# DEVELOPMENT OF A MODEL-BASED AUTO-CALIBRATION METHODOLOGY WITH GENETIC ALGORITHMS FOCUSING ON FUEL EFFICIENCY AND EMISSION COMPLIANCE OVER REAL-WORLD DRIVING CYCLES

by

Egemen Karabıyık

B.S., Mechanical Engineering, Boğaziçi University, 2014

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Mechanical Engineering
Boğaziçi University

2020

# ACKNOWLEDGEMENTS

First of all, I would like to thank my thesis supervisor Prof. Evren Samur for his great guidance and supportive attitude. Also, I sincerely thank Prof. Sinan Öncü for his valuable feedbacks which contributed a lot to this study. I would also like to thank Prof. Orkun Özener who accepted to be a jury member of my master's thesis despite his busy schedule.

I am grateful to my friends in AVL Research and Engineering Turkey for their contributions to my thesis, including Ertuğrul Akbay, Caner Bayburtlu and Uğur Yavaş for their ideas, Tugay Solmaz for his collaboration with the optimization methods, Kübra Ekinci for her guidance on neural network modeling, Sarper Özkaynak and Yücel Karaca for their support on engine modeling.

The biggest thanks goes to my beautiful wife Ayşegül Sarı Karabıyık for her patience and encouragement during my study. Finally, I would like to thank my lovely sister Ceren Karabıyık, my energetic mother Nursel Karabıyık and my supportive father İsmail Karabıyık for their trust in me.

# ABSTRACT


## DEVELOPMENT OF A MODEL-BASED AUTO-CALIBRATION METHODOLOGY WITH GENETIC ALGORITHMS FOCUSING ON FUEL EFFICIENCY AND EMISSION COMPLIANCE OVER REAL-WORLD DRIVING CYCLES

The aim of this study is to develop a methodology for the optimization of a diesel engine calibration in terms of fuel consumption while being emission compliant for a predefined route by using model-based calibration techniques with genetic algorithms. A 2 liter diesel engine of light commercial vehicle data was used to develop the proposed methodology. The model-based calibration environment was structured in Simulink with a combination of an engine control unit (ECU), internal combustion engine (ICE) and engine aftertreatment system (EAS) models. Accuracies of the models were examined and considered as adequate for this methodology development study. After the models were created, the calibration domain for a predefined route was computed in Matlab by weighting the engine operating points based on frequency and fuel consumption. These points in the ECU model were defined as the optimization domain for the genetic algorithm in Matlab. The cost function of the optimization consists of fuel consumption, performance parameters, mechanical limits of ICE, and tailpipe emission limits defined by regulations. After iterating with different genetic algorithm configurations, the fuel consumption over the predefined RDE route was decreased by 3.07% at the end of simulations performed in Simulink. The smooth calibration maps were obtained while not violating any of the limits defined in the cost function. In addition to model accuracy improvements, for further study, the models can also be embedded into an ECU and optimization can be performed automatically according to the defined route, weather and driver information.

# ÖZET

## GERÇEK SÜRÜŞ ORTAMINDA YAKIT TASARRUFU VE EMİSYON SERTİFİKASYONUNU HEDEFLEYEN, MODEL BAZLI OTOMATİK KALİBRASYON METODUNUN GENETİK ALGORİTMALAR İLE GELİŞTİRİLMESİ

Bu çalışmanın amacı, gerçek sürüş koşulları altındaki yakıt tüketimi ve emisyon limitlerini hedefleyen dizel motor kalibrasyon methodolojisini, model bazlı kalibrasyon tekniklerini ve genetik algoritmalar ile geliştirmektir. Methodun geliştirilmesinde bir hafif ticari araca ait 2 litre dizel motor verisi kullanılmıştır. Model bazlı kalibrasyon ortamı, Simulink yazılımı üzerinde elektronik kontrol ünitesi (EKÜ), içten yanmalı dizel motor ve egzoz emisyon gazı indirgeme sistem modellerinin birleşmesiyle oluşturulmuştur. Modellerin doğruluğu test sonuçlarında belirtilmiş ve bu methodoloji geliştirme çalışması için yeterli bulunmuştur. Modeller geliştirildikten sonra, kalibrasyon yapılacak bölge, Matlab içerisinde motor hızı ve tork bazlı motor operasyon noktalarının frekans ve yakıt tüketimlerine göre hesaplanmıştır. Bu noktalar, EKÜ modeli içerisindeki haritalar üzerinde genetik algoritmalar için kalibrasyon yapılacak alan olarak belirtilmiştir. Sürüş sırasındaki yakıt tüketimi ve performans hedefleri, motorun mekanik limitleri ve yasalar tarafından belirlenen emisyon limitleri optimizasyon optimizasyon amaç fonksiyonu içerisinde tanımlanmıştır. Simulink üzerinde yapılan simülasyon çalışmalarının sonuçlarına göre, farklı genetik algoritma konfigürasyonlarının denenmesi sonucunda belirlenen rota üzerinde yakıt tüketiminde 3.07% iyileştirme sağlanmıştır. Ayrıca, bu sonuç sırasında amaç fonksiyonunda belirtilen hiçbir limit aşılmamıştır.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF SYMBOLS

| | |
|---|---|
| $A$ | Coefficient of the linear equation |
| $B$ | Offset of the linear equation |
| $BP_{lim}$ | Limit for boost pressure |
| $BP_{sim}$ | Simulation value for boost pressure |
| $Bp_{x1}$ | First breakpoint on the x axis |
| $Bp_{x2}$ | Second breakpoint on the x axis |
| $Bp_{y_1}$ | First breakpoint on the y axis |
| $Bp_{y_1}$ | Second breakpoint on the y axis |
| $COT_{lim}$ | Limit for compressor out temperature |
| $COT_{sim}$ | Simulation for compressor out temperature |
| $Data_x$ | Data belongs to x axis |
| $Data_y$ | Data belongs to y axis |
| $Data_z$ | Data belongs to z axis |
| $f(u)$ | Activation function of neural networks |
| $FUEL_{lim}$ | Limit for fuel quantity |
| $FUEL_{sim}$ | Simulation value for fuel quantity |
| $J$ | Cost function |
| $k_{BP}$ | Factor for boost pressure |
| $k_{COT}$ | Factor for compressor out temperature |
| $k_{FUEL}$ | Factor for fuel quantity |
| $k_{MCP}$ | Factor for maximum cylinder pressure |
| $k_{NE}$ | Factor for $NO_x$ emission |
| $k_{NHE}$ | Factor for $NO_x$ and HC emission |
| $k_{PE}$ | Factor for PM emission |
| $k_{RP}$ | Factor for rail pressure |
| $k_{TIT}$ | Factor for turbine inlet temperature |
| $k_{TM}$ | Factor for torque match |
| $MCP_{lim}$ | Limit for maximum cylinder pressure |

| | |
|---|---|
| $MCP_{sim}$ | Simulation value for maximum cylinder pressure |
| $n$ | Number of inputs of neural networks *or* number of modeled data points |
| $NE_{lim}$ | Limit for $NO_x$ emission |
| $NE_{sim}$ | Simulation value for $NO_x$ emission |
| $NHE_{lim}$ | Limit for $NO_x$ and HC emission |
| $NHE_{sim}$ | Simulation value for $NO_x$ and HC emission |
| $NRMSE$ | Normalized Root Mean Square Error |
| $PE_{lim}$ | Limit for PM emission |
| $PE_{sim}$ | Simulation value for PM emission |
| $R^2$ | R-squared Error |
| $RMSE$ | Root Mean Square Error |
| $RP_{lim}$ | Limit for rail pressure |
| $RP_{sim}$ | Simulation value for rail pressure |
| $sign()$ | Sign of the input |
| $TIT_{lim}$ | Limit for turbine inlet temperature |
| $TIT_{sim}$ | Simulation value for turbine inlet temperature |
| $TM_{lim}$ | Limit for torque match |
| $TM_{sim}$ | Simulation value for torque match |
| $u$ | Input of artificial neural network activation function |
| $w_i$ | Weight of $i_{th}$ input of neural networks |
| $W_1$ | Weight of the first breakpoint on the x axis |
| $W_2$ | Weight of the second breakpoint on the x axis |
| $W_1$ | Weight of the first breakpoint on the y axis |
| $W_2$ | Weight of the second breakpoint on the y axis |
| $W_{(x1,y1)}$ | Weight of the point on the first breakpoints on the x and y axis |
| $W_{(x1,y2)}$ | Weight of the point on the first breakpoint on the x axis and second breakpoint the y axis |
| $W_{(x2,y1)}$ | Weight of the point on the second breakpoint on the x axis and first breakpoint the y axis |

| | |
|---|---|
| $W_{(x2,y2)}$ | Weight of the point on the second breakpoints on the x and y axis |
| $x$ | Input of neural networks *or* Input of the linear equation |
| $x_{actual}$ | Actual output value to be modeled |
| $\bar{x}_{actual}$ | Mean of the actual output value to be modeled |
| $x_{actual,max}$ | Maximum of the actual output value to be modeled |
| $x_{actual,min}$ | Minimum of the actual output value to be modeled |
| $x_{model}$ | Model output value |
| $y$ | Output of the linear equation |
| $\alpha$ | Threshold of threshold function *or* exponential power of sigmoid function |
| $\theta$ | Bias term of neural networks |
| $\kappa$ | Factor of linear function *or* ramp of ramp function |

# LIST OF ACRONYMS/ABBREVIATIONS

| | |
|---|---|
| 2D | Two Dimensional |
| ANN | Artificial Neural Network |
| ASC | Ammonia Slip Catalyst |
| AVL | Anstalt für Verbrennungskraftmaschinen List |
| BFG | BFGS Quasi-Newton |
| BFGS | Broyden–Fletcher–Goldfarb–Shanno |
| BR | Bayesian Regularization |
| BSFC | Brake Specific Fuel Consumption |
| CO | Carbon Monoxide |
| $CO_2$ | Carbon Dioxide |
| DCU | Dosing Control Unit |
| DOC | Diesel Oxidation Catalyst |
| DoE | Design of Experiment |
| DPF | Diesel Particulate Filter |
| EAS | Engine Aftertreatment System |
| ECU | Electronic Control Unit |
| EGR | Exhaust Gas Recirculation |
| EKÜ | Elektronik Kontrol Ünitesi |
| EMS | Engine Management System |
| EPA | Environmental Protection Agency |
| EU | European Union |
| FTP | Federal Test Procedure |
| GA | Genetic Algorithm |
| HC | Hydro Carbon |
| ICE | Internal Combustion Engine |
| LEV | Low Emission Vehicle |
| LM | Levenberg-Marquardt |
| LNT | Lean Nx Trap |

| | |
|---|---|
| MSE | Mean Squared Error |
| NEDC | New European Driving Cycle |
| $NH_3$ | Ammonia |
| $NO_x$ | Nitrogen Oxide |
| NRMSE | Normalized Root Mean Square Error |
| OEM | Original Equipment Manufacturer |
| PEMS | Portable Emission Measurement System |
| PID | Proportional Integral Derivative |
| PM | Particulate Matter |
| PS | Particle Swarm |
| PSO | Pattern Search Optimization |
| $R^2$ | R-squared Error |
| RDE | Real Driving Emissions |
| RMSE | Root Mean Square Error |
| RNN | Robust Neural Networks |
| RP | Resilient Backpropagation |
| SA | Simulated Annealing |
| SCG | Scaled Conjugate Gradient |
| SCR | Selective Catalyst Reduction |
| SSE | Sum Squared Error |
| UNECE | United Nations Economic Commission for Europe |
| USA | United States of America |
| WLTC | Worldwide harmonized Light Vehicles Test Cycle |

# 1. INTRODUCTION

## 1.1. Background

### 1.1.1. Engine Controls & Calibration

The complexity in automotive engines has been increasing significantly in order to meet stringent performance targets, emissions limits and fuel economy at the same time, under different ambient conditions such as high altitude or cold environment. In accordance with market requirements for engine development, engineers responded with engine electronic control units (ECUs) with complicated emission, air and combustion control algorithms. To achieve accurate control of the parameters affecting combustion and emissions, calibration engineers try to optimize the setpoint maps, actuator controller gains and other emission and performance relevant parameters in the ECU [1].

New technologies for engine control, such as exhaust gas recirculation (EGR) valves, $NO_x$ reduction catalysts or variable geometry turbines, were introduced to diesel engines. Thanks to the implementation of diesel engine technologies, the number of control parameters inside the ECUs have increased. Besides the 2D lookup tables used as setpoints of control parameters and PID gains of actuators, different control algorithms including factors and offsets have been implemented in ECUs to provide precise control over the transients [1]. In conclusion, the optimization of diesel engine calibration, for obtaining less pollutant emission and fuel consumption at the same time has become a crucial and complicated process for diesel engine calibration.

In today's diesel engine control units, there exist more than 30,000 calibratable parameters to control the emissions, fuel consumption, performance, diagnosis and auxiliary functions [1]. Therefore, calibration of a complex system like this requires a considerable amount of effort, know-how and expensive facilities. Engine dynamometers, chassis rollers and climate control chambers are some of the expensive test environ-

ments used for testing and calibration processes. To decrease the time spent and cost during the engine calibration process, an efficient method, called model-based calibration, was introduced to the calibration world. By using an accurate engine model, an emission test can be simulated as accurate as it is performed on a dynamometer. However, to reach this accuracy for all engine operating points, a considerable amount of data is required. Design of Experiments (DoEs) have been used to generate the testing matrix required to build an engine model. By using the DoE method, dynamometer costs have been decreased significantly as the number of measurement points has been decreased. After the testing matrix was performed on the dynamometer, the recorded data is utilized to build an engine response model.

Thanks to model-based simulations, calibration engineers can optimize their calibration dataset on their PCs. The focused engine operating points are defined based on the target emission cycles. These operating points are optimized locally based on the optimization targets such as emissions or fuel consumption. By using this locally optimized points, global optimization is performed based on target emission cycles [2].

### 1.1.2. Emission Reduction Strategies

To start with engine-out emissions, $NO_x$, HC, CO, $CO_2$ and particulate matter (PM) occur due to incomplete combustion of diesel fuel. Secondary systems such as fuel path and air path systems, play a big role in the reduction of toxic gases that come out of the combustion chamber [3, 4]. Variation of parameters such as boost pressure or temperature, exhaust gas recirculation, injection splitting, timing, and pressure can easily affect the pollutant ratio of exhaust flow [3, 5, 6]. In Table 1.1, these emission control parameters are listed.

Table 1.1: Engine out emission control strategies [3, 6].

| **Emission Control Parameter** | **Emission Impact** | **Additional Information** |
|---|---|---|
| Injection Timing | Reduction in $NO_x$. | Retarded combustion phasing can be used to reduce $NO_x$. |
| Injection Pressure | Reduction in PM. | While EGR can increase PM, higher injection pressure can reduce it. |
| Splitted Injections | Reduction in HC, CO, PM and $NO_x$. | Multiple injection strategies such as pilot-main-late injections can reduce HC, CO, PM and $NO_x$. |
| Exhaust Gas Recirculation(EGR) | Reduction in $NO_x$. | EGR decreases $NO_x$ by decreasing the $O_2$ content in combustion but increases HC, CO and PM. |
| Boost Pressure | Reduction in PM. | Due to the higher boost pressure,higher air-fuel ratio for settled fuel amount can decrease PM. Performance issues due to EGR, can also be solved. |
| Intake Air Temperature | Reduction in $NO_x$. | Higher intake temperature due to boosted air or EGR, can increase $NO_x$. Cooling of charged air can limit $NO_x$ while increasing power density. |

As for tailpipe emissions, they must pass through the emission cleaners called exhaust gas aftertreatment systems just before they are released to the environment from the tailpipe. Different types of catalysts exist for different types of pollutants [7]. Moreover, many smart technologies have been developed according to the needs of emission regulations. In Table 1.2, some of the latest technologies used in diesel engines are listed.

Table 1.2: Tailpipe emission control systems [3, 5–7].

| **Aftertreatment System** | **Effect on Emissions** | **Additional Information** |
|---|---|---|
| DOC: Diesel Oxidation Catalyst | High reduction of HC and CO. Oxidation of NO to $NO_2$ increases the performance of SCR and DPF. | Conversion of HC and CO to $CO_2$ and $H_2O$. Small effect on PM conversion. |
| DPF: Diesel Particulate Filter | High reduction of PM. | Cleaning of filters is done by regeneration. |
| SCR: Selective Catalyst Reduction | High reduction of $NO_x$. | Urea is injected to upstream of SCR to convert $NO_x$ to $N_2$ and $H_2O$. |
| LNT: Lean $NO_x$ Trap | Passive reduction of $NO_x$. | Decomposition of $N_2$ is done by regeneration. |
| ASC: Ammonia Slip Catalyst | Conversion of excessive $NH_3$. | Prevention of injected excessive $NH_3$ to go out from tailpipe. |

### 1.1.3. Emission Regulations & Certification Cycles

Starting from 1970 with the "Clean Air Act" in United States until today's regulations all over the world, authorities have been enforcing car manufacturers and drivers to have eco-friendly vehicles with reduced tailpipe emissions [8]. In Europe, EU Commission has been following Euro-6d Temp emission standards since 2017 while in the USA, emissions have been monitored according to Tier 3 and LEV III standards which are based on the location of the state.

Today, the emission standards are controlled by emission certification cycles selected by the authorities such as the Environmental Protection Agency (EPA) or the United Nations Economic Commission for Europe (UNECE). The emission levels of the vehicles are evaluated with different driving cycles under different conditions. The latest emission standard (Euro 6d-Temp) for our reference light commercial vehicle type N2 with diesel engine, requires two different driving cycles to be tested [9]. First one, New European Driving Cycle (NEDC), consists of steady-state driving with gentle accelerations in urban and highway phases as shown in Figure 1.1.

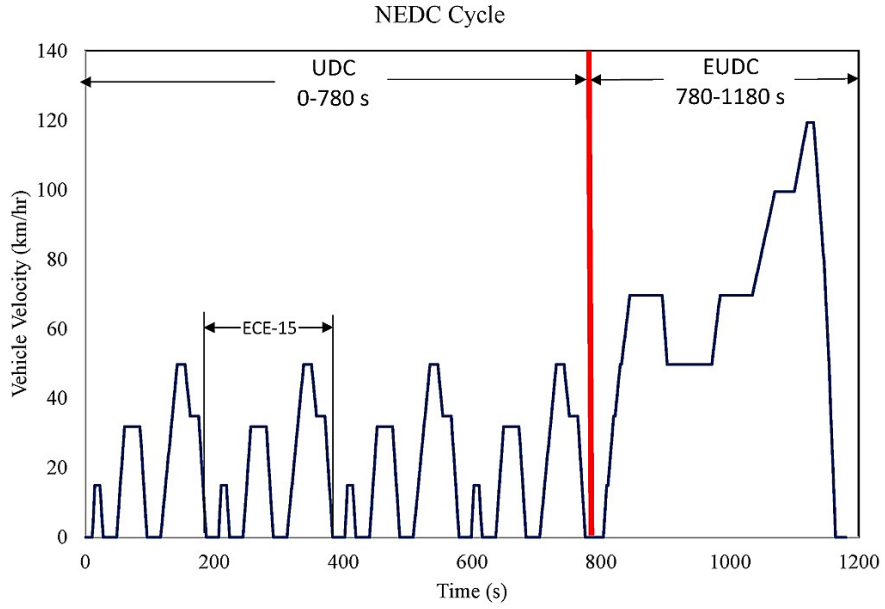



Figure 1.1: NEDC consists of two phases such as 780 seconds of urban driving cycle (UDC) and 400 seconds of extra urban driving cycle (EUDC) (reprinted from [10]).

