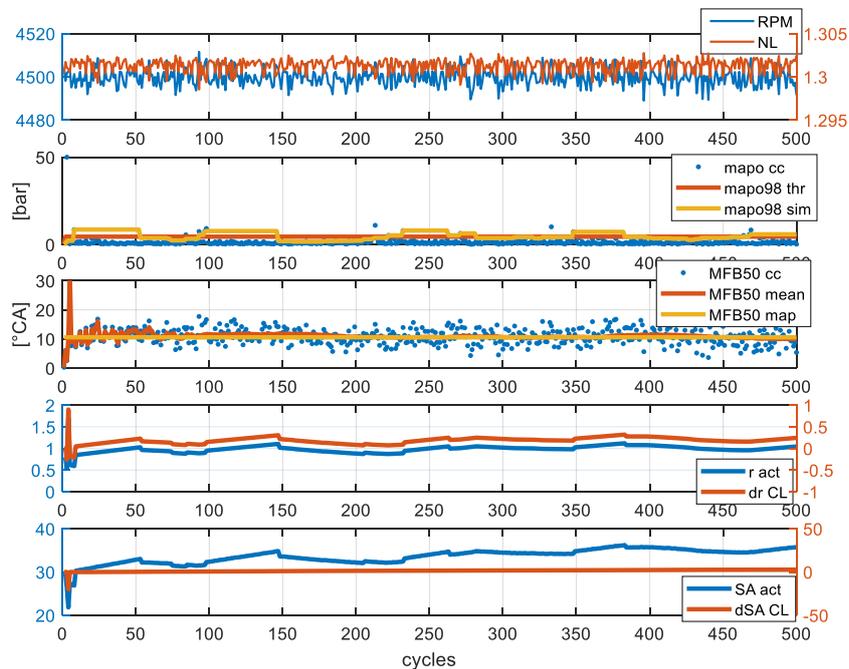


Figure 2.44: Steady state simulation results for engine point 4500 RPM, NL 1.3 with the CL on MFB50. Five subplots show (from top to bottom): RPM-NL cycle-to-cycle recorded MAPO, MAPO98 and the MAPO98 threshold, cycle-to-cycle recorded MFB50, mean MFB50 and the corresponding target, the actuated  $r$  and the  $r$  correction calculated by CL chain, the actuated SA and the SA correction calculated by the CL chain.

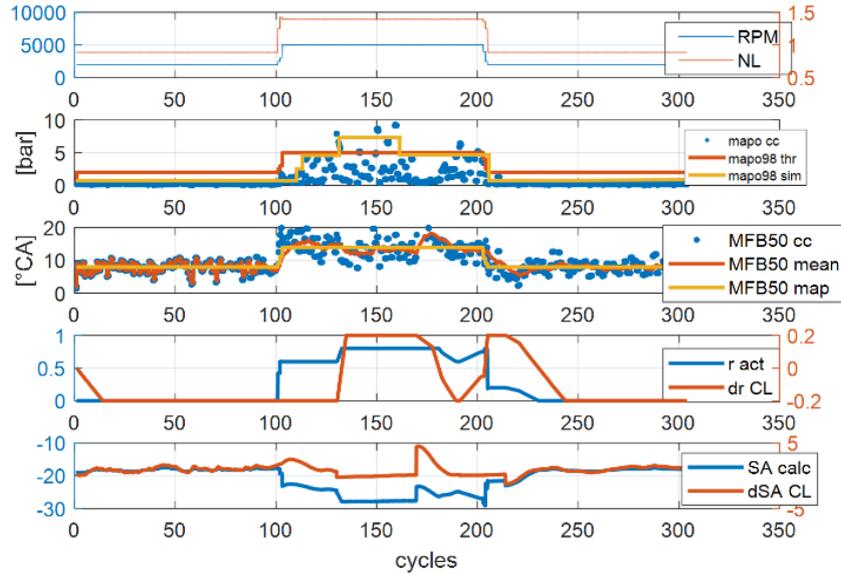


Figure 2.45: Transient simulation results with the CL on MFB50. Five subplots show (from top to bottom): RPM-NL, cycle-to-cycle recorded MAPO, MAPO98 and the MAPO98 threshold, cycle-to-cycle recorded MFB50, mean MFB50 and the corresponding target, the actuated  $r$  and the  $r$  correction calculated by CL chain, the actuated SA and the SA correction calculated by the CL chain.

Table 2.8: Root Mean Squared Errors for the validation simulations with the second version of the WI based Combustion Control.

Engine Point	RMSE MFB50 [°CA]	RMSE MAPO98 [bar]
RPM 1500 NL 0.957	0.17	0.87
RPM 2500 NL 1.43	0.32	1.89
RPM 3500 NL 1.62	0.62	1.04
RPM 4500 NL 1.3	0.45	0.93
RPM 2000-4500-2000 NL 0.88-1.49-0.88	0.88	2.34
Mean	0.49	1.41

### 2.3.5. Conclusions

In this chapter an innovative WI based combustion controller has been described. The proposed controller has been designed from the ground up and it is characterized by a model-based open loop part and a closed loop branch, which manages the injected water mass and the SA to control the knock level and the combustion phasing.

In the first part the analytical MFB50 model presented in chapter 1 is developed to include also the sensitivity to the mentioned relationship from the injected water mass changes. Such model is then implemented in the controller and it composes its OL branch and this is the main innovative contribution of such work. Three possible methods to implement such model in a 0-D environment

are presented and compared. The polynomial approach is then chosen, thanks to its greater accuracy.

The control strategy is then thoroughly described. In particular, the algorithm utilized to define the MFB50 and  $r$  maps of the OL controller and the two CL systems have been presented. The CL branch is a PI-based strategy with a gain scheduling and a dynamic saturation on the TPC. The first version of the controller has a CL part that controls only the MAPO98 through the corrections on parameters  $r$  and SA. The second version is characterized by a CL controller with an additional branch that introduces  $r$  and SA corrections to achieve a certain MFB50 target. The MFB50TPC has been introduced to efficiently manage the two contributions that define the final SA correction. In this way, the errors committed by the combustion model can be corrected by the PI controller.

In the last part the Water Injection based Combustion Control is coupled with a previously calibrated 1-D predictive combustion model and validated via SiL. The mean-cylinder combustion model has been converted in a Fast Running Model and implemented in a 0-D co-simulation environment. The two versions of the controller have been tested in RT, both under steady-state and transient conditions. The controller performance has been then evaluated with the RMSE for both controller versions, for each simulated engine point. The errors on the MFB50 target with the first controller are also a validation of the reliability of the proposed Combustion Model. Results demonstrate the robustness and the accuracy of the Water Injection based Combustion Control.

In the next section of this chapter the WICC is finally implemented in the RCP machine developed by the author (see the Appendix for details) and results of experimental validation at the engine test cell are shown.

## 2.4. Experimental Validation of Water Injection Control System

In this phase of WI activity, the performance of WICC is investigated by an experimental validation on a real engine using the Rapid Control Prototyping (RCP) platform. At first, a complete description of experimental set-up is presented with a focus on RCP system. Then a recap of WICC is provided. The experimental campaign is described in two different stages, a calibration, and a validation phase. The second one is shown in detail to discuss the capabilities of the controller in real conditions.

### 2.4.1. Experimental Setup

#### 2.4.1.1. Prototypal WI engine Setup

The calibration of the combustion model and its test bench on-line validation has been performed on the same 4-cylinder GDI TC engine with a port water injection (PWI) system, displayed in Figure 2.46.

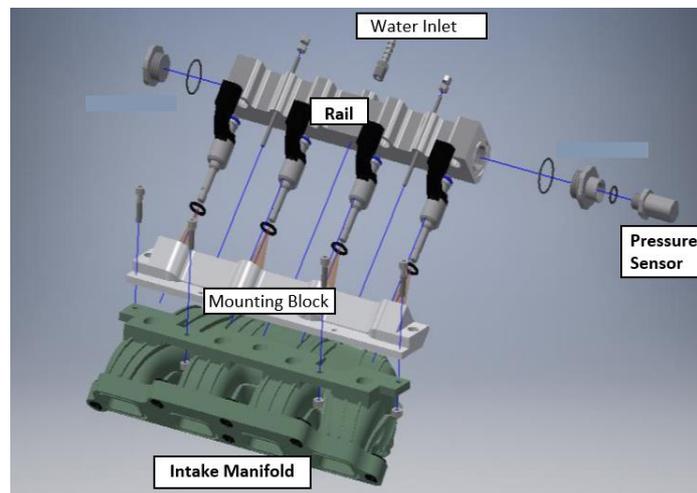


Figure 2.46: Port Water Injection System

#### 2.4.1.2. Rapid Control Prototyping Environment

A Rapid Control Prototyping platform (RCP) has been developed to manage the WI system. A synthetic scheme is displayed in Figure 2.47. During all tests, standard actuators of the engine have been managed by a development ECU. To manage the water injectors a dedicated WI Driver has been developed by a Real-Time hardware described in the Appendix coupled with the ECU. The basic algorithm designed for RT machine was initially able only to compute the necessary water injection time to obtain the externally set  $r$  value (Figure 2.48). To accomplish this goal, the hardware needs as inputs the values of RPM, fuel mass injected cycle by cycle and manifold pressure, all provided by ECU via CAN, water rail pressure, provided by dedicated sensor, and desired  $r$  value, provided by test bench system. The algorithm operates cycle by cycle for each cylinder through three steps. At first, the water mass to be injected is defined as a product of fuel mass injected and  $r$ . Then, water mass and pressure drop between water rail and manifold are used in the injector characteristic to define injection duration. At last, using the engine toothed wheel as a reference, phasing signal and a fixed Start of Injection angle (set the test-bench management system), the injection timing is

calculated and sent as a digital signal to the D/A Converter that finally transfers an analogic signal to each injector.

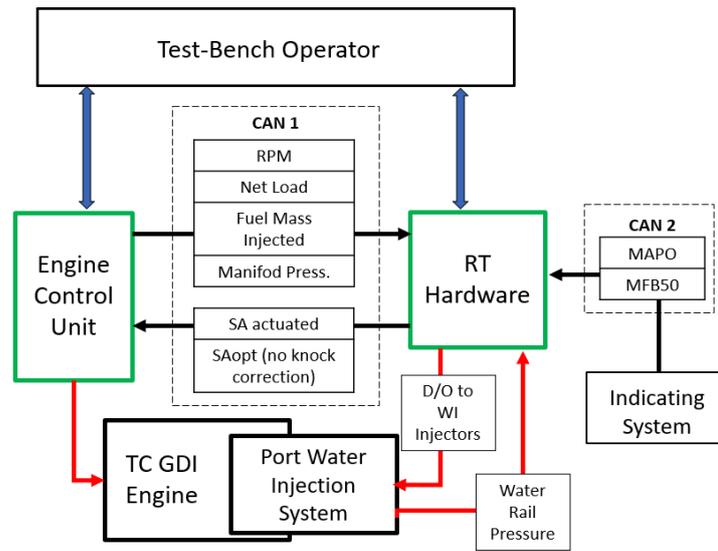


Figure 2.47: Layout of Rapid Control Prototyping.

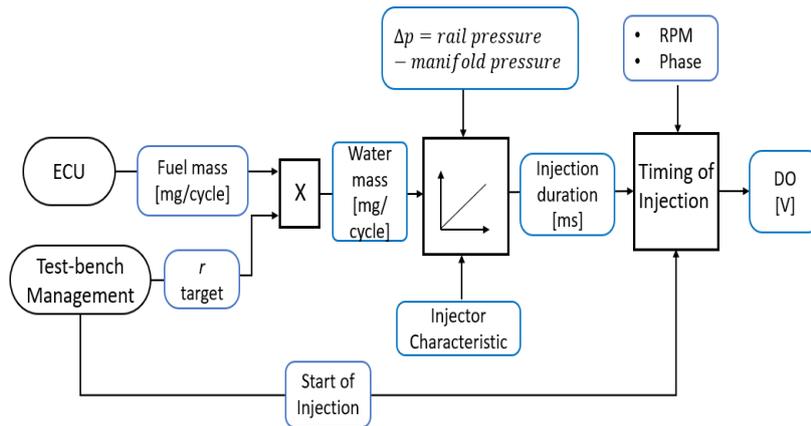


Figure 2.48: D/O management for water injectors

To test the WI controller, the algorithm developed via Software in the Loop has been integrated in the RT code described in the previous section. After this upgrade, the controller provides (cycle by cycle) the  $r$  value to the Port WI system and the new SA value to the ECU via CAN.

### 2.4.2. Controller Overview

The Water Injection Combustion Controller consists of three parts (as seen in Figure 2.49):

1.  $r$  Map and MFB50 Map are calibrated maps that provide a target value of  $r$  and MFB50 in relation to RPM and load.
2. WI Combustion Model calculates the required SA to obtain the MFB50 target considering also the effect of  $r$ .

3.  $r$ /SA Correction Management provides corrections to the computed SA value by the model and to the  $r$  target (respectively called dSA PID and drPID). The corrections are oriented to keep knock index at a threshold value (Closed Loop on Knock Index). In case of availability of MFB50 measure this part has been evolved to activate also a Closed Loop on MFB50 angle.

This approach can effectively allow to manage  $r$  and MFB50 as independent variables. In fact, every correction imposed by drPID does not affect the final MFB50 because the WI Combustion Model can take in account the variation of  $r$  and modifies the value of SA produced by model to achieve the MFB50 target. The load has been defined again as Net Load (NL), by Equation 2.10.

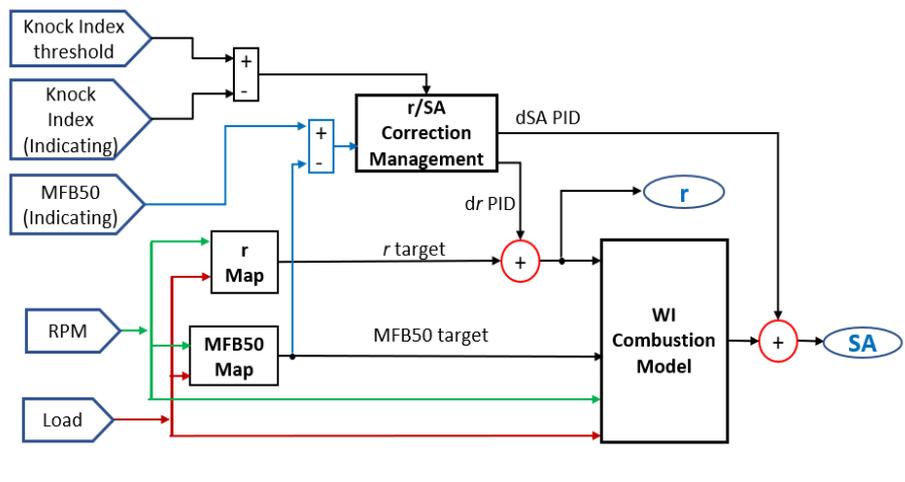


Figure 2.49: Water Injection Combustion Controller lay-out.

### 2.4.2.1. WI Combustion Model

Different approaches to model the MFB50 as a function of engine operating conditions (speed, load and SA) and injected water mass have been discussed in the previous section of this chapter. Nevertheless, the polynomial method (directly developed from the analytical MFB50 model described in the first chapter) is the most accurate approach for the MFB50 calculation. The SA can be easily deduced by reversing equations on which such model is based. The polynomial approach is briefly described below for clarity reasons.

It has been established that the relation between MFB50 and SA corresponds to a parabolic function that can be described as:

$$SA = a MFB50^2 + b MFB50 + c \quad (2.21)$$

At the same time, each parabolic coefficient  $a$ ,  $b$ ,  $c$  is also a polynomial function of both RPM and NL. Experimental data show that for the engine used during the experimental campaign, there is a stronger dependence with RPM than with NL, reason why the polynomial function used to describe each parabolic coefficient  $a$ ,  $b$  and  $c$  is the following:

$$PCoeff = p_{00} + RPM * p_{10} + NL * p_{01} + RPM * NL * p_{11} + RPM^2 * p_{20} \quad (2.22)$$

Where  $p_{00}, p_{10}, p_{01}, p_{11}, p_{20}$ , are called Surface Coefficients. As previously mentioned, WI has an influence on MFB50 / SA relation. For this reason, the effect of  $r$  into the model is added by describing each surface coefficient  $p_{xx}$  as a linear function:

$$p_{xx} = o_{xx} + r * g_{xx} \quad (2.23)$$

Figure 2.50 shows a block diagram of the combustion model. The inputs are  $r$ , RPM, NL and MFB50 while the output is the required SA to satisfy the selected MFB50 target from input.

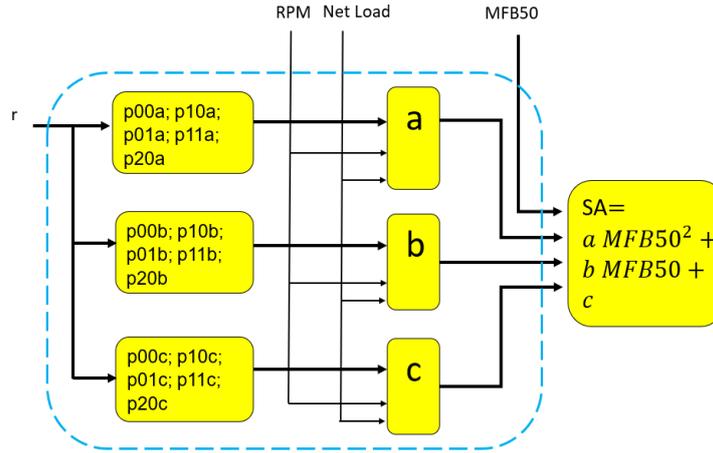


Figure 2.50: Combustion Model with Polynomial Method.

#### 2.4.2.2. $r$ /SA Correction Management

As mentioned before, this subsystem has been developed through two distinct stages. At first, a controller based on knock index has been designed to manage corrections of  $r$  ( $dr$ PID) and SA ( $dS$ A PID), as shown in Figure 2.51. The  $dr$ PID is considered by the WI Combustion Model and therefore does not affect the resulting MFB50. On the contrary  $dS$ A PID is applied to the SA provided by the model and can change the combustion phase from the target value. The management of  $dr$ PID and  $dS$ A PID is driven by the difference between the actual knock index value (MAPO98) and the established threshold, also called MAPO98 Error.

$$MAPO98 \text{ Error} = MAPO98 - MAPO98_{threshold} \quad (2.24)$$

MAPO98 Error is translated into a proportional and integral correction through a gain scheduling PI structure. The resulting sum of two contributions is then converted in a percentage value which represents the total amount of the correction also called TPC (Total Percentage Correction). The value of TPC defines the way to split  $dr$ PID and  $dS$ A PID actuation according to the following strategy.

If TPC is lower to a calibrated value (*HighTPC*), MAPO98 is considered close to MAPO98 threshold. In this case  $dr$ PID is used as function of TPC to control knock, keeping  $dS$ A PID to 0 in order not to modify the resulting MFB50. To achieve the proper value of  $r$  in every condition  $dr$ PID must be positive for TPC greater than 0 (equal to MAPO98 greater than threshold) and negative for negative TPC (equal to MAPO98 below threshold). The second condition is important to reduce the injected water mass when knock level is low enough to allow a reduction of  $r$  with respect to mapped value.

If TPC is greater than *HighTPC*, *drPID* remains fixed to a saturation value and knock level is controlled by *dSAPID*. *dSAPID* applies negative corrections (it retards SA) to move actual MFB50 to a delayed position. Furthermore, in case of high knock events on single cycles (*MAPOcc*) the TPC is forced to its maximum value by a lower saturation. This is a safety strategy to activate a fast response in case of relevant risk.

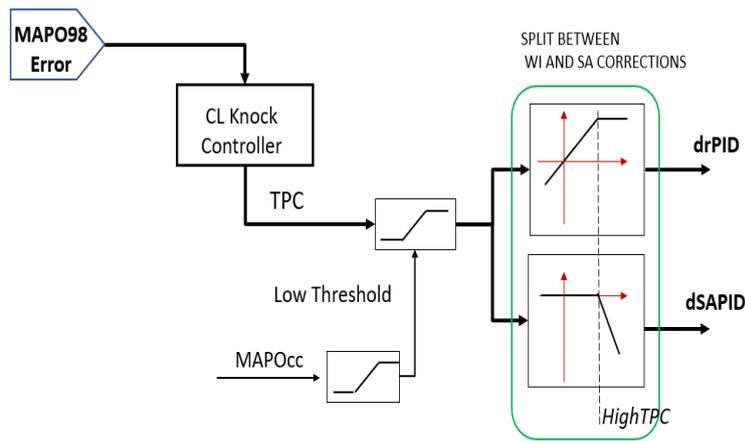


Figure 2.51: r/SA management: Closed Loop on MAPO98.

As a second stage, a new branch has been added in the subsystem to enable a closed loop on MFB50 (Figure 2.52). Like the above-mentioned chain, a MFB50 Error and a MFB50 TPC have been defined. In this configuration *dSAPID* is the sum of the correction provided by the two branches and have been called *dSAPIDknock* and *dSAPIDmfb*. These two terms operate independently, but TPC from knock branch can force the MFB50 TPC with a lower saturation (*Dis\_mfbCL*). This layout allows, in cases of high TPC, to gradually reduce positive correction of *dSAPIDmfb* that would anticipate the final SA, until the sum of two correction produce a stable final correction that can mitigate the knock level.

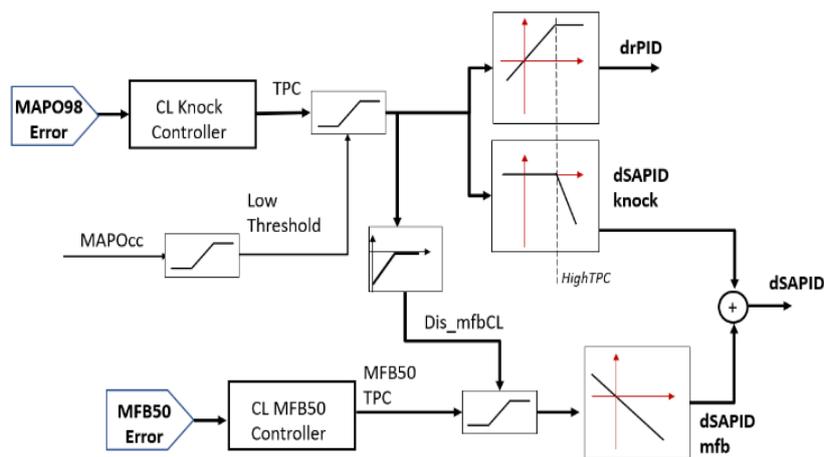


Figure 2.52: r/SA management: Closed Loop on MAPO98 and Closed loop on MFB50

### 2.4.3. Calibration Campaign

The model requires an experimental campaign to calibrate surface coefficients  $p_{00}, p_{10}, p_{01}, p_{11}, p_{20}$  for each parabolic coefficient  $a, b$  and  $c$ . At the same time each surface coefficient is comprised of a linear function with  $r$ , making a total of 30 coefficients to be determined during calibration. The least squares method has been used to resolve the equation system from which every coefficient of the model is obtained. During the experimental campaign, all the inputs and output of the model are registered, generating a dataset. From each of the entries on the dataset, corresponding to a different configuration of input and output variables, the final equation system is formed and resolved with the least squares method. Such tests go to integrate the database collected for the development of different types of combustion models, showed in Table 2.5.

During the development of the model, certain restraints have arisen regarding the RPM / NL grid in which the model can be used. It has been observed that in points with  $NL < 0.8$ , the standard deviation (STD) of MFB50 is too high for the model to be validated with results that give a calculated value of SA that is 1 CA distant from the really actuated one (limit imposed by the authors of this work regarding desired precision). For this reason, both calibration and validation have been established with values of  $NL \geq 0.9$ . The upper limits of the calibration grid have been limited by both the capacity of the turbo to generate boost and the mechanical integrity of the 3D printed polymeric intake manifold used in the engine.

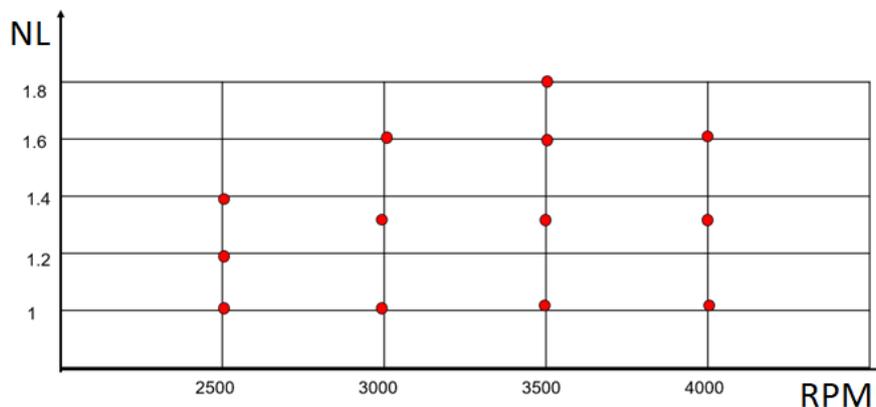


Figure 2.53: Grid of calibration points.

For each engine point seen on Figure 2.53, a sweep of SA, and  $r$  has been registered. Considering that the objective of the Combustion Controller is to first stay as close as possible to the optimum phase point (MFB50 = 8 CA ATDC), as long as knock does not exceed established threshold, the SA sweep in each point has been done in the proximity of the optimum SA without surpassing the knock threshold. It has been observed that a better calibration results of a broader SA sweep with fewer elements inside it instead of a narrow sweep but with a fine step between each point in it. A step of 1.5 CA in SA has been used between each element in the sweep. Regarding WI, the values used for calibration vary between  $0 < r < 0.8$  with a step of 0.2. In points in which it has been observed that a certain value of  $r$  was enough to stay with MFB50 at optimum phase point and still be under knock threshold, no higher amounts of water were used for calibration given the fact that the Combustion Controller will never apply such values of  $r$ .

