GENERATION OF INTERNAL COMBUSTION ENGINE MAPS AND SPARK TIMING PROFILES USING METAMODELS

by

ALI TAFRESHI

A dissertation submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN MECHANICAL ENGINEERING

2022

Oakland University
Rochester, Michigan

Doctoral Advisory Committee:

Zissimos P. Mourelatos, Ph.D. (Chair)
Brian P. Sangeorzan, Ph.D.
Dorin Drignei, Ph.D.
Jonathan Maisonneuve, Ph.D.

Dedicated

To my mother, Mary, and father, Dean, for their encouragement & continuous support.

To my brothers, Arash and Arya, who have been always there for me.

To my wife, Hoda, for her encouragement and love.

To my son, Dion, who brought immense joy to my life.

&

To my grandparents, Maman Bozorg, Agha, Mamani & Babaji.

ACKNOWLEDGMENTS

I would like to start by expressing my gratitude for my advisor, Dr. Zissimos Mourelatos for his inspiration and perpetual support. Without his guidance, this dissertation would not have been accomplished. Special thanks to my committee members, Dr. Brian Sangeorzan, Dr. Dorin Drignei, and Dr. Jonathan Maisonneuve, for reviewing my dissertation and providing invaluable feedback. I would like to deeply appreciate all the support, constructive advice and care I have received from Dr. Sangeorzan specially during the early years of my studies. Special thanks to Dr. Alkidas for his valuable input and support. Special thanks to my great colleagues, Mr. Kosanka, Dr. Nitu, Dr. Estefanous and Dr. Zheng for their support and encouragement. Thanks to my classmates and great friends, Dionysios, Vasiliki and Vasileios for their friendship and encouraging feedback.

I don’t have enough words to thank my parents, Mary and Dean. I would never be where I am today without their everlasting support, love, and encouragement. Thank you for all the trust and faith you had in me. Thanks to my brothers, Arash and Arya for giving me so much love care and confidence during these years while living apart.

Finally, I would like to thank my loving wife and best friend, Hoda, who has been by my side throughout this journey. Thank you for your confidence in me, encouragement, and your love. I truly and deeply appreciate all your effort to make life more comfortable for me and for our son, Dion Aran, whose presence brought more joy to my life

Ali Tafreshi

ABSTRACT

GENERATION OF INTERNAL COMBUSTION ENGINE FUEL MAPS AND SPARK TIMING PROFILES USING METAMODELS

by

ALI TAFRESHI

Adviser: Zissimos P. Mourelatos, Ph.D.

With the growth of computing technologies, many leading automotive companies tend to use simulation tools to reduce the number of actual engine testing for evaluating the performance of Internal Combustion (IC) engines. However, a high-fidelity engine model which is very complex and computationally demanding, is needed. In this dissertation, we present efficient and accurate metamodels to predict an engine fuel map and to also obtain the spark timing profile to generate a specified torque curve. Time-dependent Kriging metamodels using Singular Value Decomposition (SVD) and Nonlinear Autoregressive metamodels with Exogenous inputs (NARX) in conjunction with Neural Networks (NN) are developed and used. A sequential process was first developed to generate steady-state engine fuel maps using Kriging accounting for different engine characteristics at different operating conditions. The generated map predicts engine output parameters such as Brake Mean Effective Pressure (BMEP) and fuel flow rate. The Kriging metamodels are created sequentially to ensure acceptable accuracy with a small number of expensive engine simulations. Two optimization

problems are solved for full load and part load conditions, respectively. We demonstrate that the estimated fuel map is of high accuracy compared to the actual map.

The internal combustion engine is a source of unwanted vehicle vibration produced by engine mount forces which depend on the engine torque profile during a transient tip-in or tip-out maneuver. A methodology was also developed to obtain the desired engine torque profile to minimize the unwanted vibration by controlling a set of engine calibration parameters. A set of design coefficients defining a spark timing profile and the corresponding engine torque profiles are used to construct time-dependent metamodels using SVD and Kriging. The accuracy of the approach is demonstrated using GT-Power engine simulations. In addition, we developed a time-dependent NARX-NN metamodel to predict engine spark timing and cylinder pressure profiles corresponding to a desired torque profile. The NARX-NN metamodel predicts the spark timing accurately using a very small number of engine simulations.

TABLE OF CONTENTS

TABLE OF CONTENTS-Continued

TABLE OF CONTENTS-Continued

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES—Continued

NOMENCLATURE

IC	Internal Combustion
DOE	Design of Experiments
TDC	Top Dead Center
BDC	Bottom Dead Center
BMEP	Brake Mean Effective Pressure
WOT	Wide Open Throttle
NARX	Nonlinear Autoregressive Exogenous
SI	Spark Ignited
MBT	Maximum Brake Torque
N	Engine Speed
p	Pressure
V	Volume
V_d	Displacement volume
W	Work transfer
W_n	Net indicated work per cycle
W_i	Gross indicated work per cycle
W_p	Pumping work per cycle
W_f	Friction work per cycle
W_b	Brake work per cycle
A	Area
s	Stroke

P_b	Brake power
n_{cyl}	Number of cylinders
n_R	Number of crank revolutions per power stroke
T	Torque
$\dot{m}_f$	Fuel mass flow rate
$BSFC$	Brake specific fuel consumption
LHV	Lower heating value
EVO	Exhaust valve open
EVC	Exhaust valve close
IVO	Intake valve open
IVC	Intake valve close
MAP	Manifold air pressure
EGR	Exhaust Gas Recirculation
η_v	Volumetric efficiency
$\eta_{f,ig}$	Indicated fuel conversion efficiency
η_m	Mechanical efficiency
AFR	Air Fuel Ratio
MEP	Mean effective pressure
sfc	Specific fuel consumption
Q_{HV}	Heat loss
ρ_a	Air density

$\bar{S}_p$	Mean piston speed
RBF	Radial basis function
SVD	Singular value decomposition
MARS	Multivariate Adaptive Regressions Spline
F	Function
BLUP	Best linear unbiased predictor
MSE	Mean square error
MLE	Maximum Likelihood Estimate
AI	Artificial intelligence
ADALINE	Adaptive linear element
NN	Neural Network
LSTM	Long-short term memory
SGD	Stochastic gradient decent
PCA	Principal component analysis
LDDN	Layered digital dynamic network
VVA	Variable valve actuation
VVT	Variable valve timing
ICL	Intake centerline
ECL	Exhaust centerline
CA	Crank angle
RGF	Residual gas fraction

CA50	50% of the fuel mass burned
CoV	Coefficient of Variation
R^2	Coefficient of determination
ARE	Average relative error
Spk	Spark timing
TDL	Tapped Delay Line
deg	Degree
aTDCF	After top dead center firing
m	Mass
c	Damping coefficient
k	Spring coefficient
$F(t)$	Force
ω_n	Undamped frequency
ς	Damping factor
QMC	Quasi-Monte Carlo
PCE	Polynomial chaos expansion
KL	Karhunen-Loeve
PDF	Probability density function
a	Scale parameter
b	Shape parameter
μ_F	Mean

σ_F	Standard deviation
SK_F	Skewness
Ku_F	Kurtosis
ρ_{FF}	Autocorrelation function
τ	Relative time

CHAPTER ONE

INTRODUCTION

An IC engine is a mechanical system which converts chemical energy (fuel energy) to mechanical energy and produces work through thermodynamic processes. IC engines can be used in automobiles, ships, airplanes, power plants, etc. Nowadays, they are used in more than 90 percent of all produced vehicles. All IC engines are designed differently, and their performance is tested to make sure they pass certain criteria and regulations. The process of engine testing is very expensive and time consuming.

With advancements in computing technologies, the automotive industry tends to use physics-based simulation tools to design and test engines. This significantly reduces engine design cost and saves time. However, because of the complexity of many engine models a significant amount of time is devoted to develop them. To overcome this challenge, one can construct an approximate model, known as a metamodel, using a set of observations obtained from simulation at strategically chosen design points which have space filling properties. These points are obtained using the so-called Design of Experiments (DOE).

In this research, we propose a sequential process to generate a steady-state engine fuel map using Kriging metamodels. The latter account for different engine characteristics such as load and fuel consumption at different operating conditions. We also propose two different methodologies to determine the spark timing profile to achieve a desired engine torque profile during a transient tip-out maneuver. Sections 1.1 through 1.3 provide an introduction to four-stroke IC engines, DOE, and metamodels.

1.1 Four-Stroke Internal Combustion Engine

IC engines are systems that deliver mechanical work through a thermodynamic process. The mechanical work is produced by releasing fuel energy through fuel oxidization (combustion process) and converting it to thermal energy. The working fluid is a mixture of fuel and air before combustion and the byproduct after combustion is burned gas. In reciprocating engines (Figure 1.1), where the piston travels upwards and downwards in the combustion chamber, the developed gas pressure through the combustion process applies a force on the piston which is then transmitted to the crankshaft through the piston pin and the connecting rod [1]. A complete work cycle includes four strokes (Figure 1.1).

In the intake stroke, as the piston travels downwards from Top Dead Center (TDC) to Bottom Dead Center (BDC), fresh mixture of fuel and air flows into the cylinder through intake valves. In the compression stroke, the mixture is compressed while the piston travels upwards and the valves are closed. At the end of the compression stroke, combustion takes place. In the expansion stroke, the high pressure generated by combustion pushes the piston downwards and forces the crankshaft to rotate. Finally, in the exhaust stroke, the burned gases exit the cylinder through an opened exhaust valve as the piston travels upwards.

Internal combustion engines can be operated with high energy density fuels. The combustion efficiency can reach up to approximately 50% of the fuel energy that can be transferred into mechanical work. Due to their flexible designs, they can be used in a range of different applications such as in motor vehicles, airplanes, railways, ships, etc. However, there are environmental concerns associated with IC engines. The spark

ignition and diesel engines are one of the main sources of air pollution. The engine exhaust gas contains nitrogen oxides (NOx), carbon monoxide (CO) and unburned or partially burned hydrocarbon (HC) which are unfriendly to the environment and toxic for humans. Carbon dioxide (CO2), which is known as greenhouse gas, is the product of combusting fuel. Its high concentration in the atmosphere leads to global warming. For this reason, a lot of research has been conducted on ways to reduce these emissions while maintaining or increasing the efficiency of IC engines.

1.2 Design of Experiments

In recent years, the use of computer aided engineering has been a useful tool at every engineering sector using finite element analysis or computational fluid dynamics, for example. However, the effectiveness of virtual product prototypes is often hindered by the excessive computational cost of complex engineering simulations. A way to reduce the computational cost is to construct metamodels (model(s) of a model) [2], which should be sufficiently accurate to replace the computationally expensive simulation and analysis codes. The metamodel accuracy largely depends on the selected experimental design points which comprise the so-called Design of Experiments. A good DOE should have space filling properties providing a set of points that are uniformly scattered in the design space in order to maximize the extracted information from the design space. A DOE with good space filling properties can be then used to construct an accurate metamodel. The better the space filling properties of a DOE, the more accurate the metamodel will be.

1.3 Basics of Metamodeling

In today's scientific world many physical experiments can be too expensive or time consuming to conduct. Therefore, complicated physics-based models are necessary. Each of these models have multiple inputs and an output as:

$$y = f(\boldsymbol{x}), \qquad \boldsymbol{x} = \{x_1, x_2, \dots, x_s\} \in \boldsymbol{T}, \tag{1.1}$$

where y is the output, $\boldsymbol{x}$ is a vector of input variables, the function f can have an analytical formula, and $\boldsymbol{T}$ is the input design space [3]. Scientists and engineers use models to predict the behavior of systems under different inputs. Thus, the computer models become a vital part in investigating a complicated physical phenomenon.

Sometimes the physical models are very time consuming to conduct. Therefore, one of the goals is to find a proper approximate model

$$y \approx g(\boldsymbol{x}), \tag{1.2}$$

which is fast to run, accurate enough, and yields insight into the relationship between y and $\boldsymbol{x}$. Such an approximate model is called a "model of the model" or "metamodel" [3]. Since the metamodel is easier to compute and has an analytical formula, the g function can replace the actual function f for many applications. The construction of the metamodel is an interpolation problem and its accuracy depends on the Design of Experiments (DOEs) space filling properties.

1.4 Dissertation Outline

We begin in Chapter Two by describing the fundamentals of spark ignited gasoline engines, Design of Experiments (DOE), and conventional metamodeling including time dependent metamodeling techniques. DOE methods combined with

metamodeling techniques can be used in various engineering problems to provide not only accurate results but also reduce the time of experiments or simulations, significantly.

In Chapter Three, we introduce an efficient process to create fuel maps at full load and part load conditions utilizing a simulation tool and Kriging as a conventional metamodel. Fuel maps provide information about the engine characteristics which depend on the engine speed and load. The load is usually expressed in terms of the Brake Mean Effective Pressure (BMEP). An engine fuel map has two regions known as full load and part load. If an engine operates at full load, the throttle is wide open (WOT), providing the highest BMEP throughout the speed range. In part load (partially open throttle valve), it is desired to operate with the minimum fuel consumption while a targeted load is met at any engine speed.

In Chapter Four, we address clunk noise and the independent factors which can be controlled to reduce it in the driveline. We propose and compare two efficient methods to predict a spark timing profile corresponding to a desired torque profile in order to reduce clunk noise in vehicles. In the first method, we use the Singular Value Decomposition (SVD) method combined with Kriging to accurately predict the required spark timing profile and then validate the prediction. In the second method, we utilize the neural networks in a Nonlinear Autoregressive model with Exogenous inputs (NARX) combined with a feedforward Neural Network to extrapolate accurately a spark timing profile.

Finally, Chapter Five summarizes the main research contributions of this dissertation and outlines future work.

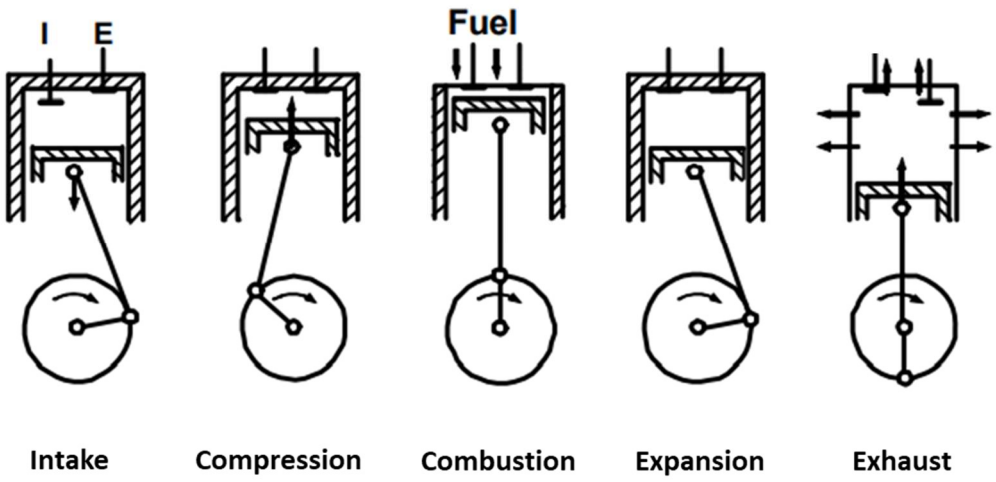

Figure 1.1: 4-stroke reciprocating engine [1]

CHAPTER TWO

RESEARCH BACKGROUND AND LITERATURE REVIEW

2.1 Overview of Spark Ignited (SI) Combustion Engines

The SI engines are four-stroke open-cycle engines in which the fuel-air mixture is combusted in the combustion chamber using a spark from a spark plug. In conventional SI engines the fuel and air are mixed outside the cylinder, either by a carburetor or by a fuel injection system at the intake manifold or ports before entering the cylinder. In order to have a stable combustion, the ratio of air mass flow to fuel mass flow must be held approximately around 14-15. The throttle valve controls the combined fuel and air flow, and thus the engine output. The maximum power at any engine speed is achieved when the throttle valve is wide open allowing maximum air flow to the cylinder. The throttle valve is partially open when lower power is required [4]. In sophisticated engines the fuel is injected directly into the combustion chamber and is mixed with fresh air in the cylinder. The mixture formation is determined by the time available between the injection event and the ignition instance [1].

The fluid mixture along with the residual gases from the previous cycle enters the cylinder when the intake valve is open during the intake stroke and then it is compressed during the compression stroke. Just before the firing TDC, the combustion process starts by an electric discharge across the spark plug and a turbulent flame develops and propagates across trapped air-fuel mixture and the residual gas in the cylinder. The flame stops propagating and extinguishes at the combustion chamber wall. During the combustion process the in-cylinder pressure increases which results in mechanical work

and generates torque. There is an optimum spark timing, for the so-called Maximum Brake Torque (MBT), which results in maximum torque depending on the in-cylinder fuel-air mixture. Any spark timing before or after MBT reduces the engine output [4].

The exhaust valve starts to open before BDC during the expansion stroke. The exhaust gases exit the cylinder during the exhaust stroke. The intake valve opens just before TDC where the gas exchange occurs. The exhaust valve closes just after gas exchange TDC. The time of opening and closing of intake and exhaust valves has significant effect on the engine efficiency, and thus fuel consumption [5-7].

2.1.1 Important Engine Characteristics

Characteristic values are useful for design engineers who are interested in evaluating the engine performance as well as assessing and comparing different engines. Figure 2.1 shows a schematic of a cylinder and a diagram known as the p-V diagram. The latter illustrates the course of in-cylinder pressure for the entire 4-stroke cycle.

The output work resulted by gas force in a cylinder can be obtained as

$$dW = p * A * ds, \tag{2.1}$$

where W is the output work, p is the in-cylinder pressure, A is the top piston surface area and s is the engine stroke.

Hence, the work in a complete cycle can be calculated as shown in equation (2.2) where V is the engine displacement volume,

$$dW = \oint p * dV \tag{2.2}$$

The net indicated work per cycle, W_n, is the work delivered to the piston over the entire cycle. It is given by

$$W_n = W_i + W_p \tag{2.3}$$

where, W_p, is the pumping work done over the exhaust and intake strokes (typically negative) and the Gross Indicated work per cycle, W_i, is the work delivered to the piston during the compression and expansion strokes. The available useful work at the crankshaft is the brake work. It is obtained by subtracting the friction work, W_f, from the net work,

$$W_b = W_n - W_f. \tag{2.4}$$

The friction work is due to engine friction and friction from the driving engine accessories. The Brake Mean Effective Pressure (BMEP) is a valuable measure of the engine's capacity to do work. It represents the brake work normalized by the displacement volume, V_d. In other words, the BMEP is independent of the engine displacement and it can be expressed as

$$BMEP = \frac{W_b}{V_d}. \tag{2.5}$$

The engine output power can be expressed as

$$P_b = \frac{W_b * N * n_{cyl}}{n_R} = \frac{BMEP * V_d * N * n_{cyl}}{n_R} \tag{2.6}$$

where N is the engine speed, n_{cyl} is the number of cylinders and n_R is the number of crank revolutions for each power stroke per cylinder. For a four-stroke engine, n_R is equal to 2. The engine brake power, P_b, can be also calculated by multiplying the engine torque, T, by the angular speed,

$$P_b = 2 * \pi * N * T. \tag{2.7}$$

The torque is a measure of the engine's ability to do work while power is the rate at which the work is done.

In engine test cells, the fuel consumption is measured using the fuel mass flow rate, $\dot{m}_f$. Another useful parameter is the Brake Specific Fuel Consumption ($BSFC$) which expresses how efficiently an engine uses the fuel supplied to produce useful work. The $BSFC$ can be expressed as

$$BSFC = \frac{\dot{m}_f}{P_b} \tag{2.8}$$

with typical units in $\frac{g}{KWh}$. The engine efficiency is given by

$$\eta = \frac{P_b}{\dot{m}_f * LHV} = \frac{1}{BSFC * LHV} \tag{2.9}$$

where LHV is the lower heating value of the fuel which defines the fuel energy content [8-10,97-98].

2.1.2 Gas Exchange and Flow in the Combustion Chamber

The gas exchange is a process taking place during the exhaust and intake strokes and it has a significant effect on the engine performance (e.g. power and $BSFC$) and also on emissions. The intake and exhaust valves control the engine breathing during gas exchange [1]. The exhaust gases must be removed from the cylinder through exhaust valves while fresh air enters the cylinder via intake valves. The following requirements usually hold for the exhaust and intake valves:

- Large opening cross section,
- Quick opening and closing,
- Optimal flow design,
- High sealing during compression and expansion, and

- High durability.

The valve timing plays an important role in engine performance. It specifies the start of opening and closing of the intake and exhaust valves as is shown in Figure 2.1. The influence of valve timing on energy losses and engine performance is controlled by the following events [8-10]:

An early Exhaust Valve Opening (EVO) results in high losses in expansion work by decreasing the exhaust work. In contrast, a late EVO results in low losses in expansion work by increasing the exhaust work. The Intake Valve Closing (IVC) has a significant influence on cylinder filling, and therefore, on engine torque characteristics. An early IVC is desirable for high torque at lower speeds. However, it causes filling losses at maximum engine speed. A late IVC causes high rated power with filling losses at low speeds. A late IVC is desirable for racing engines.

The Intake Valve Open (IVO) and the Exhaust Valve Close (EVC) usually overlap. With a large valve overlap, the fresh charge escapes the cylinder through an exhaust valve without participating in combustion reducing therefore, the efficiency of natural aspirated engines. This is the so-called scavenging loss. On the other hand, at full load condition, the residual gas is scavenged resulting in higher cylinder filling, and therefore, higher power. Increasing the valve overlap at part load increases the residual gas which can reduce the gas exchange work and also NOx emissions.

2.1.3 Fuel Map (Engine map)

One way to illustrate the operating characteristics of an IC engine over its entire load and speed range is to plot various engine parameters on an iso-plot with BMEP or engine torque on Y axis and engine speed on X axis. Some of the engine parameters that

can be shown on Z axis are BSFC, fuel flow rate, manifold air pressure (MAP), exhaust temperature, (EGR percentage, boost pressure if applicable), etc.

In general, a fuel map is generated by coupling the engine to a dynamometer in a test cell and running the engine using a regular grid of load and engine speed values. Each combination of load and speed is an operating condition. Some of the engine parameters at each operating condition, such as fuel flow rate or air flow rate, BMEP and torque, are measured. Other parameters, such as BSFC, are calculated using the measured data.

Figure 2.2 shows a fuel map example for a natural aspirated gasoline engine. The thick black curve represents the Wide-Open-Throttle (WOT) curve at full load in which the maximum possible BMEP is achieved for each engine speed. At WOT, the throttle valve is fully open allowing maximum air flow into the engine intake manifold. Of course, maximum BMEP at any engine speed is desired. For this specific example, the maximum achievable load is around 12.5 bar BMEP at 4500 rpm. Due to hardware limitations and the engine characteristics, it is not possible to operate beyond this curve (i.e. increase the BMEP further for a given rpm). The maximum BMEP occurs in the mid-speed range.

The area below the WOT curve indicates that the engine operates at part load in which the throttle is partially open to control the air-fuel mixture and to produce a certain load at a specific engine speed. In part load, the lowest fuel consumption at any engine speed and load is desired. The fuel consumption is measured as fuel mass per time $\dot{m}_f$. A more useful parameter is the Brake Specific Fuel Consumption (BSFC). The BSFC is the fuel flow rate per unit power output ($\frac{\dot{m}_f}{P}, [g/KWh]$), which measures how efficiently an

engine is using the supplied fuel to produce work [4]. Obviously, low values of BSFC are desirable. For this example, the minimum BSFC occurs around 3000 rpm and 10 bar BMEP. The minimum BSFC occurs at a lower speed and under part load.

An engine map can also be presented in terms of changes in volumetric efficiency, η_v, gross indicated fuel conversion efficiency, $\eta_{f,ig}$, and mechanical efficiency, η_m as Air Fuel Ratio (AFR), EGR (if applicable), and heat loss and friction change [11-13]. The Mean Effective Pressure (MEP), and the Specific Fuel Consumption (SFC) for four-stroke cycle engines can be expressed as in Equations (2.10) and (2.11), respectively. The Q_{HV} and ρ_a in equations (2.10) and (2.11) are lower heating value of the fuel and the inlet air density, respectively.

$$MEP = \eta_{f,ig} * \eta_v * Q_{HV} * \rho_a * \frac{Fuel}{Air} \tag{2.10}$$

$$sfc = \frac{1}{\eta_{f,ig} * Q_{HV}} \tag{2.11}$$

The full load curve varies with changes in the volumetric efficiency η_v, the reduction of η_m as the mean piston speed $\bar{S}_p$ increases, and the increase of $\eta_{f,ig}$ as $\bar{S}_p$ increases due to decreasing heat transfer per cycle. For the part load condition, starting at the minimum BSFC point, the following holds.

- Increasing speed at constant load results in an increase of BSFC due to the increasing FMEP at higher speeds which decreases mechanical efficiency η_m.
- Decreasing speed at constant load results in a BSFC increase due to the increase of heat transfer (heat loss) per cycle, and due to the friction reduction which increases η_m. Note that any fuel enrichment to maintain combustion stability or to avoid exceeding certain temperature limits to protect hardware results in a BSFC increase.

- Increasing load at constant speed increases BSFC due to fuel enrichment. The latter is necessary to increase torque while air flow increase becomes limited.
- Decreasing load at constant speed increases BSFC due to increasing friction and heat loss.

2.2 Overview of Metamodeling Techniques

The general form of a metamodel can be expressed as [3]:

$$g(\boldsymbol{x}) = \sum_{j=0}^{L} \beta_j f_j(\boldsymbol{x}), \tag{2.12}$$

where $\{f_0(\boldsymbol{x}), \dots, f_L(\boldsymbol{x})\}$ is a set of basis functions defined on the design space. The basis functions may be in the form of polynomials, splines, Radial Basis Functions (RBF) or the functions created by the covariance function in the Kriging model. The β_j's are unknown coefficients to be estimated [14-16].

In order to explain the metamodels mathematically it is important to use the following matrix notation

$$\boldsymbol{y} = (y_1, \dots, y_n)^T, \tag{2.13}$$

$$\boldsymbol{\beta} = (\beta_0, \dots, \beta_L)^T, \text{ and} \tag{2.14}$$

$$\boldsymbol{f}(\boldsymbol{x}) = \{f_0(\boldsymbol{x}), \dots, f_L(\boldsymbol{x})\}^T \tag{2.15}$$

$$\boldsymbol{F} = \begin{pmatrix} f_0(\boldsymbol{x}_1), \dots, f_L(\boldsymbol{x}_1) \\ f_0(\boldsymbol{x}_2), \dots, f_L(\boldsymbol{x}_2) \\ \vdots \quad \ddots \quad \vdots \\ f_0(\boldsymbol{x}_n), \dots, f_L(\boldsymbol{x}_n) \end{pmatrix} \tag{2.16}$$

where $\boldsymbol{y}$ is the vector of actual outputs, y, and n is the number of design sites, $\boldsymbol{\beta}$ is the vector of unknown coefficients, β, and L is the number of design space dimension, $\boldsymbol{f}(\boldsymbol{x})$

is the vector of estimated outputs as a function input data, $\boldsymbol{x}$, and $\boldsymbol{F}$ is the matrix of $\boldsymbol{f}(\boldsymbol{x})$ at different location of inputs.

A conventional metamodel predicts the value of a scalar function $y(\boldsymbol{x})$ where $\boldsymbol{x}$ is a multi-dimensional vector. Time-dependent metamodels can be constructed by combining a conventional metamodel, such as Kriging, with Singular Value Decomposition (SVD), NARX with a Neural network or NARX with Kriging. Different conventional metamodel techniques as well as time-dependent metamodeling are explained in detail next.

2.2.1 Polynomial Regression

Polynomial regression is a metamodeling technique that has been widely applied in engineering problems due to its ease in implementation and interpretation [17]. It is mainly used to obtain the overall trend of the actual function. A second order polynomial is the most used form of a polynomial regression. It can be expressed as

$$\tilde{y}(\boldsymbol{x}) = \beta_0 + \sum_{j=1}^{L} \beta_i x_i + \sum_{i=1}^{L} \sum_{j=1}^{L} \beta_{ij} x_i x_j \,, \tag{2.17}$$

where L is the number of independent parameters, $\boldsymbol{x}$ is the vector of independent parameters, and the βs are regression coefficients, obtained by the Least Square Estimation method. The latter estimates parameters by minimizing the squared of the deviations between observed values, $y(\boldsymbol{x})$, and their expected values, $\tilde{y}(\boldsymbol{x})$. The following minimization problem expresses the least square estimation

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left\{ y_i - \sum_{j=0}^{L} f_j(\boldsymbol{x}_i)\beta_j \right\}^2 \tag{2.18}$$

resulting in the following least square estimator

$$\widetilde{\boldsymbol{\beta}} = (\boldsymbol{F}^{\boldsymbol{T}}\boldsymbol{F})^{-1}\boldsymbol{F}^{\boldsymbol{T}}\boldsymbol{y} \tag{2.19}$$

where n is the number of design sites and $\boldsymbol{F}$ has full rank. The prediction of the least square estimate method is influenced by data located far away from majority of the design sites in the design space (outliers) which might reduce the prediction accuracy. Polynomial regression models have been used by a number of researchers [18-21] in designing complex engineering systems. While second-order polynomial models are most widely applied, they have limited capability to accurately model the actual function for a high number of independent parameters. Hence, higher-order polynomial models can be used to model more complicated functions. However, if the number of design sites, n, is large a higher order polynomial regression tends to oscillate between the input points. The undesirable oscillations are avoided by using other metamodels such as splines [22].

2.2.2 Splines

The splines method was introduced by Schoenberg and is widely used. In fact, this metamodeling technique laid the foundation for modern computer-aided design.

For a one-dimensional design space, the spline method uses n low-degree (e.g. cubic) polynomials between adjacent input points. If the design space is defined as

$$a = x_0 < x_1 < \cdots < x_n = b, \tag{2.20}$$

we can define a low-degree polynomial $q_0(x)$ from x_0 to x_1, then $q_1(x)$ from x_1 to x_2, and so on. The spline metamodel function $g(x)$ is obtained by fitting the q polynomials together into a single continuous curve [22]. The spline metamodel is also known as piecewise polynomial interpolation. In practice, it is recommended to use a cubic spline since it provides a continuous function as well as continuous first and second derivatives [23-26].

Spline interpolation can be used for multi-dimensional design space using a tensor product. However, the computational cost increases exponentially with the number of coefficients to be estimated. The Multivariate Adaptive Regressions Spline (MARS) was proposed by Friedman [27], in which the number of basis functions and the knot locations are determined adaptively using the data.

2.2.3 Radial Basis Functions (RBF)

The radial basis functions approach is common and can be used for a multi-dimensional space [28]. The RBFs are constructed using a linear combination of the basis functions as

$$\tilde{y}(\boldsymbol{x}) = \mu + \sum_{j=1}^{n} \beta_j f_j(\|\boldsymbol{x} - \boldsymbol{x}_j\|), \tag{2.21}$$

where μ is the mean of y. The matrix form of the above mathematical expression is

$$\tilde{\boldsymbol{y}} = \mu + \boldsymbol{F}\boldsymbol{\beta}, \tag{2.22}$$

where $\boldsymbol{F}$ is the vector of square (radial) functions. The basis functions $\boldsymbol{F}$ depend on the distance between an untried point x and the training points x_j, $j = \{1, 2, \dots, n\}$. The x_j are called the RBF centers. The vector coefficient $\boldsymbol{\beta}$ can be calculated using a least squares estimation. Table 2.1, shows some of the commonly used basis functions.

Some researchers have proposed modifications of the exact interpolation for smoothing purposes. Moody [29] and Bishop [30] proposed that the number of basis functions can be different from the number of data points. Also, the center of each basis function can be different from the sample points.

Gaussian radial basis functions is a popular form of RBF models [31]. The parameter θ_j in Gaussian RBFs is the reciprocal of the variance in a normal distribution.

Thus, it can be interpreted as the width of a Gaussian kernel. The mathematical form of Gaussian RBF is similar to a Kriging model [3].

2.2.4 Kriging

Kriging is a well-established non-parametric metamodel. It is very popular because of its acceptable accuracy for many applications and its ability to provide a prediction confidence range. The Kriging technique was first proposed by a South African geologist, D.G. Krige, in his master thesis [32] for analyzing mining data. This method has become popular in modeling computer experiments. Researchers such as Sacks and Cressie [33-34] extended the Kriging to deterministic computer experiments and Currin [35] developed a Bayesian interpolation for the Kriging methods which is used extensively in engineering problems.

