CHALMERS
UNIVERSITY OF TECHNOLOGY

Online Model Based Engine Calibration using Gaussian Process Regression

With Reference to Volvo Penta's VIRTEC Project

Master's thesis in Systems, Control and Mechatronics

ELIAS HAGENTOFT
METHISGE MAHEEL ISANKA DABARERA

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

Masters' thesis 2018:EENX30

Online Model Based Engine Calibration using Gaussian Process Regression

With Reference to Volvo Penta's VIRTEC Project

ELIAS HAGENTOFT
METHISGE MAHEEL ISANKA DABARERA

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2018

Online Model Based Engine Calibration using Gaussian Process Regression
With Reference to Volvo Penta's VIRTEC Project
ELIAS HAGENTOFT
METHISGE MAHEEL ISANKA DABARERA

Supervisors:
Jonas Sjöblom, Department of Mechanical and Maritime Studies, Division of Combustion and Propulsion Systems
Ali Ghanaati, Department of Mechanical and Maritime Studies, Division of Combustion and Propulsion Control
Ethan Faghani, Volvo Penta
Supervisor and Examiner: Jonas Sjöberg, Department of Electrical Engineering

Masters' Thesis 2018:EENX30
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2018

Online Model Based Engine Calibration using Gaussian Process Regression
With Reference to Volvo Penta's VIRTEC Project
Elias Hagentoft
Methisge Maheel Isanka Dabarera
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Engine Calibration is an important stage in the development phase of a new engine. This is expensive, time consuming and requires expert knowledge.

This thesis presents the usage of Gaussian Process Regression (GPR) as a modeling technique to mitigate the above drawbacks.

The GPR model uses operating points of the engine speed vs engine torque map as a reference to conduct a Design of Experiments using Latin Hypercube Sampling technique, for input parameters in order to predict Brake Specific Fuel Consumption as the output parameter. Six input parameters are taken into consideration for the thesis namely; engine speed, engine torque, injection timing, injection pressure, throttle position and estimated EGR valve position.

The work concerns developing a methodology rather than a collection of final models, hence an artificial neural network is used to generate data. The main finding with regarding input data is that some operating points do not allow for any variability, hence providing non-informative data for the regression methods.

One global model and three separate local models are created. The local models are designed for the low torque, part load and high torque regions. The part load and the low torque are the most accurate local models and the high torque local model is the least accurate in terms of mean squared prediction error and fit.

These models are used to construct the main deliverable, which is a modelling methodology for online engine calibration.

The Gaussian Process Regression is used in determining the importance of each of the model parameters. The results from this contradicts some well known facts and needs further investigation.

Keywords: Engine Calibration, Gaussian Process Regression, Virtual Test Cell, Online Refinement, Latin Hypercube Sampling

Acknowledgements

We would like to express our gratitude to our examiner and supervisor Jonas Sjöberg, Professor at Electrical Engineering Department at Chalmers University of Technology for his guidance in conducting the thesis. We want to thank Ethan Faghani, Project Manager at Volvo Penta and Andreas Nyman, Manager, Engine Testing 2, Volvo Penta for presenting the thesis to us and for the continuous help and guidance.The current thesis is written as a part of the Virtual Engine Calibration (VirCal) project funded by Fordonsstrategisk Forskning och Innovation (FFI) and Vinnova. We would also like to thank Jonas Sjöblom, Director of the Masters Program Automotive Engineering at Chalmers University of Technology and Ali Ghanaati at the Division of Combustion and Propulsion Control at Chalmers University of Technology for their valuable opinions and feedback. Finally, we would like to thank our families and friends for motivating us throughout this journey.

Elias Hagentoft and Maheel Dabarera, Göteborg, June 2018

Contents

List of Figures

List of Tables

List of Symbols

τ	engine torque [Nm]
ω	engine speed [rpm]
P	engine Power [W]
$\dot{m}$	massflow of fuel [g/s]
x	general model input vector [-]
x^T	the transpose of vector x [-]
$f(x), f$	function value
f_*	function value of test set
x_*	general model input vector of test set [-]
w	weight vector of a linear model [-]
$\sim$	distributed according to
N_{train}	number of samples or points of the training set
N_*	number of samples or points of the test set
$\sigma^2_{N_{train}}$	variance of additive Gaussian noise [-]
$\mathcal{N}$	normal distribution
Y	vector containing all training outputs [-]
X	vector containing all training inputs [-]
X_*	vector containing all test inputs [-]
$p(Y\|X, \mathrm{w})$	conditional probability of Y given X and w [-]
I	identity matrix
Σ_p	prior covariance matrix on the weights, w
$\phi(x)$ or $\mathbf{\Phi}(X)$	maps input x or X into a N-dimensional feature space [-]
$\mathbb{E}$	expectation
$k(x_m, x_n)$	covariance function evaluated at x_m and x_n [-]
k_{1N}	covariance function evaluated at x_1 and x_N etc [-]
$m(x)$	mean function, $m(x) = \mathbb{E}[f(x)]$ [-]
GP	Gaussian process, $f \sim GP(m(x), k(x_m, x_n))$, the function f is distributed as a Gaussian process with mean function $m(x)$ and covariance function $k(x_m, x_n)$ [-]
σ^2	signal variance hyperparameter [-]
l^2	length scale hyperparameter [-]
σ^2_{noise}	noise variance hyperparameter [-]
δ_{mn}	Kronecker Delta, δ_{mn}=1 iff $m = n$ and 0 otherwise [-]

θ	set of hyperparameters [-]
$\log(.)$	natural logarithm (base e) [-]
$K(X,X), K$	covariance matrix of all covariances between training inputs [-]
$K(X,X_*), K_*$	covariance matrix of all covariances between training and test inputs, $K(X,X_*) = K(X_*,X)$ [-]
$K(X_*,X_*)$	covariance matrix of all covariances between test inputs [-]
$\|.\|$	absolute value
$\triangleq$	an equality which acts as a definition [-]
$\bar{f}_*$	Gaussian Process posterior mean vector for the test set
$cov(f_*)$	Gaussian process posterior covariance for the test set
$k(x_*)$ or k_*	vector, short of $K(X,x_*)$, when there is only a single test point
$V[f_*]$	predicted variance of the function values f_*
CI	confidence interval
$/$	left matrix divide: $A\backslash b$ is the vector z which solves $Az = b$
Cholesky(A)	Cholesky decomposition: L is a lower triangular matrix such that $LL^T = A$
$\hat{y}$	prediction from the Gaussian Process Regression [-]
$\mu_{y_{train}}$	mean of the output data of the training set
$s_{y_{train}}$	standard deviation of the output data of the training set
$y_{train,norm}$	normalized output data of the training set [-]
$y_{test,norm}$	normalized output data of the test set [-]
$\mu_{x_{train}}$	mean of the input data of the training set
$s_{x_{train}}$	standard deviation of the input data of the training set
$x_{train,norm}$	normalized input data of the training set [-]
$x_{test,norm}$	normalized input data of the test set [-]
N	number of samples from a Latin Hypercube sampling at each operating point
N_{op}	number of operating points
N_{tot}	total number of samples
IP	injection pressure [deg]
IT	injection timing [BTDC deg]
det(A)	determinant of matrix A [-]

Abbreviations

ANN	Artificial Neural Network
ARD	Automatic Relevance Determination
BSFC	Brake Specific Fuel Consumption
BTDC	Before Top Dead Center
CI	Confidence Interval
DoE	Design of Experiments
ECU	Engine Control Unit
EGR	Exhaust Gas Recirculation
GPR	Gaussian Process Regression
HIL	Hardware-In-Loop
IP	Injection Pressure
IT	Injection Timing
LHS	Latin Hypercube Sampling
MSPE	Mean Square Prediction Error
NOP	Nozzle Opening Pressure
OP	Operating Point
PL	Part-Load
SCR	Selective Catalytic Reduction
SE	Squared Exponential
s.t	Subject To
UP	Update Point
VIRTEC	Virtual Test Cell
VirCal	Virtual Engine Calibration

1
Introduction

This chapter gives a general introduction to the importance of engine calibration and a motivation to why model based engine calibration is of interest. The chapter also gives a framework to the methodology used in this thesis. The background and motivation for the company Volvo Penta's interest in engine calibration is mentioned as well.

1.1 Background

In the development phase of new engines, an engine calibration is performed. Engine calibration is the process of adjusting engine control parameters to meet design defined criteria. Meeting strict emission standards and continuously decreasing fuel consumption are the main development targets for modern combustion engine design. To fulfill these demands, engine hardware and software architectures becomes increasingly complex. With the rising number of actuators and software functions, the number of parameters that need to be controlled and calibrated are increasing drastically [1].

Traditional engine calibration is an expensive and time consuming process and is usually conducted using physical test cells. Traditional calibration methods make use of a combination of engine dynamometer and vehicle testing in a fashion that is generally perceived as somewhat of an art.

The traditional procedure often requires expert knowledge and a high resource allocation (for example numerous costly sensors) is needed in order to perform the calibration on a real engine [2].

The need to reduce time and cost is driving the development of more advanced calibration techniques. For years Design of Experiments (DoE), have been introduced. This is a collection of techniques that can be used for efficiently structuring the calibration of engines. This methodology offers several benefits, when the number of control parameters is high and the knowledge of input to output correlation is relatively poor. The method of creating theoretical based engine models using DoE has thus become popular in engine development [1].

Another approach to reduce physical testing is to develop a high fidelity model instead of the plant. Also the final control system can be tested and validated on this model, This is called Hardware-In-Loop (HIL).

HIL simulation is a well-established technique used for model based calibration of engines, which contains parts of hardware in the simulation loop during the process of engine calibration. Thus, instead of testing the control algorithm on the mathematical model, HIL simulation allows the usage of real hardware during simulation [3]. Volvo Penta has developed a HIL simulation called Virtual Test Cell (VIRTEC) under the Virtual Engine Calibration (VirCal) project. The VIRTEC is a system which is a host for the mathematical engine model as well as the hardware (ECU and actuators). In this way it is easy to simulate different models using the same HIL.

The VIRTEC system aims to reduce the testing and calibration cost at Volvo Penta by transferring at least 20% of the calibration activities from the physical test cells to VIRTEC. This is done in order to increase the quality and to provide an opportunity to test the entire domain of engine operation (see Section 1.4) before it is delivered to the customers. The VIRTEC system is shown in Figure 1.1.

Figure 1.1: Volvo Penta Virtual Test Cell (VIRTEC) System.

A Volvo Penta diesel engine using Stage IV/Tier 4 Final technology is considered for the thesis. The engine uses a Selective Catalytic Reduction (SCR) method in combination with a light Exhaust Gas Re-circulation (EGR) system in order to meet the emission regulations (especially NOx emissions). The Volvo Penta SCR solution results in less maintenance for the customers, reducing maintenance costs and also results in better productivity [4]. More corporate-related information about the project is given in Section 6.

1.2 Data Generation

Data is generated using VIRTEC as a host for an existing Artificial Neural Network (ANN) model which was fitted to the engine. It is assumed to represent the real physical engine and to produce data.

There are six input parameters considered for the ANN model to generate output data. (further definition are given in A.2):

- Engine Speed (ω)

- Engine Torque (τ)
- Advance Angle/ Injection Timing
- Nozzle Opening Pressure (NOP) / Injection Pressure
- Throttle Position
- Estimated EGR Valve Position

Engine power is the product of the engine torque and engine speed:

$$P = \tau\omega \tag{1.1}$$

where τ (Nm) is the engine torque and ω (rpm) is the engine speed.

The output from the ANN model is the mass fuel flow rate $\dot{m}$ (g/s).

The mass fuel flow rate and the engine power constitute the Brake-Specific-Fuel-Consumption (BSFC) which is defined as follows:

$$BSFC = \frac{\dot{m}}{P} \tag{1.2}$$

$BSFC$ expresses how much fuel the engine uses in order to perform useful work [5].

1.3 Engine Model

The data described in Section 1.2 is an input for fitting the engine models, which is the topic for the rest of the report.

Different types of models can be used that are based on the dataset generated by the artificial neural network. Examples of modelling techniques are neural networks, polynomial models, linear regression, local neuro fuzzy models and support vector machines to name a few. In this thesis Gaussian Process Regression (GPR) is used, which is a statistical nonlinear regression method. The model created using GPR is called the *GPR engine model*. GPR is explained further in Section 2.2.

Figure 1.2 shows in a schematic manner, the relationship between inputs and outputs of the ANN and GPR engine model.

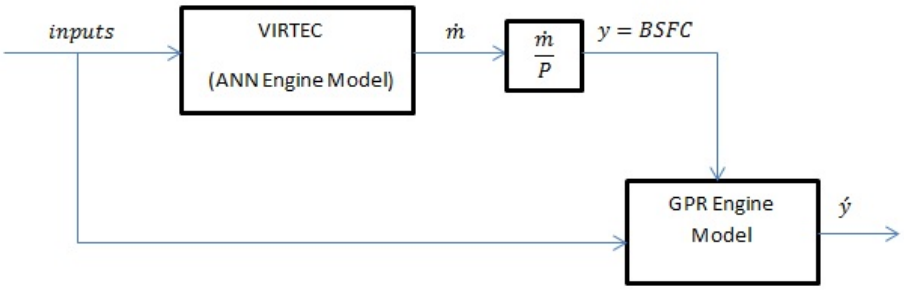

Figure 1.2: A schematic diagram of the ANN and GPR engine model with inputs and output.

The GPR Engine Model predicts for the output generated by the ANN model. $\dot{m}$ is the mass fuel flow rate, P is the engine power and y is the output, which is the BSFC and $\hat{y}$ is the prediction of the BSFC from the GPR Engine model.

1.4 Engine Operating Region and Driving Cycles

In order to setup the engine calibration according to the given dataset from a physical test cell, engine speed and torque are kept at different constant values while the other four inputs are varied throughout. These constant values define an operating point (OP). The operating region of an engine is described by a set of OPs (see Figure 1.3).

The goal of engine calibration depends on the considered operating region [6]. For the *full-load* (engine torque) region, the goal of calibration is to maximize engine torque and power. For the *part-load* region, it is to minimize fuel consumption and emissions. For the *idle* region a good engine smoothness is achieved, in order to realize the maximum driving comfort [7]. Since, most OPs lie on the part-load region, (see Section 3.1.1), the main goal of the engine calibration is to minimize *BSFC*.

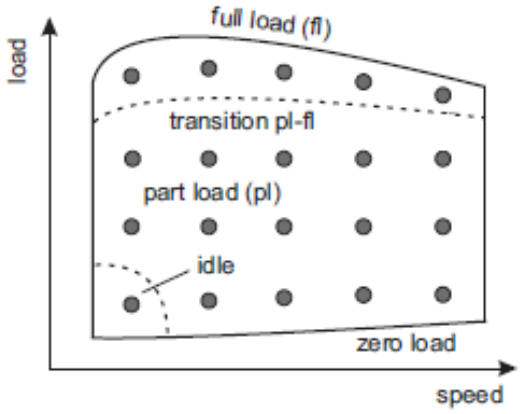

Figure 1.3: Schematic of engine operating regions with several OPs pointed out as dots. Engine speed is on the x-axis. Engine load (torque) is on the y-axis.

A *driving cycle* is represented by a sequence of OPs, and is a test run for engine calibration. It is represented by vehicle speed (m/s) versus time and contains changes in vehicle speed corresponding to different driving scenarios. Usually, the driving cycles are standardized in order to verify that the vehicle manufacturer follows the legislation regarding *BSFC* and emissions [8]. There are two types of driving cycles:

- Transient driving cycles which contains variations of engine speed and torque similar to actual road conditions and driving.
- Steady-state driving cycles which only contains periods with constant engine speed and torque. These constant periods correspond to the engine speed and torque of each OP. At each period, a stationary behaviour is observed.

In this thesis, only steady-state driving cycles are considered.

1.5 Online Refinement Algorithm for Engine Model

By using online refinement, it is possible to enhance the accuracy of the GPR engine model. This is carried out by first identifying the point in the six dimensional input space which corresponds to the highest uncertainty of the output. This defines an Update Point (UP) in the operating region at which new data is collected. The algorithm used is shown in Figure 1.4.

The uncertainty defining the UP is easily obtained by the uncertainty estimates of the GPR [10], [11].

The experiment for obtaining new data at the UP is defined as the Design of Experiments (DoE), which is described further in Section 2.1.

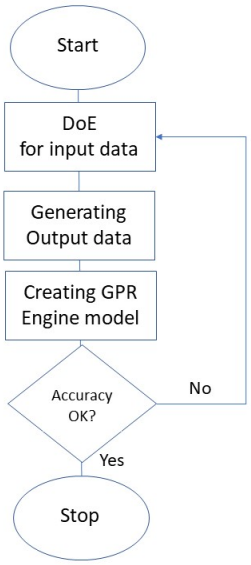

Figure 1.4: Refinement algorithm for a specific driving cycle. The loop represents the online refinement of the GPR engine model.

The criterion for the accuracy of the model upon refinement depends on the user requirements. The chosen criterion in this thesis is that the MSPE (Mean Squared Prediction Error) shall not decrease significantly upon the final iterative step of online refinement. Online Refinement is discussed further in Section 2.3 and 3.5. For illustrative purposes, the resulting decrease in MSPE is shown after two consecutive online refinements (see Section 4.2).

1.6 Limitations

The following limitations were motivated previously and are stated here again for clarity:

- Only steady-state driving cycles are considered during the project.
- For the real engine and the ANN model used, some OPs don't have variability in all input parameters. Thus those parameters were not varied at these specific OPs. The issue is described further in Section 3.1.3.

1.7 Contributions

The main deliverable of the project is a GPR engine model that can be used for an engine calibration. This calibration is intended to mainly minimize Brake Specific Fuel Consumption ($BSFC$). The model also has online capabilities as discussed in Section 1.5.
The main contribution is achieved by accomplishing a number of smaller targets:

- Incorporating an online refinement technique into the model, to improve the model.
- Conducting a DoE, that captures the entire space for varying input parameters.
- Constructing a GPR engine model, based on the DoE.

1.8 Outline

In Chapter 1 Introduction, an overview and motivation is given into model based engine calibration and engine models.

In Chapter 2 Theory, a theoretical framework of GPR is presented which is used in the rest of the thesis. The performance of the GPR models are later evaluated in Chapter 4.

Chapter 3 Methods, explains how the data used for GPR are prepared and treated in order to address non-informative trends (OPs with no variability in all input parameters) in data and match with certain assumptions inherent of GPR. Information regarding the implementation of DoE and the online refinement is also given. Local GPR engine models, covering clusters of OP's, are fitted to the data. The regions for the local models are also explained thoroughly.

In Chapter 4 Results, the performance of the GPR models are assessed as well as the implemented online refinement that come along with each GPR engine model. The result of ARD (determines the importance of each input parameter for the regression), which is a feature of GPR explained in Chapter 2 is also presented.

In Chapter 5 Future Work, possible future work for a more complete model based engine calibration is discussed.

Chapter 6 Implementation, mentions practical aspects of the thesis work, such as corporate information and the software tools.

Chapter 7 Conclusion, concludes the results of the thesis work.

2
Theory

This chapter gives a theoretical background regarding DoE as well as fundamentals, aspects and features of GPR. The chapter also explains the theoretical idea behind the implemented online refinement, and how the results of the GPR are represented. Finally a demonstration of GPR is presented.

2.1 Design of Experiments

2.1.1 Objective of Design of Experiments

A trained model can only represent the system behaviour provided by the information within a training dataset. Thus, a DoE is necessary in the first place for a proper covering or exploration of the system's input space. A space filling design is used for the DoE, see Section 2.1.3. A motivation for using a space filling design is described in Section 2.1.2. Details regarding implementation of the DoE are described in Section 3.3.

For automotive systems, measuring often requires the usage of test cells which leads to expensive measuring costs, thus the key concept of DoE schemes is to gather as much information about the input-output behaviour using as few experiments as possible.

However, optimally distributed training data strongly depend on the used regression algorithm for modeling after the data is collected. Hence, the type of model should be clear, before generating a DoE plan [12].

2.1.2 Types of DoE

Generally, DoE methods can be divided into model-based and model-free approaches (the latter is used in the thesis). If the model and its structure is known, so that a model-based approach is suitable, statistics can be used to define an optimal DoE plan, thus reducing the measuring effort. However, in this thesis no prior knowledge of the model is given and therefore a model-free DoE approach is used. Furthermore, a so-called non parametric model will be used, where the distribution of data is modelled instead of assuming a specific model structure. In essence, model-free DoE implies a space filling design, which generally aims for covering the whole input space uniformly by maximization of the minimal distance between each input point [12].

A nonlinear modeling technique as a GPR (see Section 2.2) is suitable in engine calibration since it can adapt the degree of nonlinearity itself and because it does not make as strict assumptions as a linear model with a fixed structure. Hence, it is meaningful to use a measurement design which does not make strict assumptions on the engine behavior as well, such as space fillings designs. A problem with optimal DoE plans is that they are optimal only for a specific linear model and this model has to be defined a priori before measurements in the test cell. Typically, a change of the structure of the linear model, e.g., a change of the degree of the polynomial model, is critical, and, in the worst case, not possible with such a design. Consequently, if the a priori assumptions differ from the real engine behavior, then the areas, in which the measurements are placed, are not optimal, and, in the worst case, new measurements have to be made, which is obviously time- and cost-intensive [7].

2.1.3 Latin Hypercube Sampling

The chosen space filling design is a Latin Hypercube Sampling (LHS), since it guarantees an even distribution of measurements in the input space. LHS is used in generating random sample values from a multi-dimensional distribution. LHS is the most commonly used random sampling technique for Monte-Carlo methods. Monte-Carlo methods are a computational algorithm used in different fields of engineering and relies on random sampling in order to solve a numerical problem. [13]

In LHS DoE, the design space is subdivided into an orthogonal grid which has N elements of the same length as the number of parameters. Within this multi-dimensional grid, N sub-volumes are chosen so that in each row and column of the grid only one sub-volume is selected. In each of the sub-volume, a random sampling is conducted. When selecting sub-volumes it should be done avoiding any non-existing correlations between dimensions and making sure that the samples are spread along the design space diagonal satisfying the requirements of a LHS DoE. There are different techniques employed when conducting a LHS DoE in order to minimize any non-significant correlations. Further reading about the theoretical aspect of LHS can be done on [14]. The implemented LHS DoE is described in Section 3.3.