## 2.4.4. Performance of WI Combustion Control System

The validation of WI Combustion Controller has been carried on evaluating step by step the performance of every part of the whole system.

### 2.4.4.1. Open Loop on MFB50

At first, WICC has been tested disabling the entire  $r$ /SA Correction Management subsystem, to highlight the functioning of WI Combustion Model. In fact, in this configuration there are no corrections on  $r$  and SA and the controller works in total open loop.

As shown in Figure 2.54, the test is composed by two different stages. The first one (from second 10 to second 45) features steps of MFB50 target at a fixed value of  $r$  (Ract) of 0. For each step the model modifies the actuated SA (SAact) accordingly and can produce an actual MFB50 (MFB50meas) close to the target within a range of 2 CA, even during transients. In the second stage (from second 45 to 75) the MFB50 target is set at a single value and steps of  $r$  are applied. In this case the model modifies SA to compensate the effect of water injection and the measured MFB50 remains close to the target. It is important to observe that the model's (OL) precision also varies depending on the value of  $r$ .

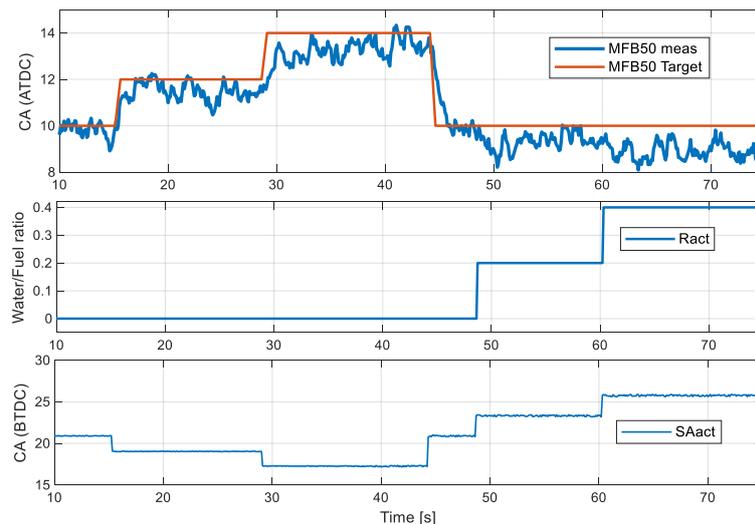


Figure 2.54: Steady state engine point 3300 RPM, 1.25 NL, Cylinder 2. On top has been displayed the match between MFB50 measured and MFB50 target, in the middle the value of  $r$  actuated, at the bottom the SA as model output to achieve the MFB50 target.

### 2.4.4.2. Closed Loop on MFB50

After the evaluation of the model performance, the CL on MFB50 branch has been tested. The procedure is similar to the previous one because the same steps of MFB50 target and  $r$  target are imposed, but this time at second 13.5 the MFB50 CL branch is activated as displayed in Figure 2.55. When the CL is activated, dSAMfb starts to apply correction on SA provided by model and SAact becomes the sum of these two terms. From this instant actual MFB50 is forced to correspond to the target value. The value of dSAMfb can also be used to evaluate the precision of the WI Combustion Model in the defining of SA. Close loop aids in correcting for the previously mentioned imprecisions of the model with increasing values of  $r$ .

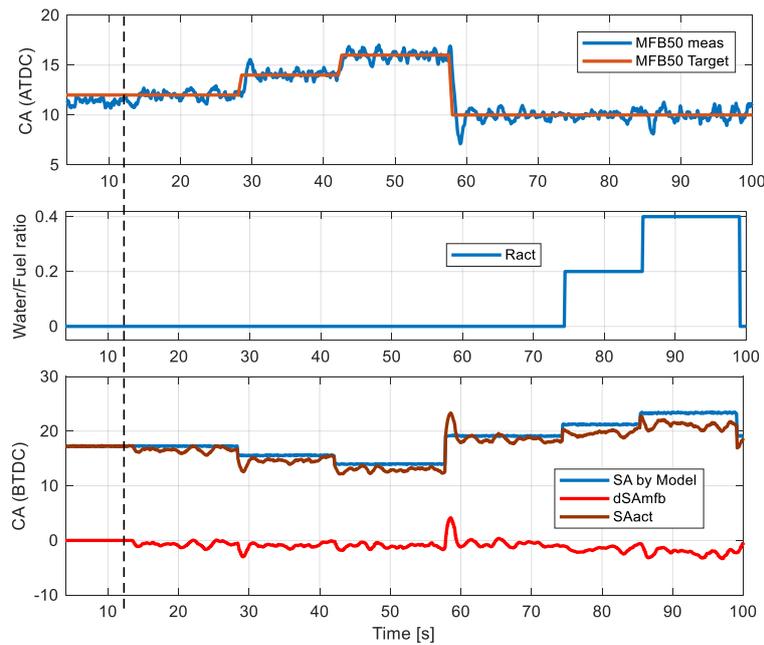


Figure 2.55: Steady state engine point 2800 RPM, 1.25 NL, Cylinder 2. At time 0 controller starts in open loop (as seen in Figure 9). After second 13.5 the controller switches in closed loop mode and the PI correction  $dSAmbf$  compensates the error on MFB50.  $SAact$  represents the sum of  $SAbyModel$  and  $dSAmbf$ .

### 2.4.4.3. Closed Loop on MFB50 and MAPO98

In the next three Figures (2.57,2.58,2.59), the behavior of the CL on  $r$ /SA Management is displayed. Two different situations are presented, one in which the correction of WI ( $dR$  PID) does not saturate (preestablished value of saturation of  $dR$  PID = 0.3) and two others in which it does.

MFB50meas corresponds to a moving average from cycle by cycle MFB50 measured angle while MFB50target is the value of the MFB50 Map. Ract and Rmap are the final  $r$  value actuated and  $r$  value obtained from the map, respectively.  $dR$  PID is the correction applied by the knock branch of the PID on actuated  $r$ , and  $dR$  PID sat is the previously mentioned saturation value imposed on the correction of  $r$  by the PID. SA act represents the finally actuated SA, that is composed of SAmodel (calculated by the model) and corrected by both  $dSAmbf$  and  $dSAknock$  which are the contributions of both closed loop branches. MAPOcc indicated the cycle by cycle value while MAPO98 and MAPO98thr are the calculated 98<sup>th</sup> percentile and its threshold value accordingly.

Figure 2.56 displays a test in which the controller operates in complete open loop until second 18 (SA controlled solely by the model and WI only by the  $r$  obtained from the  $r$  MAP). Since second 18 both CL branches (knock and MFB50) are activated. It can be seen how shortly before activating CL, knock is generally under threshold and MFB50 is slightly retarded. The response of the knock branch is an initial negative  $dR$  PID correction (reducing amount of water injected to save water) while the MFB50 branch applies positive correction ( $dSAmbf$ ) in order to get closer to the MFB50 target, given the previous slightly retarded OL response. From there on, both branches continue to keep both knock and MFB50 in target without saturating the  $dR$  PID. Values of  $dR$  PID keep the quantity of water injected close to the one set by the  $r$  MAP. The corrections given by both branches correspond only to the integral part, this is what gives the slow oscillating behavior close to each respective target. SA calculated by the model follows the form of Ract since it considers the effects of  $r$ .

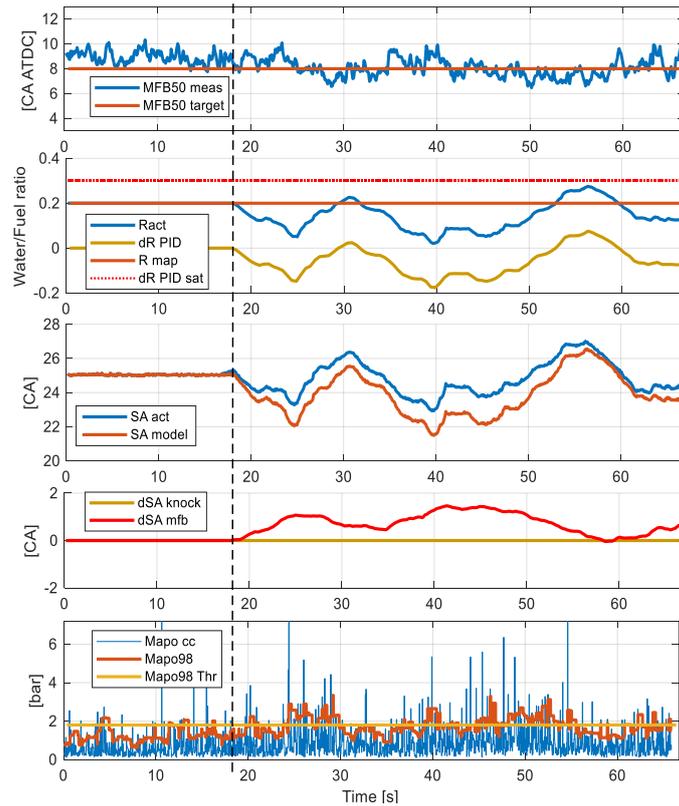


Figure 2.56: Steady state engine point 2700 RPM, 1.22 NL, Cylinder 1. At second 18 the controller switches from open loop to closed loop:  $dSA_{mfb}$  starts to compensate SA model (as seen in Figure 2.55) and correction  $dR_{PID}$  compensates R map (mapped value of  $r$ ) to keep Mapo98 (knock Index) close to threshold (Mapo98 Thr).

Figure 2.57 displays a situation in which from the start both branches are set in CL. At second 13 a sudden change in MFB50 target is imposed in order to generate a high knock tendency with the intention of stimulating a strong response of the controller. After second 13, SA changes instantly due to the model's reaction to the MFB50 target change, real MFB50 is anticipated and knock level increases. The knock branch immediately acts to lower knock by adding water and quickly saturates at  $dR_{PID} = 0.3$ . In case of saturation, MFB50 branch is forbidden to add positive contributions to SA (advancing SA). From the moment in which  $dR_{PID}$  saturates, the SA calculated by the model remains constant due to a constant MFB50 target and R map (given  $dR_{PID}$  saturation) and the SA correction from the knock branch ( $dSA_{knock}$ ) starts to retard combustion until MAPO98 is lowered to threshold. Furthermore, a delaying of MFB50 respect to the target is allowed giving the priority to maintain MAPO98 at threshold.

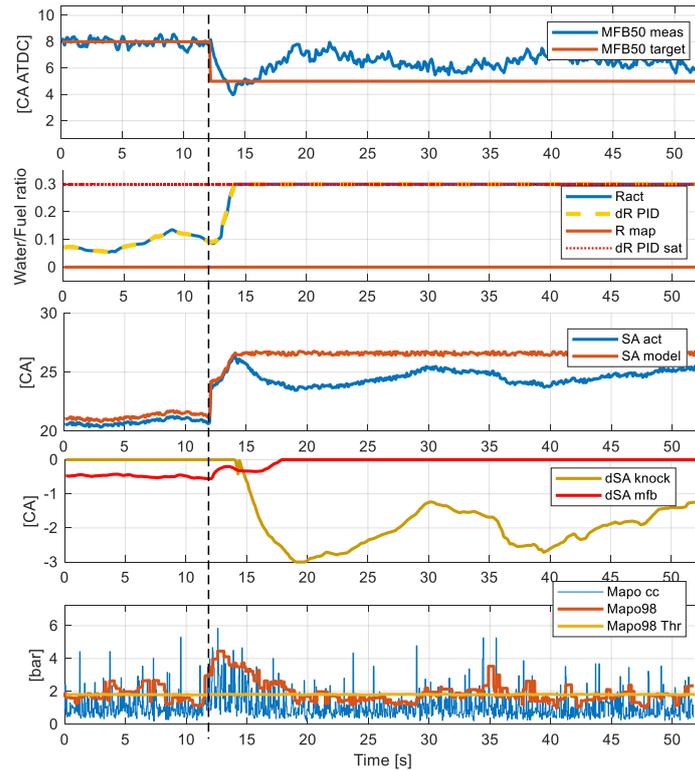


Figure 2.57: Engine point 2700 RPM, 1.22 NL, Cylinder 4. The controller is in closed loop. After second 13 MFB50 target is moved to an advanced value to stimulate high knock tendency. After 2 seconds dR PID, attempting to compensate Ract reaches saturation value (dR PID sat). After saturation, dSA mfb is limited to negative values and correction dSAknock starts to apply correction on SAact to delay MFB50.

Figure 2.58 presents a situation in which the value R map has been lowered from the calibrated one in order to obtain a higher knock level (over threshold of MAPO98). In this test the MFB50 branch is activated from the beginning and at second 35 also the knock CL branch is activated as well.

This test proves that the knock branch is able to quickly lower MAPO98 to threshold when being activated in a over threshold situation. In this case, after dR PID saturation around second 38, MFB50 can still be close to target since the amount of water injected at saturation is almost just enough to keep MAPO98 at threshold. This is also the reason why corrections of dSA knock are also small (less than 2 CA).

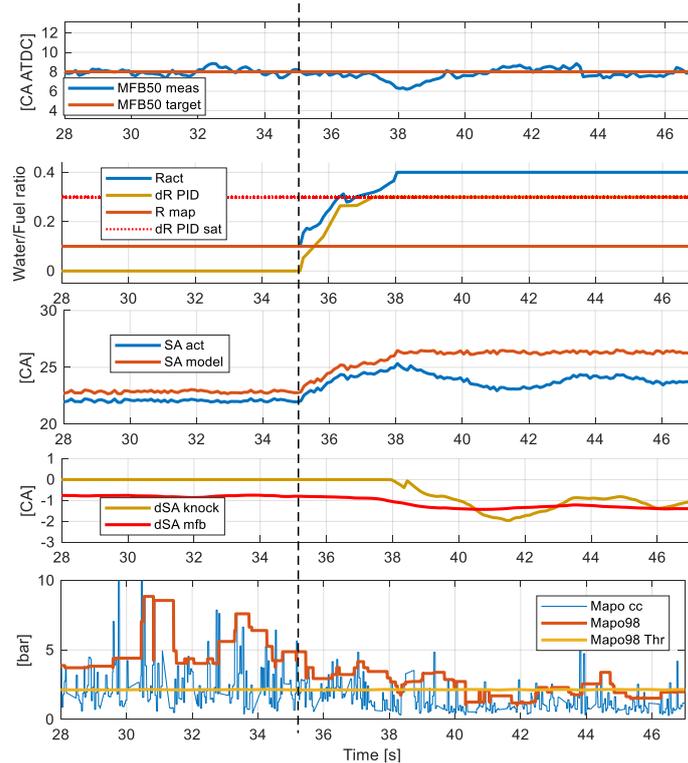


Figure 2.58: Engine point 3200 RPM, 1.3 NL, Cylinder 4. At time 0, the controller is in closed loop only for MFB50 and knock index exceeds the threshold value. After second 38 even loop on Mapo98 is closed.

#### 2.4.4.4. Controller Response Under Transient Conditions

In the next two Figures (2.59, 2.60) a transient situation is presented, reproducing a real situation in which there is a sudden change of load and RPM increase gradually.

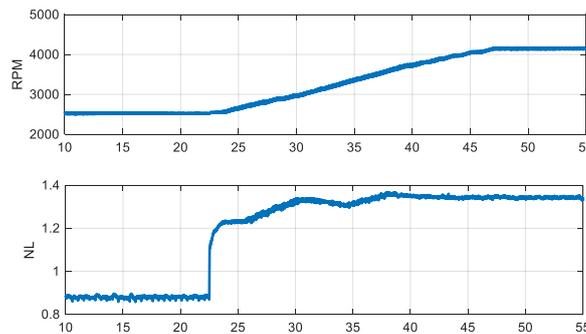


Figure 2.59: Transient test speed and load profiles.

Figure 2.60 displays how the controller can retain knock levels at MAPO98 threshold while in transient conditions as well. R map changes over time due to the everchanging engine point and the knock branch does not saturate at any moment due to a well calibrated  $r$  MAP. dR PID adds corrections that keep  $r$  values close to mapped ones. This can also be noted by the fact that when the sudden change in load is applied, MAPO98 still stays at threshold and does not go over it, demonstrating the capabilities of the open loop. Closed loop works as it should, making small adjustments to keep MAPO98 and MFB50 at target, being mostly a result of the integral part of both branches of the PID.

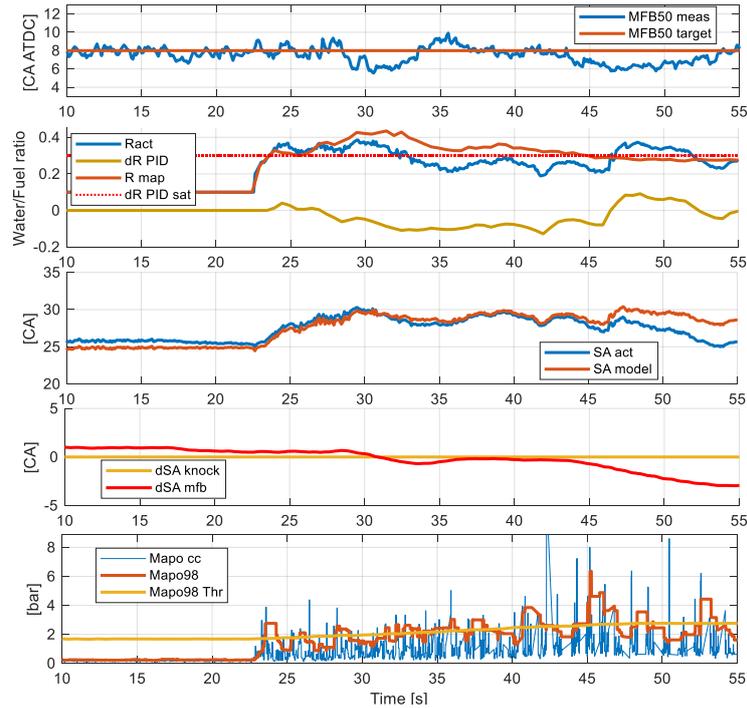


Figure 2.60: Transient test controller performance, Cylinder 1.

#### 2.4.4.5. Fuel Consumption Implications

The performance of the Combustion Controller can be also quantified in terms of fuel consumption savings. WI is utilized to mitigate knock and lower EGT while allowing for greater values of SA. When not using WI, it is necessary to enrich the mixture to achieve these same goals, with the inevitable associated increase in fuel consumption.

To evaluate this fuel consumption difference, two tests were made, one with WI and another without. Both tests were conducted at 2700rpm and 1.45 NL. In the case of WI, a value of  $r = 0.8$  was used to mitigate knock as much as possible and SA was set so that MFB50 was as close as possible to the optimum 8 CA ATDC, while remaining below an established MAPO98 limit value of 2 bar. The non-WI test was conducted in a similar way, in which  $\lambda$  and SA are set up in order to be as close as possible to optimal MFB50 and stay below the 2 bar MAPO98 limits. The results are summarized in Table 2.9.

The results with WI show an improvement in fuel consumption of 16% compared to the non-WI test, given mainly by the difference in  $\lambda$ . Even though the WI test has a more anticipated MFB50 value, the non-WI test shows a higher IMEP. The reason is related to the IMEP degradation at high  $r$  value, as reported in [2.1]. Therefore, in this case the gain of efficiency is due to both the avoiding of  $\lambda$  enrichment and the achievement of a more efficient combustion phase.

Table 2.9: Comparison between WI and non-WI tests (2700rpm, 1.45 NL).

	With WI	Without WI
Water / Fuel ratio ( $r$ )	0.8	0
MFB50 [CA ATDC]	12	18
MAPO98 [bar]	1.9	2
IMEP [bar]	16.7	17.3
Lambda	1	0.89
EGT [°C]	786	818
Indicated Torque [Nm]	185	187
Fuel Consumption [kg/h]	11.3	13.5
Indicated Specific Fuel Consumption [g/kWh]	216	257

### 2.4.5. Conclusions

In this section a detailed evaluation of a Water Injection (WI) Combustion Controller performance has been carried out via a step by step approach. The implementation of the polynomial combustion model developed and presented in section 2.3 in a real-time RCP environment is presented.

At first, a completely Open Loop configuration has been tested, showing that the Combustion Model is able to produce a MBF50 angle close to MFB50 target with a precision of 2 CA even in presence of WI. Furthermore, the model in combination with the proposed closed-loop knock controller branch has been used to manage the knock level, applying both water injection and SA corrections. Through this configuration, the feasibility of knock mitigation by adding water without sensible variation in MFB50 (if WI is not saturated) has been demonstrated. In fact, the presence of variations of water-to-fuel ratio ( $r$ ) does not significantly affect the combustion phase due to the model compensation, and the knock level can be stabilized to the established threshold.

The addition of a closed-loop correction on MFB50 by in-cylinder pressure signal allows to better implement the controller, keeping the combustion phase on the target as long as  $r$  value is not saturated.

Hence, controller needs the combustion phase index (MFB50) that is not commonly available on the final on-vehicle application, while, as mentioned in chapter 1.4, knock intensity is typically estimated with accelerometers or ionization current. Nevertheless, innovative algorithm for the combustion phase estimation from the accelerometric signal described in section 1.4.2 can be efficiently applied to provide MFB50 index even to this WI controller. This is undoubtedly one of unsolved activities of this PhD course that will be developed during next years.

As stated in the introduction, main advantage resulting from the use of the WI is undoubtedly the important reduction of exhaust gas temperature that allows to avoid the mixture enrichment at critical engine points. Indeed, the control of such temperatures is another key topic in the field of combustion efficiency management, due to the direct influence of such efficiency on the wasted heat through the exhaust system. The third chapter of this document finally deals with exhaust gas temperature modelling and the analytical approach presented in previous chapters and used to describe different combustion indexes is also applied to develop a control-oriented model such temperature.

## References

- 2.1. Cavina, N., Rojo, N., Businaro, A., Brusa, A. et al., "Investigation of Water Injection Effects on Combustion Characteristics of a GDI TC Engine," SAE Int. J. Engines 10(4):2017, doi:10.4271/2017-24-0052.
- 2.2. Ranuzzi, F., Cavina, N., Brusa, A., De Cesare, M. et al., "Development and Software in the Loop Validation of a Model-based Water Injection Combustion Controller for a GDI TC Engine," SAE Technical Paper 2019-01-1174, 2019, doi:10.4271/2019-01-1174.
- 2.3. Ranuzzi, F., Cavina, N., Scocozza, G., Brusa, A. et al., "Experimental Validation of a Model-Based Water Injection Combustion Control System for On-Board Application," SAE Technical Paper 2019-24-0015, 2019, doi:10.4271/2019-24-0015.
- 2.4. Gamma Technology Inc., 2017. GT Gamma Technology.
- 2.5. Pauer, T., Frohnmaier, M., Walther, J., Schenk, P. et al., "Optimization of gasoline engines by water injection", 37th INTERNATIONAL VIENNA MOTOR SYMPOSIUM, Wien, 2016.
- 2.6. Wenig, M., Grill, M., and Bargende, M., "A New Approach for Modeling Cycle-to-Cycle Variations within the Framework of a Real Working-Process Simulation", SAE Technical Paper 2013-01-1315, 2013, doi:10.4271/2013-01-1315.
- 2.7. Millo, F., Rolando, L., Pautasso, E., and Servetto, E., "A Methodology to Mimic Cycle to Cycle Variations and to Predict Knock Occurrence through Numerical Simulation", SAE Technical Paper 2014-01-1070, 2014, doi:10.4271/2014-01-1070.
- 2.8. Leppard W. R., Individual-Cylinder Knock Occurrence and Intensity in Multi-Cylinder Engines, SAE Technical paper, 820074, 1982.
- 2.9. Cavina, N., Corti, E., Minelli, G., Moro, D. et al., "Knock Indexes Normalization Methodologies", SAE Technical Paper 2006-01-2998, 2006, doi:10.4271/2006-01-2998.
- 2.10. Xudong, Z., Yang, W., Shuaiqing, X. et al., "The engine knock analysis - An overview", Applied Energy, Volume 92, Pages 628-636, 2012, <http://doi.org/10.1016/j.apenergy.2011.11.079>.
- 2.11. Businaro, A., Cavina, N., Corti, E., Mancini, G., Moro, D., Ponti, F., Ravaglioli, V., "Accelerometer Based Methodology for Combustion Parameters Estimation", , Energy Procedia, Volume 81, December 2015, Pages 950-959, <https://doi.org/10.1016/j.egypro.2015.12.152>.

### 3. Exhaust Gas Temperature Modelling

### 3.1. Introduction and Literature Review

Generally speaking, the development of real time controllers with the aim to increase combustion efficiency cannot ignore a reliable estimation of resulting exhaust gas temperature, especially in modern turbo-charged downsized engines that reach high values of specific power, causing a significant increase of the exhaust gas temperature. Such parameter plays a key role in the overall powertrain environmental impact because it strongly affects both the catalyst efficiency and the turbine durability. In fact, common techniques to properly manage the turbine inlet gas temperature are based on mixture enrichment, which causes both a steep increase in specific fuel consumption and a decrease of catalyst efficiency. At the test bench, exhaust gas temperature is typically measured using thermocouples that are not available for on-board application, and such information is processed to calibrate open-loop look-up-tables. A real-time, reliable, and accurate exhaust temperature model would then represent a strategic tool for improving performance of the engine control system. For this reason, such topic has been deepened during PhD research activity and interesting results have been achieved by applying the analytical approach used to develop models described in previous chapters, such as for knock intensity index calculation, for the combustion phase (MFB50) and maximum in-cylinder pressure estimation and so on. The resulting model has been developed to integrate the exhaust gas temperature calculation within the adaptive damage-based combustion controller described in section 1.4.1. Results described in this chapter have been published in [3.1].