Assume that $\boldsymbol{X}_i, i = 1,\cdots,n$ are design points over an s-dimensional domain T, and $y_i = y(\boldsymbol{X}_i)$ is the associated output to $\boldsymbol{X}_i$. The Gaussian Kriging model consists of a polynomial model (regression model) and a stochastic process and is defined as

$$y(\boldsymbol{X}) = \underbrace{\sum_{j=0}^{L} \beta_j f_j(\boldsymbol{X})}_{A} + \underbrace{z(\boldsymbol{X})}_{B}, \tag{2.23}$$

where part *A* represents the polynomial model and part *B* is a random error which is a Gaussian process with zero mean, $E[X]$=0, variance, σ^2, and correlation function $r(\boldsymbol{\theta}; \boldsymbol{X}_i, \boldsymbol{X}_j)$, i.e.

$$z(\boldsymbol{X}) = \sigma^2 r(\boldsymbol{\theta}; \boldsymbol{X}_i, \boldsymbol{X}_j). \tag{2.24}$$

The correlation function $r(\boldsymbol{\theta}; \boldsymbol{X}_i, \boldsymbol{X}_j)$ in Equation (2.25) is a pre-specified positive definite bivariate function of $z(\boldsymbol{X}_i)$ and $z(\boldsymbol{X}_j)$

$$r(\boldsymbol{\theta}; \boldsymbol{X}_i, \boldsymbol{X}_j) = Corr\left(z(\boldsymbol{X}_i), z(\boldsymbol{X}_j)\right) \tag{2.25}$$

and can be expressed as

$$r(\boldsymbol{\theta}; \boldsymbol{X}_i, \boldsymbol{X}_j) \equiv r(\boldsymbol{\theta}, d) \equiv \prod_{k=1}^{s} r_k(\theta_k, d_k) \tag{2.26}$$

where r_k is the correlation function defined in terms of the unknown correlation parameter, $\boldsymbol{\theta} = \{\theta_1, \dots, \theta_k\}$, and the distance d_k between the kth components of the two data points $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ $(d_k = (\boldsymbol{X}_i - \boldsymbol{X}_j)_k)$, and s is the number of input variables. For instance, the following function

$$r(\theta, d_k) = \exp\left\{-\sum_{i,j=1}^{s} \theta_k {d_k}^2\right\} \tag{2.27}$$

is the Gaussian correlation function.

The model of Equation (2.23) is referred to as the universal Kriging model. In the literature, the model

$$y(\boldsymbol{X}) = \mu + z(\boldsymbol{X}) \tag{2.28}$$

is referred to as the ordinary Kriging model, which is the most commonly used Kriging model in practice. The parameter μ is the average response of the output training points.

2.2.4.1 Prediction using Kriging

The Kriging prediction uses the best linear unbiased predictor theorem. A predictor $g(\boldsymbol{x})$ of $f(\boldsymbol{x})$ or $\tilde{y}(\boldsymbol{x})$ of $y(\boldsymbol{x})$ is expected to have the following properties:

- A predictor $\tilde{y}_0$ of $y_0 \equiv y\left(x^{(0)}\right)$ is a linear predictor if it is of the following form

$$\tilde{y}_0 = \sum_{i=1}^{n} c_{0i}(\boldsymbol{X}) y_i = \boldsymbol{c_0}^T \boldsymbol{y} \tag{2.29}$$

- It is also important to obtain predictors that are almost equal the quantity of interest everywhere on the design space. This property is expressed mathematically by requiring the predictor $\tilde{\boldsymbol{y}}$ to be unbiased. A predictor $\tilde{\boldsymbol{y}}$ is unbiased if

$$E[\tilde{\boldsymbol{y}}_0 - \boldsymbol{y}_0] = 0. \tag{2.30}$$

- A criterion is necessary for the comparison of the quality of competing predictors. A predictor $\tilde{\boldsymbol{y}}$ is called Best Linear Unbiased Predictor (BLUP) if it has the minimal Mean Squared Error (MSE) among all linear unbiased predictors.

$$MSE = \arg min\, E[(\tilde{\boldsymbol{y}}_0 - \boldsymbol{y}_0)^2] \tag{2.31}$$

The expectations in Equations (2.30) and (2.31) are taken with respect to the assumed joint Gaussian distribution of $\binom{y_0}{\boldsymbol{y}}$.

The problem of identifying the best predictor consists of finding the optimal weight vector $c_i(\boldsymbol{x})$ that satisfies the above three criteria; i.e.

$$\begin{gathered} c_0 \equiv \arg\min E[(\boldsymbol{c_0}^{\boldsymbol{T}}\boldsymbol{y} - y_0)^2] \\ \text{s.t. } E[\boldsymbol{c_0}^{\boldsymbol{T}}\boldsymbol{y} - y_0] = 0 \end{gathered} \tag{2.32}$$

The random error between $\boldsymbol{y}_0$ and its linear prediction can be shown by replacing $\boldsymbol{y}$ and $\boldsymbol{y}_0$ by their assumed expressions from Equation (2.23)

$$\begin{aligned} \tilde{y}_0 - y_0 &= \boldsymbol{c_0}^{\boldsymbol{T}}\boldsymbol{y} - y_0 \\ &= \boldsymbol{c_0}^{\boldsymbol{T}}(\boldsymbol{F\beta} + \boldsymbol{Z}) - \left(f_0{}^T\boldsymbol{\beta} + Z_0\right) \\ &= \boldsymbol{c_0}^{\boldsymbol{T}}\boldsymbol{Z} - Z_0 + (\boldsymbol{c_0}^{\boldsymbol{T}}\boldsymbol{F} - f_0{}^T)\boldsymbol{\beta}. \end{aligned} \tag{2.33}$$

The unbiased property requires that

$$\begin{aligned} E[\tilde{y}_0 - y_0\,] &= E\left[\boldsymbol{c_0}^{\boldsymbol{T}}\boldsymbol{Z} - Z_0 + (\boldsymbol{c_0}^{\boldsymbol{T}}\boldsymbol{F} - {f_0}^T)\boldsymbol{\beta}\right] \\ &= E[\boldsymbol{c_0}^{\boldsymbol{T}}\boldsymbol{Z} - Z_0] + \left(\boldsymbol{c_0}^{\boldsymbol{T}}\boldsymbol{F} - {f_0}^T\right)\boldsymbol{\beta} = 0 \end{aligned} \tag{2.34}$$

The first expected term in Equation (2.34) is zero due to the zero-mean assumption for the Gaussian process z

$$E[\boldsymbol{c_0}^{\boldsymbol{T}}\boldsymbol{Z} - Z_0] \equiv 0, \tag{2.35}$$

Therefore, the non-biased constraint reduces to

$$\boldsymbol{c_0}^{\boldsymbol{T}}\boldsymbol{F} + {f_0}^T = 0. \tag{2.36}$$

By combining Equations (2.32) and (2.36), the MSE prediction is obtained as

$$\begin{aligned} E[(\tilde{y}_0 - y_0)^2] &= E[(\boldsymbol{c_0}^{\boldsymbol{T}}\boldsymbol{Z} - Z_0)^2] \\ &= E\left[\boldsymbol{c_0}^{\boldsymbol{T}}\boldsymbol{Z}\boldsymbol{Z}^{\boldsymbol{T}}\boldsymbol{c_0} + {Z_0}^2 - 2\boldsymbol{c_0}^{\boldsymbol{T}}\boldsymbol{Z}Z_0\right] \\ &= \boldsymbol{c_0}^{\boldsymbol{T}}E[\boldsymbol{Z}\boldsymbol{Z}^{\boldsymbol{T}}]\boldsymbol{c_0} + E\left[{Z_0}^2\right] - 2\boldsymbol{c_0}^{\boldsymbol{T}}E[\boldsymbol{Z}Z_0]. \end{aligned} \tag{2.37}$$

Note that $E[\boldsymbol{Z}\boldsymbol{Z}^{\boldsymbol{T}}] = \sigma^2\boldsymbol{R}$, $E\left[{Z_0}^2\right] = \sigma^2$ and $E[\boldsymbol{Z}Z_0] = \sigma^2\boldsymbol{r}_0$. Therefore, the MSE prediction can be written as:

$$\begin{aligned} E[(\tilde{y}_0 - y_0)^2] &= \boldsymbol{c_0}^{\boldsymbol{T}}\sigma^2\boldsymbol{R}\boldsymbol{c_0} + \sigma^2 - 2\boldsymbol{c_0}^{\boldsymbol{T}}\sigma^2\boldsymbol{r}_0 \\ &= \sigma^2(1 + \boldsymbol{c_0}^{\boldsymbol{T}}(\boldsymbol{R}\boldsymbol{c_0} - 2\boldsymbol{r}_0)). \end{aligned} \tag{2.38}$$

The Best Linear Unbiased Prediction (BLUP) of $\tilde{y}_0$ is the solution of an equality constrained optimization problem stated by Equation (2.32). In order to solve this optimization problem a vector of Lagrange multipliers $\boldsymbol{\lambda}_0 \equiv \boldsymbol{\lambda}(\boldsymbol{x}^{(0)})$ is introduced. The Lagrangian function is applied to the equality constraint in (2.24) during the minimization of MSE in (2.26) and can be written as

$$L(\boldsymbol{c}_0, \boldsymbol{\lambda}_0) = \sigma^2\left(1 + \boldsymbol{c_0}^{\boldsymbol{T}}(\boldsymbol{R}\boldsymbol{c_0} - 2\boldsymbol{r}_0)\right) + \boldsymbol{\lambda}_0(\boldsymbol{c_0}^{\boldsymbol{T}}\boldsymbol{F} - {f_0}^T). \tag{2.39}$$

The gradient of Equation (2.39) with respect to $\boldsymbol{c}_0$ and $\boldsymbol{\lambda}_0$ is

$$\begin{cases} \nabla_{\boldsymbol{c_0}} L = 2\sigma^2(\boldsymbol{R c_0} - \boldsymbol{r}_0) + \boldsymbol{\lambda}_0 \boldsymbol{F} = 0 \\ \nabla_{\boldsymbol{\lambda}_0} L = (\boldsymbol{F^T c_0} - f_0) = 0. \end{cases} \tag{2.40}$$

The above equations form a linear system and can be written as

$$\begin{bmatrix} \boldsymbol{R} & \boldsymbol{F} \\ \boldsymbol{F^T} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{c_0} \\ \tilde{\boldsymbol{\lambda}}_0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_0 \\ f_0 \end{bmatrix} \; with \; \tilde{\boldsymbol{\lambda}}_0 = \frac{\boldsymbol{\lambda_0}}{2\sigma^2}. \tag{2.41}$$

By solving the system of equation above, the following $\tilde{\boldsymbol{\lambda}}_0$ and $\boldsymbol{c_0}$ are obtained

$$\begin{aligned} \tilde{\boldsymbol{\lambda}}_0 &= (\boldsymbol{F^T R^{-1} F})^{-1}(\boldsymbol{F^T R^{-1} r}_0 - f_0) \\ \boldsymbol{c_0} &= \boldsymbol{R^{-1}}\left(\boldsymbol{r}_0 - \boldsymbol{F}(\boldsymbol{F^T R^{-1} F})^{-1}(\boldsymbol{F^T R^{-1} r}_0 - f_0)\right) \\ &= \boldsymbol{R^{-1}}(\boldsymbol{r}_0 - \boldsymbol{F}\tilde{\boldsymbol{\lambda}}_0). \end{aligned} \tag{2.42}$$

The expression for the mean of $\tilde{y}_0$ is obtained by replacing the $\boldsymbol{c_0}$ from Equation (2.42) in Equation (2.29) as follows

$$\begin{aligned} \widetilde{y_0}(\boldsymbol{x}) &= \boldsymbol{c_0}^{\boldsymbol{T}}\, \boldsymbol{y} \\ &= (\boldsymbol{r}_0 - \boldsymbol{F}(\boldsymbol{F^T R^{-1} F})^{-1}(\boldsymbol{F^T R^{-1} r}_0 - f_0))\boldsymbol{R^{-1}}\, \boldsymbol{y} \\ &= \boldsymbol{r}_0{}^T \boldsymbol{R^{-1}}\, \boldsymbol{y} - [(\boldsymbol{F^T R^{-1} F})^{-1}\boldsymbol{F^T R^{-1} r}_0 + (\boldsymbol{F^T R^{-1} F})^{-1} f_0]^{\boldsymbol{T}} \boldsymbol{F^T R^{-1}}\, \boldsymbol{y} \\ &= f_0{}^{\boldsymbol{T}} \underbrace{(\boldsymbol{F^T R^{-1} F})^{-1}\boldsymbol{F^T R^{-1}}\, \boldsymbol{y}}_{\tilde{\boldsymbol{\beta}}} + \boldsymbol{r}_0{}^T \boldsymbol{R^{-1}}(\boldsymbol{y} - \boldsymbol{F}\, \underbrace{(\boldsymbol{F^T R^{-1} F})^{-1}\boldsymbol{F^T R^{-1} y}}_{\tilde{\boldsymbol{\beta}}}) \\ &= f_0{}^{\boldsymbol{T}}\tilde{\boldsymbol{\beta}} + \boldsymbol{r}_0{}^T \boldsymbol{R^{-1}}(\boldsymbol{y} - \boldsymbol{F}\tilde{\boldsymbol{\beta}}) \end{aligned} \tag{2.43}$$

Note that $\boldsymbol{R}$ is symmetric. Similarly, the expression for predicted MSE (variance) is obtained by introducing $\boldsymbol{u}_0 \equiv \boldsymbol{F^T R^{-1} r}_0 - f_0$ in Equation (2.38)

$$\begin{aligned} MSE &= \sigma^2(1 + \boldsymbol{c_0}^{\boldsymbol{T}}(\boldsymbol{R c_0} - 2\boldsymbol{r}_0)) \\ &= \sigma^2[1 + (\boldsymbol{r}_0 - \boldsymbol{F}(\boldsymbol{F^T R^{-1} F})^{-1}\boldsymbol{u}_0)^T \boldsymbol{R^{-1}}((\boldsymbol{r}_0 - \boldsymbol{F}(\boldsymbol{F^T R^{-1} F})^{-1}\boldsymbol{u}_0) \\ &\qquad - 2\boldsymbol{r}_0)] \end{aligned} \tag{2.44}$$

$$= \sigma^2[1 + \underbrace{(\boldsymbol{r}_0 - \boldsymbol{F}(\boldsymbol{F^T R^{-1} F})^{-1}\boldsymbol{u}_0)^T \boldsymbol{R^{-1}} (\boldsymbol{r}_0 - \boldsymbol{F}(\boldsymbol{F^T R^{-1} F})^{-1}\boldsymbol{u}_0)}_{\text{"(a-b)}^T\text{K(a+b)=a}^T\text{Ka-b}^T\text{Kb" since } K \text{ is symmetric}}]$$

$$= \sigma^2[1 - (\boldsymbol{r}_0{}^T \boldsymbol{R^{-1}} \boldsymbol{r}_0 - (\boldsymbol{F}(\boldsymbol{F^T R^{-1} F})^{-1}\boldsymbol{u}_0)^T \boldsymbol{R^{-1}} \boldsymbol{F}(\boldsymbol{F^T R^{-1} F})^{-1}\boldsymbol{u}_0]$$

$$= \sigma^2[1 - \boldsymbol{r}_0{}^T \boldsymbol{R^{-1}} \boldsymbol{r}_0 + \boldsymbol{u}_0{}^T \underbrace{(\boldsymbol{F^T R^{-1} F})^{-1}(\boldsymbol{F^T R^{-1} F})}_{\text{I}} (\boldsymbol{F^T R^{-1} F})^{-1}\boldsymbol{u}_0]$$

$$= \sigma^2[1 - \boldsymbol{r}_0{}^T \boldsymbol{R^{-1}} \boldsymbol{r}_0 + \boldsymbol{u}_0{}^T (\boldsymbol{F^T R^{-1} F})^{-1}\boldsymbol{u}_0].$$

2.2.4.2 Estimation of Unknown Parameters

The unknown parameters $\boldsymbol{\beta}$ and σ^2 in Equations (2.43) and (2.44) are obtained using the Maximum Likelihood Estimation (MLE) approach. MLE also estimates the parameters $\boldsymbol{\theta}$ of the probability density of y which is Gaussian. In MLE, the following likelihood function is maximized

$$L(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}) = (2\pi\sigma^2)^{-\frac{n}{2}} |\boldsymbol{R}(\boldsymbol{\theta})|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{F\beta})^T \boldsymbol{R}(\boldsymbol{\theta})^{-1}(\boldsymbol{y} - \boldsymbol{F\beta})\right\}. \quad (2.45)$$

This is equivalent to minimizing its opposite natural logarithm. Equation (2.46) expresses the log-likelihood of $L(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta})$

$$-logL(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}) = \frac{n}{2}\log(\sigma^2) + \frac{1}{2}\log|\boldsymbol{R}(\boldsymbol{\theta})| + \frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{F\beta})'\boldsymbol{R}(\boldsymbol{\theta})^{-1}(\boldsymbol{y} - \boldsymbol{F\beta}). \quad (2.46)$$

The maximum likelihood estimation problem can be expressed as:

$$\left(\tilde{\boldsymbol{\beta}}(\boldsymbol{\theta}), \widetilde{\sigma^2}(\boldsymbol{\theta})\right) = arg \min_{(\tilde{\boldsymbol{\beta}}, \widetilde{\sigma^2})} - logL(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}). \quad (2.47)$$

Solving Equation (2.47) with respect to $\boldsymbol{\beta}(\boldsymbol{\theta})\ and\ \sigma^2(\boldsymbol{\theta})$ eventually leads to the so-called generalized least squares estimate of the vector of $\tilde{\beta}$ and $\widetilde{\sigma^2}$. The gradient of Equation (2.45) with respect to β for a given $\boldsymbol{\theta}$ is

$$\nabla_{\beta}\, L(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2,\boldsymbol{\theta}) = 0$$

$$\frac{1}{\sigma^2}(\boldsymbol{F^T R(\theta)^{-1} y} - \boldsymbol{F^T R(\theta)^{-1} F\beta}) = 0 \tag{2.48}$$

which leads to the following least squares estimate of $\hat{\beta}_{mle}$

$$\tilde{\beta}_{mle}(\boldsymbol{\theta}) = (\boldsymbol{F}^T\boldsymbol{R}(\boldsymbol{\theta})^{-1}\boldsymbol{F})^{-1}\boldsymbol{F}^T\boldsymbol{R}(\boldsymbol{\theta})^{-1}y. \tag{2.49}$$

Similar to $\hat{\beta}_{mle}$, the maximum likelihood estimator for $\widetilde{\sigma^2}$ can be obtained by taking the derivative of the log-likelihood with respect to σ^2

$$\frac{\partial L(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2,\boldsymbol{\theta})}{\partial\sigma^2} = 0$$

$$-\frac{n}{2}\frac{1}{\sigma^2} + \frac{1}{2\sigma^4}(\boldsymbol{y} - \boldsymbol{F\beta})^{\boldsymbol{T}}\boldsymbol{R(\theta)^{-1}}(\boldsymbol{y} - \boldsymbol{F\beta}) = 0 \tag{2.50}$$

$$\widetilde{\sigma^2}(\boldsymbol{\theta}) = n^{-1}\big(\boldsymbol{y} - \boldsymbol{F}\tilde{\boldsymbol{\beta}}\big)^{\boldsymbol{T}}\boldsymbol{R(\theta)^{-1}}\big(\boldsymbol{y} - \boldsymbol{F}\tilde{\boldsymbol{\beta}}\big)$$

Note that the parameters $\tilde{\beta}$ and $\widetilde{\sigma^2}$ depend on $\boldsymbol{\theta}$. Plugging the two solutions of Equations (2.49) and (2.50) into Equation (2.46) provides a new expression that depends only on $\boldsymbol{\theta}$

$$\begin{aligned} -logL(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2,\boldsymbol{\theta}) &= \frac{n}{2} + \frac{n}{2}\log(2\pi) + \frac{n}{2}\log\left(\widetilde{\sigma^2}(\boldsymbol{\theta})\right) + \frac{1}{2}\log|\boldsymbol{R}(\boldsymbol{\theta})| \\ &= \frac{n}{2}\log(\psi(\boldsymbol{\theta})) + \frac{n}{2}(\log(2\pi) + 1) \end{aligned} \tag{2.51}$$

where the following reduced likelihood function is used

$$\psi(\boldsymbol{\theta}) = \widetilde{\sigma^2}(\boldsymbol{\theta})|\boldsymbol{R}(\boldsymbol{\theta})|^{1/n}. \tag{2.52}$$

The maximum likelihood estimate of $\boldsymbol{\theta}$ is obtained by the following global minimizer

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\theta}\psi(\boldsymbol{\theta}). \tag{2.53}$$

The global optimization problem in Equation (2.53) is solved by numerical global optimization techniques. In this research, the DACE MATLAB toolbox is employed [36]. It uses the BOXIMIN algorithm to solve the optimization problem of Equation (2.53).

2.2.5 Time-dependent Metamodeling using SVD

Metamodels can also be used to predict time-dependent functions using Kriging and Singular Value Decomposition (SVD).

SVD is a matrix decomposition method which allows the representation of a time-dependent function as a linear combination of time-dependent basis functions. The latter are obtained by an eigenvalue decomposition of a set of discretized time-dependent sample functions. SVD can be combined with Kriging to obtain a time-dependent metamodeling technique. In that, Kriging provides an interpolation between a set of scalar designs and the coefficients of the linear combination in SVD.

Figure 2.3 shows an example of an SVD-Kriging parametric time-dependent metamodeling for a case involving a two-dimensional design $\boldsymbol{d}$. Each design site results in a time-dependent function. An SVD-Kriging time-dependent metamodel is trained using a set of designs (green dots) and then it is used to obtain the time-dependent function corresponding to a design (red dot) which is not included in the training set.

We assume that the time-dependent function can be characterized (spanned) by m time-dependent sample functions in the time interval $t \in [t_{min}, t_{max}]$. The latter is discretized using a uniform grid of n discrete time instances. Each sample function depends on the parameter vector $\boldsymbol{D} = [d_1, d_2, \dots, d_L]$; i.e. the function is defined if the vector $\boldsymbol{D}$ is known. The m available sample functions correspond to m designs sites. The discretized m sample functions form a $m \times n$ response matrix $[X]$ as

$$[X] = \begin{bmatrix} x(D_1, t_1) & \cdots & x(D_1, t_n) \\ \vdots & \ddots & \vdots \\ x(D_m, t_1) & \cdots & x(D_m, t_n) \end{bmatrix}. \tag{2.54}$$

Each row of $[X]$ represents a discretized time function corresponding to a design $\boldsymbol{D}$. Each column corresponds to a particular discrete time. Using SVD, we can decompose the response matrix $[X]$ as

$$[X] = [U][S][V]^T, \tag{2.55}$$

where $[U]$ is a column-orthogonal $m \times m$ matrix in which each column is a left eigenvector of $[X]$, $[V]$ is an orthogonal $n \times m$ matrix of the right eigenvectors of $[X]$ and $[S]$ is a diagonal $m \times m$ matrix of the singular values of $[X]$ in decreasing order. Equations (2.56) to (2.58) show an expanded form of $[U]$, $[S]$, and $[V]$

$$[U] = \begin{bmatrix} u_{11} & \cdots & u_{1m} \\ \vdots & \ddots & \vdots \\ u_{m1} & \cdots & u_{mm} \end{bmatrix}, \tag{2.56}$$

$$[S] = \begin{bmatrix} s_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & s_{mm} \end{bmatrix}, \tag{2.57}$$

$$[V] = \begin{bmatrix} v_{11} & \cdots & v_{1m} \\ \vdots & \ddots & \vdots \\ v_{n1} & \cdots & v_{nm} \end{bmatrix}. \tag{2.58}$$

Note that we can retain only the *k* dominant singular values in Equation (2.57) and truncate the non-dominant singular values to zero. This improves efficiency, and more importantly reduces the dimensionality of the problem.

To obtain a set of basis functions in SVD which span any time-dependent function of parameters D, the space D is space-filled with *m* design sites using a DOE algorithm. The matrix $[X]$ is then constructed by discretizing the time functions corresponding to the m designs according to Equation (2.56).

In parametric time-dependent metamodeling (Figure 2.3), the time-dependent response for a design D_p which is not included in the m designs used for training, can be obtained by interpolating each mode (column) of matrix $[U]$. The vector $\{u_p\}$ is the interpolated row of matrix $[U]$ which corresponds to the design D_p. Kriging is used to obtain $\{u_p\}$ with high accuracy. The time-dependent response is then obtained as

$$\{x_p\} = \{u_p\}[S][V]^T, \tag{2.59}$$

where $\{x_p\}$ is the row vector of the discretized time-dependent function corresponding to the design point D_p. The latter can be viewed as an interpolated row of the response matrix $[X]$ corresponding to an interpolated row $\{u_p\}$ of matrix $[U]$.

In inverse parametric time-dependent metamodeling, the time-dependent response $\{x_p\}$ is known, and the design $\boldsymbol{D}$ is of interest. In this case, the interpolated row $\{u_p\}$ corresponding to $\{x_p\}$ is first calculated as

$$\{u_p\} = \{x_p\}[V][S]^{-1} \tag{2.60}$$

and then used as input to Kriging to interpolate the unknown design point. Figure 2.4 shows schematically the inverse parametric time-dependent metamodeling.

2.2.6 Time-dependent Metamodeling using NARX-net

Time-dependent metamodels can also be constructed using a combination of Neural Networks (NN) and Nonlinear Autoregressive models with Exogenous inputs (NARX), called NARX-net. In the following sections, we provide an introduction to neural networks and NARX. We also demonstrate the efficiency of NARX-net in developing time-dependent metamodels.

Neural Networks (or deep learning) is a machine learning approach enabling the creation of Artificial Intelligence (AI) systems using raw data [37]. Neural networks are composed of neurons connected to each other in multiple layers [38-41]. Among all layers there is an input layer, an output layer and potentially multiple hidden layers.

The earliest NNs were introduced in 1940s for cybernetics applications. During this time different linear models were developed which were motivated by neuroscience. McCulloch-Pitts neuron model [42] was the earliest linear model of the brain. Other well-known linear models were the perceptron [43] and the Adaptive Linear Element (ADALINE) [44] which were able to predict outputs given inputs by learning a set of weights. However, linear models have many limitations. For example, they cannot predict complex nonlinear problems accurately. Also, they cannot learn the Exclusive OR function, known as XOR function [55].

The next generation of NNs started in the 1980s with the "connectionist" approach. The general idea of this approach is that many simple computational units can result in intelligent behavior when connected together. Major accomplishments achieved during this period. The method of backpropagation [46] was introduced to train single or multi-hidden layer neural networks. Also, modeling sequences was another significant achievement during this period. Long short-term memory (LSTM) [47] network was developed to resolve some fundamental mathematical difficulties in modeling long sequences. Now, this method is widely used to predict dynamic profiles in different areas such as engineering and economy. However, in the connectionist period, the developed algorithms were very complex resulting in a computationally expensive analysis when a sizable amount of data is used.

The currently used term, neural network (deep learning) started about 2006 due to the need to analyze larger amounts of data using powerful computers and different techniques to train networks. Hilton [48] developed a neural network called a deep belief network which was able to be trained efficiently using a method called greedy layer-wise pretraining. His coworkers [49,50] helped to generalize this method using various test examples. With their findings, now researchers are able to train deeper neural network for much complex problems using powerful computers. Currently, different forms of neural networks are used for many applications. As an example, reinforcement learning is an extension of NNs which is widely used in engineering controls applications such as robotics and autonomy in the automotive industry [51, 52].

NNs, similar to other metamodeling techniques, optimize a compositional function

$$\underset{A_j}{\operatorname{argmin}}(f_M(A_M, \dots, f_2(A_2, f_1(A_1, x)) \dots) + \lambda_g(A_j)) \tag{2.61}$$

using stochastic gradient descent and back propagation algorithms. Each matrix A_k denotes the weights connecting the neural network from the kth to $(k+1)$th layer. This system of equations can be highly undetermined and is regularized by $g(A_j)$. The composition and regularization of Equation (2.61) is critical to generate expressive representations of the data and to prevent overfitting. In the following sub-sections, we will discuss one-layer and multi-layer NNs and how they are trained.

2.2.6.1 One-Layer Neural Networks

In order to demonstrate one-layer NN, we will use the "dogs and cats" example. The output $\boldsymbol{y}$ where

$$\boldsymbol{y} = \{dog, cat\} = \{+1, -1\} \tag{2.62}$$

labels each data vector with an output $\boldsymbol{y}\epsilon\{\pm 1\}$. In this case, the output layer is a single node. The goal is to determine a mapping so that each data vector $\boldsymbol{x}_j$ is labeled correctly by $\boldsymbol{y}_j$.

Figure 2.5 shows a single layer network for binary classification between dogs and cats. The output layer is a realization with $\boldsymbol{y}\epsilon\{\pm 1\}$. A linear mapping between the input image space and the output layer can be constructed for training data by solving $\boldsymbol{A} = \boldsymbol{Y}\boldsymbol{X}^T$.

The easiest mapping is a linear mapping between the input images $\boldsymbol{x}_j \epsilon \mathbb{R}^n$ and the output layer. This gives a linear system $\boldsymbol{A}\boldsymbol{X} = \boldsymbol{Y}$ of the form

$$\boldsymbol{A}\boldsymbol{X} = \boldsymbol{Y} \rightarrow [a_1\ a_2 \dots\ a_n]\begin{pmatrix} \vdots & \vdots & & \vdots \\ x_1 & x_2 & \cdots & x_p \\ \vdots & \vdots & & \vdots \end{pmatrix} = [+1\ \ +1 \dots -1\ \ -1] \tag{2.63}$$

where each column of the matrix $\boldsymbol{X}$ is a dog or cat image and the columns of $\boldsymbol{Y}$ are its corresponding labels. Note that $\boldsymbol{A}$ and $\boldsymbol{Y}$ are vectors, since the output layer is a single node. Therefore, our goal is to determine the matrix (vector) $\boldsymbol{A}$ with components a_j . The simplest solution is to take the pseudo-inverse of the data matrix $\boldsymbol{X}$

$$\boldsymbol{A} = \boldsymbol{Y}\boldsymbol{X}^T. \tag{2.64}$$

Therefore, we can use either a least square estimation or LASSO methods to solve the system of Equations (2.64). This example was coded in MATLAB [53] to show the dogs and cats classification performance of single NNs using least square estimation and LASSO. The results showed that both methods' performance was poor in terms of classifying dog and cat images.

2.2.6.2 Multi-Layer Networks and Activation Functions

In general, the following non-linear transformations can be used from input to output

$$y = f(\boldsymbol{A}, x) \tag{2.65}$$

where $f(.)$ is a predefined activation function (transfer function) which is either differentiable or piecewise differentiable. The most commonly used activation function is currently the $ReLU$, which we denote by $f(x) = ReLU(x)$. Some standard activation functions are shown in Table 2.2.

With a nonlinear activation function $f(x)$, or if there are multiple layers, standard linear optimization routines such as the pseudo-inverse and LASSO can no longer be used [53]. Instead, stochastic gradient descent and backpropagation techniques can be used to train the NNs; i.e. determine the value of weights in each matrix A_k of Equation (2.64).

The prediction performance of the dogs and cats classification example using a hyperbolic tangent (nonlinear) transfer function improved with an acceptable accuracy of 85%.

2.2.6.3 The Backpropagation Algorithm

In order to train a NN we need to determine the weights which can be calculated using optimization methods such as gradient decent or stochastic gradient decent for higher efficiency in more complex optimization problems [54].

The backpropagation algorithm is used to calculate the gradient used in gradient-based optimization problem to determine the weights of the network. It is based on the simple mathematical principle of the chain rule for differentiation. Figure 2.6 illustrates

the backpropagation algorithm for a one-node, one hidden layer network. This is a simple example to understand the backpropagation algorithm.

Given the functions $f(.)$ and $g(.)$ with weighting constants a and b (see Figure 2.6) the output error, E, between the predicted output y which depends on the functions $f(.)$ and $g(.)$ and the weights a and b, and the target output y_0 produced by the network can be computed as

$$E = \frac{1}{2}(y_0 - y)^2. \tag{2.66}$$

The objective is to find a and b to minimize the error, E. Hence, the minimization requires

$$\frac{\partial E}{\partial a} = -(y_0 - y)\frac{dy}{dz}\frac{dz}{da} = 0. \tag{2.67}$$

It is important to note that the compositional nature of the network along with the chain rule forces the optimization to backpropagate the error through the network. For instance, the terms $\frac{dy}{dz}\frac{dz}{da}$ show how this backpropagation occurs. The chain rule can be computed using $f(.)$ and $g(.)$. Backpropagation results in an iterative, gradient descent update rule as shown below

$$\begin{aligned} a_{k+1} &= a_k + \delta\frac{\partial E}{\partial a_k} \\ b_{k+1} &= b_k + \delta\frac{\partial E}{\partial b_k} \end{aligned} \tag{2.68}$$

where δ is the so-called learning rate and $\frac{\partial E}{\partial a}$ along with $\frac{\partial E}{\partial b}$ in Equation (2.68) can be explicitly computed using Equation (2.67). The sequential iteration algorithm is executed

to convergence. Below, we summarize the process of training a NN using the backpropagation algorithm:

- A NN is constructed with a labeled training set.
- A set of random initial weights are generated. It is important to know that we must not initialize the weights to zero. If the initial weights are equal to zero, after each update, the calculated weights of each neuron will be identical, since the gradients will be identical. Furthermore, NNs might stuck at local optima where the gradient is zero. To have a better chance to obtain the global minimum, random weight initialization is commonly used.
- The network produces an output y using training data set. Then, the derivatives with respect to each network weight is computed using backprop formulas (Equation (2.67)).
- The network weights are updated as in Equation (2.68) for a given learning rate δ.
- We return to step (3) and continue iterating until a maximum number of iterations is reached or convergence is achieved.