2.2 Gaussian Process Regression

In order to explain the working principles of GPR, a theoretical framework regarding Bayesian inference and regression as well as Gaussian Processes needs to be established first.

2.2.1 Gaussian Process

A Gaussian Process consists of a set of random variables, where every finite collection of these variables has a multivariate normal distribution (all finite linear combinations of the variables are normally distributed). The distribution corresponding to

a Gaussian process is the joint distribution of infinitely many random variables, and is therefore a distribution over functions. These stochastic processes are defined by their mean and covariance functions,

$$\begin{aligned} k(x_m, x_n) &= \mathbb{E}[(f(x_m) - m(x_m))(f(x_n) - m(x_n))] \\ m(x) &= \mathbb{E}[f(x)] \end{aligned} \tag{2.1}$$

where $k(x_m, x_n)$ is the covariance function which captures how correlated, the input x_m are to the nearest input x_n and how fast the correlation decreases as the distance between the two points increases [15].

Covariance functions (also called kernels) are the key ingredient in using Gaussian processes. They encode most assumptions about the form of function that we are modeling. In general, covariance represents some form of distance or similarity. Consider two input points (locations) x_i and x_j with corresponding observed values y_i and y_j. If the inputs x_i and x_j are close to each other, we generally expect that y_i and y_j will be close as well and vice versa. This measure of similarity is embedded in the covariance function and is called 'smoothness' [16].

The mean function specify how the mean, $m(x) = \mathbb{E}[f(x)]$ is calculated. In most applications there is no prior knowledge about the mean behaviour and thus a popular choice is to assign $m(x) = 0$. This assignment expresses that the initial guess for the function output at any input is zero. This assignment may sound restrictive, but offsets and simple trends can be subtracted out before modeling by normalizing so a zero mean is obtained (see Section 3.2.3), hence in practice it is not so restrictive. Consequently, the covariance function is then of greater importance than the mean function and is what really affects regression [17],[18].
In the implemented GPR, the mean function is assigned according to: $m(x) = 0$, which in summary simplifies the modelling.

A Gaussian Process is denoted as:

$$f(x) \sim GP(m(x), k(x_m, x_n)) \tag{2.2}$$

A simple example of a Gaussian Process is the Bayesian linear regression model $f(x) = \phi(x)^T \mathrm{w}$ with prior $\mathrm{w} \sim \mathcal{N}(0, \Sigma_p)$. For the mean and covariance function, the following is obtained:

$$\begin{aligned} \mathbb{E}[f(x)] &= \phi(x)^T \mathbb{E}[\mathrm{w}] = 0 \\ \mathbb{E}[f(x_m) f(x_n)] &= \phi(x)^T \mathbb{E}[\mathrm{w}\mathrm{w}^T] \phi(x_n) = \phi(x_m)^T \Sigma_p \phi(x_n) \end{aligned} \tag{2.3}$$

thus $f(x_m)$ and $f(x_n)$ are jointly Gaussian distributed with zero mean and covariance given by $\phi(x_m)^T \Sigma_p \phi(x_n)$. A commonly used covariance function is the Squared Exponential function (SE), which is described in (2.4).
It can be shown that the SE corresponds to a Bayesian linear regression model with an infinite number of basis functions. This theoretical result motivates the popular use of the SE and also illuminate the connection between a linear regression and a regression based on a Gaussian Process [15].

2.2.2 Modeling via concept of smoothness

In Figure 2.1 a theoretical example illustrates the core idea behind a GPR. The kernel can produce numerous functions because of infinite combinations of different values for the hyperparameters (see Section 2.2.3) inherent for the used kernel and this will affect the assumed smoothness of the function. The prior functions are shown in the left. An infinite amount of prior functions are available, but for practical reasons only a finite amount is shown here. After obtaining data, the posterior probability can be calculated and as can be seen in the middle plot, the number of probable functions decreases when the posterior is calculated.

A characteristic property of smoothness is that the uncertainty (2.12) increases as the distance increases between each data point in the input domain. This is reasonable, as a smooth function has very similar function values that are very close to each other in the input domain, or in other words, they are highly correlated. Respectively, function values are expected to be less similar the further away the data points are in the input domain. To the right in Figure 2.1 the uncertainty estimates are displayed for the predictions, these uncertainty estimates are described in Section 2.2.7. A question that then arise is, how smooth is the true function? That is what a GPR tries to answer in order to fit the data accurately.

The choice of covariance function is often based on some known property of the function data. If the data is smooth then (2.4) is a suitable choice. If the data is periodic, a periodic covariance function is best and if the data is very irregular there are covariance functions that are more suited for this behaviour as well. For a more detailed treatment regarding covariance functions see [15].

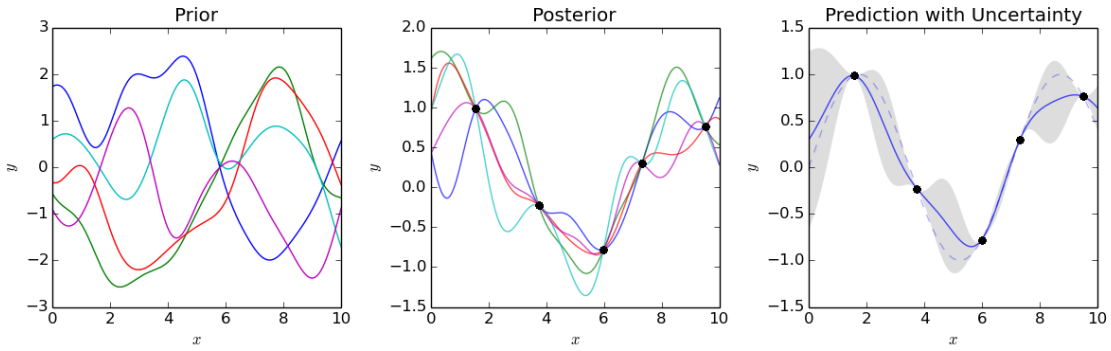

Figure 2.1: To the left five examples of prior functions are shown. On the x-axis is the input and on the y-axis is the output or the function value. (Figure reference: Wikimedia Commons, `https://commons.wikimedia.org/wiki/File:Gaussian_Process_Regression.png`, 2018-07-12)

In the middle of Figure 2.1 the predictions after obtaining training data (black dots), to the right the predictions are shown with corresponding uncertainty estimates.

2.2.3 Hyperparameters

A very popular kernel due to its flexibility is the Squared-Exponential (SE), which is defined by:

$$k_{SE}(x_m, x_n) = \sigma^2 \exp\left(\frac{-(x_m - x_n)^2}{2l^2}\right) + \delta_{mn}\sigma^2_{noise} \tag{2.4}$$

The hyperparameters are σ, l and σ_{noise}. σ^2 is the signal variance and determines the average distance of the function away from its mean. A small value of the signal variance characterize functions that stay close to their mean value, larger values allow more variation, it can be seen as a scaling factor.

l^2 is the length scale and describes how smooth the function is. A small length-scale value implies that the function values can change quickly, hence a large value characterize functions that change slowly. For example, a sine wave of low angular frequency would have a small length-scale and sine wave with high angular frequency would have a large length-scale. This is not related to transient or stationary behaviour, it's only a description of how the function values changes. σ^2_{noise} is the noise variance and δ_{mn} is the Kronecker Delta, thus the noise variance is only applied when $m = n$. The noise variance allows GPR to model the noise present in the training data [19].

In Figure 2.2 a theoretical example illustrates the effect of the hyperparameters on the prior function.

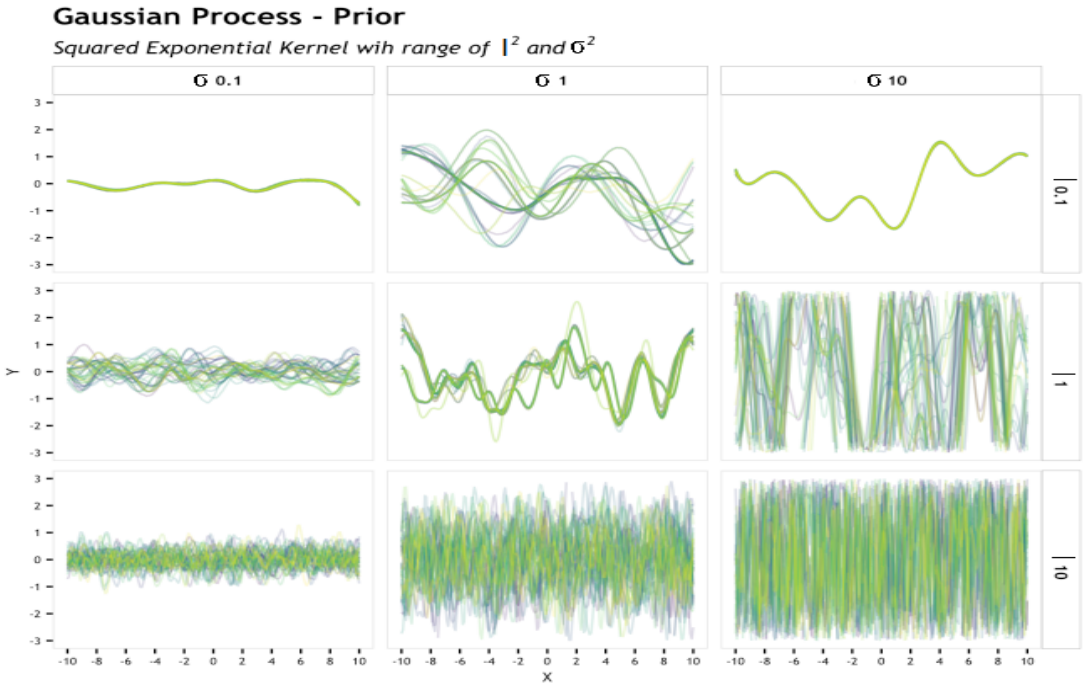

Figure 2.2: Different combinations of values for the hyperparameters. On the x-axis is the input and on the y-axis is the output or the function value. (Figure reference: `https://matthewdharris.com/2016/05/16/`, 2018-07-14)

It can be seen in Figure 2.2 that a bigger σ results in bigger amplitudes and a bigger l results in greater wiggles of the function.

2.2.4 Training

The training of a GPR is based on selecting the optimal model structure according to Bayesian inference. The inference consists of several levels. The lowest level consists of parameters, for instance the weights in a Neural Network. The second level is related to the hyperparameters, which affects the distribution of parameters at the first level. An example of the second level is the occurrence of weight decay in a Neural Network. Finally, the top level consists of a set of possible model structures [15]. To execute the complete inference, posterior probabilities are calculated for each level at a time.

The training involves a non-convex optimization task in minimizing the marginal likelihood and usually gradient based optimization techniques are used. A suggested technique is the iterative method of Conjugate Gradients (see [20]). In the implemented GPR, the method of Conjugate Gradients is used. An interesting trait of the marginal likelihood is that it incorporates a trade-off between model complexity and fit. Because of this reason, the marginal likelihood is of great importance for the training of GPR. For more information regarding training of GPR, see `http://www.gaussianprocess.org/gpml/chapters/RW5.pdf`.

2.2.5 Automatic Relevance Determination

The SE kernel (2.4) can easily be adjusted to include several input variables in the following manner:

$$k_{SE-ARD}(x_m, x_n) = \sigma^2 \exp\left(\frac{-\sum_{a=1}^{D}(x_{ma} - x_{na})^2}{2l_a^2}\right) + \delta_{mn}\sigma_{noise}^2 \tag{2.5}$$

where D is the number of input variables. The terms l_a^2 are still length scales and σ and σ_{noise} are signal variance and noise variance as in (2.4). If a particular l_a^2 becomes high, the function becomes relatively insensitive to the corresponding input variable x_{ma}. Hence, it becomes possible to detect input variables that have little or no effect on the regression (and vice versa) by looking at the absolute value of l_a. A smaller absolute value of l_a indicate an input variable that is of importance and a greater absolute value of l_a indicate an input variable that is less important. This determination of the importance of a certain input is called Automatic Relevance Determination (ARD), SE with ARD is abbreviated as SE-ARD [21]. In the implemented GPR, SE-ARD is used due to its flexibility.

2.2.6 Prediction

After the GPR is trained, predictions are made. The joint distribution of the training outputs Y and the function values f_* at test locations under the prior (kernel) is

given by:

$$\begin{bmatrix} Y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \sigma^2_{N_{train}} I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right) \tag{2.6}$$

where X_* contains all test inputs. $K(X,X)$ is the resulting covariance matrix consisting of all covariances between training inputs, this covariance matrix is generated by (2.5) and has the following form:

$$K(X,X) = \begin{bmatrix} k_{11} & \dots & k_{1N_{train}} \\ \vdots & \ddots & \vdots \\ k_{N_{train}1} & \dots & k_{N_{train}N_{train}} \end{bmatrix} \tag{2.7}$$

If the training inputs are arranged in ascending order with respect to the distance to the first training input, then k_{N1} and k_{1N} will both tend to zero as this represent the idea of smoothness, see Section 2.2.1. k_{11} and k_{NN} are self correlations and thus never zero. $K(X,X_*)$ is the resulting covariance matrix consisting of all covariances between training and test inputs (note that $K(X,X_*) = K(X_*,X)$ caused by definition of a covariance matrix. This covariance is as well generated by (2.5) and has the following form:

$$K(X,X_*) = \begin{bmatrix} k_{11_*} & \dots & k_{1N_*} \\ \vdots & & \vdots \\ k_{N1_*} & \dots & k_{NN_*} \end{bmatrix} \tag{2.8}$$

$K(X_*,X_*)$ is the resulting covariance matrix of all covariances between test inputs and has in principle the same form as (2.7) [15].

The term $\sigma^2_{N_{train}} I$ in (2.6) represents the noise term on the training inputs, the noise is only present on the self correlations and this can be represented by multiplying the noise term with an identity matrix, I.

The predictive equations are given by:

$$\bar{f}_* \triangleq \mathbb{E}[f_*|X,Y,X_*] = K(X_*,X)[K(X,X) + \sigma^2_{N_train} I]^{-1} Y \tag{2.9}$$

$$cov(f_*) = K(X_*,X_*) - K(X_*,X)[K(X,X) + \sigma^2_{N_{train}} I]^{-1} K(X,X_*) \tag{2.10}$$

(2.9-2.10) can be written in a more compact form by introducing $K = K(X,X)$ and $K_* = K(X,X_*)$ and only consider the case when there is only one test input x_*, let $k(x_*) = k_*$ to denote the covariance between the test input and the N_{train} training points. By this compact notation, the following is obtained:

$$\bar{f}_* = k_*^T (K + \sigma^2_{N_{train}} I)^{-1} Y \tag{2.11}$$

$$V[f_*] = k(x_*,x_*) - k_*^T (K + \sigma^2_{Ntrain} I)^{-1} k_* \tag{2.12}$$

where $V[f_*]$ is the predicted variance of the function value f_*. It can be noted that the predicted mean $\bar{f}_*$ can be seen as a linear combination of kernels according to:

$$\bar{f}(x_*) = \sum_{i=1}^{N_{train}} \alpha_i k(x_i, x_*) \tag{2.13}$$

where $\alpha = (K + \sigma^2_{N_{train}} I)^{-1} Y$ [15].

2.2.7 Uncertainty Estimates

One attractive feature of the GPR is the ability to obtain uncertainty estimates of the regression model. These estimates comes from (2.12). Through (2.12-2.13) a 95% confidence interval (CI), can be created via the predicted variances according to [22],[23]:

$$CI = \bar{f}_* \pm 1.96\sqrt{V[f_*]} \tag{2.14}$$

The appearance of the confidence interval from 2.14 can vary depending if noise is present or not. When no noise is present, the prediction variances are thus very small where measurements are taken. When noise is present, an uncertainty always exist and consequently the confidence interval is on average broader than the noise-free case.

2.2.8 Prediction Algorithm

Now it is possible to present the algorithm for calculating predictions via GPR. In Figure 2.3 the algorithm is presented. In order to execute the algorithm some objects are needed. These are training inputs, training outputs, a chosen covariance function, a predicted noise level (noise variance) and test inputs. If these objects are provided, the GPR can be trained and afterwards predictions can be made according to steps 1-7.

In (2.11) it can clearly be seen that a matrix inversion is required to make predictions. This matrix inversion is numerically cumbersome and therefore another alternative is sought after that improves numerical stability and efficiency. A good alternative is the Cholesky Decomposition, see Section A.1.

ALGORITHM: PREDICTIONS FOR GAUSSIAN PROCESS REGRESSION

Input: X(training inputs), Y(training outputs), k(covariance function), $\sigma^2_{N_{train}}$(noise level), x_*(test input)

1. L=Cholesky$(K + \sigma^2_{N_{train}} I)$
2. $\alpha = L^T \backslash (L \backslash Y)$
3. $\bar{f}_* = k_*^T \alpha$
4. $v = L \backslash k_*$
5. $V[f_*] = k(x_*, x_*) - v^T v$
6. $\log(p(Y|X)) = -\frac{1}{2} Y^T \alpha - \sum_i \log(L_{ii}) - \frac{n}{2} \log(2\pi)$
7. **return:** $\bar{f}_x$ (mean), $V[f_*]$(variance), $\log(p(Y|X,\theta))$ (log marginal likelihood)

Figure 2.3: Algorithm for predictions using GPR.

It is assumed in Figure 2.3 that the GPR is already trained before step 1. The implementation in the algorithm addresses the issue with the matrix inversion described in (2.12) and (2.13) by using a Cholesky decomposition (see Section A.1), instead of directly inverting the matrices. The notation on line 4 denotes that v is the solution to the system of linear equations $Lv = k_*$ and so forth. For multiple test inputs, lines 3-5 are repeated.

2.3 Online Refinement

The information from (2.14) can be used to construct uncertainty estimates in forms of confidence intervals. These estimates can then be used to determine where to place additional measurements in order to improve the model prediction according to specifications mentioned in Section 1.5. The extra measurements should be placed in the regions where the confidence interval is largest in width, here the uncertainty of the model prediction is the biggest. The extra measurements are called Update Points (UPs). In this way, a procedure can be developed that integrates modeling and taking measurements in an online fashion. The suggested procedure is described in Section 3.5.

2.4 Representation of Results

Here it is explained how the result of the GPR is represented.

2.4.1 Regression Plots

The regression data is split into training and test data. This is described in further detail in Section 3.2.2. The training and test data corresponds to two sets of inputs and outputs. The training data is denoted as: $(x_i, y_i), i = 1, ..., N_{train}$ and test data as $(x_j, y_j), j = 1, ..., N_*$. The prediction from GPR is denoted as: $\hat{y} = GPR(x_j)$ where the prediction can be seen as a function of the test input where the function is the GPR.
The test data is unseen data for the GPR during training, and thus constitute a reasonable challenge for the GPR to be used as a measure of performance. Only the training data is used to train the GPR. If the GPR is successful, the predictions should approximately be equal to the test outputs y_j, more equal the better. In Figure 2.4 this information is shown in a regression plot.

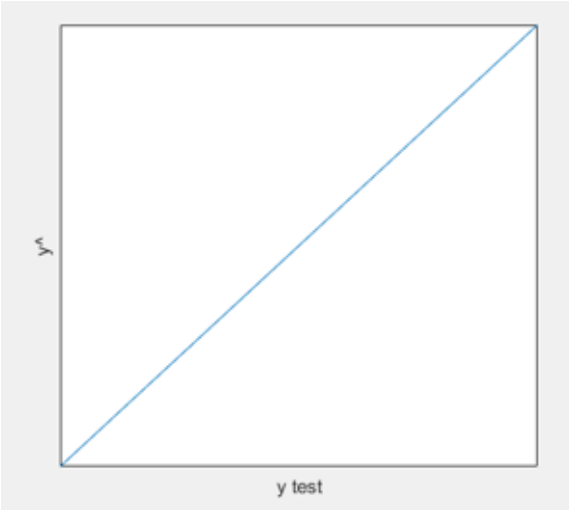

Figure 2.4: Regressionplot, the test output is on the x-axis, on the y-axis is the prediction from GPR. This would be an ideal case where the predictions are exactly equal to the test output.

It can be seen from Figure 2.4 it represents an ideal case where the regression is perfect. The line that fits the predictions would have a fit of one.

2.4.2 Mean Squared Prediction Error

Another measure of the regression results is the Mean Squared Prediction Error (MSPE), the MSPE is defined as:

$$MSPE = \frac{1}{N_*} \sum_{j=1}^{N_*} (\hat{y}_j - y_j)^2 \tag{2.15}$$

where N_* is the number of samples of the test set. $\hat{y}_j$ is the prediction from the GPR and y_j is the test data output.

The regression is deemed better the lower the MSPE is, which indicates that the predictions and the test output are on average similar [25].
A regression is deemed successful if the fit of the regression plot is high and the MSPE is low. It is difficult to say if the MSPE is low or not, however by comparing the MSPE with respect to the MSPE of the training data (which would always be lower), one gets an idea of the accuracy.

2.4.3 Demonstration of GPR

A short demonstration is presented that illustrates how a GPR works together with an online refinement according to the overall mentioned procedure as a proof of concept.

Consider the case where the function to be regressed is $y = \cos(x)^2$,$x \in [0, 4\pi]$, 250 points are evenly taken from the interval. The function is also affected by an additive stochastic noise with a standard deviation of 0.2. The function is plotted

in Figure 2.5.

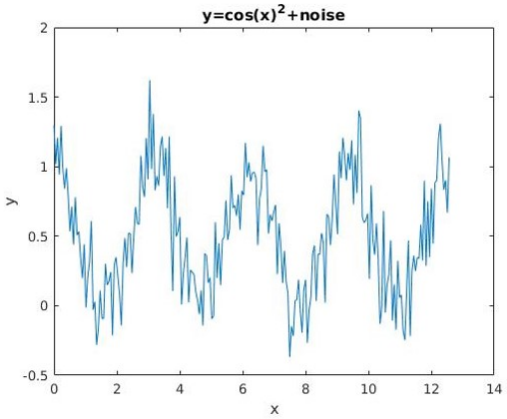

Figure 2.5: The function to be regressed together with additive stochastic noise with a standard deviation of 0.2. On the x-axis is the input and on the y-axis is the function value.

The goal of the regression is to the predict the function values despite the presence of noise.