As a replacement of NEDC, Worldwide harmonized Light Vehicles Test Cycle (WLTC) has been introduced with Euro 6 emission standards. As opposed to NEDC, this driving cycle is more realistic thanks to more transient driving characteristics. There exist 3 different WLTC categories for different vehicle classes in terms of power/weight ratio [11]. In Figure 1.2, WLTC class 3 version 3.2 is illustrated.

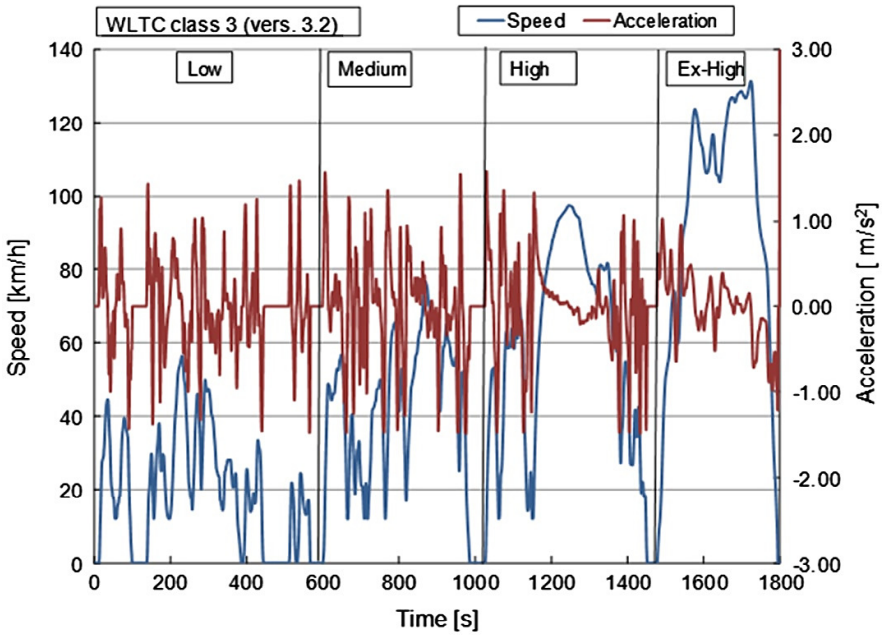



Figure 1.2: Worldwide harmonized Light Vehicles Test Cycle (WLTC) (reprinted from [11]).

The last conformity test is called real-world driving emissions (RDE) cycle. As can be understood from its name, RDE is considered as the most representative cycle for real life driving. While the NEDC and WLTC are performed on the chassis dyno in the laboratory, the RDE cycle is performed on real roads [12]. The measurements of exhaust gases are recorded by a portable emission measurement system called PEMS. Any driven cycle with PEMS recording can be considered as RDE cycle as long as it satisfies the rules defined by the regulations. These rules ensure that the driven cycle is representative of the characteristics of urban, rural and highway road types with normal driving conditions (not too aggressive, not too smooth) and under normal environmental conditions (0°C $<$ Ambient Temperature $<$ 30°C and Altitude $<$ 700 m) [13]. In case of extreme conditions like high altitude or cold ambient, the emission limits are extended with factor of 1.6 [9]. The RDE validity criteria are described in

1.3.

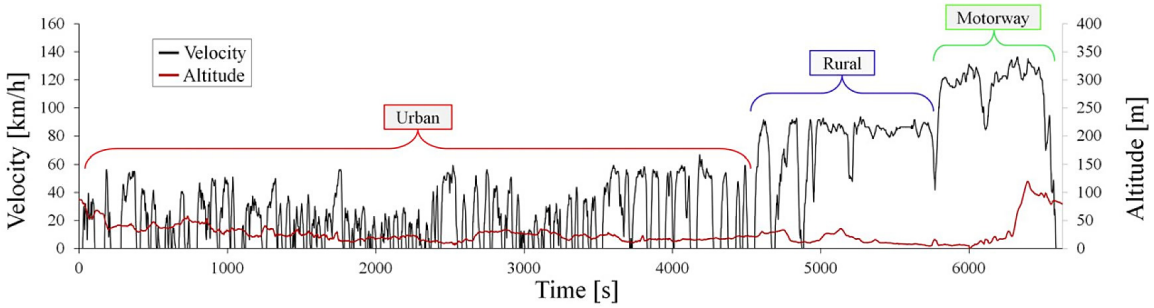



Figure 1.3: Real-World Driving Emissions (RDE) Cycle (reprinted from [12]).

Table 1.3: RDE cycle validity criteria in terms of speed, distance and altitude. [13]

| **Limits** | **Unit** | **Urban** | **Rural** | **Highway** |
|---|---|---|---|---|
| Speed interval | km/h | Speed $\leq$ 60 | 60 $<$ Speed $\leq$ 90 | 90 $<$ Speed |
| Distance share | % | 29 - 44 | 23 - 43 | 23 - 43 |
| Total trip duration | min | 90 - 120 | 90 - 120 | 90 - 120 |
| Average speed with stops | km/h | 15 $<$ Speed $<$ 40 | | |
| Maximum total stop duration | % | 6 $<$ % in time $<$ 30 | | |
| Maximum stop time for each stop | sec | 300 | | |
| Maximum speed | km/h | | | 160 |
| Maximum duration for speed $>$ 145 km/h | % | | | 3 (of highway time) |
| Cumulative positive elevation | m/100 km | 1200 | 1200 | 1200 |
| Elevation difference of start and stop | m | 100 | 100 | 100 |

Moreover, driver-based excessive differentiation is prevented by dynamic trip requirements. Extremely smooth driving, to pretend releasing low emissions or extremely aggressive driving to show a high level of toxic emissions, is taken into consideration by trip dynamics rules as well [14]. The examples for violation of low and high dynamic boundary conditions are illustrated in Figure 1.4.

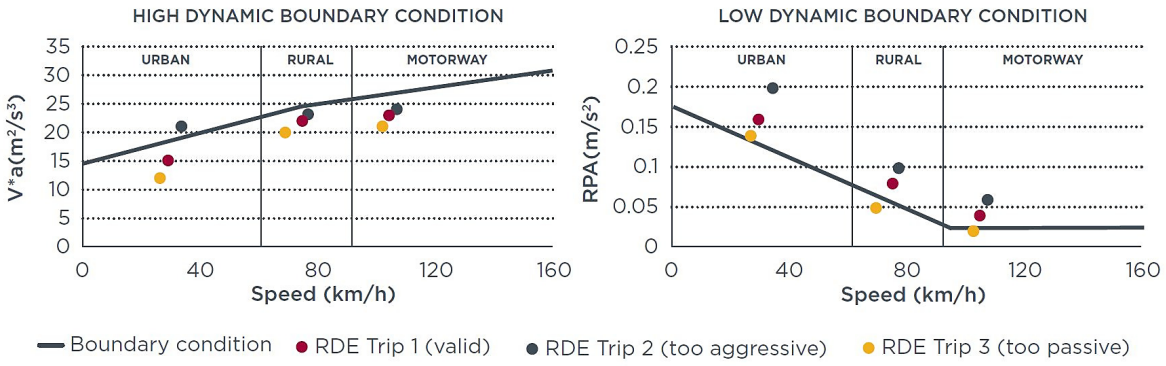



Figure 1.4: RDE cycle validity criteria in terms of trip dynamics (reprinted from [14]).

### 1.1.4. Artificial Neural Networks

The history of Artificial Neural Networks (ANNs) started in 1943, with the binary threshold unit proposed by McCulloch and Pitts [15]. Recently, ANNs have become popular in many disciplines to model complex real-world problems including non-linearity and imprecise information. ANNs are used widely in pattern recognition, clustering, function approximation, forecasting, control and optimization problems [16].

The artificial neuron depicted in Figure 1.5 has n inputs ($x_1$,$x_2$,$x_3$,...$x_n$) and one bias term ($\theta$). Each input term is multiplied with corresponding weights ($w_1$,$w_2$,$w_3$,...$w_n$) before the summation operation. These weighting factors in artificial neurons resemble the synaptic connections in biological neurons [17]. The summation result passes through an activation function (f(u)) to produce the output (y) signal.

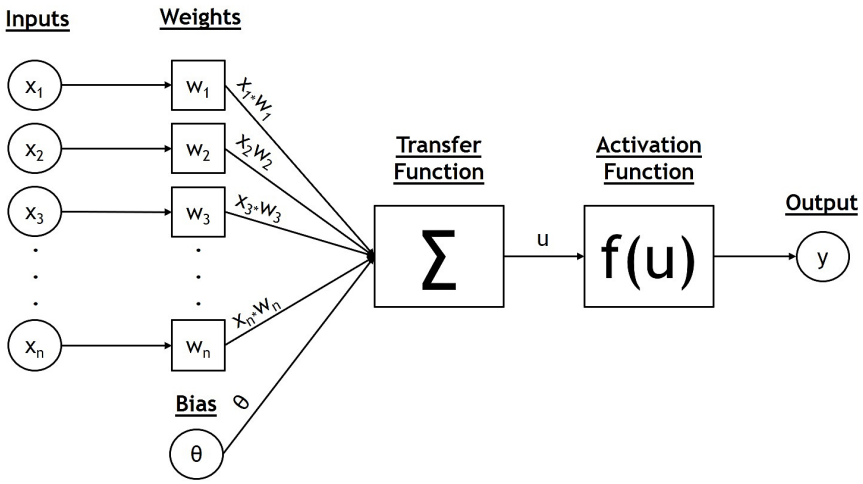



Figure 1.5: Artifical neuron structure.

The relation between the inputs and the output are given with the following formulas of transfer and activation functions.

- **Transfer function:**

$$u = \left( \sum_{i=1}^{n} w_i * x_i \right) + \theta \tag{1.1}$$

- **Activation function:**

$$y = f(u) \tag{1.2}$$

The first proposed activation function by McCulloch and Pitts, was a simple threshold function [15]. However, there exist more activation functions in the literature now. These are the commonly used activation functions:

- **Threshold function:**

$$f(u) = \begin{cases} 0, & a <= 0 \\ 1, & a > 0 \end{cases} \tag{1.3}$$

- **Linear function:**

$$f(u) = \kappa u \tag{1.4}$$

- **Ramp function:**

$$f(u) = \begin{cases} 0, & u <= 0 \\ u/\kappa, & 0 < u < \kappa \\ 1, & u > \kappa \end{cases} \tag{1.5}$$

- **Sigmoid function:**

$$f(u) = \frac{1}{1 + e^{-\kappa a}} \tag{1.6}$$

- **Hyperbolic tangent function:**

$$f(u) = \frac{e^{u} - e^{-u}}{e^{u} + e^{-u}} \tag{1.7}$$

Sometimes, a single neuron is not enough to solve complicated nonlinear problems. To overcome these problems, a multi-layer network of neurons can be constructed by connecting more than one neuron to each other consecutively [17]. In Figure 1.6, a feedforward multi-layer neural network structure, consisting of an input layer, hidden layers and an output layer, is illustrated.

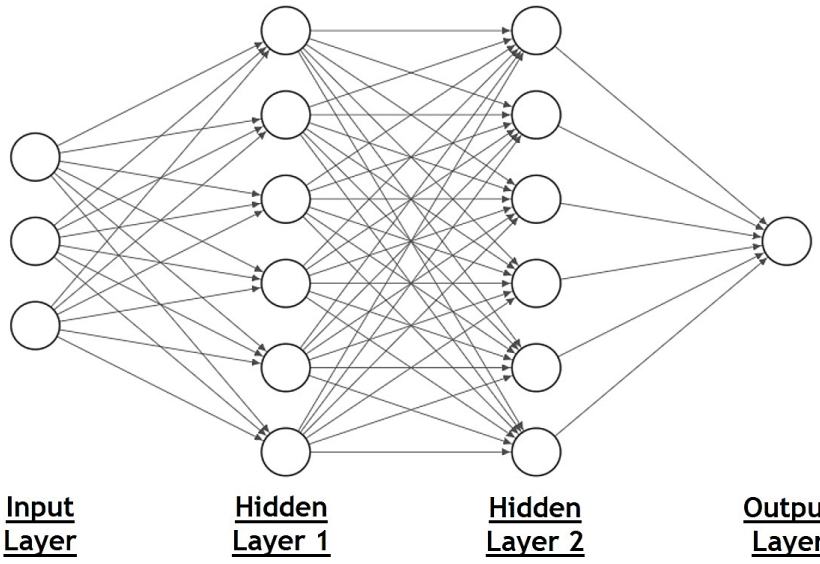



Figure 1.6: Multi-layer feedforward ANN structure.

In addition to single and multi-layer neural networks, there exist many different networks such as Hopfield networks, radial basis function networks, competitive networks, Kohonen networks, recurrent networks, adaptive resonance theory networks, neuro-fuzzy networks and etc. [18]. In the scope of the learning algorithms of ANNs, error-backpropagation is considered as the most commonly used supervised learning algorithm. It is based on gradient descent for points with the minimum error which is mainly a function of weights. The algorithm works in two steps: going forward through layers to calculate the output, and then going back to update the weights based on the calculated error [16]. The learning scheme is demonstrated in Figure 1.7.

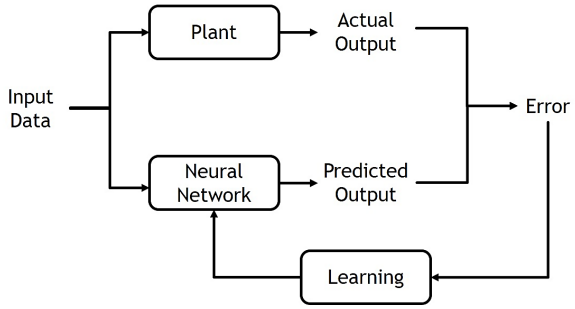



Figure 1.7: Backpropagation learning scheme.

### 1.1.5. Genetic Algorithms

To solve optimization problems, there exist many numerical methods such as simple gradient method, steepest descent method, and sequential quadratic programming. However, these methods have some limitations such as a local minima or low performance in case the wrong initial value is chosen. Furthermore, these numerical methods can only be used in optimization problems which are defined as continuously differentiable functions [19]. As opposed to numerical methods, genetic algorithms (GAs) can avoid being trapped in local minima due to their derivative-free behavior. Moreover, GAs can alos solve problems when the objective function is nonlinear, stochastic, discontinuous or nondifferentiable [20].

In Figure 1.8, the workflow of the genetic algorithm is illustrated. The update of the population starts with the selection of individuals randomly from the current population to be parents. The crossover function uses these randomly matched parents, that have lower cost, for creating the children for the next generation. In case the convergence criteria are met, successive generations can evolve the population to create the best solution for the optimization [20].

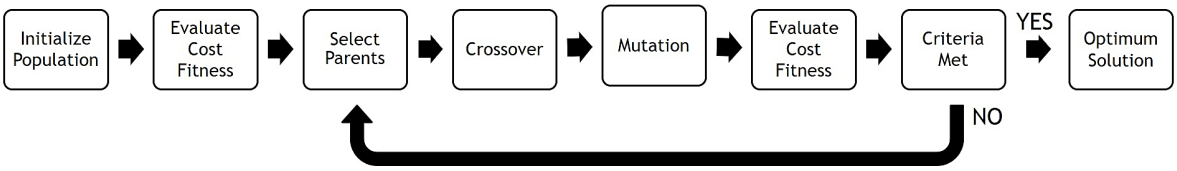



Figure 1.8: The Genetic Algorithm workflow.

In addition to GA, there exist other derivative-free algorithms such as Pattern Search (PS), Particle Swarm Optimization (PSO), Simulated Annealing (SA) and etc. Furthermore, these algorithms can be hybridized to increase the speed of convergence as well as the chance to find the global minimum [21]. However, in this study, we will focus on genetic algorithms and their optimum configuration.

## 1.2. Motivation

Together with the legislative requirements, car manufacturers also have their own specific targets in terms of fuel consumption, performance, and drivability characteristics. These hard targets make calibration process long and costly. To decrease cost and time spent on calibration of engine control parameters, model-based approaches were developed. Despite their increasing popularity, majority of the calibration process is still being done by real-world tests instead of model-based simulations.

Most of the model-based engine calibration methodologies rely on physical models of engine and aftertreatment systems. However, building the physical models requires extreme know-how on engine behavior, input data and time. Also, the physical models work very slowly compared to the empirical models due to the computation time of highly complicated mathematical models inside the physical models. Once the models are built, the target engine operating points are defined by the calibration engineers according to certification emission cycles such as NEDC or WLTC. During the optimization, these points are weighted based on their frequency. In case of definition of optimization domain by frequency relation, some points can be discarded. Even though they are not the most often encountered ones, these engine operating points may have a larger impact on optimization cost more than the frequent points.

The remaining points are calibrated manually with different real-world driving data by the calibration engineers based on the emission test results. The main problem with this method is that it takes too much time for an engineer to consider all interdependent control variables in the engine control unit. Also, the calibration values can differ based on the calibration engineers' experience, feeling and subjective analysis which may create robustness issues during the validation process with fleet data. Once the calibration dataset is frozen for production, it stays the same for the whole lifetime of ECU regardless of route or driver characteristics. To explain, there is still a margin to improve emission compliant calibrations for better fuel economy for specifically defined engine operating points, in other words, specific routes driven with specific driving characteristics.

## 1.3. Aims of the Study

The main objective of my thesis is to create a model-based engine calibration methodology which aims to decrease fuel consumption while taking emission limits into consideration by updating the reference calibration dataset by genetic algorithms. To reach this objective, the specific aims of this study are defined as below:

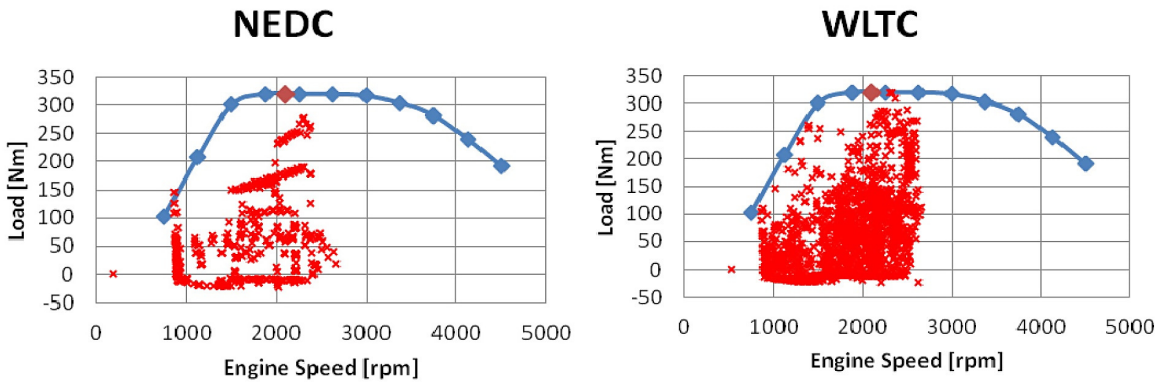
**Aim 1:** Create a model-based engine calibration environment by modeling ECU, ICE and EAS systems thanks to empirical data and neural networks. A reduced parameter set which are crucial for the optimization targets, are modeled with enough accuracy. The developed model-based environment is used to simulate the whole system with the calibration datasets generated by genetic algorithms.

**Aim 2:** Generate a calibration dataset automatically, specifically for a defined route, according to the cost function defined within genetic algorithms. Emission limits, calibration engineers' know-how, engines' mechanical constraints and car manufacturers' targets are included inside the cost function. The objective is to create a more fuel-efficient calibration dataset than the reference calibration within a shorter time than the manual calibration methods.