The backpropagation procedure can also be applied to a deeper net. Consider a network with M hidden layers labeled z_1 to z_m with the first connection weight a between x and z_1. The generalization of Figure 2.6 and Equation (2.67) is given by

$$\frac{\partial E}{\partial a} = -(y_0 - y)\frac{dy}{dz_m}\frac{dz_m}{dz_{m-1}}\ldots\frac{dz_2}{dz_1}\frac{dz_1}{da}. \tag{2.69}$$

A full generalization of backpropagation method including multiple layers and multiple nodes per layer is illustrated in Figure 2.7. The objective is to determine the matrix elements (weights) of each matrix A_j .

Denoting all the weights to be updated by the vector $\boldsymbol{w}$, where $\boldsymbol{w}$ contains all the elements of the matrices A_j illustrated in Figure 2.7, we have

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \delta \nabla E \tag{2.70}$$

where the gradient of the error ∇E, through the composition and chain rule, produces the backpropagation algorithm for updating the weights and reducing the error. Expressed in a component-by-component way

$$\boldsymbol{w}_{k+1}^j = \boldsymbol{w}_k^j + \delta \frac{\partial E}{\partial \boldsymbol{w}_k^j} \tag{2.71}$$

where the equation holds for the jth component of the vector $\boldsymbol{w}$. The term $\frac{\partial E}{\partial \boldsymbol{w}^j}$ produces the backpropagation through the chain rule.

2.2.6.4 The Stochastic Gradient Descent Algorithm

The gradient descent algorithm is used to determine the fitting coefficients, $\boldsymbol{\beta}$, in a nonlinear regression formula with the general form of

$$f(\boldsymbol{x}) = f(\boldsymbol{x}, \boldsymbol{\beta}) \tag{2.72}$$

by minimizing the error. In NNs, the parameters $\boldsymbol{\beta}$ are the network weights, so we can reformulate Equation (2.72) as

$$f(\boldsymbol{x}) = f(\boldsymbol{x}, \boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_M) \tag{2.73}$$

where the $\boldsymbol{A}_j$ are the connectivity matrices from one layer to the next NN layer. In other words, among M hidden layers $\boldsymbol{A}_1$ connects the first and second layers.

As mentioned earlier, the goal of training the NN is to minimize the error between the network prediction and the actual data. The standard RMSE is obtained as

$$\underset{A_j}{\operatorname{argmin}}\ E(\boldsymbol{A}_1, \boldsymbol{A}_2, \dots, \boldsymbol{A}_M) = \underset{A_j}{\operatorname{argmin}} \sum_{k=1}^{n} (f(x_k, \boldsymbol{A}_1, \boldsymbol{A}_2, \dots, \boldsymbol{A}_M) - y_k)^2 \tag{2.74}$$

where the minimization is performed by setting the partial derivative with respect to each matrix component to zero. In other words, we require $\frac{\partial E}{\partial (a_{ij})} = 0$ where $(a_{ij})_k$ is the ith row and jth column of the kth matrix ($\boldsymbol{k} = 1, 2, \dots, M$). This can be shown as the gradient $\nabla f(\boldsymbol{x})$ of the function with respect to the NN parameters. Note that $f(.)$ is the function evaluated at each of the n data points. The Newton-Raphson iteration scheme can be used to find the minimum using the following iterative scheme

$$\boldsymbol{x}_{j+1}(\delta) = \boldsymbol{x}_j - \delta \nabla f(\boldsymbol{x}_j) \tag{2.75}$$

where δ is a parameter determining how far a step should be taken along the gradient direction. In NNs, this parameter is called the learning rate.

To perform the computation of Equation (2.74) the Stochastic Gradient Decent (SGD) does not estimate the gradient in Equation (2.75) using all n data points. Instead, a single, randomly chosen data point, or a subset for batch gradient descent, is used to approximate the gradient at each step of the iteration. Thus, we can reformulate Equation (2.74) as

$$\begin{aligned} (\boldsymbol{A}_1, \boldsymbol{A}_2, \dots, \boldsymbol{A}_M) &= \sum_{k=1}^{n} E_k(\boldsymbol{A}_1, \boldsymbol{A}_2, \dots, \boldsymbol{A}_M) \\ E_k(\boldsymbol{A}_1, \boldsymbol{A}_2, \dots, \boldsymbol{A}_M) &= (f_k(\boldsymbol{x}_k, \boldsymbol{A}_1, \boldsymbol{A}_2, \dots, \boldsymbol{A}_M) - \boldsymbol{y}_k)^2 \end{aligned} \tag{2.76}$$

where $f_k(.)$ is the fitting function for each data point, and the elements of the matrices $\boldsymbol{A}_j$ are determined using the optimization process.

Now, the gradient descent iteration algorithm of Equation (2.75) can be reformulated as

$$\boldsymbol{w}_{j+1}(\delta) = \boldsymbol{w}_j - \delta \nabla f_k(\boldsymbol{w}_j) \tag{2.77}$$

where $\boldsymbol{w}_j$ is the vector of all network weights from $\boldsymbol{A}_j$ $(j = 1,2, \dots, M)$ at the jth iteration, and the gradient is computed using only the kth data point and $f_k(.)$. Thus, only a single data point is randomly selected and used. At the next iteration, another randomly selected point is used to compute the gradient and update the solution. The algorithm may require multiple passes through all the data to converge, but each step is now easy to evaluate versus the expensive computation of the Jacobian which is required for the gradient. If instead of a single point, a subset of points is used, then we have the following batch gradient descent algorithm

$$\boldsymbol{w}_{j+1}(\delta) = \boldsymbol{w}_j - \delta \nabla f_K(\boldsymbol{w}_j) \tag{2.78}$$

where $K \in [k_1, k_2, \dots, k_p]$ denotes the p randomly selected data points k_j used to approximate the gradient.

2.2.6.5 Nonlinear Autoregressive Networks with Exogenous Inputs (NARX)

NARX is a mathematical structure to predict the output of time-dependent problems. In recent years, many researchers have used NARX to model highly nonlinear systems such as heat exchangers [55], waste water treatment plants [56], time series [57]. Researchers have also combined conventional metamodels with NARX depending on the type of problems to solve. Some studies have been conducted by combining Kriging with NARX [58-60] and some other research works used a polynomial approximation with NARX [61]. But the majority of researchers have used a neural network with NARX [62,63].

Neural networks are widely used for static applications where time is constant. However, they can also be used to predict dynamic profiles when time varies. In static networks there is no feedback elements and the networks contain no delays. Hence, the output is calculated directly from the input. In other words, the output of the static networks is updated for the corresponding input. In dynamic networks on the other hand, the output depends not only on the current input, but also on the current or previous inputs and outputs. In other words, the dynamic network has memory. Hence, the dynamic networks are more powerful than static networks.

The dynamic networks can be classified into two categories. The first category includes networks which have only feedforward connections. If the network does not have any feedback connections, then only a finite amount of history affects the response. Figure 2.8 shows an example of dynamic feedforward NNs.

The second category includes networks which have feedback connections. These types of networks are called Nonlinear Autoregressive with Exogenous inputs (NARX) and typically they have longer response than feedforward-dynamic networks. Figure 2.9 shows a schematic of the NARX structure.

Equation 2.79 shows the mathematical expression of NARX model

$$\tilde{y}(t) = f(\tilde{y}(t-1), \tilde{y}(t-2), \dots, \tilde{y}(t-n_y), u(t-1), u(t-2), \dots, u(t-n_u)) \quad (2.79)$$

where the next value of the dependent output signal, $\tilde{y}(t)$, is calculated based on previous values of the output signal and previous values of an independent (exogenous) input signal $u(t)$. This implementation also allows for a vector ARX model, where the input and output can be multidimensional.

The MATLAB software provides a toolbox which is designed to train a class of network called the Layered Digital Dynamic Network (LDDN) [64]. Any network that can be structured in the form of an LDDN can be trained with this toolbox. Each layer of LDDN is composed of set of weight matrices, tapped delay lines, basis vector, net input function, and transfer functions. The weight can be connected from inputs or other layers of the network. The net input function combines the output and the bias to produce the net input. The transfer function is also known as activation function.

Note that a change in the weight has a direct effect on the output and it causes an immediate change. Also, since some of the inputs to the layer are functions of the weights the prediction is indirectly affected by any change in the weights. Thus, we need to use dynamic back propagation to avoid this issue. However, the error for dynamic networks can be more complex compared to static networks and training is more likely to be trapped in local minima. Therefore, it is suggested to train the network several times to achieve an optimal result.

The training of a NARX model can be done in two steps. The first step creates a series-open loop architecture, in which the actual output is used instead of using feedback of the estimated output. This method results in a one-step-ahead prediction. However, the advantage of using this approach is that the input to feedforward network is more accurate and static backpropagation can be used which has less complexity and of course is less time consuming compared to dynamic backpropagation method. When the NARX model is trained it can be used in a closed loop parallel configuration which is useful for multi-step-ahead prediction. Figure 2.10 shows the series parallel architecture (top) and the closed loop parallel architecture (bottom).

Note that the result of NARX model can be different each time a neural network is trained. This is because during the training process the initial weights and bias are randomly selected and different divisions of the data are used for training which ultimately results in a different solution. Thus, different neural networks trained on the same problem can generate different outputs for the same input. Therefore, it is recommended to retrain the model to ensure that a neural network accuracy is acceptable.

Example 1: A Magnetic Levitated System

Here we provide an example to illustrate how a NARX neural network predicts the dynamic behavior of a magnetic levitated system. The example illustrates how a neural network combined with NARX can predict the time-dependent behavior of a magnetic levitated system using a control current [64].

The control current is an external input time series while the past magnet position feedback time series is used as the output to the NARX to predict the future value of the magnet position. Due to the nonlinear behavior of this system the NARX neural network model is a good candidate to use. This example is conducted in the MATLAB environment.

To start, we load the data and separate them into two arrays of input time series, X, and output time series, T. Then the NARX neural network is created by defining the following model parameters listed in Table 2.3. The NARX model has one hidden layer and one output layer with twenty neurons. Also, five tap delays of the control current (input signal) and the magnet position (output signal) are sufficient to predict the system's dynamic behavior. More hidden layers, neurons or tap delays can be used for more complex dynamic systems.

Before training the network, the five tapped delay states should be specified. Therefore, the first two signals of X (input) and T (output or feedback) are used as two delay signals. Note that the array of T is used as input and target series during training the model. The MATLAB command *preparets* is used to prepare the model for training by separating the input and output delay states from the remaining signals, automatically. The model is ready for training after the preparation step is completed.

During the training phase, the timesteps are divided into training, validation, and test sets. The purpose of using training set is to train and teach the behavior of this control system. During the training process, the code monitors the prediction performance of the validation set. The training process continues as the prediction of the validation set is improved. The training process stops if no improvement is observed. Figure 2.11 shows the training process. It shows the neural network structure including the input, output, number of layers, number of lags, and number of neurons. Also, it provides information about the algorithm being used to train the model and how the accuracy of the model is evaluated. Some useful plots become available after the model is trained which can provide visual information on the performance and prediction accuracy of the model.

Figure 2.12 illustrates the performance improvement during training. The performance is evaluated for each of training, validation and test sets based on the mean square error (logarithmic scale in this case). As is shown the model stopped at epoch 264 which is six epochs after the best validation performance (epoch 258). The provided default number of six validation checks, which is the convergence criterion for this example, can be modified for better accuracy. Figure 2.13 shows a comparison between

the network's response and the actual position of the magnet. It can be seen that the model outputs are almost identical to the target data. The maximum MSE is about 0.09.

So far, we have shown and discussed the prediction of the open loop NARX model. Now, we convert the model to a closed loop form. Figure 2.14 shows a schematic of the closed loop NARX. Here, the network uses only specified initial actual magnet positions, and then utilizes its own prediction as feedback to the loop. As is shown in Figure 2.15, the magnet position at each time step is not predicted accurately compared to open loop form. This result is expected since we are using the inputs which include the error associated with the model predictions. However, the model is able to predict the behavior of the system for most of the time steps with an acceptable error.

Example 2: A Nonlinear Duffing Oscillator

A single degree of freedom Duffing oscillator which is a vibratory system with a linear and nonlinear stiffness [65] is used to show the capability of NARX neural network to predict the response displacement and velocity (outputs) of the system excited by a Gaussian force (input) in the time domain. Figure 2.16 provides a schematic of the Duffing oscillator.

The equation of motion is

$$m\,\ddot{x}(t) + c\,\dot{x}(t) + k_1 x(t) + k_2 x^3(t) = F(t) \tag{2.80}$$

where, $x(t)$ is the output displacement and $F(t)$ is the input force which is a random process. The mass m is 25 Kg, the linear damping c is 3 Kg/s, and the spring coefficients k_1 and k_2 are 10^2 and 10^4 N/m, respectively. The $k_2 x^3(t)$ is the nonlinear stiffness term. The undamped frequency of the linear system is $\omega_n = \sqrt{k_1/m} = 2$ rad/s, and the damping factor is $\varsigma = 2m\omega_n = 0.15$.

The input $F(t)$ is a stationary and Gaussian random process. Sample realizations of the $F(t)$ process are generated using the Karhunen-Loeve Expansion (KLE) method based on a provided Power Spectral Density (PSD). The outputs (displacement and velocity) are obtained first by integrating the equation of motion (Eq. 2.68) over time using the 4th order Runge Kutta method. Using KLE we generated 80 input trajectories of the stationary, Gaussian process $F(t)$ for a period of 90s. The equation of motion was then integrated numerically to obtain the corresponding trajectories of the output. Figure 2.17 shows eighty trajectories of $F(t)$.

In this example, we first use 74 trajectories to train a NARX neural network and then we predict the response for the other 6 input trajectories. To construct the NARX neural network we used 5 lags, 2 hidden layers and 6 neurons for each hidden layer. Each trajectory is discretized using an increment of 0.1 seconds. The training process stops when the number of epochs reaches 100 to make sure the *MSE* is very low.

Figures 2.18 to 2.23 compare the predicted and actual displacement and velocity of the Duffing oscillator system for the untried trajectories. The comparison shows that the predicted displacements and velocities for the 75th to 80th force trajectories are very accurate.

2.3 Summary

This chapter provided first a brief introduction to SI combustion engines. In addition, it provided a review of different metamodeling techniques including conventional (time-independent) and time-dependent metamodels. The popular Kriging stochastic metamodeling technique was discussed in detail. Finally, neural networks and the NARX network approach were reviewed. The former can be used to predict the

output of time-independent problems while the latter is used to predict the output of time-dependent problems. A magnetic levitation example was also presented to show how a NARX neural network can be constructed and trained to predict the time-dependent magnet position corresponding to a given control signal. In addition, a Duffing oscillator example was presented reinforcing the very good predictive capabilities of a NARX neural network.

Table 2.1: Common basis functions

Basis function names	Basis functions
Linear	$\|x - x_j\|$
Cubic	$\|x - x_j\|^3$
Thin-plate spline	$\|x - x_j\|^2 * \log\|x - x_j\|$
Gaussian	$\exp(-\theta_j\|x - x_j\|^2)$

Table 2.2: Activation functions

Activation functions	Name of the functions
$f(x) = x$	-Linear
$f(x) = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases}$	-Binary step
$f(x) = \dfrac{1}{1 + \exp(-x)}$	-Logistic (soft step)
$f(x) = \tanh(x)$	-TanH
$f(x) = \begin{cases} 0 & x \leq 0 \\ x & x > 0 \end{cases}$	-Rectified linear unit (ReLU).

Table 2.3: Characteristics of NARX neural network

Number of hidden layers	Number of neurons	Tap delays
1	20	5

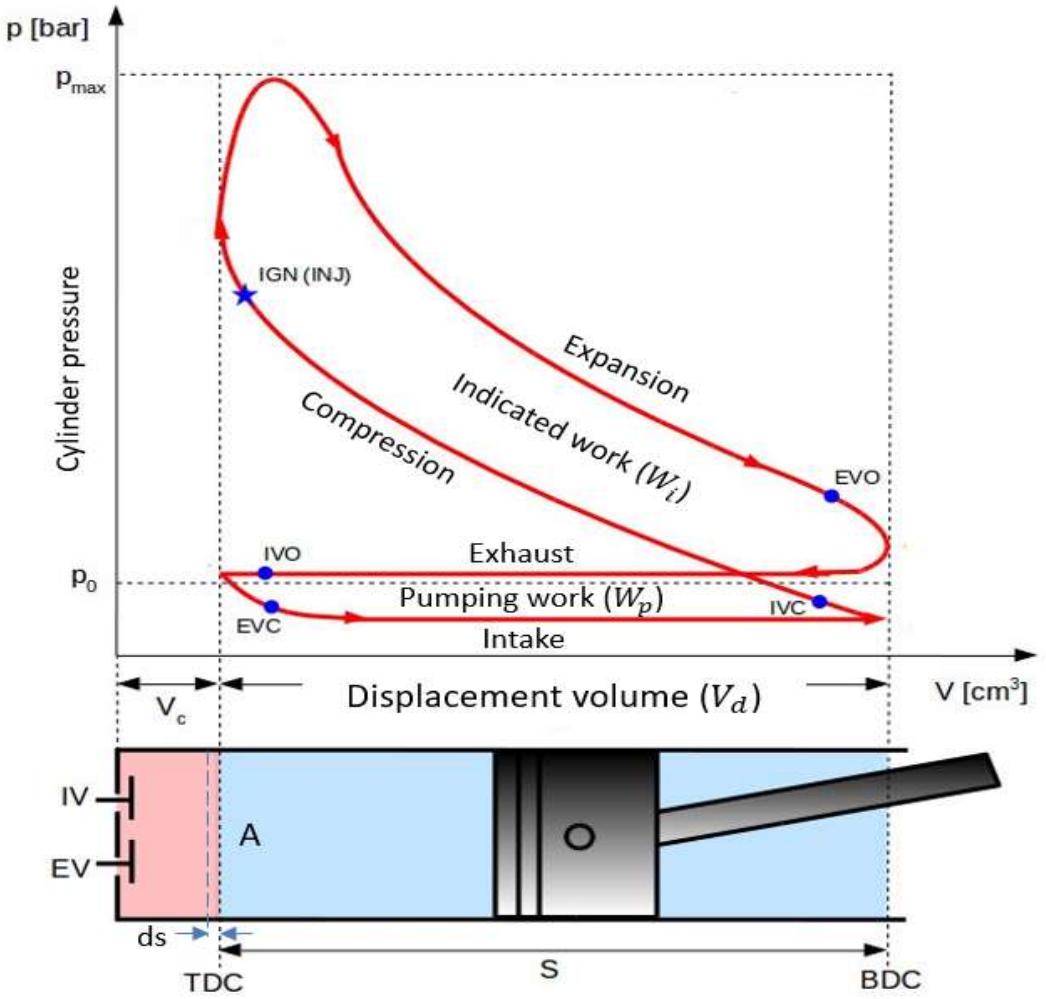

Figure 2.1: Schematics of p-V diagram and engine cylinder

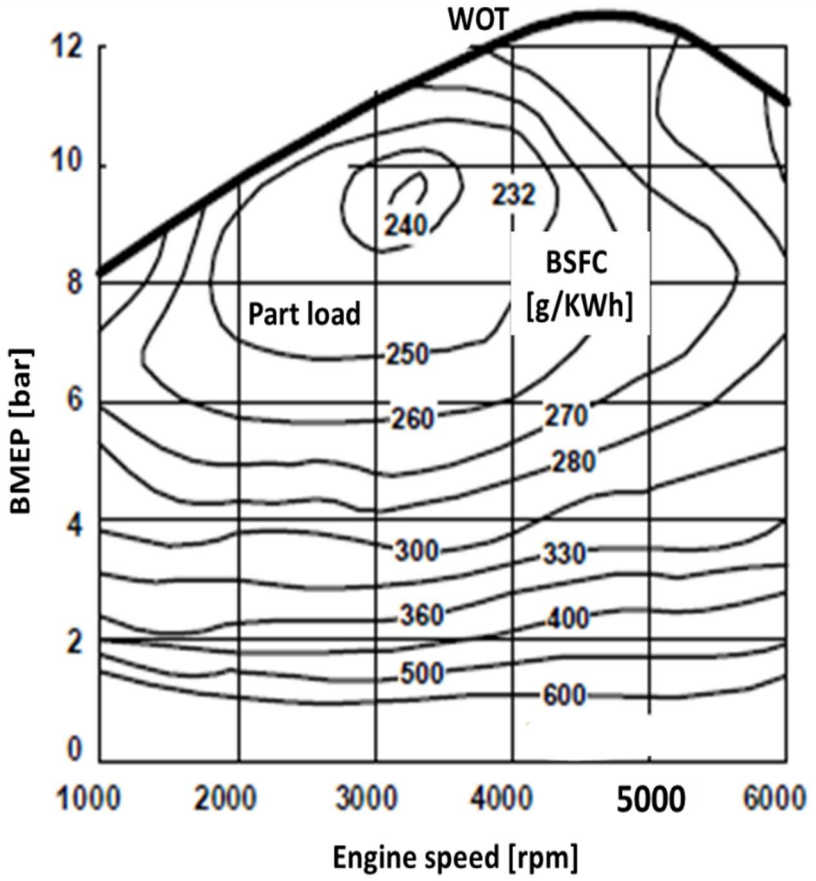

Figure 2.2: Example of a fuel map for a natural aspirated gasoline engine [1]

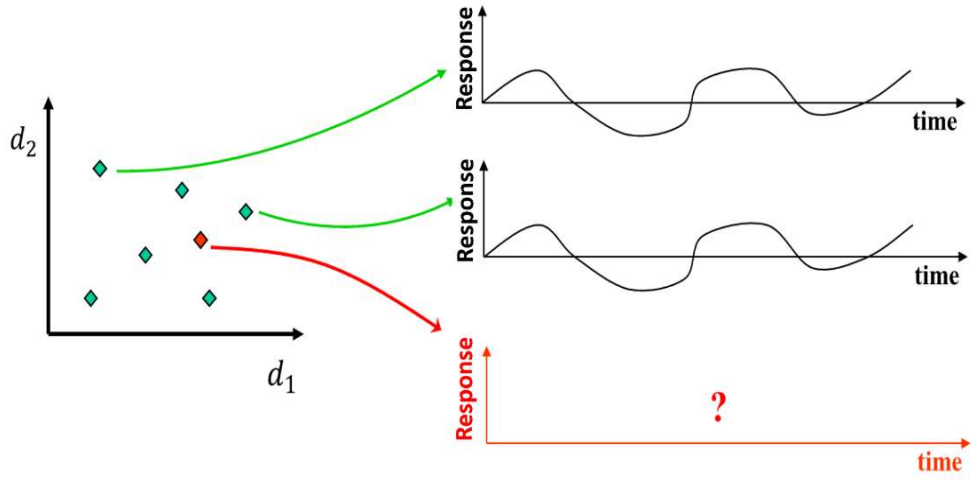

Figure 2.3: Illustration of parametric time-dependent metamodeling

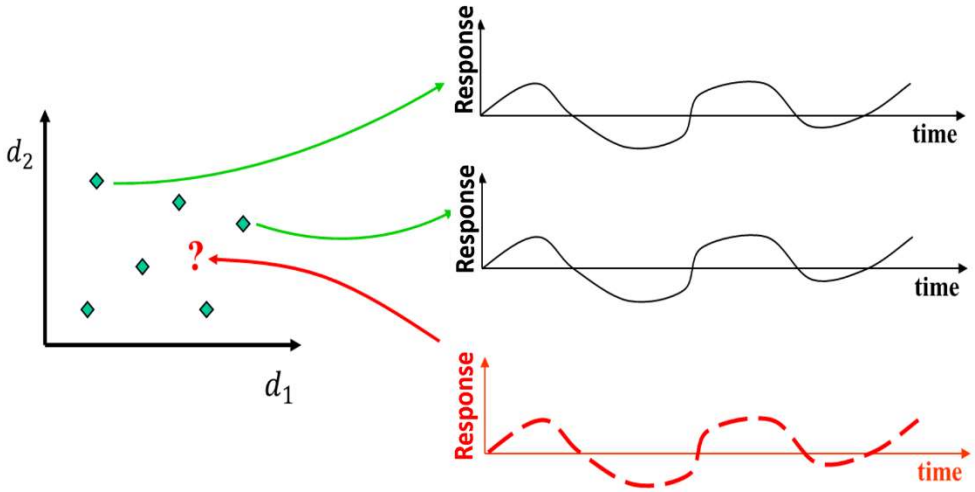

Figure 2.4: Illustration of inverse parametric time-dependent metamodeling

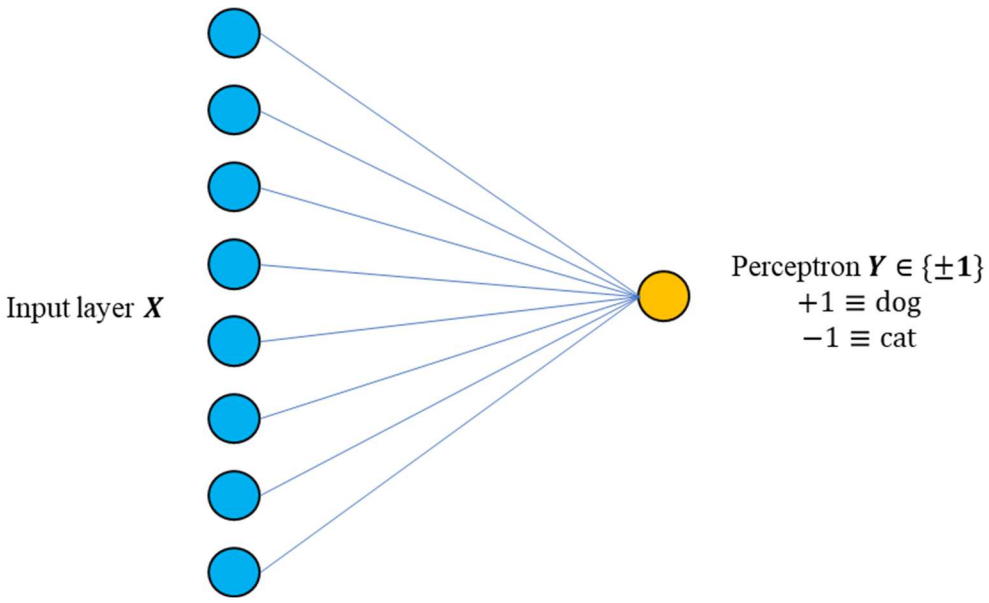

Figure 2.5: Single layer network for binary classification between dogs and cats

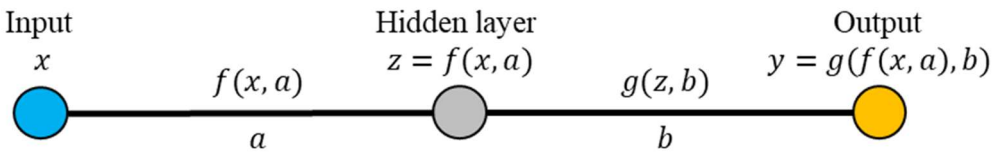

Figure 2.6: One-node, one-hidden layer network [53]

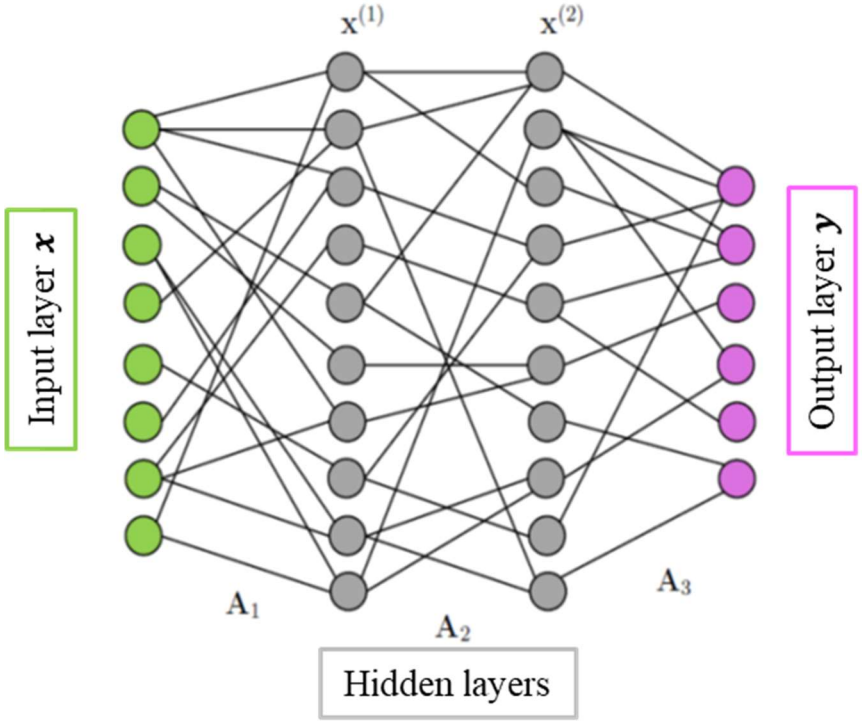

Figure 2.7: Generalized NNs structure with multiple layers and nodes [53]

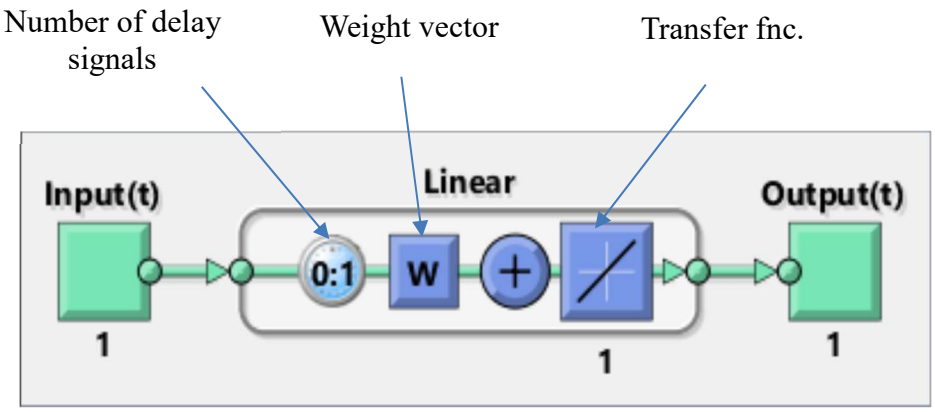

Figure 2.8: Example of a feedforward liner transfer function network with a tapped delay line on input