The regression starts with a data preparation according to Section 3.2, the input and output data are both normalized according to Section 3.2.3. The reason for normalizing the inputs is because it ensures proper scalings of the covariance matrices (2.7-2.8) and also preempts numerical issues related to the inversion of the covariance matrix mentioned in Section 2.3. The reason for normalizing the output is to obtain zero mean output data, which simplifies the choice of the mean function for the Gaussian process as the mean function can reasonably be set to zero, see Section 2.2.1. The histograms of the training and test outputs are shown in Figure 2.6 to indeed show that the outputs are normalized in a way that matches the simplification regarding zero mean.

The histograms of the training and the test outputs are shown in Figure 2.6.

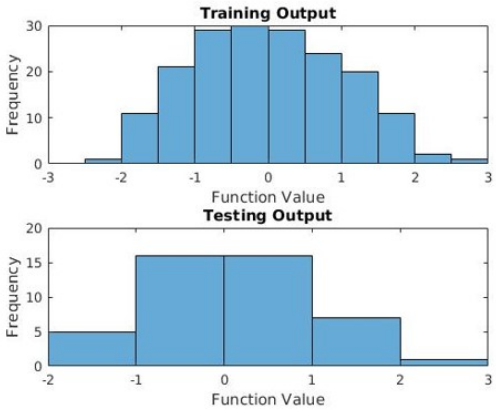

Figure 2.6: Histogram of the values for the training and test outputs. The total number of the different values (frequency) is on the y-axis and the output value on the x-axis.

The prediction from the GPR is shown in Figure 2.7, this is shown as a regression plot mentioned in Section 2.4.1. The GPR has the ability to account for noise via the right term in (2.5), this term corresponds to the standard deviation of the noise present. How the noise affects the predictions is embedded in (2.11), the estimated noise standard deviation is obtained via the training of GPR.

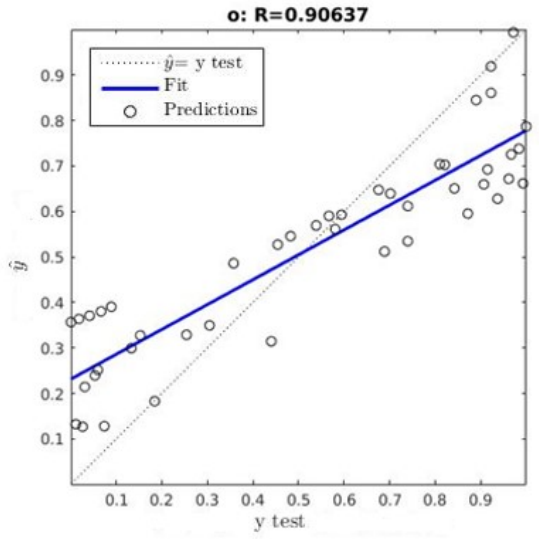

Figure 2.7: Predictions from the GPR, test outputs (without noise) are on the x-axis and predicted outputs on the y-axis.

In Figure 2.7 R is the slope of the resulting line that tries to fit all points. The circles are the predictions. The MSPE is 0.0354. The estimated noise standard deviation is 0.95.

After performing an online refinement according to Section 2.3, the new predictions are shown in Figure 2.8. 50 UPs are added. The exact procedure of the online refinement has been described in Section 3.5.

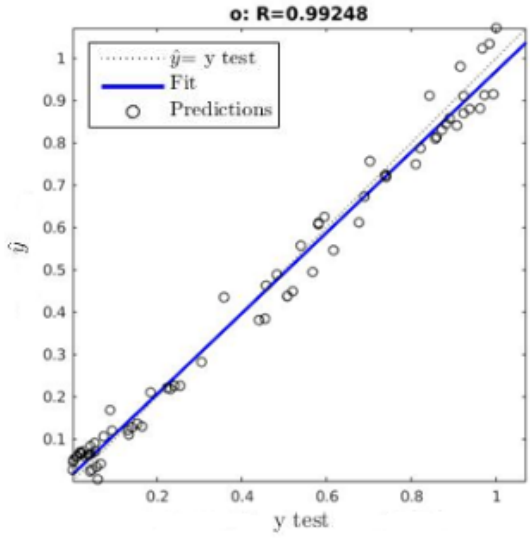

Figure 2.8: Predictions from the GPR after adding UPS, test outputs (without noise) are on the x-axis and predicted outputs on the y-axis.

In Figure 2.8 R is the slope of the resulting line that tries to fit all points. The circles are the predictions. The MSPE is 0.0021. The estimated noise standard deviation is 0.22.

It is seen that the predictions after adding UPs have improved and the estimated noise standard deviation is close to the real value (0.2).

In order to more clearly illustrate how the predictions relate to the test targets, a histogram of the residuals (predictions-test target) is plotted. The residuals should on average be zero, if the predictions are good. A histogram of the residuals (prediction-test target) is shown in Figure 2.9.

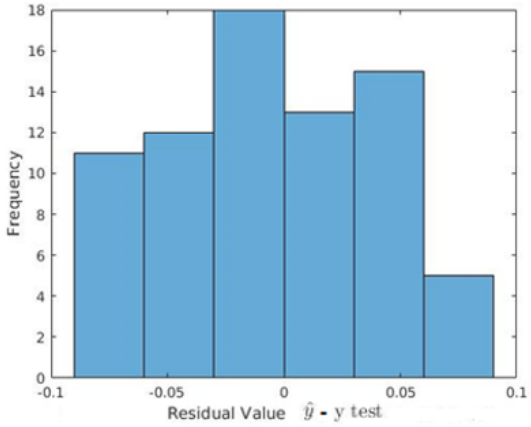

Figure 2.9: The number of how frequent each residual value is occurring (frequency) on the y-axis and the residual value on the x-axis. The residual is the prediction-test output in scaled units.

The uncertainty estimates from (2.12) and (2.13) can be used to get information about the uncertainty of the regression. In Figure 2.10 confidence interval from (2.14) is created before and after the online refinement. The training points are shown and the predictions as well. The predictions are interpolated between every predicted value. The appearance of the confidence intervals are discussed in Section 2.2.7.

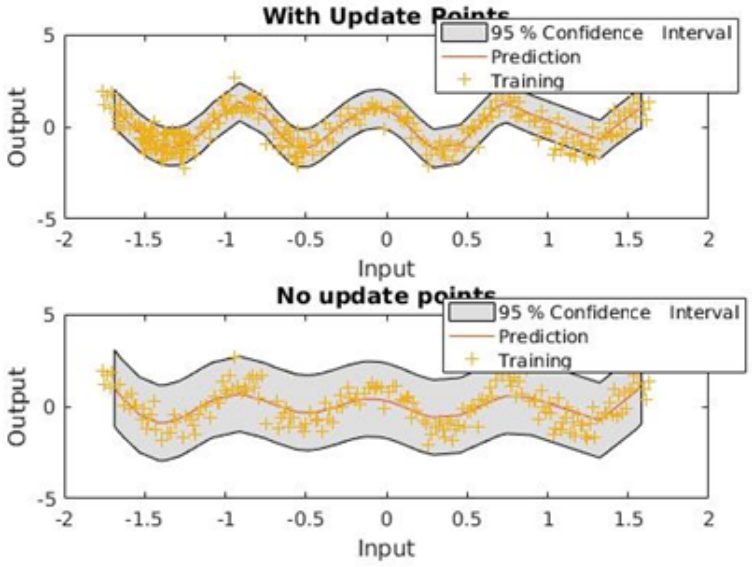

Figure 2.10: The confidence intervals generated from the GPR. On top is the confidence interval after the online refinement and on bottom is the confidence interval before. Training points are shown along predictions. On the x-axis is the input and on the y-axis is the output or function value.

It can be seen from Figure 2.10 that the confidence interval is more narrow in width after adding UPs, this is as desired according to Section 2.3. The UPs are placed in the regions where the confidence interval is the broadest. It can also be seen that the predicted function looks a bit different after adding UPs.

The determined UPs are shown in Figure 2.11 which corresponds to the regions with the greatest uncertainty (broadest confidence intervals). The confidence intervals appear to be constant in width, this is because the prediction variances did not differ much in numerical values. However a difference still exists which GPR acts upon.

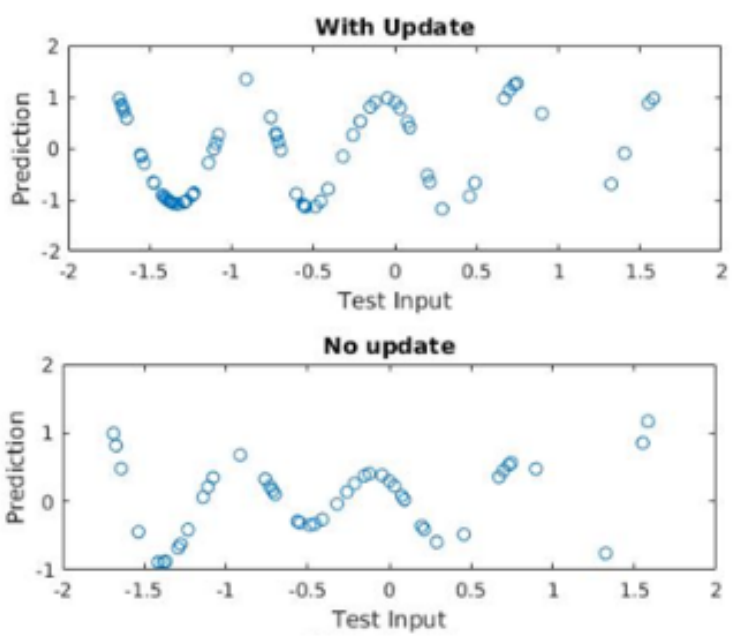

Figure 2.11: Illustration of the placement of the UPs. The test inputs on the x-axis and the predictions on the y-axis.

The UPs are indicated where more test inputs are placed, these regions are visible by an increased color density (blue) in Figure 2.11. The rest of the UPs are used as training data.

Finally, the comparison between the noise free function and the predictions is shown in Figure 2.12.

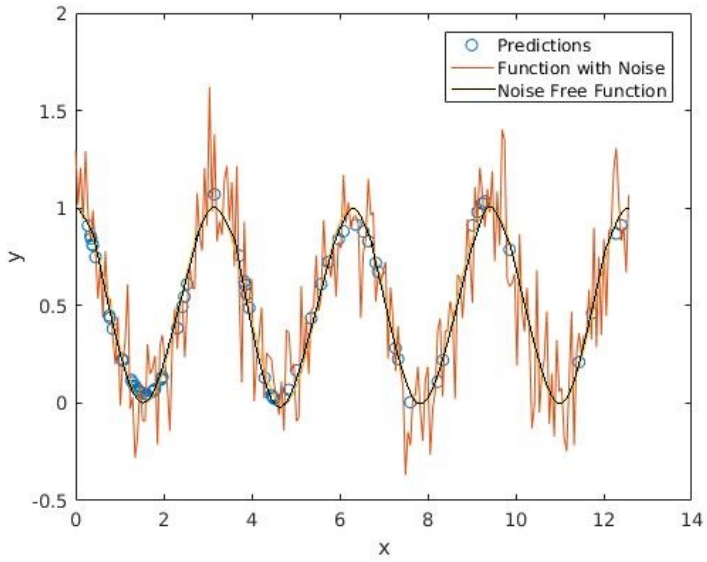

Figure 2.12: Comparison between the noise free function and the predictions. On the x-axis is the input and on the y-axis is the output or function value. The function is also shown with noise.

3
Methods

This chapter presents the methodology used in the thesis. An insight into the given data and how it is prepared to conduct the GPR is explained first.

3.1 Data

The dataset used is analyzed further in this section.

3.1.1 Operating Points

An OP is a point in the operating region defined by its engine speed and engine torque. Since, the entire engine speed vs. engine torque map is considered in the thesis there are 220 OPs in total including the repeating points, at which the engine speed and engine torque value is repeated. In Figure 3.1 all the OPs are plotted on a Speed vs Torque map.

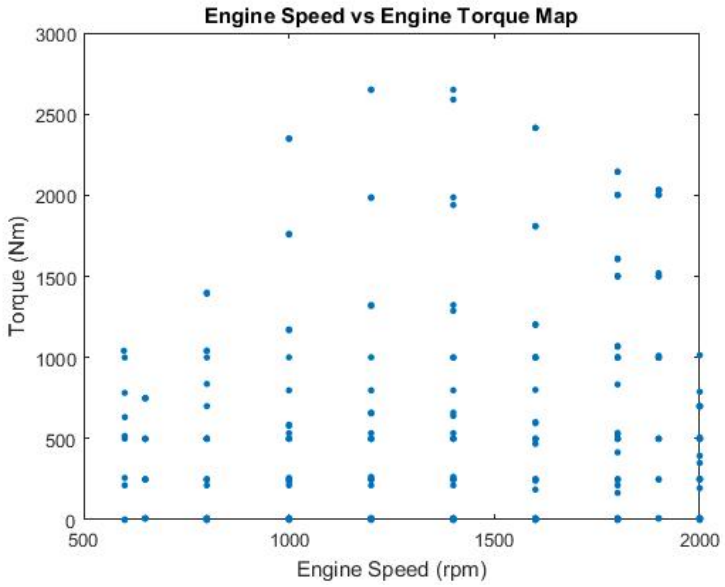

Figure 3.1: OPs on a speed vs torque map. Engine speed [rpm] is on the x-axis and engine torque [Nm] on the y-axis. There are 220 OPs in total.

The OPs can be categorized into different sections according to the torque values. When separating the OPs, three main regions are identified: Low torque region, Part

Load region and the High torque region. Local models are developed for each of these regions and the criteria for dividing into local models is described in Section 3.4.2.

3.1.2 Repeated Operating Points

There are 220 OPs in total and it can be seen in Figure 3.1 (Eg: Engine Speed 1400 rpm, Engine Torque 0 Nm; Engine Speed 600 rpm, Engine Torque 0 Nm; Engine Speed 650 rpm, Engine Torque 500 NM) that there are OPs in the dataset that contains repeated measurement values, since the counted total number of OPs in Figure 3.1 is not 220. At a repeated OP measurements are taken more than one round at that specific OP. One measurement round corresponds to the measurements that results in the data structure in Table 3.1.

The extra measurements are considered for modeling as they can provide extra information about the engine behaviour at those OPs. The extra measurements also ensure a more accurate modeling of any noise present, which is embedded in the right term of (2.5).

3.1.3 Variability Requirement

In the dataset containing the OPs according to Figure 3.1, at some OPs, the allowed actuator ranges for the varying four inputs (see Table 3.1) are not able to allow any variability when using the LHS. As an example, if the EGR position is kept at either 0% or 100% at certain OPs as seen in Figure 3.2, it will result in all inputs generated by LHS to be constant.

According to the discussion in Section 3.4 there needs to exist variability in the generated LHS in order for the regression method to be successful. In Figure 3.2-3.5 the input variability is shown by presenting all LHS-samples at each OP according to the data structure in Table 3.1.

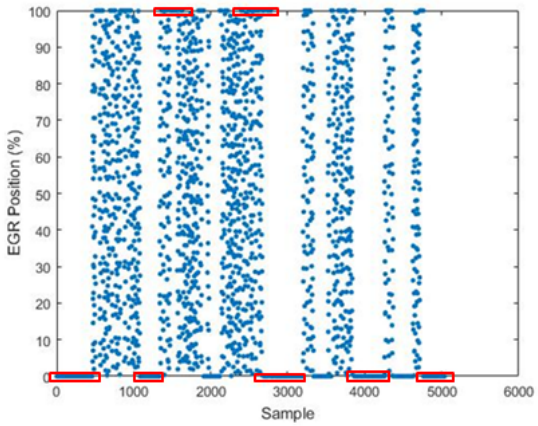

Figure 3.2: The variability of EGR position along the dataset. On the x-axis is the sample number (the corresponding row number of the LHS over all OPs). On the y-axis is the input value.

It can be seen in Figure 3.2 that there are several points with no variability along the dataset, these are the regions with flat lines. Around the first 1000 samples corresponds to the low torque region, between 1000 to around 4000 corresponds to the part load region and the rest to the high torque region. The regions are described in Section 3.4.2. The highlighted section shows the data samples with no variability.

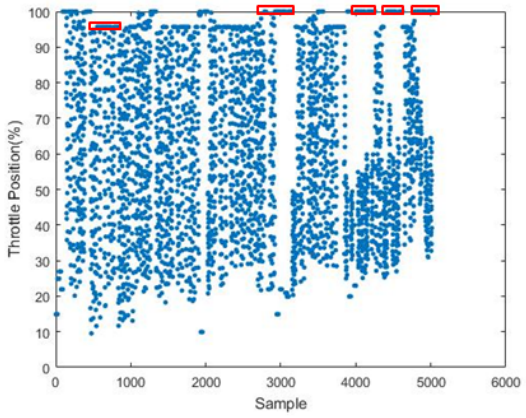

Figure 3.3: The variability of throttle position along the dataset. On the x-axis is the sample number (the corresponding row number of the LHS) over all OPs. On the y-axis is the input value.

It can also be seen in Figure 3.3 that there are several points with no variability along the dataset.

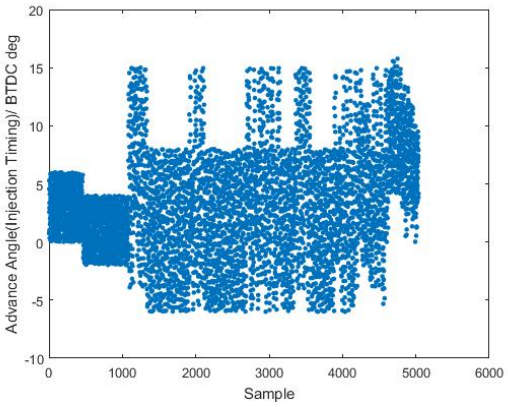

Figure 3.4: The variability of advance angle (injection timing) along the dataset. On the x-axis is the sample number (the corresponding row number of the LHS) over all OPs. On the y-axis is the input value.

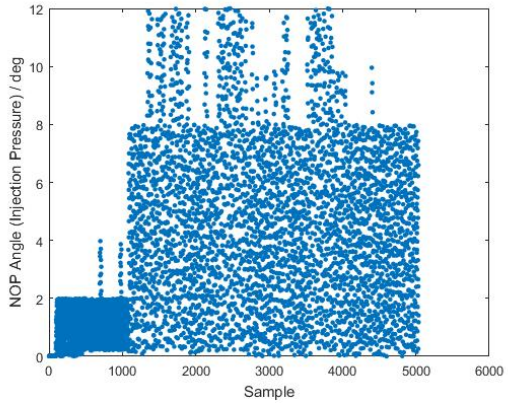

Figure 3.5: The variability of injection pressure (NOP angle) along the dataset. On the x-axis is the sample number (the corresponding row number of the LHS) over all OPs. On the y-axis is the input value.

Table 3.1 illustrates how the data was set up after the DoE.

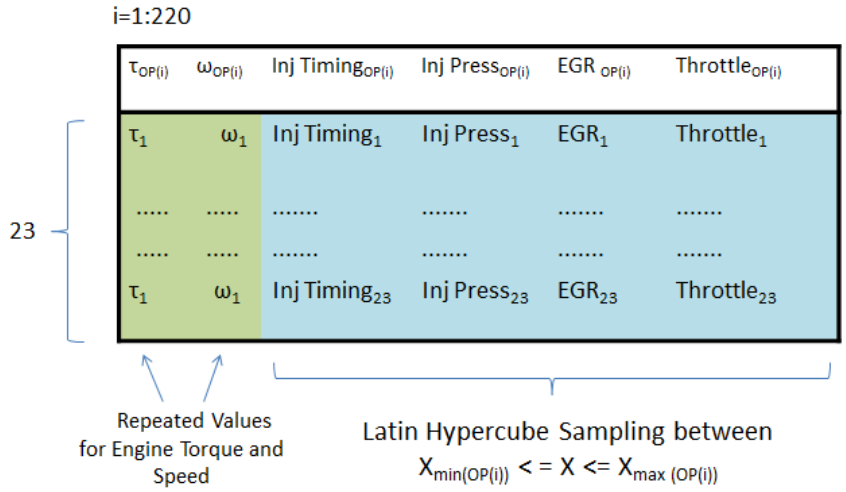

Table 3.1: Table explaining the data structure at every OP.

There are 220 OPs in total and at each of the OPs, the engine speed and torque is kept constant while the other four inputs are varied according to LHS DoE. 23 samples are taken at each OP by LHS. For every OP, there are limits for the actuators in which they are allowed to vary. This information about the actuator ranges were obtained by examining the dataset that was used to create the ANN model. The LHS is generated between these ranges for the four inputs at every OP.

The lack of input variability in the dataset at certain OPs contradicts with the concept of GPR. A theoretical explanation shed light on the issue and is explained further in Section 3.4.

The inputs need to have variability in order for the GPR to work correctly. In Figure 3.6 the flat predictions (see Section 3.4) are discussed for a small dataset example.

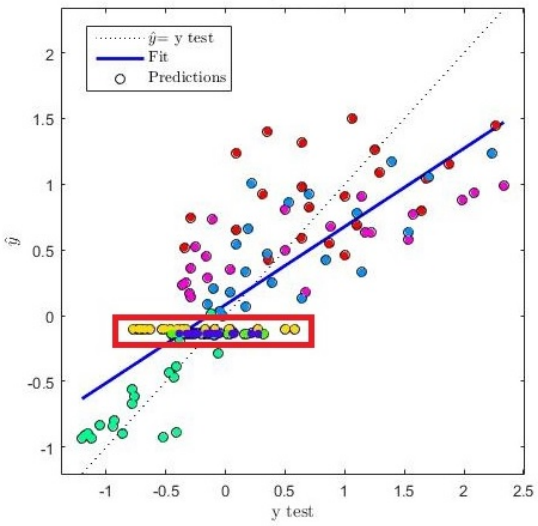

Figure 3.6: The highlighted 'flat lines' correspond to the OPs with no variability. Predictions with the same color belongs to the same OP.

When investigating further into the properties of these flat lines; the value at each of these flat lines corresponded to the mean output value at each of these OPs. Occurrence of these 'flat lines' is also a reason for the development of local models which is described in Section 3.4.2.