Some kinds of exhaust temperatures models are widely used in the automotive field and are typically implemented in Engine Control Units (ECUs). An accurate estimation of such parameters over the entire engine operating range for both steady-state and transient conditions is needed in order to prevent turbine failure in turbo-charged engines, and in general to optimize the after-treatment system efficiency, as highlighted by Fu and Chen [3.2]. The latter reason is particularly meaningful for modern three-way catalysts that work properly in a limited temperature range. Outside such temperature interval there is an evident drop for both conversion efficiency (for lower temperature values) and catalyst useful life (for higher temperature values). As well-known, the latest emission regulations for passenger cars are forcing engine manufacturers to achieve maximum efficiency for both the combustion process and the after-treatment systems [3.3]. Moreover, the exceedance of the maximum gas temperature at turbine inlet must be avoided, to prevent turbine failure. For such reasons, an accurate and reliable exhaust temperature model is a key tool in managing the main control levers to achieve desired engine-out gas temperature. The most widespread Spark-Ignition (SI) engine strategy for catalyst heating is spark retardation, while common techniques to lower exhaust gas temperatures are both the use of non-stoichiometric Air-to-Fuel Ratio (AFR) and load limitation, as well explained in [3.4]. For the mentioned control strategies, the engine-out temperature calculation is typically coupled with models for the estimation of the catalyst thermal behavior.

Majority of modelling effort found in literature deals with catalyst warmup and efficiency prediction and with temperature calculation at the turbine inlet [3.5, 3.6]. Some works develop methods based on 1-D approach that solve mass-energy balance equations for every sub volume within which exhaust pipes are divided. For example, the physics-based 1-D thermocouple model described in [3.7] outputs the estimated temperature by solving the thermal balance related to the heat exchanged by convection between exhaust gases and the thermocouple tip, and by conduction and radiation with the pipe wall. Nevertheless, such method is not compatible with RT application due to the high computational effort needed to solve the energy balance equation and this makes such 1-D models typically more suitable for engine components design and for the development and validation of

custom control algorithms via Software-in-the-Loop simulation. On the other hand, accurate knowledge of temperature distribution during the engine cycle allows to highlight further criticisms for turbine but such kind of studies needs the calibration and validation of 3-D CFD models [3.8, 3.9, 3.10]. Instead, focus of this work is the real-time exhaust gas temperature modelling for SI engines.

The innovative analytical approach developed by the author is described in detail in this chapter and an empirical model is developed and validated comparing the signals of the Thermocouples (TCs) installed in the engine exhaust manifold, with the modelled ones. Such model is based on the identification of appropriate polynomial functions to interpolate exhaust temperature values measured under steady-state operating conditions. Polynomial fittings of temperatures recorded in standard conditions (mapped values for both SA and lambda) do not consider the effects of deviations in SA or lambda targets under real engine operation. For such reason, the sensitivities to combustion phase and AFR are modelled through polynomial functions that scale the modelled temperatures as a function of SA or lambda deviations with respect to the nominal values. The analytical approach derives from control-oriented models described in section 1.3.3 [3.11] in which main combustion indexes (such as the 50% Mass Fraction Burnt Angle, or MFB50, the maximum in-cylinder pressure, and the Indicated Mean Effective Pressure or IMEP) are modelled using polynomial equations. The major advantage of the analytical approach is an extremely low computational effort required to run the resulting model, which makes it particularly suitable for Real-Time (RT) applications. Lookup Table calibration or artificial Neural Network (ANN) may be considered as valid alternatives to the polynomial method, and they represent the state-of-the art approaches [3.12]. However, it can be stated that limited literature in 0-D exhaust gas temperature modelling exists, while 1-D and 3-D modelling have been more extensively explored [3.13, 3.14]. The analytical model, based on polynomial functions, is the innovative contribution presented in this chapter since it ensures a higher accuracy (due to the absence of discretization of the independent variables) and a lower calibration effort. In fact, when the polynomial degree of the fitting function is fixed, the number of calibration points can be significantly reduced. More generally, if the Lookup-Table and ANN-based models can be associated to the typical *black-box* approach, it can be asserted that the proposed methodology is more similar to a *grey-box* one, due to preliminary analysis of variable trend.

A simplified methodology to reproduce the TC temperature signal also for transient conditions is the focus of the second part of this chapter. The TC output and dynamics are affected by three main factors: TC lumped thermal capacity, conduction heat transfer between TC tip and pipe wall through the mounting support, and radiation heat transfer between TC and the surrounding pipe wall [3.15]. While TC thermal inertia has no effects in steady-state conditions, and it acts like a first-order system when the engine point changes, conduction and radiation heat transfers affect both steady-state and transient TC output. Indeed, they typically introduce a negative offset on the stationary measured temperature, causing a systematic underestimation [3.16]. Nevertheless, the steady-state TC measurement can be considered significant for the purpose of this work because the limit values provided by turbine and catalyst manufacturers are defined by using thermocouples in steady-state conditions. This means temperature limits for such components already include the negative offset given by performing measurements with TCs, thus, they can be taken as reference. Under dynamic conditions, the TC thermal capacity is probably the most influent factor on the initial response, and it may be synthesized by its time constant. Moreover, the conduction heat transfer between the TC and the walls of the exhaust runner additionally deviates the TC tip temperature from that of the exhaust gases under dynamic conditions, slowing down the reaching of the steady state value [3.17]. A similar effect is due to the heat transfer caused by radiation between the TC tip and the pipe wall, which is particularly relevant since the heat transfer is proportional to the temperature difference

raised to the 4<sup>th</sup> power. Therefore, conduction and radiation heat transfers between TC and pipe wall, together with TC thermal inertia and the mixing of the novel exhaust gases ejected from the combustion chamber with those within the runners, cause the resulting experimental trend of temperature readings under transient conditions. Such exhaust gas mixing and the effect of the TC time constant are responsible for the initial fast part of the TC response during a steep temperature transition. The slower behavior is, instead, given by conduction and radiation with the pipe wall [3.18, 3.19]. This model includes two Moving Averages (MAs) that are calibrated to reproduce the resulting trend of the TC signal. The analytic exhaust gas temperature model coupled with the TC dynamics model are validated through the comparison between the calculated values and the measured ones, for both steady-state and transient conditions, and the error is evaluated.

## 3.2. Experimental Campaign

Experimental tests were carried out on the 8-cylinder used also for the calibration and validation of the analytical engine simulator (section 1.3.3) and the adaptive damage-based combustion controller (section 1.4.1). Engine characteristics are thus collected in Table 1.8. Data used for development and calibration of the model were recorded performing spark and lambda sweeps on the entire engine operating range, and the mean cylinder was considered for the indicating indexes. The tested operating conditions are shown in Figure 3.1. For each engine point, the spark sweep was performed while maintaining the mapped lambda value, and the measurements were performed in steady state conditions, including about 200 engine cycles for every operating point. An analogous approach was used for lambda sweeps: the AFR value was changed while fixing a given value of the MFB50 and engine load, via adjustment of the actuated SA and the requested torque, respectively. Moreover, some transient operating conditions were tested while the bench signals (such as the exhaust temperature readings from the TCs, engine speed and load) as well as the main combustion and control indexes were logged for the validation process. Such data were collected on engine points that are not included in the initial calibration database, to also check the reliability of the proposed methods. The exhaust runners and manifold were instrumented with Mineral-Insulated-Metal-Sheathed (MIMS) thermocouples. The main combustion indexes were calculated from the in-cylinder pressure signals: CHAMPION charge amplifier and OBI-M2 indicating system by Alma Automotive were used for signal conditioning and acquisition. Indeed, each cylinder was equipped with a piezoelectric transducer, and the signal was sampled with a frequency of 200 kHz. The calculation of MFB50, IMEP and Pmax uses the low-pass filtered pressure trace with a cut-off frequency of 3 kHz. For confidentiality reasons, all the variables and parameters were normalized or partially offset.

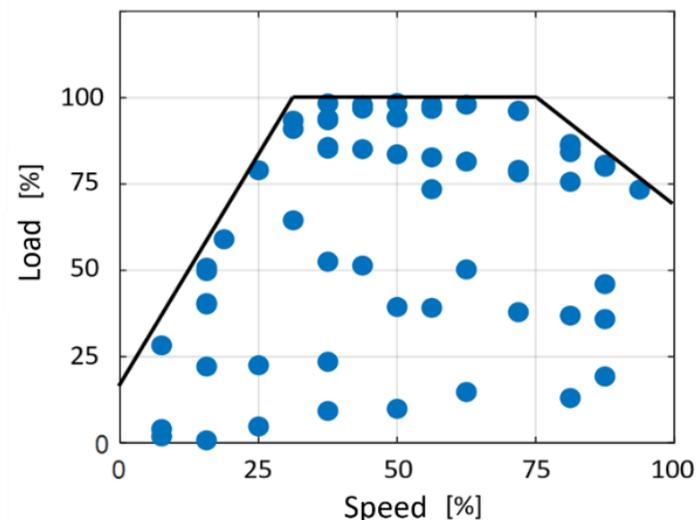


Figure 3.1: A total of 60 engine points on the Speed-Load range were tested for the calibration process. For each point a spark sweep was performed, and for 14 of these, a lambda sweep was carried out.

### 3.3. Analytical Model

The proposed model has been calibrated to minimize errors between the experimental exhaust manifold gas temperature and the calculated one under steady-state conditions. The temperature of gases within the exhaust manifold is considered.

In the following section the development of the analytical model and the thermocouple dynamics model are thoroughly described. It is important to mention that the empirical model of the TC dynamics was developed to improve the quality of the calculated values when compared against experimental signals. This becomes unnecessary when the aim of the model is to determine the exhaust gas temperature on-board the vehicle, for instance within a RT control strategy for a Turbo-Charged engine that must prevent the turbine failure through mixture enrichment. In this case the implementation of the TC dynamics model acts as a first-order system, hence it causes the calculation of temperatures that are coherent with TC output but lower than the “real” ones during fast transient conditions characterized by increasing power (from low to high load), according to what stated in the introduction. This means that for the prevention of turbine overheating such model should always calculate the real gas temperature, rather than the measured temperature. On the contrary, the TC dynamic model is needed for the exhaust temperature model validation under transient conditions because the TC signal is the only measurable feedback of such engine variable. A comparison between the modelled and the logged temperatures without the contribution of the TC dynamics representation would only be meaningful for steady-state conditions.

The proposed approach provides the analysis of the effects that the engine operating condition (both engine speed and load), the combustion phase (defined through the MFB50 index that represents the angular value for which the normalized Cumulative Heat Release curve meets the 0.5 value) and the lambda value produce on the exhaust gas temperature. Moreover, such effects were studied separately and included in the model by adding each contribution to reach the temperature for fixed engine speed, load, MFB50 and lambda. In other words, such model was developed by using the effects-separation method: the steady-state gas temperature, estimated for a given engine operating point, with the calibrated lambda value and a reference value of MFB50, is multiplied by two functions that are designed to adjust the reference temperature when the AFR or the combustion phase deviate from nominal conditions.

Figure 3.2 shows the block-layout of the analytical model, as implemented in Simulink. The reference temperature (obtained for the reference values of combustion phase and lambda) is calculated with the T\_EXH\_REF block, that uses the engine speed (ES) and the engine load (EL) as inputs. EL is defined as the relative air charge inside the combustion chamber, independent of engine displacement, and it is calculated as the ratio between the actual air mass in the combustion chamber and the air mass at normalized conditions ( $P_{ref}=1013\text{hPa}$ ,  $T_{ref}=273\text{K}$ ). Such temperature is multiplied by two gains that consider the effect of a different MFB50 (MFB\_GAIN block) and lambda ( $\lambda$ \_GAIN block), respectively. The result of such product is the gas temperature within the exhaust manifold (i.e., the one that would be measured with a MIMS TC under steady-state conditions). The MFB50 index is calculated by an empirical model that was discussed in a previous work of the authors [3.11], in order to produce an algorithm that does not need indicating indexes as input and can therefore be implemented in an engine controller. This makes the proposed approach a valid solution also for on-board applications. In the following paragraphs, each block of the analytical exhaust gas temperature model is described in detail.

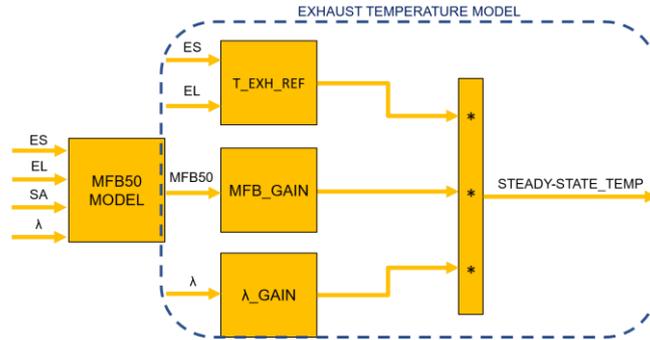


Figure 3.2: Block-layout of the exhaust gas temperature model. The reference temperature is multiplied by two factors that depend on MFB50 and lambda values respectively. The MFB50 model from [3.11] allows to calculate the combustion phase without the need for indicating signals.

### 3.3.1. Reference Temperature Model

Every spark sweep was carried out by applying wide SA variations while maintaining lambda at its calibrated value. In this way, it was possible to calculate exhaust gas reference temperatures on the entire engine operating range. These temperatures represent the ones that would theoretically be measured within the exhaust manifold under steady-state conditions, when lambda is equal to the mapped value and the normalized MFB50 is equal to a fixed reference value. MFB50 is evaluated in terms of arbitrary units (for which the unitary variation corresponds to a certain amount of crank angle degrees) while the temperatures were normalized with respect to the maximum and indicated as percentage values (considering degrees Celsius before normalization). The reference combustion phase was assumed equal to 2 because all the spark-sweeps were centered around such value. In this way, the calculation of the corresponding temperature was always done through interpolation of the experimental values, without performing an extrapolation that could be a source of error. The trend of these temperatures with respect to engine speed and load is shown in Figure 3.3, where the blue dots indicate the values calculated through the interpolation of adjacent experimental data and the surface represents the overall fitting function. The interpolation process is needed due to missing experimental temperature values for MFB50 exactly equal to 2. The effect caused by the combustion phase is then accurately excluded in this way from the reference temperature definition. With this function, the influence of the engine speed and load is introduced in the exhaust gas temperature model.

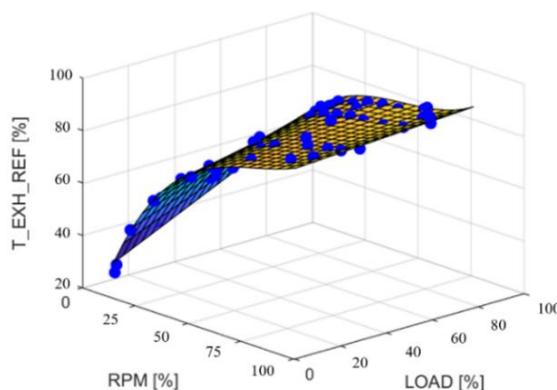


Figure 3.3: Trend of the reference temperatures (lambda=map, MFB50=2) on the normalized RPM-Load range. The blue dots represent values corresponding to the MFB50=2 calculated through the interpolation of the experimental data, while the surface is the poly31 fitting function.

The choice of the polynomial order is carried out by identifying the optimal compromise between the fitting performance and the function complexity. If, by increasing the polynomial order, the accuracy improvement is negligible, the last formulation that guaranteed a significant performance step is chosen. More generally, the definition of the polynomial order is also driven by the analysis of the independent variable trend. The goodness of the fitting process has been evaluated through the R2 and the RMSE indexes defined by Equations 3.2 and 3.3 [3.20]. Table 3.1 collects polynomial order, R2, and Root-Mean-Squared-Error (RMSE) for the reference temperatures (correlation and errors are evaluated between experimental and fitted values). The last polynomial order that guarantees an increment higher than  $10^{-2}$  in terms of R2 is chosen for the fitting process. The importance of selecting the as low as possible polynomial order is not only related to the computational effort needed by the model execution. Indeed, major complexity comes from the numerical optimization of power function for final implementation in RT machine. Second or third power force programmer to declare the output of that function with a wide number of digits and this opposes the necessity to minimize the impact on RAM of Rapid Control Prototyping (RCP) system. Such effort must be as low as possible. The RMSE index is calculated for percentage quantities (normalized for confidentiality reasons). Thus, the output of Equation 3.3 is a percentage value. The nomenclature '*polylv*' indicates that the polynomial is characterized by a degree *l* for the first independent variable (*X*=engine speed) and a degree *v* for the second one (*Y*=engine load). All the possible order combinations were tested from *poly11* to *poly33*. Equation 3.1 represents the analytical form of the fitting function (*poly31*):

$$T_{EXH\_REF} = p_{00} + p_{10}X + p_{01}Y + p_{20}X^2 + p_{11}XY + p_{30}X^3 + p_{21}X^2Y \quad (3.1)$$

In which:

- $T_{EXH\_REF}$  is the dependent variable
- $X$  and  $Y$  represent engine speed and load
- $p_{lv}$  is the coefficient for the element in which there is a degree  $l$  for  $X$  and a degree  $v$  for  $Y$ .

Equations 3.2 and 3.3 specify R<sup>2</sup> and RMSE analytical formulations, respectively:

$$R^2 = \left( \frac{\sum (c_i - \bar{c})(e_i - \bar{e})}{N (\sigma_c \sigma_e)} \right)^2 \quad (3.2)$$

$$RMSE = \sqrt{\frac{\sum (c_i - e_i)^2}{N}} \quad (3.3)$$

Where:

- $N$  is the number of samples
- $c_i/e_i$  is the generic calculated/experimental value
- $\bar{c}/\bar{e}$  is the calculated/experimental mean value
- $\sigma_c/\sigma_e$  is the calculated/experimental standard deviation

Table 3.1: Sensitivity analysis to the fitting polynomial order, for the reference temperature. The green row highlights the best polynomial order.

Polynomial	R2	RMSE [%]
poly11	0.828	8.331
poly12	0.926	5.562
poly21	0.972	3.381
poly13	0.927	5.612
poly31	0.995	1.903
poly22	0.974	3.301
poly23	0.985	2.588
poly32	0.996	1.813
poly33	0.997	1.767

The robustness of this fitting process through a *poly31* polynomial was tested by reducing the number of engine points used for the calibration process and by verifying the performance on the total amount of experimental data. Figure 3.4 shows R2 values for the fittings with a different number of used engine points: it is possible to highlight that the accuracy of the model does not significantly change when a lower number of spark sweeps is used for the calibration. By reducing the calibration data, the engine points are still selected in order to be spread out enough to cover the whole engine operating range.

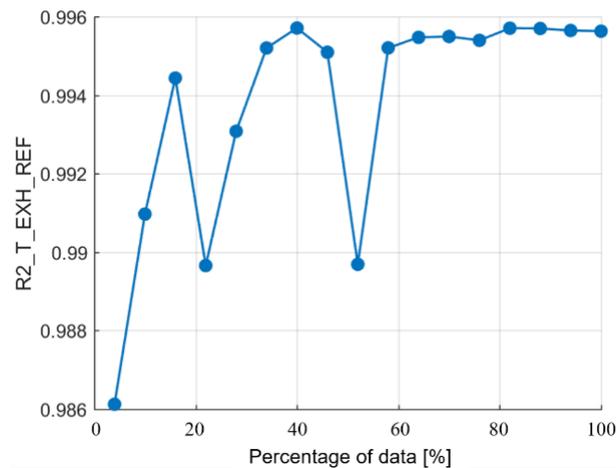


Figure 3.4: R2 index calculated with a different number of experimental points used for the model calibration. The percentage values indicate the amount of engine points used for fitting procedure with respect to the entire database. 100% corresponds to 60 engine points (i.e. 60 spark sweeps), 20% to 12.

### 3.3.2. MFB50 Dependency Model

The next step in the development of the analytical model consists of the analysis of the spark-sweeps database and for each tested operating condition (fixed engine speed and load) the exhaust gas temperature trend for the mean cylinder was studied and fitted with a polynomial equation with respect to the combustion phase. Figure 3.5 displays the average temperature for different SA values recorded under steady-state conditions (blue dots) and the fitting curve (red line), for the engine point at 4000 rpm and full load.

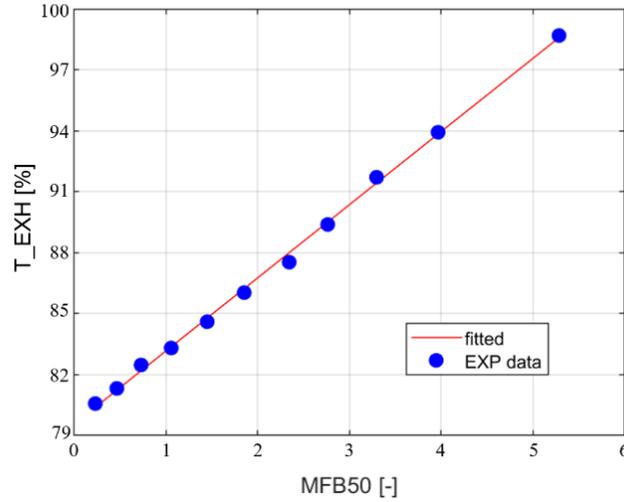


Figure 3.5: Exhaust manifold gas temperature as a function of combustion phase (MFB50), for fixed engine speed and load. The blue dots and the red line represent the experimental values logged for different values of the SA and the fitting polynomial function respectively.

As observable from Figure 3.5, the polynomial degree is equal to 1 (linear function), due to the accuracy already provided by such kind of function. For such fitting process, the mean R2 values and the RMSE indexes for all engine points have been collected in Table 3.2.

Table 3.2: R2 and RMSE mean values for all spark-sweeps. The exhaust gas temperature was fitted as a function of MFB50 with a linear polynomial function, for each tested engine speed and load.

R2	RMSE [%]
0.987	0.472

The function chosen for the fitting has the simple form shown in Equation 3.4:

$$Texh = p_0 + p_1 MFB50 \quad (3.4)$$

Where:

- $Texh$  is the dependent variable (the gas temperature in the exhaust manifold, as measured with the TC)
- $MFB50$  is the combustion phase
- $p_l$  is the coefficient that multiplies the combustion phase at the  $l$  power.

This polynomial function can be considered as a corrective function of the modeled temperature when the combustion phase changes. In fact, by fixing an arbitrary reference value of MFB50 and normalizing the polynomial function with respect to temperature value that corresponds to such reference, it is possible to obtain a new function that defines the multiplying factor. In other words, when MFB50 is equal to the reference value the gain is equal to 1, while if MFB50 is lower or higher than the reference, the gain is lower or higher than 1, respectively. Figure 3.6 shows the same data as in Figure 3.5 after the normalization with respect to the value of the temperature that corresponds to the MFB50 equal to 2. The red line represents the fitting function defined by Equation 3.5:

$$MFB\_GAIN = p_{0\_MFB} + p_{1\_MFB} MFB50 \quad (3.5)$$

Where  $p_{0\_MFB}$  and  $p_{1\_MFB}$  are the coefficients of the MFB\_GAIN function. This procedure allows to identify the gas temperature along the exhaust manifold for fixed operating conditions (engine speed, load, the calibrated - or nominal - lambda value, and any given value of the combustion phase), according to the following Equation 3.6:

$$T_{exh} = T_{EXH\_REF}(ES, EL) * MFB\_GAIN(MFB50) \quad (3.6)$$

Where  $T_{EXH\_REF}$  indicates the reference temperature, as defined by Equation 3.1 in the previous section, at reference (mapped) SA and AFR, for any given engine speed and load ( $ES$  and  $EL$  respectively). The new term in Equation 3.6 implements the combustion phase dependency, which is now added to the model.

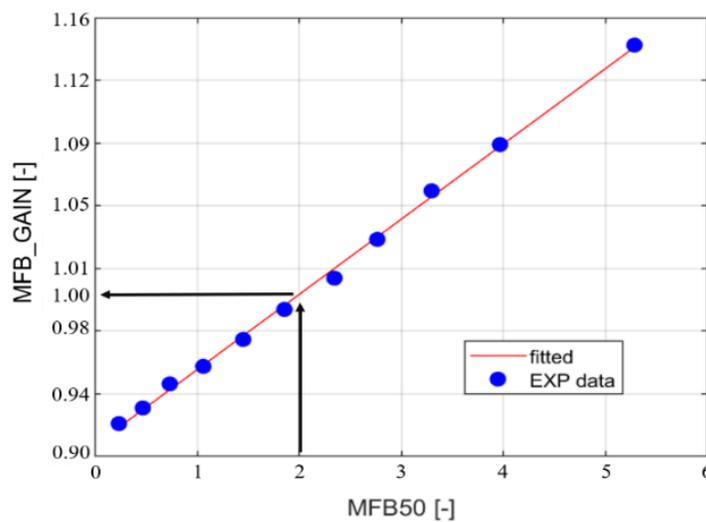


Figure 3.6: Exhaust gas temperature is normalized with respect to the value in correspondence with the reference MFB50. Thus, the red line represents the correction function of the temperature for a given operating condition.

The MFB\_GAIN fitting function was then considered and the coefficients  $p_{0\_MFB}$  and  $p_{1\_MFB}$  were hence separately identified for all the 60 tested engine points. Such analysis demonstrated that the numerical values for gains and offsets are all very similar. This means that the trend of such coefficients with respect to the engine speed and load can be neglected. This observation allows the use of mean  $p_{0\_MFB}$  and  $p_{1\_MFB}$  on the entire engine operating range. Figures 3.7 and 3.8 display such values for each test and the indicated numbers refer to the normalized temperatures fitted with respect to the MFB50 values considered as arbitrary unit. In such figures there are two red dashed lines that indicate the upper and lower limit of the rounded dispersion of data.