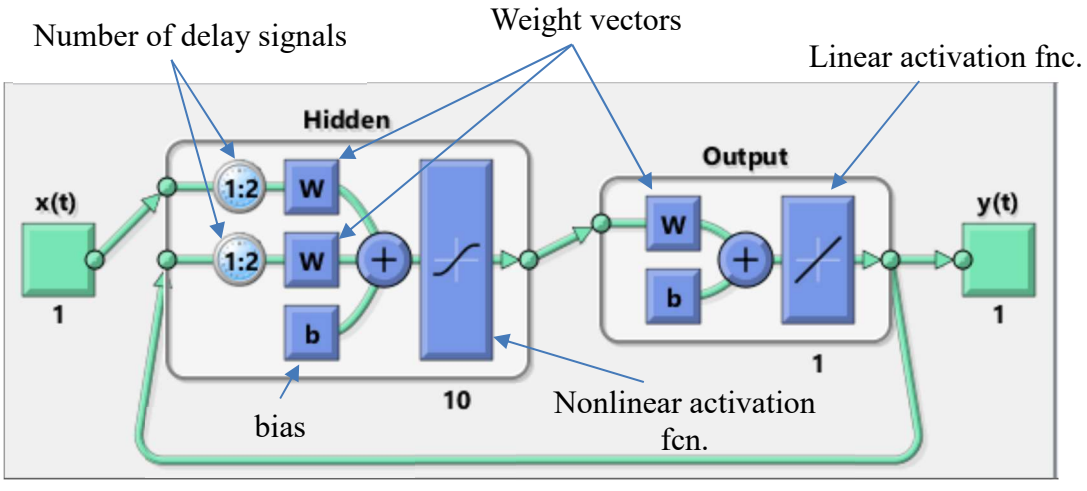

Figure 2.9: Example of a NARX model structure

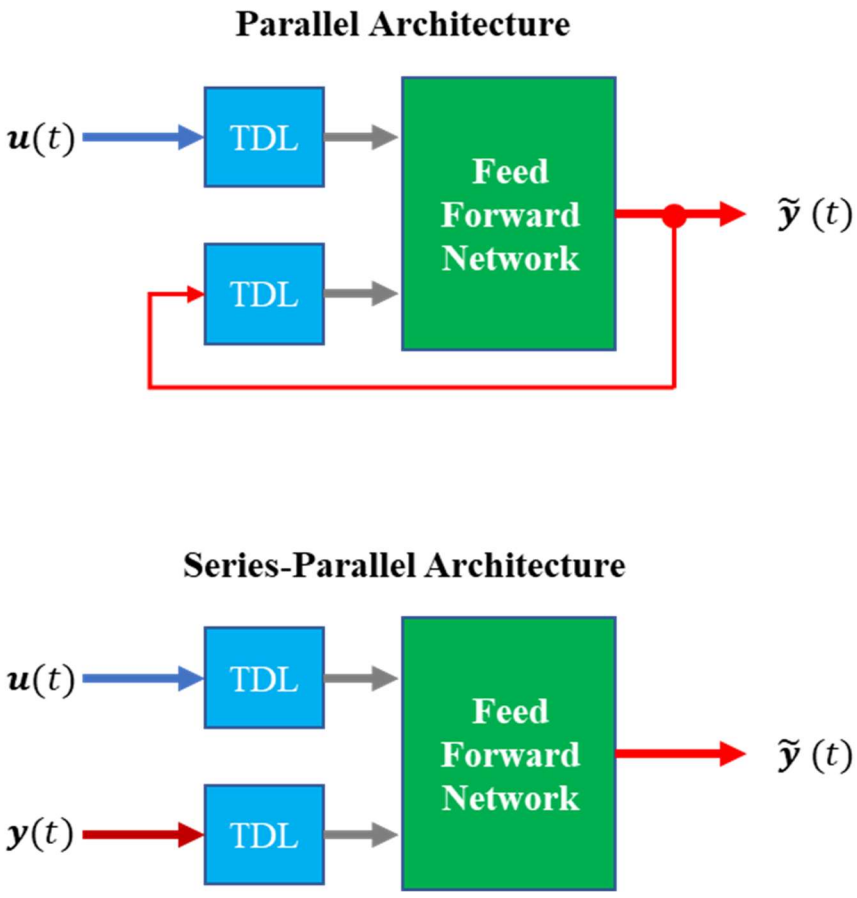

Figure 2.10: Schematics of parallel architecture (top) and series-parallel architecture (bottom)

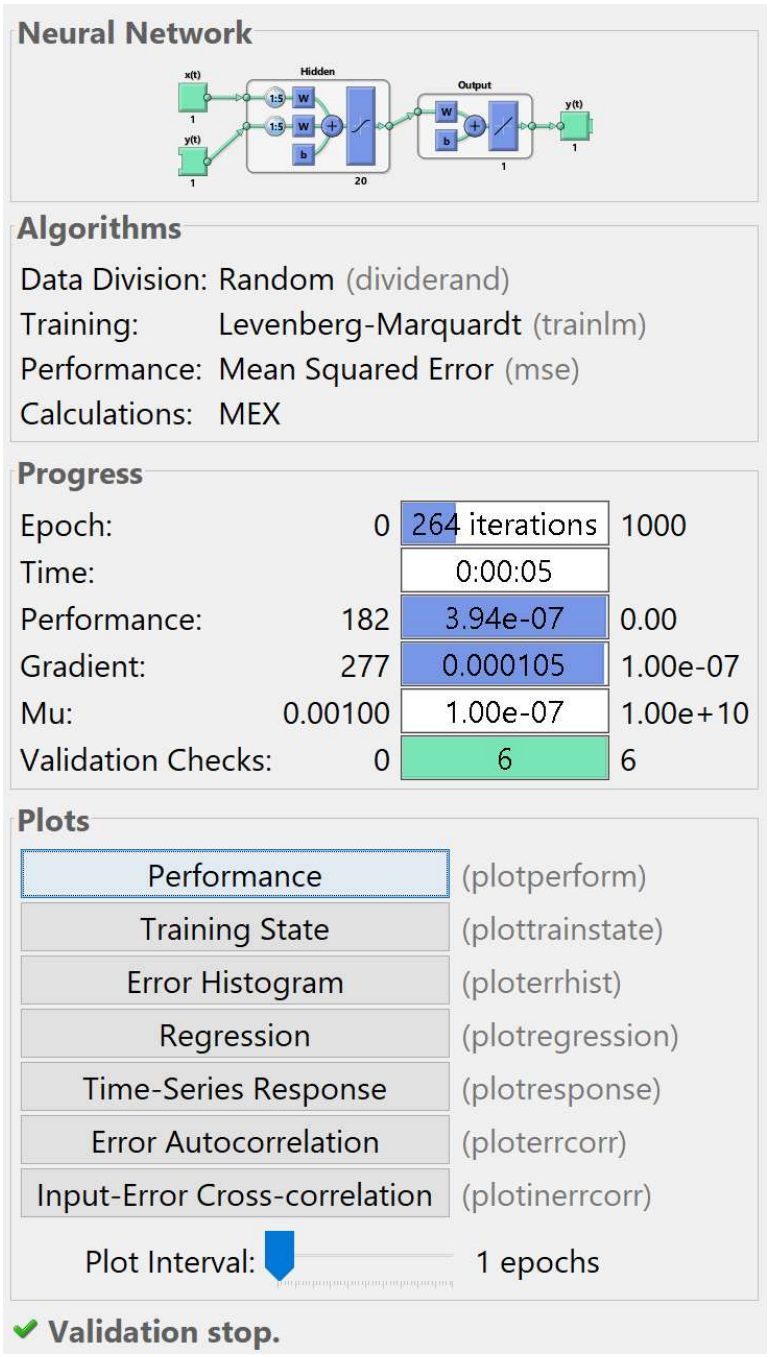

Figure 2.11: NARX neural network structure and performance of Example 1

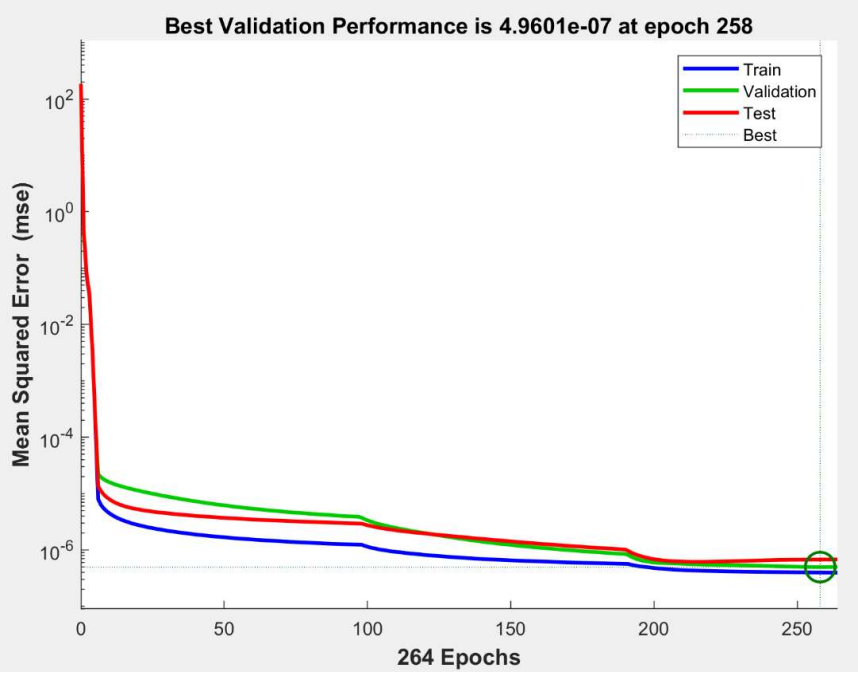

Figure 2.12: Prediction performance in Example 1

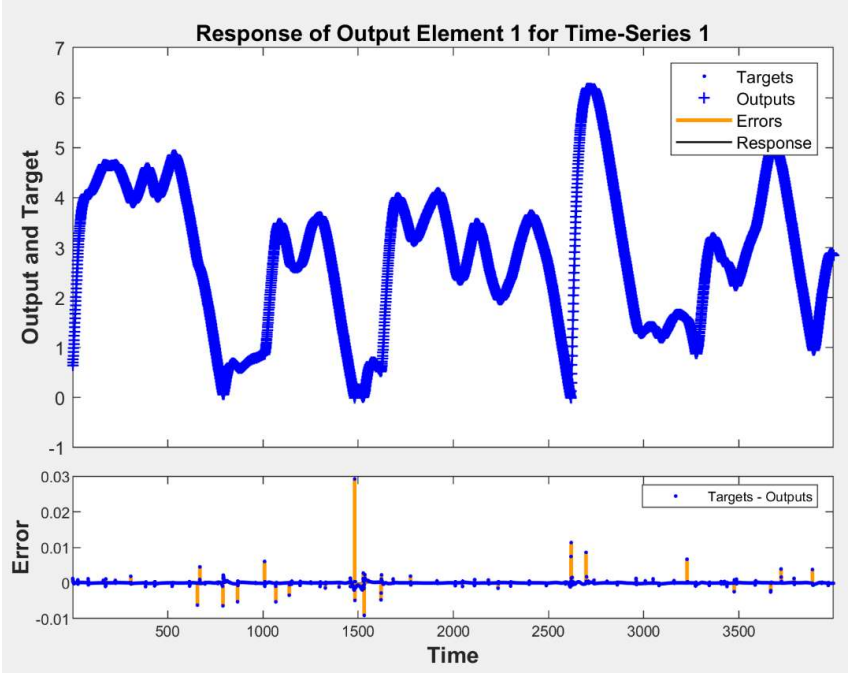

Figure 2.13: Comparison between the target and prediction data

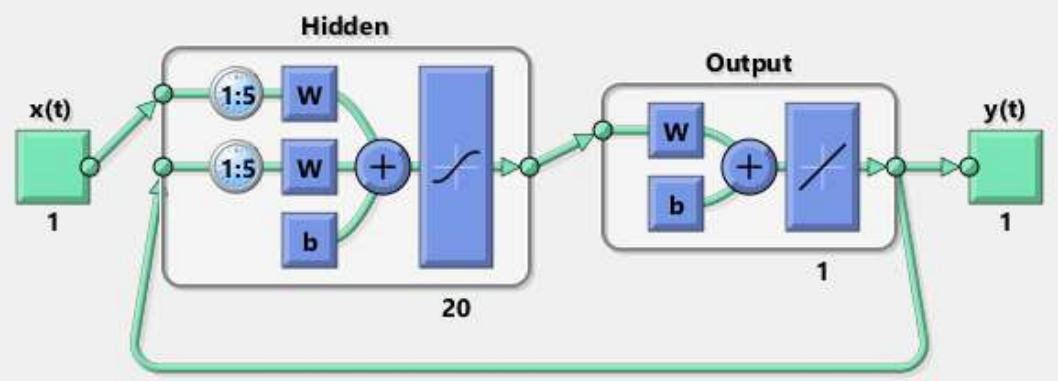

Figure 2.14: Schematic of closed loop form of NARX model in Example 1

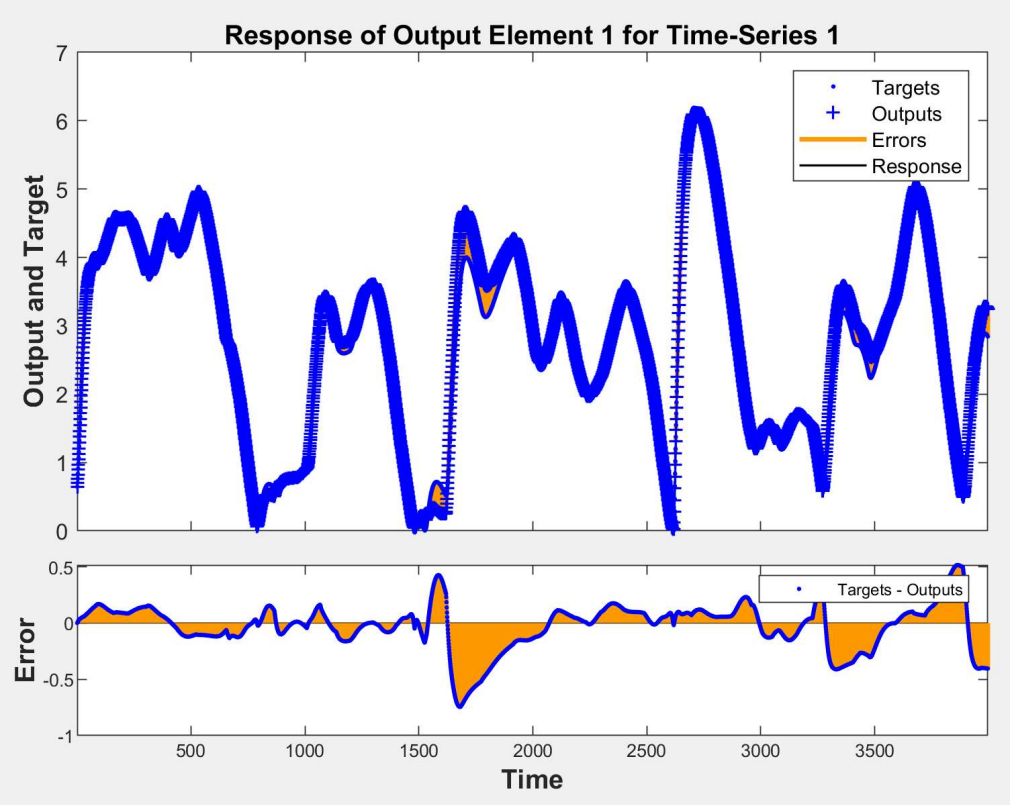

Figure 2.15: Comparison between target and predicted data using closed loop form of NARX model

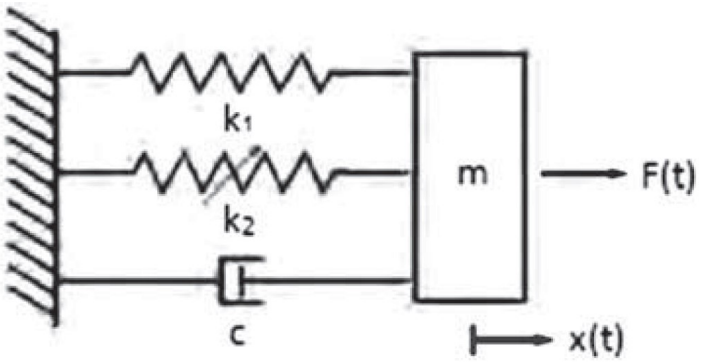

Figure 2.16: Schematic of Duffing oscillator

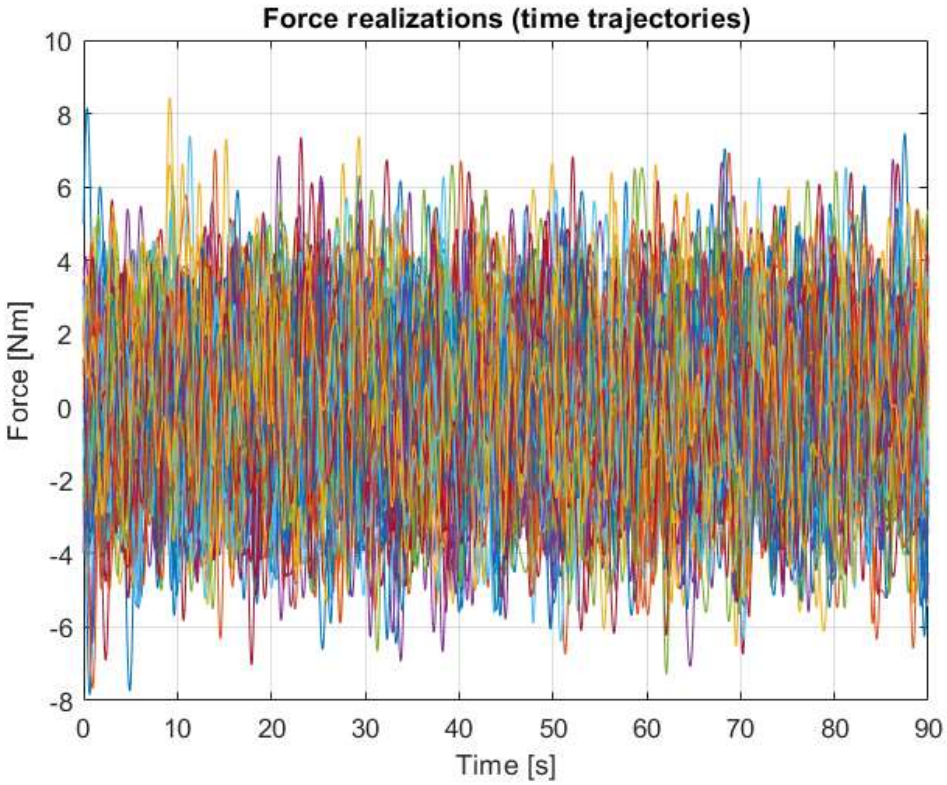

Figure 2.17: Eighty force (input) trajectories from 0 to 90s

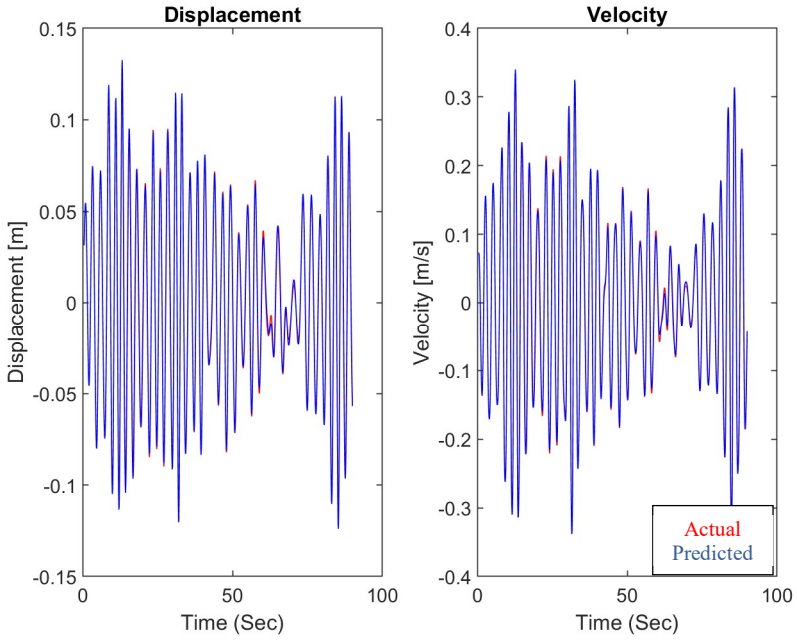

Figure 2.18: Comparison between actual and predicted displacement and velocity of the Duffing oscillator - trajectory 75

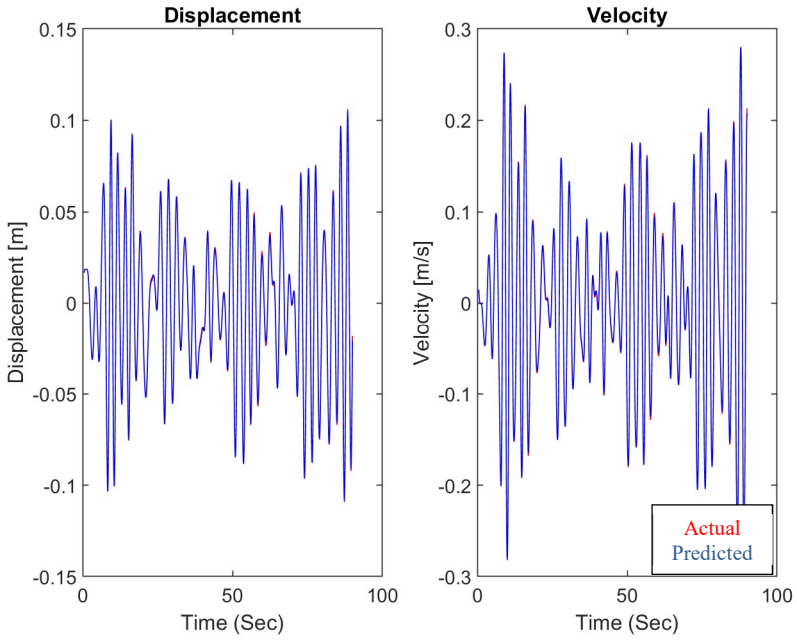

Figure 2.19: Comparison between actual and predicted displacement and velocity of the Duffing oscillator - trajectory 76

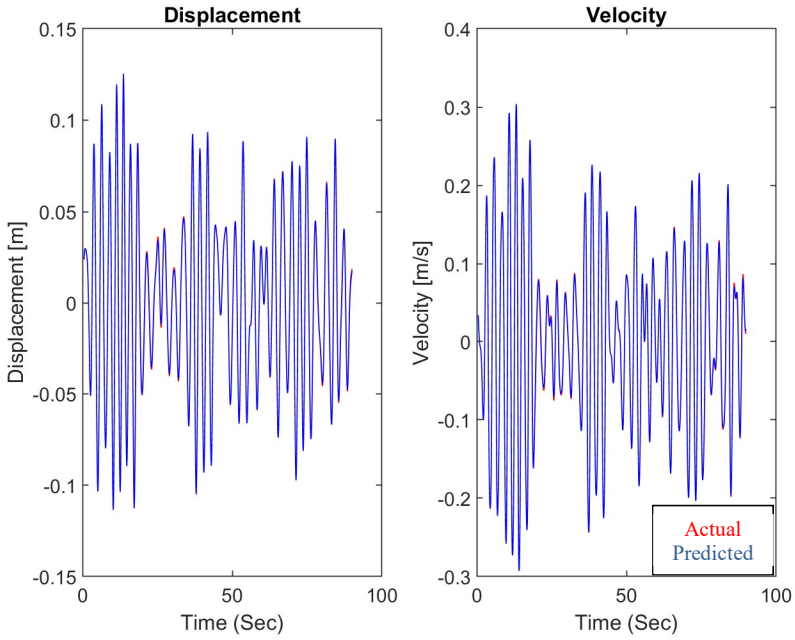

Figure 2.20: Comparison between actual and predicted displacement and velocity of the Duffing oscillator - trajectory 77

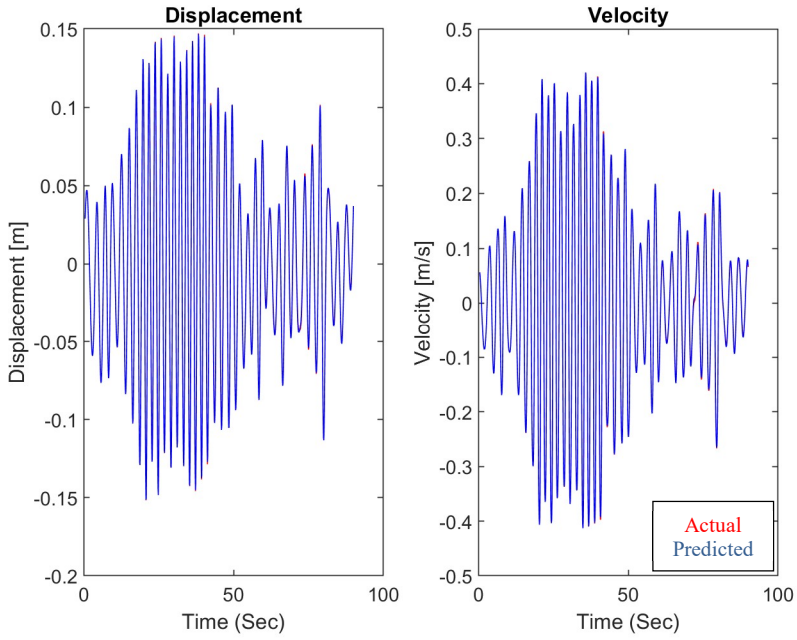

Figure 2.21: Comparison between actual and predicted displacement and velocity of the Duffing oscillator - trajectory 78

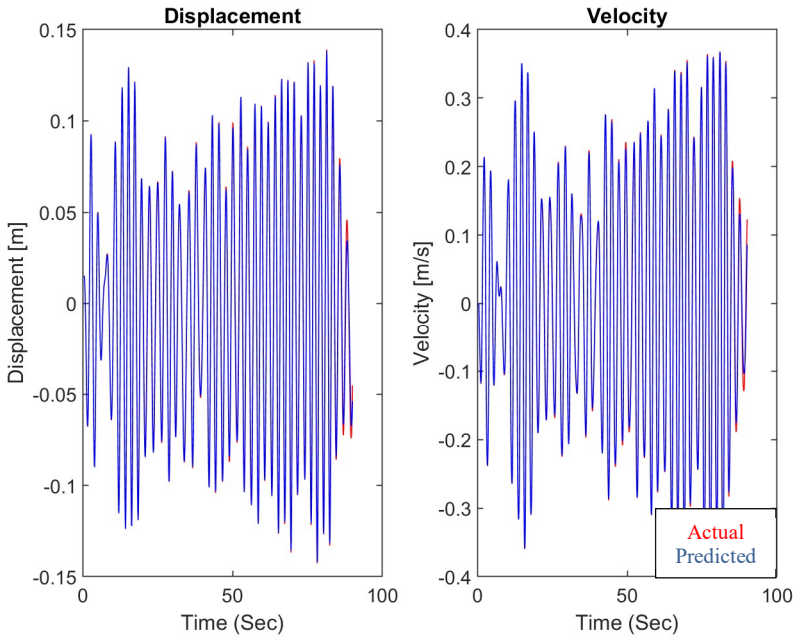

Figure 2.22: Comparison between actual and predicted displacement and velocity of the Duffing oscillator - trajectory 79

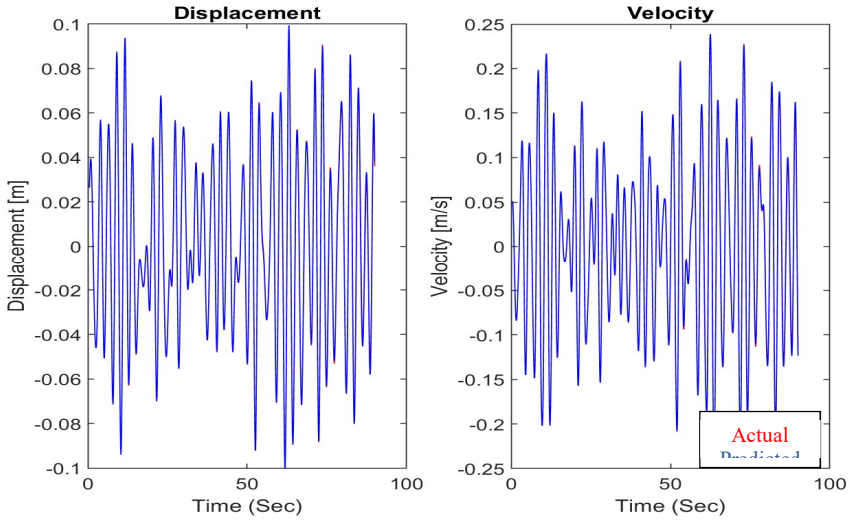

Figure 2.23: Comparison between actual and predicted displacement and velocity of the Duffing oscillator - trajectory 80

CHAPTER THREE

METHODOLOGY TO DEVELOP INTERNAL COMBUSTION ENGINE FUEL MAPS

Fuel maps provide information about the engine characteristics which depend on the engine speed and load. The load is usually expressed in terms of the Brake Mean Effective Pressure (BMEP). An engine fuel map has two regions known as full load and part load. If an engine operates at full load, the throttle is wide open (WOT), providing the highest BMEP throughout the speed range. In part load (partially open throttle valve), it is desired to operate with the minimum fuel consumption while a targeted load is met at any engine speed. A number of independent variables must be controlled in order to achieve maximum BMEP at full load and minimum fuel rate at part load.

In Spark Ignition (SI) engines, these independent variables can be the spark timing, the cam timing, the throttle angle, and the Exhaust Gas Recirculation [66-68]. In turbocharged engines the wastegate is also another control factor. Advances in technology have led to improvements in engine performance and fuel economy, but they increase the complexity to generate the fuel map. For instance, the intake variable cam timing can improve the full load performance by varying the intake cam phasing. In contrast, a more sophisticated system using Variable Valve Actuation (VVA) increases torque generation, improves fuel economy, and reduces pollution by adding three independent variables (valve timing, lift, and duration of intake and exhaust valves) in generating a fuel map [69, 70]. Increasing the number of independent variables results in a time consuming and expensive process to obtain the fuel map.

An alternative approach to build a fuel map is to use a physics-based simulation tool which is highly correlated with engine data. Taking advantage of simulation is cost efficient and is less time consuming if the goal is to optimize input variables only for a couple of operating conditions. Fu et al. [71] used simulation to investigate the effect of cam strategies on fuel economy only at certain part load conditions. However, as the number of operating conditions increase, the number of computer experiments also increases, and more time is required for simulations to converge. To tackle this issue, Wu et al. [72] used a high-fidelity simulation tool combined with an Artificial Neural Network (ANN) to maximize the torque at full load for a variable Valve Timing (VVT) engine with dual-independent cam phasers by optimizing the intake and exhaust cam phasing, the spark timing, and the fuel-air equivalence ratio. In their study, 95% of full factorial DOE (Design of Experiments) points were used to train the ANN at full load.

In this research, a computationally efficient process is introduced to create a fuel map at both full load and part load conditions using a physics-based simulation tool to train a number of metamodels. Design of Experiments [73-77] combined with metamodeling techniques [78, 79] offer a promising venue to achieve optimized designs [80, 81] rather efficiently and accurately. DOEs place points (design sites) strategically in the design space in order to "space fill" the entire space. A metamodel is then built using the responses at the DOE points (training points). The metamodel predicts the response at any point in the design space. The objective of this study is to build an engine fuel map using metamodels to accurately predict the minimum fuel rate and the maximum BMEP at part load and full load conditions, respectively.

3.1 Overview of Fuel Map Generation

We use a four-cylinder 2.0L naturally aspirated gasoline engine with a port fuel injection system. Each cylinder has two intake and two exhaust valves. The engine is equipped with a VVT system in which the intake and exhaust camshaft positions are independently controlled. Figure 3.1 shows the Intake Centerline Location (ICL) and Exhaust Centerline Location (ECL) to represent the intake and exhaust camshaft positions, respectively. The ICL is defined as the distance between the Top Dead Center (TDC) and the centerline of the intake camshaft lobe. The ECL is defined as the distance between TDC and the centerline of the exhaust camshaft lobe. They are both measured in degrees of crank angle (CA). The cam-phasing range varies from 80 to 140 CA degrees from before-TDC and after-TDC gas exchange for both the ECL and ICL. Table 3.1 lists the main specifications of the engine.

There are various control parameters (design factors), such as ICL, ECL, and EGR valve, that can be controlled to obtain the minimum fuel consumption at part load, and the maximum BMEP at full load. Here, we use the ICL and ECL as independent control factors at any combination of engine speed and BMEP.

In addition to minimizing fuel consumption at part load and maximizing BMEP at full load, there are other parameters such as the Residual Gas Fraction (RGF) and the CA50 (50% of the fuel mass is burned) which influence the combustion quality and must be considered when generating a fuel map. A high RGF in the cylinder slows down the combustion process, increasing the potential for a poor combustion. The RGF increases by increasing the valve overlap (retarding ECL and advancing ICL) due to the pull back of exhaust gas from the exhaust port by the downward motion of the piston. In a test cell,

the engine stability is measured by the Coefficient of Variation (CoV) of IMEP, which is worsened as the RGF increases by retarding ECL. Because the CoV of IMEP cannot be provided by simulation, the limit of 25% of RGF is considered as the maximum RGF beyond which the combustion is considered unstable.

The CA50 can be controlled by spark timing. At different engine speeds, there is a spark timing (and thus CA50) which results in the Maximum Brake Torque (MBT) called the "MBT timing." Studies have shown that the MBT occurs when the CA50 is between 4 and 12 degrees CA after TDC firing, depending on the operating conditions. In this paper, the spark timing is controlled such that the CA50 is at 8 CA after TDC firing. However, at higher loads and lower engine speeds there is a risk of engine knock. To avoid it, the spark timing should be retarded which consequently retards the CA50. In this case, the engine cannot run at MBT. Retarding CA50 deteriorates the combustion quality. Here, the maximum limit of CA50 is 25 degrees CA after TDC firing to avoid poor combustion.