# 2. LITERATURE REVIEW

## 2.1. Route-Based Calibration

In route based calibration activities, an efficient method called DoE for testing and calibration is used to collect sufficient amount of data to build an empirical engine model. Building an empirical engine model by using only a few measurements, instead of scanning the full space of engine operating points, make this method cost and time efficient for understanding the engine responses [22]. The space of the DoE is defined mainly by using the target emission cycles. In the last decade NEDC has been the main target cycle for emission calibration. Later, authorities offered WLTC to observe transient effects on emissions [23]. Recently, with the latest Euro 6d-Temp regulations, RDE has been introduced to have a more real-life representative emission cycle. Therefore, today's DoE's are shaped by these 3 cycles.




(a) NEDC (b) WLTC

Figure 2.1: Distribution of engine operating points of (a) NEDC and (b) WLTC (reprinted from [23]).

M. Castagné et al. [24] described different calibration methods based on DoE, especially focusing on the NEDC area. They examined local calibration approaches, global calibration approaches, and map smoothening methods as well. F. Millo et al. [25] investigated GA based optimizer with single and multiobjective cost functions to lower the brake specific fuel consumption (BSFC), engine-out soot and $NO_x$ emissions.

The calibration key points were defined based on the frequently encountered engine operating points during NEDC and WLTC, instead of using RDE key points as a basis. In 2008, a new engine calibration process called "Rapid Transient Calibration" was validated on a heavy-duty diesel engine [26]. The minimum fuel consumption was aimed during the Federal Test Procedure (FTP) with a cost function consisting of the mechanical constraints and engine-out emission-based constraints. Tailpipe emission limits couldn't be examined due to a lack of an aftertreatment system model.

As explained in the study of P. Polterauer and H. Waschl [27], just switching in between the optimum calibration datasets which were calibrated independently to have less emission or less fuel, couldn't decrease the multiobjective costs such as fuel consumption and $NO_x$ emissions at the same time. However, switching directly to a calibration dataset which is optimum in terms of fuel consumption while not exceeding the emission or the mechanical limits, created the motivation for this research. Although they did not focus on the calibration method, they were focusing on optimal control problems of optimum calibration dataset selection while considering the route based information. Due to the trade-off between the $NO_x$ emissions and fuel consumption, they utilized multiple calibration datasets, which were optimized for fuel consumption and $NO_x$ emissions separately. During the drive, the calibration switching problem is solved as a Boolean linear program focusing on either less fuel consumption or less $NO_x$ emission. As a result, 0.25% of fuel economy was gained while the emission constraints were violated by 10%.

A different approach was proposed by A. Malikopoulos [28] for optimal engine calibration in real-time for specific driving cycles. In this method, there exists a self-learning algorithm for controllers to select the optimum value in terms of fuel and emission. As an example, injection timing was calibrated in real-time via the proposed self-learning method. The results showed improvements in fuel consumption and emissions. However, due to the highly dynamic controller parameter changes, drivability could be an issue for this method. Also, since the controllers are learning from transients.

## 2.2. Model-Based Simulation

In the literature, there exist different methods for engine modeling such as physical, empirical or semi-physical models as G. Sujesh et al. [29] mentioned in his extensive research on the history of diesel engine modeling. Guzzella and Onder [30] explained all the components of mean value and discrete-event models in their book. Physical and semi-physical methods rely on physical modeling of all engine components and parameters by using high computation capable PC. This method requires a lot of time and know-how on diesel engines to build an accurate response model. As opposed to physics-based models, building data-driven empirical models are relatively easy and fast. Nowadays, neural networks and polynomial regression models have become increasingly popular in engine modeling. Rolf Iserman et al. described all steps of model-based development with experimental data. The procedure starts with variation space design and continues with signal excitation, main measurement, raw data analysis, training of the data and model validation, respectively [31].

In 2006, Alonso et al. [19] utilized a large number of stationary engine tests to build fuel consumption and engine-out emission models by artificial neural networks. To evaluate diesel engine control parameters' influences on emissions, Tauzia et al. [32] used semi-physical models. They investigated injection pressure and timing, EGR rate, boost pressure and pilot injection mass influence on $NO_x$ and soot emissions. Ismail et al. [33] modeled engine out responses of 9 main parameters of a light-duty diesel engine by using artificial neural networks. Although the inputs were similar to the diesel engine model presented in this research, the diesel engine was fueled with bio-diesel blends and there was no aftertreatment system or optimization algorithm combined for any optimization. In 2014, investigation of the effect of engine control parameters on diesel emissions and performance was done by utilization of neural network modeling [34]. Asprion et al. [35] combined the physical $NO_x$ formation model with a mean value engine model just to optimize the $NO_x$ emissions at the engine out. A set of transient maneuvers was used to build an artificial neural network-based dynamic engine model for a heavy-duty diesel engine [26]. Another investigation was done by Grahn et al. [36] on the effect of implementing boost pressure deviation based

transient correction factors to the Engine Management System (EMS). Although there are 4 separate models consisting of driver, vehicle, engine and EMS models, there was no aftertreatment system model implemented to the system.

## 2.3. Engine Optimization Methods

Once the engine model is obtained, the engine operating points for optimization and their weights are defined based on the target driving cycle. Then, the setpoint maps for emission and fuel consumption affecting parameters such as air mass flow, boost pressure, fuel injection quantity and angle, are optimized according to the emission regulations and the Original Equipment Manufacturer's (OEM's) specific targets. As the number of optimization parameters increased, the required time for the optimization, as well as the complexity is increased.

Diesel engine calibration optimization is a very complex problem also because of performance and fuel consumption targets, emission limits and engine's physical limitations. Recently, genetic algorithms (GAs) were introduced to the calibration optimization process in order to meet the requirements of multiple and often conflicting objectives and different constraints. The genetic algorithm was generally used in calibration, based on mechanical constraints with a multiobjective optimization function based on fuel consumption and engine-out emissions.

In 2006, a large number of stationary engine tests was utilized to build fuel consumption and engine-out emission models by artificial neural networks [19]. Artificial neural networks were combined with genetic algorithms for optimizing fuel consumption while taking emission constraints. A model-based calibration procedure for diesel engine calibration optimization was proposed by Nikzadfar and Shamekhi recently [37]. Full load and part load calibration problems were solved independently by utilization of separate cost functions within the GA. In 2018, Arya and Millo [38] proposed a multi-objective optimization method targeting less $NO_x$, soot and fuel consumption during the WLTC. The main objective was to find the best calibration from a group of already optimized calibrations according to their smoothness and effect on emissions.

By using genetic algorithms, they succeed to achieve similar smoothness levels compared to conventional methods. Moreover, 10%, 5%, 1% reduction in $NO_x$ emissions, soot emissions and fuel consumption were achieved respectively.

Another derivative-free optimization method, PSO, was applied to model-based boost pressure outer control-loop calibration by R. Sauermann et al. [39]. PSO succeed to find the global minimum after 50 iterations. The optimization parameters were selected as proportional and integral gains of the PI controller. The overshoot of 10% was given as a constraint in the cost function. Instead of genetic algorithm, M. Grahn et al. [36] used a grid search algorithm by utilization of only 3 controllable parameters (boost pressure, injection timing and lambda at intake manifold) with steady and transient lookup tables. According to the comparison of the results of transient EMS optimization with a steady-state approach, around 1% of fuel economy was gained during the simulations of the pre-defined cycles such as NEDC or FTP while considering the emission limits.

As for online learning of calibration methodology, in 2010, A. Malikopoulos et al. [28] aimed to convert the calibration dataset to an autonomous intelligent system that learns the optimal setpoints of engine controllers in real-time. In this theoretical method, engine operation was treated as a Markov decision process. Although the results of model-based approach were promising in terms of fuel economy, it might create several problems in real-life applications due to online learning. When the number of interdependent control parameters under learning process is increased, the higher dimensionality and learning time problem may occur. One year later, this problem was addressed with a decentralized learning control scheme [40]. In addition to self-learning of injection timing, variable geometry turbocharger vane position was also added to his learning scheme. This was achieved by the use of a learning process that can map the probability of the values of the control parameters which can change simultaneously. The main issue with this method is that meeting all the emission, fuel consumption and performance criteria for real-world driving scenarios requires more than two engine control parameters.

# 3. MATERIALS & METHODS

## 3.1. Approach

The target of this study is to create an automatic calibration methodology for fuel efficiency by using the genetic algorithms in combination with empirical models of ECU, ICE and EAS to have the focus on real driving emissions limits specifically for pre-defined routes. In the proposed method, in addition to extended mechanical constraints, the legal Euro 6d-Temp tailpipe $NO_x$ emission limits [41] were also taken as constraints while aiming to decrease the fuel consumption over the defined drive cycle.

The workflow of the proposed calibration methodology is demonstrated below in Figure 3.1. The process starts with engine operating points traveled over the target route. Instead of calibrating each point on each emission-related 2D look-up tables, the focus in this research was to identify only the most important points by the frequency distribution of the cycle points in terms of engine speed and torque setpoint. On these cycle points, the calibration values which are coming from engine speed and torque setpoint based 2D lookup tables, have the upper and lower limits to have smooth transitions over the cycle points. Then, engine-out emissions, fuel consumption and mechanical constraints are estimated by the ICE model which was developed in AVL Cameo with neural networks. AVL Cameo is a calibration software developed by AVL List GmbH. The main tasks done by this calibration tool are creating DoEs, automatic testing and measuring, raw data analysis, modeling of engine responses and cycle optimization [42]. Before the optimization starts, the tailpipe emissions are estimated by a neural network based EAS model. At the end of cycle simulation, all the mechanical constraints, cumulated emission mass flows and fuel quantity are fed to the optimizer to calculate the cost function. Finally, a genetic algorithm based optimizer proposes new set of calibration values just for the target engine operating points to decrease the cost which is based on fuel consumption mainly. The improvements in fuel efficiency were assessed by the simulation of the predefined cycle with each generated

calibration compared to the reference dataset while monitoring the violations in the limits of mechanical components and emissions.

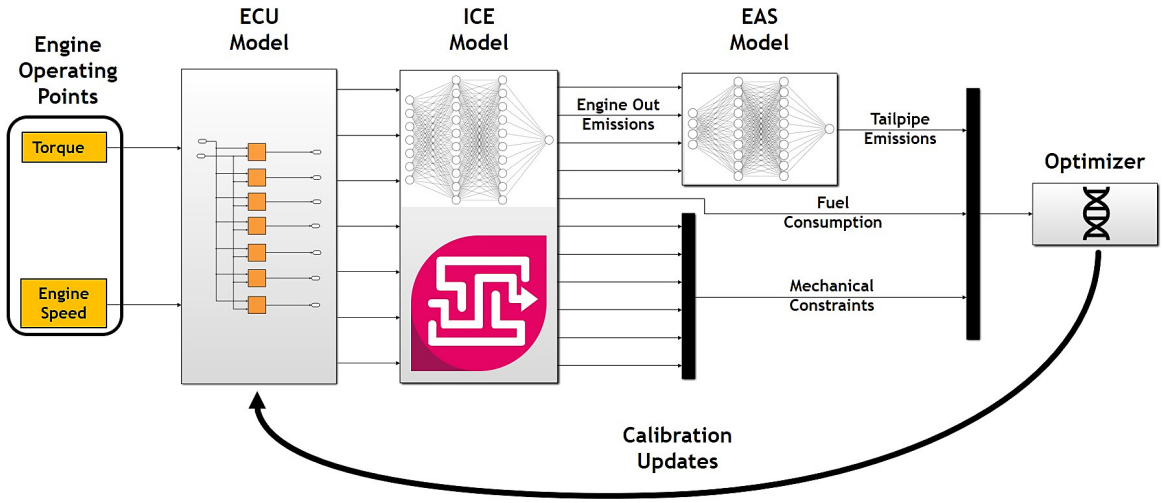



Figure 3.1: The scheme of the model-based calibration environment.

## 3.2. Modeling

To start with modeling, a 2 liter diesel engine calibration dataset was used as a reference for ECU parameters. Control parameter maps in the ECU were modeled as 2D look-up tables in Simulink. Then, by using DoE measurements, steady-state engine responses were modeled in AVL Cameo. In the end, transient RDE cycle data was utilized to generate neural network based SCR model. As for the main development environment, Matlab 2019a was used since it is easy to merge models of the ECU, ICE and SCR on Simulink. A combination of these models eases the model-based calibration activities. In addition to the models, cycle analyzer and genetic algorithm were implemented to the calibration environment.

### 3.2.1. ECU Model

As a first step, ECU steady-state maps are built as 2D lookup tables of main control parameters which have a significant effect on emission and fuel consumption. The lookup tables were filled with the values carried out from the reference calibration

dataset which was calibrated by conventional calibration methods. The following control parameters are tabulated based on engine speed and torque setpoint as shown in Figure 3.2:

(i) Start of the main injection angle,

(ii) Start of pilot injection angle,

(iii) Pilot injection quantity,

(iv) Main injection quantity,

(v) Mass airflow,

(vi) Boost pressure,

(vii) Rail pressure.

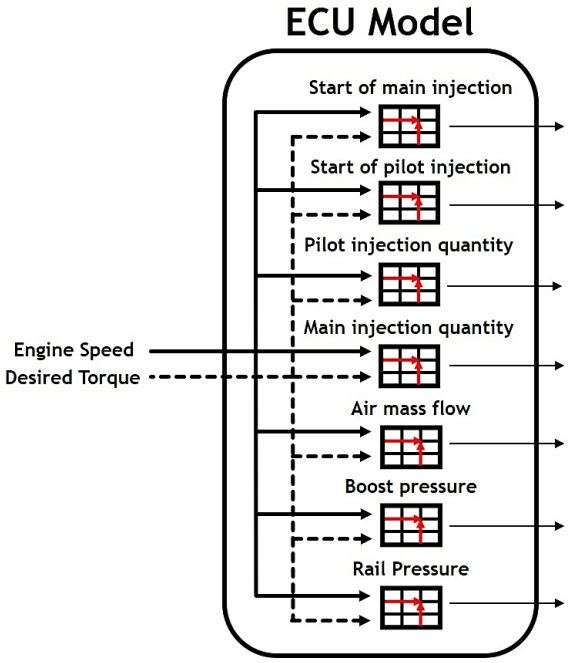



Figure 3.2: ECU model inputs and outputs.

As shown in Figure 3.2, engine speed and torque traces are interpolated in these 2D lookup tables to have corresponding output signals of injection quantity, injection angle, air mass flow, rail and boost pressure. These outputs are fed to the engine model directly to estimate the engine out temperature, pressure and emission traces. Since the EGR rate is controlled by the fresh air mass flow entering the cylinder, instead of direct EGR rate control, air mass flow map was used to control the EGR rate.

### 3.2.2. Engine Model

The output signals coming from the lookup tables are then fed to the engine model to simulate the engine responses such as engine out temperature, pressure or emissions. As opposed to the physics-based engine modeling, the empirical modeling is much faster and easy. Therefore, data-driven modeling was chosen as modeling approach. To generate empirical engine response models, AVL Cameo software was used. The reason to choose AVL Cameo is that it is time-efficient and easy to use in terms of preprocessing of data and application of different modeling options. A neuro-fuzzy modeling method called robust neural networks (RNN) was chosen as the modeling method. These neuro fuzzy models consist of several polynomials fitted to each local fuzzy area.

To model the required parameters for this study, steady-state dynamometer data was utilized. It was very crucial to set up a correct input-output correlation to have an accurate empirical model. Apart from engine speed, 7 engine control parameters were used as inputs for neural networks. These inputs are defined as starting angles of main and pilot injections, quantities of main and pilot injections, air mass flow, boost pressure and rail pressure. In order to have a comprehensive engine model, DoE data was used. The main reason to use the DoE method was to cover the behavior of as many operating points as possible with the least amount of test points. The DoE test points were created while considering mechanical constraints such as full load curve based fuel injection quantity limitation.

The workflow of modeling in AVL Cameo was demonstrated below in Figure 3.3.

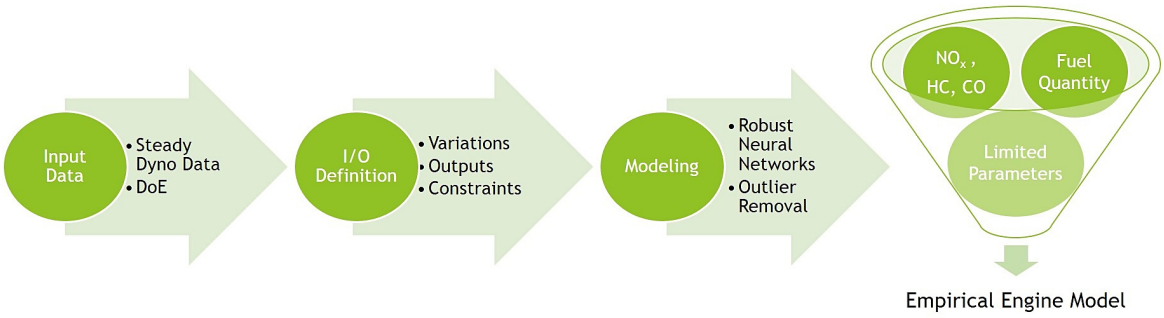



Figure 3.3: AVL Cameo modeling workflow.

After the measurements on the testbed were performed, automatically detected outliers were removed. Afterwards the neural network model was exported from AVL Cameo to Simulink for further simulations in the optimization loop. Moreover, the exported model is used to observe the effect of calibration parameters on emissions and fuel consumption. Remaining 1037 data points were used to train the neural networks. In Figure 3.4, the inputs and the outputs of the engine model were demonstrated.

In addition to engine speed and control parameters coming from lookup tables; engine coolant temperature, oil temperature and boosted air temperature could have been used as inputs to the engine model since they were also changed slightly during the DoE tests due to the temperature control problem on the engine testbed. However, these parameters are not controlled via ECU maps. Therefore, during the optimization, these temperatures were assumed as constants and did not have any effect on the engine model outputs. The constraint parameters are modeled as well by using the neural networks trained by the DoE data. Together with mechanical constraints like turbine speed, maximum cylinder pressure, compressor out or turbine inlet temperature; $NO_x$, PM and HC emissions were also modeled. The engine-out emission responses were fed to the EAS model to generate tailpipe emissions which are monitored during the optimization process.

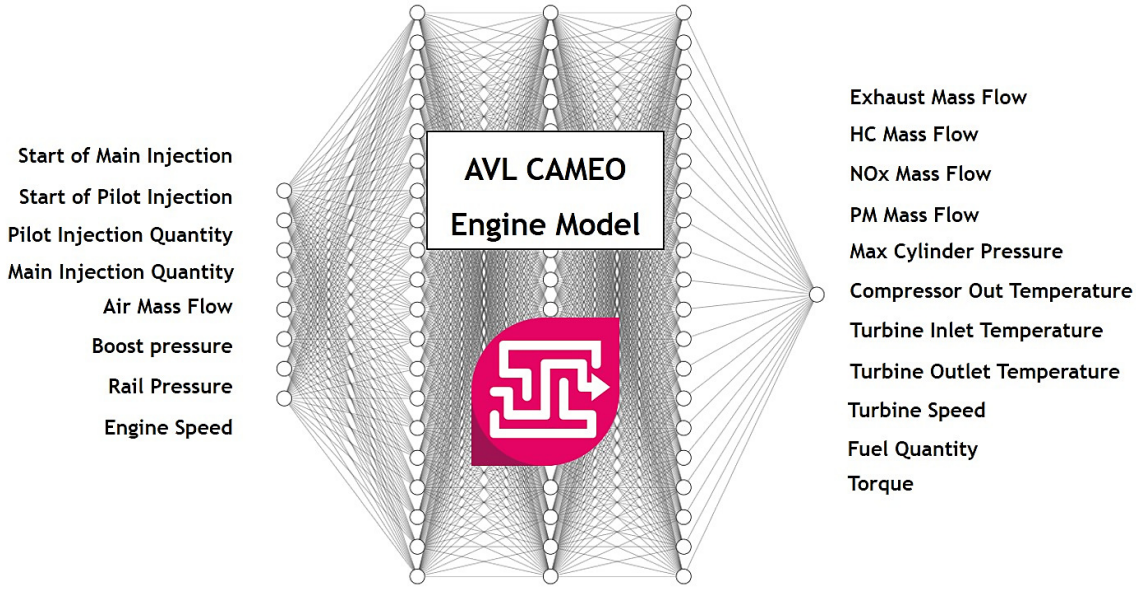



Figure 3.4: Engine model inputs and outputs.