In order to solve this data issue, OPs which resulted in no input variability are removed. However the number of these OPs are few. In the real application this issue needs to be addressed, but due to time constraints, it was needed to omit these OPs. The treatment of these OPs will be a task for future investigations.

3.1.4 Artificial Neural Network

As mentioned in Section 1.2, an ANN model is assumed to represent the real physical engine and is used to generate output data.

The ANN was trained by the Neural Network toolbox in MATLAB. The data is split into a training (70%), test (15%) and validation (15%) set. The dataset for the ANN was provided by Volvo Penta and it was obtained from a physical test cell run. The ANN has ten hidden layers, seven inputs nodes and one output node. The inputs and outputs for the ANN are the same parameters used in generating the model using GPR.

In Figure 3.7-3.9, the performance of the ANN is represented.

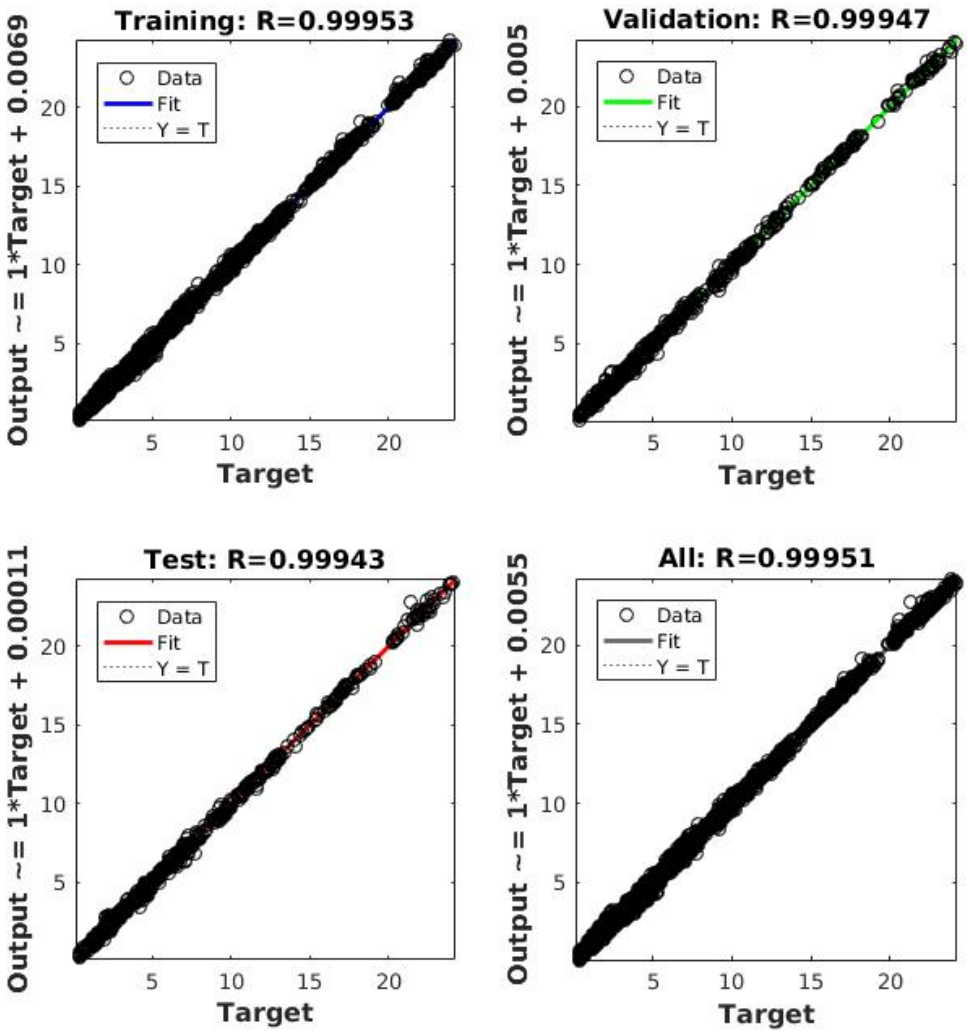

Figure 3.7: Regression plots for the training, validation, test data and all of them combined in the last plot. 'Target' on the x-axis is the known output. On the y-axis is the prediction. The resulting straight line that go through the predictions is also shown.

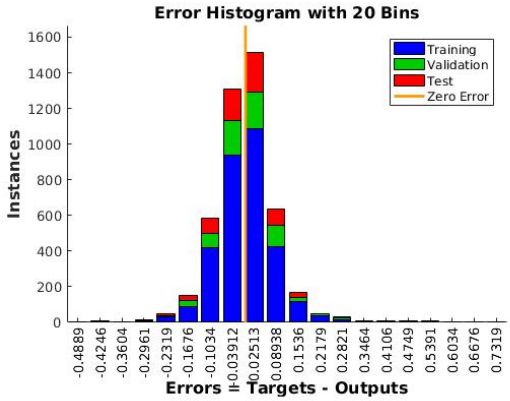

Figure 3.8: Residual histogram for the training, validation and test data.

Typically, the training set for the ANN is fixed. Training the ANN on each item of the set once is an epoch. By having 153 epochs as the case here, each element on the dataset would have 153 individual training trials when training. The algorithm used for training is the Levenberg-Marquardt.

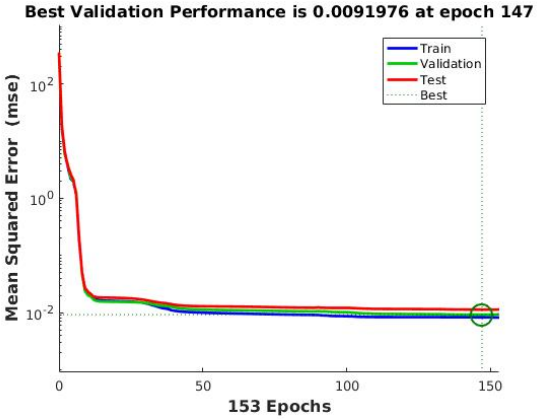

Figure 3.9: The decrease of MSE as a function of epochs for the training, validation and test set.

3.1.5 Output Data

All data originates from OPs, where the engine torque and speed are fixed, thus only fuel flow affects the BSFC (see (1.2)) at each OP. Hence, considering fuel flow in this context is equivalent to looking at BSFC at each OP. When looking at (1.2) it is seen that the denominator in the equation for BSFC consists of the engine power (product of the engine speed and torque). When considering BSFC at OPs in the low torque region (where the torque is close to zero), the BSFC consequently tend towards infinity. Hence, it is more practical to look at the fuelflow, which is in the numerator of the equation for BSFC.

In a histogram it is seen that the distribution of the data for the fuel flow is not evenly distributed, see Figure 3.10.

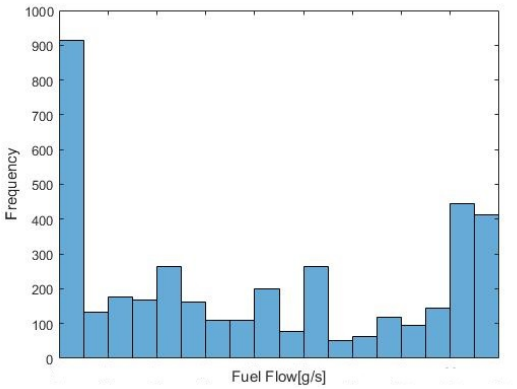

Figure 3.10: Histogram of fuel flow data. The frequency of fuel flow is on the y-axis. The range of fuel flow [g/s] is on the x-axis. The fuel flow values are hidden for confidentiality.

The distribution of fuel flow data for all LHS samples, over all OPs is illustrated in Figure 3.11.

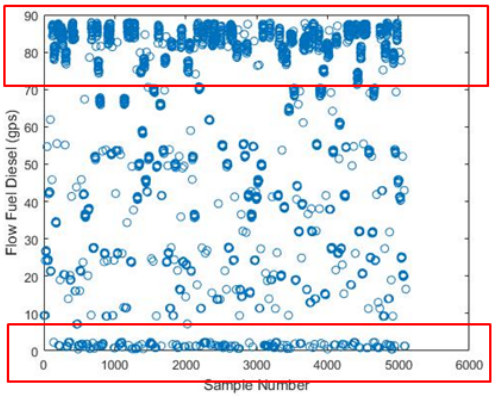

Figure 3.11: Fuel flow [g/s] is on the y-axis. On the x-axis is the sample number (the corresponding row number of the LHS) over all OPs.

The upper red box in Figure 3.11 correspond to the high torque region and the lower to the low torque region. The region between the red boxes correspond to the part load region. It can be seen from both Figure 3.10 and Figure 3.11 that there are a high frequency of fuel flow around 0-5 g/s and 80-90 g/s.

In Figure 3.12 fuel flow is plotted against engine torque.

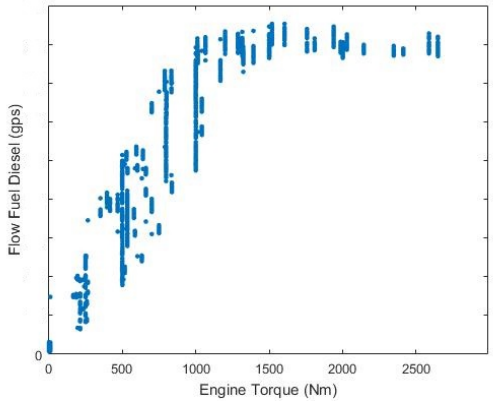

Figure 3.12: Fuel flow [g/s] on the y-axis and engine torque [Nm] on the x-axis. Vertical bars (constant engine torque) exist due to the LHS-samples for the respective OPs.Flow Fuel Diesel values are hidden for confidentiality.

The predictions over the test set for the data in Figure 3.12 is shown in Figure 4.1.

3.1.6 Data Constraints

There are certain restrictions and constraints existing in the dataset that affects the model.

The maximum number of inputs that can be considered with the current dataset is limited to 4 and at every OP, each varying input parameter; injection timing, injection pressure, throttle position and EGR valve position are bounded by a minimum and a maximum value. Hence, when generating data the LHS needs to be carried out within these ranges.

Engine speed and engine torque are constant at each OP by definition. Hence they are not taken into consideration for LHS.

As mentioned in Section 3.1.3, some input parameters do not have variability in some OPs thus creating constraints for LHS. These limitations in input data affects the GPR modeling process.

3.2 Data Preparation

The methods used for data preparation are presented. The data preparation is performed before executing the GPR.

3.2.1 Randomization

All the data (input-output) in its current sequence is randomized before conducting the GPR. The randomization shuffles the order between different OPs (not the data within each OP).

Randomization is a commonly used technique in data science and prevents bias. Randomization techniques eliminate the bias by using the probability theory [26].

It should be noted that four inputs at each OP from LHS (see Table 3.1) are already randomized. The randomization is done with a specific end goal to distribute the data throughout the whole range of the speed vs torque map in Figure 3.1. Hence, this will result in a good representation of the entire dataset in both the training and testing dataset (see Section 3.2.2). In this way the regression problem is set up in terms of an interpolation instead of a extrapolation, thus the entire region of the dataset will be modelled. This is preferred since models for entire regions are created for local models, see Section 3.4.2.

The randomization will make each training round of GPR unique, and thus its predictions. However, the predictions from the GPR will not differ significantly between training rounds. The similarity between training rounds can be seen by considering the marginal likelihood which is maximized during the training of GPR (see Section 2.2.4).

In Table 3.2 the marginal likelihoods are presented that resulted from some training rounds of the GPR for the global model (see Section 4.1.1) as an example.

Fit	Marginal Likelihood
0.99973	$6.545458 \cdot 10^3$
0.99967	$6.571981 \cdot 10^3$
0.99977	$6.516765 \cdot 10^3$

Table 3.2: The fit of different global model runs along with marginal likelihood in each of them. It is seen that the values of the marginal likelihood are very similar, implying similar predictions.

3.2.2 Division of Training/Validation Data

The randomized data is split into a training and a testing set. The training is set to 80% and the testing for 20%. The training set is denoted by: $(x_i, y_i), i = 1, ...N_{train}$ and test set by: $(x_j, y_j), j = 1, ...N_*$. 80/20 is a commonly occurring ratio and this is also referred to as the Pareto principle. Usually, it is a good starting point to use these values as the ratio. The fraction of patterns reserved for the testing set is inversely proportional to the number of free adjustable parameters. Hence, it is concluded that roughly 1/6 of the total dataset should be reserved for testing and the rest should be reserved for training [27].

3.2.3 Normalization

The normalization is performed according to:

$$y_{itrain,norm} = \frac{y_i - \mu_{y_{train}}}{s_{y_{train}}}, i = 1, ...N_{train} \tag{3.1}$$

$$y_{jtest,norm} = \frac{y_j - \mu_{y_{train}}}{s_{y_{train}}}, j = 1, ..., N_* \tag{3.2}$$

where y_{itrain} is the ith output from the training set, $\mu_{y_{train}}$ is the mean of the output data from the training set and $s_{y_{train}}$ is the standard deviation of the output data from the training set.

y_{jtest} is the jth output from the test set. Accordingly, $y_{itrain,norm}$ and $y_{jtest,norm}$ are the normalized output from respective sets at each OP.

The training and test inputs are normalized as well:

$$x_{itrain,norm} = \frac{x_i - \mu_{x_{train}}}{s_{x_{train}}}, i = 1, ...N_{train} \tag{3.3}$$

$$x_{jtest,norm} = \frac{x_j - \mu_{x_{train}}}{s_{x_{train}}}, j = 1, ..., N_* \tag{3.4}$$

where the notation is the same as (3.1-3.2) but with inputs instead of outputs.

Both the training and testing sets are normalized relative to the mean and standard deviation of the training set ($\mu_{y_{train}}, \mu_{x_{train}}$ and $s_{y_{train}}, s_{x_{train}}$). Samples from the test set are normalized relative to the mean and standard deviation of the training set, instead of the test set in order to place the samples from the test set on the same scale as the samples from the training set. Scaling samples from test set to the training set's mean and standard deviation may eliminate differences that could be present in the training and testing sets [28].
Normalization is explained further in Section 2.4.3.

3.3 Design of Experiments

For each OP of the engine speed vs. engine torque map, a number of samples from a LHS has been implemented and follows (3.5). LHS has been presented more in detail in Section 2.1.3.

$$N = \frac{N_{tot}}{N_{op}} \tag{3.5}$$

where N is the number of samples from LHS at each OP, N_{tot} is the total number of samples, and N_{OP} is the number of OP:s. The number of samples are restricted in order to achieve online capabilities. A dataset of 5000 is considered resulting in 23 samples at each of the 220 OPs. Although 23 is a coarse number for the DoE, this number was used due to practical reasons such as time and computational capability. Upon, conducting an online refinement the initial dataset is increased with the

addition of the UPs. Online Refinement is explained in Section 3.5.

When conducting the DoE, appropriate boundaries for the inputs at each OP are obtained by analyzing the data provided by Volvo Penta. In the dataset, the limits for the inputs at each OP are provided. The range is calculated by subtracting the minimum value from the maximum value at every OP for each DoE input. The LHS will be scaled at every OP according to these ranges.

The DoE plots for each of the inputs; injection timing (advance angle) Figure 3.13, injection pressure (NOP angle) Figure 3.14, EGR position Figure 3.15 and throttle position in Figure 3.16. In these plots the engine speed and torque are on the x and y-axis in order to illustrate the spread of the input values over the speed vs torque map in Figure 3.1. It can be seen from these plots that the spread is indeed space filling in the input space as the LHS is presented to be in Section (2.1.2-2.1.3).

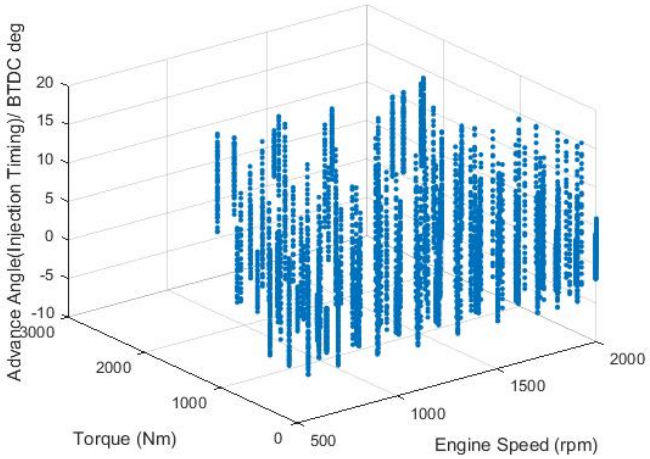

Figure 3.13: LHS DoE performed for advance angle (injection timing). On the x-axis, engine speed [rpm]. Engine torque [Nm] is on the y-axis. On the z-axis advance angle (injection timing) is represented. Each circle represents a point.

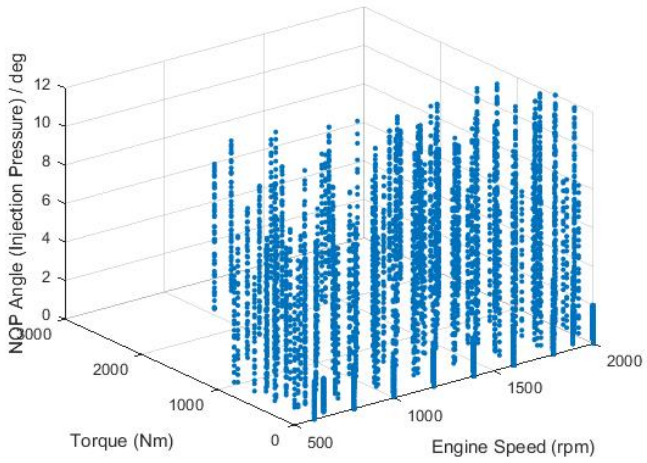

Figure 3.14: LHS DoE performed for NOP angle. On the x-axis, engine speed [rpm]. Engine torque [Nm] is on the y-axis. On the z-axis NOP angle (injection pressure) is represented. Each circle represents a point.

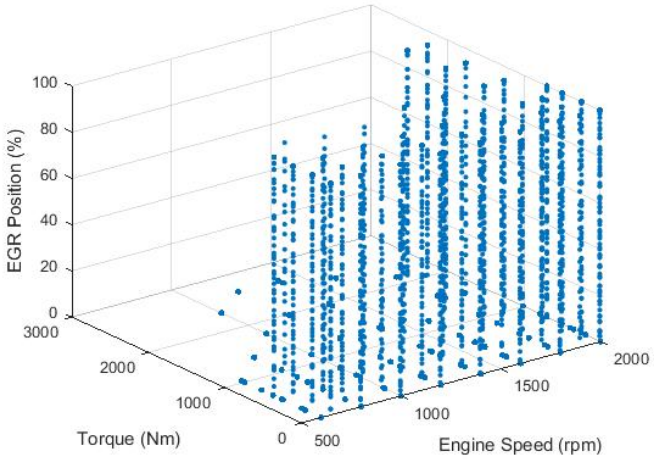

Figure 3.15: LHS DoE performed for EGR position. On the x-axis, engine speed [rpm]. Engine torque [Nm] is on the y-axis. On the z-axis EGR position is represented. Each circle represents a point.

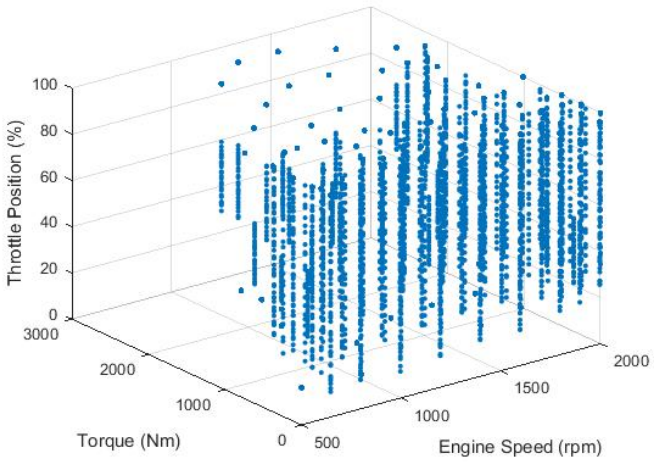

Figure 3.16: LHS DoE performed for throttle position. On the x-axis, engine speed [rpm]. Engine torque [Nm] is on the y-axis. On the z-axis throttle position is represented. Each circle represents a point.

3.4 Gaussian Process Regression

3.4.1 Covariance Consideration

The covariance function used (2.5) assumes a smooth covariance. What is meant by smoothness is explained in Section 2.2.1. A disadvantage with the usage of a smooth covariance function although commonly used, is that the resulting covariance matrix (2.7) is usually poorly conditioned, which therefore tends to make the covariance matrix non-invertible.

A smooth covariance requires the two nearby points in range to be strongly correlated. Thus, the row/columnn in the covariance matrix (2.7) corresponding to the point will tend towards autocovariance. If the two points become gradually closer, the covariance matrix (2.7) will tend towards singularity.

The predictions from the GPR is dependent on the matrix containing the covariance between training and test data as well as the inverse of the covariance matrix of training data (see (2.11)). If these covariance matrices (2.7-2.8) are close to being singular, the predictions will be constant or flat, (see Figure 3.6).

However, a smooth covariance can be treated in a way that greatly increases the numerical stability see Figure (2.3). Instead of calculating the inverse of the covariance matrix of training data, the computation of its Cholesky Decomposition is performed (see Section A.1).

3.4.2 Local Models

It is concluded that the values (see Table 3.3 and Figure 3.10, 3.12) in between 0-5g/s corresponded to OPs at low engine torque and in between 80-90g/s correspond to high engine torque. In Figure 3.12 the fuel flow is plotted against the engine torque and the existence of the regions with high and low engine torque can be distinguished according to Table 3.3.

This insight motivated the investigation of three different local models, for the low and high engine torques and for the part load region.

The three local models are based on different engine torque regions as it is mentioned in Section 3.1.1. These local models are developed due to prediction clusters (see Section 4.1.1) existing in certain regions of the global model. Table 3.3 represents the criteria upon which the local models are created.

Local Model	Engine Speed [rpm]	Engine Torque [Nm]	No. of OPs
Low Torque	600-2000	0 - 8.8	47
Part Load	600-2000	165-1400	138
High Torque	1000-1900	1500-2650	35

Table 3.3: Criteria for local models.