In part load, the fuel rate is minimized for a specified BMEP and RPM by varying the ECL and ICL. Details are provided in Section 3.2. However, for given BMEP, RPM, ECL and ICL, the air-fuel mixture may not be sufficient to produce enough load to meet the requested BMEP. As a result, the engine operation is not acceptable. For this reason, the output BMEP from a metamodel (i.e. $\widetilde{\text{BMEP}}$) should be within ±5 kPa of the requested BMEP. This is imposed using a constraint as illustrated in the optimization problem of Equation (3.2) in Section 3.2.

Figure 3.2 shows an example of the fuel rate map at part load. The map is for the engine of Table 3.1. The throttle angle is controlled to maintain the manifold air pressure

and air flow for given engine speed and BMEP. The injected fuel is controlled to maintain the stoichiometric fuel-air ratio throughout the part load operating condition. A knock controller is also used to adjust the spark timing to avoid knock. At any possible combination of engine speed and BMEP, the ICL and the ECL are varied with a 10 degree CA increment within the allowed phasing range in order to estimate the fuel mass flow rate and other engine parameters such as RGF and CA50. We observe that the fuel rate increases as the engine speed and the BMEP increase. We also observe that the engine cannot achieve a BMEP higher than 800 kPa at 1000 rpm due to hardware limitations and the engine characteristics (top-left corner of Figure 3.2).

Figure 3.3 shows the fuel rate as a function of ICL and ECL at 2000 rpm and 200 kPa of BMEP. At this condition, without considering the RGF and CA50 values, the minimum fuel rate is achieved when the intake and exhaust valves have the maximum overlap. However, according to Figure 3.4 the RGF level is highest when the valve overlap is maximum. Hence, the possibility of poor combustion is high. Figure 3.5 shows that CA50 is below the limit of 25 degrees CA. Therefore, it is unacceptable to operate the engine in the region where the ECL is between 80 and 110 degrees CA before TDC and the ICL is between 80 to 110 degrees CA after TDC, with the RGF being greater than 25%. Hence, the most retarded ICL (140 degrees CA) and the most advanced ECL (80 degrees CA) result in the minimum fuel rate at 2000 rpm and 200 kPa BMEP.

In summary, to obtain the fuel map, we first need to build four metamodels – one for each of BMEP, RGF, CA50 and $\dot{m}_f$. The first three are used for full load and the last is used for part load. The inputs to the metamodels are the engine speed, BMEP, ICL and ECL. Note that BMEP appears in the input and output lists. The input value is assumed

and the output value is the actual realizable value based on combustion and engine characteristics. This differentiation is important in part load conditions. Using the metamodels, the actual fuel map is determined by solving a different constrained optimization problem for full and part load conditions using the ECL and ICL as design variables (see Sections 3.1 and 3.2 for details). Table 3.2 lists the inputs, outputs, objectives and constraints for part load and full load conditions.

To determine the accuracy of the developed fuel map, a full factorial DOE of the engine speed, BMEP, ICL and ECL design space is used to evaluate the realizable BMEP at full load and the minimum $\dot{m}_f$ at part load under constraints ensuring good combustion characteristics. For each design site of the DOE, the GT Power software is run to obtain the optimal BMEP for full load and $\dot{m}_f$, for part load. These optimal values are used to establish the accuracy of the estimated fuel map using the developed four metamodels. Considering that running GT Power at a design site can be computationally demanding, the metamodels are built using the minimum possible number of GT Power runs (i.e., minimum number of design sites where GT Power evaluates the optimal BMEP and $\dot{m}_f$, outputs). To build the full factorial DOE, Table 3.3 shows the upper and lower limit and the number of levels for each input variable. The range and number of levels for each input variable were determined based on experience to generate the actual fuel map with acceptable accuracy. Note that at full load, in order to achieve maximum BMEP, we must force the throttle controller to keep the throttle wide open to allow for maximum airflow. Therefore, we target a maximum BMEP of 1500 kPa. This BMEP value is not achievable due to hardware limitations, but it forces the throttle to stay wide open.

Based on Table 3.3, the total number of full factorial DOE points are 3,234 - 10 BMEP x 6 engine speeds x 7 ICL x 7 ECL = 2940 design sites for part load, and 6 engine speeds x 7 ICL x 7 ECL = 294 design sites for full load. As we have mentioned, running GT Power many times can be computationally expensive. Therefore, it is important to have a methodology to obtain a fuel map with less computational effort while maintaining accuracy. In the following sections, we propose a method which helps to generate a fuel map with a relatively small number of computer experiments, while maintaining accuracy.

3.2 Proposed Methodology to Estimate Fuel Map

The proposed methodology to estimate the fuel map of a VVT engine uses Kriging metamodels and consists of two major steps. First, we use a sequential approach to build four metamodels of acceptable accuracy using a relatively small number of engine combustion simulations. The latter are performed using the GT Power software which is the industry standard. The number of GT Power runs is kept to a minimum in order to improve efficiency since each GT Power run (or alternatively an actual engine experiment) is computationally intensive hindering therefore, the practicality of the proposed approach.

In the second step, we estimate the engine fuel map at full load and part load conditions using the developed metamodels, by solving two constrained optimization problems. For full load, the WOT curve is determined by maximizing the BMEP under constraints of RGF and CA50. The constraints ensure acceptable combustion quality. The ICL and ECL are used as design variables. For part load, the mass fuel rate is minimized under the same RGF and CA50 constraints at each BMEP-engine speed condition which

are used as design parameters. An additional constraint is used to make sure the achievable BMEP is close to the target BMEP (design parameter in optimization). Details for both full load and part load are provided in Section 3.3.

The first step in creating a metamodel is to determine the initial DOE points which must be preferably space filling. The size of the design domain is established using the lower and upper bounds of each design factor. In this paper, the design factors are the engine speed, BMEP, ICL, and ECL.

Figure 3.6 shows a schematic of a fuel map and the selected initial points (red dots) for part load. At every corner of the BMEP-engine speed projection of the map, we consider the four corner points (low and high values) of the ICL and ECL. As a result, we have 16 initial design points for part load where GT Power will run. In contrast, for full load we start the metamodeling process with 8 designs (combinations of upper and lower bounds of engine speed, ICL, and ECL). Section 3.3.1 provides more details.

GT Power is a high-fidelity physics-based combustion simulation tool. The responses from GT Power at each of the 16 DOE points are used to obtain three Kriging metamodels - one for the fuel rate, one for CA50, and one for the RGF output. For each metamodel, Kriging provides a mean prediction (most likely value), and a Mean Square Error (MSE) or variance of prediction. The prediction accuracy of each metamodel is improved sequentially by adding more training points to the training point set of the previous iteration. The added training points (one or more) have the same highest MSE. Every time one or more training points, are added to the previous set of training points, a new metamodel is generated. We consider this a new iteration in the sequential process of

metamodel generation until convergence is achieved. For convergence, the maximum MSE becomes less than a threshold.

After converged Kriging metamodels are obtained for fuel rate, CA50 and RGF, the acceptable operating regions of the ICL and ECL domain are determined so that the constraints mentioned in Table 3.2 are satisfied. To generate the final fuel map for part load, the minimum fuel rate is obtained by varying the ICL and ECL for each engine speed and BMEP. Similarly, for full load, the maximum torque (or equivalently BMEP) is obtained in terms of ICL and ECL for different engine speeds.

3.3 Case Study: Generation of Fuel Map at Full Load

In this section, we use a case study to demonstrate how the proposed methodology is used to estimate the engine fuel map for full load and part load conditions. Using the case study, we also provide additional details on the process of building an engine map. The engine of Table 3.1 is used as an example.

3.3.1 Generation of Metamodels and Fuel Map at Full Load

In building the metamodels, 8 designs (combinations of upper and lower bounds of engine speed, ICL, and ECL) are initially selected (first iteration), and GT Power is run to calculate the BMEP, RGF and CA50 at each design point. The GT Power values are used to develop Kriging metamodels for $\widetilde{BMEP}$, $\widetilde{RGF}$, and $\widetilde{CA50}$, and their corresponding $\widetilde{MSE}$s at selected prediction points (Table 3.4). The grid of $\widetilde{MSE}$ prediction points is finer than the grid used to obtain the actual fuel map (Table 3.3) in order to estimate the maximum $\widetilde{BMEP}$ for each engine speed more accurately. In the next iteration, new metamodels are created using the design sites from the previous iteration plus a new DOE set (one or more points). The latter includes the design site (or sites)

from the previous iteration, that have the maximum MSE of $\widetilde{BMEP}$. The process continues until the three metamodels converge (i.e. the maximum MSE is below a predetermined tolerance).

Figure 3.7 shows the maximum predicted MSE of $\widetilde{BMEP}$ at each iteration. The values of the maximum predicted MSE of *BMEP* was very high after the first few iterations. However, as the number of iterations increases the MSE reduces indicating that the prediction accuracy increases. In this study, the sequential process of metamodel generation is terminated after 100 iterations with the average maximum MSE of the last 20 iterations being stabilized around 200 kPa^2. Convergence is achieved if the relative error of predicted MSE satisfies the following inequality for each prediction point where the index i refers to the iteration number.

$$\left|\frac{\widetilde{MSE}^{i+1}_{\widetilde{BMEP}} - \widetilde{MSE}^{i}_{\widetilde{BMEP}}}{\widetilde{MSE}^{i}_{\widetilde{BMEP}}}\right| * 100 \leq 1.5 \tag{3.1}$$

We have observed that the prediction accuracy for both full and part load is acceptable by satisfying the convergence criterion of Equation (3.1). The same is true for Equation (3.3) later on. If these convergence criteria hold, the prediction accuracy of the metamodels satisfies the constraints of Table 3.5. The predicted quantities in Table 3.5 ($\widetilde{BMEP}$, $\widetilde{\dot{m}_f}$, $\widetilde{RGF}$, and $\widetilde{CA50}$) are obtained from the optimization problems (3.2) and (3.4).

In order to evaluate the prediction accuracy, the coefficient of determination R^2 and the Root Mean Square Error RMSE are calculated for the output parameters $\left\{\widetilde{BMEP}, \widetilde{RGF}, \widetilde{CA50}, \widetilde{fuel\ rate}\right\}_i, i = \{1, \dots, N\}$ where N is the number of prediction

points of Table 3.3. Figures 3.9 and 3.10 show the R^2 and $RMSE$ as the four metamodels are trained using more points.

The $RMSE$ of Figure 3.8 indicates that as the metamodels are trained using more training points, the average absolute error between the actual fuel map and the estimated fuel map using the metamodels, reduces and the error stabilizes after 60 iterations. Figure 3.9 shows a poor correlation between the actual data and the prediction in the first iterations but after adding more training points the R^2 reaches almost 1 after 60 iterations indicating a high correlation between the actual and estimated fuel maps.

At full load, we seek the ICL and ECL that maximizes the BMEP for each engine speed, subject to constraining the RGF to be less than 25% and the CA50 to be less than 25 degrees CA after TDC firing. The allowed ICL phasing range is between 80 to 140 degrees CA after TDC gas exchange, and the allowed ECL phasing range is between -140 to -80 degrees CA which is before the TDC gas exchange. The optimum ICL and ECL values are thus obtained by solving the following optimization problem.

$$\max_{\underline{\boldsymbol{x}}} \widetilde{BMEP}(\underline{\boldsymbol{x}}, RPM)$$

$$\text{such that:} \quad \begin{aligned} &\widetilde{RGF}(\underline{\boldsymbol{x}}, RPM) < 25\% \\ &\widetilde{CA50}(\underline{\boldsymbol{x}}, RPM) < 25\ degCA \\ &-140 \leq ECL \leq -80 \\ &80 \leq ICL \leq 140 \end{aligned} \tag{3.2}$$

$$\underline{\boldsymbol{x}} = \{ECL, ICL\}$$

A genetic algorithm is used to obtain the optimal solution.

Figure 3.10 illustrates the difference between the actual WOT curve and the predicted WOT curve when the metamodels were trained using 121 design sites after 100 iterations. The figure shows that as the number of training points increases, the prediction of maximum BMEP at each engine speed approaches the actual WOT curve.

Figures 3.11 and 3.12 show a good match between the predicted and actual RGF, and CA50. As is expected, the RGF percentage is low at higher loads and the difference between the prediction and the actual RGF is less than 5%. The trend of predicted CA50 is similar to the actual values for each engine speed. It is retarded at lower engine speeds to avoid knock. The difference between actual and predicted CA50 is less than 5 degrees CA which is considered acceptable.

Figures 3.13 and 3.14 show the ICL and ECL respectively, for the maximum BMEP at full load. For less than 50 iterations, the ICL and ECL do not match the actual data (thick black line) at some engine speeds. However, as the number of iterations increases, the ICL and ECL match the actual data throughout the entire engine speed with the maximum difference being less than 10 degrees CA.

Based on the results in Figures 3.8 through 3.14, we observe that at least 75 iterations are required to obtain the full load WOT curve with an acceptable accuracy using metamodels.

3.3.2 Generation of Metamodels and Fuel Map at Part Load

In part load, similarly to full load, the 16 initial design sites (see Figure 3.6) are defined using combinations of lower and upper values of engine speed, BMEP, ICL, and ECL. The actual data at the 16 initial design sites are used to train four metamodels to predict the fuel rate, BMEP, RGF and CA50 outputs and their MSEs. Design sites with

the maximum prediction MSE are sequentially added until convergence. The objective in part load is to minimize the fuel rate. Figure 3.15 shows the maximum MSE of predicted fuel rate at each iteration.

A zoom in of the blue box region shows that the MSE values decrease at each iteration, *i*, until convergence. The latter is achieved if the maximum relative error of predicted MSE of fuel rate $\dot{m}_f$ among all prediction points, satisfies the following inequality

$$\left| \frac{\widehat{MSE}^{i+1}_{\tilde{m}_f} - \widehat{MSE}^{i}_{\tilde{m}_f}}{\widehat{MSE}^{i}_{\tilde{m}_f}} \right| * 100 \leq 2. \tag{3.3}$$

The MSE values decreased at each iteration and stabilized around 0.25 at iteration 85. The prediction process was terminated at this iteration.

Figures 3.16 and 3.17 show the $RMSE$ and R^2 for the fuel rate, *BMEP*, *RGF*, and *CA50*. We observe that the average absolute error between the actual outputs and predicted outputs reduces as the metamodels are trained with more training points. Although the RMSE of fuel rate, *RGF*, and *CA50* show convergence after 80 iterations, adding more training points could further reduce the RMSE of *BMEP*. Figure 3.17 shows that, in general, as training points are continuously added, the correlation between the actual and predicted outputs improves.

At part load condition, we seek the ICL and ECL that minimize the fuel rate at each combination of engine speed and BMEP under the constraints of RGF being less than 25% and CA50 being less than 25 degrees CA after TDC firing. We must also make sure that the predicted BMEP is within ±5 kPa of the targeted BMEP. The allowed ICL phasing range is between 80 to 140 degrees CA after TDC gas exchange and the allowed

ECL phasing range is between -140 to -80 degrees CA which is before TDC gas exchange. The optimum values of ICL and ECL can be thus obtained by solving the following optimization problem

$$\min_{\underline{\boldsymbol{x}}} \widetilde{\dot{m}_f}(\underline{\boldsymbol{x}}, RPM, BMEP)$$

$$\text{such that:} \quad \begin{aligned} &\widetilde{RGF}(\underline{\boldsymbol{x}}, \underline{\boldsymbol{P}}) < 25\% \\ &\widetilde{CA50}(\underline{\boldsymbol{x}}, \underline{\boldsymbol{P}}) < 25\ degCA \\ &\left|\widetilde{BMEP}(\underline{\boldsymbol{x}}, \underline{\boldsymbol{P}}) - \text{BMEP}\right| \leq 5\ kPa \\ &-140 \leq \text{ECL} \leq -80 \\ &80 \leq \text{ICL} \leq 140 \end{aligned} \tag{3.4}$$

Figure 3.18 shows the relative error between the actual $\dot{m}_{f\ opt}(\underline{\boldsymbol{x}})$ and the predicted $\widetilde{\dot{m}_{f\ opt}}(\underline{\boldsymbol{x}})$ at part load using 111 training points after 85 iterations. We observe an acceptable accuracy of the predicted fuel rate. However, the fuel rate relative error is high at the bottom-left corner of the map. Note that at low speed and low load, the fuel rate is very low and therefore, the relative error is high (see Figure 3.2).

Figures 3.19 and 3.20 show the difference between actual and predicted RGF and CA50, respectively, in terms of engine speed and BMEP. Both plots show very good accuracy of RGF and CA50 for part load.

Figure 3.21 shows the difference of ICL between the actual and predicted part load map. At specific operating conditions, the ICL at the predicted minimum fuel rate does not match the actual ICL. For example, at 5000 rpm and 300 kPa BMEP, even though the relative error between the actual and predicted fuel rate is minimum (around ±1%) their corresponding ICL are very different by approximately 40 degrees. This

observation implies that for certain operating conditions the metamodel was not able to represent the physics since the minimum fuel rate does not correspond to the proper ICL.

Figure 3.22 shows the difference of ECL between the actual and predicted part load map. The difference between the ECLs are slightly high, but lie in an acceptable range except in a few operating points (blue region with 100 kPa BMEP at 4000 and 5000 rpm).

For this example, it was observed that the number of selected training points using the maximum MSE criterion is higher at the upper and lower bounds of the engine speed and BMEP while less points were selected in between the upper and lower bounds. Figure 3.23 shows the number of selected training points at each operating condition. Because more training points are placed at the upper and lower bounds, less information is provided for the middle of the map resulting in a reduced prediction accuracy. The training points of Figure 3.23 are selected using the maximum predicted MSE of the fuel rate. To reduce the higher error in the middle of the map, more training points are needed.

3.4 Summary

A new process was introduced to generate an engine fuel map with less computational effort while maintaining accuracy. Both the full load and part load conditions were considered. Kriging metamodels were first developed for the engine BMEP, fuel rate, RGF and CA50. The metamodels were then used in two separate optimization problems to obtain the full load and part load parts of the map using constraints which ensure acceptable combustion characteristics. The estimated fuel map was compared with the actual map that was created using a full factorial DOE to demonstrate its accuracy. For full load, we observed that:

- A small number of DOE points (121 points) was enough to generate the WOT curve. This number is smaller than some existing “optimized” methods used in the automotive industry which use around 300 DOE points. Note that a computationally intensive run of the GT Power software is required for each DOE point (training point for the metamodels).
- The accuracy of the estimated map was very good.

For part load, we observed that:

- Only 110 DOE points were used to train the developed metamodels compared to almost 3000 points existing approaches require.
- The estimated maps for fuel rate, BMEP, RGF and CA50 were very accurate compared to actual maps.
- The ICL and ECL cam position maps, for the predicted fuel rate map were not consistent with the actual data at certain operating conditions. This implies that more training points are needed to fill the design space and train the metamodels in order to improve the prediction performance.

The presented approach reduces the computational time to generate the fuel map by a factor of 14 compared to a heuristically optimized conventional method used by the industry.

In future studies, we aim to further improve the accuracy of the estimated map at part load.

Table 3.1: Critical engine parameters

Parameter name	Values and descriptions
Displacement	2.0 [L]
Number of cylinders	4
Bore / Stroke	86 / 86.07 [mm]
Compression ratio	9.5:1
Max. intake valve lift	10.2 [mm]
Max. exhaust valve lift	10.2 [mm]
Allowed ICL phasing range	80 to 140 degCA aTDC
Allowed ECL phasing range	-140 to -80 degCA aTDC
Fuel injection type	Port Fuel Injection

Table 3.2: Details on fuel map generation

Operating conditions	Part load	Full load
Inputs	Engine speed, BMEP, ICL, ECL	Engine speed, BMEP, ICL, ECL
Outputs	$\dot{m}_f$, BMEP, RGF, CA50	BMEP, RGF, CA50
Constraints	$\lvert \text{BMEP} - \text{BMEP}_{\text{req}} \rvert \leq 5$ kPa, $\text{RGF} \leq 25\%$, $\text{CA50} \leq 25$ deg CA	$\text{RGF} \leq 25\%$, $\text{CA50} \leq 25$ deg CA
Objective	Minimize $\dot{m}_f$	Maximize BMEP

Table 3.3: Range and levels of each input variable

Operating conditions	Part load			Full load		
DOE information	Lower limit	Upper limit	# of levels	Lower limit	Upper limit	# of levels
Engine speed [rpm]	1000	6000	6	1000	6000	6
BMEP [kPa]	100	1000	10	NA	1500	1
ICL [degCA aTDC]	80	140	7	80	140	7
ECL [degCA aTDC]	-140	-80	7	-140	-80	7

Table 3.4: Range and levels of prediction points

Operating conditions	Full load		
DOE information	Lower limit	Upper limit	# of levels
Engine speed [rpm]	1000	6000	6
ICL [degCA aTDC]	80	140	31
ECL [degCA aTDC]	-140	-80	31

Table 3.5: Criteria for acceptable accuracy of fuel map

Convergence criteria	Operating conditions
$\left\lvert\frac{\widetilde{BMEP} - BMEP}{BMEP}\right\rvert * 100 \leq 5$	Full load
$\left\lvert\frac{\widetilde{\dot{m}_f} - \dot{m}_f}{\dot{m}_f}\right\rvert * 100 \leq 5$	Part load
$\left\lvert\widetilde{RGF} - RGF\right\rvert \leq 5$	Full and part loads
$\left\lvert\widetilde{CA50} - CA50\right\rvert \leq 5$	Full and part loads