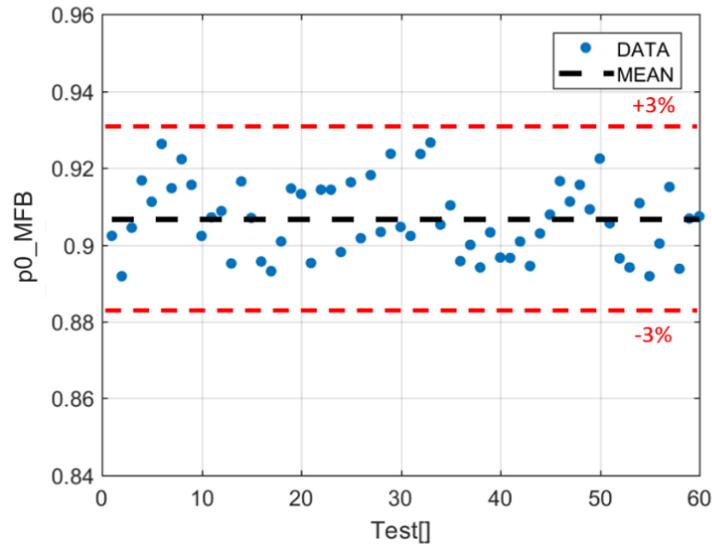


Figure 3.7: Numerical values of coefficient  $p0\_MFB$  identified for each spark-sweep. The black dashed line indicates the mean value, while the two red dashed lines represent the upper and lower limit of the rounded dispersion of data, which lie within  $\pm 3\%$ .

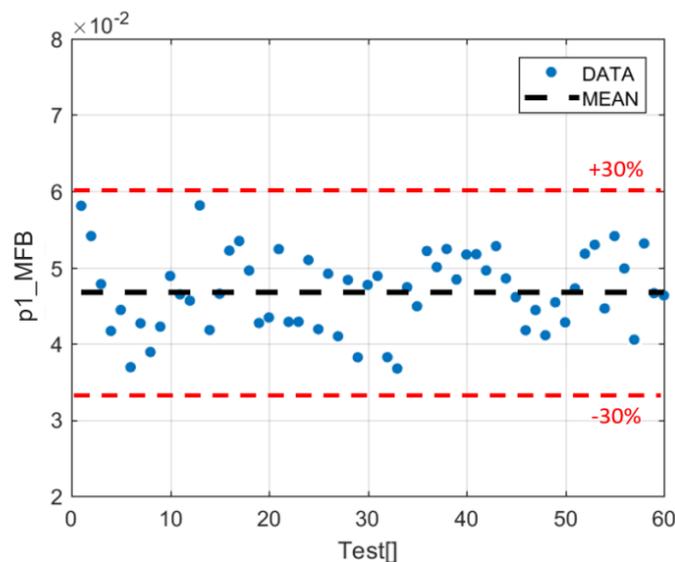


Figure 3.8: Numerical values of the coefficient  $p1\_MFB$  for each spark-sweep. The black dashed line indicates the mean value, while the two red dashed lines represent the upper and lower limit of the rounded dispersion of data, which lie within  $\pm 30\%$ .

The absolute percentage error between each fitting function (the specific values of MFB50\_GAIN function calculated for each spark sweep) and the one that derives from the approximation of the coefficients with their mean value (the mean MFB50\_GAIN function) were then evaluated. The error using the mean value is equal to 0.43% and this result legitimates such simplification.

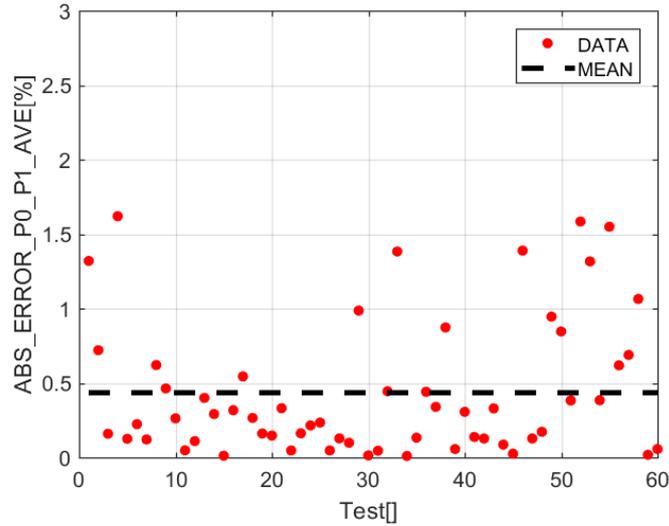


Figure 3.9: Absolute error for all the spark sweeps, given by the approximation of coefficients  $p_0$  and  $p_1$  with their corresponding mean value. The mean value of the error is equal to 0.43% and this result confirms the possibility of simplifying each fitting function with the average polynomial coefficients.

A sensitivity analysis was therefore carried out by imposing a fixed error on  $p_{0\_MFB}$  and  $p_{1\_MFB}$  and studying the resulting error on the MFB\_GAIN. Table 3.3 collects the mean error on the MFB\_GAIN function when a given percentage error is imposed on each coefficient. This study demonstrates that an approximated gain of the linear polynomial (the coefficient  $p_{1\_MFB}$ ) causes a much lower error on the resulting function than the one obtained with the same percentage error on the offset (the coefficient  $p_{0\_MFB}$ ). This observation allows to clarify how the approximation of both coefficients has the same effect on the final error, even if the dispersion of  $p_{1\_MFB}$  is ten times higher.

Table 3.3: Sensitivity analysis of the MFB\_GAIN function to the error imposed on the  $p_{0\_MFB}$  and  $p_{1\_MFB}$  mean values.

Error on P0_MFB [%]	Mean Error on MFB_GAIN [%]
10	1.44
20	5.97
30	9.50
40	14.03
50	18.57
60	23.10
Error on P1_MFB [%]	Mean Error on MFB_GAIN [%]
10	0.04
20	0.09
30	0.15
40	0.19
50	0.25
60	0.31

### 3.3.3. Lambda Dependency Model

The methodology used for AFR impact on exhaust gas temperature variations is similar to the approach used for the combustion phase. The experimental database consists in 14 lambda sweeps carried out on a portion of the engine points represented in Figure 3.1. Tests were performed mainly

on the portion of the engine operating field characterized by the highest speed and load, because it is in this portion of the engine map that turbine inlet temperatures need to be closely controlled. Thus, in these operating conditions it is particularly important to guarantee an accurate estimation of exhaust gas temperature. Figure 3.10 shows the engine points on which lambda sweeps were carried out. Such tests were performed while controlling both combustion phase and engine load. In fact, combustion phase and load are affected by AFR but, during this kind of test, it is important to separate the effect of AFR on the exhaust gas temperature from those of MFB50, engine speed and load. This was achieved by checking the effective load and the combustion phase when lambda was changed, by adjusting SA to maintain MFB50 close to the normalized reference value (equal to 2), and by keeping constant IMEP. Indeed, as already mentioned in the first section of this chapter, the proposed approach implements the effects-separation method. Such experimental campaign needs to collect data to analyze mixture quality effects on exhaust temperature, excluding the influence of combustion efficiency that depends on combustion phase. For this reason, MFB50 is maintained equal to the reference value used to calibrate the reference temperature model.

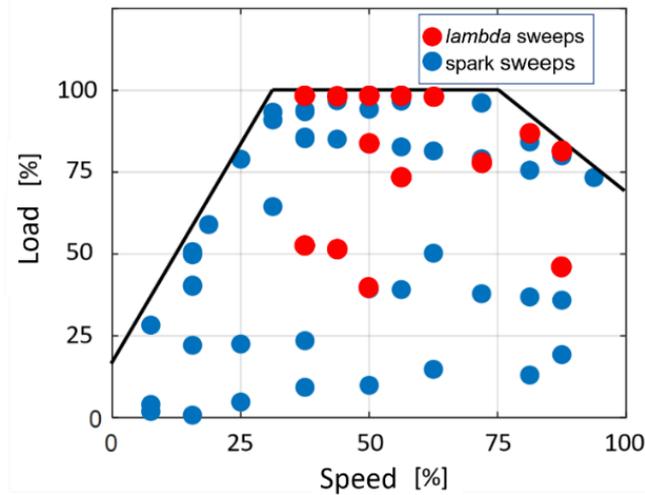


Figure 3.10: Red dots indicate the 14 engine points in which lambda sweeps were performed. The blue points represent the operating conditions for which a spark-sweep was carried out.

Exhaust gas temperatures recorded during steady state lambda sweeps were described through a linear polynomial. Temperatures were normalized with respect to mapped lambda conditions. Reference temperature for the calibrated lambda has to coincide with the value recorded during the spark sweep on the same engine point because lambda sweeps were carried out maintaining the MFB50 equal to 2. Figure 3.11 shows the function of the corrective factor that has unitary value when lambda matches the mapped value. The trend of the normalized exhaust gas temperatures on lambda was described through a linear function, even if for some engine points a degree 2 equation could improve R2 and RMSE indexes. Nevertheless, the magnitude of the improvement is weighted with the computational cost increase due to the calculation of some squared quantities that strongly impacts on the numerical optimization cost of the algorithm when it has to be implemented in a RT machine or in an ECU. For this reason, a simplified approach is preferred.

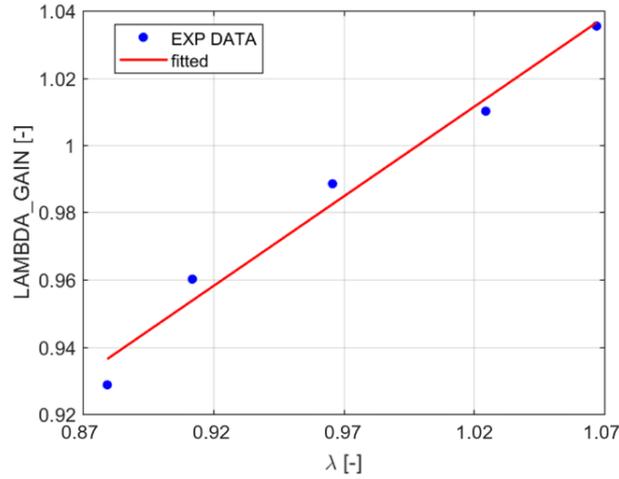


Figure 3.11: Exhaust gas temperatures were normalized with respect to the reference value. Blue dots indicate the normalized gas temperature values while the red line represents the corresponding fitting function.

The LAMBDA\_GAIN function is defined by Equation 3.7:

$$LAMBDA\_GAIN = p_{0\_LAMBDA} + p_{1\_LAMBDA}\lambda \quad (3.7)$$

Where  $p_{0\_LAMBDA}$  and  $p_{1\_LAMBDA}$  are the coefficients of the LAMBDA\_GAIN function. Thus, introducing lambda dependency, the exhaust gas temperature is calculated according to the following Equation 3.8:

$$T_{exh} = T_{EXH\_REF}(ES, EL) * LAMBDA\_GAIN(\lambda) * MFB\_GAIN(MFB50) \quad (3.8)$$

Where  $T_{EXH\_REF}$  indicates the reference temperature, that is defined as a function of engine speed ( $ES$ ) and load ( $EL$ ) in nominal conditions. The selected LAMBDA\_GAIN function is characterized by a polynomial degree equal to 1 and the accuracy of the fitting process was evaluated through R2 and RMSE indexes. The mean values for all the 14 tested engine points are collected in Table 3.4.

Table 3.4: R2 and RMSE indexes mean values. The normalized exhaust gas temperatures were described with a linear polynomial function.

R2	RMSE [%]
0.971	0.702

Coefficients  $p_{0\_LAMBDA}$  and  $p_{1\_LAMBDA}$  were analyzed and the corresponding mean values were calculated. The error caused by the approximation of all the polynomials with the average one was assessed: figures 3.12 and 3.13 display the values of the coefficients  $p_{0\_LAMBDA}$  and  $p_{1\_LAMBDA}$  for each lambda sweep and the value of the rounded maximum percentage dispersion.

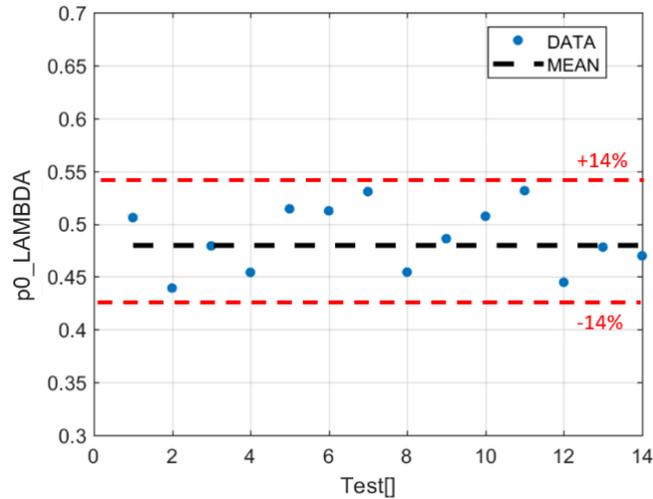


Figure 3.12: Numerical values of the coefficient  $p0\_LAMBDA$  for each lambda sweep. The black dashed line indicates the mean value, while the two red dashed lines represent the rounded limits of the data dispersion, which lie within  $\pm 14\%$ .

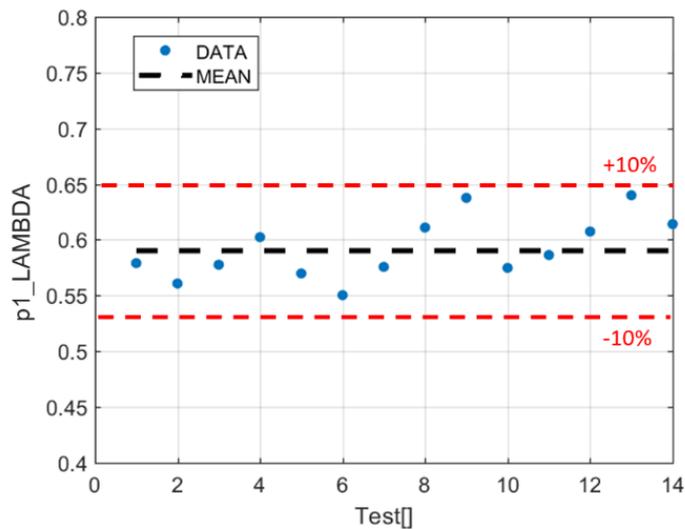


Figure 3.13: Numerical values of the coefficient  $p1\_LAMBDA$  for each lambda sweep. The black dashed line indicates the mean value, while the two red dashed lines represent the rounded limits of the data dispersion, which lie within  $\pm 10\%$ .

As for the MFB50\_GAIN block, the accuracy of the mean function (the function defined by the mean offset and gain) was checked by calculating the error for each lambda sweep and carrying out a sensitivity analysis of the LAMBDA\_GAIN to the imposed error on the coefficients  $p0\_LAMBDA$  and  $p1\_LAMBDA$ . The error is defined as the mean absolute value of the difference between each LAMBDA\_GAIN function obtained for every lambda sweep and the mean function. Figure 3.14 shows such error for each test and Table 3.5 collects data about the sensitivity analysis.

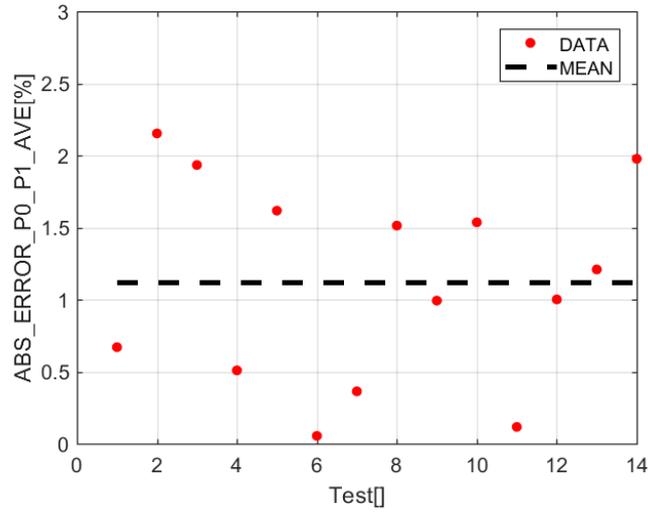


Figure 3.14: Absolute error for each lambda sweep given by approximation of coefficients  $p0\_LAMBDA$  and  $p1\_LAMBDA$  with the corresponding mean value. The mean error is equal to 1.11%: this result confirms the possibility to simplify each fitting function with the average polynomial.

Table 3.5: Sensitivity analysis of  $LAMBDA\_GAIN$  function to the error imposed on  $p0\_LAMBDA$  and  $p1\_LAMBDA$  mean values.

Error on P0_LAMBDA [%]	Mean Error on MFB_GAIN [%]
10	0.77
20	1.51
30	2.25
40	3.00
50	3.72
60	4.46
Error on P1_LAMBDA [%]	Mean Error on MFB_GAIN [%]
10	0.23
20	0.98
30	1.63
40	2.41
50	3.20
60	3.99

### 3.4. Model Validation Under Steady-State Conditions

The complete algorithm composed by the reference temperature model, the MFB\_GAIN, and the LAMBDA\_GAIN functions, was then validated through comparison of steady-state temperatures related to the whole experimental database and the corresponding modelled values. The error between all recorded exhaust gas temperatures (for both spark and *lambda* sweeps) and the values calculated by supplying the model with the mean control and indicating parameters (engine speed, load, MFB50 and *lambda*) is shown in Figure 3.15. Every point represents the difference between the steady state TC measurement and the calculated value, for each engine operating condition. Results confirm the accuracy of the model for all tested engine speed-load conditions, and also for all the actuated SA and *lambda* values.

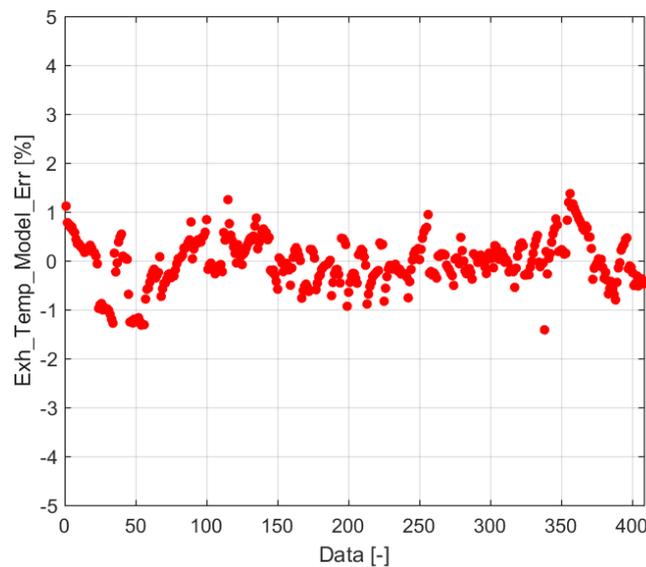


Figure 3.15: Error between measured and modelled steady-state exhaust gas temperatures.

#### 3.4.1. TC Dynamics Model

This section deals with the development of the TC dynamics model and the method used for its implementation within Simulink. Such model uses the steady-state exhaust gas temperature as input, and it applies a calibratable filter to reproduce TC dynamics during transient conditions. In this way the calculated temperature can be directly compared with the TC signal. Figure 3.16 shows a qualitative trend of the gas temperature measured with a TC when the engine operating condition is changed drastically. The real exhaust gas temperature changes quickly but the TC output is affected by its own dynamics which include mainly its time constant, conduction and radiation between TC tip and manifold wall. Moreover, as introduced in the first part of this work, also gas mixing within exhaust runners and manifold influences the temperature trend under transient conditions.

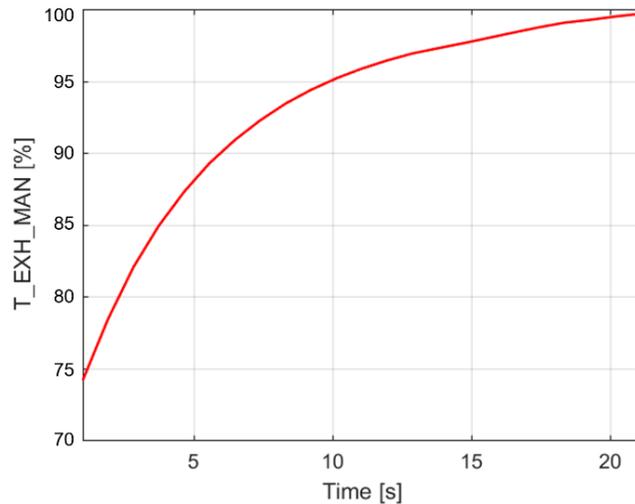


Figure 3.16: Normalized TC temperature trend during a fast transient condition.

The behavior of the TC output is initially characterized by a fast response, followed by a slower transient to reach steady state [3.18]. The TC time constant and the gas mixing within the exhaust manifold cause the faster part of the dynamics, while conduction and radiation phenomena are responsible for the slower one. The approach used to reproduce such trend is based on the algebraic sum of the output of two different filters, that were implemented in the form of two weighted moving averages. The one with the lower weight corresponds to the faster filter, and the one with the higher weight corresponds to the slower one. Through a proper calibration it is possible to reproduce TC dynamics. For both moving averages, the weight applied to the current value was defined as the value complementary to 1 w.r.t. the calibrated weight. The output of each moving average is then weighted further before summing the faster and the slower contribution, in order to calculate the final modeled temperature. These last two weights are again defined in such a way that their sum is equal to 1.

The calibration parameters of such TC dynamic model are the weight of the old value for the faster and the slower moving average, and the one relative to the sum of these two contributions. Figure 3.17 and 3.18 display the Simulink representation of the TC dynamics model. Figure 3.17 shows how the moving averages were implemented in the simulation environment, and the calibratable weight is highlighted. Such parameter is defined as a percentage number and it indicates the influence of the old value in the calculation of the output, with respect to the current value. The difference between such weight divided by 100 and the constant 1 defines the weight applied to the current value. A switch block was introduced to avoid the propagation of an inconsistent output. Figure 3.18 shows an overview of the sum of the output of the faster and slower moving averages. The calibratable weight was arbitrarily assigned to the output of the slower filter.

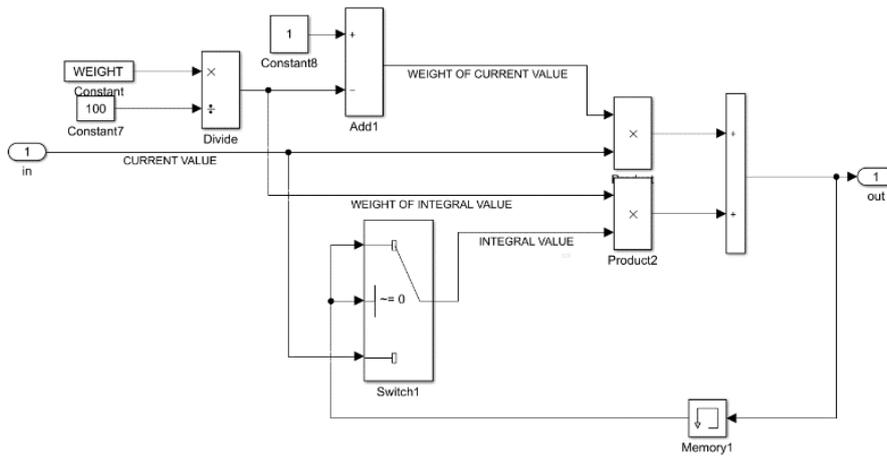


Figure 3.17: Simulink block scheme used to implement moving averages.

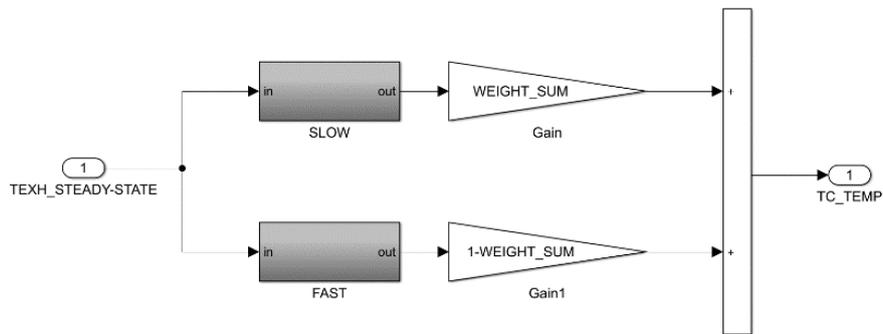


Figure 3.18: Simulink blocks used for the sum of moving the average outputs.

### 3.5. Model Validation under Transient Conditions

The TC dynamics model was coupled with the analytical one and the complete model was validated by comparing the measured and the calculated exhaust manifold temperatures under transient conditions. The parameters of the TC dynamic model were calibrated with dedicated tests. However, it is important to mention that the focus of this section is mainly to validate the analytical exhaust gas temperature model, and the TC model is useful in order to have an output signal directly comparable with the actual TC reading. The reason is that the final purpose of this model is to be implemented in a control strategy for the RT estimation of the gas temperature at the turbine inlet, and for this kind of application the TC dynamics model can be excluded, due to the underestimation of the gas temperature under transient conditions. The experimental database used for the validation of the complete models is composed of two tests performed at the test bench, characterized by different engine speed and load transients. Moreover, some SA and lambda steps were externally imposed to verify the quality of the combustion phase and AFR dependencies modelling.

The steady-state exhaust manifold temperature calculated by the analytical model was used as input for the TC dynamic one. Figure 3.19 shows the block scheme of the complete model.



Figure 3.19: The exhaust manifold temperature model was coupled with the TC dynamic model in order to reproduce the TC signal also under transient conditions.

The engine and the complete analytical model were supplied with the same speed, load, SA, and lambda profiles, and the experimental and modelled gas temperatures were compared. Two different tests were carried out in order to cover the high speed-high load operating range. This is because such area is the critical one for inlet turbine temperature: an accurate prediction of the exhaust gas temperature is crucial. Tests under transient conditions allow to stress the model also on engine points that are not included in the calibration database. Achieving a limited estimation error on the entire test is a further demonstration of the validity of the proposed method.

#### 3.5.1. Results and Discussion

Figures 3.20 and 3.22 show the main actuations of each test (engine speed, load, SA, and lambda), while in Figures 3.21 and 3.23 the corresponding calculated and experimental temperatures are shown. As for MFB50, also SA is evaluated in terms of arbitrary units, for which the unitary variation corresponds to a certain amount of crank angle degrees. Error in figures 3.21 and 3.23 is calculated as the difference between experimental and calculated value.