The three local models are created based on torque characteristics. The distribution of OPs in each local model is not uniform and as it could be seen more OPs are present in the part load region.

3.5 Online Refinement

3.5.1 Algorithm

The purpose of an online refinement is to improve the accuracy of regression by usage of additional measurements. For more information see Section 2.3. The improvement of the model by Online Refinement is represented by the increase of the fit of the model and by the decrease of the MSPE of the model. In Figure 3.17 the implemented algorithm for Online Refinement is presented.

ALGORITHM: ONLINE REFINEMENT

1. Select the M output predictions with the highest predicted variance according to (2.12). The overall process that generates predictions is described in Section 2.3.
2. Identify the inputs that correspond to these predictions.
3. A linear interpolation is conducted in between data samples with highest predicted variance in order to obtain more measurements. – Update Points (UPs).
4. Add the UPs to input data and obtain new output data.
5. Train and run the GPR for the new dataset, the newly formed input and output data is split into training and test data according to Section 3.2.2.

During refinement, the new data set including the UPs is randomized again between OPs and then divided into training and testing dataset.

Figure 3.17: Algorithm 2. Description of the implemented algorithm for online refinement.

4
Results and Discussion

This chapter presents the results and findings of the thesis. A discussion explaining the reasons for the results is also included. It is then followed by the results from global and local models. Findings specific to the GPR method are presented finally.

4.1 Gaussian Process Regression

4.1.1 Global Model

As mentioned in Chapter 3 under Section 3.2, the original dataset is prepared before running the GPR. As it is mentioned earlier in Section 3.1.3, OPs with no variability are removed. Also, as discussed in Section 3.2, 6 input parameters have been considered for the final model. A justification for the use of 6 inputs is given in Section 4.3. The GPR model is constructed by having BSFC as output and engine speed, engine torque, injection pressure (NOP angle), injection timing (advance angle), EGR position and throttle position as inputs. The regression plots (see Section 2.4.1) are shown below.

An important factor to note here is that due to the complexity of the model by having multiple inputs, representation of the results from online refinement using confidence interval is not possible as in Figure 2.10. Although, confidence intervals is used as a method of representing results in the demonstration (see Section 2.4.3), it's not possible here since the confidence interval should be plotted against all inputs and this would result in a seven-dimensional plot (6 inputs and 1 output). Thus, in Figure 4.2 prediction variance is shown instead in order to represent the uncertainty estimate of the global model.

Figure 4.1 presents the predicted fuel flow values by GPR. It should be noted that only 20% of the dataset is considered for testing and 80% for training (see Section 3.2.2). Hence only the test set of Figure 3.12 is predicted.

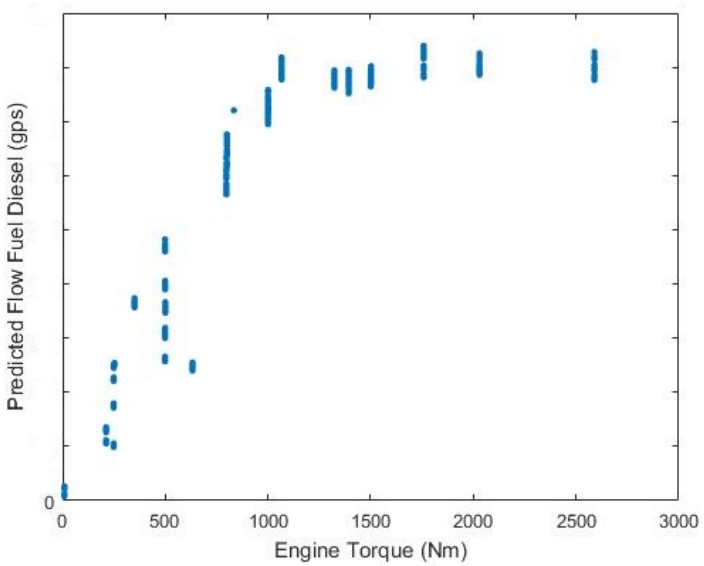

Figure 4.1: The predicted fuel flow against engine torque. On the x-axis is engine torque [Nm] and on the y-axis is the predicted fuel flow [g/s]. Vertical bars (constant engine torque) exist due to the LHS-samples for the respective OPs.

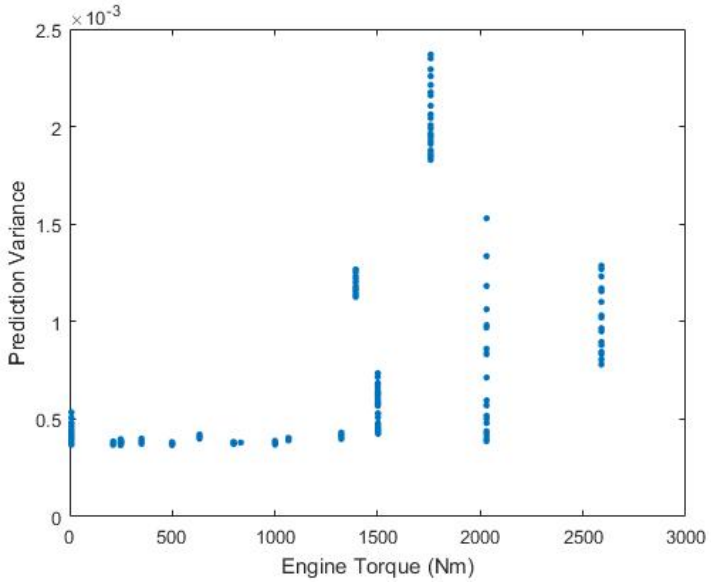

Figure 4.2: The prediction variance of fuel flow against the engine torque. On the x-axis is engine torque [Nm] and on the y-axis is the predicted variance. Vertical bars (constant engine torque) exist due to the LHS-samples for the respective OPs.

As it can be seen by comparing Figure 3.12 with Figure 4.1, the predicted fuel flow is accurate in modeling the actual fuel flow data in the regions considered. Since 20% of the entire dataset is used for testing, Figure 3.12 is not replicated entirely in Figure 4.1.

In the Figure 4.2, the variance of the predictions are shown and these values are at the highest in the high torque region (after 1500 Nm, see Section 3.4.2) and lowest in the part load region (165-1400 Nm). The small spike seen around 0 Nm and the points up to 165 Nm corresponds to the low torque region.

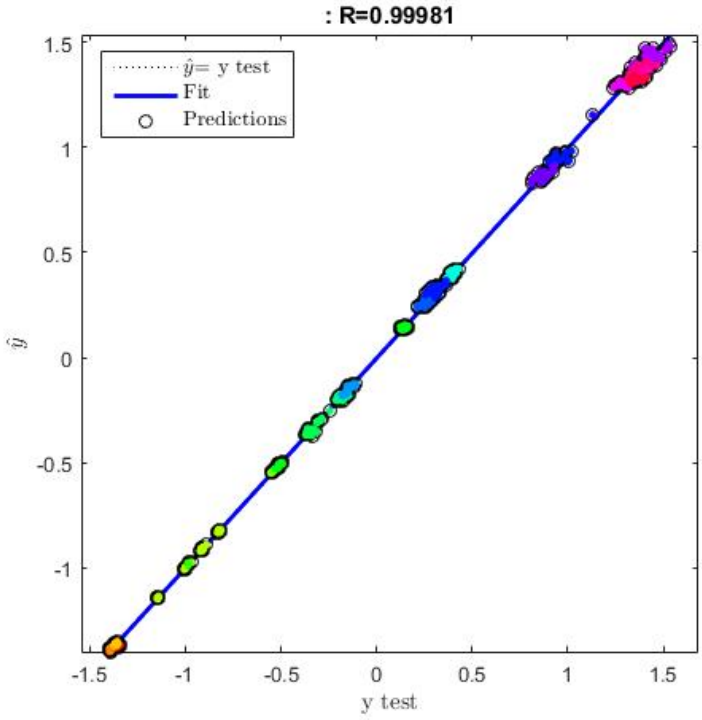

Figure 4.3: Predictions from the GPR for the initial dataset for the global model. Test outputs are on the x-axis and predicted outputs on the y-axis.

In Figure 4.3 R is the slope of resulting line that tries to fit all points. The circles are the predictions. Each colour correspond to an OP of the test dataset. The MSPE is $3.3873 \cdot 10^{-4}$.

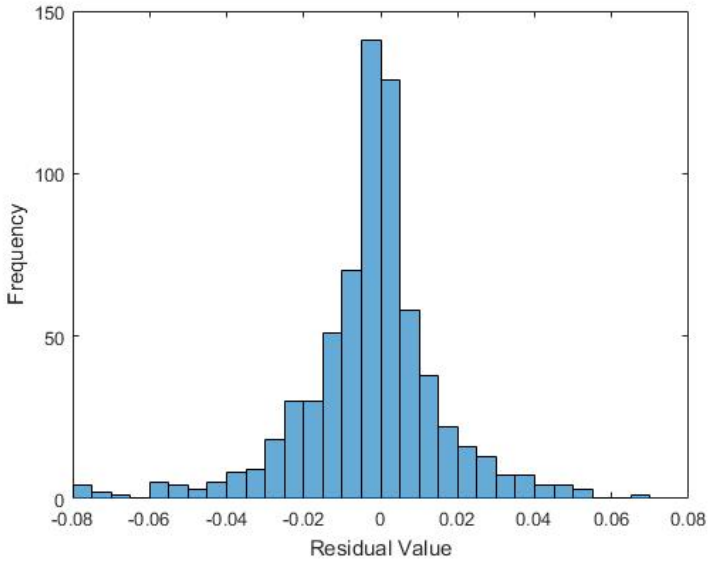

Figure 4.4: Residual plot for the global model. The number of how frequent each residual value is occurring (frequency) on the y-axis and the residual value (prediction-test output) in scaled units on the x-axis.

The predictions from the GPR is shown in Figure 4.3 and the residual plot is shown in Figure 4.4. In the residual plot it can be seen that the errors are mostly centered around small values, which is positive.
Since, the entire global model is considered, 220 OPs are taken into account when constructing the model. The MSPE of BSFC (predicted-actual) for the initial dataset is $3.3873 \cdot 10^{-4}$. In order to improve the results from this model, an online refinement is carried out. The procedure of the online refinement is discussed in Section 3.5. The results from an online refinement is also discussed further below in Section 4.2.

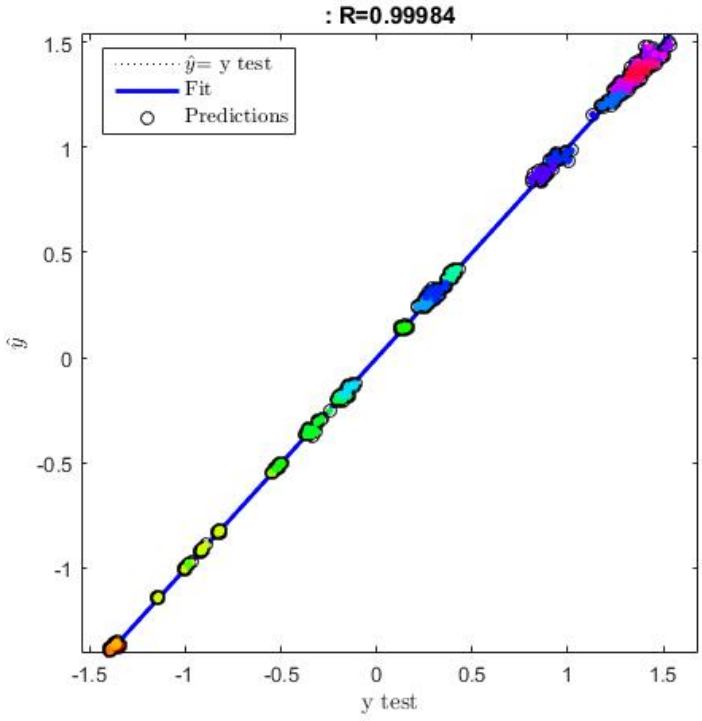

Figure 4.5: Predictions from the GPR after the first online refinement for the global model. Test outputs are on the x-axis and predicted outputs on the y-axis.

In Figure 4.5 R is slope of the resulting line that tries to fit all points. The circles are the predictions. Each colour correspond to an OP of the test dataset. It can be seen that the fit increases after the first refinement and the MSPE decreases to $3.0004 \cdot 10^{-4}$ from $3.3873 \cdot 10^{-4}$.

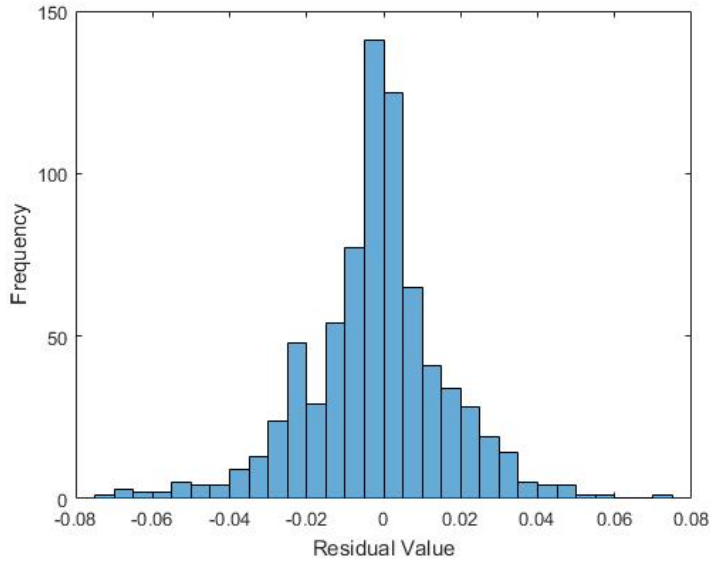

Figure 4.6: Residual plot for the global model after the first online refinement. The number of how frequent each residual value is occurring (frequency) on the y-axis and the residual value (prediction-test output) in scaled units on the x-axis.

The MSPE value of BSFC (predicted-actual) of the model after the first online refinement decreased to $3.0004 \cdot 10^{-4}$. The predictions from the online refinement is shown in Figure 4.5 where a slight increase in the fit can be observed. The residual value plot is shown in Figure 4.6, the residual plot mostly look similar to the residual plot before refinement, but with more samples around smaller residuals but that still resulted in lower MSPE.

The UPs can be placed at existing OPs, which can then be seen by an increased number of predictions of the same color in comparison to the initial GPR. At a closer look, it can be seen that UPs are placed in the upper right corner of Figure 4.3 in comparison to Figure 4.5. These UPs are of the same color, which indicate that UPs are placed at some OPs. These phenomena is even more clear between Figure 4.9 and Figure 4.11. UPs can also be placed at new OPs, which would then be indicated by predictions of new colors that are not present in the initial GPR. This can be seen in comparison between Figure 4.21 and Figure 4.23.

It is visible by careful inspection that the predictions looks very similar between Figure 4.3 and Figure 4.5 where no UPs have been placed. This consequently indicate that the training of the GPR also tends to be similar between runs, given the same data (where there are no UPs). The similarity between training rounds can be seen by considering the marginal likelihood which is maximized during the training of GPR (see Section 2.2.4). In Table 3.2 the marginal likelihoods are presented that resulted from some training rounds of the GPR.

Another online refinement is done to further improve the model. The same procedure as the first refinement is used here.

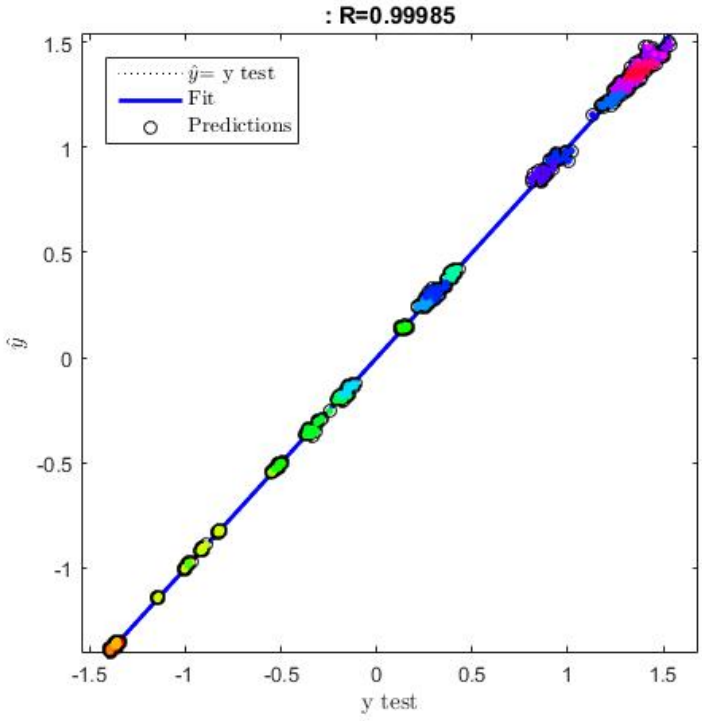

Figure 4.7: Predictions from the GPR after the second online refinement for the global model. Test outputs are on the x-axis and predicted outputs on the y-axis.

In Figure 4.7 R is slope of the resulting line that tries to fit all points. The circles are the predictions. Each colour correspond to an OP of the test dataset. It can be seen that the fit increases after the second refinement compared to the first refinement and the MSPE decreases to $2.9895 \cdot 10^{-4}$ from $3.0004 \cdot 10^{-4}$.

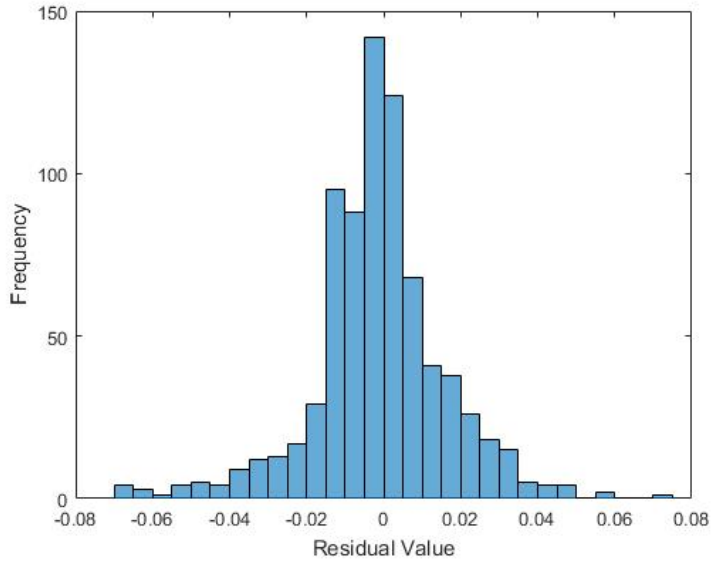

Figure 4.8: Residual plot for the global model after the second online refinement. The number of how frequent each residual value is occurring (frequency) on the y-axis and the residual value (predictions-test output) in scaled units on the x-axis.

The MSPE value of BSFC (predicted-actual) after the second online refinement has further reduced and is now at $2.9895 \cdot 10^{-4}$ in scaled units, the fit have increased slightly as well. The predictions from the second online refinement is shown in Figure 4.7 and the residual value plot is shown in Figure 4.8. In the residual plot it can be seen that the previous spike around residual value -0.02 is now gone but another spike has instead appeared around -0.016, the frequency around some smaller values has also increased. This occurrence is deemed as more positive.

Thus, it can be concluded that the objective of conducting an online refinement: to improve the model prediction and to decrease the MSPE have been achieved.

Even though, the results look quite convincing in the global model, when looking more closely at the predictions, it could be seen that there are a cluster of datapoints in certain regions of the model. When zoomed in, the fit of these clusters is worse, which suggest that the model predictions are worse at certain OPs. The fit of the global model is still high due to the large number of datapoints at better predicted regions. These points heavily influence the overall fit to become high. Hence, it is decided to look into local models. For more background information see 3.4.2.

4.1.2 Low Torque Region

The Low Torque region contains 47 OPs. These OPs have an engine speed of 600-2000 rpm and engine torque between 0-8.8 Nm. The initial regression model is presented first and then the model is shown after two consecutive online refinements.

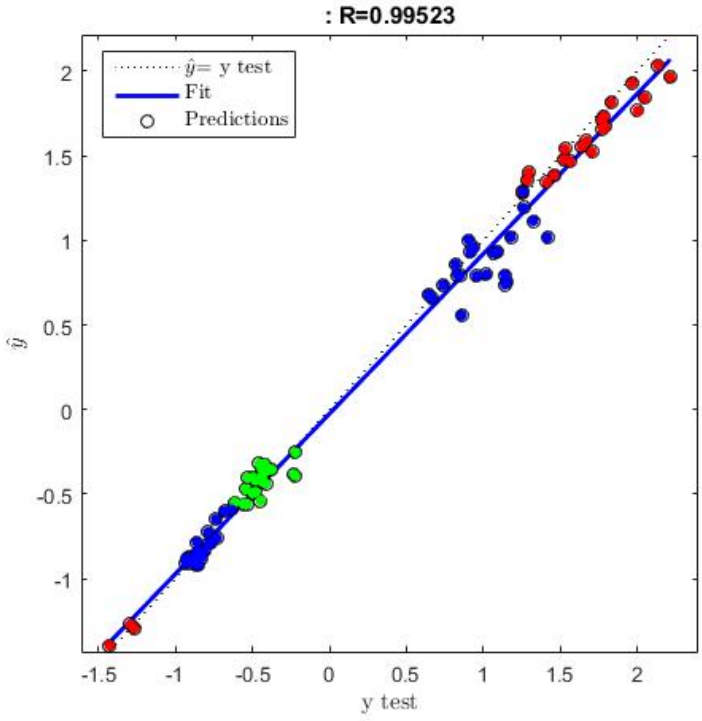

Figure 4.9: Predictions from the GPR for the initial dataset of the low torque region. Test outputs are on the x-axis and predicted outputs on the y-axis.

In Figure 4.9 R is slope of the resulting line that tries to fit all points. The circles are the predictions. Each colour correspond to an OP of the test dataset. The MSPE is 0.0161.