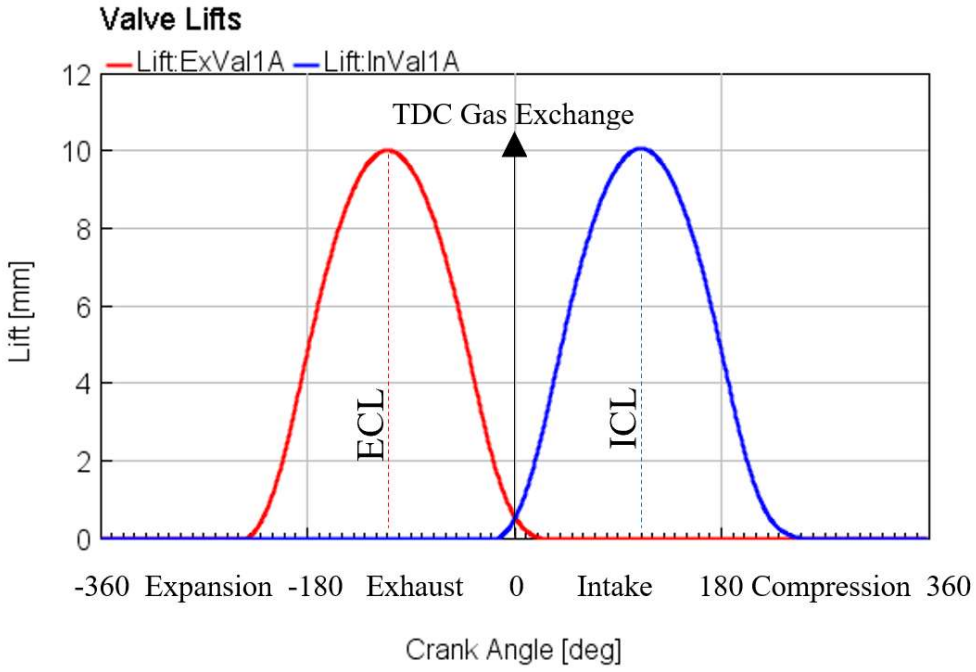

Figure 3.1: Definition of intake and exhaust camshaft positions

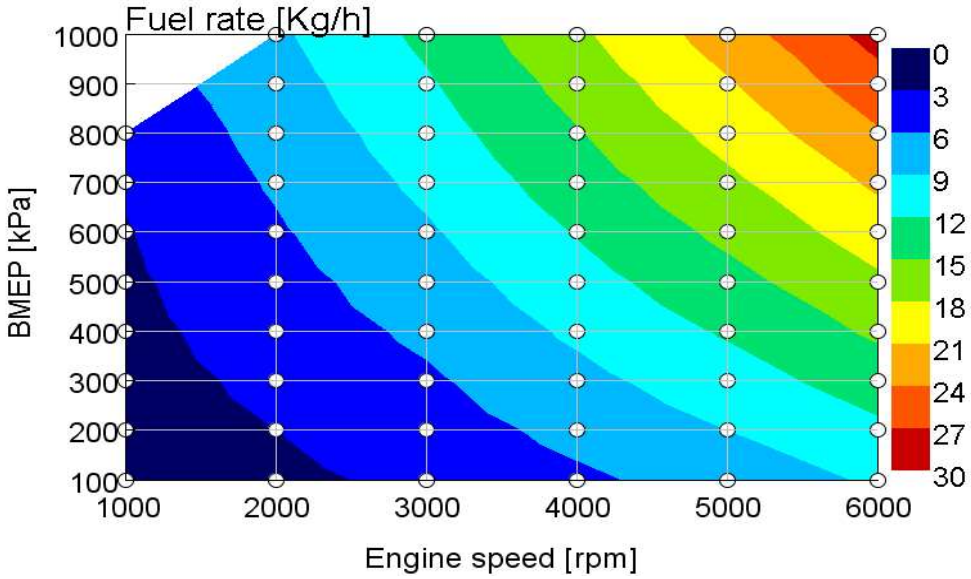

Figure 3.2: Fuel flow rate at part load condition

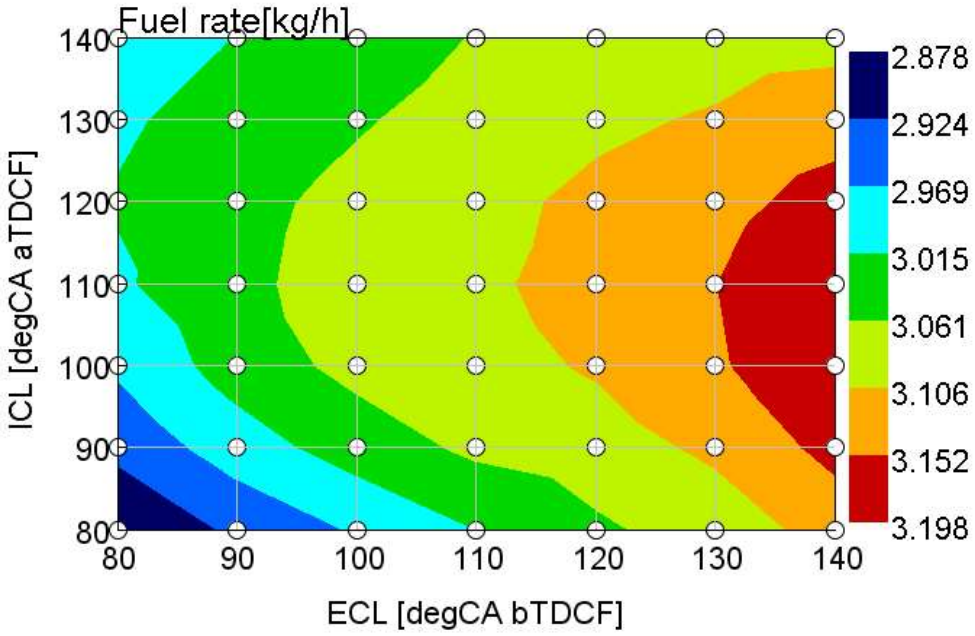

Figure 3.3: Fuel flow rate at 2000 rpm and 200 kPa of BMEP

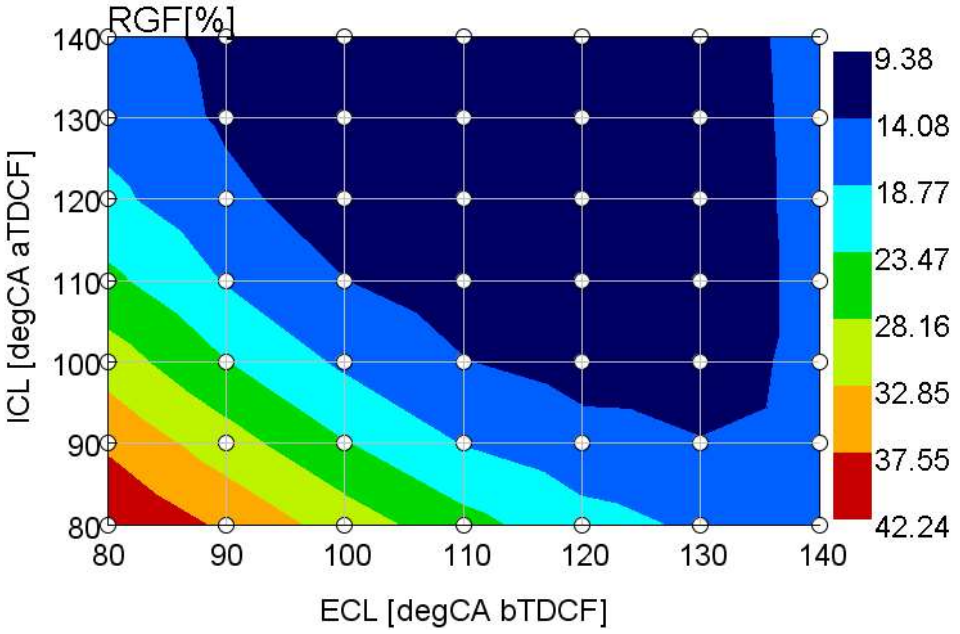

Figure 3.4: RGF at 2000 rpm and 200 kPa of BMEP

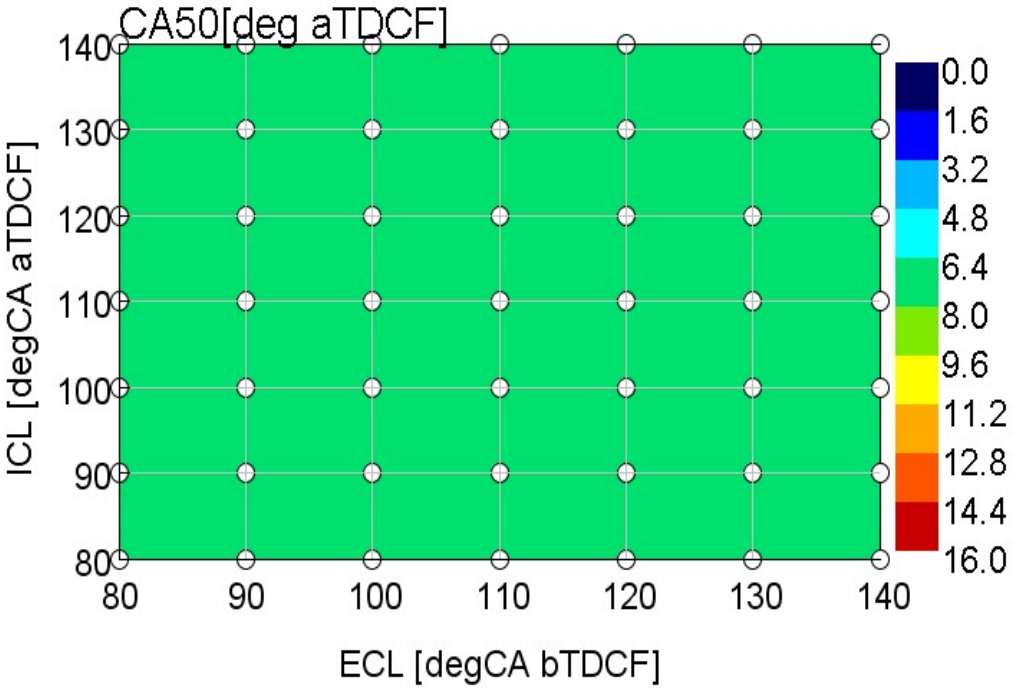

Figure 3.5: CA50 at 2000 rpm and 200 kPa of BMEP

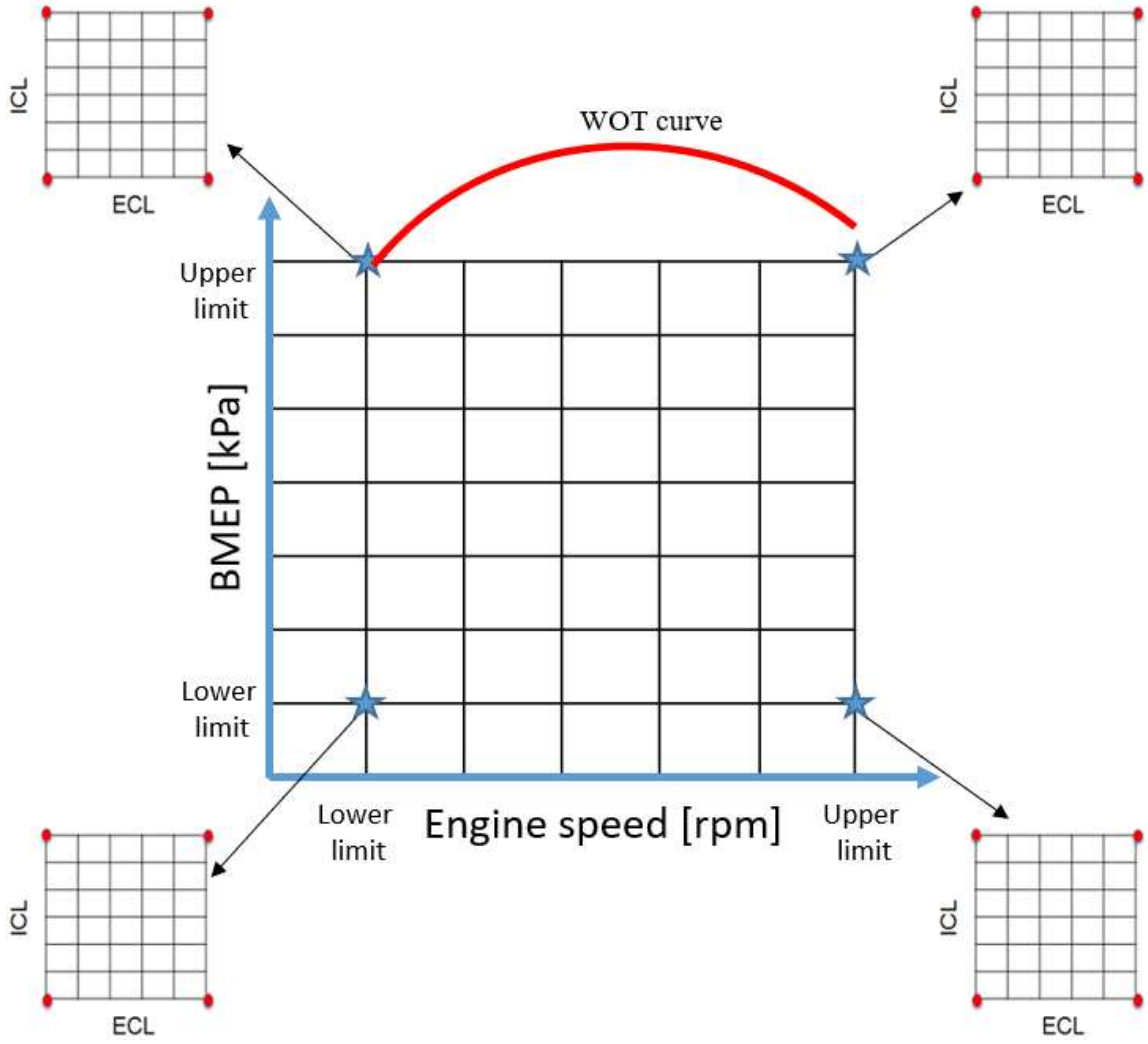

Figure 3.6: Selection of initial training points for part load

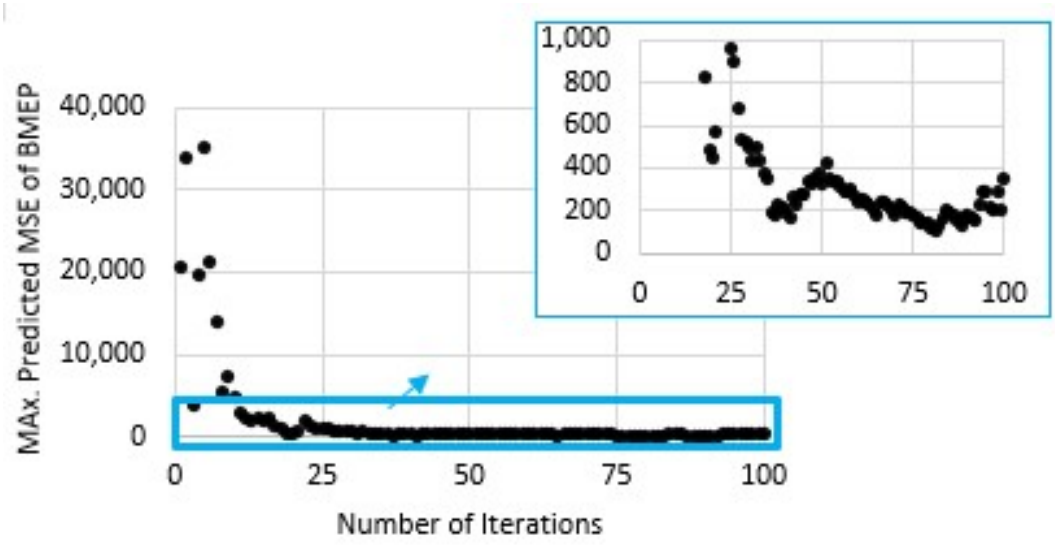

Figure 3.7: Predicted maximum MSE of BMEP at each iteration

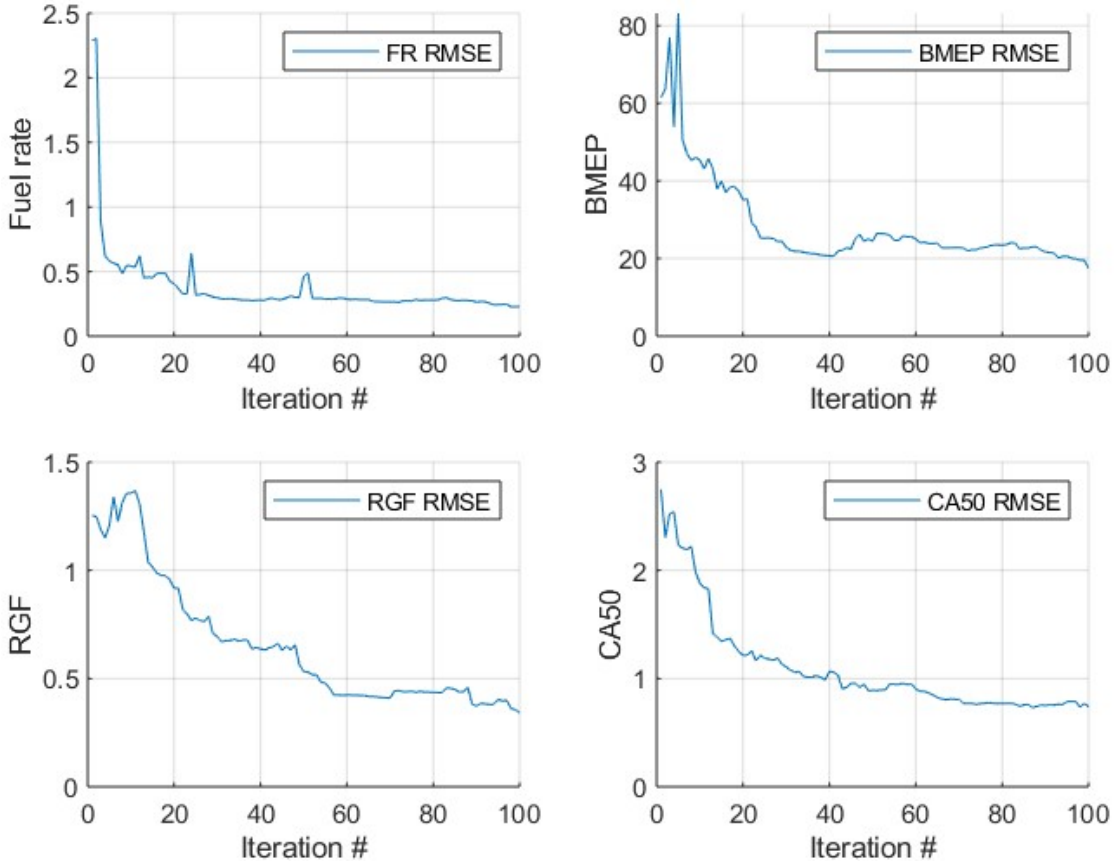

Figure 3.8: RMSE of actual vs. predicted outputs at each iteration

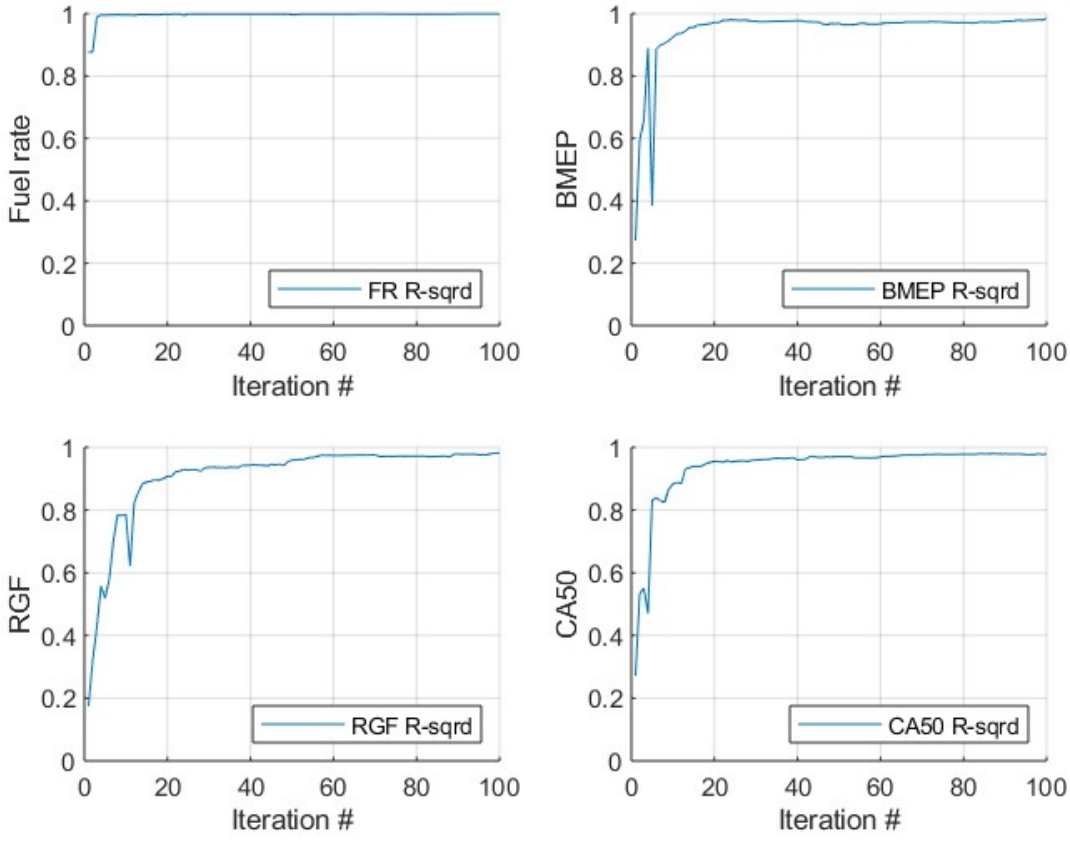

Figure 3.9: R^2 of actual vs. predicted outputs at each iteration

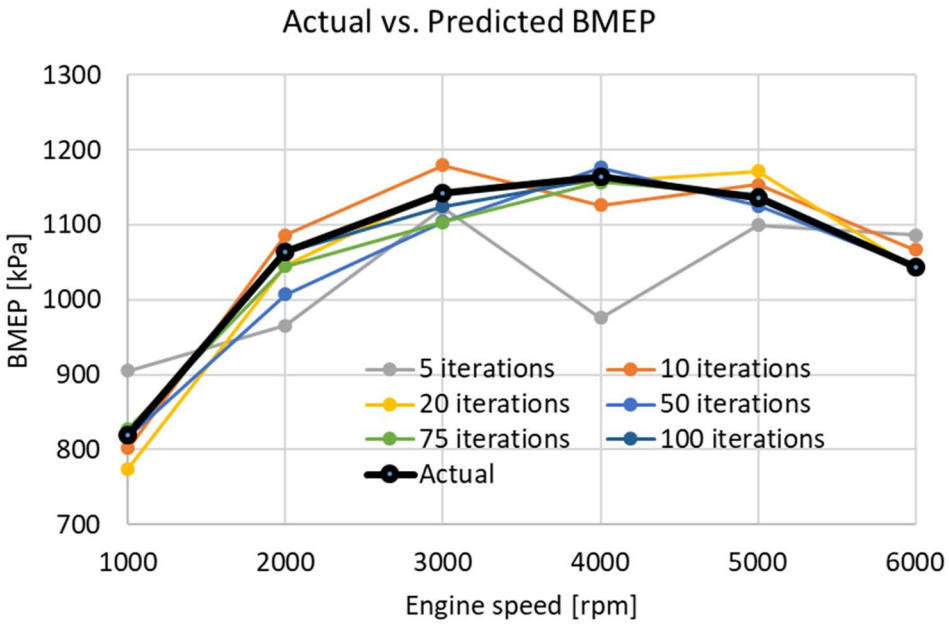

Figure 3.10: Predicted vs. actual BMEP at different iterations

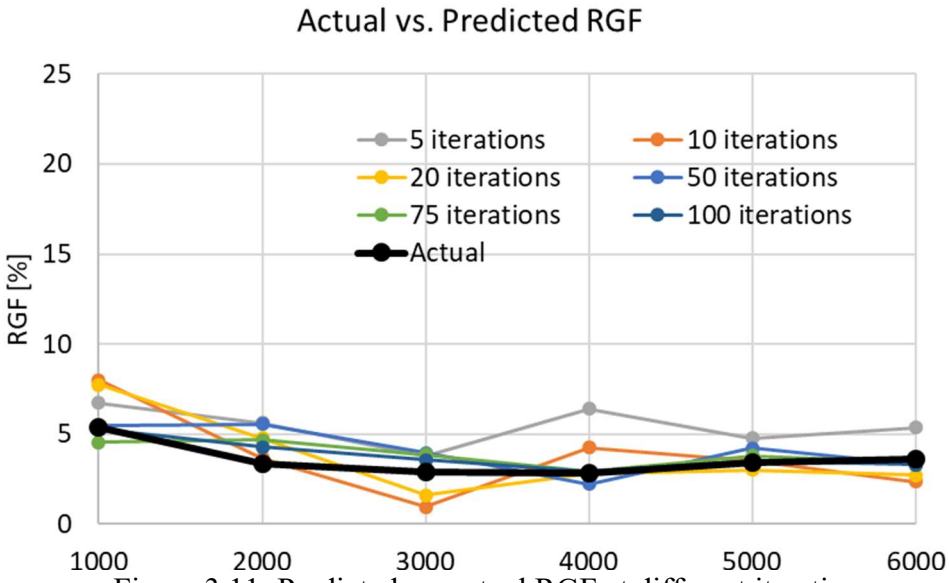

Figure 3.11: Predicted vs. actual RGF at different iterations

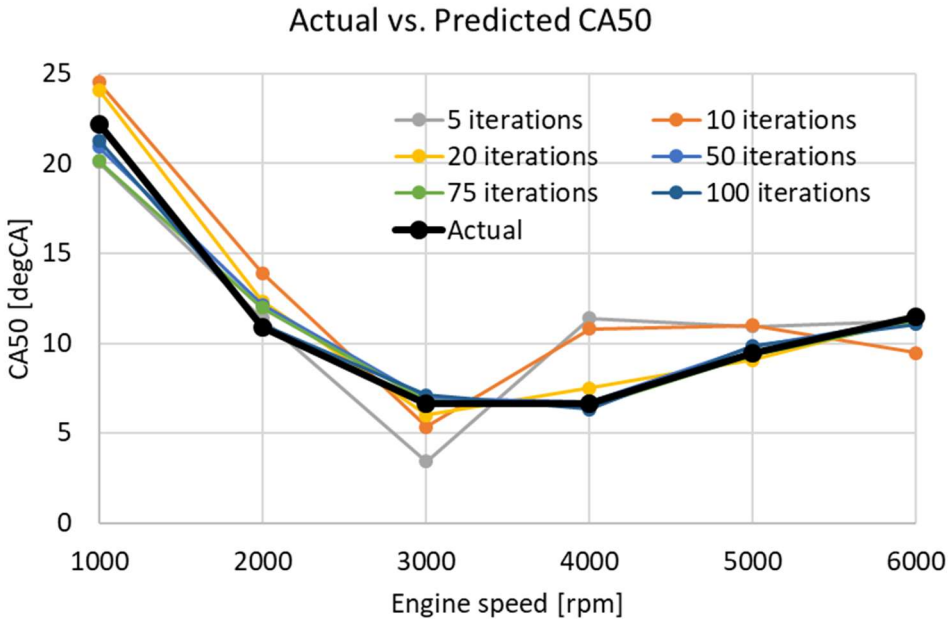

Figure 3.12: Predicted vs. actual CA50 at different iterations

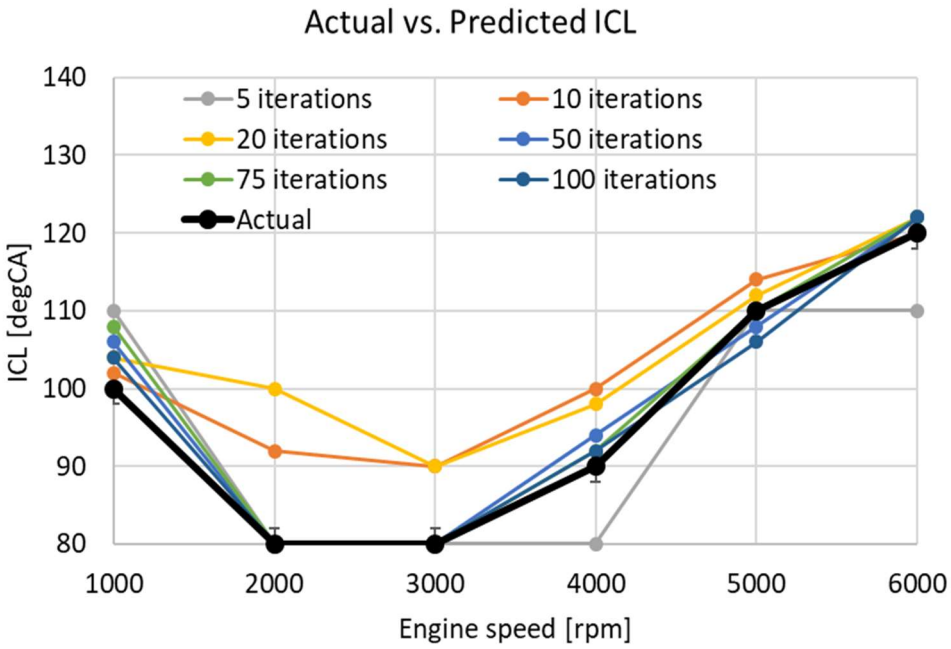

Figure 3.13: ICL for actual and predicted WOT curve at different iterations

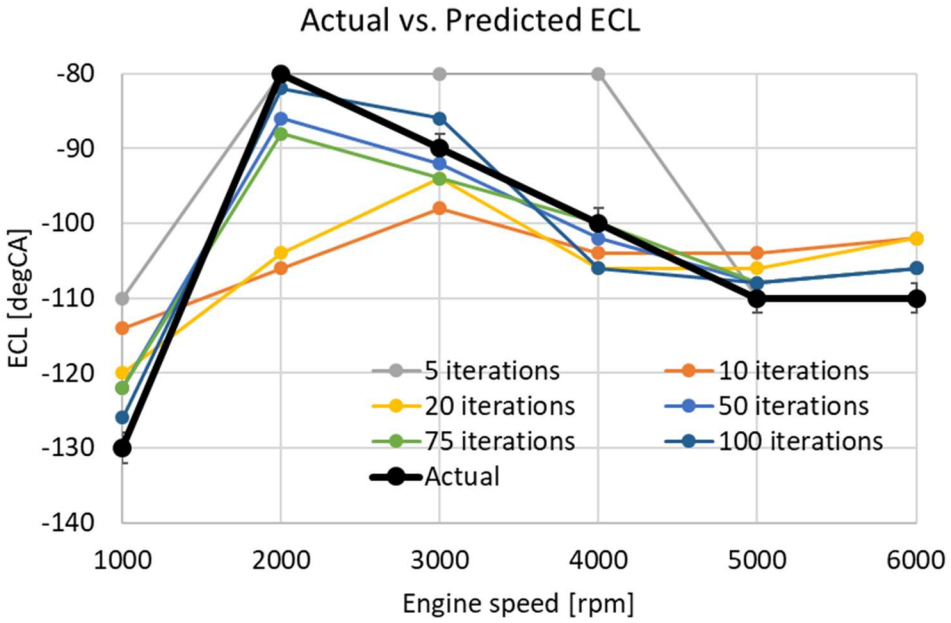

Figure 3.14: ECL for actual and predicted WOT curve at different iterations

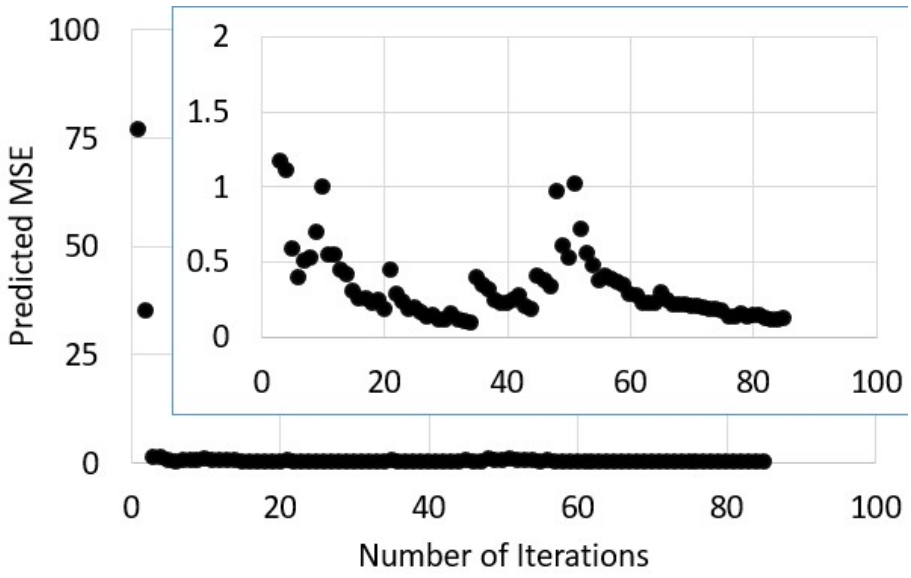

Figure 3.15: MSE of fuel rate at each iteration

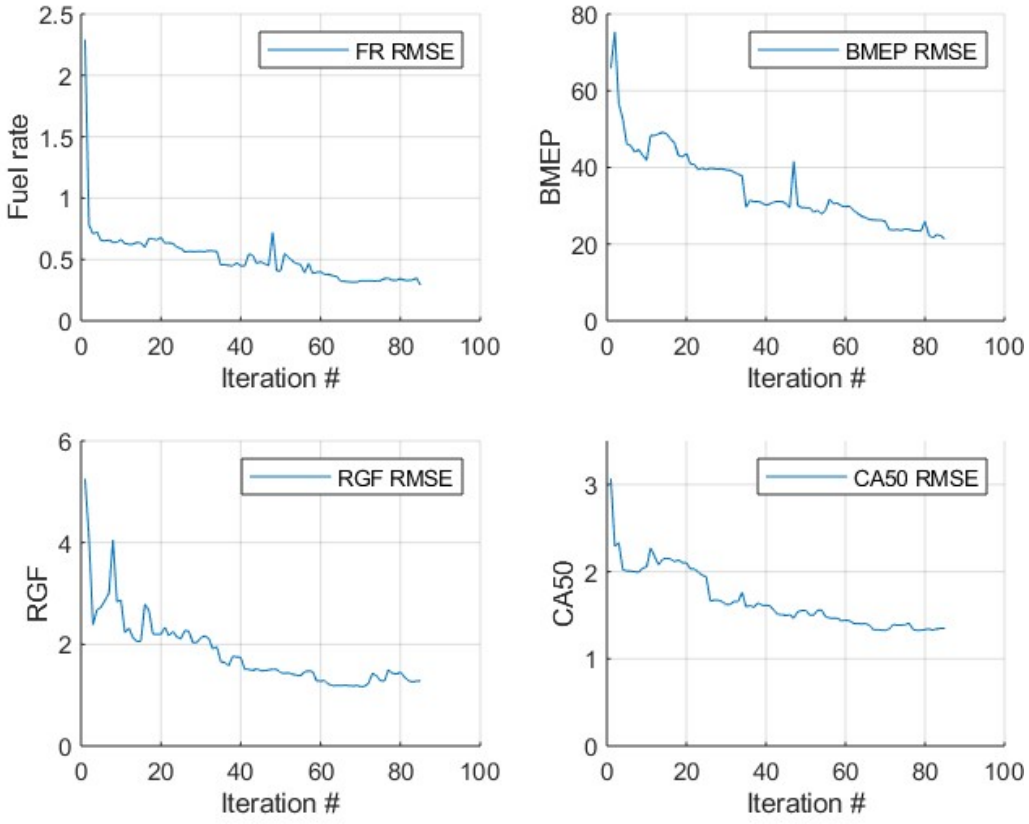

Figure 3.16: RMSE of actual vs. predicted outputs at each iteration

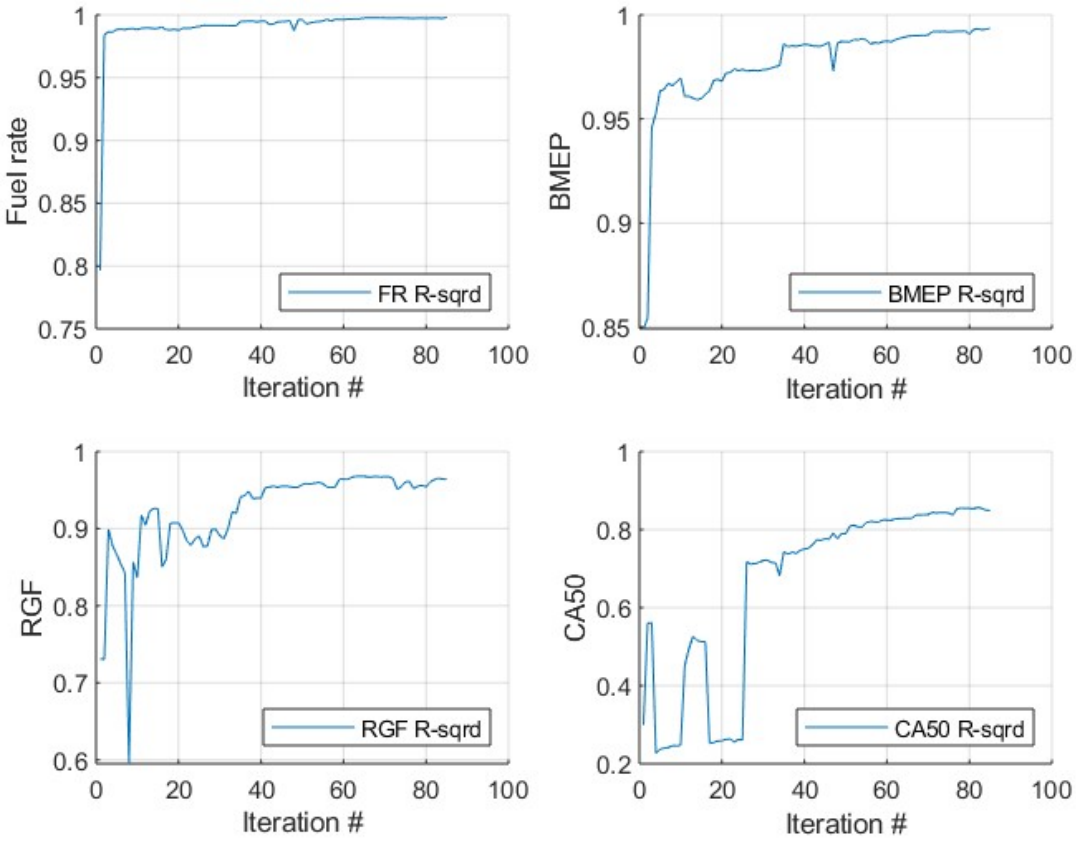

Figure 3.17: R^2 of actual vs. predicted outputs at each iteration

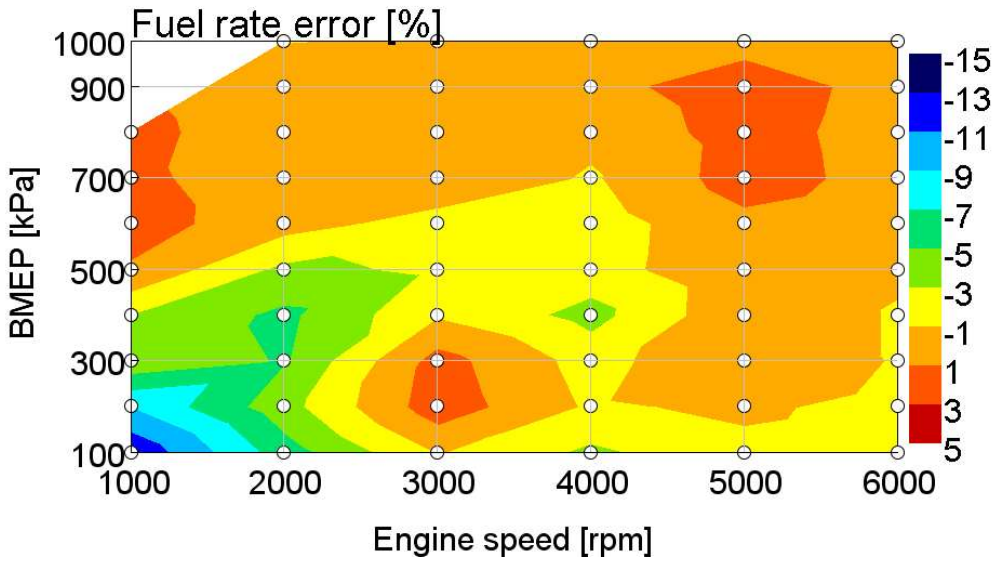

Figure 3.18: Relative error between actual and predicted fuel map

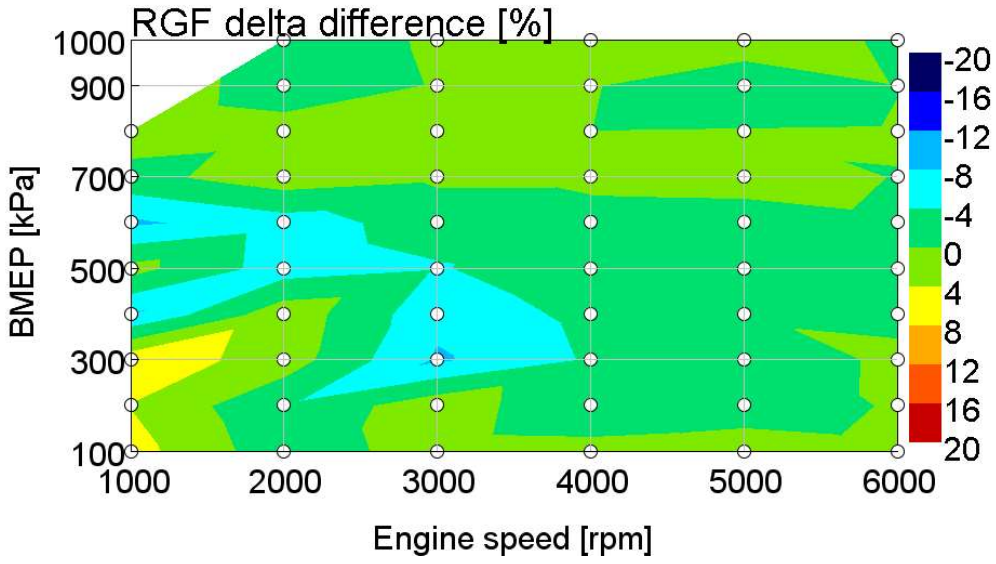

Figure 3.19: Difference between actual and predicted RGF

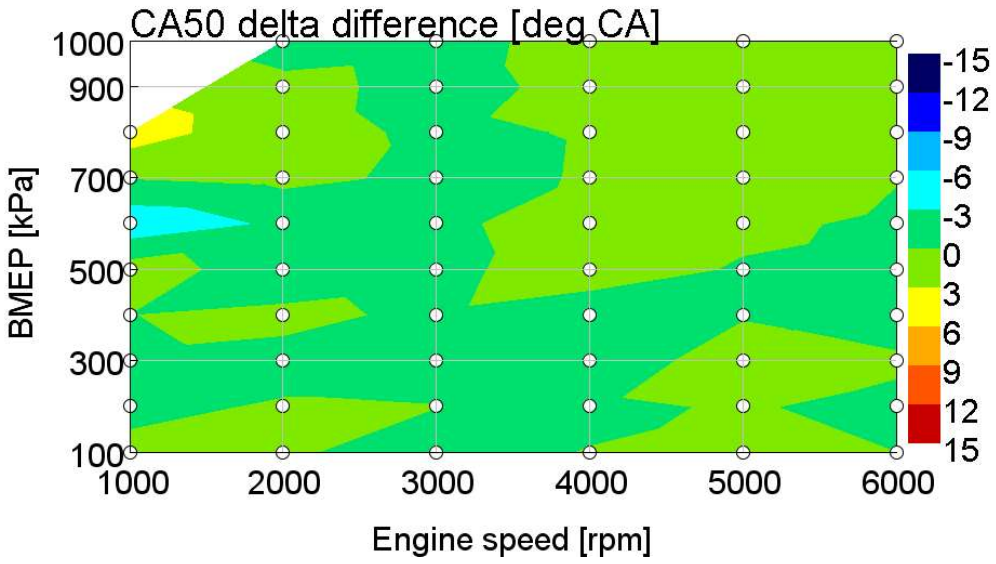

Figure 3.20: Difference between actual and predicted CA50

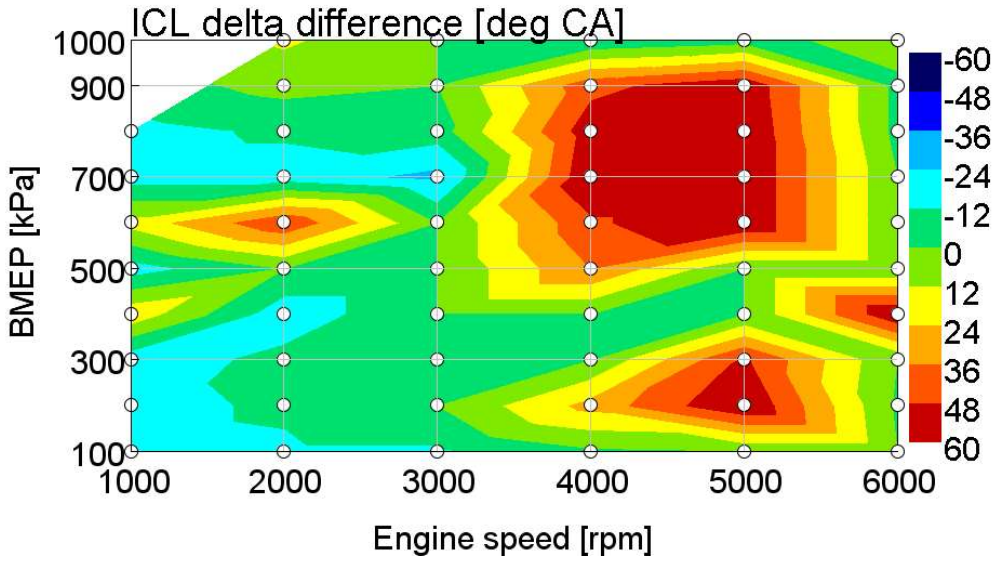

Figure 3.21 Difference between ICL of actual and predicted part load map

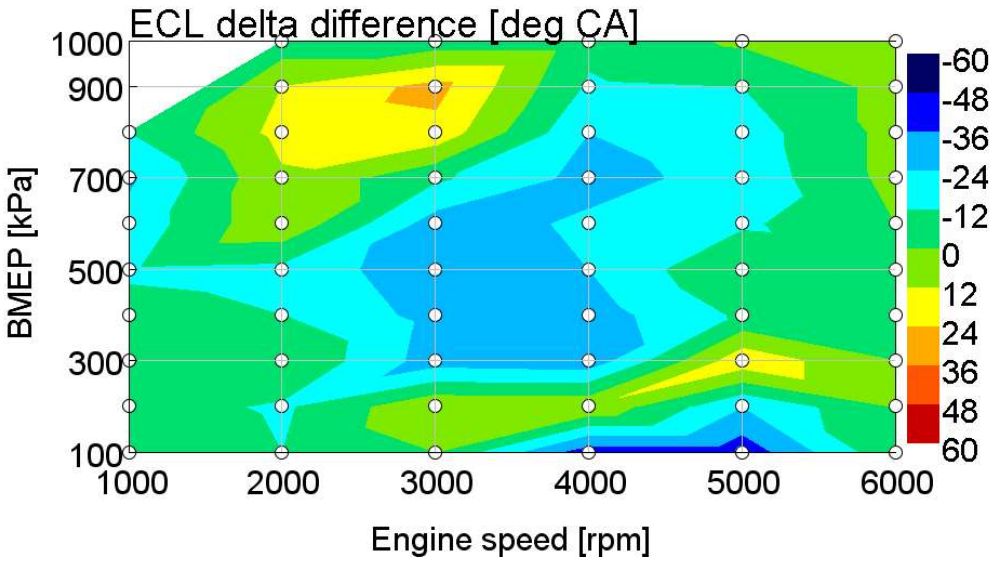

Figure 3.22: Difference between ECL of actual and predicted part load map

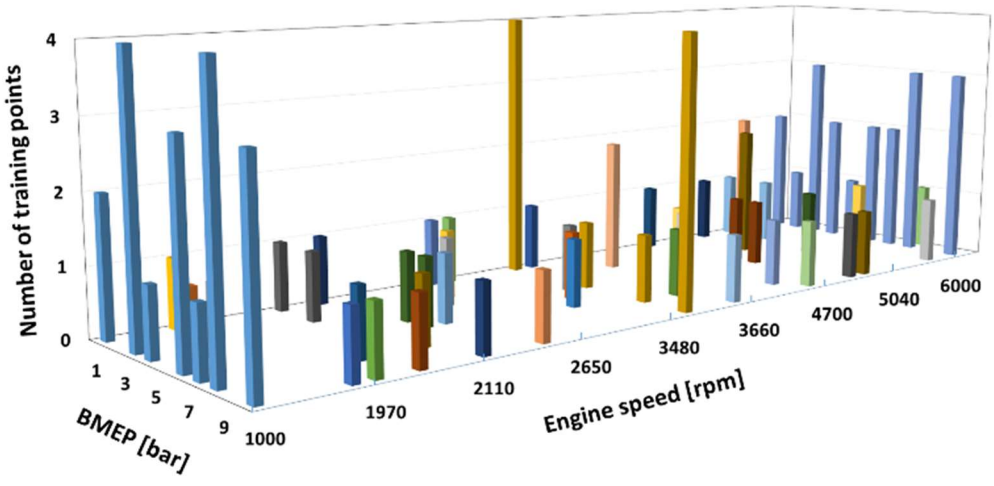

Figure 3.23: Number of training points at different operating conditions to train the metamodels

CHAPTER FOUR

METHODOLOGIES TO PREDCIT TRANSIENT SPARK TIMING PROFILE FOR A TIP-OUT MANEUVER

A powertrain consists of an engine, transmission, and driveline. The engine generates power which is transferred to the wheels through the transmission and the driveline. The power however, generated by the engine can cause oscillating forces which can lead to unwanted vibration and noise. Gear rattle and driveline clunk are examples of unwanted noise in the transmission and driveline respectively that are caused by the engine.

Clunk is the noise generated by torsional vibrations in the driveline. These torsional vibrations are originated from abrupt changes in the engine output torque such as in tip-in/tip-out maneuvers. These two dynamic behaviors of vehicle are referred to the transient conditions when the driver suddenly requests for a quick vehicle acceleration /deceleration by depressing/releasing the accelerator pedal. In tip-in maneuver more torque is requested while in tip-out maneuver less torque is desired. The literature provides ways to reduce the clunk phenomenon [82-85]. One proposed solution is to reduce the lash in the gear set. This reduces the acceleration of the driver gear teeth in the lash zone which in turn minimizes the clunk. However, lash reduction is not an efficient solution because a finite amount of lash is always required for the gears to rotate. Thus, there is no guarantee to eliminate the clunk noise [86, 87]. Also, reducing or removing the lash requires new designs which are costly and have physical limitations. Other studies have showed that increasing driveline damping reduces the relative speed

between the gear sets as they collide by adding resistance to the gear set in the lash zone. However, this approach reduces the driveline efficiency and fuel economy.

An efficient way to mitigate this problem is to use torque shaping, also known as torque management, where we specify an engine output torque profile to minimize the driveline clunk. Different studies have suggested how to specify the desired torque shape [83, 86-89]. However, it is important to know how to generate a specific engine torque considering that the engine output torque depends on certain independent parameters. The value of each independent parameter for any conditions is determined using engine calibration.

4.1 Overview of Torque Shaping During Tip-out Maneuver

Most car manufacturers follow different but similar approaches to develop and produce vehicles. Engine calibration is a very important step in product development. The engine is calibrated in order to pass a variety of tests and satisfy certain regulations over a range of conditions.

There are many different steps involved in calibrating an engine for steady-state and transient conditions. One step of steady-state calibration is to generate a fuel map also known as engine map [90]. A fuel map consists of two regions known as the full load and part load and it provides engine characteristics in terms of engine speed and load [91]. Specifically, the fuel map reports how to achieve a minimum fuel consumption at part load and maximum load at full load conditions by controlling a set of independent design parameters for the entire space of engine operating conditions. Traditionally, a fuel map is generated by running the actual engine in a test cell. However, an alternative approach is to generate it using physics-based model simulations considering that the

physics-based model is well correlated with the actual engine. This virtual approach reduces the product development cost and time significantly. The generated engine maps using simulations should be however, tested and verified using a real engine in a test cell.

Unlike steady-state conditions, the load and speed vary through time during a transient operation which is common for actual driving. Consequently, actuators such as throttle, spark timing, etc. are constantly trying to meet the requested dynamic load and speed. The rate of change of the actuator may not be however, the same as the change in engine parameters such as air flow, Manifold Air Pressure (MAP), and combustion parameters. Thus, using a model developed for steady-state conditions might not be suitable for transient conditions. Figure 4.1 shows a transient maneuver when the actuators use the steady-state fuel map. This is a tip-out scenario in which the desired torque request, shown in black, drops from 110 Nm at five seconds and reaches -20 Nm at six seconds while the engine is running on stoichiometric conditions.

Before the five second time point, when the tip-out maneuver starts, the engine output torque (in red) is in steady-state and in good agreement with the desired torque since the actuators are not varying with time. After five seconds, the desired torque drops and reaches -20 Nm at six seconds. Actuators such as the throttle angle, spark timing, ICL (Intake cam Center Line) and ECL (Exhaust cam Center Line) update their values instantaneously using the steady-state fuel map data. The actuator changes from 5 to 6s, result in the engine output torque. The Figure 4.1 shows clearly that the engine output torque does not follow the target torque since the actuator commands were obtained from the steady-state fuel map.

In order to control actuators such as throttle valve, cam timings, and EGR valve (or waste gate diameter in turbocharged engines), sophisticated controllers are required in addition to the steady-state fuel map in order to obtain the desired torque shape. The controllers for transient maneuvers are complex depending on the engine system and also require a lot of engine data. Depending on the actuator type, fuel and air path control system reactions are different and need certain calibration work. Many researchers and engineers have attempted to improve the transient behavior of an engine under different conditions [92-95].

In this research, we use two different approaches to determine the spark timing profile to achieve a desired engine torque profile during a tip-out maneuver. These methods are both less complicated compared to the required controllers for transient maneuvers and requires less amount of data. Using these methods, we can predict and use a spark timing profile as the only input parameter for a tip-out maneuver to generate a desired torque profile.

We use GT-Power as the engine simulation tool. The standalone engine model is a four-cylinder 2.0L naturally aspirated gasoline engine with a port fuel injection system. Each cylinder has two intake and two exhaust valves. The engine is equipped with a VVT system in which the intake and exhaust camshaft positions are independently controlled. Table 4.1 provides the engine parameters.

4.2 Creation of Desired Torque Profile Using SVD and Kriging (Method 1)

In this section, we apply the time-dependent metamodeling approach to generate the required spark timing profile to obtain a desired torque profile. The section discusses all steps and simultaneously demonstrates the proposed approach using an example.

We use a standalone engine system model from the GT-Power library. The model simulates a tip-out maneuver at constant engine speed of 2000 rpm. It is assumed that the torque convertor is locked as the vehicle moves with constant speed. At $\mathbf{t_0 = 5}$ s, the desired torque at the flywheel starts to drop from 110 Nm and reaches -20 Nm at $\mathbf{t = 6}$ s. The objective is to predict a spark timing profile which produces a desired torque.

We assume that a third-order polynomial is adequate to represent the spark timing profiles which can be used to determine the unique spark timing profile which results in the desired torque profile. Equation (4.1) provides a parametric description of the spark timing profiles $\boldsymbol{s}\ (\boldsymbol{t})$ as a function of time

$$\boldsymbol{s}\ (\boldsymbol{t}) = \boldsymbol{d_0} + \boldsymbol{d_1} * (\boldsymbol{t} - \boldsymbol{t_0}) + \ \boldsymbol{d_2} * (\boldsymbol{t} - \boldsymbol{t_0})^{\boldsymbol{2}} \ + \ \boldsymbol{d_3} * (\boldsymbol{t} - \boldsymbol{t_0})^{\boldsymbol{3}}. \tag{4.1}$$

The time domain is $\boldsymbol{t} \in [\mathbf{5}, \mathbf{6}]$ and

$$\boldsymbol{D_p} = \{\boldsymbol{d_1^{(p)}}, \boldsymbol{d_2^{(p)}}, \boldsymbol{d_3^{(p)}}\} \tag{4.2}$$

provides the three independent coefficients in Equation (4.1) to be determined. The subscript or superscript *p* indicates a design point which is not included in the design points of the training set. The initial spark timing at $\boldsymbol{t_0 = 5s}$ corresponding to the torque at $\boldsymbol{t_0}$ is known

$$\boldsymbol{s}\ (\boldsymbol{t_0}) = \ \boldsymbol{d_0} \ = \ \boldsymbol{-20.19\ degCA\ aTDCF}. \tag{4.3}$$

Figure 4.2 shows the flowchart of the overall process to determine the independent coefficients of $\boldsymbol{D_p}$ defining the desired spark timing and the corresponding torque profiles with acceptable accuracy.

The group-based space-filling DOE algorithm of [96] is used. This DOE algorithm creates multiple groups of points where each group has space-filing properties by itself and unions of groups also have space-filling properties. The advantage of using

this DOE approach is that the minimum number of design sites is used to determine $\boldsymbol{D_p}$. This is important because for each design site, Equation (4.1) is used to determine the spark timing and then the computationally intensive GT-Power is used to obtain the corresponding torque profile. In this study, we generated 32 groups of 4 design sites each. Thus, the total number of generated design sites is 32*4=128. The designs are grouped in the following matrix $[\boldsymbol{D}]$ as

$$[\boldsymbol{D}] = \begin{bmatrix} \boldsymbol{d_1^{(1)}} & \boldsymbol{d_2^{(1)}} & \boldsymbol{d_3^{(1)}} \\ \vdots & \vdots & \vdots \\ \boldsymbol{d_1^{(128)}} & \boldsymbol{d_2^{(128)}} & \boldsymbol{d_3^{(128)}} \end{bmatrix}. \tag{4.4}$$

The first 4 rows of $[\boldsymbol{D}]$ include the first 4 design sites, the next 4 rows include the design sites of group 2, etc. The first 4 designs are used as training points for the first iteration (i = 1) of the algorithm of Figure 4.2.

Table 4.2 shows the upper and lower range of any design. The ranges are properly selected to generate a number of different spark timing profiles which can produce potential torque profiles for the tip-out maneuver.

Figure 4.3 shows the generated spark timing profiles (left) for the first group of 4 DOE points. It also shows the desired torque profile (thick black line). The corresponding torque profiles for the 4 spark timing profiles are shown on the right. The torque profiles were obtained by running GT-Power using the generated spark profiles.

At this point, we check the quality of the generated training points using two criteria. The first criterion makes sure that each generated spark timing does not result in engine misfire. The second criterion ensures that the desired torque profile is within the "cloud" of torque profiles from the training designs ensuring interpolation. The training points which produce a spark timing profile which is retarded beyond 45 degCA aTDCF

are removed because they cause an engine misfire. Figure 4.4 shows a case in which a misfire occurs.

The cylinder pressure traces indicate that the combustion occurs in each cylinder from 5 to 5.75s as the spark timing retards. The Indicated Mean Effective Pressure (IMEP) which is the work delivered to the piston over the compression and expansion stroke per cycle is greater than zero. When the spark timing passes the misfiring limit (t >5.75s), no combustion occurs in the cylinders and the IMEP becomes zero indicating engine misfire. So, in Figure 4.4, the training points producing a spark timing profile greater than 45 degCA aTDCF should be removed from the set of training points.