While some of the mechanical constraints or objective parameters were modeled, some of them were controlled directly by the 2D look-up tables inside the ECU model. Rail pressure, boost pressure, air mass flow and injection quantity are the parameters that were monitored just before they entered the engine model.

### 3.2.3. EAS Model

As for the EAS modeling, SCR was chosen as a base EAS model. SCR was used to convert $NO_x$ emissions by $NH_3$ dosing at the upstream of SCR. Along the diesel engine's exhaust line, usually OEMs implement DPF and DOC systems at the upstream of the SCR to filter out the PM, HC and CO emissions before the tailpipe. However, in this research, the focus was just on the SCR system because PM, HC or CO at the tailpipe is not considered as a big challenge for the OEMs anymore due to the higher efficiency of DPF and DOC systems compared to SCR conversion efficiency. To estimate the HC and PM mass flows at the tailpipe during the simulations, the overall filtration and conversion efficiency for DPF and DOC systems were set to 95% and 90% according to [43] and [44] respectively.

The inputs for the SCR system were defined based on the $NO_x$ conversion efficiency related parameters such as exhaust mass flow, exhaust temperature, $NH_3$ dosing and SCR upstream $NO_x$ mass flow. However, $NH_3$ dosing was excluded from the model inputs. The reason is that $NH_3$ dosing is controlled by a closed-loop control algorithm by the dosing control unit (DCU) and the measurement data is not enough to build an accurate DCU model. As for the exhaust temperature at the SCR upstream, the measurement data was used directly as input since the SCR temperature was not modeled due to the lack of data.

To model the $NO_x$ conversion efficiency in the SCR, the measurement data of the predefined route was used. Since steady-state data cannot be representative for a highly dynamic chemical reaction such as $NO_x$ conversion, transient data was chosen to train neural networks to create a model applicable to transients. Before the training, the data were preprocessed such that the parts where the $NO_x$ tailpipe sensor did not reach the dew point were excluded not to train the model with improper data. After the initial data was preprocessed, the remaining 6362 data points were scattered in terms of engine operating points to understand how much of the engine operation area was covered in the cycle. It can be seen from Figure 3.5 that the example cycle could cover most of the operation range. More data will improve both the representativeness and the accuracy of the model.

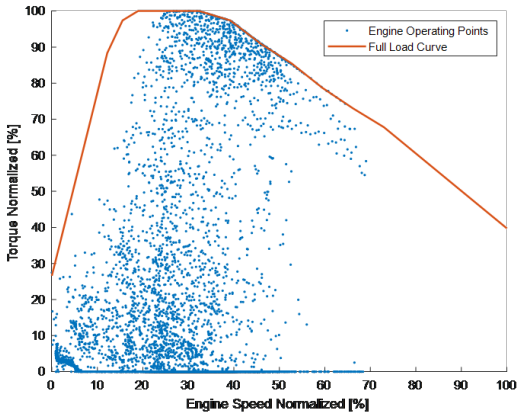



Figure 3.5: Engine operation coverage of SCR model input data.

Matlab's Neural Network Toolbox was used for modeling the SCR behavior. The SCR model scheme is shown below with all inputs and $NO_x$ mass flow at the downstream of SCR as the output.

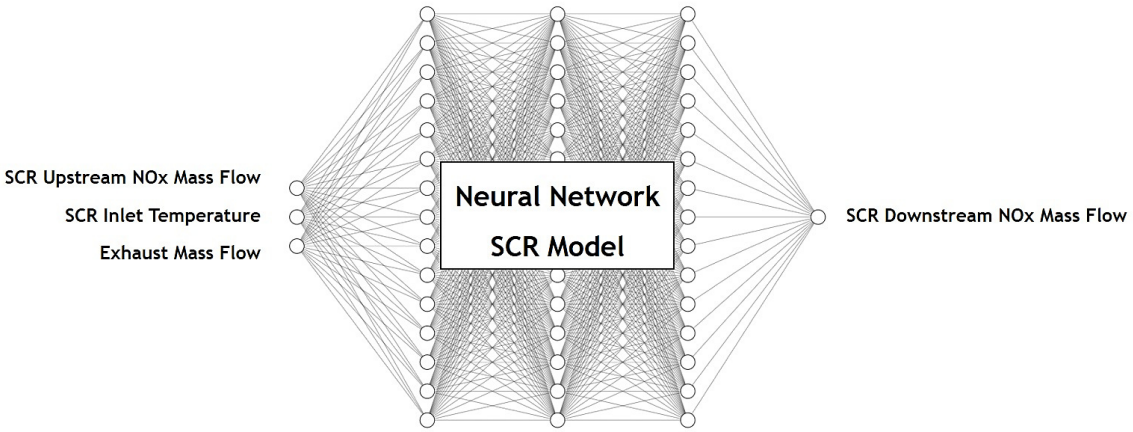



Figure 3.6: SCR model inputs and the output.

During the SCR efficiency modeling, there were 120 loops iterated to find the optimum neural network training options provided by the toolbox. The configuration parameters listed below were altered to find the best fit with the least computation time.

- **Number of hidden layers:** 1,2,3
- **Number of neurons in each layer:** 5,10,15,20
- **Training function:** Levenberg-Marquardt (LM), Bayesian Regularization (BR), BFGS Quasi-Newton (BFG), Resilient Backpropagation (RP), Scaled Conjugate Gradient (SCG)
- **Performance criteria:** Sum Squared Error (SSE), Mean Squared Error (MSE)
- **Activation functions:** Sigmoid in Hidden Layers, Linear in Output Layer

## 3.3. Optimization

### 3.3.1. Optimization Domain Definition

In this research, we focused on specific real-world driving cycles such as daily routes for commuting or repetitive routes for trucks driven in construction areas, to generate fuel-efficient but emission compliant calibration datasets. The road measurement used in this study is a valid RDE cycle satisfying the conditions given in Table 1.3. This RDE cycle consists of similar distance shares of urban (34%), rural (33%) and highway (33%) phases. The urban, rural and highway phases, vehicle speed and altitude profiles are given in Figure 3.7.

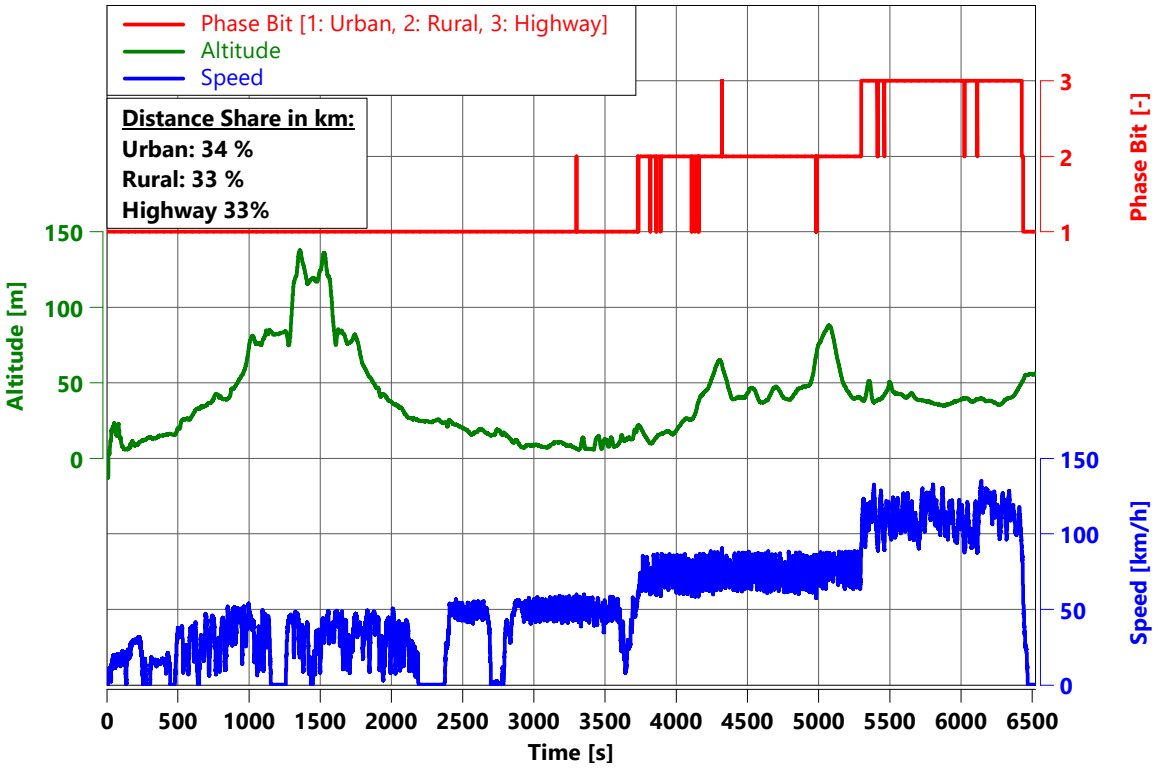



Figure 3.7: Vehicle speed and altitude profile of the reference cycle.

To calculate the engine operating point weights of a drive cycle, two different methods were used. In the first method, the percentage of time spent, in other words, frequency is calculated for each breakpoint on the 2D lookup tables whereas in the second method, the percentage of fuel consumption is calculated for each breakpoint.

Although the distributed data is different for each method, the distribution algorithm is same for both as shown in Figure 3.8. It starts with feeding up the cycle traces in terms of engine speed and torque. Then, to calculate the weights of each breakpoint, distances between the data point and the closest breakpoints are used. Finally, the data point is distributed to the breakpoints accordingly with each breakpoint weight. The data distribution was performed in 3 steps. In the following formulas, one data point ($Data_z$) of engine operating point ($Data_x$,$Data_y$), is distributed according to calculated weights ($W_{(x1,y1)}$,$W_{(x1,y2)}$,$W_{(x2,y1)}$,$W_{(x2,y2)}$) of each neighbour breakpoint ($Bp_{x_1}$, $Bp_{x_2}$, $Bp_{y_1}$, $Bp_{y_2}$).

(i) Definition of Neighbour Breakpoints: $Bp_{x1}$, $Bp_{x2}$, $Bp_{y_1}$, $Bp_{y_2}$

(ii) Calculation of Weights:

$$W_{x1} = 1 - \frac{(Data_x - Bp_{x1})}{(Bp_{x2} - Bp_{x1})} \tag{3.1}$$

$$W_{x2} = 1 - W_{x1} \tag{3.2}$$

$$W_{y_1} = 1 - \frac{(Data_y - Bp_{y_1})}{(Bp_{y_2} - Bp_{y_1})} \tag{3.3}$$

$$W_{y_2} = 1 - W_{y_1} \tag{3.4}$$

$$W_{(x1,y1)} = W_{x1} * W_{y_1} \tag{3.5}$$

$$W_{(x1,y2)} = W_{x1} * W_{y_2} \tag{3.6}$$

$$W_{(x2,y1)} = W_{x2} * W_{y_1} \tag{3.7}$$

$$W_{(x2,y2)} = W_{x2} * W_{y_2} \tag{3.8}$$

(iii) Data Distribution on Neighbour Breakpoints:

$$Data_{(x1,y1)} = W_{(x1,y1)} * Data_z \tag{3.9}$$

$$Data_{(x1,y2)} = W_{(x1,y2)} * Data_z \tag{3.10}$$

$$Data_{(x2,y1)} = W_{(x2,y1)} * Data_z \tag{3.11}$$

$$Data_{(x2,y2)} = W_{(x2,y2)} * Data_z \tag{3.12}$$

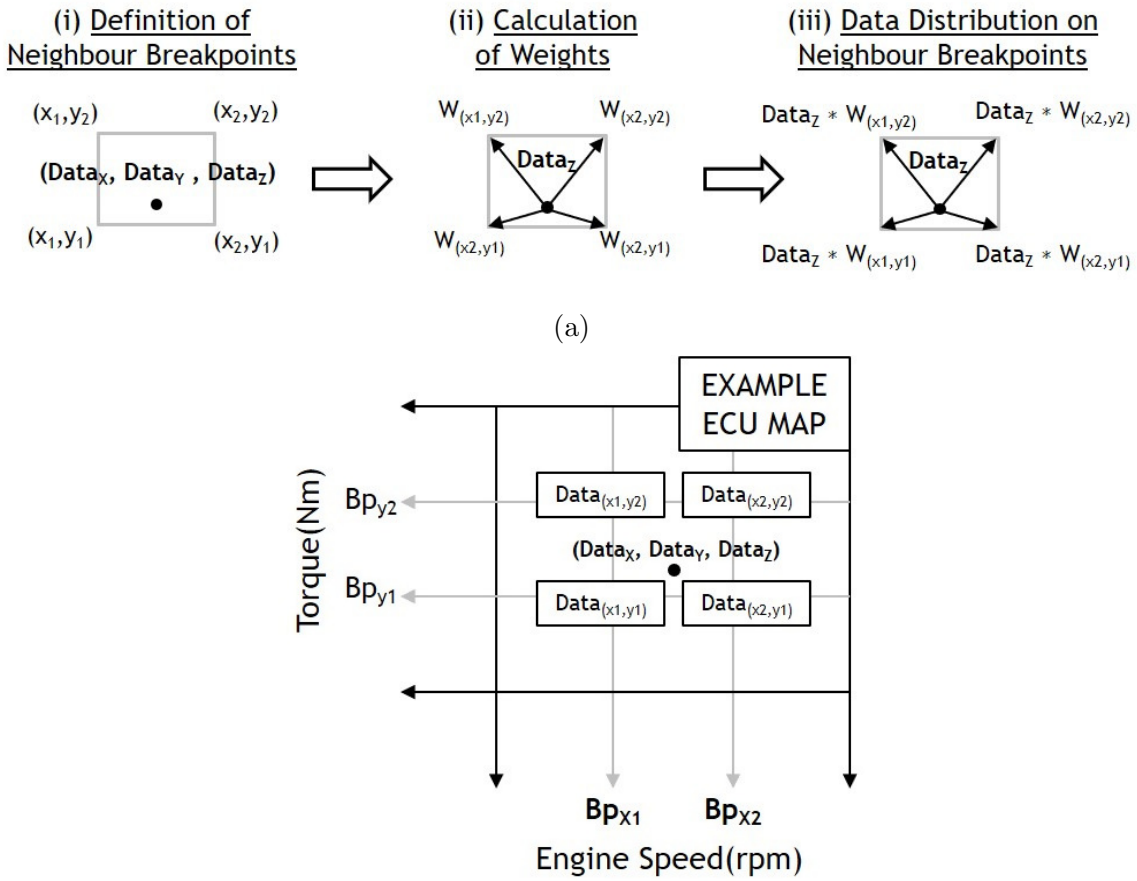



Figure 3.8: (a) Data distribution steps start with neighbour breakpoint definition and continues with weight calculation. At the end, the data is distributed to the breakpoints according to each corresponding weight. (b) The resulted data is represented on an example ECU map.

At the end of distribution, time spent or fuel consumption on each breakpoint are accumulated and then divided by the total time spent or total fuel consumption over the cycle to convert time or mg to percentage. The most weighted points on the lookup tables are chosen as the domain for optimization. In Figure 3.9, the reference cycle data was used to visualize the distribution of time spent on engine speed - torque map with normalized breakpoints which were scaled from 0 to 100 whereas in Figure 3.10, the distribution of fuel consumption percentage over the driven cycle is shown. As it can be seen from the Figure 3.9 and Figure 3.10, the most weighted engine operating points are colored darker. These points are selected as domain for the calibration improvement.


Figure 3.9: Distribution of time spent on a normalized engine speed - torque map. The color of a point is getting darker as the frequency increases.

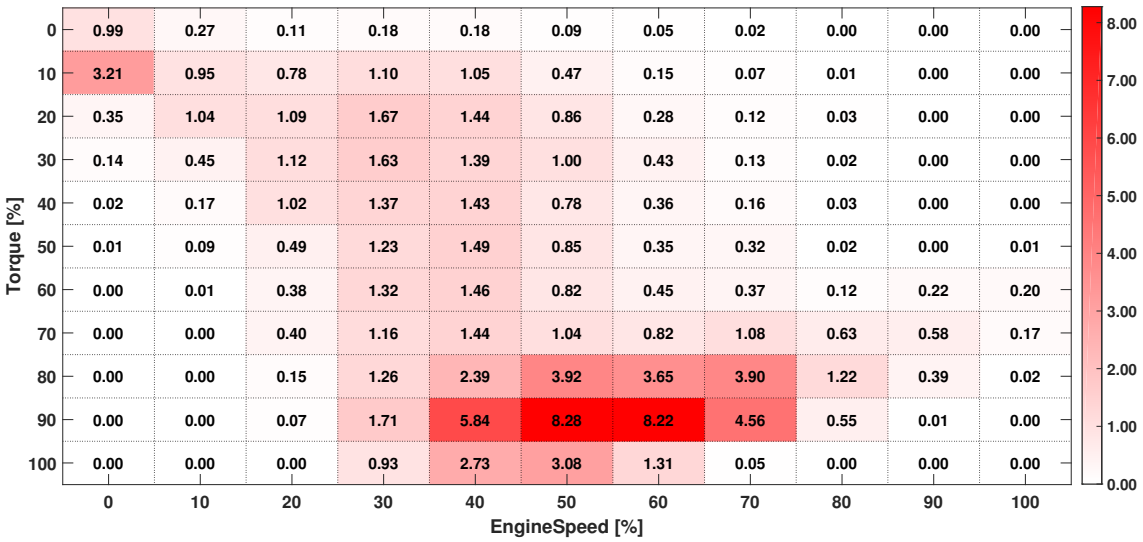



Figure 3.10: Distribution of fuel consumption on a normalized engine speed - torque map. The color of a point is getting darker as the fuel consumption increases.

### 3.3.2. Optimization Objective and Constraints

In the engine calibration optimization problems, there exist different constraints in terms of mechanical, performance and emissions. In Table 3.1, all the limitations and the main cost parameter defined in the optimization problem are listed.

Table 3.1: Optimization objective and constraints.

| **Mechanical Constraints** | **Emission Constraints** | **Performance Constraint** | **Main Objective** |
|---|---|---|---|
| Maximum Cylinder Pressure | $NO_x$ Mass Flow | Desired Torque Match | Fuel Injection Quantity |
| Rail Pressure | PM Mass Flow | | |
| Boost Pressure | HC + $NO_x$ Mass Flow | | |
| Turbine Inlet Temperature | | | |
| Compressor Out Temperature | | | |
| Turbine Speed | | | |

Mechanical constraints represent the physical limits of the engine components. It is crucial to be within these limits of the engine components to prevent the engine from damage. For instance, boost pressure of the air should not exceed a certain threshold which can result in over-boost and compressor surge at the end. Moreover, rail pressure along the common rail system must be below the endurance limit of the injection systems. It is also crucial to monitor whether maximum in-cylinder pressure is above the cylinder endurance limits or not during the combustion. As for the compressor out and turbine inlet temperatures, it is very important to stay in the safe operating range of intake and exhaust manifold temperatures. In case the turbine inlet temperature becomes higher than the temperature limit which turbine cannot stand for a long time, it will damage the turbine vanes. In case the compressor outlet

temperature increases above the thermal limit, the thermal stress on the components along the intake manifold will be increased and system will be damaged again. In addition to thermal stress, the increase in compressor out temperature is not desired because the air density is also decreasing as well as the amount of fresh air entering the cylinders. Another mechanical constraint that was taken into consideration during the optimization was turbine overspeed. If the turbine speed exceeds its mechanical limits for a long time, the turbine vanes will be damaged again. Therefore, all the limits mentioned above were taken from the hardware and OEM specifications and implemented as penalties into the objective function. Since both of the boost and rail pressure setpoints were controlled via ECU maps, they were limited from these maps instead of implementing them into the objective function.