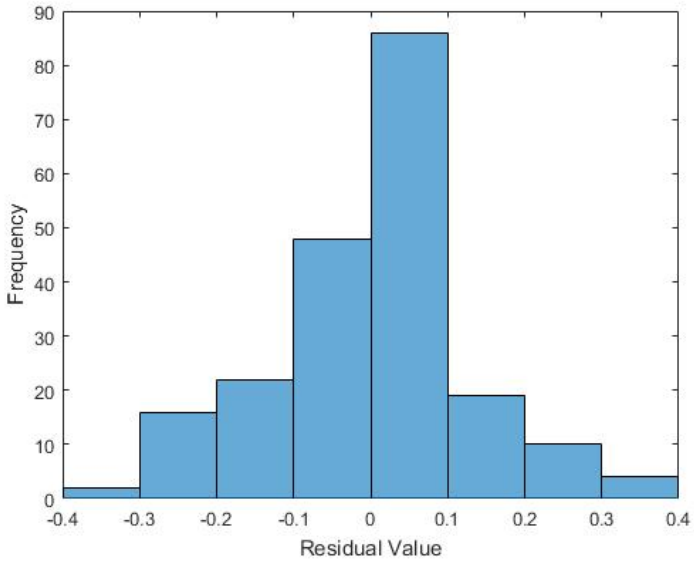

Figure 4.10: Residual plot for the low torque-model. The number of how frequent each residual value is occurring (frequency) on the y-axis and the residual value (prediction-test output) in scaled units on the x-axis.

The MSPE for BSFC (predicted-actual) in scaled units is 0.0161 for the initial model. The predictions from the initial regression is shown in Figure 4.9 and the residual value plot is shown in Figure 4.10. The residual plot is mostly centered around small residual values. An online refinement is done to improve the existing model.

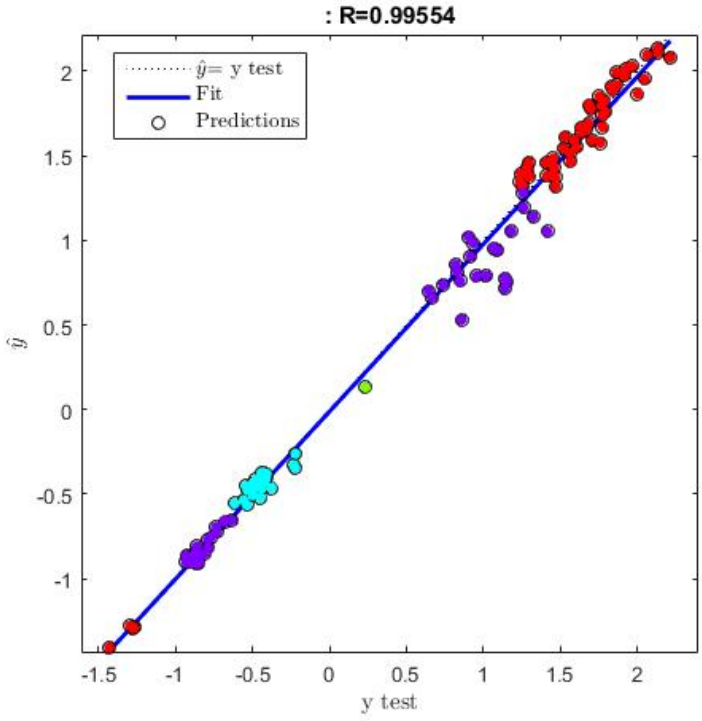

Figure 4.11: Predictions from the GPR after the first online refinement for the low torque-model. Test outputs are on the x-axis and predicted outputs on the y-axis.

In Figure 4.11 R is slope of the resulting line that tries to fit all points. The circles are the points. Each colour correspond to an OP of the test dataset. It can be seen that the fit increases after the first refinement and the MSPE decreases to 0.0093 from 0.0161.

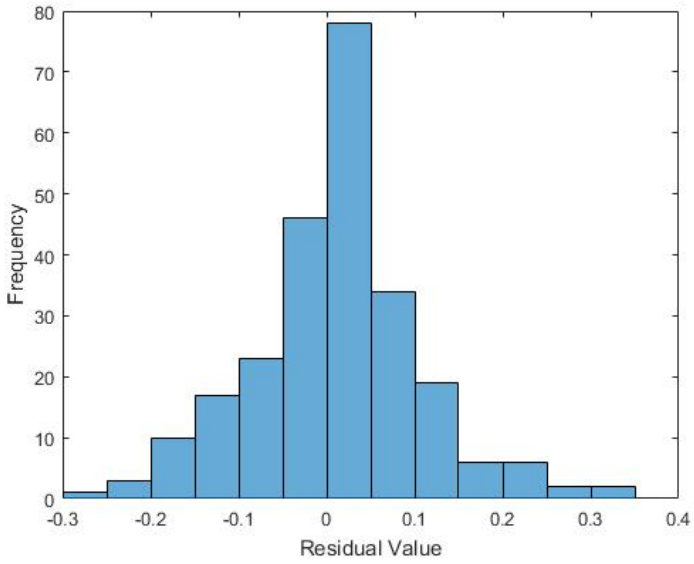

Figure 4.12: Residual plot for the low torque-model after one online refinement. The number of how frequent each residual value is occurring (frequency) on the y-axis and the residual value (predictions-test output) in scaled units on the x-axis.

The MSPE for BSFC (predicted-actual) after the first refinement decreased to 0.0093, the fit has also increased slightly. The predictions after the first refinement is shown in Figure 4.11 and the residual plot is shown in Figure 4.12. In the residual plot, more staples can be seen in comparison to the initial GPR, this is suspected due to addition of UPs. The residuals mostly have the same distribution as before, but with a small increase in the frequency around the residual value 0.05. In order to improve the local model further, a second online refinement is performed.

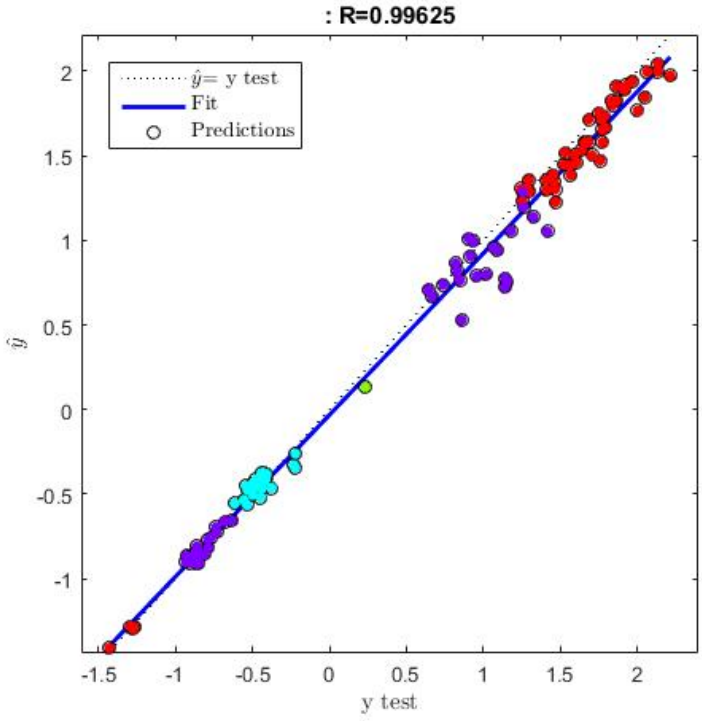

Figure 4.13: Predictions from the GPR after the second online refinement for the low torque-model. Test outputs are on the x-axis and predicted outputs on the y-axis.

In Figure 4.13 R is slope of the resulting line that tries to fit all points. The circles are the predictions. Each colour correspond to an OP of the test dataset. It can be seen that the fit increases after the second refinement compared to the first refinement and the MSPE decreases to 0.0091 from 0.0093.

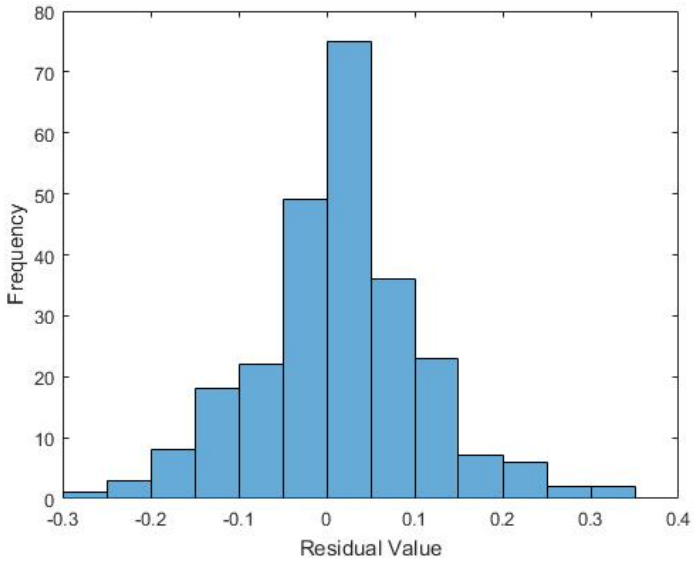

Figure 4.14: Residual plot for the low torque-model after the second online refinement. The number of how frequent each residual value is occurring (frequency) on the y-axis and the residual value (predictions-test output) in scaled units on the x-axis.

The MSPE for BSFC (predicted-actual) in scaled units after the second online refinement further decreased to 0.0091. The predictions from the second refinement is shown in Figure 4.13 and the residual plot is shown in Figure 4.14. The residual plot look similar to the residual plot in Figure 4.12.

In the low torque region the fit of the model is improving too and the MSPE is decreasing with successive online refinement rounds. However, in comparison with the global model, it could be noted that the MSE values of the local model are higher. Also, the fit of the model is slightly less fit than the global model. This is due to the distribution of OPs in the local model and due to the variability of OPs in the local model. The importance of variability in OPs is discussed in Section 3.1.3.

4.1.3 Part Load Region

138 OPs out of the total 220 OPs are located in the part load region. The engine speed in this region varies from 600-2000 rpm and the engine torque varies from 165-1400 Nm. The initial regression model is presented first and then the model is shown after two consecutive online refinements.

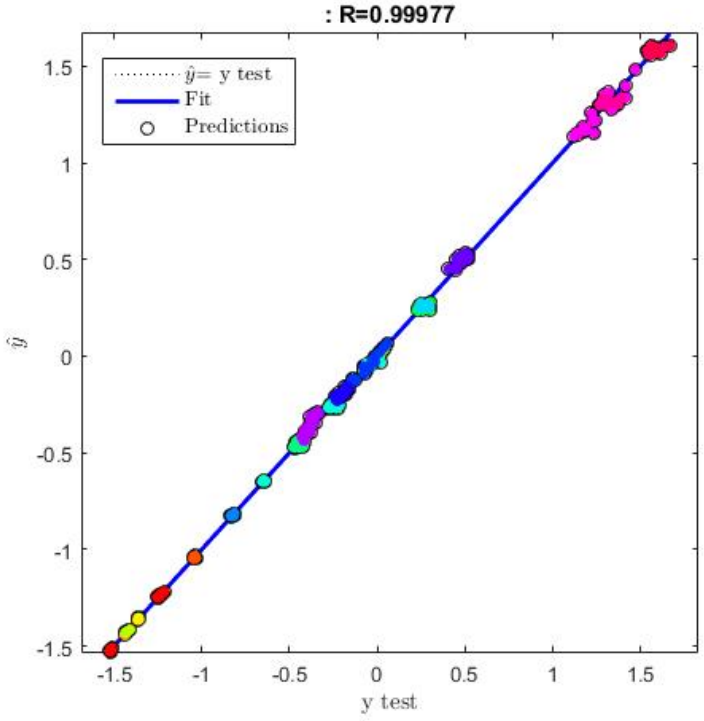

Figure 4.15: Predictions from the GPR for the initial dataset of the part load region. Test outputs are on the x-axis and predicted outputs on the y-axis.

In Figure 4.15 R is slope of the resulting line that tries to fit all points. The circles are the predictions. Each colour correspond to an OP of the test dataset. The MSPE is $4.5400 \cdot 10^{-4}$.

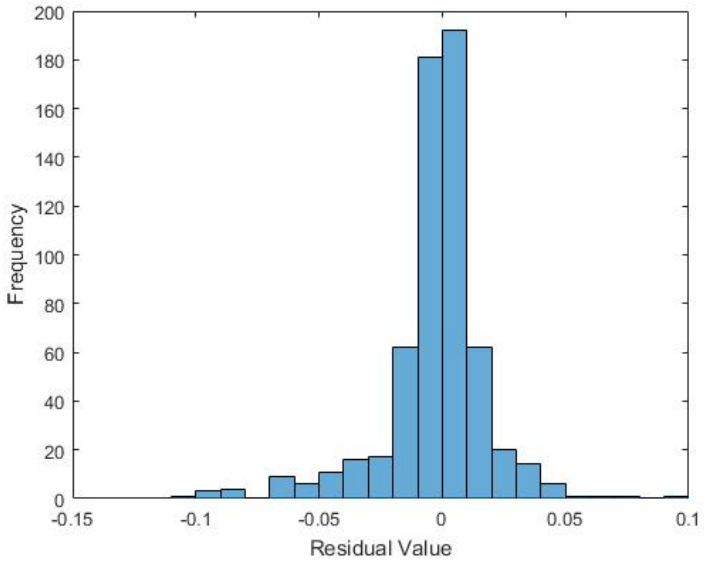

Figure 4.16: Residual plot for the part load model. The number of how frequent each residual value is occurring (frequency) on the y-axis and the residual value (prediction-test output) in scaled units on the x-axis.

The MSPE for BSFC (predicted-actual) for the initial regression is $4.5400 \cdot 10^{-4}$. The prediction from the initial regression is in Figure 4.15 and the residual value plot is shown in Figure 4.16. In the residual plot, it can be seen that there are a high and sharp frequency around small residual values and very much less around higher values. This is a favorable residual distribution. An online refinement is done in order to improve the model even further and to reduce the MSPE.

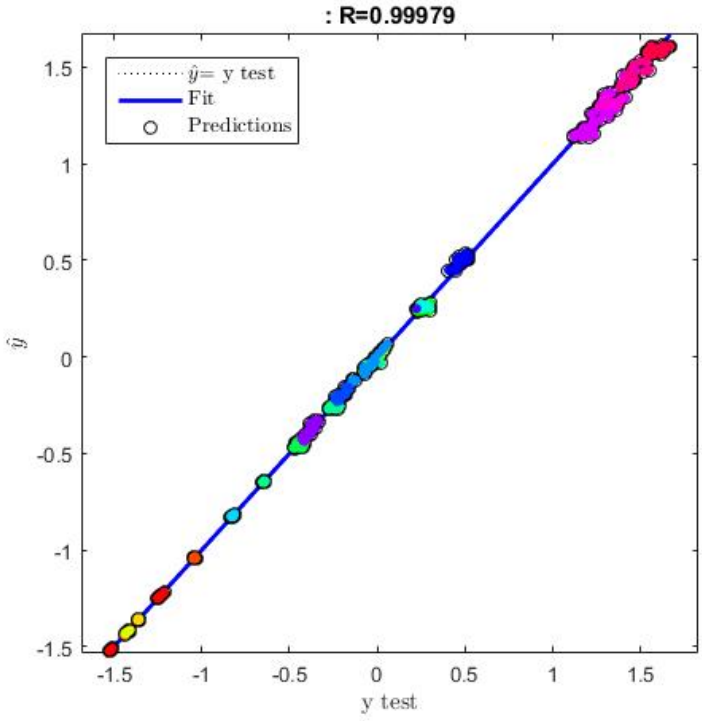

Figure 4.17: Predictions from the GPR after the first online refinement for the part load model. Test outputs are on the x-axis and predicted outputs on the y-axis.

In Figure 4.17 R is slope of the resulting line that tries to fit all points. The circles are the predictions. Each colour correspond to an OP of the test dataset. It can be seen that the fit increases after the first refinement and the MSPE decreases to $4.0808 \cdot 10^{-4}$ from $4.5400 \cdot 10^{-4}$.

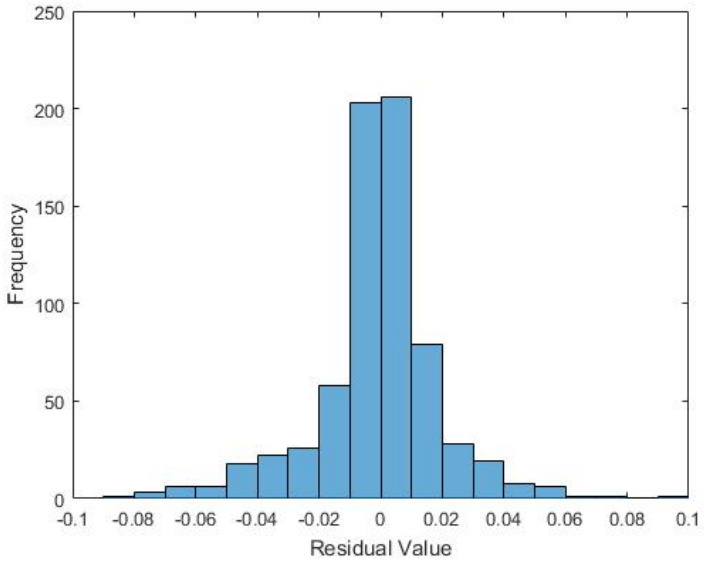

Figure 4.18: Residual plot for the part load model after one online refinement. The number of how frequent each residual value is occurring (frequency) on the y-axis and the residual value (predictions-test output) in scaled units on the x-axis.

The MSPE for BSFC (predicted-actual) after one online refinement decreased to $4.0808 \cdot 10^{-4}$, the fit barely increased. The prediction from the online refinement is shown in Figure 4.17 and the residual value plot is shown in Figure 4.18. In the residual plot, an increased frequency around the smallest residual values can be observed, which is positive. A second online refinement is done in order to improve the model even further.

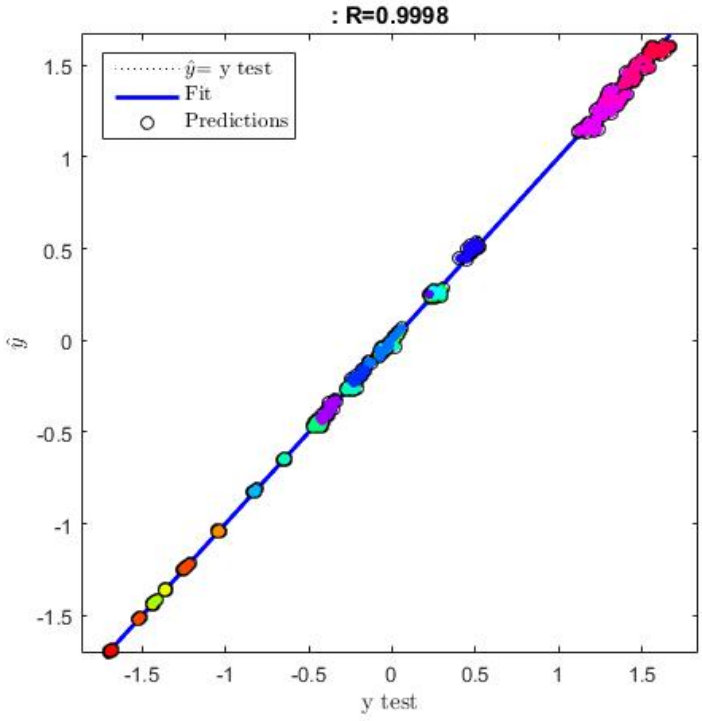

Figure 4.19: Predictions from the GPR after the second online refinement for the part load model. Test outputs are on the x-axis and predicted outputs on the y-axis.

In Figure 4.19 R is slope of the resulting line that tries to fit all points. The circles are the predictions. Each colour correspond to an OP of the test dataset. It can be seen that the fit increases after the second refinement compared to the first refinement and the MSPE decreases to $3.9825 \cdot 10^{-4}$ from $4.0808 \cdot 10^{-4}$.

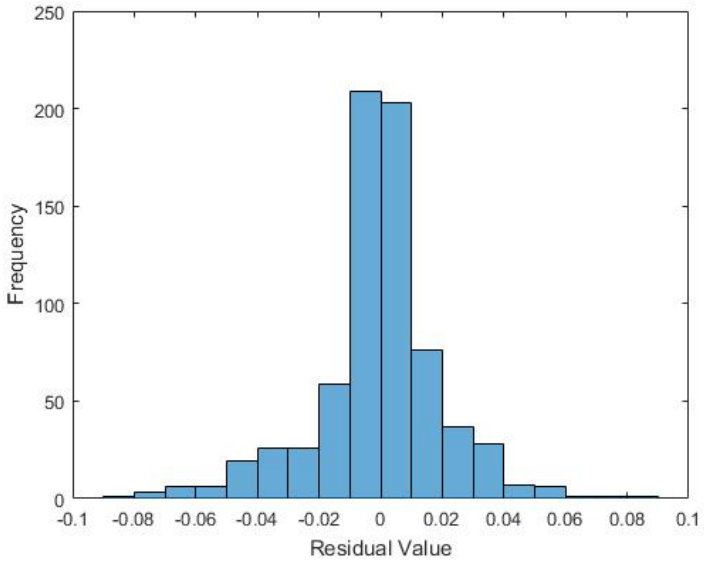

Figure 4.20: Residual plot for the part load model after the second online refinement. The number of how frequent each residual value is occurring (frequency) on the y-axis and the residual value (prediction-test output) in scaled units on the x-axis.

The MSPE for BSFC (predicted-actual) after the second online refinement further decreased to $3.9825 \cdot 10^{-4}$, the fit also increased barely. The prediction from the second online refinement is shown in Figure 4.19 and the residual value plot is shown in Figure 4.20. The residual plot look similar to Figure 4.18. Due to having large number of OPs, the part load region has the biggest effect on the global model. This is clear when looking at the fit of the local model in the part load region and the MSPE values. These values are almost identical to the values of the global model.

4.1.4 High Torque Region

35 OPs are present in the High Torque region. The engine speed in the high torque region varies within 1000-1900 rpm and the engine torque between 1500-2650 Nm. The initial regression model is presented first and then the model is shown after two consecutive online refinements.