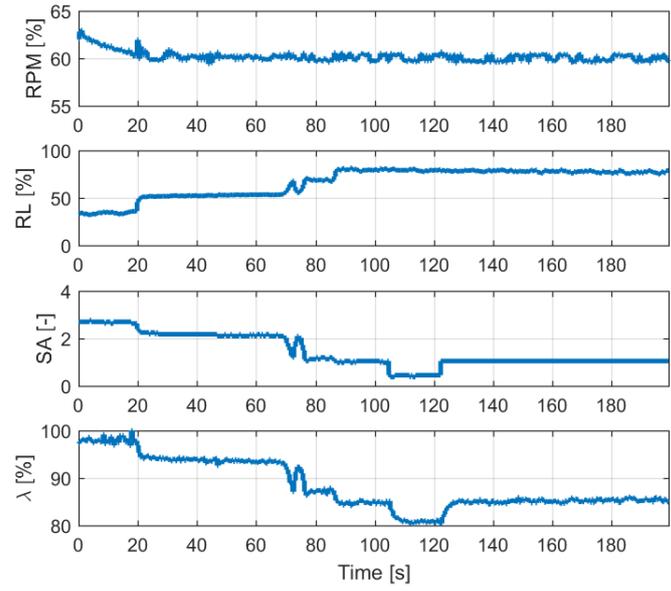


Figure 3.20: Normalized Engine speed, load, SA, and lambda profiles for test 1.

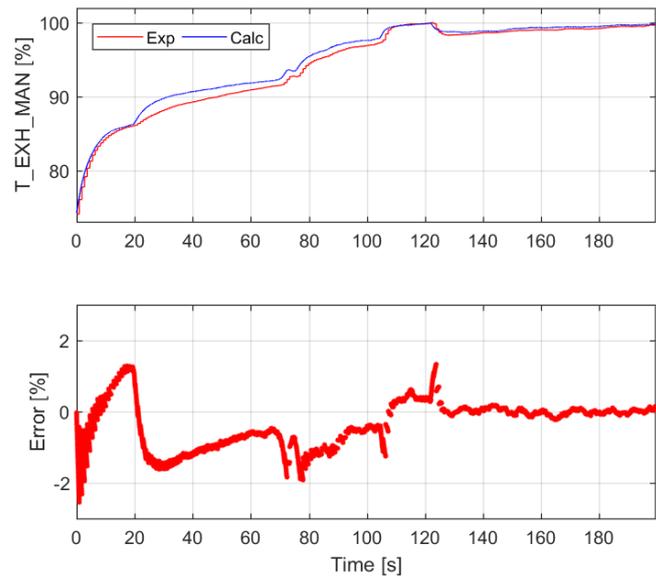


Figure 3.21: Performance of the complete analytical model under transient conditions (test 1).

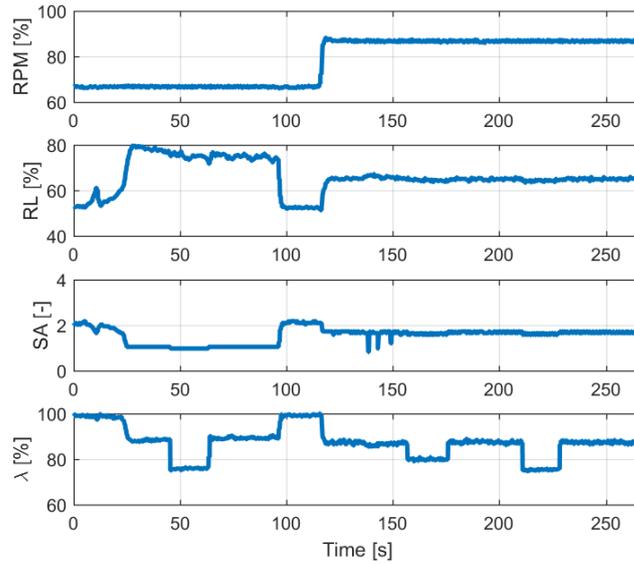


Figure 3.22: Normalized Engine speed, load, SA, and lambda profiles for test 2.

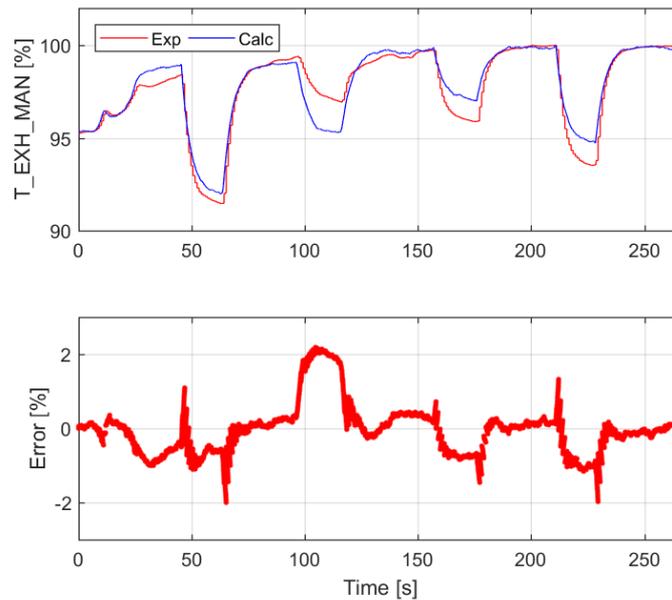


Figure 3.23: Performance of the complete analytical model under transient conditions (test 2).

The results clearly confirm the high accuracy and reliability of the complete model. The error is typically within the range  $\pm 2\%$ . However, it can be observed that the widest percentage error is caused by lambda steps, especially for the test 2. This is due to the linear trend of the LAMBDA\_GAIN function but, as anticipated above, the extremely low value of the error does not justify an increase in the function complexity. In fact, the mean absolute error for the validation tests is equal to 1.35%, proving the high accuracy of the proposed approach. Due to the type of tests designed for the model validation, this value represents the average error obtainable over the entire operating range. The discontinuous shape exhibited by the TC signal during fast transients is due to the low frequency temperature sampling, and this causes an unavoidable gap between the calculated and the experimental signals. This means the error can be reduced further by increasing the acquisition rate.

The ratio between the actual and the model execution time interval is equal to 312. This implies that such model can be executed 312 times faster than real time, on a standard laptop equipped with an Intel Core i7 6600U CPU and 8GB of RAM. Such result demonstrates that the developed model is particularly suitable to be implemented in a RT control strategy, due to the extremely low computational effort required for its execution.

### 3.6. Conclusions

A novel analytical approach for the exhaust gas temperature modelling is investigated in this chapter of the document. The empirical method described in chapter 1 for combustion index modelling is extended to obtain an accurate and reliable model for the exhaust gas temperature estimation under steady-state conditions, and an analytical model of the sensor dynamics was developed to extend the validity of the approach also under transient conditions. The dependencies to SA and lambda are introduced and a wide experimental database is collected for the calibration process. Moreover, the robustness of the proposed approach is demonstrated through a sensitivity analysis of the model accuracy w.r.t. the number of engine points used for the calibration.

The analytical approach is based on the analysis of the effects of the main control parameters (SA and lambda) that influence the exhaust gas temperature, and on their modelling by implementing the effects-separation method. The effects of engine speed and load on the exhaust manifold temperature were captured with a relatively low-degree polynomial function that reproduces the experimental data for fixed MFB50 and for the mapped values of lambda. Linear polynomial functions for the calculation of the multiplying factors to adjust the reference temperatures with respect to a different MFB50 or lambda were then calibrated and added to the Simulink model. The proposed empirical approach guarantees a high level of accuracy, with percentage errors included within the range of  $\pm 1\%$  under steady-state conditions. Such result is even more remarkable considering the extremely low level of complexity of the resulting code. The very simple modelling approach makes such model particularly suitable to be implemented in control applications designed to manage the turbine inlet gas temperature in Turbo-Charged engines.

A control-oriented TC dynamics model is also introduced in this chapter to reproduce the TC signal also under transient conditions. In fact, TC dynamics is affected by physical phenomena such as the TC thermal inertia, gas mixing within exhaust runners and manifold, and conduction and radiation heat exchanges between the TC tip and the pipe wall, factors which act like low-pass filters during fast temperature transients. The TC behavior was modelled with two weighted moving averages that were calibrated to work like a faster and a slower response on the input signal, respectively. The outputs of these filters were then weighted and added in order to obtain the resulting signal. Nevertheless, the TC dynamics model was developed and implemented for simulation purposes only, because it leads to underestimating the real exhaust gas temperature under transient conditions. For this reason, such model would be excluded from a possible implementation in a RT control strategy. The TC dynamic model is finally coupled with the exhaust gas temperature model and calibrated to minimize the error between the experimental and the calculated signal. The complete model is tested under transient conditions, comparing the TC and the modelled temperature profiles, and calculating the percentage error. Two different tests were carried out at the test bench, by varying the main parameters that influence the exhaust gas temperature, i.e. engine speed and load, SA, and lambda. Also in this case, the accuracy and the robustness of the complete model is demonstrated, registering absolute percentage errors mainly below 2%, with a mean absolute error equal to 1.35%. Moreover, the complete model can be executed 312 times faster than the real time, on a standard laptop, confirming it is particularly suitable for RT application. The analytical exhaust gas temperature model can be easily reversed to convert a target value of the exhaust gas temperature at the turbine inlet into a MFB50 value that can be used as target for a combustion controller. It can be thus implemented in the adaptive damage-based combustion controller as the second “actor” that can calculate a target MFB50: the controller will end up targeting the highest value between that calculated by the piston damage model and that calculated by the exhaust gas temperature model.

## References

- 3.1. Brusa, A., Mecagni, J., Cavina, N., Corti, E. et al., "Development and Experimental Validation of a Control-Oriented Empirical Exhaust Gas Temperature Model," SAE Technical Paper 2020-24-0008, 2020.
- 3.2. Fu, H., Chen, X., Shilling, I., and Richardson, S., "A One-Dimensional Model for Heat Transfer in Engine Exhaust Systems," SAE Technical Paper 2005-01-0696, 2005, <https://doi.org/10.4271/2005-01-0696>.
- 3.3. Regulation (EU) 2019/631, European Commission, [https://ec.europa.eu/clima/policies/transport/vehicles/regulation\\_en](https://ec.europa.eu/clima/policies/transport/vehicles/regulation_en).
- 3.4. Zhao, F., "Technologies for Near-Zero-Emission Gasoline-Powered Vehicles", Warrendale, Society of Automotive Engineers, Inc., (2006), 10.4271/R-359, ISBN 978-0-7680-4301-3.
- 3.5. Eriksson, L., "Mean Value Models for Exhaust System Temperatures," SAE Technical Paper 2002-01-0374, 2002, <https://doi.org/10.4271/2002-01-0374>.
- 3.6. Nasser, S. and Playfoot, B., "A Turbocharger Selection Computer Model," SAE Technical Paper 1999-01-0559, 1999, <https://doi.org/10.4271/1999-01-0559>.
- 3.7. Fulton, B., Van Nieuwstadt, M., Petrovic, S., and Roettger, D., "Exhaust Manifold Temperature Observer Model," SAE Technical Paper 2014-01-1155, 2014, doi:10.4271/2014-01-1155.
- 3.8. Dahlstrom, J., Andersson, O., Tuner, M., and Persson, H., "Experimental Comparison of Heat Losses in Stepped-Bowl and Re-Entrant Combustion Chambers in a Light Duty Diesel Engine," SAE Technical Paper 2016-01-0732, 2016, doi:10.4271/2016-01-0732.
- 3.9. Olmeda, P., Martin, J., Garcia, A., Blanco, D. et al., "Evaluation of EGR Effect on the Global Energy Balance of a High Speed DI Diesel Engine," SAE Technical Paper 2016-01-0646, 2016, doi:10.4271/2016-01-0646.
- 3.10. Smith, L., Preston, W., Dowd, G., Taylor, O. et al., "Application of a First Law Heat Balance Method to a Turbocharged Automotive Diesel Engine," SAE Technical Paper 2009-01-2744, 2009, doi:10.4271/2009-01-2744.
- 3.11. Brusa, A., Cavina, N., Rojo, N., Cucchi, M. et al., "Development and Validation of a Control-Oriented Analytic Engine Simulator," SAE Technical Paper 2019-24-0002, 2019, doi:10.4271/2019-24-0002.
- 3.12. Malbec, L., Le Berr, F., Richard, S., Font, G. et al., "Modelling Turbocharged Spark-Ignition Engines: Towards Predictive Real Time Simulators," SAE Technical Paper 2009-01-0675, 2009, doi:10.4271/2009-01-0675.
- 3.13. Martin, D. and Rocci, B., "Virtual Exhaust Gas Temperature Measurement," SAE Technical Paper 2017-01-1065, 2017, doi:10.4271/2017-01-1065.
- 3.14. Son, S. and Kolasa, A., "Estimating Actual Exhaust Gas Temperature from Raw Thermocouple Measurements Acquired During Transient and Steady State Engine Dynamometer Tests," SAE Technical Paper 2007-01-0335, 2007, <https://doi.org/10.4271/2007-01-0335>.
- 3.15. S. V. Patankar, "Numerical Heat Transfer and Fluid Flow", 1980, ISBN: 978-0070487406.
- 3.16. Cavina, N., "Measurement of Exhaust Gas Temperatures: Theoretical and Experimental Analysis", Proceedings of the ASME 2002 Internal Combustion Engine Division Fall Technical Conference, New Orleans, Louisiana, USA, September 8–11, 2002, pp. 557-565, ASME, <https://doi.org/10.1115/ICEF2002-539>.
- 3.17. Papaioannou, N., Leach, F., and Davy, M., "Effect of Thermocouple Size on the Measurement of Exhaust Gas Temperature in Internal Combustion Engines," SAE Technical Paper 2018-01-1765, 2018, doi:10.4271/2018-01-1765.

- 3.18. C. D. Henning and R. Parker, "Transient response of an intrinsic thermocouple", J. Heat Transfer. May 1967, 89(2): 146-152, <https://doi.org/10.1115/1.3614337>.
- 3.19. Gat, U., Kammer, D., Hahn, O. J., "The effect of temperature dependent properties on transient measurement with intrinsic thermocouple", International Journal of Heat and Mass Transfer, Volume 18, Issue 12, December 1975, Pages 1337-1342, [https://doi.org/10.1016/0017-9310\(75\)90246-X](https://doi.org/10.1016/0017-9310(75)90246-X).
- 3.20. Matlab Documentation, The Mathworks.

## 4. Conclusions

Topics described in this document may look like unconnected from each other, but they have a common thread that is the increase of combustion efficiency in modern SI engines.

Innovative solutions for knock intensity measurement and control have been realized starting from the white paper and validated at the engine test bench, dealing with the entire development process of the algorithm. Issues related to the exhaust gas temperature management in modern ICE are even analyzed and deepened, developing a control-oriented model for the RT estimation of gas temperature at the turbine inlet.

Generally speaking, a novel analytical methodology for analysis and description of trend of main indexes related to the combustion process has been defined and applied to a very wide range of engine variables. At first, such method has been used to model some synthetic combustion indexes demonstrating the high accuracy and reliability of the proposed approach. Low computational effort needed by such models makes them particularly suitable to be implemented in an advance combustion phase controller, where combustion indexes that are not available on final on-board application are calculated by specific models. The controller implements an innovative concept that couples knock intensity management (evaluated as piston surface erosion) with the combustion efficiency optimization. Analytical approach has been then applied to other activities, such as the exhaust gas temperature modelling, providing again a powerful solution for the development of control-oriented models.

On the other hand, prototyping solutions for knock intensity mitigation and exhaust gas temperature reduction are realized, tested, and controlled. A prototyping Port Water Injection system and the related model-based controller have been developed in the first phase of PhD course. Such experience allowed to study the impact of a non-native technology on combustion process, even in terms of integration of an additional control algorithm within a standard combustion control system that runs in the ECU. Moreover, main steps for the realization of prototyping components have been experienced.

These works have been always coupled with several hours of experimental activities at the test bench, that lead even to the development of novel solutions with the final purpose to improve available tools and solve, when possible, some faced problems. The most relevant goal is that of the custom RT machine building, with which a powerful, cheap, and user-friendly target PC has been realized. Custom RT target device allowed even the fast experimental validation of complex prototyping control algorithm, without spending days for a long numerical optimization of the code. A development CAN network with which RCP machine receives all variables needed and overwrites some ECU control parameters has been realized at the test bench to rapidly implement and test custom control algorithm. The RT machine and CAN network development process is described in detail in the following Appendix.

Finally, it can be asserted that PhD course has been translated into a significant experience in SI engine experimental field and control system development and testing.

## 5. Appendix: Simulink Real Time Target Machine Development

## 5.1. Introduction and Motivations

Increasing complexity of engine control algorithms and the always more numerous sensors and actuators mounted on modern ICE are forcing Control Unit (CU) developers and manufacturers to push up targets for computational and communication capabilities. At the same time, the number of on-vehicle control devices is increasing to manage all the different systems that compose a complete powertrain. Cost is thus the other key feature to build a competitive product.

New requirements for on-board CUs inevitably impact even on development hardware for automotive field, such as development Engine CU (ECU) and Real-Time (RT) target machines for Rapid Control Prototyping (RCP) applications or signal and data logging. During last years, majority of companies that produce this kind of device (National Instruments, Speedgoat, Matrox and Intel, for instance) are providing always more powerful modular machines with a connectivity layer that can be configured to best meet user needs. Of course, cost of the final product is typically very high (order of magnitude of  $10^4$  €) and the computational power remains limited by the fanless processing unit with which mainboards are commonly equipped, to guarantee the compactness standards. Limitations on CPU performance become limitations on port number for most demanding communication protocols.

A custom RT machine has been built during the last two years of PhD course to face mentioned lacks of commercial products. Device hardware is based on the motherboard of a desktop PC while the RT Operating System (RTOS) is the kernel of Simulink RT (SRT), compiled as a Microsoft Disk Operating System (MS DOS) application. Target machine has been configured with Controller Area Network (CAN) and analog and digital I/O ports, installing specific PCI boards. User defined driver blocks have been developed to expand Simulink RT library and a dedicated case for the Printed Circuit Board (PCB) has been 3-D printed.

Custom RT machine is based on Simulink xPC platform, that is part of a family of software products that can be used to create real-time systems. Nevertheless, Mathworks ended up developing and providing such tool at the end of 2017, when Speedgoat became the official producer of SRT compatible HWs. Documentation and support about xPC platform are thus very poor and all the tutorials have been delated from the network. In the R2018a Matlab release SRT kernel is maintained but both boot options for RTOS and the Simulink driver block library are partially updated for Speedgoat devices and boards. This meant that the utilization of a recent Matlab version needed the development of user defined driver blocks to extend device connectivity by installing production PCI boards of Speedgoat competitors, such as Softing, Humosoft, Contec and so on. For the same reason, bibliographic references of this chapter are very poor.

It is important to mention that the following description has not the final purpose to be an exhaustive “manual” of the Simulink Real-Time platform, but it wants to be just a discussion of features that have been figured out during the PhD course, due to the absolute lack of a related official documentation, derived from recent Mathworks collaboration with Speedgoat. Author thinks it is however an extremely powerful and cheap solution to build a handy RCP system that can be very useful for a wide range of applications.

Simulink RT xPC [4.1] is a platform to build custom RT devices directly programmable with Simulink with a connectivity layer that can be configured with external PCI boards compatible with driver block included in Simulink RT library. In addition to the hundreds of hardware driver blocks that ship with xPC third parties and the final user can create additional blocks and functionalities. The xPC Driver Authoring Tool [4.2] is a graphical user interface to code generation tools that

automates and simplifies the development of custom drivers. Such development platform is the tool used to introduce a powerful novel contribution to the xPC target system. Indeed, SRT kernel can be generated for a limited number of commercial motherboards, even if the main limit comes from the available drivers for ethernet controller, and not from CPU or other components of the PCB. One of the most relevant results has been thus achieved developing drivers for an external mini PCI express (mPCIe) ethernet board by using the Driver Authoring Tool. Such drivers can be then selected when the RTOS is generated. It is a key step and an important contribution to the old xPC platform because it removes the limit of the pre-installed ethernet controller and makes the SRT kernel compatible with any desktop PC motherboard. Indeed, two motherboards of different brands (with even different CPU brand) have been tested installing the external ethernet board and both demonstrated compatible with the RTOS.

Custom RT machine is undoubtedly a powerful tool for RCP applications, and it is currently utilized at the engine test bench of Industrial Engineering Department for the development and validation of innovative combustion control algorithm. Positives of this device can be summarized in the following points:

1. **Cost:** the final cost of a configured machine is lower than 2k€ but it can be further reduced by using an old and unused desktop PC. This solution is thus much cheaper than a commercial RT device. Fixed the cost of a commercial RT machine, a motherboard with a huge number of PCI slots and an extremely powerful CPU can be purchased.
2. **Computational power:** even using an old desktop PC motherboard the available computational power of a standard cooled CPU is however much higher than that provided by a RT machine processor. This means that a higher port number for serial-based communication protocols can be managed, w.r.t. standard target device. High performance of a desktop PC CPU allowed even to test the combustion control system described in section 1.4.1 on a 8-cylinders engine, that would not have been possible on a commercial target machine, without a long work of numerical optimization of the code. Even the Water-Injection combustion control system described in section 2.3 has been therefore validated at the test bench via RCP with SRT machine.
3. **Recycling:** as mentioned above, computational power of a CPU of an old desktop PC is undoubtedly higher than that of a fanless unit of a production RT target machine. This allows to build a really powerful device even with an unused PC that can be reutilized for this kind of application.
4. **Compatibility with Simulink RT Explorer:** SRT Explorer is the user interface that runs on the host PC and it allows to visualize connected HWs and browsing root directories of target boot disk, available compiled SRT applications that can be deployed on the target, application that is running on target, details of application execution such as the percentage CPU load and all the tools needed for the development of a instrumented panel for signal visualization and parameter manipulation.

RT xPC platform provides therefore the possibility to generate the .a21 file associated to the corresponding RT application for the remote management of signals and parameters via Universal Measurement and Calibration Protocol (XCP, [4.4]). Moreover, when the algorithm is validated, it can be compiled as a stand-alone application that is automatically executed when the target machine

is supplied. Such features demonstrate how capabilities of the system are comparable with those provided by the most widespread commercial products for the automotive field.

On the other hand, main negatives are synthesized below:

1. **FPGA:** installation of a PCI board with a development Field Programmable Gateway Array (FPGA) is not supported by all the types of motherboard. A FPGA board can be used independently to log and treat signals at high frequency and send synthetic indexes via serial communication to the RT machine. Nevertheless, configuration of a FPGA application with this system is a complicated process and this lack disables the execution of algorithm with a frequency over 100kHz (it is the maximum execution frequency of a basic Simulink model).
2. **Compactness:** even if a specific case that contains just the PC motherboard is used, machine volume is higher than that of a certain type of commercial fanless devices. This is not true when industrial PCs become the reference. Indeed, such HWs are sometimes used in the automotive field.
3. **Modularity:** limited number of PCI ports in standard PC motherboard acts as a limit for the maximum number of external boards that can be mounted.

However, weight of point number 2 and 3 is strongly related to the final application. For example, when the RT target machine is used for a specific application in an engine test room, as in our case, compactness and modularity are not relevant features, due to the wide available space and the need of a limited number of serial ports.

Hence, this chapter deals with the procedure used to build mentioned RT target machine, starting from the development of drivers for the external mPCIe ethernet board used to compile the SRT kernel, explaining how a RT application is built and deployed on target and then showing the final machine configuration.

## 5.2. xPC Driver Authoring Tool

The driver authoring environment works as a Simulink S-function [4.3] builder for the creation custom library blocks. Key elements of a S-function are three main routines:

- A start routine is called before model execution
- An outputs routine is called once per sample time
- A terminate routine is called after execution ends

Such tool has been used to develop driver block for the external PCI ethernet board with the Realtek 8125 controller, not available as a standard option for SRT kernel in the R17a Matlab release (Figure 5.1).



Figure 5.1: External mPCIe Ethernet board with Realtek 8125 controller.

The SRT kernel is thus compiled for a PC with that ethernet board set as the main port of the motherboard. In this way, installing mentioned board on any PC makes such motherboard compatible with SRT OS. Driver is composed by two different functions: the receiver converts a vector of message bits and a vector of time bits durations of the ethernet frame into an output signal of decimal numbers that change over time, while the sender block takes the vector of decimal numbers and prepares the message and time vector to be sent to the other device. The core of the driver code is provided by the board manufacturer in a CD-ROM and code has thus to be built according with inputs, outputs and parameters included in the native file.

Figure 5.2 shows the startup of `xpcdrivertool` command, where the user can define the driver name, input/output ports of the corresponding Simulink block, hidden and editable parameters, work variables and related C file template and block that have to be generated. In this case just the C file is built, because such driver has not to be used in a Simulink model, but it is only a file that is then added in a specific directory of Matlab installation folder. However, when the S-function is ready, the corresponding driver block can be generated in any moment.

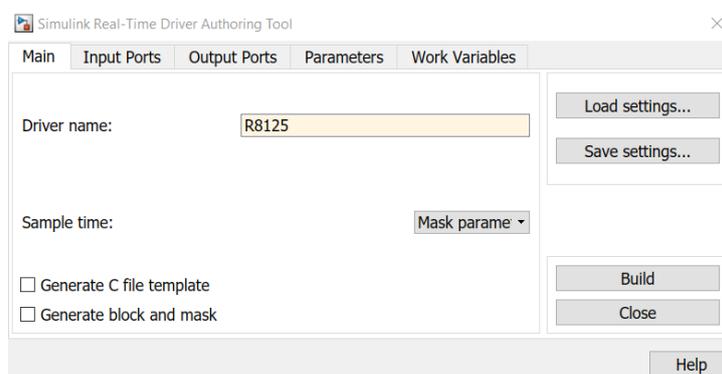


Figure 5.2: Startup of `xpcdrivertool`.