As opposed to the studies in the literature, in which optimizations targeted the engine-out emissions, in this study, engine-out emissions were not even considered since it was not regulated by authorities. In addition, tail-pipe emissions were defined as constraints in the cost function. The Euro 6d-Temp regulation limits were taken as a reference during the optimization because the cycle that was used in the optimization was considered as a real-world driving cycle for a light commercial vehicle which is in the N1/CL3 vehicle category. Therefore, the limits at the end of the driven cycle were set as 125 mg/km, 4.5 mg/km and 215 mg/km for $NO_x$, PM and HC+$NO_x$ mass flows respectively. The emission constraints of $NO_x$, PM and HC were implemented to the objective function as penalty terms as well. In case the emission limits are violated with the new set of calibration parameters offered, it will be rejected during the optimization since the emission constraint was defined as a hard limiter.

Performance is another factor in calibration which is important for OEMs. Each OEM has its own pedal characteristic which is called brand DNA in engine calibration world. This characterized desired torque comes from the pedal maps defined in the ECU. In this research, the pedal maps are not changed not to destroy pedal characteristics. However, from torque to quantity maps are modified to lower the total fuel quantity injected. While decreasing the injection quantity for fuel efficiency; the start of injection angles for pilot and main, boost pressure and rail pressure setpoints were

calibrated to reach the same torque as it comes from the pedal maps. During the optimization process, the difference in the torque desired and the actual torque coming from the engine model was monitored. In case of high variation in the actual and desired torque, the objection function was penalized again. In case the optimization domain is selected within the full load range, the penalization gets tighter to protect the full load characteristics.

The smoothness of a calibration map is very crucial to have robust setpoint control in ECU especially during the transients. Therefore, smoothness was maintained by limiting the optimization parameters. Two different limitation method was applied to the optimization as listed below:

- **Soft limitation:** The optimization parameter can be increased or decreased as much as the maximum difference between consecutive values along the same x-axis breakpoint.
- **Hard limitation:** The optimization parameter cannot be higher or lower than the values corresponding to upper or lower y-axis breakpoints. The upper or lower limitation was selected according to monotonically increasing or decreasing behavior of the curve along the x-axis.

### 3.3.3. Optimization Cost Function

After the domain on the calibration maps was found, the optimization of these points was performed according to the mechanical or performance-related constraints, emission limitations and the main cost parameter. The main objective of the optimization was defined as the fuel economy on a defined route. While maintaining the best fuel economy over a route, it was decided not to violate any of the mechanical, emission or performance-related limits. For this reason, the cost-function was designed as including all the constraints as penalization terms added to the main objective term as shown below.

$$
\begin{aligned}
J = {} & k_{FUEL}\left(\frac{FUEL_{sim}}{FUEL_{act}}\right)^2 \\
& + k_{MCP}(1 - sign(MCP_{lim} - MCP_{sim})) \\
& + k_{RP}(1 - sign(RP_{lim} - RP_{sim})) \\
& + k_{BP}(1 - sign(BP_{lim} - BP_{sim})) \\
& + k_{TIT}(1 - sign(TIT_{lim} - TIT_{sim})) \\
& + k_{COT}(1 - sign(COT_{lim} - COT_{sim})) \\
& + k_{NE}(1 - sign(NE_{lim} - NE_{sim})) \\
& + k_{PE}(1 - sign(PE_{lim} - PE_{sim})) \\
& + k_{NHE}(1 - sign(NHE_{lim} - NHE_{sim})) \\
& + k_{TM}(1 - sign(TM_{lim} - TM_{sim}))
\end{aligned}
\tag{3.13}
$$

The weighting factor terms ($k_x$'s) in the cost function are all set to 1 since we assume all these terms as hard limiters. The penalty terms are designed such that when a limit is violated the sign term in the equation results in -1 so that the whole term increases the cost by 2.

### 3.3.4. Optimization Method

To optimize the total injected fuel quantity over a predefined route, global optimization methods were thought since the problem was constrained and nonlinear. Matlab's Global Optimization Toolbox provides a wide range of optimization methods such as pattern search, particle swarm and genetic algorithm to solve nonlinear problems. In this research, we focused on the genetic algorithm solver since it is a derivative-free, stochastic solver. Although, more function evaluations are required for reaching the global minimum, by use of GA compared to other solvers, the GA solver has an easy syntax and provides relatively wider range of optimization options. Therefore, one of the aims of this study was to find the most efficient genetic algorithm

optimization configuration. The simulation process consisting of all models and the implementation of the optimization step is demonstrated in Figure 3.11.

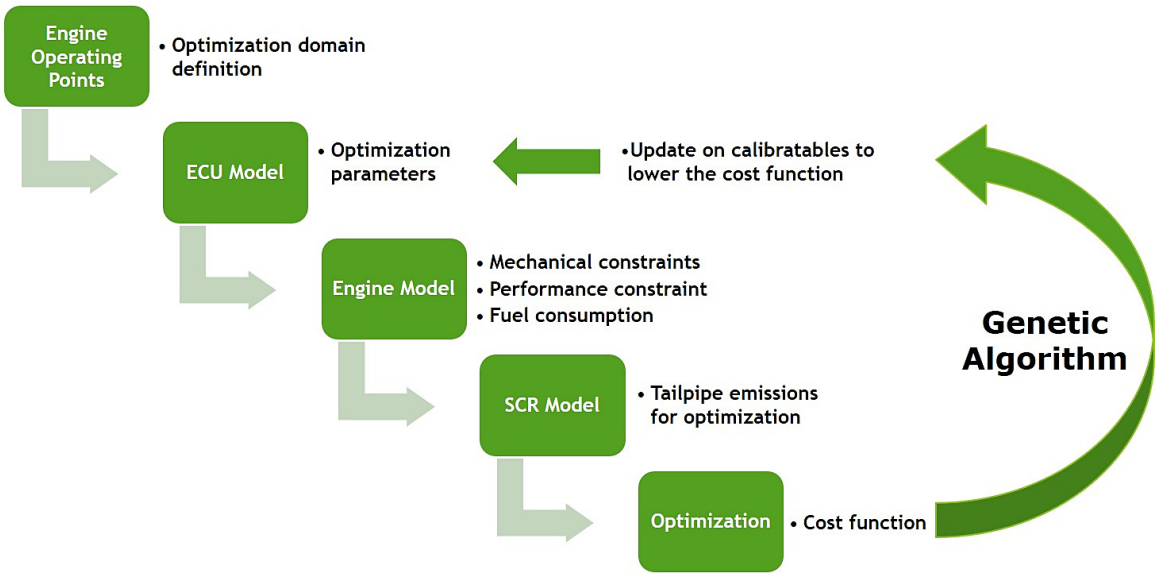
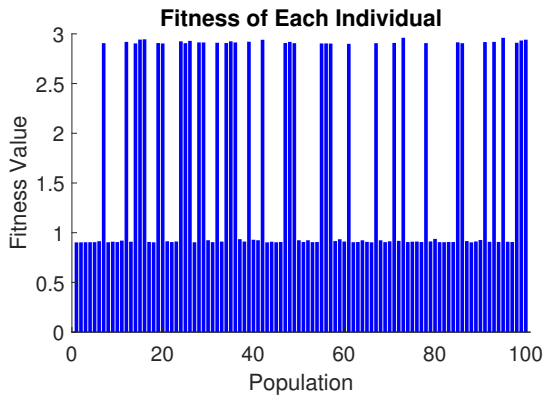



Figure 3.11: Optimization work flow.

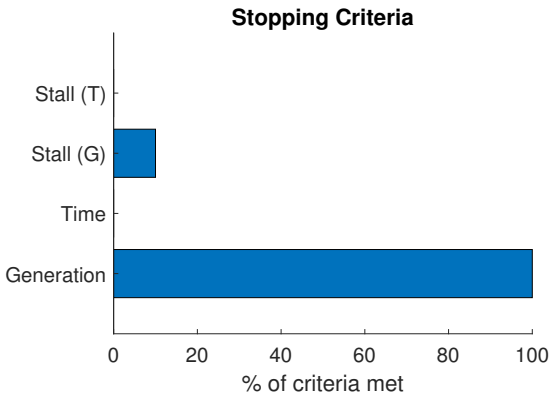
While using GA as solver, the optimization options, that have big impact on cost function, were iterated to find the best configuration for the solver in terms of cost and computation time. The following optimization options were iterated:

- **Domain definition method:** Frequency Based, Fuel Consumption Based
- **Limitation method:** Soft Limitation, Hard Limitation
- **Number of ECU maps optimized:** 7
- **Number of optimization parameters from each map:** 4, 9
- **Number of generations:** 50, 100, 150
- **Population size:** 10,20
- **Mutation function:** Uniform, Gaussian, Adaptive Feasible
- **Crossover function:** Scattered, Heuristic, Single Point, Two-Point, Arithmetic
- **Creation function:** Uniform, Linear Feasible

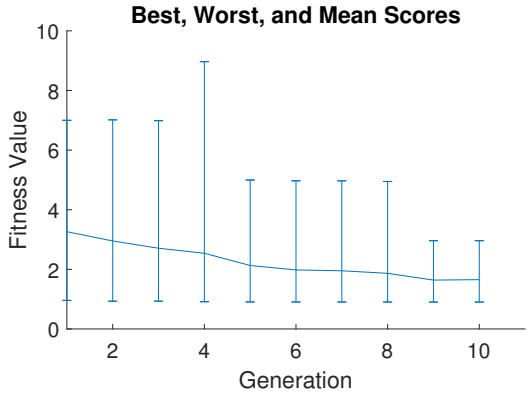
During the optimization, the progress was monitored in terms of stopping criteria, fitness of each individual, best, mean and worst scores of each generation as shown in Figure 3.12.




(a)




(b)




(c)

Figure 3.12: Optimization progress in terms of (a) Fitness of each individual, (b) Stopping criteria distribution and (c) Best, mean and worst fitness of each generation.

# 4. RESULTS

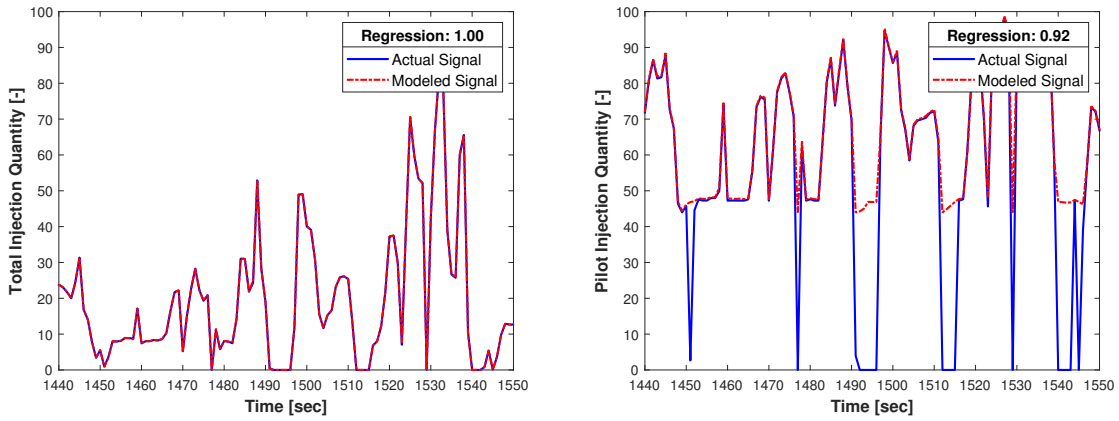
## 4.1. Model Validation

### 4.1.1. ECU Model Accuracy

The ECU model was validated with the pre-defined transient cycle. All the modeled parameters were normalized into [0,100] scale due to confidentiality of the data used in this research. The accuracies of the modeled parameters were evaluated by regression values of the model and actual values. The regression value (R) for a model with n samples, is calculated based on the Pearson's correlation formula as following:

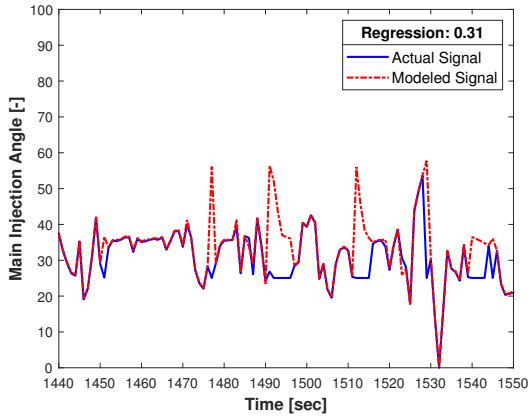
$$R = \frac{\sum_{i=1}^{n}((x_{model,i} - \bar{x}_{model,i})(x_{actual,i} - \bar{x}_{actual,i}))}{\sqrt{(\sum_{i=1}^{n}(x_{model,i} - \bar{x}_{model,i})^2)(\sum_{i=1}^{n}(x_{actual,i} - \bar{x}_{actual,i})^2)}} \tag{4.1}$$

At least 0.90 regression value was desired for each modeled parameter. As it can be seen in Figures 4.1 and 4.2, all the modeled parameters, except the main and pilot injection angles, have good accuracy.

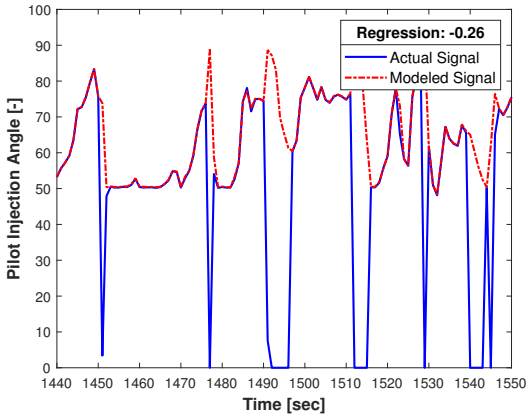



(a) (b)

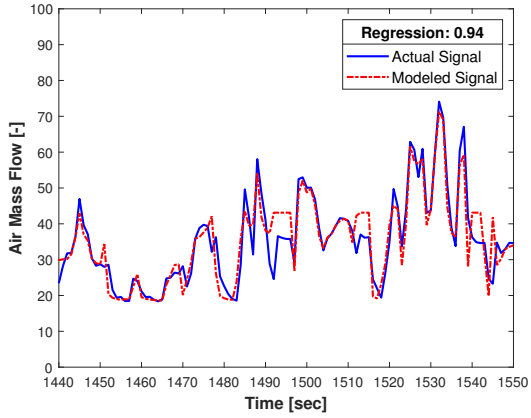
Figure 4.1: Model accuracies of (a) Total injection quantity and (b) Pilot injection quantity, during the pre-defined transient cycle are illustrated.



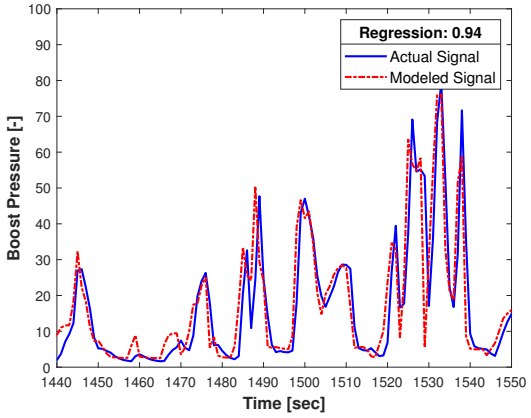

(a)



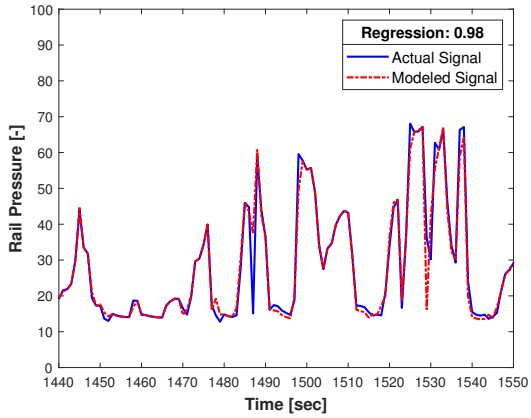

(b)



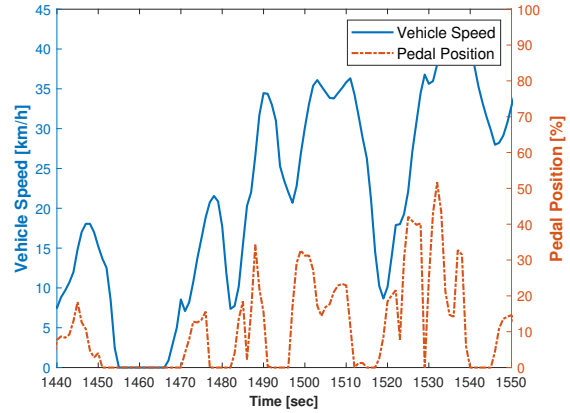

(c)




(d)




(e)




(f)

Figure 4.2: Model accuracies of (a) Main injection angle, (b) Pilot injection angle, (c) Air mass flow, (d) Boost pressure, and (e) Rail pressure model accuracies during the pre-defined transient cycle are illustrated as well as (f) Vehicle speed and pedal position.

Although the regression value of pilot injection quantity model is higher than 0.90, it has the same behavior as injection angles at the same instances where their models differentiate from actual values. Moreover, the boost pressure model value is following the actual signal with a small delay, especially during the high transients. This behavior was expected due to actuator delays. This small delay was neglected since the model accuracy of the boost pressure was 0.94 and considered as enough for this methodology development.

After the low accuracy points were investigated, the root cause was identified as fuel cut-off regions during the drive. These regions occurred during when there was no load (pedal position = 0) but the engine was not at idle speed. Therefore, the pilot injection quantity and the main and pilot injection angles coming from 2D look-up tables were corrected by a fuel cut-off detection model implemented to the ECU model as illustrated in Figure 4.3.

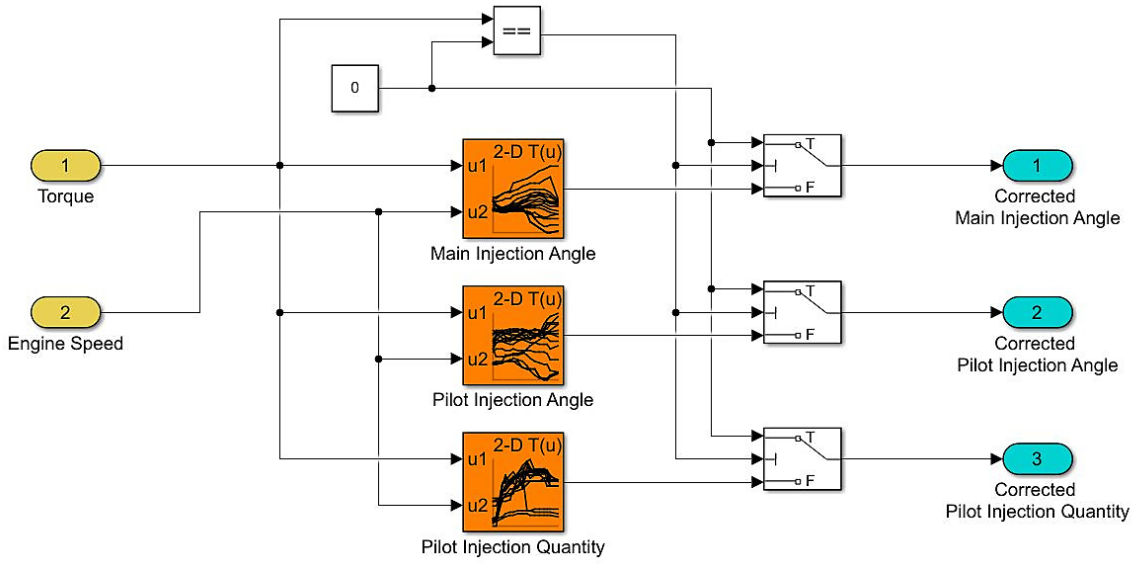



Figure 4.3: Fuel cut-off model.