Also, before creating a metamodel using SVD-Kriging interpolation, it is necessary to have a set of training points which produce torque profiles which surround the desired torque profile so that the latter is obtained by interpolation. This is ensured by the second quality criterion. If this condition is not satisfied more training points are added until the condition is met. Figure 4.5 shows the spark timing and the torque profiles corresponding to 6 training points after 4 iterations. The desired torque profile is located within the upper and lower range of the training torque profiles from 5 to 6s. Also, there are no spark timing profiles retarded beyond 45 degCA aTDCF indicating no misfire. Note that the number of training points after 4 iterations should be 4*4=16 but 10 profiles were removed to avoid misfiring leaving only 6 training profiles in Figure 4.5.

The 6 torque profiles at iteration 4 are used to define a response matrix $[\boldsymbol{X}]$ according to Equation (4.5). The time domain $\boldsymbol{t} \in [\mathbf{5}, \mathbf{6}]$ was discretized using a uniform grid of 100 time instances. The response matrix $[\boldsymbol{X}]$ was decomposed into the $[U]$, $[S]$, and $[V]$ matrices using SVD as

$$[\boldsymbol{X}]_{\mathbf{6\times100}} = [\boldsymbol{U}]_{\mathbf{6\times6}}\,[\boldsymbol{S}]_{\mathbf{6\times6}}\,[\boldsymbol{V}]^{\boldsymbol{T}}{}_{\mathbf{6\times100}}\,. \tag{4.5}$$

The number of singular values in matrix $[\boldsymbol{S}]$ is equal to the number of rows in the response matrix $[\boldsymbol{X}]$. If the number of singular values is large, the first few dominant singular values are kept and the remaining are ignored. As a criterion, the ratio of the last kept singular value over the first one may be less than 0.01. The desired torque response was also discretized using the same uniform grid of 100 time instances to form the row vector $\{\boldsymbol{x_p}\}$. The desired row $\{\boldsymbol{u_p}\}$ of the modal matrix [U] is then provided by Equation (2.59) discussed in Chapter 2.

In order to calculate the required spark timing (Equation 4.1) to obtain the desired torque profile, the three components of vector $\boldsymbol{D_p} = \{\boldsymbol{d_1^{(p)}}, \boldsymbol{d_2^{(p)}}, \boldsymbol{d_3^{(p)}}\}$ must be estimated. The parameter $\boldsymbol{d_0}$ in Equation (4.1) is provided by Equation (4.3). Three metamodels are developed for $\boldsymbol{d_1^{(p)}}$, $\boldsymbol{d_2^{(p)}}$, and $\boldsymbol{d_3^{(p)}}$. For each of the three metamodels, the rows of matrix $[\boldsymbol{U}]$ are used as input, and each column of [*D*] is used as output to train a Kriging metamodel. The three metamodels are then used to obtain the three unknown components of vector $\boldsymbol{D_p}$ which form $\{\boldsymbol{u_p}\}$. The latter is provided from Equation (2.59). The vector $\boldsymbol{D_p}$ is then used to calculate the spark timing using Equation (4.1). The spark timing is used as input to the GT-Power software to calculate the corresponding torque profile.

Figure 4.6 shows the spark timing and corresponding torque profiles at the fourth iteration. The dashed curve on the left shows the spark timing profile corresponding to the predicted coefficients in $\boldsymbol{D_p}$. The dashed curve of the right panel shows the corresponding torque profile. The desired torque profile is shown in solid black.

Figure 4.6 clearly shows a sizeable difference between the desired torque (solid-black curve) and the torque (dashed-black curve) resulting from the predicted spark timing. A large difference between the actual and predicted torque profiles indicates that either the number of training points is not sufficient or there are some training points (profiles) which "contaminate" the design space reducing the accuracy of the prediction. Such points are eliminated. For that, the time instance $\boldsymbol{t_{err}}$ is determined where the difference between the desired and the predicted torque profiles $\boldsymbol{\Delta T_{pred}}$ becomes maximum. Figure 4.7 shows $\boldsymbol{t_{err}}$ and $\boldsymbol{\Delta T_{pred}}$ for the fourth iteration. For each torque profile in the training set, the maximum difference between the desired torque and the torque profile at $\boldsymbol{t_{err}}$ is also determined. If the latter is greater than $\boldsymbol{\Delta T_{pred}}$, the torque profile is eliminated from the training set. The eliminated torque profiles are mark with a red dot at $\boldsymbol{t_{err}}$. The elimination of torque profiles from the training set is performed in the next iteration before applying the first and the second criteria.

Figure 4.8 shows the spark timing and corresponding torque profiles at iteration 5 after criteria 1 and 2 are applied. The figure indicates that the prediction accuracy is improved from the fourth iteration. The predicted spark timing in the fifth iteration produces a torque profile closer to the desired torque profile. This iterative process (Figure 4.2) continues until convergence is achieved if $\boldsymbol{-10\ Nm \leq \Delta T_{pred} \leq -10\ Nm}$.

Figure 4.9 shows good agreement between predicted and desired spark timing as well as between predicted and desired torque profiles. This was achieved at iteration 8 which is the last iteration. The metamodels were trained using 4 design sites after applying criteria 1 and 2. The maximum difference between the predicted and desired

torque profiles is $\Delta \boldsymbol{T_{pred}} = \boldsymbol{4.6\ Nm}$ at $\boldsymbol{t_{err}} = \boldsymbol{6\ s}$ which satisfies the convergence criterion.

4.3 Using NARX and Neural Networks to Create a Spark Timing and Cylinder Pressure Profiles Corresponding to a Desired Torque (Method 2)

As discussed in Section 2.2.6, neural networks provide a powerful metamodel which may be used to predict and interpolate in static (not time dependent) design space. Furthermore, the combination of NARX and neural network provides a framework that can be used to predict dynamic profiles (temporal design space). In contrast to the SVD-Kriging interpolation metamodeling technique, the NARX model may be used to extrapolate a dynamic system where no data was used to train the metamodel.

In this section, we utilize the NARX neural networks to predict the spark timing profile corresponding to the desired torque profile as discussed in Section 4.2. We also show how to predict the cylinder pressure profile corresponding to a desired torque. In this work, the cylinder pressure profile is predicted to mainly demonstrate the powerful capability of NARX neural network to predict and simulate a periodic, highly nonlinear dynamic profile with a small number of training profiles.

4.3.1 Spark Timing Prediction Using NARX Neural Networks

Figure 4.10 shows the flowchart of the overall process to determine the desired spark timing profile with acceptable accuracy using a NARX Neural Network (NARX-NN) metamodel.

In this study, 64 design sites are provided using a group-based space-filling DOE algorithm. Consequently, 64 synthetic spark timing profiles and their corresponding torque profiles are generated using GT-power. It is assumed that there are only 64

synthetic profiles available. Note that in the current approach, after generating the synthetic data, no simulation activity takes place. This is important since in many real case scenarios either there are no simulation tools available (only test data are available) or there is no correlated simulation model.

Subsequently, we verify that there is no design site in the training set which results in engine misfire. The final training set is thus provided by eliminating the design sites where misfiring takes place. The initial training set which is used to train the metamodel includes 8 training points resulting in 8 spark timing and their corresponding torque profiles.

To train the NARX-NN model, we must define the input and output and various parameters used in the training process. The torque sequence (time trajectory) is the input and the spark timing sequence is the output. For the NN in NARX, the number of hidden layers and epochs are 25 and 1000, respectively. The number of neurons per hidden layer is 10. For the NARX, we use a tapped delay line with 10 delays for both input and output. The number of delay signals, number of hidden layers, and number of epochs was determined by performing a sensitivity analysis to achieve an accurate prediction with a small number of training data. In this study, we ignored the validation check in order to use all available data for training. The training process continued until the maximum number of epochs was reached.

The training was performed using the open loop form (series-parallel network configuration). Figure 4.11 shows the neural network training summary using 8 training points. After the training was completed the model was converted to the closed loop form

to predict the spark timing corresponding to a desired torque profile which was not used for training.

Figure 4.12 shows the predicted spark timing (dotted line) against the actual spark timing (solid line) for only 8 training points. The difference between the actual and predicted spark timing is very small until $t = 40\ sec$ but the prediction accuracy reduces for $t > 40\ secs$. In order to increase the accuracy, we must add more training profiles to the initial training set. To keep the number of training profiles to a minimum, we add one training point to the previous training set until the model is converged.

The model is considered converged if the Averaged Relative Error (ARE) of predicted spark timings for two consecutive iterations is less than 1%

$$\frac{\Sigma_{t_0=1}^{t=90}\frac{\left|\mathrm{spk}_i^{(t)}-\mathrm{spk}_{i-1}^{(t)}\right|}{\mathrm{spk}_{i-1}^{(t)}}}{t} \leq 1\% \tag{4.6}$$

where, spk is the spark timing, i is the number of training points, and t is the time step. Figure 4.13 shows the prediction of spark timing against the actual spark timing for 9 training points. Figure 4.14 shows the ARE as the number of training points increases. As expected, the ARE decreases as the number of training points increases.

The model is converged using only 13 training points. Adding more training points does not improve the prediction accuracy. Figure 4.15 compares the predicted and actual spark timing profile at the 13th iteration. The difference between these two profiles is very small.

4.3.2 Cylinder Pressure Profile Prediction using NARX Neural Networks

In this section, we demonstrate the capability of NARX-NN to predict and simulate periodic and highly nonlinear dynamic profiles with a small number of training

profiles. For that we simulate the cylinder pressure profile corresponding to the desired torque profile discussed earlier in this chapter. The process is similar to the spark timing prediction. However, the output is the cylinder pressure profile (Figure 4.10).

The synthetic cylinder pressure profiles which are used to train the NARX-NN correspond to the training torque and spark timing profiles discussed in the previous section. These profiles are recorded from cylinder one at every 2ms from 5 to 6s. A small time-step is used so that we capture small variations of pressure in the cylinder within one second as the spark timing retards. The input to the NARX-NN is torque sequence (time trajectory) and the output is the cylinder pressure sequence. The number of hidden layers and epochs are 10 and 250, respectively. The number of neurons of each hidden layer is 10. A tapped delay line with 50 delays (equivalent to 100 ms) is used for both input and output. The goal is to predict the remaining sequences (450 sequences) of cylinder pressure profile. The selected training parameters are reasonable as the prediction performance (MSE) stabilizes at its minimum very quickly.

Figure 4.16 shows the neural network training log when 9 training points are used. As mentioned before, the model trained for 250 epochs. Figure 4.17 shows the MSE decrease with the number of epochs. Figure 4.18 compares the targeted and predicted training points (top plot) and the error between them (bottom plot). It is observed that the errors are very small for the entire time series.

Figures 4.19 and 4.20 compare the target and predicted cylinder pressure profiles and the error between them, respectively. It is observed that the NARX-NN predicted the target cylinder pressure profile with very good accuracy. The average error is less than 0.1 bar and the maximum absolute error is approximately 1 bar.

4.4 Summary

Two new iterative methods were developed to determine the spark timing profile for a spark-ignition engine to achieve a desired engine torque profile during a tip-out maneuver. The first method uses time-dependent metamodeling based on SVD and Kriging. The spark timing profiles are described by a third-order polynomial as a function of time using three unknown coefficients. A design of the three unknown coefficients is generated using a DOE which provides groups of design points with space-filling properties by themselves and in unions. Each design site defines a spark timing profile which is then used to generate an engine torque profile by running the GT–Power engine simulation software. To improve accuracy and efficiency (i.e. minimize the number of GT-Power runs), the design sites which result in engine misfire are removed from the training set. A spark-ignition engine example was used to demonstrate the proposed approach. The results showed good accuracy in determining the spark timing profile to produce a desired torque profile. This was achieved with only 32 runs (8 groups of 4 design sites each) of GT-Power.

The second method uses NARX Neural Networks to predict the spark timing profile and cylinder pressure profile corresponding to a desired torque profile. Sixty four design sites were provided using the group-based space-filling DOE algorithm and consequently 64 synthetic spark timing profiles and their corresponding torque profiles were generated using GT-power. The profiles with misfiring conditions were removed. A neural network was used with 25 hidden layers and 10 neurons in each hidden layer while the NARX model used 10 tapped delay lines. The sequential process of prediction started with 8 initial training profiles and the model converged using 13 training profiles.

Finally, for the prediction of a cylinder pressure profile, the NARX-NN model was setup using 50 tapped delay lines, 10 hidden layers with 10 neurons in each hidden layer and 250 epochs. The sequential process of prediction started by 8 initial training profiles and the model converged using 9 training profiles.

The time-dependent metamodels were able to predict the spark timing profile accurately. However, the NARX-NN model converged faster with a smaller number of training profiles and with a higher accuracy. It should also be noted that the NARX-NN model can be trained with test data if simulation tools are not available.