By clicking on the Input Ports tab (Figure 5.3), we can add the input ports needed for the block. The name used for the port becomes the variable name in the C program used to read the port. Scrolling the Type pulldown, the port and variable type can be defined. The three checkboxes Start, Output and Terminate tell the code generator in which part of function variables are needed. Since input and output ports can only be accessed within output subroutine the tool pre-selects the proper checkboxes, while for parameters and work variables the user can select in which subroutine the variable is recalled. Figure 5.3, 5.4 and 5.5 show Input Ports, Output Ports and Work Variables tabs for R8125 ethernet driver as example. Parameters tab in this case is empty because the Simulink block is not built, and no parameters need to be edited after the driver is compiled. Instead, C code of the driver provided by the board manufacturer is analyzed to define variables that have to be shared between subroutines and need to be declared in the Work Variables tab.

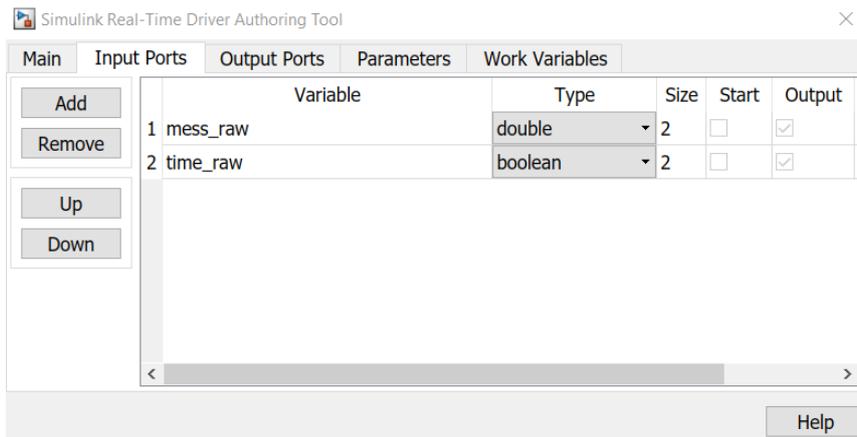


Figure 5.3: Input Ports tab for R8125 ethernet driver.

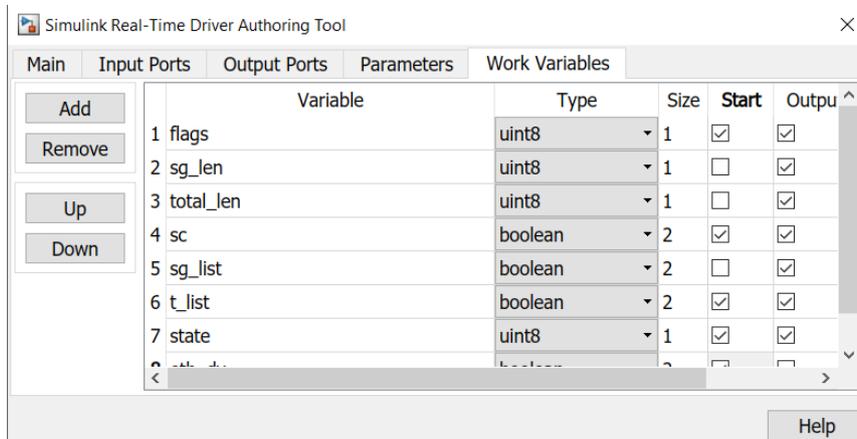


Figure 5.4: Work Variables tab for R8125 ethernet driver.

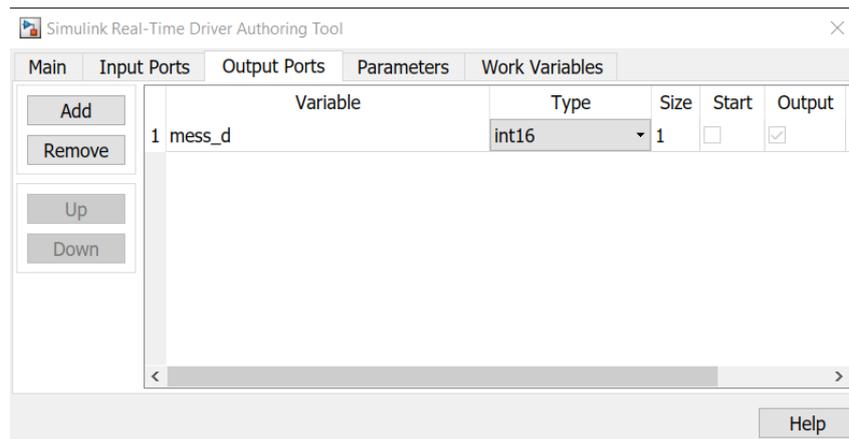


Figure 5.5: Output Ports tab for R8125 ethernet driver.

Returning to the Main tab some fields have to be filled in. The sample time with which the driver block is executed can be inherited or supplied by the block mask but in this case block and mask are not generated and the checkbox is empty. It can be chosen if the C file template and the block and mask are generated. As mentioned, for this specific workflow it is generated the C file template but held off on generating the block and mask. Hence, tool generates a .c stub file and a .h header file by clicking Build button. Figure 5.6 displays the Main tab set in this way.

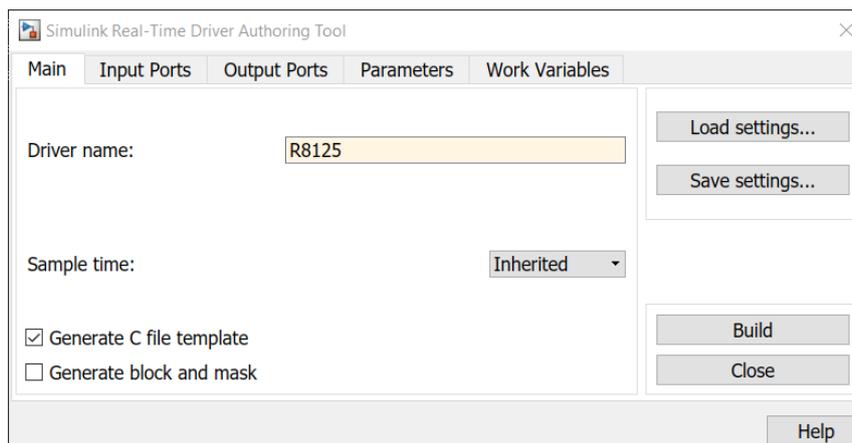


Figure 5.6: Main tab for the C file template generation.

Figure 5.7 shows the initialization of the generated the C file template as example. Code included the header file, R8125.h. There are even prototypes for R8125Start, R8125Output and R8125Terminate. Bodies of functions themselves are templates. The actual driver code that accesses the hardware needs to be written and it is taken from the native driver of the ethernet board. The arguments contain pointers to all the ports and work variables specified within the GUI, using the names indicated. Code that should only run on the target (such as the RT xPC kernel functions) must be protected by an `#ifndef MATLAB_MEX_FILE / #endif` construct. This is true for code that is inside the Start, Output and Terminate functions as well as code that resides outside those functions. Consequently, such part of the code must not be canceled from the C file. The body of each function consists of a comment inside a `#ifndef` that have to be replaced with driver code.

```

sfcn_R8125.c
37  * Need to include simstruc.h for the definition of the SimStruct and
38  * its associated macro definitions.
39  */
40  #include "simstruc.h"
41
42  /* Specific header file(s) required by the legacy code function */
43  #include "R8125.h"
44
45  #define EDIT_OK(S, P_IDX) \
46      (!((ssGetSimMode(S)==SS_SIMMODE_SIZES_CALL_ONLY) && mxIsEmpty(ssGetSFcnParam(S, P_IDX))))
47
48  #define SAMPLE_TIME (ssGetSFcnParam(S, 0))
49
50
51  /* Utility function prototypes */
52  static boolean_T IsRealMatrix(const mxArray *m);
53
54  #define MDL_CHECK_PARAMETERS
55  #if defined(MDL_CHECK_PARAMETERS) && defined(MATLAB_MEX_FILE)
56  /* Function: mdlCheckParameters =====
57  * Abstract:
58  * mdlCheckParameters verifies new parameter settings whenever parameter
59  * change or are re-evaluated during a simulation. When a simulation is
60  * running, changes to S-function parameters can occur at any time during
61  * the simulation loop.
62  */
63  static void mdlCheckParameters(SimStruct *S)
64  {
65      /* Check the parameter 1 (sample time) */
66      if (EDIT_OK(S, 0)) {

```

Figure 5.7: C file template for R8125 driver.

When C code is completed, it can be encrypted through the command `pcode` followed by the name of the file. Such procedure is needed to save the file with an extension that makes it compatible with the code that reads the available ethernet driver, user specifications (such as the IP number associated to the target machine, checkboxes that enable USB support, multi-core execution of RT applications and so on) and create the boot disk image. Figure 5.8 display the SRT Explorer window and it demonstrates that the driver code is properly read by the system, because the driver for Realtek 8125 ethernet controller is available.

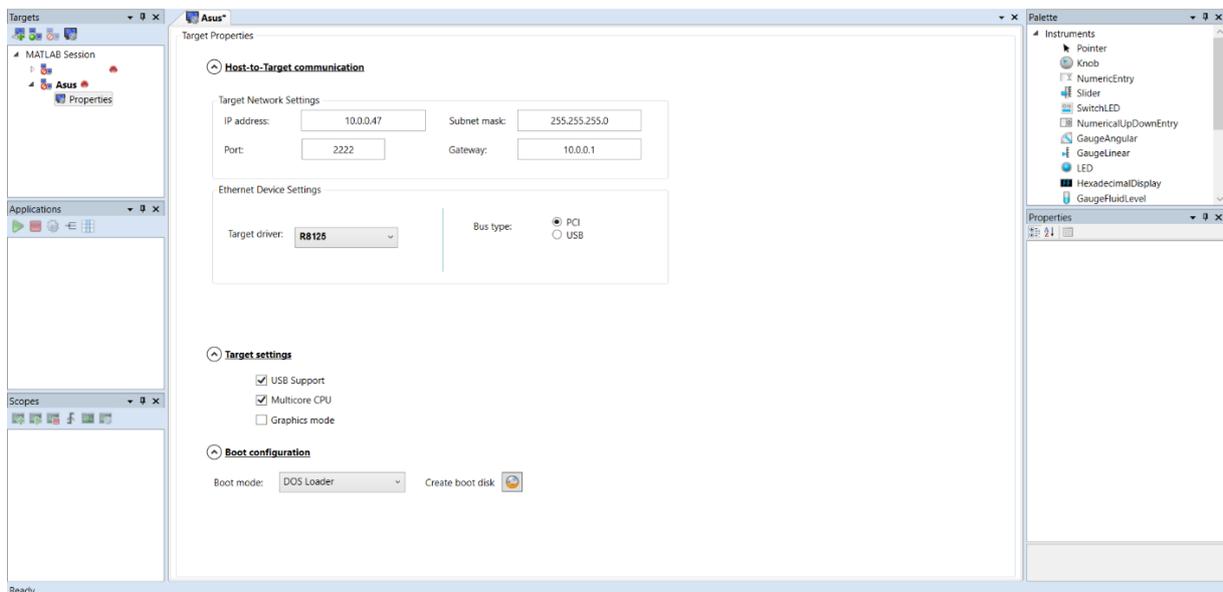


Figure 5.8: SRT Explorer with the R8125 ethernet driver available.

## 5.3. Real-Time Software

The RT xPC software environment includes many features to help prototyping, testing and deploying real-time application. Elements of a RT system are the following:

- **Real-Time Kernel:** BIOS, BIOS-extension, kernel, and loader
- **Real-Time Application:** memory model, and task execution time
- **Signal Acquisition:** signal monitoring, signal logging to the target or host PC memory, and signal tracing on the host PC screen
- **Parameter Tuning:** interactive, scripts and batch procedures

### 5.3.1. *Real-Time Kernel*

RT OS can be launched as an independent boot disk image installed on a removable disk or via network, from the host PC. For these options xPC target does not require DOS, Windows, Linux, or any another operating system on the target PC. Instead, you boot the target PC with a boot disk that includes the highly optimized xPC kernel. Nevertheless, the xPC Embedded Option (DOS Loader and Stand-Alone Mode) requires DOS and a DOS license at boot time. Indeed, if the RT kernel is generated as MS DOS application (.com or .exe) a Microsoft DOS or Windows are needed just to start the kernel boot sequence, that suppresses the primary system. This option has been proven more reliable when tested on different PCs and it has been chosen for custom RT machine building.

#### 5.3.1.1. **Target Boot Disk**

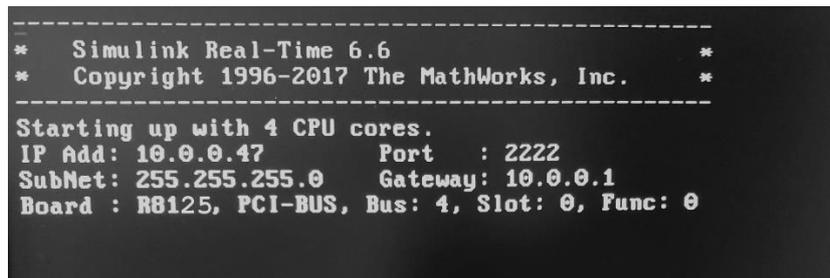
The boot disk eliminates the need to install software, modify existing software configurations, or access the hard disk on the target PC. This arrangement allows user to use the target PC for testing real-time applications, and then when tests are finished, he can use the target PC again as a desktop computer. Software is not permanently installed on the target PC, unless user is deliberately using the xPC Embedded Option and install a stand-alone application on the hard disk or flash memory. This option has been proven more reliable when tested on different PCs and it has been chosen for custom RT machine building, due to the system installation on a no longer used HW as desktop PC. Target boot disk is thus prepared with the free version of MS DOS and the RT kernel is installed as DOS loader.

#### 5.3.1.2. **Target PC BIOS**

Selecting a newer BIOS allows the customization of settings for a better control over the real-time behavior of the system. For example, it can be suppressed checking for a keyboard, and switched off any power save features. The RT xPC kernel runs only on a PC compatible system and a key component of every PC compatible system is the BIOS. The BIOS is the only software component which is needed by the xPC kernel. After the BIOS is loaded, it searches the target boot disk for a bootable image (executable). This bootable image includes a 16-bit part and a 32-bit part. The 16-bit part runs first because the CPU is still in real mode. It prepares the descriptor tables and in addition to other things switches the CPU to protected mode. Next, the 32 bit part runs. It prepares the target PC environment for running the kernel and finally starts the kernel. After loading the kernel, the target PC does not make calls to the BIOS or DOS functions. The resources (for example, interrupt controller, UART, and counters) on the CPU motherboard are addressed entirely through I/O addresses.

### 5.3.1.3. Real-Time Kernel Features

After the kernel starts running, it displays a welcome message with information about the host-target connection. Figure 5.9 shows the displayed screen when the kernel is properly started, and system is ready for application execution. The kernel activates the application loader and waits to download a target application from the host PC. The loader receives the code, copies the different code sections to their designated addresses, and sets the target application ready to start. You can now use xPC Target functions and other utilities to communicate with the target application. It is important to mention, that after the CPU switches to protected mode (32-bit), none of the xPC Target components switch the CPU back to real mode (16-bit). The generated real-time application and the real-time kernel are compiled as Windows NT applications with a flat memory model. This provides full 32-bit power without time consuming 16-bit segment switching and DOS extenders.



```
-----  
*   Simulink Real-Time 6.6   *  
*   Copyright 1996-2017 The MathWorks, Inc.   *  
-----  
Starting up with 4 CPU cores.  
IP Add: 10.0.0.47      Port   : 2222  
SubNet: 255.255.255.0  Gateway: 10.0.0.1  
Board  : RB125, PCI-BUS, Bus: 4, Slot: 0, Func: 0
```

Figure 5.9: Screen of launched RT kernel.

### 5.3.1.4. Target PC Heap

The initialization code of the target application reserves the remaining unused RAM as heap. The memory available for the heap is 4 MB less than the entire RAM installed by default (1 MB for the application; 3 MB for the kernel). Normally, the largest part of the heap is used by signal logging because logging acquires and stores data during the entire run. However, user can define the amount of memory available for data logging in the Simulation Parameters dialog box.

## 5.3.2. Real-Time Application

Real-Time Workshop, Stateflow Coder, RT xPC, and a C compiler create a real-time application (target application) from a Simulink and Stateflow model. Target applications created with Real-Time Workshop and xPC run in real time on a standard PC without using a Windows operating system. The target application runs in real time on the target PC and has the following characteristics:

- **Memory model:** the target application is compiled as a Windows NT application with the flat memory model. This executable is then converted to an image suitable for xPC Target, and it provides full 32-bit power without time consuming 16-bit segment switching and DOS extenders. Also, it does not rely on DOS or any other Microsoft operating system.
- **Task execution time (TET):** the target application is capable of high-speed, real-time task execution. A small block diagram can run with a sample time as fast as 10  $\mu$ s (100 kHz). Model size, complexity, and target PC hardware affect maximum speed (minimal sample time) of execution.

### 5.3.3. *Signal Acquisition*

The xPC Target real-time kernel stores signal data from the target application in RAM on the target PC, on its hard drive or on the host PC disk. You can use this signal data to analyze and visualize signals. xPC Target supports the following types of signal acquisition:

- **Signal monitoring:** this is the process for acquiring signal data without time information. In this mode, you can get the current value of one or more signals. The data is not acquired in the real-time task but in the background task. The advantage of this process is that collecting data does not add any computational load to running the real-time application. For example, if you have a LED Gauge in a Simulink model on the host PC, you could use signal monitoring to display the status of the signal.
- **Signal logging:** this is the process of acquiring signal data while a target application is running, and then visualizing the collected data after the target application stops running. The data is collected in the real-time task and acquired samples are associated with a time stamp. After the run reaches its final time or you manually stop the run, the host PC makes a request to upload data from the target PC. You can then visualize signals by plotting data on the host PC, or you can save data to a disk.
- **Signal tracing:** this is the process of acquiring and visualizing signal data while a target application is running. The data is collected in the real-time task and acquired samples are associated with a time stamp. It allows you to acquire signal data and visualize it on the target PC or to upload the signal data and visualize it on the host PC while the target application is running. The flexibility of this acquisition type is very similar to the behavior of a digital oscilloscope.

### 5.3.4. *Parameter Tuning*

Most Simulink blocks have parameters that user can change before or while your target application is running. Parameters can be varied with the following modes:

- **Interactive:** xPC Target supports interactive tuning of parameters while the target application is running in real time. The changes to parameters are immediately reflected in the signal outputs.
- **Scripts and batch procedures:** xPC Target also includes commands to change parameters during a run or between runs. By writing and running a script on the host PC that incrementally changes a parameter and monitors a signal output, you can optimize the value of that parameter.

Signal monitoring and tracing and parameter tuning can be performed by developing custom graphical instrumented interface with the SRT Explorer panel. Signals and Parameters can be quickly associated to each instrument with the drag-and-drop gesture. As mentioned above, Explorer panel allows to browse in the target root directory, to download/upload files from/to target PC and visualize specifications and loaded applications of connected HW. Figure 5.10 and 5.11 show an example of instrumented panel for an automotive application and browsing capabilities of such system.



Figure 5.10: Example of instrumented panel developed with SRT Explorer.

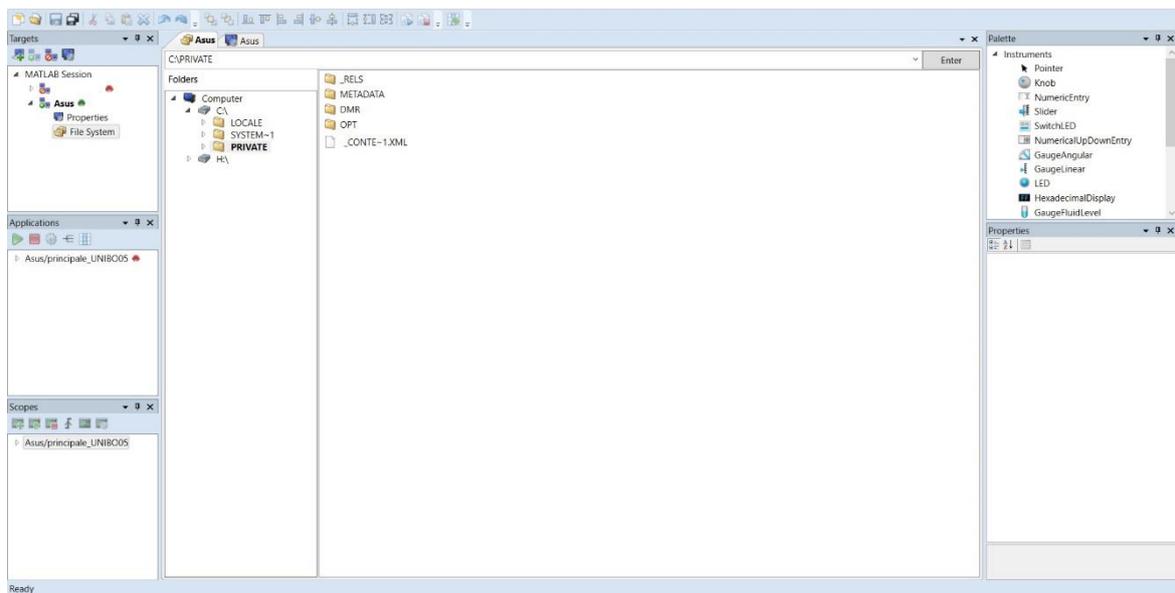


Figure 5.11: Visualizing root directories and loaded application on target PC with SRT Explorer.

## 5.4. Real-Time Hardware

The hardware environment consists of a host computer, target computer, I/O boards in the target computer, and a serial or network connection between the host and target computers. Knowing the different types of computers and I/O supported by xPC Target will help you to setup a development environment that meets your needs. This section includes the following topics:

- **Host PC:** desktop PC, or notebook PC
- **Target PC:** desktop PC, industrial PC, PC 104, or CompactPCI
- **Host-Target Connection:** RS232 serial or TCP/IP network
- **I/O Driver Support:** analog, digital, CAN, GPIB, RS232, UDP, counters, timers, and signal conditioning

### 5.4.1. *Host and Target PCs*

User can use any PC that runs a Microsoft Windows platform supported by The MathWorks as the host PC. Also, it must contain a free serial port or an Ethernet adapter card.

Considering the developed ethernet driver described above, user can use any PC with any Intel or AMD processor as the target computer. Also, it must contain a free mPCIe port on the PCB, for the external R8125 ethernet board installation. The target PC can be one of the following:

- **Desktop PC:** this computer is booted from a special target boot disk created by xPC Target. When user boots the target PC from the target boot disk, xPC Target uses the resources on the target PC (CPU, RAM, and serial port or network adapter) without changing the files already stored on the hard drive. After user has ended up using desktop computer as a target PC, he can reboot your computer without the target boot disk and resume normal use of your desktop computer.
- **Industrial PC:** this computer is booted from a special target boot disk, or with the xPC Target Embedded Option, booted from a hard disk or flash memory. When using an industrial target PC, you can select PC104, PC104+, CompactPCI, or single-board computer (SBC) hardware.

No special target hardware is thus needed. However, the target PC must be a fully PC compatible system and contain a serial port or an Ethernet controller compatible with xPC Target.

### 5.4.2. *Host-Target Connection*

xPC Target supports two connection-and-communication protocols between the host PC and the target PC: serial and network.

- **Serial:** the host and target computers are connected directly together with a serial cable using their RS232 ports. This cable is wired as a null modem link that can be up to 5 meters long and with a transfer rate between 1200 and 115200 Baud. We provide a null modem cable with the xPC Target software
- **Network:** the host and target computers are connected through a network. The network can be a LAN, the Internet, or a direct connection using a cross-over Ethernet cable. Both the host and target computers are connected to the network with Ethernet adapter cards

using the TCP/IP protocol for communication. When using a network connection, the target PC can use the Ethernet adapter card provided with xPC or one of the supported cards. The data transfer rate can be 10 megabits/second or 100 megabits/second.

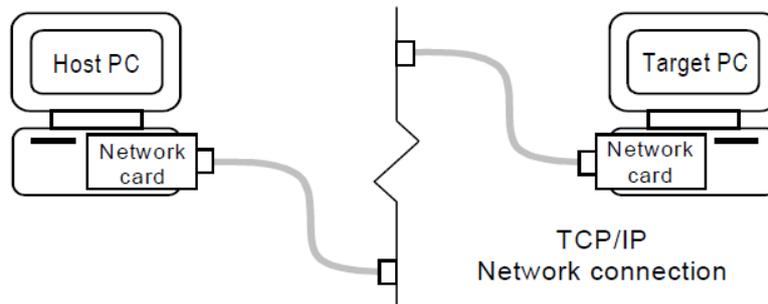


Figure 5.12: TCP/IP network connection scheme between host and target PC.

### 5.4.3. I/O Driver Support

xPC Target supports a wide (an extensible) range of I/O boards. The list of supported I/O boards includes ISA, PCI, PC/104, PC/104+, and CompactPCI hardware. The drivers are represented by Simulink blocks and SRT driver block library can be extended with the authoring tool, as described above. User interaction with the drivers is through Simulink blocks and the parameter dialog boxes.