After implementing the fuel cut-off model, the accuracies of the pilot injection quantity, main and pilot injection angles were increased to 0.98, 0.92 and 0.97 respectively. As shown in Figure 4.4, all the modeled parameters have good correlation with actual values over the pre-defined cycle.

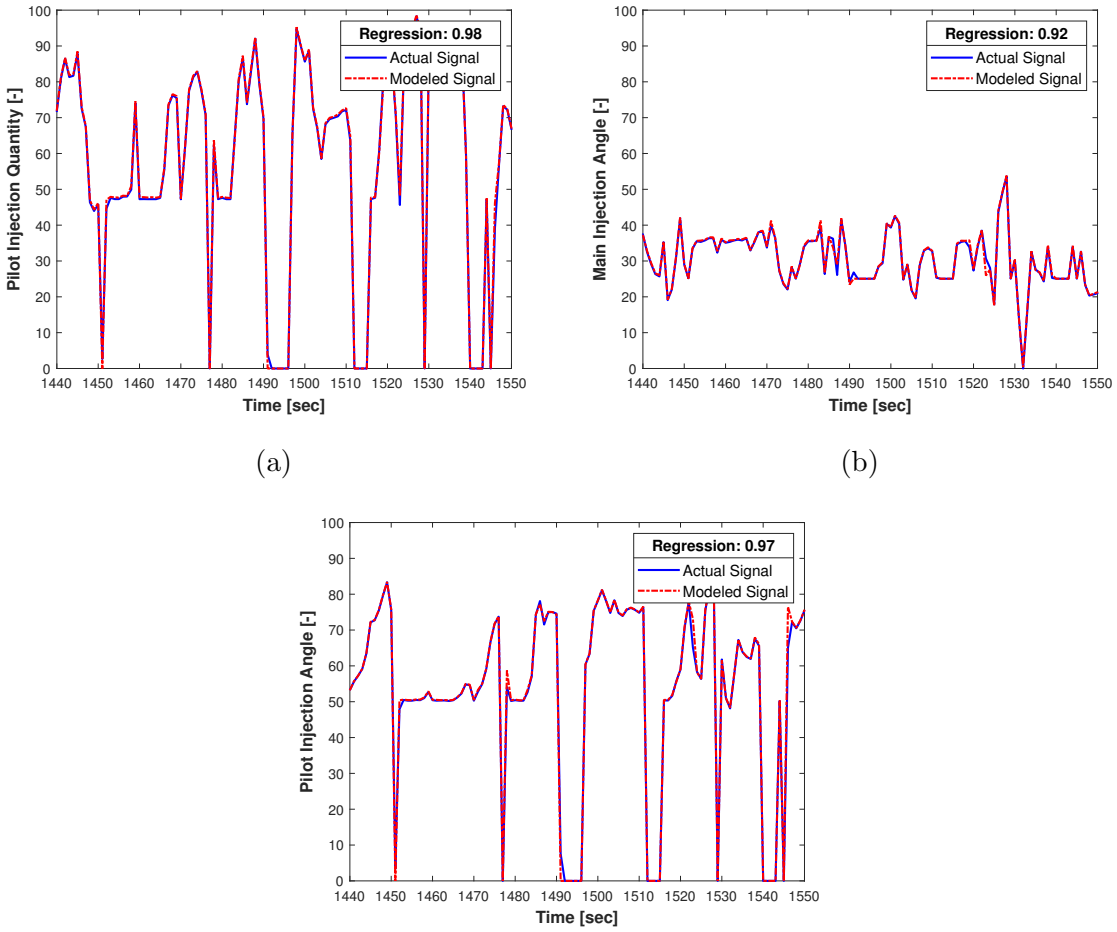



Figure 4.4: After fuel cut-off model implementation, improved model accuracies of (a) Pilot injection quantity, (b) Main injection angle and (c) Pilot injection angle are illustrated.

### 4.1.2. Engine Model Accuracy

The accuracy of the models were illustrated in Figures 4.5 and 4.6. The accuracy plots consist of actual values distributed along the x-axis, model values distributed along the y-axis Also, upper and lower thresholds for 10% error were demonstrated with lines.

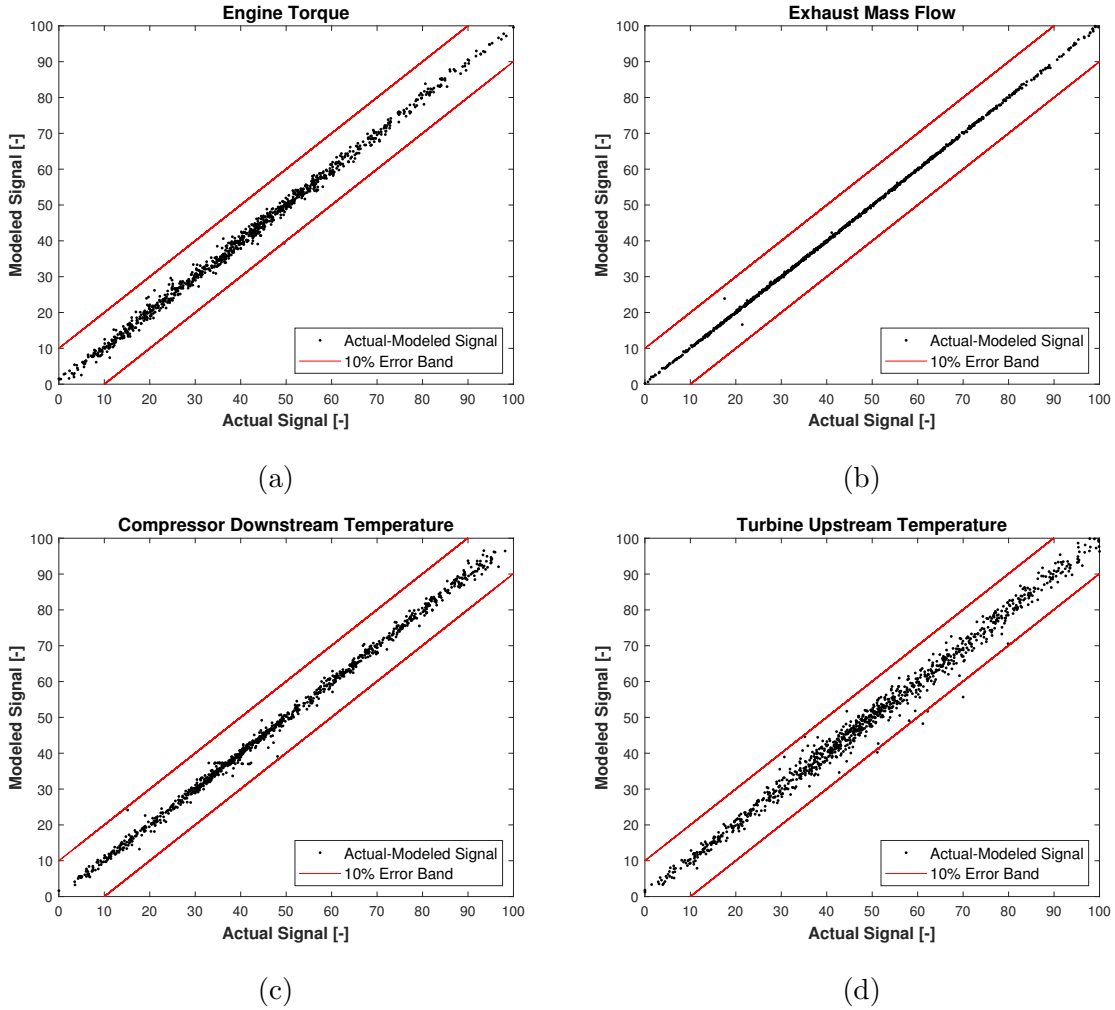



Figure 4.5: The model responses of (a) Compressor outlet temperature, (b) Turbine inlet temperature, (c) Compressor outlet temperature and (d) Turbine inlet temperature are compared with actual values.

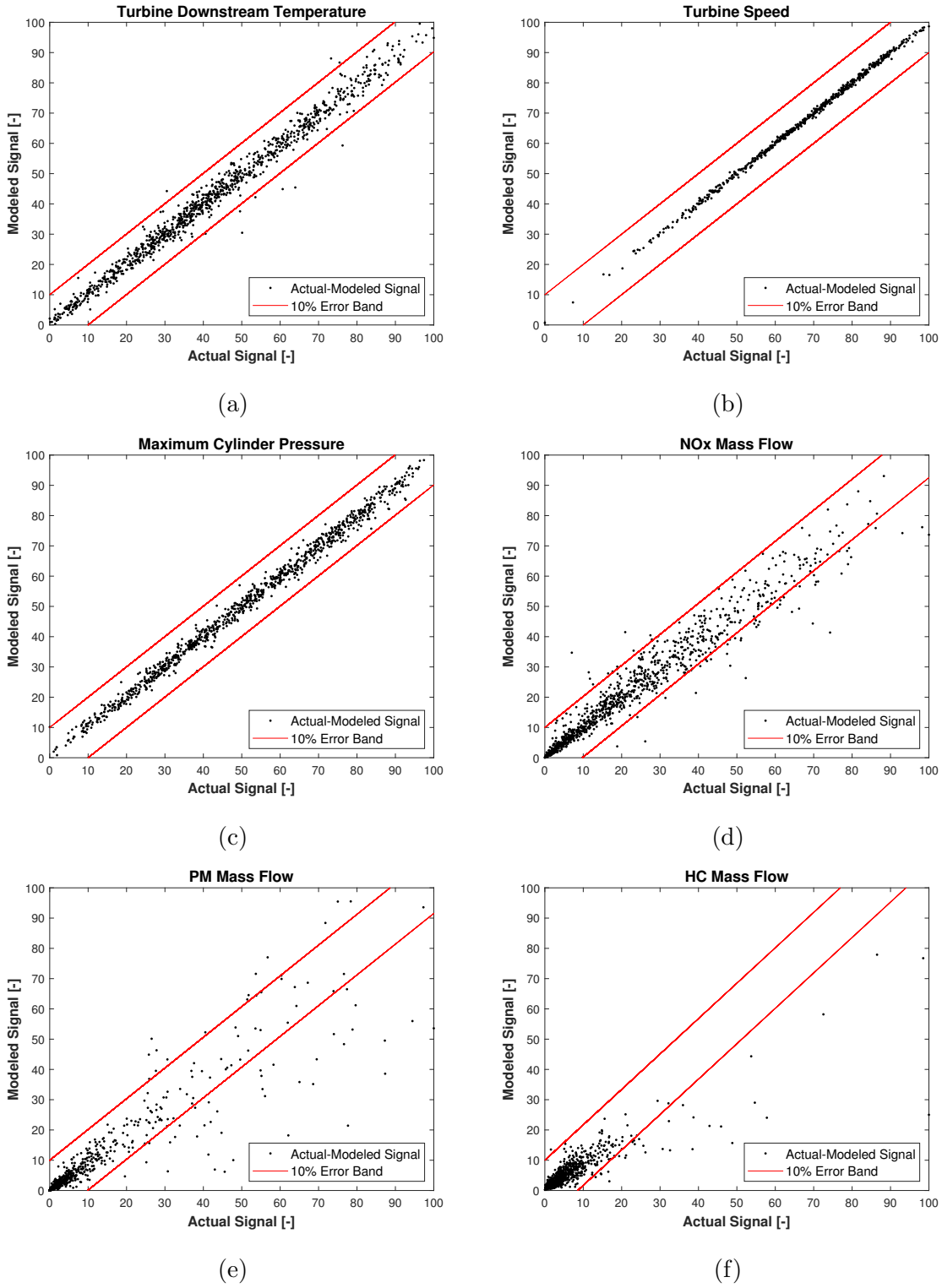



Figure 4.6: The model responses of (a) Turbine outlet temperature, (b) Turbine speed, (c) Maximum cylinder pressure, (d) $NO_x$ mass flow (e) PM mass flow and (f) HC mass flow are compared with actual values.

Except for the emission mass flow models, the error between the engine model responses and the actual values were generally within 10%. Although there exist some outliers in the model, the model quality was considered as adequate based on the model performance parameters represented in Table 4.1. For the evaluation of the model performance, the following parameters were calculated:

- **The Slope of Best Fit**

  According to the line equation given in Equation 4.2, when the slope of the best fit (a) equals 1, it means that the model is 100% accurate. Thus, it is expected to have the slope of the best fit as close as possible to 1.

$$y = ax + b \tag{4.2}$$

- **Root Mean Square Error(RMSE)**

  The Root Mean Square Error (RMSE) is defined as shown in the below formula:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_{actual,i} - x_{model,i})^2}{n}} \tag{4.3}$$

  While evaluating the model accuracy, RMSE was desired to be close to 0.

- **Normalized Root Mean Square Error(NRMSE)**

  To compare the RMSE of the parameters with different units, non-dimensional forms of the RMSE can be used. In the literature, there are two methods to normalize the RMSE:

  (i) Normalization based on the range of the actual values:

$$NMRSE = \frac{RMSE}{(x_{actual,max} - x_{actual,min})} \tag{4.4}$$

(ii) Normalization based on the mean of the actual values:

$$NMRSE = \frac{RMSE}{\bar{x}_{actual}} \tag{4.5}$$

In this study, since the parameters are already normalized according to the minimum and the maximum values due to confidentiality, the normalization method based on the average was used to have a different performance measure than RMSE. While evaluating the model accuracy, NMRSE value for each model was desired to be close to 0.

- **R-squared Error($R^2$)**

  R-squared is defined as shown in the below formula:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(x_{model,i} - x_{actual,i})^2}{\sum_{i=1}^{n}(x_{model,i} - \bar{x}_{actual,i})^2} \tag{4.6}$$

  A perfect match of the model and the actual values can be defined as $R^2$ equals to 1. Therefore, during the model evaluations, it was desired to have $R^2$ values greater than 0.900.

In Table 4.1, the modeled parameters were evaluated based on the defined performance parameters. All the models, except HC and PM emissions, have $R^2$ values greater than 0.900.

Table 4.1: Engine model accuracy.

| **Modeled Parameters** | **Slope of Best Fit** | **RMSE** | **NRMSE** | **$R^2$** |
|---|---|---|---|---|
| Engine Torque | 1 | 1.235 | 0.030 | 0.997 |
| Exhaust Mass Flow | 1 | 0.330 | 0.008 | 1.000 |
| Compressor Outlet Temperature | 1 | 1.123 | 0.023 | 0.998 |
| Turbine Inlet Temperature | 1 | 2.108 | 0.043 | 0.991 |
| Turbine Outlet Temperature | 1 | 2.774 | 0.065 | 0.985 |
| Turbine Speed | 1 | 0.653 | 0.010 | 0.999 |
| Maximum Cylinder Pressure | 1 | 1.914 | 0.038 | 0.993 |
| $NO_x$ Mass Flow | 1.02 | 4.325 | 0.195 | 0.949 |
| PM Mass Flow | 1.03 | 6.127 | 0.763 | 0.835 |
| HC Mass Flow | 1.17 | 2.743 | 0.526 | 0.785 |

### 4.1.3. EAS Model Accuracy

While modeling the SCR, 120 loops were iterated to find the optimum neural network training options. Number of layers, number of neurons in each layer, training method and performance criteria were the parameters that were altered. To train the model, the target was to find the least cumulated error at the end of the cycle and the highest linear regression between the predicted and the actual SCR downstream $NO_x$ mass flow values in training, test and validation datasets. Therefore, the measurement data was divided randomly into 3 parts. The portion of training, test and validation data were 70%, 15% and 15% respectively.

Although the training time for each training was calculated, it was not considered as the main performance criteria while choosing the best training option because the size of the training data was so small that it could not end up with high computation times. For future studies with higher amounts of training data, it should be considered as one of the main performance criteria.

In Table 4.2, the first 20 optimum training options out of 120 were listed according to overall regression values of each model whereas the result of the each iteration was given in Table 4.3. While taking cumulated error % and training time into account, $3^{rd}$ iteration was selected to be used in this model-based calibration process. In this optimum iteration, there are 15 neurons in each 3 hidden layer, trained with Levenberg-Marquardt algorithm. The other reason to choose $3^{rd}$ option as optimum was that it created a safety margin thanks to its 3.05% cumulated $NO_x$ overestimation. Moreover, the time spent on training was almost the half of the time required for training of the $1^{st}$ option.

Table 4.2: SCR model training options.

| **No.** | **Fitness Function** | **Training Function** | **Neuron Number** | **Hidden Layer Number** |
|---|---|---|---|---|
| 1 | SSE | LM | 20 | 3 |
| 2 | SSE | LM | 20 | 2 |
| 3 | MSE | LM | 15 | 3 |
| 4 | MSE | LM | 20 | 3 |
| 5 | MSE | BFG | 20 | 2 |
| 6 | SSE | RP | 20 | 3 |
| 7 | MSE | RP | 20 | 3 |
| 8 | MSE | LM | 20 | 2 |
| 9 | SSE | BR | 15 | 3 |
| 10 | MSE | BR | 15 | 3 |
| 11 | MSE | LM | 10 | 2 |
| 12 | MSE | LM | 15 | 2 |
| 13 | SSE | BR | 20 | 2 |
| 14 | MSE | BR | 20 | 2 |
| 15 | SSE | BR | 15 | 2 |
| 16 | MSE | BR | 15 | 2 |
| 17 | SSE | BR | 20 | 3 |
| 18 | MSE | BR | 20 | 3 |
| 19 | SSE | SCG | 20 | 2 |
| 20 | MSE | BFG | 10 | 2 |

Table 4.3: SCR model training results.

| **No.** [-] | **Cumulated Error** [%] | **Training Regression** [-] | **Validation Regression** [-] | **Test Regression** [-] | **Time Spent** [sec] |
|---|---|---|---|---|---|
| 1 | 3.74 | 0.88 | 0.84 | 0.86 | 18.90 |
| 2 | 6.86 | 0.86 | 0.85 | 0.87 | 7.16 |
| 3 | 3.05 | 0.88 | 0.79 | 0.81 | 9.43 |
| 4 | 0.86 | 0.86 | 0.83 | 0.84 | 18.37 |
| 5 | 3.50 | 0.85 | 0.85 | 0.87 | 10.71 |
| 6 | 3.09 | 0.86 | 0.83 | 0.85 | 2.17 |
| 7 | 3.09 | 0.86 | 0.83 | 0.85 | 2.08 |
| 8 | 1.85 | 0.86 | 0.83 | 0.87 | 7.70 |
| 9 | 1.16 | 0.86 | 0.83 | 0.83 | 15.51 |
| 10 | 1.16 | 0.86 | 0.83 | 0.83 | 18.11 |
| 11 | 10.30 | 0.85 | 0.84 | 0.83 | 2.58 |
| 12 | 18.02 | 0.85 | 0.83 | 0.86 | 4.45 |
| 13 | 2.38 | 0.84 | 0.85 | 0.87 | 19.95 |
| 14 | 2.38 | 0.84 | 0.85 | 0.87 | 19.64 |
| 15 | 0.36 | 0.85 | 0.83 | 0.87 | 10.63 |
| 16 | 0.36 | 0.85 | 0.83 | 0.87 | 10.83 |
| 17 | 3.62 | 0.85 | 0.83 | 0.86 | 41.88 |
| 18 | 3.62 | 0.85 | 0.83 | 0.86 | 41.64 |
| 19 | 0.77 | 0.84 | 0.85 | 0.86 | 3.84 |
| 20 | 2.53 | 0.85 | 0.85 | 0.83 | 5.18 |

The regression plots shown in Figure 4.7 represents how well the option 3 model fits with actual signals. Each of the separated training, validation and testing data has 0.88, 0.79 and 0.81 regression values respectively. The overall regression was calculated as 0.86.