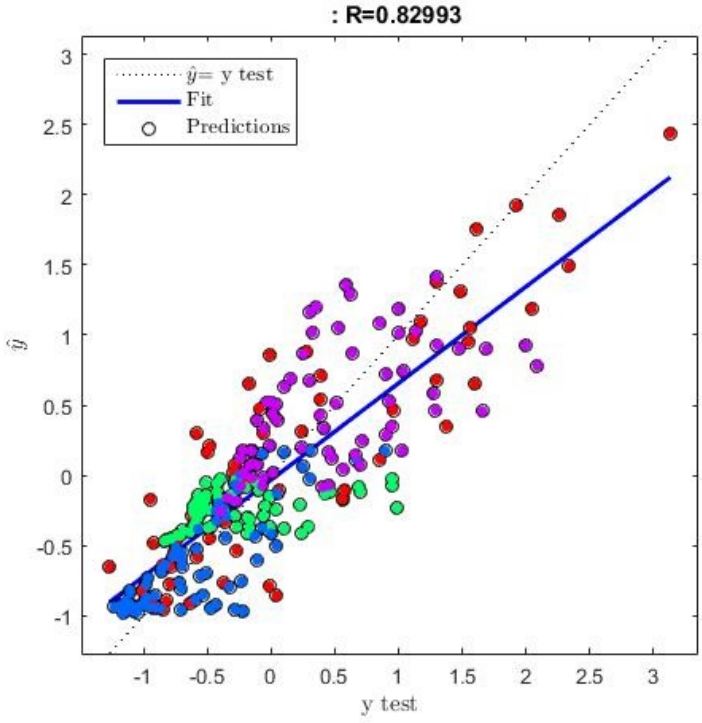

Figure 4.21: Predictions from the GPR for the initial dataset of the High Torque region. Test outputs are on the x-axis and predicted outputs on the y-axis.

In Figure 4.21 R is slope of the resulting line that tries to fit all points. The circles are the predictions. Each colour correspond to an OP of the test dataset. The MSPE in scaled units is 0.2092.

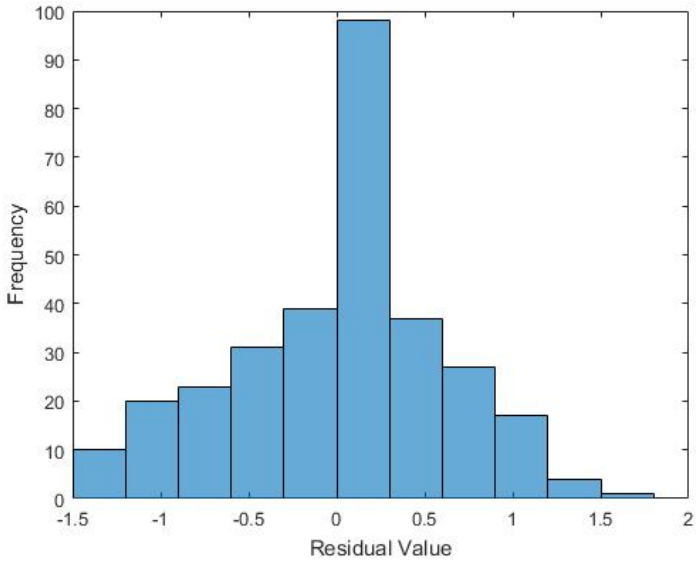

Figure 4.22: Residual plot for the high torque-model. The number of how frequent each residual value is occurring (frequency) on the y-axis and the residual value (prediction-test output) in scaled units on the x-axis.

The MSPE for BSFC (predicted-actual) for the initial regression is 0.2092, higher than the other models shown. The prediction from the regression model is in Figure 4.21 and the residual value plot is in Figure 4.22. In the residual plot it can be seen that the residuals are mostly centered around the residual values 0 to 0.25, which is considerably higher than the residual values for the other models. In fact all residuals in the distribution are of higher values than the previously showed models. An online refinement is performed to increase the performance of the model.

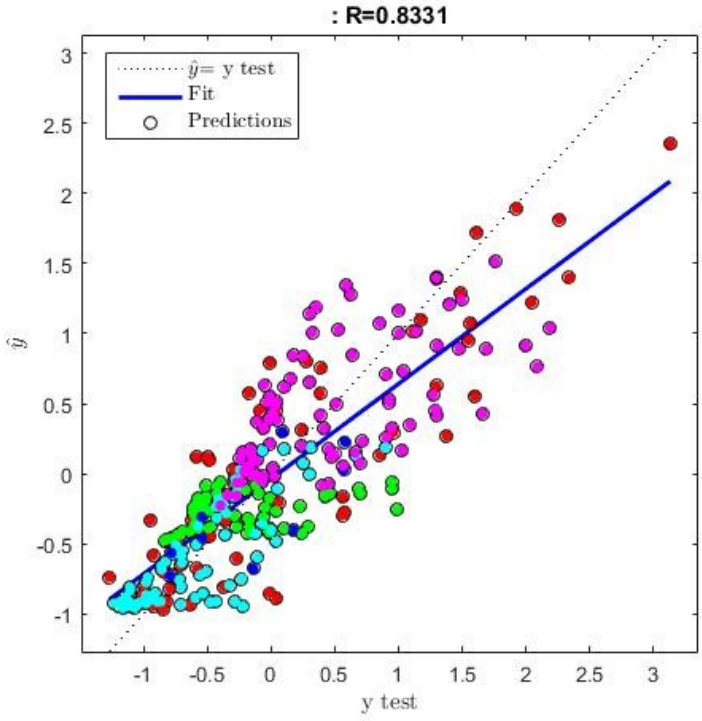

Figure 4.23: Predictions from the GPR for the high torque-model after the first online refinement. Test outputs are on the x-axis and predicted outputs on the y-axis.

In Figure 4.23 R is slope of the resulting line that tries to fit all points. The circles are the predictions. Each colour correspond to an OP of the test dataset. It can be seen that the fit increases after the first refinement and the MSPE decreases to 0.1851 from 0.2092.

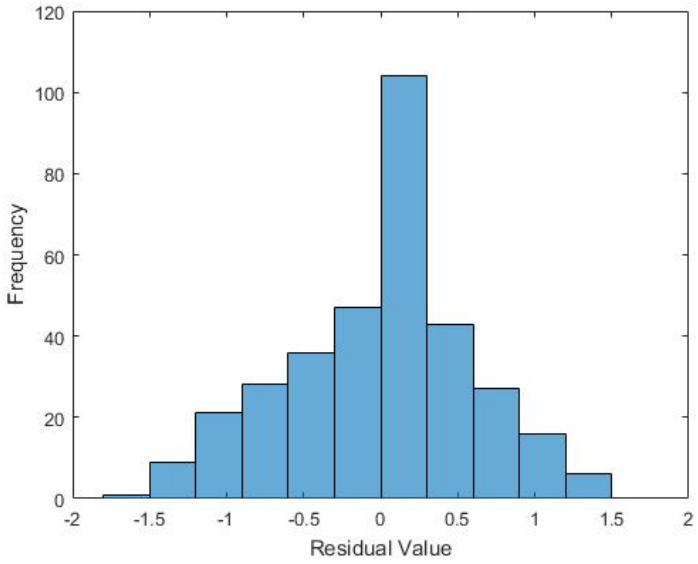

Figure 4.24: Residual plot for the high torque-model after one online refinement. The number of how frequent each residual value is occurring (frequency) on the y-axis and the residual value (prediction-test output) in scaled units on the x-axis.

The MSPE for BSFC (predicted-actual) after the first online refinement decreased to 0.1851, the fit also increased. The predictions from the refinement is shown in Figure 4.23 and the residual value plot is shown in Figure 4.24. In the residual plot it can be seen that the residual distribution look similar to Figure 4.22 but with an increased frequency at some of the smaller residual values.

A second online refinement is done to improve the results even further.

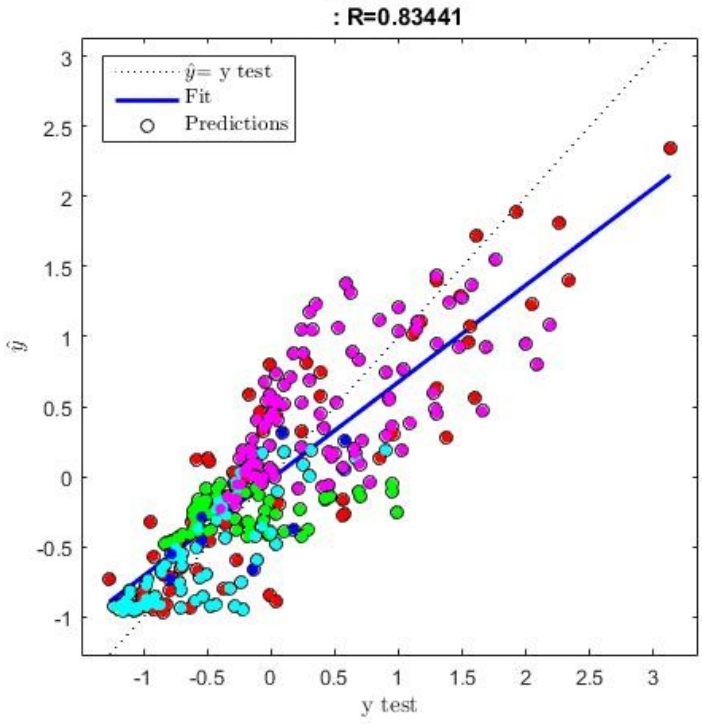

Figure 4.25: Predictions from the GPR for the high torque-model after the second online refinement. Test outputs are on the x-axis and predicted outputs on the y-axis.

In Figure 4.25 R is slope of the resulting line that tries to fit all points. The circles are the predictions. Each colour correspond to an OP of the test dataset. It can be seen that the fit increases after the second refinement compared to the first refinement and the MSPE decreases to 0.1756 from 0.1851.

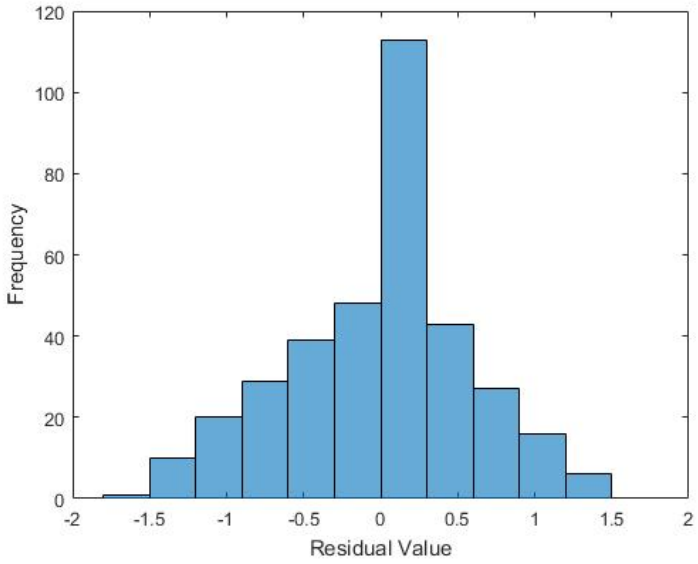

Figure 4.26: Residual plot for the high torque-model after the second online refinement. The number of how frequent each residual value is occurring (frequency) on the y-axis and the residual value (prediction-test output) in scaled units on the x-axis.

The MSPE for BSFC (predicted-actual) after the second online refinement further decreased to 0.1756, the fit slightly increased also. The predictions from the second online refinement is shown in Figure 4.25 and the residual value plot is shown in Figure 4.26. The residual plot look similar to Figure 4.24 but with an increase of the frequency at some of the smaller residual values.

A conclusion that could be drawn from the results is that a GPR for the region with high engine torque is not as successful as other models in predicting the outputs. A possible reason is maybe due to the difference in the ranges of inputs in the high torque region than the low torque and part load region. The OPs in the high torque region are spread across a larger region (see Figure 3.1 from 1500 Nm and upwards) and also sometimes multiple OPs are placed closer to each other. The number of OPs in the high torque region is also less than the other regions (see Table 3.3).

It can now be understood why all the UPs are placed in the upper right corner of the regressionplots for the global model in Figure 4.5 and 4.7, they correspond to the high-torque region. As the high torque model is the hardest and most uncertain to predict, the predicted variance (2.12) is at the highest in that region, thus the UPs will be placed there according to the online refinement method (see Figure 3.17).

4.2 Online Refinement

The objective of conducting an online refinement is to improve the model online. As it could be seen for instance in Figure 4.5 and Figure 4.7, the fit of the model improves after the online refinement. Also, the MSPE which is another measure of the model is reduced by conducting the online refinement. The reduction of MSPE values for the above models is summarized in Table 4.1.

	Global Model	Low Torque	Part Load	High Torque
Initial Model	$3.3873 \cdot 10^{-4}$	0.0161	$4.5400 \cdot 10^{-4}$	0.2092
First Refinement	$3.0004 \cdot 10^{-4}$	0.0093	$4.0808 \cdot 10^{-4}$	0.1851
Second Refinement	$2.9895 \cdot 10^{-4}$	0.0091	$3.9825 \cdot 10^{-4}$	0.1756

Table 4.1: The MSPE reduction in scaled units upon online refinement for the Global model as well as for the Low, Part load and High torque region-models.

The unscaled MPSE for all the initial models are shown in Table 4.2.

	Global Model	Low Torque	Part Load	High Torque
Initial Model	0.303	0.001	0.205	0.68

Table 4.2: The unscaled MPSE for all the initial models.

The decrease in the scaled MSPE can be more visually represented by looking at the Figures 4.27, 4.28, 4.29 and 4.30.

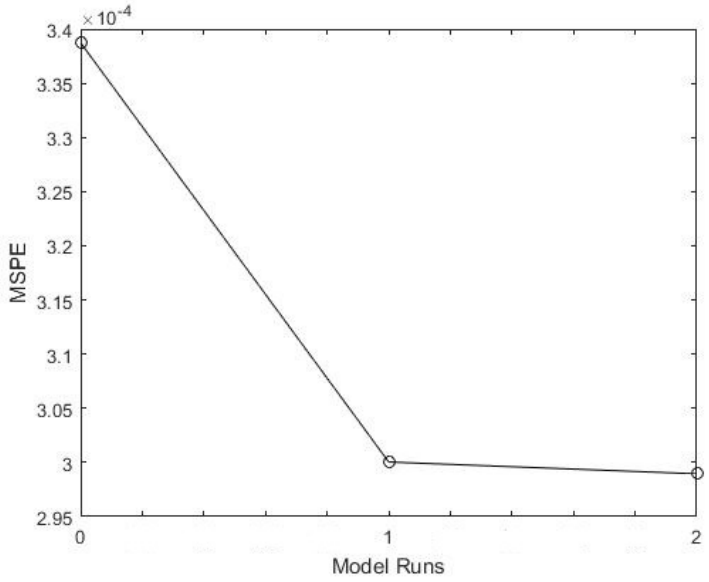

Figure 4.27: The decrease in scaled MSPE for the Global model. Number of GPR runs are on the x-axis, and MSPE on the y-axis.

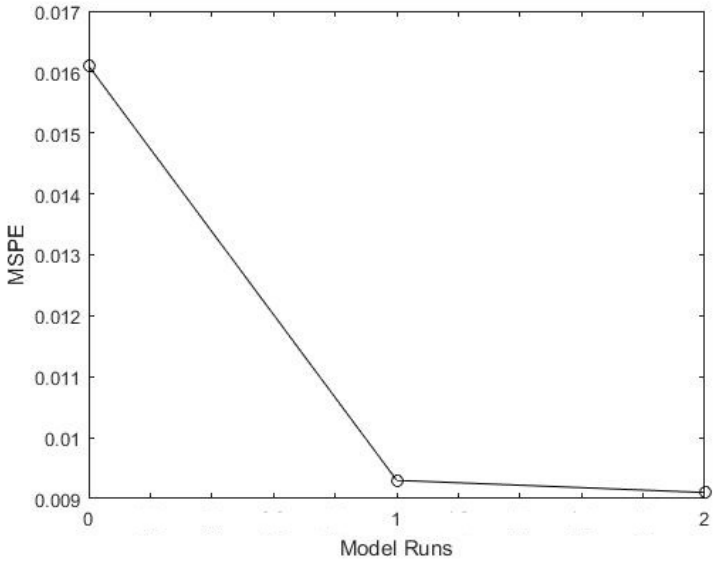

Figure 4.28: The decrease in scaled MSPE for the Low Torque region-model. Number of GPR runs are on the x-axis, and MSPE on the y-axis.

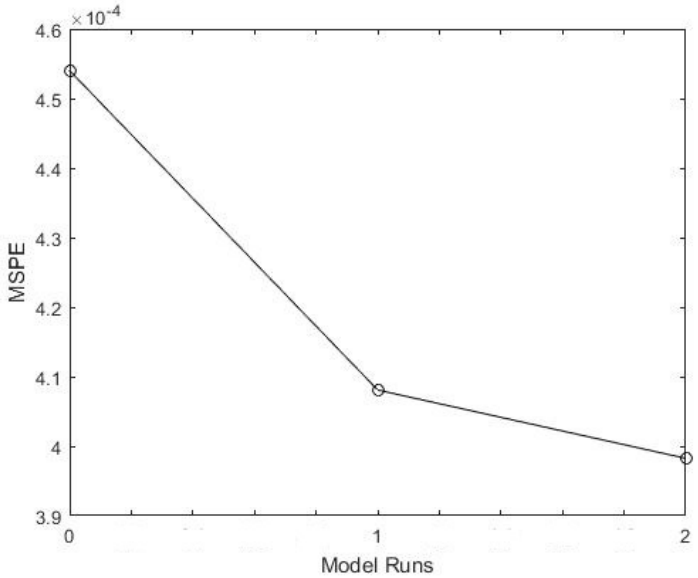

Figure 4.29: The decrease in scaled MSPE for the part load region-model. Number of GPR runs are on the x-axis, and MSPE on the y-axis.

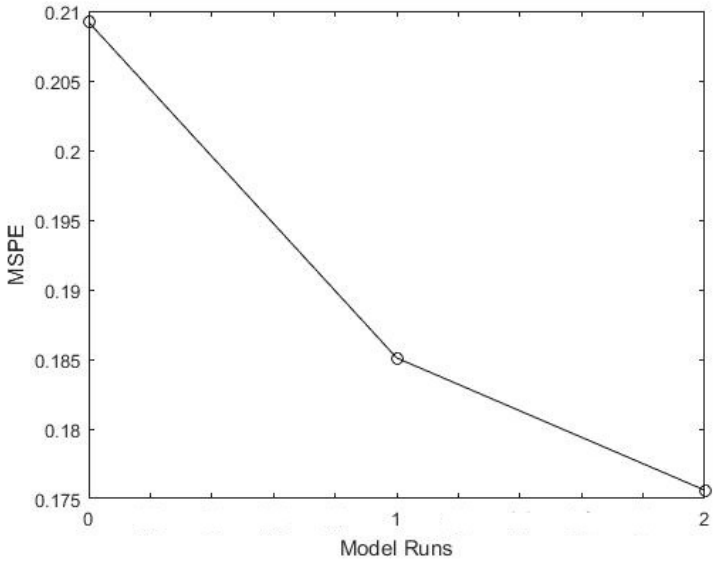

Figure 4.30: The decrease in scaled MSPE for the High Torque region-model. Number of GPR runs are on the x-axis, and MSPE on the y-axis.

The MSPE values of the global and the three local models are decreasing with the refinement runs. This, in combination with improved fit indicate that the model gets better at predicting the system due to refinement. This is an desired result as the objective of an online refinement is to improve the model online. When looking more closely at the MSPE decrease it could be seen that, the decrease in MSPE is smaller from the first refinement to second refinement in comparison from the initial model to the first refinement. This trend is least apparent in the high torque region, but true for all models.

4.3 Automatic Relevance Determination

ARD is a feature in GPR that allows the user to determine the relative importance of different parameters used for the model. The theoretical background of the ARD is described in the theory chapter under the Section 2.2.5. As mentioned, ARD enables the possibility to determine which input variables have little effect as well as those who have a significant effect on regression, by looking at the absolute value of the lengthscale l for each input parameter. Table 4.3 shows the results of ARD by showing the values of the lengthscales for the Global, Low torque, Part load and High torque-models.

In Table 4.3 corresponding values of the lengthscale for each input parameter is listed for every model. A smaller absolute value of the lengthscale indicate an input parameter of greater importance for regression (and vice versa).

As it could be seen from Table 4.3, a justification for the selection of two extra inputs can be understood. The two initial inputs, injection timing (advance angle)

	Global Model	Low Torque	Part Load	High Torque
Engine Speed	1.3840	0.6917	1.3531	2.9382
Engine Torque	−0.4556	4.4551	0.2607	1.4265
Injection Timing (Advance Angle)	7.0540	5.0650	8.3570	4.7413
Injection Pressure (NOP)	6.1166	5.1650	8.7849	6.6663
Throttle Position	2.0854	−0.3440	1.8109	1.1426
EGR Position	2.4016	1.2072	2.6278	1.1846

Table 4.3: The ARD results for the four models after refinement.

and injection pressure (NOP angle) have the highest absolute values of the lengthscales, which suggests that they are the two most irrelevant inputs when modeling the system. This finding is contrary to the well known fact that injection pressure and timing are important parameters for modelling engines, thus further investigation is needed. Both throttle position and EGR position have much less absolute values of the lengthscales in comparison, which in return means that they are much more relevant parameters for the model.

4.4 Noise Standard Deviation

As explained in the demonstration in Section 2.4.3 the GPR accommodates for noise apparent in the data. A measure of the noise in each of the separate models can be obtained by looking at the noise hyperparameter σ_{noise} (noise standard deviation) in (2.5) achieved from training.

Model	Noise Standard Deviation (σ_{noise})
Global Model	$8.0 \cdot 10^{-5}$
Low Torque	$3.22 \cdot 10^{-2}$
Part Load	$3.025 \cdot 10^{-4}$
High Torque	$5.26 \cdot 10^{-1}$

Table 4.4: Values of the noise hyperparameter (noise standard deviation) in scaled units for each of the separate models after refinement.

It can be seen in Table 4.4 that the highest noise standard deviation is at the high torque region local model. The lowest noise standard deviation is at the global model. This is an expected result as the global model is the most successful model seen over its whole range in terms of MSPE and fit (but not necessarily when zoomed in into certain regions) and the high torque model, the least successful model. The noise in the high torque region is significantly higher than the noise in the global model and will consequently make the high torque region more difficult to predict. Although the GPR can accommodate for the noise in the data, the GPR will still

encounter greater difficulty predicting with noisy data than compared to noise-free data.