- **I/O board library:** the I/O driver library contains Simulink blocks for xPC Target. User can drag-and-drop blocks from the I/O library and connect I/O drivers to the model the same way as he would connect any standard Simulink block
- **I/O support:** the I/O device library supports over 150 standard boards. I/O boards plug into the target PC expansion bus, PC104 stack, or industrial PC chassis. Also, there is support for modules that plug into IP or PMC carrier boards. xPC Target supports the following I/O functions:
  - **Analog input (A/D) and analog output (D/A):** interface sensors and actuators to your target application.
  - **Digital input and output:** Interface to switches, on/off devices, and communicate information in parallel
  - **Audio and Video:** driver blocks for audio and video are available in the library. Audio signals can be logged or generated, while images of an external camera connected with the target can be visualized, saved, analyzed and shared in real time with host PC
  - **Thermocouples:** thermocouple signal can be recorded by installing Measurement Computing PCI-DAS-TC board, with 16 input channels
  - **RS232 support:** Use the COM1 or COM2 ports for serial communication with external devices
  - **CAN support:** it can be used CAN-AC2, CAN-AC2-PCI, and CAN-AC2-104 boards from Softing GmbH AG with xPC Target CAN drivers to interface with a CAN fieldbus network. This interface provides communication through a CAN network between target applications and remote sensors and actuators. The xPC Target CAN drivers are compatible with CAN specification 2.0A and 2.0B and use the dynamic object mode

- **GPIB support:** special RS232 drivers support communication with a GPIB control module from National Instruments to external devices with a GPIB connector
- **UDP support:** communicate with another system using the standard UDP/IP network protocol
- **Counter-Timers:** use the counter-timer blocks for measuring pulse and frequency with modulation applications
- **Watchdog:** monitor an interrupt or memory location and reset the computer if an application does not respond
- **Incremental encoder:** change motion into numerical information for determining position, direction of rotation, and velocity
- **Shared memory:** use shared memory blocks with multiprocessing applications

## 5.5. Rapid Control Prototyping Process

In this section steps to design and build an algorithm on a host PC, and then run and test the application on a target PC are described. xPC Target functions include interactive control of the target application, acquisition of signal data, and tuning of parameters while running in real time.

The rapid prototyping process includes the following sequence of tasks:

1. **Create a Simulink and Stateflow model:** user can create block diagrams in Simulink using standard drag-and-drop operations, and then enter values for the block parameters and select sample rates
2. **Simulate the model in nonreal time:** Simulink uses a computed time vector to step the model. After the outputs are computed for a given time value, Simulink immediately repeats the computations for the next time value. This process is repeated until it reaches the stop time. Because this computed time vector is not connected to a hardware clock, the outputs are calculated in nonreal time as fast as your computer can run. The time to run a simulation can differ significantly from real time.
3. **Create an executable target application:** Real-Time Workshop, Stateflow Coder, xPC Target, and a C compiler create the target application that runs on the target PC. This real-time application uses the initial parameters from the Simulink model that were available at the time of code generation. If a Multi-Core kernel is installed on target machine (with a multi-core CPU) Simulink model can be configured assigning specific tasks to each CPU core. Model must be built as a group of connected Model Reference blocks that are deployed on target as independent tasks that send and receive just data that must be shared. Figure 5.13 shows the Concurrent Execution panel where Model Reference blocks are assigned to different tasks. In this specific example first task (and then first available CPU core) is dedicated to CAN communication management while second to control algorithm

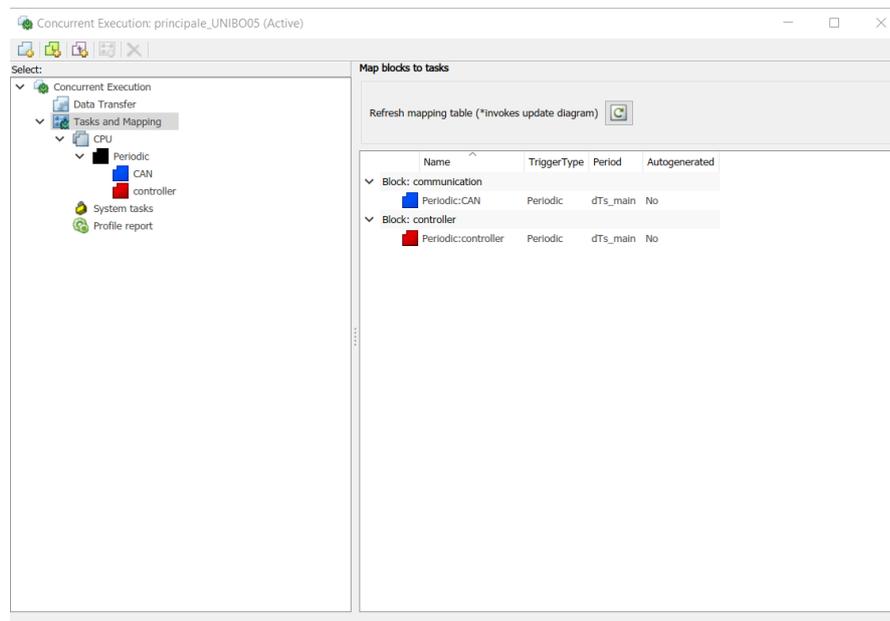


Figure 5.13: Concurrent Execution panel.

4. **Execute the target application in real time:** target PC is booted using an xPC Target boot disk that loads the xPC Target real-time kernel. After booting the target PC, you can build

and download a real-time application. xPC Target provides the necessary software that makes use of real-time resources on the target PC hardware. Based on your selected sample rate, xPC Target uses interrupts to step the model at the proper rate. With each new interrupt, the target application computes all of the block outputs from your model.

5. **Acquire signals:** acquire signal data using xPC Target Scopes. User can create xPC Target Scopes and acquire data from the target application by:
  - xPC Target: using the xPC Target Remote Control Tool and Scope Manager window to create scopes, and using the Simulink viewer to add signals.
  - MATLAB: entering commands in the MATLAB window
  - Simulink: adding xPC Target Scope blocks to your Simulink model. Note xPC Target does not support normal Simulink scope blocks in external mode. Instead, use xPC Target scope blocks
  - Target PC: using commands in the target PC command window
  - Web browser: using the xPC Target Web browser interface
  - Simulink GUI: adding Gauge blocks from the Dials & Gauges blockset to a Simulink user interface model with xPC Target From blocks.
6. **Tune parameters:** user can tune parameters with same tools available for the signal acquisition

## 5.6. Embedded Application Development

Often, control systems and digital signal processing applications are developed for use in production where a limited number of deployed systems are required. Whether deploying one or one hundred systems, the embedded option provides a convenient approach that allows to implement resulting system on low-cost custom target hardware and execute that automatically when the device is turned on. Indeed, when user has completed development and testing, he can use the target application as a real-time system that runs on a dedicated target PC without the need to connect to the host computer.

The xPC Target Embedded Option consists of a specific mode of operation. Target PC boots into DOS, starts the DOS program xpcboot.com from autoexec.bat, and then starts the kernel from xpcboot.com. Moreover, if Simulink model is compiled selecting the Stand-Alone option on the associated target, boot sequence runs both the kernel and the target application from the hard disk on the target PC. The target application runs completely independent from the host PC. In other words, the Stand-Alone mode combines the target application with the kernel and boots them together on the target PC from the hard disk drive, or flash memory. The host PC does not have to be connected to the target PC. Following points summarize main steps for the stand-alone system building:

1. Select the Stand-Alone mode from the xPC Target Setup window on SRT Explorer (Figure 5.14)

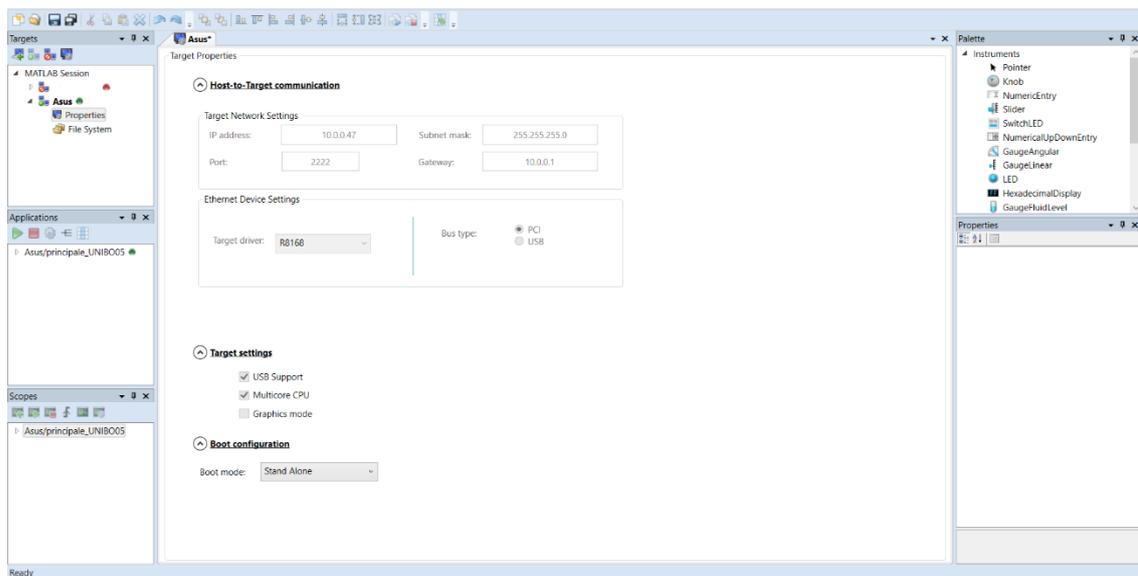


Figure 5.14: Target boot mode set as Stand-Alone.

2. Select the same target machine in the Simulink Real-Time Options tab of the model configuration parameters (Figure 5.15)

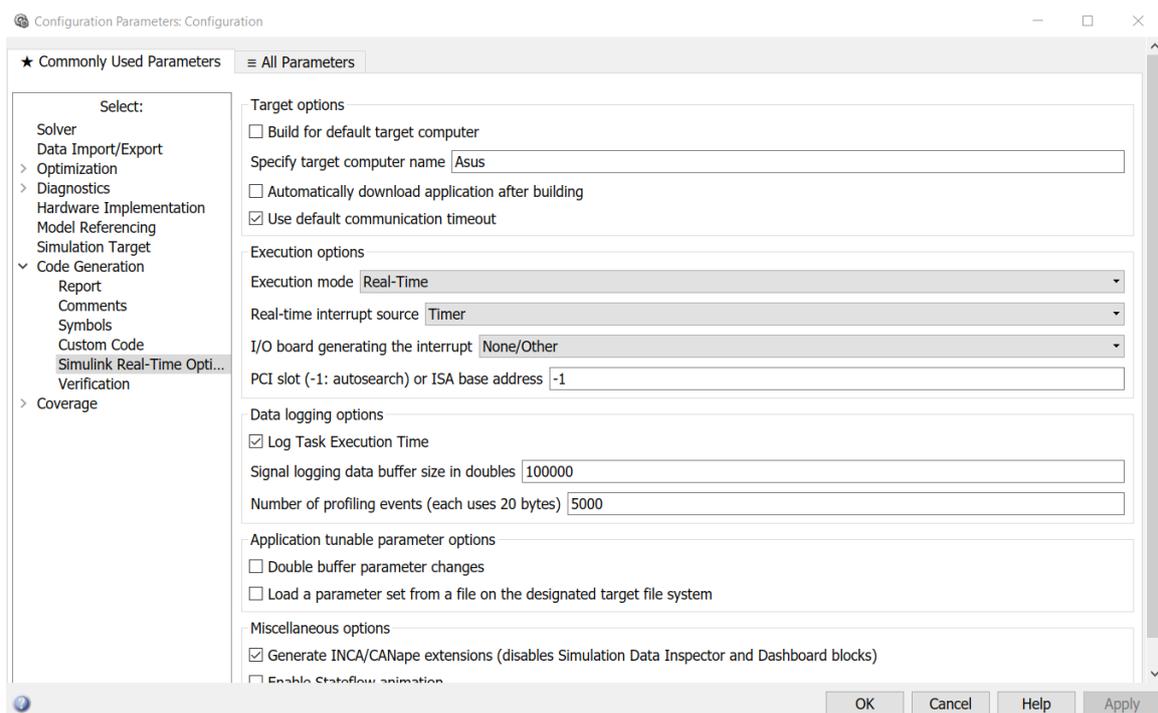


Figure 5.15: The target PC is specified in model Configuration Parameters.

3. Build a kernel/target application
4. Copy DOS system files, utilities, kernel/application files and helper files to the boot disk. If booting from an alternate boot device, in the autoexec.bat file, remove the line that loads xpcboot.com.
5. Boot the target PC with a bootable DOS disk from the floppy disk. If booting from an alternate boot device, copy files to the alternate boot device. In the autoexec.bat file, add the line that loads xpcboot.com. Target automatically runs both RT kernel and application when turned on (Figure 5.16).

```

-----
*   Simulink Real-Time 6.6                               *
*   Copyright 1996-2017 The MathWorks, Inc.             *
-----
Starting up with 4 CPU cores.
IP Add: 10.0.0.47      Port   : 2222
SubNet: 255.255.255.0  Gateway: 10.0.0.1
Board : R8125, PCI-BUS, Bus: 4, Slot: 0, Func: 0
Unpacking application...
Unpacking application finished
Loading application C:\principale_UNIB005
Application loaded - principale_UNIB005
real-time mode, multi-tasking
Stop time is Inf
Static data logging: time tet
Initializing application...
      downloading CAN-AC2-PCI firmware on board 1
      downloading CAN-AC2-PCI firmware finished
Initializing application finished.
Model thread stack is 2048K bytes
Execution started (sample time: 0.003000)

```

Figure 5.16: Target screen output for Stand-Alone mode.

## 5.7. User Remote Interaction Options

The xPC Target environment has a modifiable interface to the target PC. User can use this interface from MATLAB or Simulink, and he can use other development environments to create stand-alone client applications independent of MATLAB. Because of this open environment, there are several ways to interact with the target application from the host and target PCs.

This section deepens the following methods for the remote interaction with RT application:

- **xPC Target Remote Control Tool:** use the xPC Target Remote Control Tool to set xPC Target environment properties, control the target application, add xPC Target Scopes, and change tunable parameters
- **MATLAB Command-Line Interface:** enter xPC Target functions on the host PC
- **Simulink External Mode Interface:** connect a Simulink block diagram to the target application using Simulink External Mode, and then use the block diagram for parameter tuning
- **Simulink with xPC Target Scope Blocks:** add xPC Target Scope blocks to your model for signal tracing
- **Target PC Command-Line Interface:** enter xPC Target functions on the target PC
- **Web Browser Interface:** use Microsoft Internet Explorer to connect to the target PC from any computer on the network with the target PC
- **Custom GUI with Simulink, Dials, and Gauges:** create a model with the Dials & Gauges Blockset, and xPC Target To and From blocks. This user interface (UI) model runs in Simulink on the host PC
- **Custom GUI with xPC Target API:** create a client application that interfaces with a target application using any development environment that can call functions from a DLL
- **XCP communication protocol:** create the .a2l file associated with RT application to interface target computer to Vector CAN Application Environment (CANape) or ETAS INCA, using the XCP communication.

### 5.7.1. *xPC Target Remote Control Tool*

xPC Target offers a graphical user interface (GUI) for interacting with a target application. This GUI is built using xPC Target functions and MATLAB Handle Graphics®. To open the xPC Target GUI, in the MATLAB Command Window, type `xpcrcctool`.

The xPC Target Remote Control Tool includes the following functions:

- **Environment:** use the xPC Target Setup window to change properties in the xPC Target environment. Properties include communication between the host and target computers, type and location of your C compiler, and target PC properties
- **Control:** build, download, and run a target application. Change start and sample times without regenerating code and get statistical performance information during or after the last run.
- **Signal acquisition:** use the xPC Target Simulink Viewer to interactively add scopes of type host or target and add or remove signals. Use the xPC Target Scope Manager to set the triggering mode of a scope. A scope created on the target PC acquires data from the target application and stores the data for display on the host PC or target PC.

- For scopes of type host: Host Scope Manager periodically uploads data packets and displays them in a MATLAB figure.
- For scopes of type target: scope displays signal traces on the target PC screen. Because xPC Target uses highly optimized graphic routines, signal tracing has the same fast display and update rates that you normally observe when using a digital oscilloscope
- **Parameter tuning:** use the xPC Target Simulink Viewer to interactively change tunable parameters in your target application.

### 5.7.2. *MATLAB Command-Line Interface*

User can interact with the xPC Target environment through the MATLAB command-line interface. Enter xPC Target functions in the MATLAB Command Window on the host PC. Also, it can be written a custom M-file scripts that use xPC Target functions for batch processing. xPC Target has more than 90 MATLAB functions for controlling the target application from the host computer. These functions define, at the most basic level, what you can do with the xPC Target environment. The GUIs provided with xPC Target are for completing the most common tasks. They use the xPC Target functions but do not extend their functionality. The command-line interface provides an interactive environment that you can extend.

The MATLAB command-line interface includes the following functions:

- **Environment:** create a boot disk and directly change the environment properties without using a graphical interface
- **Control:** reboot the target PC, download a target application, start and stop target applications, and change start and sample times without regenerating code. Get statistical performance information during or after the last run. Add/remove scopes, add/remove signals to scopes, and define triggers for scope display
- **Signal acquisition:** trace signals for viewing while the target application is running and monitor signal values without time information. Transfer logged signal data to the MATLAB workspace by uploading from the host PC to the target PC between runs
- **Parameter tuning:** change parameters while the target application is running, and use xPC Target functions to change parameters in between runs

### 5.7.3. *Simulink External Mode*

Use Simulink in external mode to connect your Simulink block diagram to your target application. The block diagram becomes a graphical user interface to the target application running in real time. By changing parameters in the Simulink blocks, you also change parameters in the target application.

The Simulink external mode interface includes the following functions:

- **Control:** control is limited to connecting the Simulink block diagram to the target application, and starting and stopping the target application
- **Parameter tuning:** elect external mode and change parameters in the target application by changing parameters in the Block Parameters dialog boxes. Once user changes a value, the

new value is immediately downloaded to the target PC and replaces the existing parameter while the target application continues to run.

#### 5.7.4. *Simulink with xPC Target Scope Blocks*

An alternative to interactively adding scopes to the target PC is to add xPC Target Scope blocks to the Simulink model. After the download process, these blocks create scopes on the target PC during initialization of the target application. User can choose to display data on either the host PC or target PC.

- **Signal acquisition:** add scopes to the target PC by adding xPC Target Scope blocks to Simulink model. In the Block Parameters dialog box, select the scope mode and set the trigger.

#### 5.7.5. *Target PC Command-Line Interface*

User can interact with the xPC Target environment through the target PC command window. Enter commands in the command-line on the target PC. This interface is useful with stand-alone applications that are not connected to the host PC.

The target PC command-line interface includes the following functions:

- **Control:** start and stop the target application and change the stop time and sample time
- **Signal acquisition:** acquiring signal data is limited to viewing signal traces and signal monitoring on the target PC screen
- **Parameter tuning:** changing parameters is limited to changing the scalar parameters in your model.

#### 5.7.6. *Web Browser Interface*

If the target PC is connected to a network (TCP/IP), you can use a Web browser to interact with the target application from any computer connected to the network. Also, if the target PC is connected to the host PC with an RS-232 cable and is using the TCP/IP to RS-232 gateway, user can use a Web browser on the host PC.

The Web browser interface includes the following functions:

- **Control:** start and stop the target application, and change the stop time and sample time
- **Signal acquisition:** signal tracing is limited to viewing a snapshot of a screen captured from the target PC screen. Add scopes of type target, add or remove signals, and set triggering modes. Also, signal values can be monitored
- **Parameter tuning:** change parameters in an HTML form, and then submit that form to make the changes in your target application.

### 5.7.7. *Custom GUI*

Dials, gauges, and xPC Target interface blocks can be used to create a user interface (UI) model in Simulink. Custom GUIs can be created even as independent program that can link in a DLL or that can include COM objects.

#### 5.7.7.1. **Custom GUI with Simulink**

This GUI is directly created in Simulink environment. Host Simulink model runs in normal mode and in nonreal time. The xPC Target interface blocks connect this Simulink model to the target application, which runs in real time.

The Simulink Dials and Gauges Blockset interface includes the following functions:

- **Signal acquisition:** use Gauge blocks to visualize signal data
- **Parameter tuning:** use Dial blocks to change model parameters.

#### 5.7.7.2. **Custom GUI with xPC Target Application Programming Interface (API)**

Create a GUI application interface to a target application using any development environment which can link in a DLL. Use the GUI application to control, tune parameters, and acquire signal data from a target application. The Custom GUI runs on the host PC and communicates with the target application on the target PC using RS-232 or TCP/IP communication. A GUI application can be a console or Windows application using ActiveX components.

### 5.7.8. *XCP Communication Protocol*

User can use a target computer as an electronic control unit (ECU) for a Vector CANape system. Using a target computer in this way, a Vector CANape/INCA system can read signals and parameters from a real-time application running on the target computer via XCP. Such approach for remote control of RT application is often used in automotive field, especially at the test bench, due to the utilization of INCA system to interface ETK module installed on production ECU.

In this section the XCP protocol and procedure for the generation of INCA project related to the RT application are briefly introduced.

#### 5.7.8.1. **Universal Measurement and Calibration Protocol (XCP)**

XCP protocol was standardized by an ASAM working committee (Association for Standardisation of Automation and Measuring Systems), that is an organization of automotive OEMs, suppliers and tool producers. Such protocol succeeds CCP (CAN Calibration Protocol) and the conceptual idea of the CAN Calibration Protocol was to permit read and write access to internal ECU (or other RT systems) data over CAN. XCP was developed to implement this capability via different transmission media. Then one speaks of XCP on CAN, XCP on FlexRay or XCP on Ethernet. The primary applications of XCP are measurement and calibration of internal ECU parameters. Here, the protocol offers the ability to acquire measured values “event synchronous” to processes in ECUs. This ensures consistency of the data between one another. XCP was defined based on the ASAM interfaces model [4.4]. The Interface 1 is the so called “ASAM MCD-1 MC” between ECU and

measurement & calibration system. This interface describes the physical and the protocol-specific parts. Strictly speaking, a distinction was made between interfaces ASAP1a and ASAP1b here. The ASAP1b interface, however, never received general acceptance and for all practical purposes it has no relevance today. The XCP protocol is so flexible that it can practically assume the role of a general manufacturer-independent interface. For example, today all measurement and calibration hardware manufacturers offer systems (xETK, VX1000, etc.) which can be connected via the XCP on Ethernet standard. An ASAP1b interface – as it was still described for CCP – is no longer necessary. The Interface 2 is the “ASAM MCD-2 MC” A2L ECU description file. As already mentioned, XCP works in an address-oriented way. Read or write accesses to objects are always based on an address entry. Ultimately, however, this would mean that the user would have to search for his ECU objects in the Master based on the address. That would be extremely inconvenient. To let users work with symbolic object names, for example, a file is needed that describes the relationship between the object name and the object address. Such interface is fundamental to identify target PC within the local subnet and to make host PC able to properly read and write application signals and parameters, respectively. Such file is an output of the RT application compilation.

XCP is based on the Master-Slave principle. The ECU is the Slave, and the measurement and calibration tool is the Master. A Slave may only communicate with one Master at any given time; on the other hand, the Master can simultaneously communicate with many Slaves, as reported in Figure 5.17.

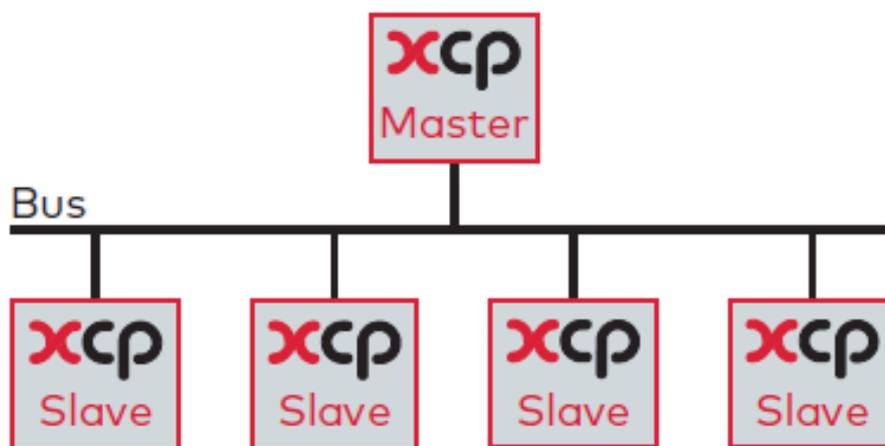


Figure 5.17: Master-Slave principle (from [4.4]).

Read and write accesses to memory contents are available with the mechanisms of the XCP protocol. The accesses are made in an address-oriented way. Read access enables measurement of parameters from RAM, and write access enables calibration of the parameters in RAM. XCP permits execution of the measurement synchronous to events in the ECU. This ensures that the measured values correlate with one another. With every restart of a measurement, the signals to be measured can be freely selected. For write access, the parameters to be calibrated must be stored in RAM.

### 5.7.8.2. Configuring the model for A2L file generation

To support the XCP communication layer, the SRT software provides:

- An XCP server process in the real-time application that runs on target HW

- A generator that produces A2L (ASAP2) files that INCA can load into the workspace database. The generated file contains signal and parameter access information for the real-time application

Following procedure is that tested and used for recent control algorithm developed during PhD course to add in INCA workspace the project associated with the stand-alone application that the xPC target machine runs. It assumes that user have already validated the control algorithm and that it has to be just calibrated.

Such steps allow to create the final INCA project in the active workspace, starting from the validated control algorithm developed in Simulink environment:

1. All Simulink model Signals and Parameters that have to be available to be monitored or modified through INCA Experiment must be properly defined in the Matlab Workspace:
  - **Simulink Signals:** on the specific wire of Simulink model signal name must be typed and the first checkbox (from the top) needs to be selected in the properties tab, to create the corresponding workspace object. Even the Test Point must be checked, and the Exported Global option must be selected for the storage class of the corresponding Simulink object. Figure 5.18 indicates how Simulink Signal properties must be configured.