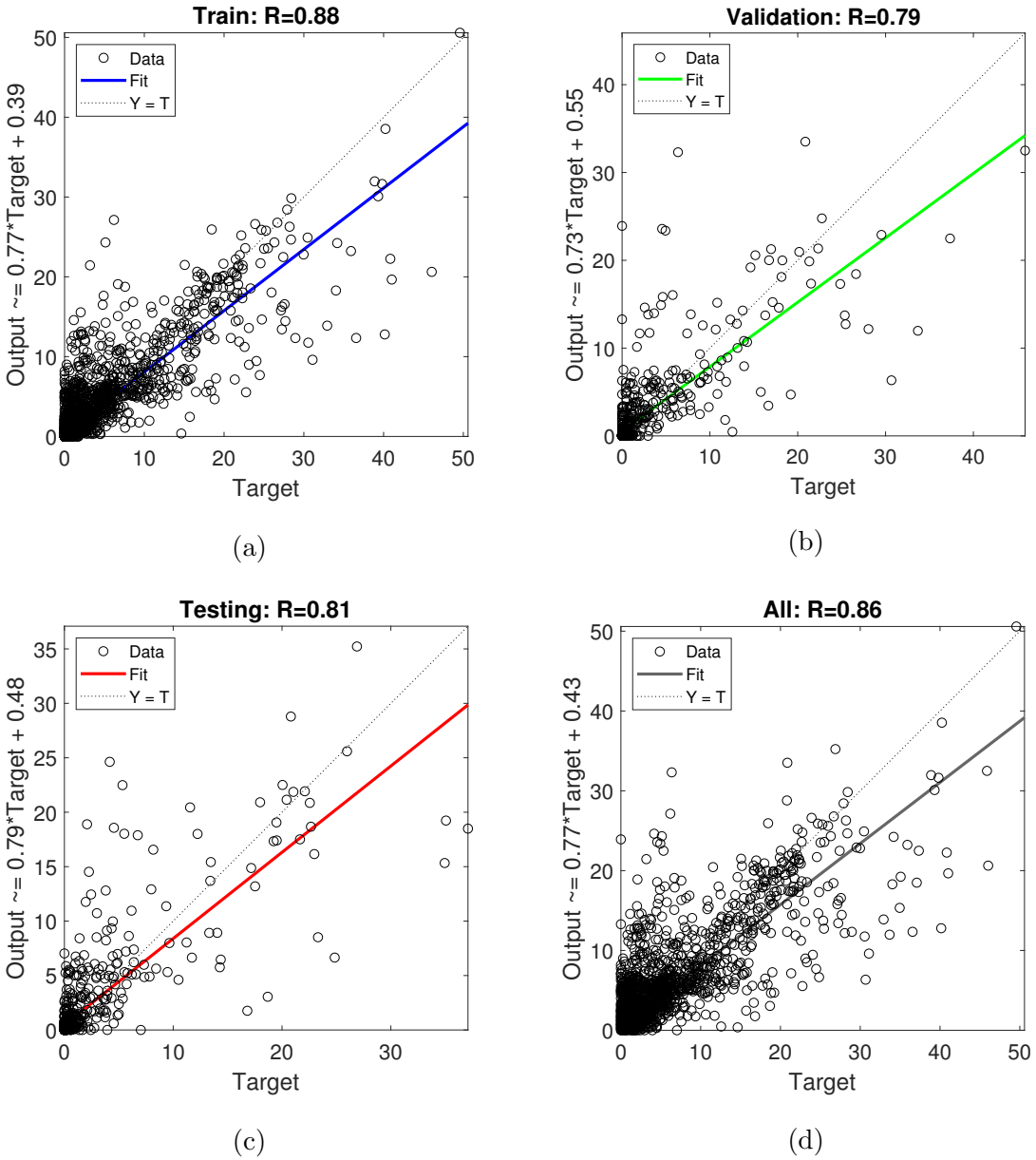
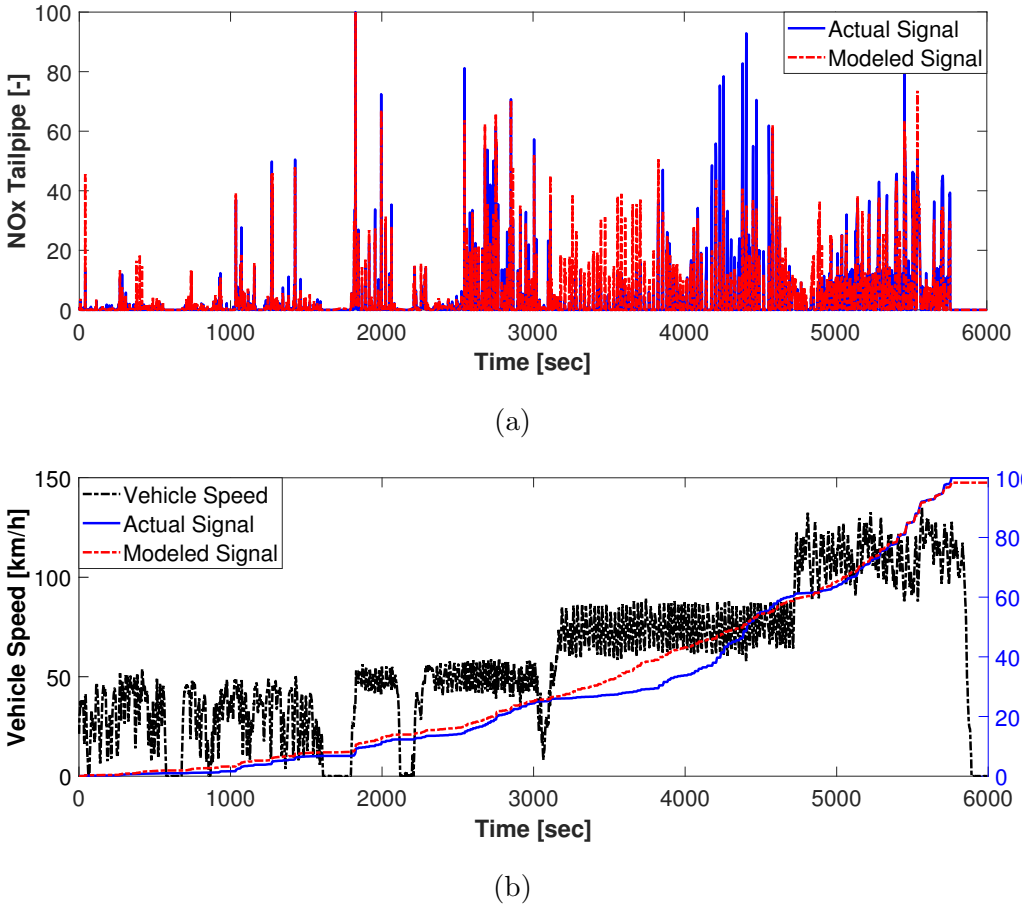



Figure 4.7: SCR model responses of (a) Training data, (b) Validation data, (c) Testing data and (d) All data are represented with linear regression equations shown on the y axis of each corresponding plot.

In Figure 4.8, the transient performance of the model in the pre-defined dynamic route was shown by $NO_x$ tailpipe emissions in terms of time-based and integrated normalized values due to the confidentiality of the data. Although, the model cannot follow the actual values precisely, especially in between $3250^{th}$ and $4500^{th}$ seconds, the model was considered as enough for this study since the regression value is higher than the desired (0.8). This model behavior was expected due to the extremely transient maneuvers and lack of $NH_3$ dosing as an input to the model.




(a)

(b)

Figure 4.8: According to (a) Time-based normalized $NO_x$ values and (b) Integrated normalized $NO_x$ tailpipe values and the vehicle speed trace, the model accuracy becomes worse as the transients increase.

## 4.2. Optimization Results

To find the optimum set of genetic algorithm options in Matlab, the configuration parameters were iterated in 35 optimization loops such as described in the MATERIALS and METHODS chapter. In this section, the results were evaluated firstly based on the lowest cost and then based on the time required for optimization. In addition, emissions, torque match and smoothness of the maps were also considered to have a representative calibration map for real life. Moreover, the constraints were all satisfied in all optimization trials. This can be understood from the best fitness values, shown in Table 4.5. All listed best fitness values are lower than 1, which means no penalization came from constraints.

The optimization results were investigated by comparing the similar iterations' results which differentiate from each other by just one setting. Therefore, to compare the effect of each configuration, 20 different iteration options and their results were given in Table 4.4 and Table 4.5 respectively.

Table 4.4: Optimization options.

| No. | Creation Function | Crossover Function | Mutation Function | Population Size | Maximum Generations | Limitation Method | Domain Definition | Domain Size |
|---|---|---|---|---|---|---|---|---|
| 1 | Uniform | Scattered | Adaptive Feasible | 50 | 10 | Soft Limit | Fuel Based | 4x7 |
| 2 | Uniform | Scattered | Adaptive Feasible | 50 | 10 | Soft Limit | Fuel Based | 9x7 |
| 3 | Uniform | Scattered | Adaptive Feasible | 100 | 10 | Soft Limit | Fuel Based | 9x7 |
| 4 | Uniform | Scattered | Adaptive Feasible | 150 | 10 | Soft Limit | Fuel Based | 9x7 |
| 5 | Uniform | Scattered | Adaptive Feasible | 50 | 20 | Soft Limit | Fuel Based | 9x7 |
| 6 | Uniform | Scattered | Adaptive Feasible | 100 | 20 | Soft Limit | Fuel Based | 9x7 |
| 7 | Uniform | Scattered | Gaussian Distribution | 50 | 10 | Soft Limit | Fuel Based | 4x7 |
| 8 | Uniform | Scattered | Uniform Distribution | 50 | 10 | Soft Limit | Fuel Based | 4x7 |
| 9 | Uniform | Scattered | Adaptive Feasible | 50 | 10 | Hard Limit | Frequency Based | 9x7 |
| 10 | Uniform | Scattered | Adaptive Feasible | 50 | 10 | Hard Limit | Fuel Based | 9x7 |
| 11 | Uniform | Scattered | Adaptive Feasible | 100 | 10 | Hard Limit | Fuel Based | 9x7 |
| 12 | Uniform | Scattered | Adaptive Feasible | 150 | 10 | Hard Limit | Fuel Based | 4x7 |
| 13 | Uniform | Scattered | Adaptive Feasible | 150 | 10 | Hard Limit | Fuel Based | 9x7 |
| 14 | Uniform | Scattered | Adaptive Feasible | 250 | 10 | Hard Limit | Fuel Based | 9x7 |
| 15 | Uniform | Heuristic | Adaptive Feasible | 50 | 10 | Hard Limit | Fuel Based | 9x7 |
| 16 | Uniform | Single Point | Adaptive Feasible | 50 | 10 | Hard Limit | Fuel Based | 9x7 |
| 17 | Uniform | Two Point | Adaptive Feasible | 50 | 10 | Hard Limit | Fuel Based | 9x7 |
| 18 | Uniform | Arithmetic | Adaptive Feasible | 50 | 10 | Hard Limit | Fuel Based | 9x7 |
| 19 | Adaptive Linear | Scattered | Adaptive Feasible | 50 | 10 | Hard Limit | Fuel Based | 9x7 |
| 20 | Uniform | Scattered | Adaptive Feasible | 150 | 10 | Hard Limit | Frequency Based | 9x7 |

Table 4.5: Optimization results.

| **No.** [-] | **Best Fitness** [-] | **Average Fitness** [-] | **Fuel Saved** [%] | **Increment in $NO_x$** [%] | **Increment in PM** [%] | **Increment in HC** [%] | **Increment in HC+$NO_x$** [%] | **Torque Match [NRMSE]** | **Time Spent [sec]** |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.94 | 1.02 | 3.24 | 12.16 | 0.40 | -0.09 | 11.24 | 0.040 | 6335.0 |
| 2 | 0.90 | 1.19 | 4.95 | 20.95 | -22.74 | -0.12 | 19.38 | 0.041 | 5655.3 |
| 3 | 0.90 | 1.65 | 5.01 | 42.85 | -25.31 | -0.26 | 39.65 | 0.042 | 11006.3 |
| 4 | 0.88 | 1.33 | 6.09 | 17.27 | -21.32 | -0.13 | 15.98 | 0.045 | 17118.1 |
| 5 | 0.90 | 0.90 | 4.99 | 36.17 | -25.14 | -0.12 | 33.47 | 0.041 | 12174.2 |
| 6 | 0.88 | 0.91 | 6.00 | -14.93 | 15.12 | -0.19 | -13.84 | 0.047 | 42154.4 |
| 7 | 0.87 | 1.99 | 6.75 | -7.71 | 295.15 | 0.06 | -7.13 | 0.065 | 5561.6 |
| 8 | 0.93 | 0.98 | 3.34 | -5.55 | 50.21 | -0.12 | -5.14 | 0.043 | 6536.6 |
| 9 | 0.99 | 0.99 | 0.58 | -0.06 | -4.05 | -1.56 | -0.17 | 0.024 | 5312.5 |
| 10 | 0.95 | 0.95 | 2.79 | -4.58 | -16.36 | -0.18 | -4.25 | 0.029 | 5269.0 |
| 11 | 0.94 | 0.95 | 2.88 | 14.52 | -21.40 | -0.19 | 13.43 | 0.029 | 10808.6 |
| 12 | 0.96 | 0.96 | 1.96 | 17.35 | -14.00 | -0.11 | 16.05 | 0.027 | 15178.2 |
| 13 | 0.94 | 0.94 | 3.06 | 16.53 | -20.74 | -0.23 | 15.29 | 0.029 | 15552.1 |
| 14 | 0.94 | 0.94 | 3.07 | -11.19 | -7.59 | -0.20 | -10.37 | 0.031 | 27710.6 |
| 15 | 0.97 | 0.97 | 1.61 | -9.06 | -6.23 | -0.12 | -8.40 | 0.027 | 5620.1 |
| 16 | 0.96 | 0.96 | 2.02 | 25.43 | -14.89 | -0.14 | 23.53 | 0.027 | 5401.4 |
| 17 | 0.96 | 0.96 | 2.10 | 0.82 | -9.43 | -0.13 | 0.75 | 0.028 | 5349.9 |
| 18 | 0.97 | 0.97 | 1.38 | 17.62 | -3.20 | -0.10 | 16.30 | 0.026 | 5355.5 |
| 19 | 0.95 | 0.95 | 2.79 | -4.58 | -16.36 | -0.18 | -4.25 | 0.029 | 5940.5 |
| 20 | 0.98 | 0.99 | 0.94 | -0.42 | -5.62 | -3.16 | -0.62 | 0.024 | 16557.9 |

The effect of each optimization configuration parameter is given as follows:

**Creation Function:** Comparison of the results of the $10^{th}$ and $19^{th}$ iterations show that changing the creation function did not affect on the cost function. Although the time spent on optimization decreased by 30 sec when the creation function was changed from Uniform to Adaptive Linear.

**Crossover Function:** Results of the $10^{th}$, $15^{th}$, $16^{th}$, $17^{th}$ and $18^{th}$ iterations indicate that Scattered option resulted in the best fuel efficiency by 2.79% whereas the Arithmetic option turned out to be the worst by 1.38% decrement in fuel consumption. As for the optimization times, Scattered option was the fastest with 5269.0 sec optimization time. The optimization held by Two Point, Arithmetic, Single Point and Heuristic options lasted for 5349.9, 5355.5, 5401.4, 5620.1 seconds respectively. Even

though any of the optimization options did not violate any of the emission limits, $NO_x$ emissions were increased by 25.43%, 0.82% and 17.62% for Single Point, Two Point and Arithmetic options, respectively. The creation function did not have a major impact on torque match as it differs from one option to another at most 0.003 in terms of NRMSE value.

**Mutation Function:** To compare the effect of mutation function on the cost, time and emissions, $1^{st}$, $13^{th}$ and $14^{th}$ iterations were performed. As can be seen from the results, the Gaussian mutation function is the most effective option in terms of fuel efficiency by 6.75% and optimization time of 5561.6 sec. For the sake of fuel efficiency, the torque match was destroyed with 0.065 NRMSE value. The torque match is so crucial especially for the fuel-based optimization domains since the full load area is selected as the optimization domain. With 0.065 NRMSE value, the desired full load torque cannot be maintained. The PM emission was also affected badly such that it increased by 295.15%. However, similar results were observed in Uniform Distribution and Adaptive Feasible options. Respectively, 3.34% and 3.24% fuel saved at the end of $14^{th}$ and $1^{st}$ optimizations. $NO_x$ emission increased by 12.16% with Adaptive Feasible option whereas 7.71% and 5.5% decrease observed with Gaussian and Uniform Distribution options. The slowest option, Uniform Distribution, lasted 201.6 sec more than Adaptive Feasible and 975 sec more than Gaussian Distribution options.

**Population Size:** $10^{th}$, $11^{th}$, $13^{th}$ and $14^{th}$ iterations were investigated to understand the effect of population size on fuel economy and optimization time mainly. The population size was increased from 50 to 250 while maintaining the other settings. As increasing the population size of each generation, the time spent during the optimization increased with almost linearly behavior. For the population size of 50, 100, 150 and 250, the optimizations lasted for 5269.0, 10808.6, 15552.1 and 27710.6 seconds. Increasing the population size from 50 to 250 increased the fuel efficiency from 2.79% to 3.07%. However, the increment was so small such as 0.01% while altering the population size from 150 to 250. On the other hand, the time spent increased dramatically as well. Therefore, 150 was selected as the maximum limit for the other trials. The

torque match was slightly affected by the population size such that for the population size of 50, 100 and 150 NRMSE was 0.029 whereas it was 0.031 for the population size of 250.

**Maximum Generations:** The maximum generation number is important to have different mutated populations in each generation to decrease the average and best fitness of the cost function. According to the results of $3^{rd}$ and $6^{th}$ iterations, increasing the maximum generation from 10 to 20 shows a decrement in best fitness from 0.90 to 0.88, average fitness from 1.65 to 0.91 and an increment in fuel saved by from 5% to 6%. However, by doubling the generation number, the time spent was almost quadrupled such as increasing from 11006.3 seconds to 42154.4 seconds. Although the $NO_x$ delta emission from the reference cycle was decreased from 42.85% to -14.93%, PM emission was increased from -25.31% to 15.12%. The HC emission was slightly affected as increasing by 0.07%. Also, increasing the maximum generation had a positive effect on the torque match by decreasing the NMRSE value from 0.047 to 0.042.

**Limitation Method:** As for the limitation method, the optimization parameters' upper and lower possible values are limited with hard and soft limit options which were explained in the MATERIALS and METHODS chapter. The soft limit was very effective in terms of fuel efficiency such that it almost doubled it from 3.06% to 6.09% in the $13^{th}$ and $4^{th}$ iterations. Nevertheless, the torque match was decreased drastically with an increase in the NRMSE value from 0.029 to 0.045. In both soft and hard limit options, emission results are similar. The $NO_x$ emission increased by 17.27% and 16.53% whereas PM emission decreased by 21.32% and 20.74% and HC emission decreased by 0.13% and 0.23% in soft and hard limited iterations respectively. The time spent on optimization with hard limits is 1566 seconds less than the optimization time of the soft limit option.