5
Future Work

There are a few inquiries concerning the thesis work that stayed unanswered and are hence proposed as future research questions.

Only the modeling aspect of a model based engine calibration is considered for this thesis. The second step, which consists of optimization, needs to be considered in the future. However, a basic foundation is laid into the optimization below in Section A.3.

For now, only BSFC is considered as an output to the system. Even though, NOx emission are considered as well, further investigation needs to be carried out into the usage of different output parameters.

The addition of more input parameters would be interesting to examine in order to see if the performance of GPR would increase, thus enabling capture of more input-output behaviour. It's also interesting to see if this improvement would stop after a certain number of added input parameters.

The addition of more DoE samples at every OP is also interesting to investigate further, the models should improve further by adding more DoE samples. A preliminary investigation is shown for the high torque model in Section A.5.

Alternative approaches for online refinement can be looked into in future. One alternative is for instance to use the predicted variance (2.12) in a different fashion. Instead of looking at the predicted variance of the outputs (observations), it may be possible to optimize and search for the test input that would maximize the predicted variance and then obtain new measurements at those corresponding regions. Those regions should correspond to areas where more measurements are needed in order to decrease the uncertainty of the model with the least measurements possible. For more information see [29].

Interesting opportunities exist for an online optimization when a GPR is used for the modeling, some promising optimization approaches are described further in detail in [7].

6
Implementation

This chapter discusses practical aspects of the thesis work. The chapter also gives a background to the software used throughout the work.

The thesis was performed mainly at Volvo Penta (Gropegårdsgatan 11), since the VIRTEC system is located here. A significant part of the thesis was also performed at the Chalmers University of Technology, in order to get direct feedback from the supervisors and the examiner. The thesis was carried out as a part of the bigger VIRTEC/VIRCAL project, which aims to build a virtual in-house solution for the test cells. Ethan Faghani, Project Manager at Volvo Penta and a supervisor of the thesis has a good knowledge about the limitations and scope of the thesis and about the expected results from the thesis. Since the thesis is aimed at solving an industrial problem, it was decided to spend as much time as possible within the industry, and develop a solution that suits the requirements of the industry.

Jonas Sjöberg, professor in Mechatronics and attached to the Electrical Engineering department at Chalmers, was the examiner for the thesis while Jonas Sjöblom and Ali Ghanaati from the Automotive Engineering Department along with Ethan Faghani, from Volvo Penta were the other supervisors for the project.

The entire project was conducted using MATLAB/Simulink software. A final code consisting approximately of 700 lines was developed for this thesis. The Engine Open Loop Simulink block (Black box model for the engine) was incorporated into the code. The project can be easily expanded to include more parameters as suggested in Future Work by expanding the code.

Running the Global model with online refinement takes in average 1.5 hours. All the local models will consume less time than the Global model to run. The timings for the execution are in reference to a Intel(R) Core(TM) i5-6600K 3.5 GHz with 16GB RAM. The LHS DoE was implemented via the lhsdesign function in MATLAB.

The GPR (`http://www.gaussianprocess.org/gpml/code/matlab/doc/`) was implemented using the Gaussian Processes for Machine Learning (GPML) Toolbox in MATLAB by Carl Edward Rasmussen (Department of Engineering, University of Cambridge) and Hannes Nickisch (Max Planck Institute for Biological Cybernetics). The toolbox is well known in the field of Machine Learning and is free to download.

7
Conclusion

This chapter discusses conclusions of the results.

The main deliverable from the thesis is an online model for engine calibration. A model is constructed for 6 inputs and one output. Measurements are taken from 220 OPs along the entire engine speed vs, engine torque map used for engine calibration.

A discovery is that some OPs do not contain any input variability, hence providing non-informative data for any regression method. These OPs are not modelled.

The model covers different torque regions: low torque, part load and high torque. The part load and low torque are the most accurate and the high torque region the least accurate. The OPs are scattered throughout the high torque region, thus creating a greater challenge for the GPR. There are also more repeated OPs in the high torque region as well. This data is also more noisy than data for the other torque regions. Although GPR can accommodate for noisy data, in comparison noise free data is modelled better.

The part load region consisted of the greatest number of OPs with a tight spread and this data is also less noisy. These traits aids the GPR in making good predictions.

A flexible and a transparent model is developed using GPR, that has the features of uncertainty estimates of the output parameters in order to add UPs where the confidence interval is at the highest width. An online refinement results in a refined model with a better fit and lesser MSPE.

A feature of GPR is ARD, which distinguish how relevant each of the input parameter are to the model. Injection timing and injection pressure are the input parameters having the least effect on the model. The ability of GPR to handle noisy data, can be seen when looking at the noise standard deviations in each of the models.

The application developed synchronizes the execution of DoE, GPR and online refinement into a single code.

Bibliography

[1] Roepke, Karsten *Design of Experiments for Engine Calibration*, IAV GmbH, Carnotstrasse 1, Berlin, Germany, 1-2, 2014 `https://www.jstage.jst.go.jp/article/sicejl/53/4/53_322/_pdf`2018-11-7

[2] Rask, Eric and Sellnau, Mark *Simulation-Based Engine Calibration: Tools, Techniques, and Applications*, SAE International, 1, 2004

[3] Basic, Marko *On hardware-in-the-loop simulation* 44th IEEE Conference on Decision and Control, European Control Conference 3194-3198, IEEE, 2005 `https://ieeexplore.ieee.org/document/1582653/authors#authors`

[4] `https://www.volvopenta.com`

[5] Šarkan, Branislav, Skrúcaný, Tomas and Majerová, Jana, *Possibilities of Measuring the Brake Specific Fuel Consumption in Road Vehicle Operation*, The Faculty of Operation and Economics of Transport and Communications – University of Zilina , the Slovac Republic, 2014, Scientific Proceedings XII International Scientific-Technical Conference "trans and Motauto '14

[6] Atkinson, Chris and Mott, Gregory, *Dynamic model-based calibration optimization: An introduction and application to diesel engines*, SAE Technical Paper, 2005

[7] Berger, Benjamin, *Modeling and optimization for stationary base engine calibration*, Technische Universität München, 2012

[8] Nyberg, Peter, *Evaluation, Generation, and Transformation of Driving Cycles*, Linköping Studies in Science and Technology Dissertations, No. 1669, Department of Electrical Engineering, 2015 Chapter 1

[9] Castagné, Michael, Bentolila, Yohan, Chaudoye, Fabien, Hallé, Adrien, Nicolas, Frederic and Sinoquet, Delphine *Comparison of Engine Calibration Methods Based on Design of Experiments (DoE)*, Oil Gas Science and Technology – Rev. IFP, Vol. 63 (2008), No. 4, pp. 563-582, Institut français du pétrole 2008 17-18 `https://ogst.ifpenergiesnouvelles.fr/articles/ogst/pdf/2008/04/ogst07105.pdf`2018-10-28

[10] Stuhler, Harald, Kruse, Thomas, Stuber, Axel, Gschweitl, Kurt, Piock, Walter, Pfluegl, Horst and Lick, Peter, *Automated model-based GDI engine calibration adaptive online DoE approach*, SAE Technical Paper, 2002

[11] Kviklien, Adriana, Jurkonis, Rytis, Ressner, Marcus, Hoff, Lars, Jansson, Tomas, Janerot, Sjöberg, Birgitta, Lukoševičius, Arūnas and Ask, Per, *Modelling of nonlinear effects and the response of ultrasound contrast micro bubbles: simulation and experiment*, Ultrasonics, 42, 1-9, 301–307, Elsevier, 2004

[12] Tietze, Nils, *Model-based Calibration of Engine Control Units Using Gaussian Process Regression*, der Technischen Universität Darmstadt, 11-12, 2015

[13] Shields, Michael D and Zhang, Jiaxin, *The generalization of Latin hypercube sampling*, Reliability Engineering & System Safety, V.148, Elsevier 96–108, 2016,

[14] Cavazzuti, Marco, *Optimization Methods.From Theory to Design Scientific and Technological Aspects in Mechanics*, Springer, 13–42, 2013

[15] Rasmussen, Carl Edward and Williams, Christopher, *Gaussian Processes for Machine Learning*, the MIT Press, 2006, ISBN 026218253X, 2006 Massachusetts Institute of Technology. http://www.gaussianprocess.org/gpml/chapters/2018-07-13

[16] Strappini, Francesca, Gilboa, Elad, Pitzalis, Sabrina, Kay, Kendrick, McAvoy, Mark, Nehorai, Arye and Snyder, Abraham, *Adaptive Smoothing Based on Gaussian Processes Regression Increases the Sensitivity and Specificity of fMRI Data*, Human Brain Mapping, 2016, Wiley Periodicals, Inc. Chapter-Materials and Methods-GP Regression https://www.ese.wustl.edu/~nehorai/paper/Gilboa_Adaptive_Smoothing_HBM_2016.pdf2018-07-13

[17] Osborne, Michael, *Gaussian Processes for Prediction*, Robotics Research Group, 2007, Department of Engineering Science, University of Oxford, Chapter 3.2 http://www.robots.ox.ac.uk/~mosb/public/pdf/1133/PARG-07-01.pdf2018-07-13

[18] Snelson, Edward, *Flexible and efficient Gaussian process models for machine learning*, M.A., M.Sci., Physics, University of Cambridge, UK (2001), Gatsby Computational Neuroscience Unit, University College London, Chapter 1.4 http://www.gatsby.ucl.ac.uk/~snelson/thesis.pdf2018-07-13

[19] Roberts, Stephen, Osborne, Michael, Ebden, Mark, Reece, Steve, Gibson, Neale and Aigrain, Suzanne, *Gaussian Processes for Timeseries Modelling*, Department of Engineering Science, 2. Department of Astrophysics, 2012 University of Oxford. Chapter 3.1,3.3.1 http://www.robots.ox.ac.uk/~sjrob/Pubs/philTransA_2012.pdf2018-07-14

[20] Quiñonero, Joaquin, Rasmussen, Carl Edward and Williams, Christopher, *Approximation Methods for Gaussian Process Regression*, 2007, Chapter 4 http://homepages.inf.ed.ac.uk/ckiw/postscript/lskm_chap.pdf2018-07-14

[21] Duvenaud, Kristjanson, *Automatic Model Construction with Gaussian Processes*, 2014 University of Cambridge Chapter 2.3.3 https://www.cs.toronto.edu/~duvenaud/thesis.pdf2018-07-14

[22] Schiller, Friedrich, *Gaussian Process Regression for Uncertainty Estimation on Ecosystem Data*, Universität at Jena, Max-Planck-Institut für Biogeochemie Jena, 2011, Chapter 2.2, pp.25 http://www.clib-jena.mpg.de/theses/bgc/BGC11003.pdf2018-07-14

[23] Ebden, Mark, *Gaussian Processes for Regression: A Quick Introduction*, 2008, Chapter 3 https://www.robots.ox.ac.uk/~mebden/reports/GPtutorial.pdf2018-07-14

[24] MathWorks, *Gaussian Process Regression Models*, 2018, https://www.robots.ox.ac.uk/~mebden/reports/GPtutorial.pdf2018-07-14

[25] Ljung, Lennart *System Identification An Inverse Problem in Control Potentials and Possibilities of Regularization*, http://users.isy.liu.se/en/rt/ljung/ipa/sysid_mai.pdf, 2018-07-4, 2013

[26] Rahim, Yahay and Sahib, Shahrin, *Randomization techniques in privacy studies, Computing and Convergence Technology (ICCCT), 2012 7th International Conference*, IEEE, 524–526, 2012

[27] Guyon, Isabelle, *A scaling law for the validation-set training-set size ratio*, AT&T Bell Laboratories, Citeseer, 1–11, 1997,

[28] Caywood, Matthew, Roberts, Daniel, Colombe, Jeffrey, Greenwald, Hal and Weiland, Monica, *Gaussian Process Regression for Predictive But Interpretable Machine Learning Models: An Example of Predicting Mental Workload across Tasks*, Frontiers in human neuroscience, volume 10 , 2017, Frontiers

[29] Schulz, Eric, Speekenbrink, Maarten and Krause, Andreas, *A tutorial on Gaussian process regression with a focus on exploration-exploitation scenarios*, 2016, `https://www.biorxiv.org/content/biorxiv/early/2016/12/19/095190.full.pdf`, 2018-06-05

A

Appendix

A.1 Cholesky Decomposition

The Cholesky decomposition of a symmetric and positive definite matrix A decomposes A into a product of a lower triangular matrix L and its transpose according to:

$$LL^T = A \tag{A.1}$$

where L is called the Cholesky factor. The Cholesky decomposition is useful for solving linear systems with symmetric, positive definite coefficient matrix A. In order to solve $Ax = b$ for x, the triangular system, $Ly = b$ is first solved by forward substitution and then the triangular system $L^T x = y$ by back substitution. The solution is denoted as $x = {}_{L^T}\backslash({}_{L}\backslash b)$ where the notation ${}_{A}\backslash b$ is the vector x which solves $Ax = b$. Both the forward and backward substitution steps require $\frac{n^2}{2}$ operations, when A is of size $n \times n$. The computation of the Cholesky factor L is considered to be numerically extremely stable and takes $\frac{n^3}{6}$ time, so it is the preferred method whenever it can be applied. Note that the determinant of a positive definite symmetric matrix can be now be efficiently calculated by:

$$\det(A) = \prod_{i=1}^{n} L_{ii}^2 \quad \text{or} \quad \log(\det(A)) = 2\sum_{i=1}^{n} \log(L_{ii}) \tag{A.2}$$

where L is the Cholesky factor from matrix A [15].

A.2 All Inputs

The full list of inputs:

1. Engine Speed - Revolutions Per Minute (RPM) of the crankshaft.
2. Engine Torque - Crankshaft torque.
3. Injection Advance Angle - Crank angle degrees relative to Top Dead Center (TDC) of the compression stroke. TDC is the highest point of the piston during a compression stroke. It is related to the injection timing. The fuel should be injected close to TDC.
4. Nozzle Opening Pressure (NOP) - Pressure inside the unit injectors are controlled by the nozzle for fuel injection. The nozzle covers the holes through which diesel flows out from. The pressure which lifts up the nozzle is called the Nozzle Opening Pressure.

5. EGR Valve Position - Exhaust Gas Re-circulation (EGR) is a way of reducing the NO_x emissions through re-circulation of the exhaust gases into the engine cylinders. The formation of NO_x is highly dependent on higher temperatures, and it is then desired to limit the temperatures during the combustion. By recirculating the exhaust gases, which consist of inert gases, the oxygen concentration becomes diluted and the combustion process is slowed down. This results in lowered local temperatures and thus decrease the formation of NOx. EGR is mostly used when the NO_x concentration is reaching the maximum permitted level. The NO_x concentration is measured by the NO_x sensor, which is located at the exhaust gas outlet.
6. Throttle Position - For diesel engines, the throttle regulates the air flow into the engine and hence is a mean of controlling the engine power. Throttle Position Sensor (TPS) is used in measuring the throttle position of the engine. TPS is controlled by the ECU in Electronic Throttle Control (ETC).
7. Waste Gate Position - The air pressure that is too high inside the turbocharger is relieved via the waste gate valve, in order to prevent damage to the turbine blades. This valve is mostly controlled when the engine torque is high.

A.3 Optimization

The optimization can be described as a mapping problem that consists of mapping the control inputs to the optimal outputs that minimizes an objective function at each OP. A local optimization at each OP can be implemented [7].

Initially, the objective function will be the BSFC and control inputs (within the allowable range for each OP) are injection timing, injection pressure, EGR position and throttle position. The optimization problem is briefly described in Figure A.1.

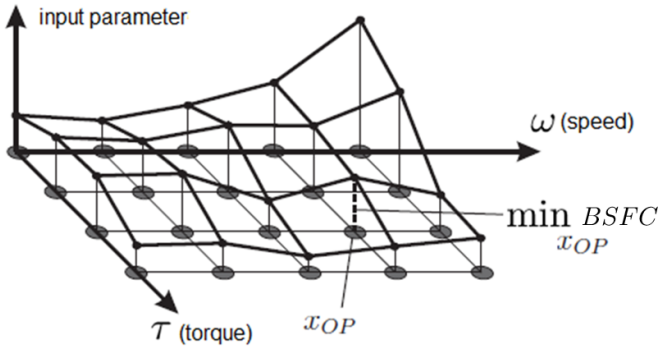

Figure A.1: Graphical representation of the optimization problem.

At each OP along the engine speed vs engine torque map in Figure A.1, an optimization is carried out to minimize the fuel flow at each OP. Thus, for each input on every OP along the speed vs torque map, a value of each input parameter that minimizes BSFC can be found. The solution (input values) corresponds to the global optima of BSFC at each OP. The resulting surface is called an engine map.

The optimization problem can be described according to:

$$\begin{aligned} \min\{BSFC((\tau,\omega)_i, IP, IT, EGR, Throttle)\} & \\ i = 1, ..., N_{op} & \\ \text{s.t} \quad IPmin_i \leq IP \leq IPmax_i & \\ ITmin_i \leq IT \leq ITmax_i & \\ EGRmin_i \leq EGR \leq EGRmax_i & \\ Throttlemin_i \leq Throttle \leq Throttlemax_i & \end{aligned} \tag{A.3}$$

where IP is the injection pressure, IT is the injection timing and fuel flow is simply the flow of fuel. Fuel flow is a function of engine speed (ω) and torque(τ), these variables are fixed at every OP while the other variables can be varied at each OP. EGR refers to the EGR position and $Throttle$ refers to the throttle position. $IPmin_i, ITmin_i$, $EGRmin_i, Throttlemin_i$, $IPmax_i, ITmax_i, EGRmax_i$ and

$Throttlemax_i$ contains the boundaries for the control inputs at each OP. The solution consists of the collection of input parameter values (IP, IT, EGR and Throttle) at each OP that combined will minimize BSFC.

There are numerous different optimization techniques available to solve the optimization problem. For an introduction into these techniques and the basics of optimization for engine calibration, see [7]. Note that there can be other objective functions than BSFC and there can also be multi-objective functions (for instance BSFC and NOx emissions), this would lead to a multi-objective optimization. A single objective-optimization can only optimize one objective, while all other objectives have to be integrated into constraints. (A.3) could for instance integrate NOx emissions by adding additional constraints.

A.4 The Complete Online Calibration Process

An online model based engine calibration process consists of three major components. The first two components are considered in the thesis but with the addition of the optimization component, the entire engine calibration process can be represented as:

- Design of Experiments (DoE).
- Modeling (system identification) of the system using GPR with online refinement.
- Online Optimization of input parameter settings.

The process also has a minor fourth step, which consists of generating engine maps of the solution from optimization (A.1), these engine maps will be stored on the Engine Control Unit (ECU) of the vehicle [7].
The three components are related to each other in the way described by Figure A.2.

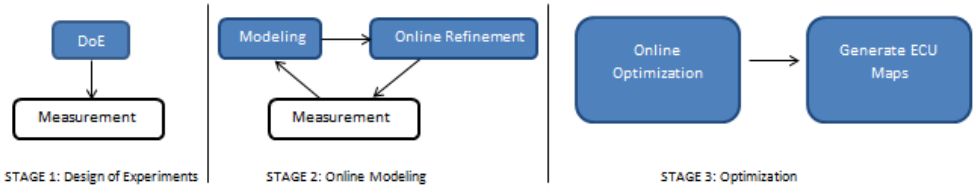

Figure A.2: The entire online model based engine calibration process.

In Figure A.2, stage 1 first starts with a DoE that is formulated on the computer which is then followed by measurements in the test cell. Stage 2 consists of an iterative loop between modeling, online refinement and measurements in the test cell. After one online refinement, additional measurements are taken at regions of interests. Stage 3 consists of the online optimization which tries to solve the optimization problem described in Section A.3. The online optimization is then

followed by the generation of the ECU maps which are illustrated in Figure A.1. Blue boxes occurs in the computer and white boxes in the test cell (but in this thesis executed by an ANN engine model).

A.5 Added DoE samples for the high torque model

An extra run was executed for the high torque model where more DoE samples was taken at each OP (no refinement was performed). The number of samples was $23 \times 5 = 115$. The resulting regression is shown in Figure A.3.

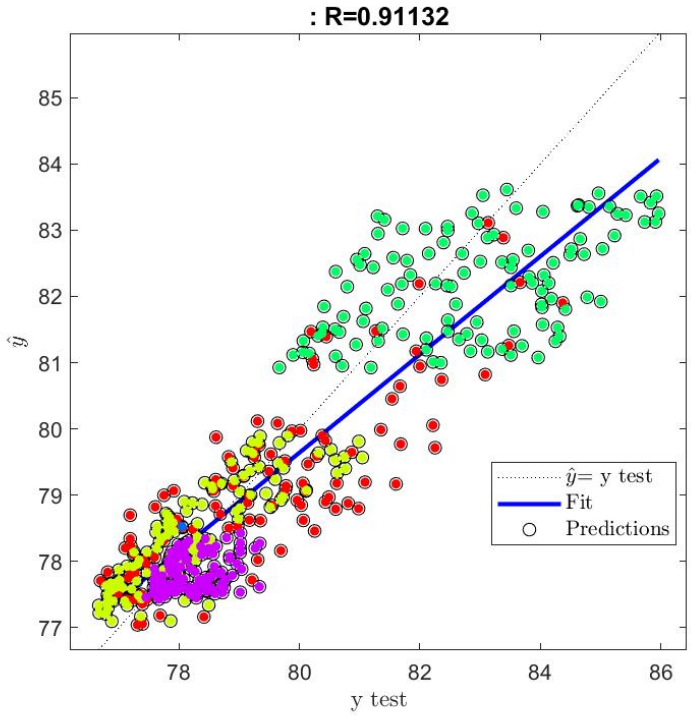

Figure A.3: Predictions from the GPR for the High Torque region with 115 DoE samples at each OP. Test outputs are on the x-axis and predicted outputs on the y-axis.

In Figure A.3, R is slope of the resulting line that tries to fit all points. The circles are the predictions. Each colour correspond to an OP of the test dataset.

The MSPE in scaled units is 0.2166. An improvement can be seen in the fit, however the MPSE has increased, but more samples are taken which may affects the spread of data and hence the MPSE.

Unscaled MPSE (115 DoE samples at each OP): 0.7398
Unscaled MPSE (23 DoE samples at each OP): 0.68