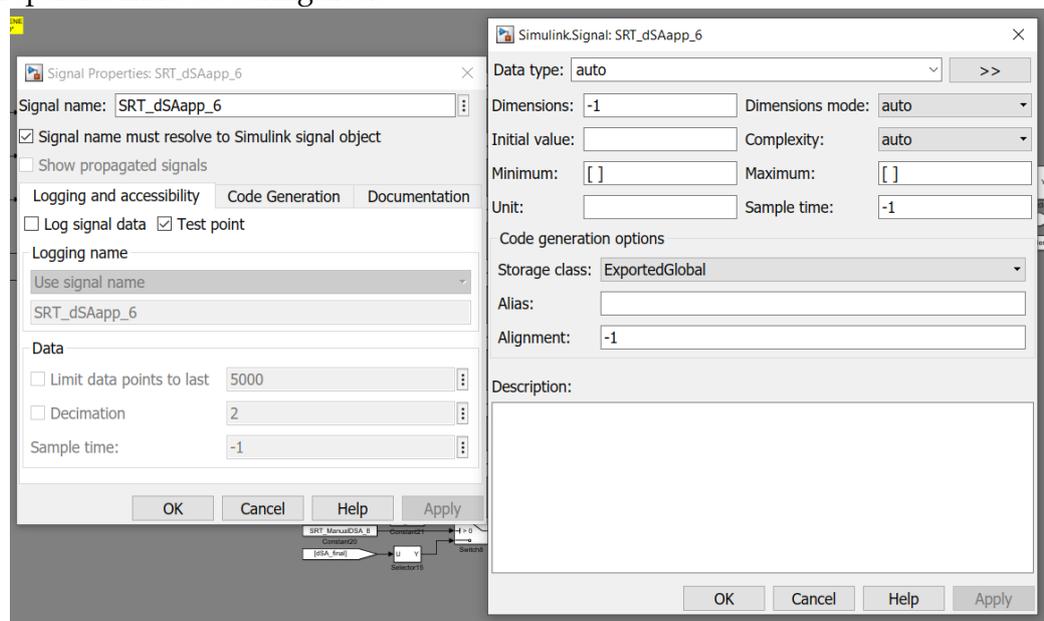


Figure 5.18: Simulink Signal configured for A2L file generation.

- **Simulink Parameter:** each parameter that needs to be tuned from INCA workspace must be defined as a Simulink Parameter object in the Matlab Workspace. Default value, data type and maximum and minimum value can be set in the related properties tab. As for Simulink Signal, the storage class must be set as Exported Global. Figure 5.19 shows how Simulink Parameter properties must be configured.

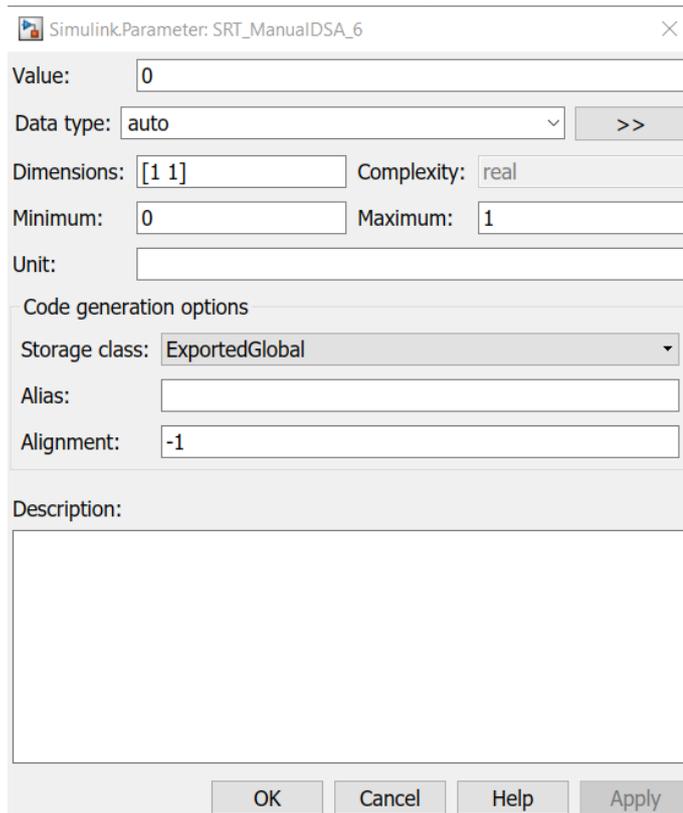


Figure 5.19: Simulink Parameter configured for A2L file generation.

2. Simulink model is compiled as a SRT application for the specific target machine and the checkbox for the INCA/CANape extension generation must be selected, as indicated in Figure 5.20.

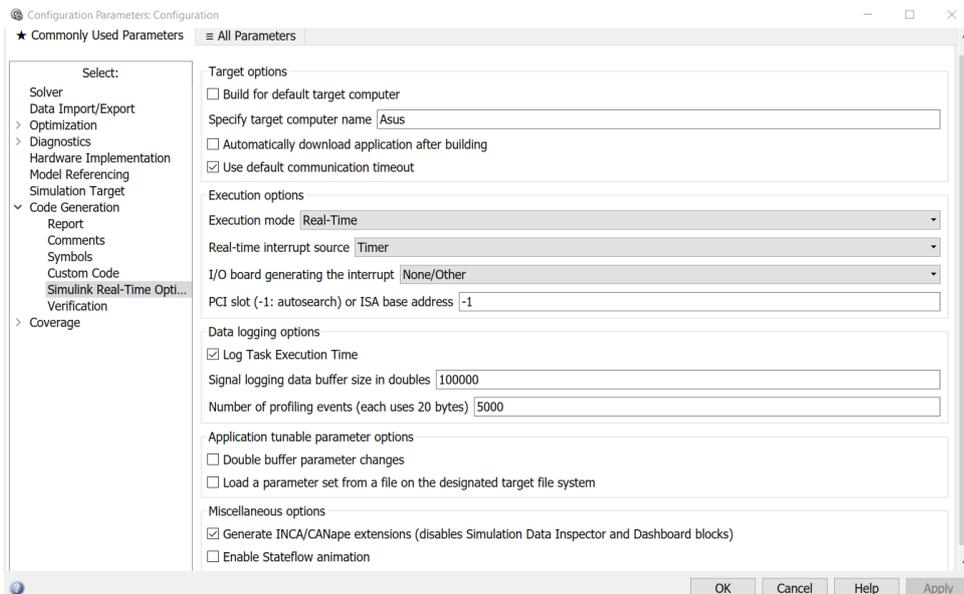


Figure 5.20: Generation of A2L file when RT application is built.

3. When model is correctly compiled the A2L file is generated in the directory that contains RT model

- A2L file is used to create a new project associated to the target device. INCA project is created by selecting the A2L file and neglecting HEX one. In this case INCA PC is the host hardware, and the connection uses Ethernet ports for the XCP communication. Figure 5.21 shows the project and HW added to the active workspace.



Figure 5.21: INCA workspace with SRT device added via XCP.

- Hardware can be finally initialized, and both Working Page and Reference Page are uploaded on the host PC from target device, due to the absence of pages description file. Such file can be created later, when a user defined calibration set is generated by modifying parameters from the INCA Experiment. Figure 5.22 indicates the correct option for the first alignment of host Memory Pages.

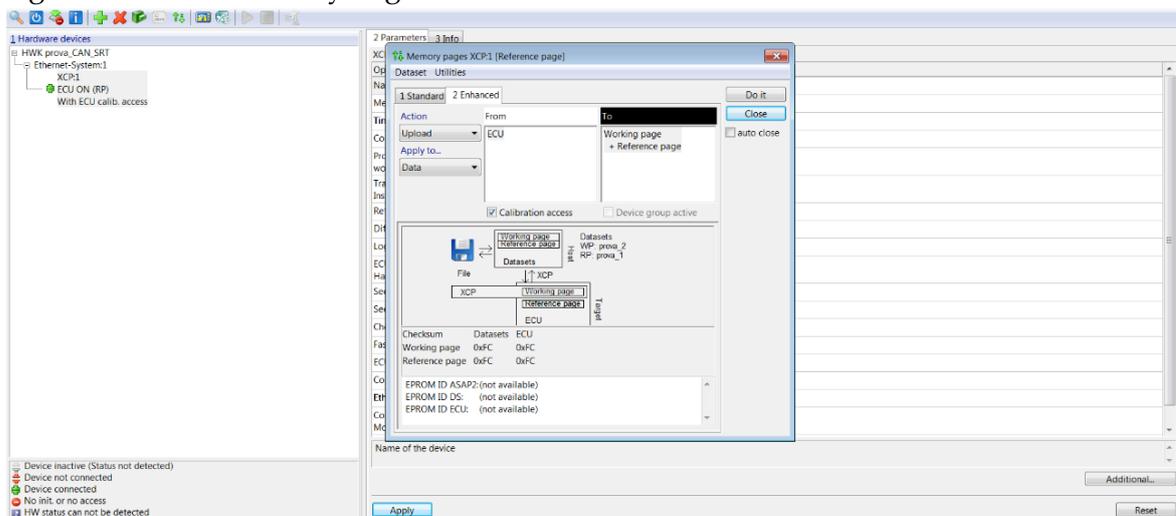


Figure 5.22: Memory pages for XCP connection.

6. Finally, target signals and parameters defined at step 1 can be added to the INCA Experiment (Figure 5.23).

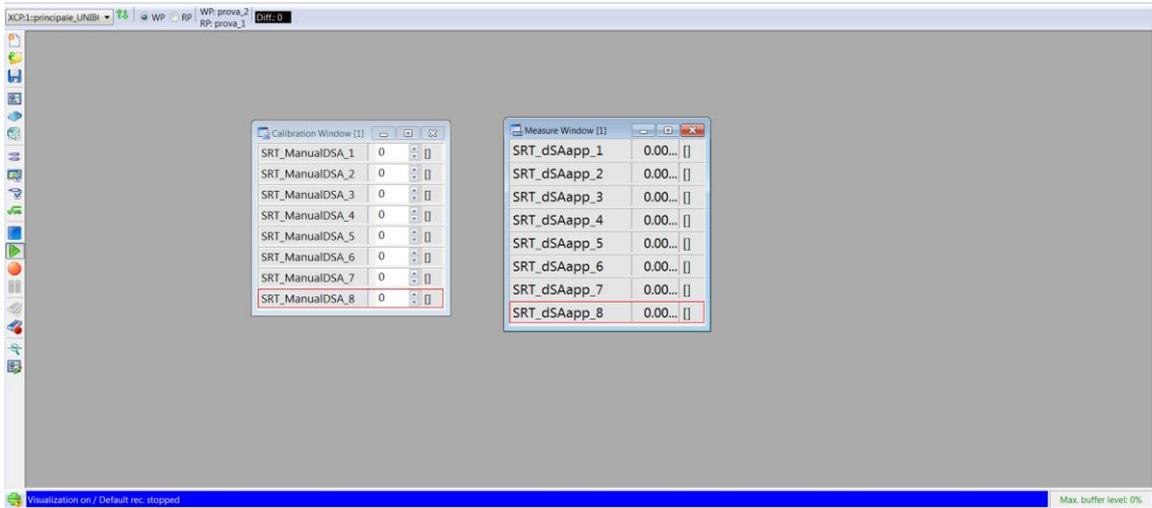


Figure 5.23: INCA Experiment with SRT device signals and parameters.

## 5.8. Custom Device Configuration

First final version of custom SRT target machine is realized with an Asus M3N78-VM motherboard with a quad-core AMD CPU. The PCB has 4 PCI slot available. In particular, there are 2 express PCI ports and 2 standard PCI ports (Figure 5.24).



*Figure 5.24: ASUS M3N78-VM motherboard.*

An USB pen is used as hard drive and two PCB supply connectors have been coupled to minimize the volume of resulting machine.

A dedicated case is designed with a 3D CAD application and built in ABS with a 3D printer. Case is realized to contain just the motherboard PCB and external PCI boards. ABS has been chosen for its good stiffness. Indeed, considering that such machine is used in an engine test bench room, the material fragility is compatible with the final application. Thickness of external walls is 2mm and in this way, resulting dimensions of RT machine are the following:

- Width: 210mm
- Depth: 210mm
- Height: 170mm

This makes target machine very compact and portable. Case is designed in two parts: a lower and an upper case to allow opening and closing the case. Its geometry allows to use such device with a vertical or horizontal orientation. On the lower part profiles for the PCB support screws (on the bottom), the supply LED and switch hole (on the left), the PCB supply connector hole (on the right) and the motherboard integrated, and external ports are prepared. On the upper case there is the grid for the CPU fan. Two parts are assembled just with interference and screws to allow a quick access to PCB and external boards if needed. Figure 5.25 displays some images of case.

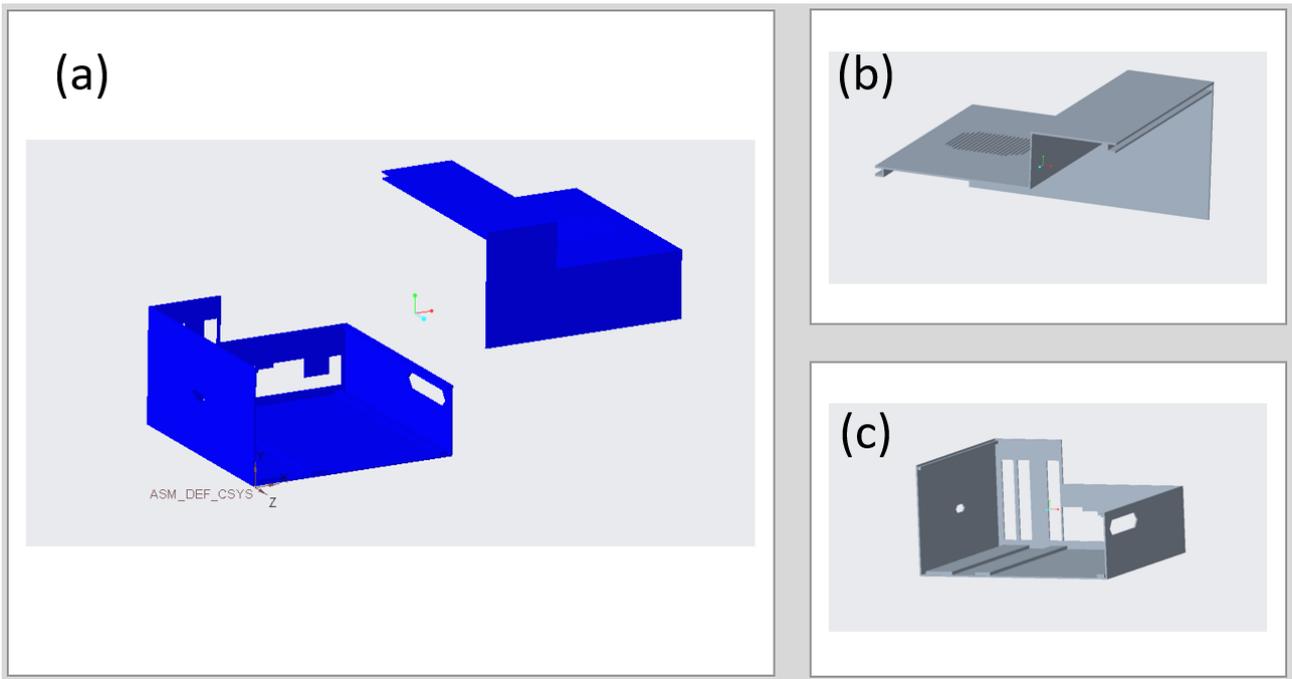


Figure 5.25: Target device case: exploded view (a), upper case (b), lower case (c).

RT machine is configured for the specific activity that leads to test innovative spark timing (and injected water mass) control systems, such as the model-based algorithms described in chapter 1.4 and 2.4. A Softing CAN board and Contec analog and digital I/O board are mounted on Asus motherboard. Softing board (Figure 5.26\_a) provides two high-speed CAN ports while Contec (Figure 5.26\_b) one has 12 differential analog input (or 16 single ended AI), 4 digital I/O and 2 analog output channels that can be connected with a break-out board for the 37 pin DSUB connector.



Figure 5.26: (a) Softing CAN AC2 PCI board and (b) Contec AD12-16E.

R8125 ethernet board is installed in the free mPCIe port. Figures 5.27, 5.28, 5.29 show some images of RT machine while Figure 5.30 reports the break-out board for analog and digital I/O connector.

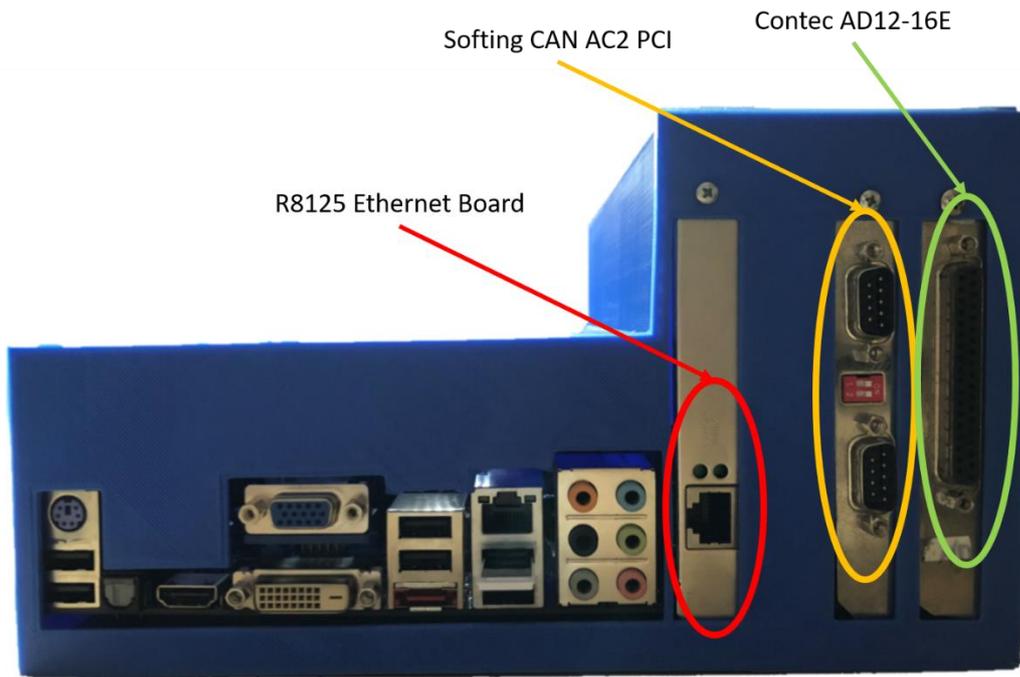


Figure 5.27: Rear view of RT machine.

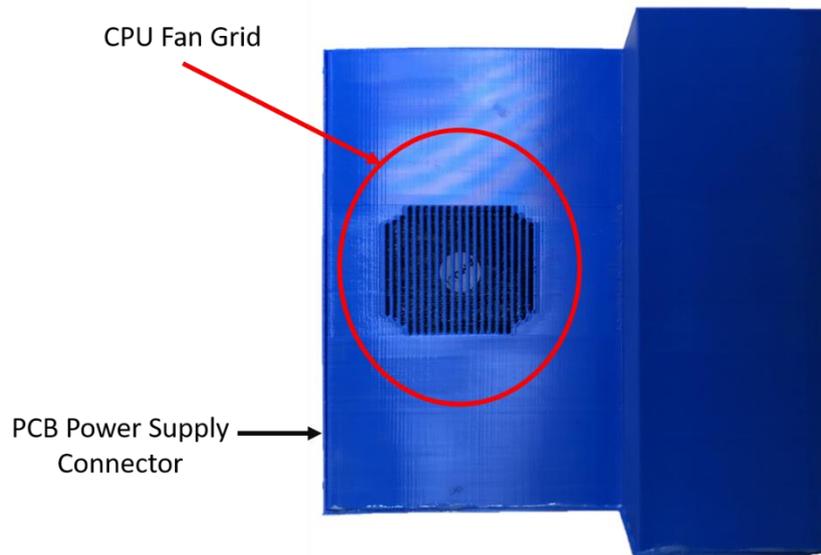


Figure 5.28: Top view of RT machine.

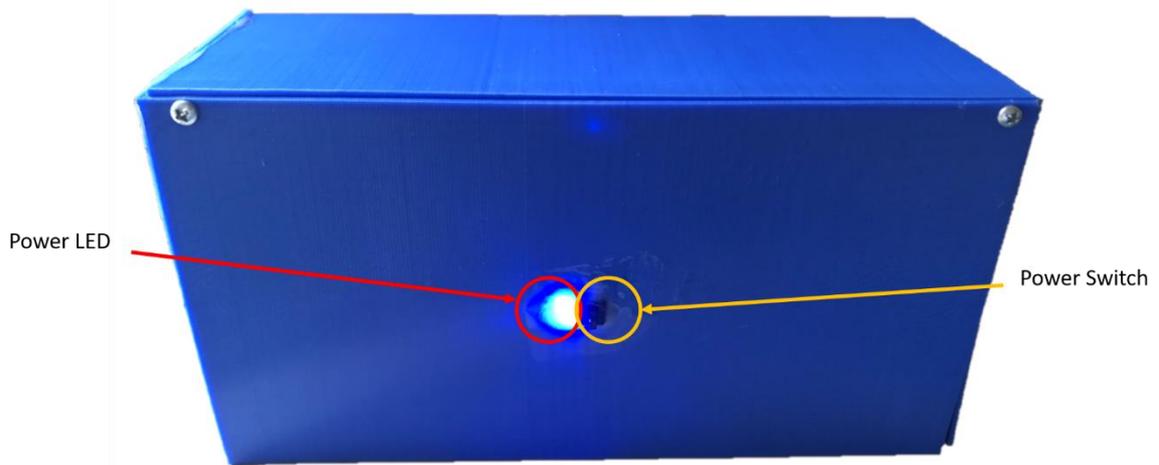


Figure 5.29: Side view of RT machine.

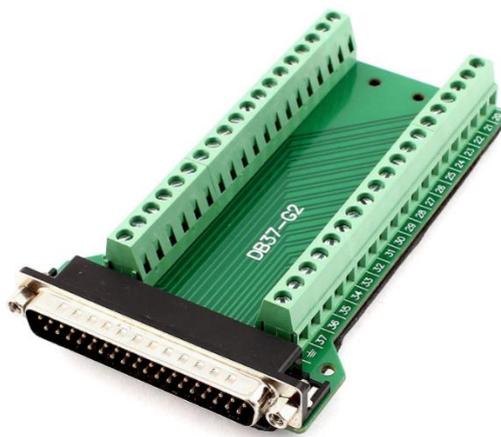


Figure 5.30: 37pin DSUB break-out board.

## 5.9. Conclusions

Custom target machine realized during PhD course allows to exploit high computational power of cooled CPU for desktop PC to test advanced control algorithm in real time, with an extremely cheap solution. Device can be connected to the ECU and other systems of test bench room via CAN and some engine control parameters and signals can be directly controlled, logged and analyzed. Custom RT target has been applied for the validation of the Water-Injection control algorithm developed for a prototyping Port Water Injection system that was the focus of an activity that has been performed in collaboration with Magneti Marelli and it is currently used in the context of another important collaboration with Ferrari. The device is now a robust component of the instrumentation used at the test bench. Figure 5.31 displays CAN network used at the test bench room of Industrial Engineering Department and connected devices are reported.

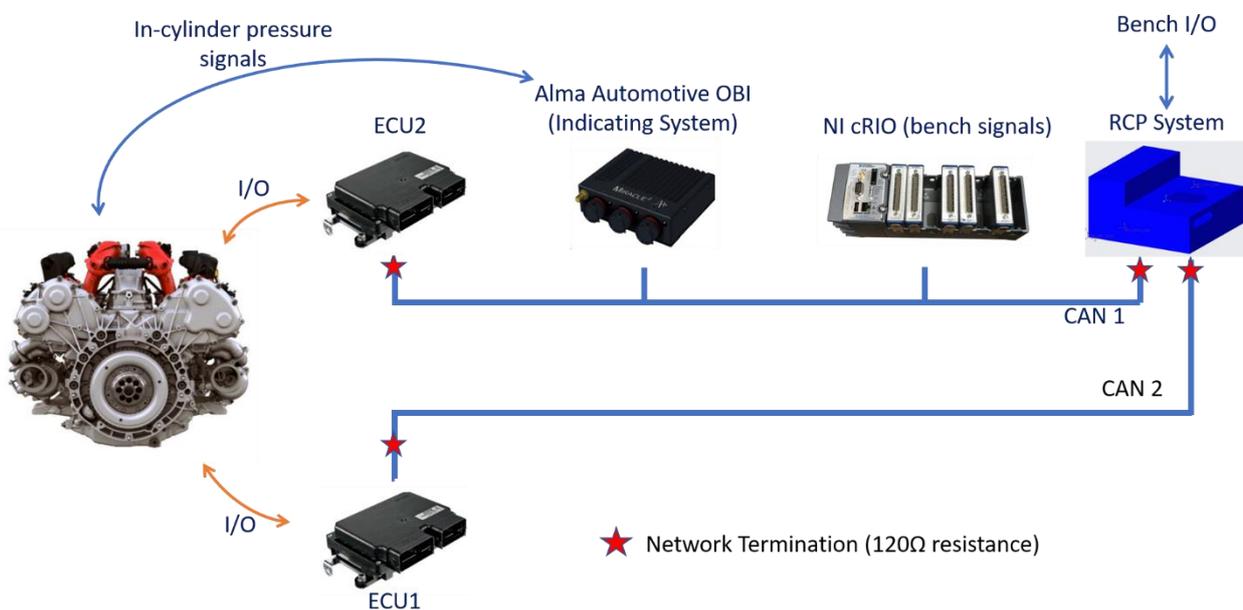


Figure 5.31: CAN networks layout and connected devices.

## References

- 4.1. Simulink Real Time online Documentation, The Mathworks, <https://it.mathworks.com/products/simulink-real-time>.
- 4.2. Simulink xPC DAT online Documentation, The Mathworks, <https://it.mathworks.com/matlabcentral/fileexchange/20415-xpc-target-driver-authoring-tool-tutorial>.
- 4.3. Matlab Documentation, The Mathworks, <https://it.mathworks.com/help/matlab>.
- 4.4. Patzer, A., Zaiser, R., "XCP-The Standard Protocol for ECU Development", Vector online documentation.