**Domain Definition Method:** The domain of the calibration was defined according to the engine operating points which have the highest contribution in terms of fuel consumption or time spent during the cycle. The $13^{th}$ and $20^{th}$ iterations were

compared to understand the effect of different domain selection, mainly on fuel consumption and torque match. The fuel-based option defines domain around the full load area generally. Therefore, changing the calibration around full load can have a major effect on the torque match. As can be seen from the results, the torque match is better in frequency based options compared to fuel based options. The NRMSE values for the torque matches in the $20^{th}$ and $13^{th}$ iterations are recorded as 0.024 and 0.029. For the sake of torque match, the fuel efficiency is better with fuel based domain definition option. The increase in fuel saved is 0.94% in frequency based iteration whereas it is 3.06% in the iteration with the fuel based domain. The reason, why the frequency based domain is less fuel efficient, is the domain itself. According to frequency based domain definition, the domain was defined around the zero torque area. Therefore, there was no big margin to reduce the fuel consumption.

**Domain Size:** Although the number of calibration maps was constant, the number of points on each calibration map were iterated in between 4 and 9. By comparing $1^{st}$ and $2^{nd}$ or $12^{th}$ and $13^{th}$ iterations, it can be said that increasing the number of optimized parameters can increase the fuel efficiency. The fuel efficiency was increased from 1.96% to 3.06% in $12^{th}$ and $13^{th}$ iterations. However, the relation of optimization time and the domain size was not as expected. Although, the time spent was increased from 15178.2 seconds to 15552.1 seconds while going from $12^{th}$ to $13^{th}$ whereas it was decreased from 6355.0 seconds to 5655.3 seconds while going from $1^{st}$ and $2^{nd}$ iteration. This could be due to the different excessive loads on the PC during the optimization.

As a summary, all the iterations with soft limit options represent higher fuel efficiency. However, the differentiation from the desired torque in these iterations is excessive. Also, the results show that the fastest iterations were the ones with the lowest number of population size as expected.

# 5. DISCUSSION

According to the simulation results, the fuel consumption was decreased by 3.07%, while still being within the Euro 6d-Temp regulation tailpipe emission limits, and satisfying mechanical and performance constraints defined in MATERIALS & METHODS chapter.

Since the optimization cost and constraints are calculated according to the model outputs, the model accuracy has a big impact on the results. Therefore, the model accuracy of each system can be improved to have safer calibration datasets at the end of this optimization. Especially, engine-out emission models of PM and HC can be more accurate with proper measurements taken on the engine testbed. Moreover, $CO_2$ emissions can also be taken into consideration since the limits for $CO_2$ will be decreased in the next legislation package of Euro 6d which will be valid in Europe from 2020. Due to the lack of measurement data, $CO_2$ was not considered in this study.

The comparison of different modeling options with neural networks contributed to the SCR model accuracy and the time spent on modeling. As an improvement, $NH_3$ dosing quantity with a closed-loop control by a simplified DCU model can be implemented to the SCR model. Due to the lack of data and closed-loop system complexity, $NH_3$ dosing and DCU model was not be considered in this study. Validation results indicate that during the steady-state parts, the tailpipe NOx model is more accurate compared to transients. This can be improved by adding transient corrections to the SCR model in order to catch the high transients with better accuracy. Moreover, to have more representative tailpipe PM and HC emissions, DPF and DOC can be modeled by filtration and conversion efficiency parameters such as exhaust gas temperature and mass flow. Also, DPF model can be used to observe engine behavior during regeneration mode. To control the DPF regeneration, the release conditions and post injection parameters can be implemented to the ECU model. By building an accurate DPF model, oil dilution can be monitored, and defined as a constraint to the cost function.

Optimization results show that different calibration areas have different margins in terms of optimization cost. To demonstrate, fuel-based domain definition resulted in less fuel consumption compared to frequency-based domain definition. The frequency-based domain selects the optimization domain mainly from zero desired torque range in which there is no margin for fuel-efficiency. On the other hand, the calibration domain based on the fuel consumption, has more margin for fuel-efficiency but this domain corresponds to the full load area in which the torque match is crucial. Therefore, hard constraints were put to maintain the highest torque match possible. Also, the smoothness of the maps is maintained by defining strong upper and lower limits to the optimization parameters. However, this hard limitation increases the optimization cost compared to soft limitation. To decrease the cost with an acceptable smoothness, neighbours of the optimization domain can be adapted as well. Furthermore, the level of smoothness can be defined as a constraint in the cost function.

Optimization loops performed with different configurations to find the optimum calibration parameters as well as the optimum set of optimization configurations. This optimum combination of configurations can be considered valid just for the engine optimization problem for a predefined cycle since the dependency, non-linearity and accuracy of the problem inputs, such as models, can differ from one optimization problem to another. The optimum configuration was selected according to fuel efficiency, torque match, emission violations and the time spent during the optimization. Although the iterations with soft limit represent the highest fuel efficiency, they are lack of torque match. Therefore, the optimum configuration option was selected from the optimizations performed with hard limit options in which the desired torque can be maintained. By comparing the hard-limited iterations according to their best fitness values, $14^{th}$, $13^{th}$ and $11^{th}$ iterations were defined as the top three iterations with the same fitness value of 0.94. Out of these three options, the best one can be selected according to the focus of the calibration. In case the main focus is fuel consumption and $NO_x$ emission reduction, the $14^{th}$ iteration should be selected as the best option due to 3.07% reduction in fuel consumption and 11.19% reduction in $NO_x$ emission.

If time efficiency is a big concern, the $13^{th}$ iteration can be selected since it provides 3.06% fuel efficiency within less time.

Direct implementation of this calibration environment to the modern ECUs is not easy due to the amount of data, and the time spent for modeling and optimization. On the other hand, all the modeling and optimization activities can be performed on the cloud much faster. Therefore, additional hardware, which can be a bridge between the cloud and the ECU, can be integrated for data transfer. Moreover, the optimization method can be extended to specific driving conditions of that moment by implementing more input parameters to the models such as driver, traffic jam, altitude profile of the defined route and the weather. Although this information can also be gathered on the cloud, more inputs will increase the system complexity and the amount of data required to build accurate models. Nevertheless, this calibration methodology can be turned into a commercial product with these hardware and software integrations.

# 6. CONCLUSION

In this study, a novel methodology for the optimization of a diesel engine calibration was developed. The optimization target was fuel consumption while being emission compliant for a predefined route. The optimization was performed automatically by using model-based calibration techniques with genetic algorithms. It is believed that the proposed method is able to provide a representative solution for the real-life calibration problem.

First, the model-based calibration environment was built as a combination of ECU, ICE and SCR models. 2D lookup tables were used to build the ECU model whereas the ICE and SCR models were created with neural networks. Different neural network structures were tried to find the optimum configuration with the best fit. All the models were validated and according to validation results, they all considered as accurate enough for this methodology development. Second, the optimization domain, constraints and cost function were defined. Finally, different options were iterated to find the best fitness value for the cost, in other words, the best fuel efficiency. The optimization results were compared in detail in terms of fuel efficiency, time spent, torque match and emissions as well. The best optimization configuration consists of a population of 250 individuals, 10 maximum generations, hard limits of lower and upper thresholds for optimization parameters, Scattered crossover function, Uniform creation function, Adaptive Feasible mutation function, fuel consumption based domain definition and 9x7 domain size. This optimum configuration set of optimization results in reduction in fuel consumption, $NO_x$, PM and HC by 3.07%, 11.19%, 7.59% and 0.20% respectively. Also, smoothness was maintained by hard limits and the torque match was considered as enough with 0.031 NMRSE value.

## 6.1. Originality and Contributions

The originality of this study comes mainly from the single objective cost function for the genetic algorithm. Instead of targeting multi-objectives such as emissions and fuel consumption at the same time, a single objective was proposed as just targeting fuel efficiency with constraints of emissions, performance, mechanical and smoothness limits. Another original idea proposed in this study is to define the calibration domain based on the fuel consumption weights of the operating points instead of the frequency of them. Although this was risky due to high torque request in full load area, the torque match was defined as a performance constraint in the cost function. The investigation of different options for neural network modeling and optimization with the genetic algorithm can be also considered as a valuable outcome for future studies. Furthermore, optimizing the dataset on an RDE cycle, with constraints defined as it is in the Euro 6d-Temp tailpipe emission limits makes this fuel-efficient calibration dataset more representative for real-world driving conditions.

## 6.2. Outlook and Future Work

The proposed method has several advantages over similar approaches in the literature. Fuel consumption based optimization domain definition with a certain torque match limit during the optimization resulted in relatively better fuel efficiency compared to conventional frequency based domain definitions as defined in RESULTS chapter. For the ANN based SCR modeling, the most accurate and time efficient structure was selected out of 120 combinations. Similar to modeling, the GA solver options were also investigated in order to find the lowest cost with the minimum time. Both of the outcomes from the modeling and optimization iterations can be used in the future model-based calibration methodologies.

The model accuracies of engine-out emission models can be improved by more measurement data. Transient factors can be implemented to the ECU and ICE models to improve the model performance on transients. Moreover, a closed-loop control model of $NH_3$ dosing quantity can be integrated into the SCR model to increase the tailpipe $NO_x$ accuracy. To reduce the need for measurement data, the SCR temperature model can be integrated. As an improvement on HC and PM tailpipe emission model accuracies, DPF and DOC can be modeled by filtration and conversion efficiency related parameters such as exhaust temperature and mass flow. Also, instead of using one RDE cycle, more data can be used to define the optimization domain to cover more driving scenarios on the same route.

As another future work, the optimization domain can be defined automatically for frequently commuted routes according to previously recorded data of engine operating points, GPS coordinates, weather and traffic information. All the data can be sent to the cloud with hardware connected to the ECU. Once enough data is collected, all the required models can be built and optimization can be performed automatically on the cloud for each predefined route. The generated calibration can be downloaded into the ECU via the connected hardware which provides the communication with the cloud.

# REFERENCES


1. Isermann, R., *Engine Modeling and Control*, Springer, Berlin, Germany, 2014.

2. Langouat, H., J. Treur, D. Sinoquet and Q.-H. Tran, "Engine calibration: multi-objective constrained optimization of engine maps", *Optimization and Engineering*, Vol. 12, pp. 407–424, 2011.

3. Brijesh, P. and S. Sreedhara, "Exhaust emissions and its control methods in compression ignition engines: A review", *International Journal of Automotive Technology*, Vol. 14, pp. 195–206, 2013.

4. Heywood, J. B., *Internal combustion engine fundamentals*, McGraw-Hill, New York, NY, USA, 1988.

5. Johnson, T. V., "Diesel Emission Control in Review", *SAE International Journal of Fuels and Lubricants*, Vol. 2, pp. 1–12, 2009.

6. Majewski, W. A. and M. K. Khair, *Diesel Emissions and Their Control*, SAE International, Warrendale, Pennsylvania, USA, 2006.

7. Azam, A., S. Ali and A. Iqbal, "Emissions from Diesel Engine and Exhaust After Treatment Technologies", *4th International Conference on Energy, Environment and Sustainable Development*, 2016.

8. Davidson, J. and J. M. Norbeck, *An Interactive History of the Clean Air Act, 1st Edition*, Elsevier, London, 2012.

9. Hooftman, N., M. Messagie, J. V. Mierlo and T. Coosemans, "A review of the European passenger car regulations – Real driving emissions vs local air quality", *Renewable and Sustainable Energy Reviews*, Vol. 86, pp. 1–21, 2018.

10. Androne, C.-S. and S. B. Rao, "Vehicle Simulation for Powertrain System Testing", *Master's Thesis*, 2017.

11. Tutuianu, M., P. Bonnel, B. Ciuffo, T. Haniu, A. M. N. Ichikawa, J. Pavlovic and H. Steven, "Development of the World-wide harmonized Light duty Test Cycle (WLTC) and a possible pathway for its introduction in the European legislation", *Transportation Research Part D: Transport and Environment*, Vol. 40, pp. 61–75, 2015.

12. Triantafyllopoulos, G., D. Katsaounis, D. Karamitros, L. Ntziachristos and Z. Samaras, "Experimental assessment of the potential to decrease diesel NOx-emissions beyond minimum requirements for Euro 6 Real Drive Emissions (RDE) compliance", *Science of The Total Environment*, Vol. 618, pp. 1400–1407, 2018.

13. EU, C. R., "Amending Regulation (EC) No 692/2008 as Regards Emissions From Light Passenger and Commercial Vehicles (Euro 6)", *Official Journal of the European Union*, Vol. 82, p. 1–98, 2016, `https://eur-lex.europa.eu/eli/reg/2016/427/oj`.

14. Transportation, I. C. O. C., "Real-Driving Emissions test procedure for exhaust gas pollutant emissions of cars and light commercial vehicles in Europe", *Policy Update*, 2017, `https://theicct.org/sites/default/files/publications/EU-RDE_policy-update_Jan2017_vF.pdf`.

15. McCulloch, W. S. and W. Pitts, "A Logical Calculus of the Ideas Immanent in Nervous Activity", *Bulletin Of Mathematical Biophysics*, Vol. 5, p. 115–133, 1943.

16. Basheera, I. and M. Hajmeerb, "Artificial neural networks: fundamentals, computing, design, and application", *Journal of Methods Microbiological*, Vol. 43, p. 3–31, 2000.

17. Halıcı, U., "From Biological to Artificial Neuron Model", *EE543, Artificial Neural Networks, Lecture Notes*, pp. 1–14, `http://users.metu.edu.tr/halici/`

courses/543LectureNotes/lecturenotes-pdf/ch1.pdf.

18. Jain, A. K., J. Mao and K. Mohiuddin, "Artificial Neural Networks: A Tutorial", *IEEE Computer*, Vol. 29, pp. 31–44, 1996.

19. Alonso, J. M., F. Alvarruiz, J. M. Desantes, L. Hernández, V. Hernandez, and G. Molto, "Combining Neural Networks and Genetic Algorithms to Predict and Reduce Diesel Engine Emissions", *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION,*, Vol. 11, pp. 46–55, 2007.

20. Mathworks, *Global Optimization Toolbox, User's Guide*, 2019, https://www.mathworks.com/help/pdf_doc/gads/gads_tb.pdf.

21. Chambers, L., *The Practical Handbook of Genetic Algorithms, Applications, 2nd Edition*, Chapman and Hall/CRC, Boca Raton, FL, USA, 2000.

22. Mallamo, F., M. Badami and Millo, "Application of the Design of Experiments and Objective Functions for the Optimization of Multiple Injection Strategies for Low Emissions in CR Diesel Engines", *SAE 2004 World Congress & Exhibition*, 2004.

23. Gaballo, M. R., M. Giodice, A. Diano, F. Fersini, F. Miccolis, S. Mannal and S. Motz, "Application of a Modular Simulation Approach: Optimizations from Combustion to Vehicle Management", *12th International Conference on Engines & Vehicles*, 2015.

24. Castagne, M., Y. Bentolila, F. Chaudoye, A. Halle, F. Nicolas and D. Sinoquet, "Comparison of Engine Calibration Methods Based on Design of Experiments (DoE)", *Oil & Gas Science and Technology*, Vol. 63, pp. 563–582, 2008.

25. Millo, F., P. Arya and F. Mallamo, "Optimization of Automotive Diesel Engine Calibration Using Genetic Algorithm Techniques", *Energy*, Vol. 158, pp. 807–819, 2018.

26. Atkinson, C., M. Allain and H. Zhang, "Using Model-Based Rapid Transient Calibration to Reduce Fuel Consumption and Emissions in Diesel Engines", *SAE World Congress & Exhibition*, 2008.

27. Polterauer, P. and H. Waschl, "An implementable multi-calibration strategy to address real driving emissions of CI engines", *2017 American Control Conference (ACC)*, 2017.

28. Malikopoulos, A. A., D. N. Assanis and P. Papalambros, "Optimal Engine Calibration for Individual Driving Styles", *SAE World Congress & Exhibition*, 2008.

29. Sujesh, G. and S. Ramesh, "Modeling and control of diesel engines: A systematic review", *Alexandria Engineering Journal*, Vol. 57, pp. 4033–4048, 2018.

30. Guzzella, L. and C. H. Onder, *Introduction to Modeling and Control of Internal Combustion Engine Systems, 2nd Edition*, Springer, Berlin, Germany, 2000.

31. Isermann, R. and H. Sequenz, "Model-based development of combustion-engine control and optimal calibration for driving cycles: general procedure and application", *IFAC-PapersOnLine*, Vol. 49, pp. 633–640, 2016.

32. Tauzia, X., A. Maiboom and H. Karaky, "Semi-physical models to assess the influence of CI engine calibration parameters on NOx and soot emissions", *Applied Energy*, Vol. 208, pp. 1505–1518, 2017.

33. Ismail, H. M., H. K. Ng, C. W. Queck and S. Gan, "Artificial neural networks modelling of engine-out responses for a light-duty diesel engine fuelled with biodiesel blends", *Applied Energy*, Vol. 92, pp. 769–777, 2012.

34. Nikzadfara, K. and A. H. Shamekhi, "Investigating the relative contribution of operational parameters on performance and emissions of a common-rail diesel engine using neural network", *Fuel*, Vol. 125, pp. 116–128, 2014.

35. Asprion, J., O. Chinellato and L. Guzzella, "Optimisation-oriented modelling of the NOx emissions of a Diesel engine", *Energy Conversion and Management*, Vol. 75, pp. 61–73, 2013.

36. Grahn, M., K. Johansson and T. McKelvey, "Model-based diesel Engine Management System optimization for transient engine operation", *Control Engineering Practice*, Vol. 29, pp. 103–114, 2014.

37. Nikzadfara, K. and A. H. Shamekhi, "Investigating a new model-based calibration procedure for optimizing the emissions and performance of a turbocharged diesel engine", *Fuel*, Vol. 242, pp. 455–469, 2019.

38. Arya, P., F. Millo and F. Mallamo, "A fully automated smooth calibration generation methodology for optimization of latest generation of automotive diesel engines", *Energy*, Vol. 178, pp. 334–343, 2019.

39. Sauermann, R., D. Boja, F. Kirschbaum and O. Nelles, "Particle Swarm Optimization for Automotive Model-Based Calibration", *6th IFAC Symposium Advances in Automotive Control*, 2010.

40. Malikopoulos, A. A., D. N. Assanis and P. Papalambros, "Real-Time Self-Learning Optimization of Diesel Engine Calibration", *Journal of Engineering for Gas Turbines and Power*, Vol. 131, pp. 455–469, 2009.

41. Parliament, E. and of the Council of 20 June 2007, "Type Approval of Motor Vehicles with Respect to Emissions From Light Passenger and Commercial Vehicles (Euro 5 and Euro 6) and on Access to Vehicle Repair and Maintenance Information", *Official Journal of the European Union*, Vol. 171, p. 1–16, 2007, `https://eur-lex.europa.eu/eli/reg/2007/715/oj`.

42. Gmbh, A. L., *AVL CAMEO, All-In-One Powertrain Calibration*, `https://www.avl.com/documents/10138/885957/01814_Cameo_PA3023E_web.pdf/00252e94-e667-4b12-9f7a-c06c6c0281bc`.

43. Wirojsakunchai, E., E. Schroeder, C. Kolodziej, D. E. Foster, N. Schmidt, T. Root, T. Kawai, T. Suga, T. Nevius and T. Kusaka, "Detailed Diesel Exhaust Particulate Characterization and Real-Time DPF Filtration Efficiency Measurements During PM Filling Process", *SAE World Congress & Exhibition*, 2007.

44. Knafl, A., S. B. Busch, M. Han, S. V. Bohac, D. N. Assanis, P. G. Szymkowicz and R. D. Blint, "Characterizing Light-Off Behavior and Species-Resolved Conversion Efficiencies During In-Situ Diesel Oxidation Catalyst Degreening", *SAE 2006 World Congress & Exhibition*, 2006.