Table 4.1: Important engine parameters

Parameter name	Values and descriptions
Displacement	2.0 [L]
Number of cylinders	4
Bore / Stroke	86 / 86.07 [mm]
Compression ratio	9.5:1
Max. intake valve lift	10.2 [mm]
Max. exhaust valve lift	10.2 [mm]
Allowed ICL phasing range	440 to 500 degCA aTDCF
Allowed ECL phasing range	220 to 280 degCA aTDCF
Fuel injection type	Port Fuel Injection

Table 4.2: Range of spark timing profile coefficients in the DOE

Input parameter	d_1	d_2	d_3
Low limit	0	0	-100
Upper limit	100	100	100

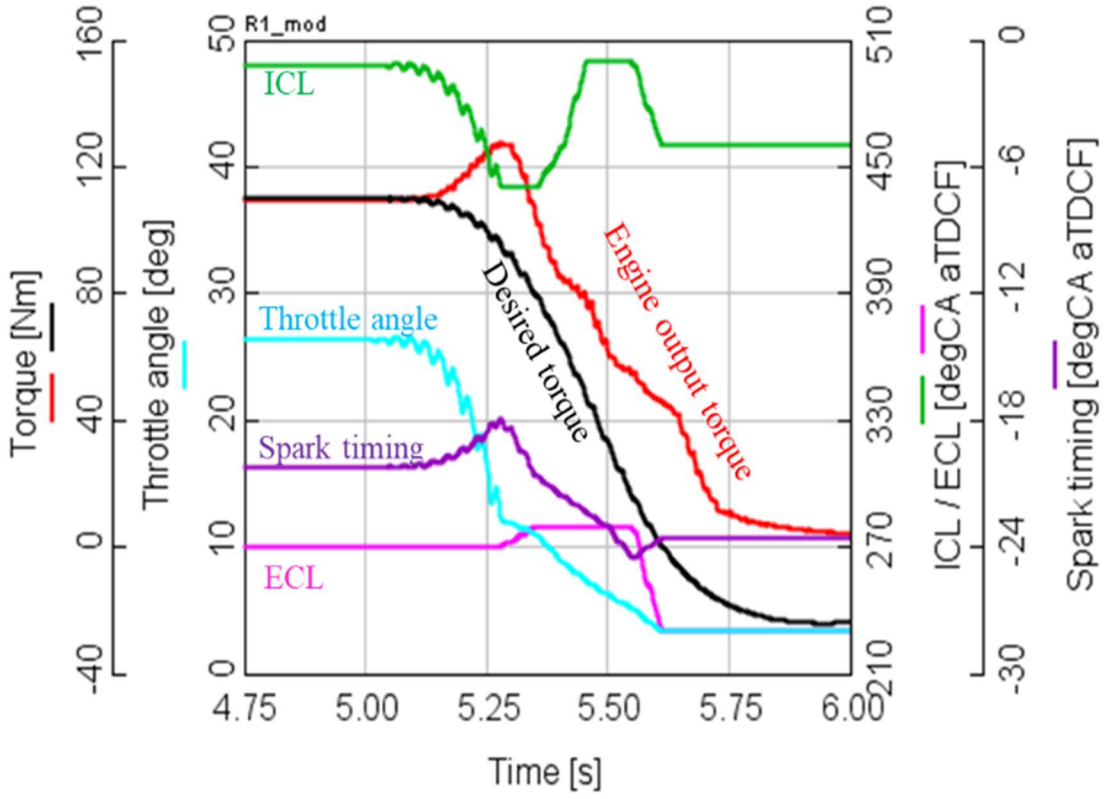

Figure 4.1: Example of transient tip-out maneuver at constant engine speed using fuel map input

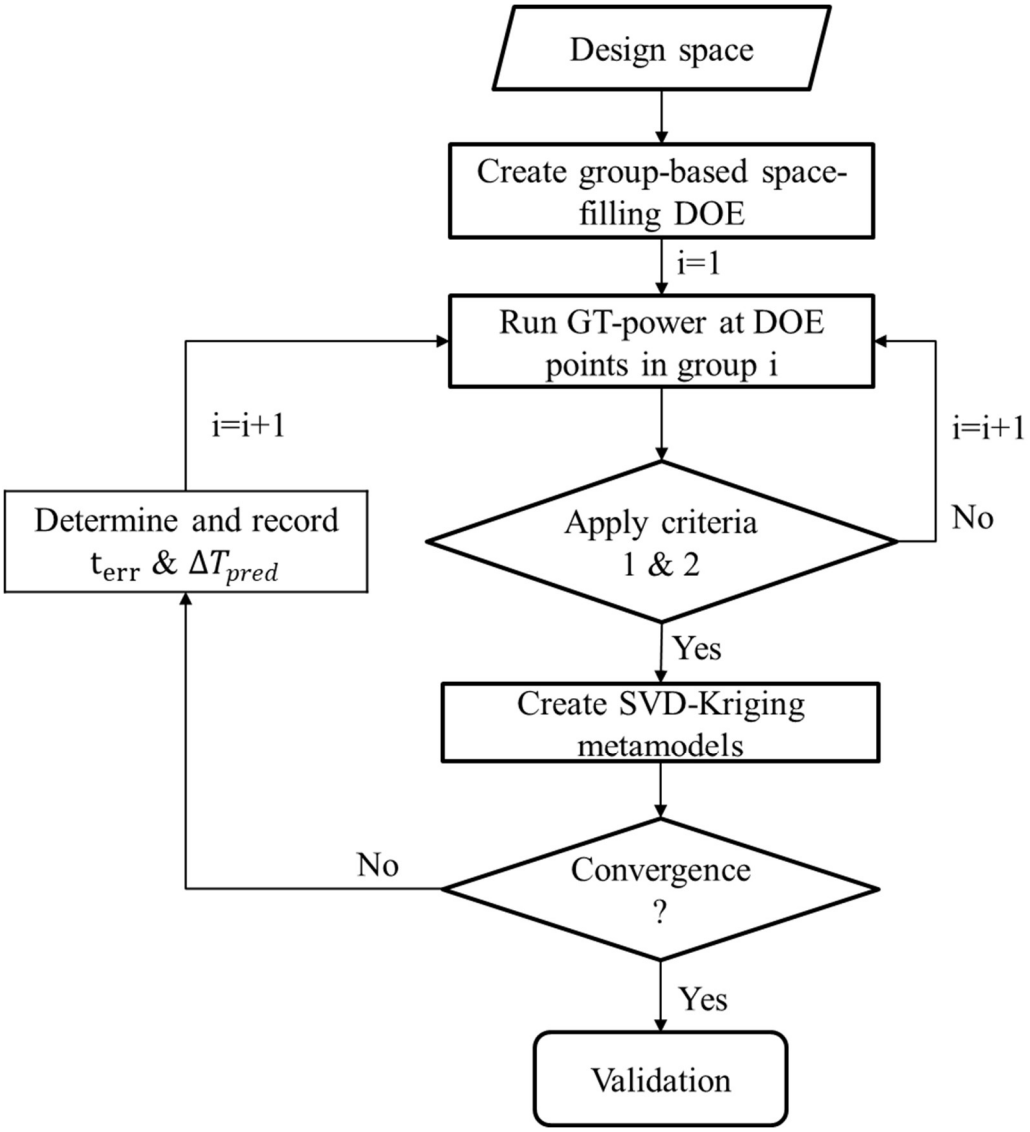

Figure 4.2: Flowchart of proposed methodology (method 1)

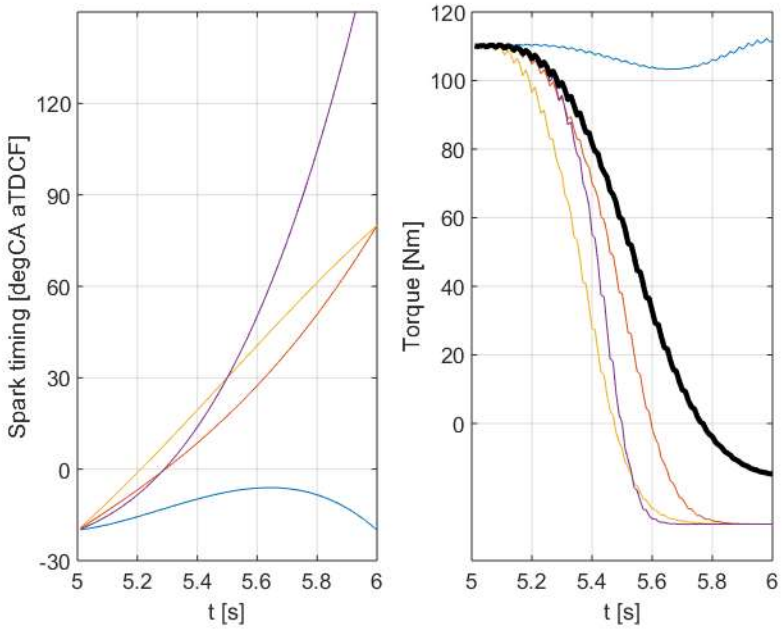

Figure 4.3: Spark timing profiles (left) and corresponding engine torque profiles (right) for first iteration

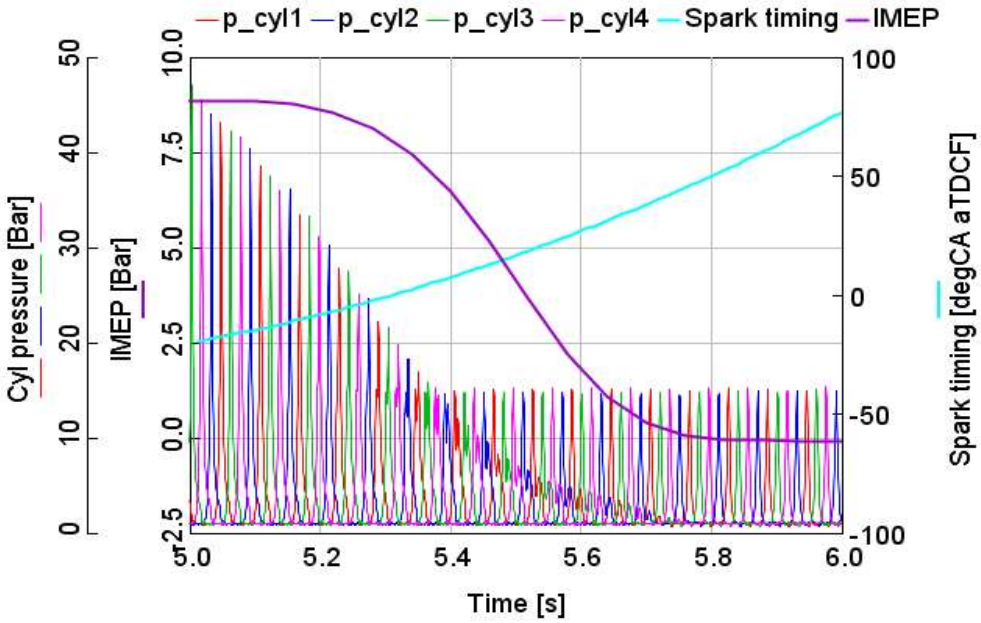

Figure 4.4: An example of engine transient maneuver with misfire

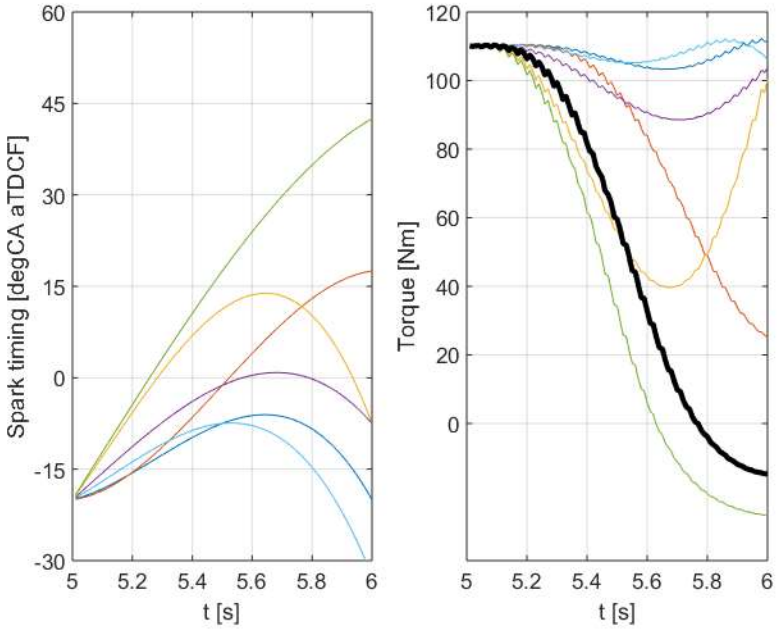

Figure 4.5: Spark timing and torque profiles at iteration four

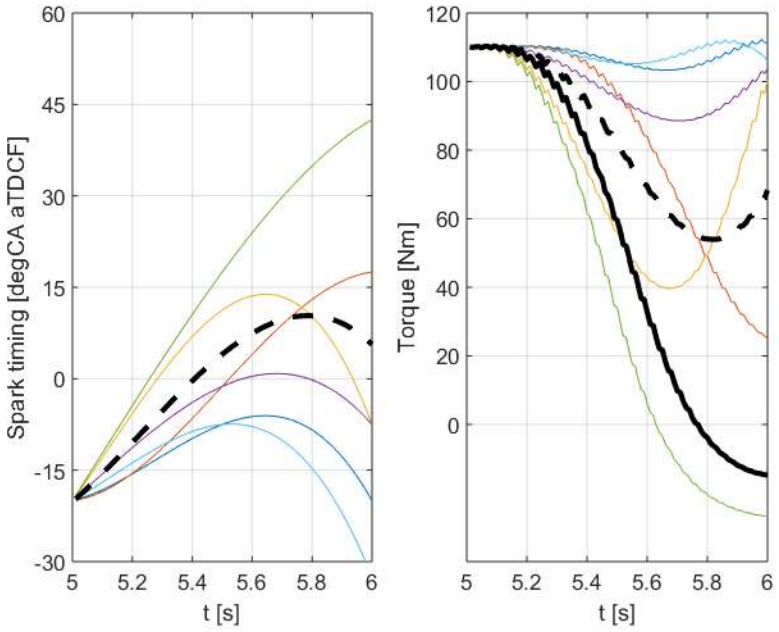

Figure 4.6: Spark timing prediction at fourth iteration

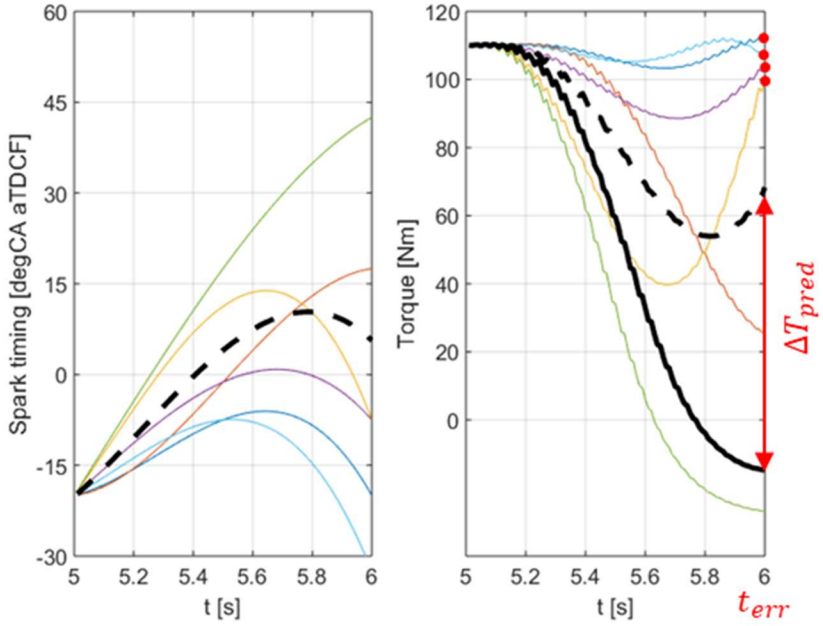

Figure 4.7: Calculation of $\boldsymbol{t_{err}}$ and $\boldsymbol{\Delta T_{pred}}$ at fourth iteration

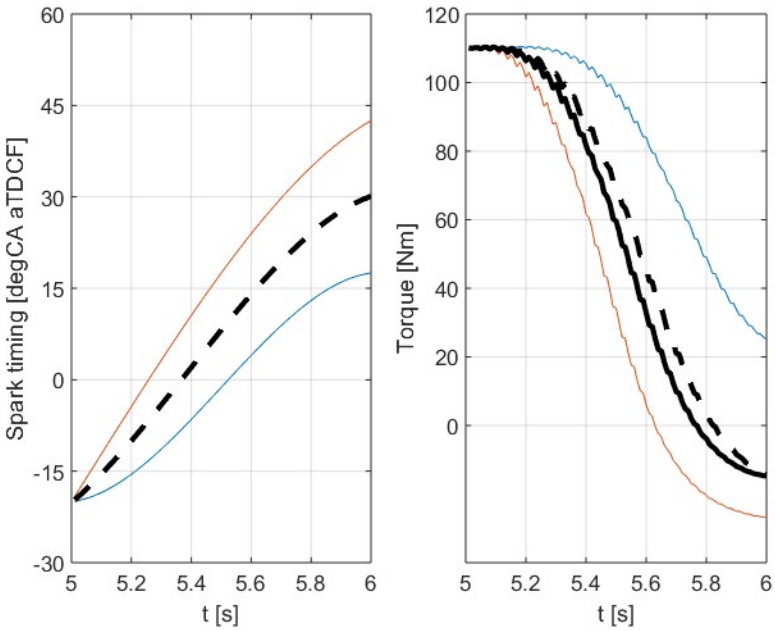

Figure 4.8: Spark timing prediction at fifth iteration

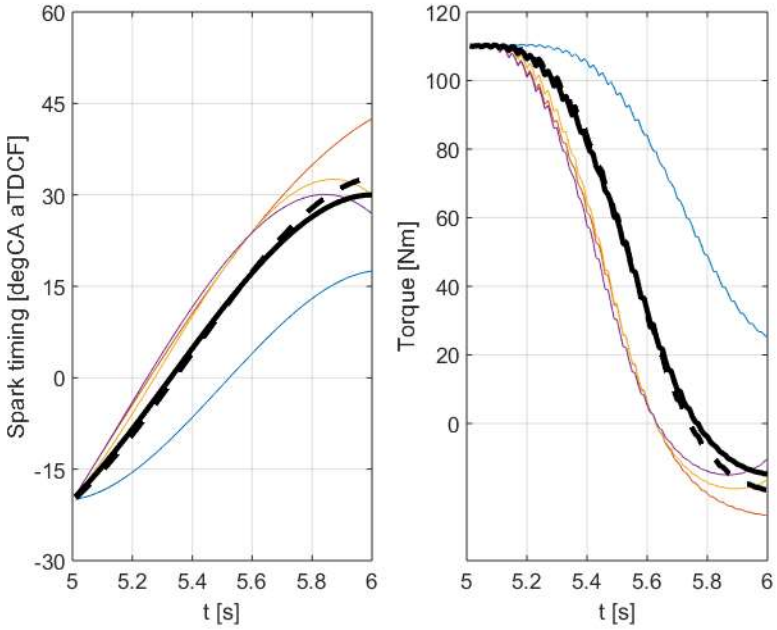

Figure 4.9: Comparison between desired and predicted spark timing at iteration 8

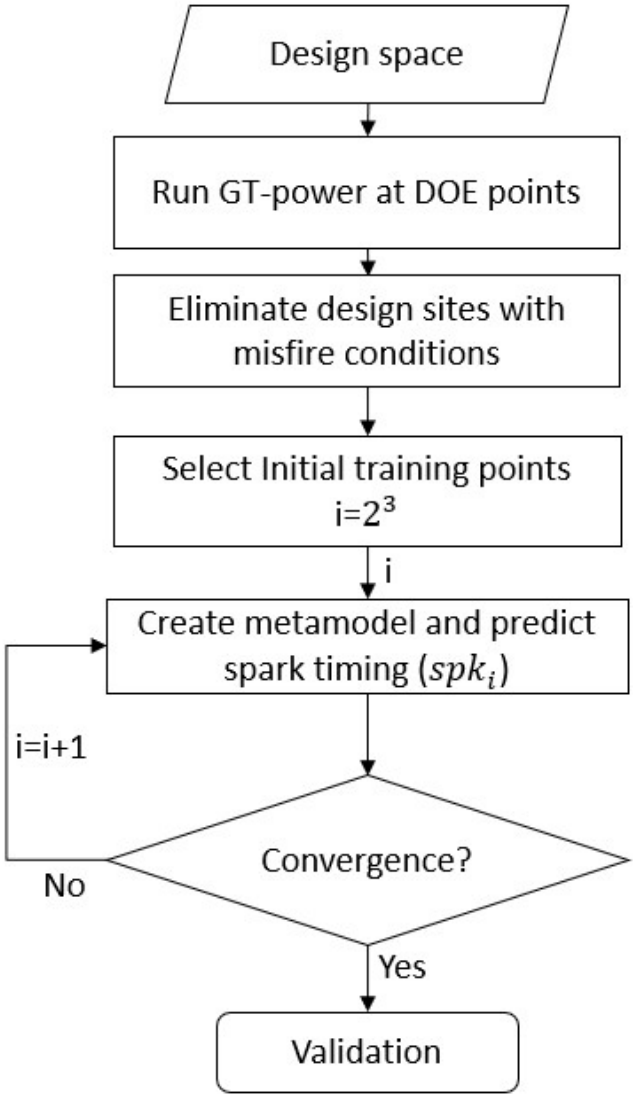

Figure 4.10: Flowchart of proposed methodology to generate spark timing profile using NARX (method 2)

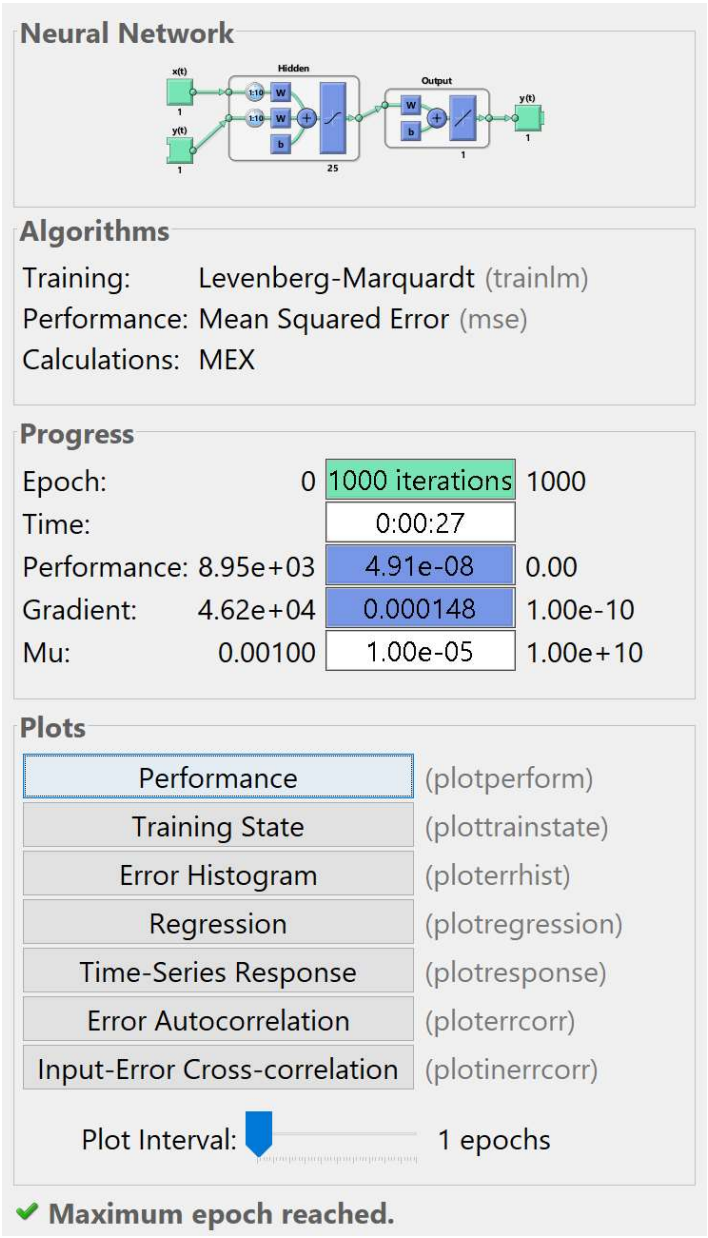

Figure 4.11: Neural network training summary using 8 training points

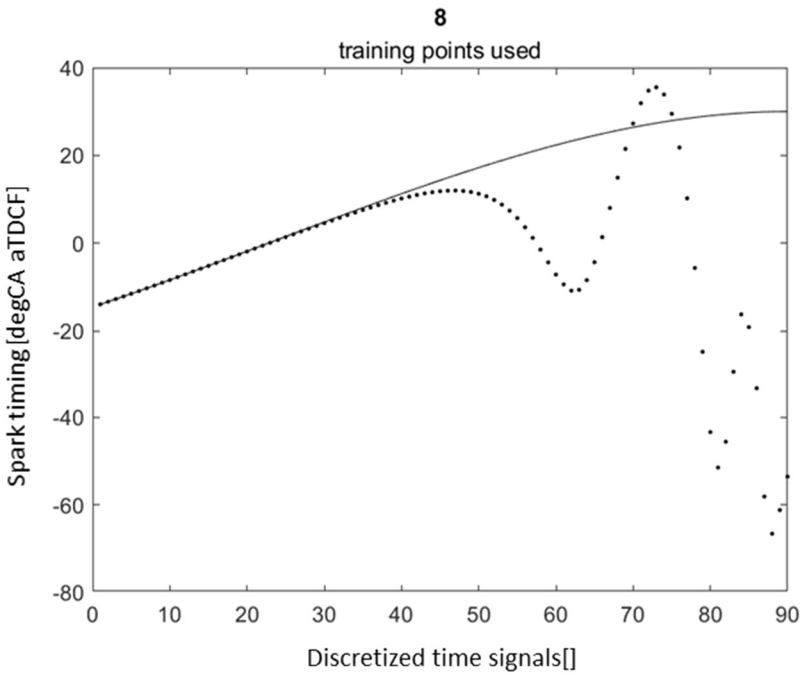

Figure 4.12: Predicted vs. actual spark timing using 8 training points

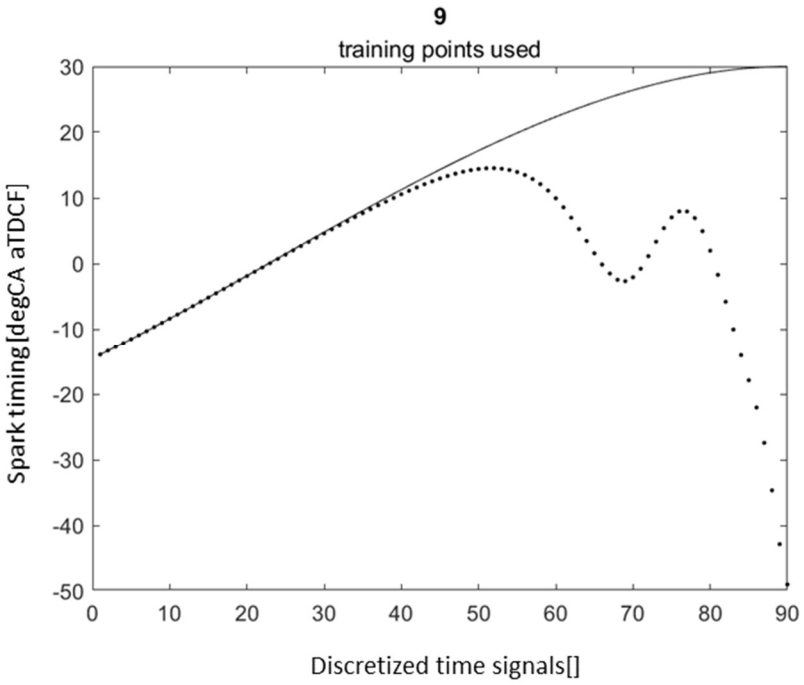

Figure 4.13: Predicted vs. actual spark timing using 9 training points

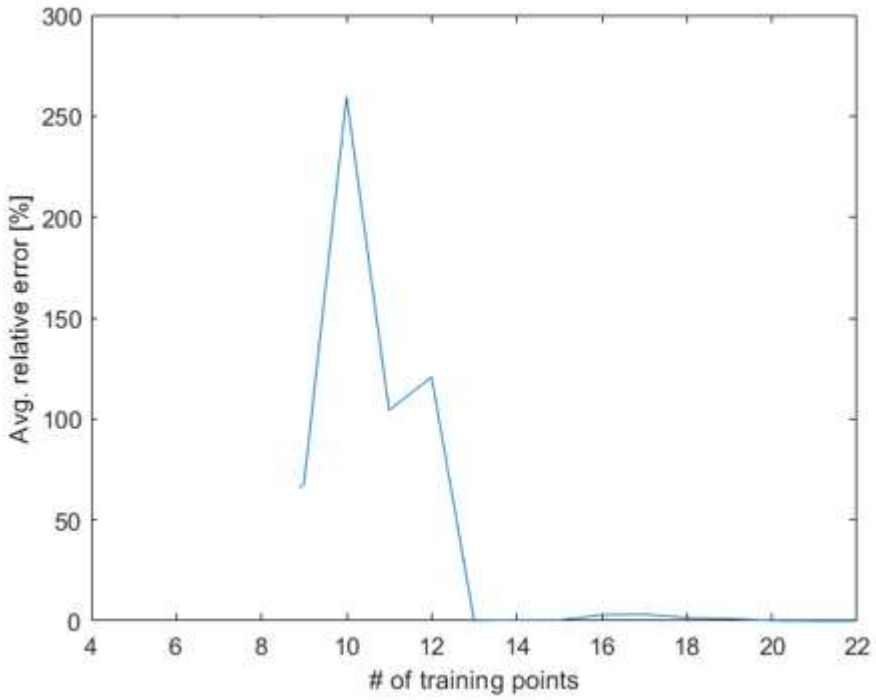

Figure 4.14: Average relative error for different number of training points

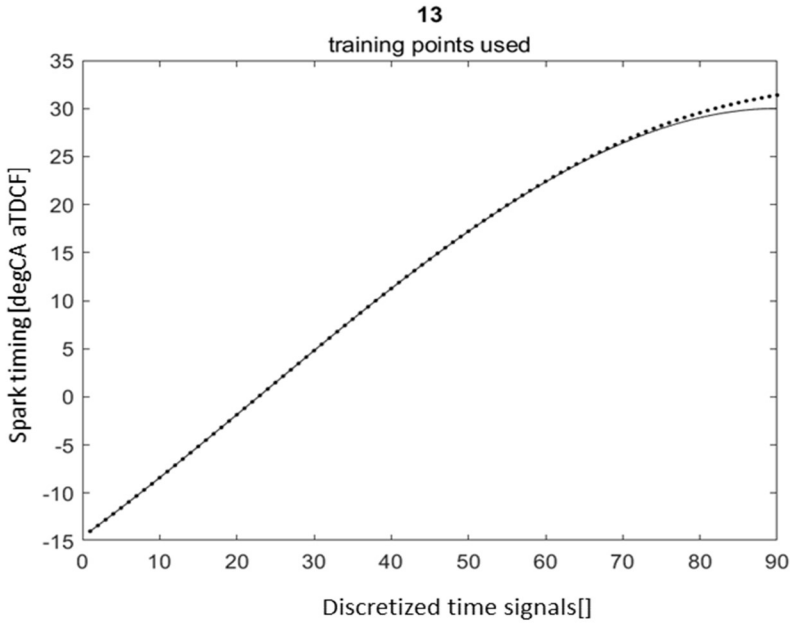

Figure 4.15: Comparison between actual and predicted spark timings for converged model at iteration 13

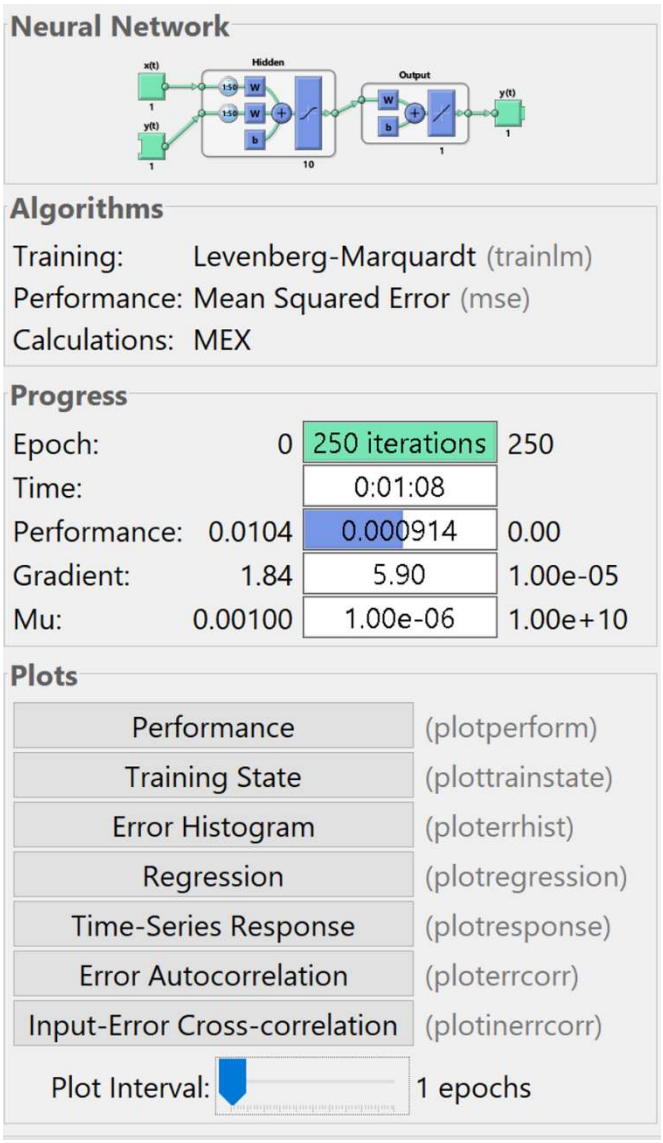

Figure 4.16: Neural network training log using 9 training points

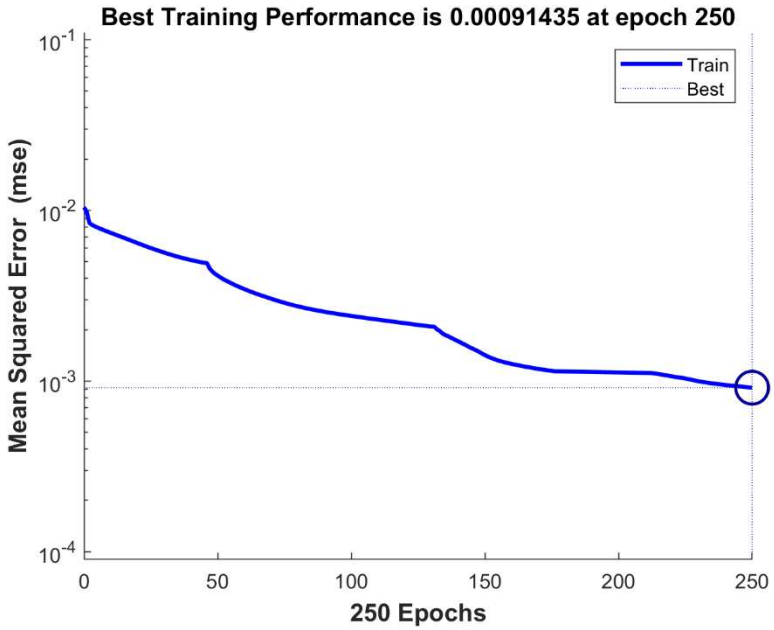

Figure 4.17: Training performance using 9 training points

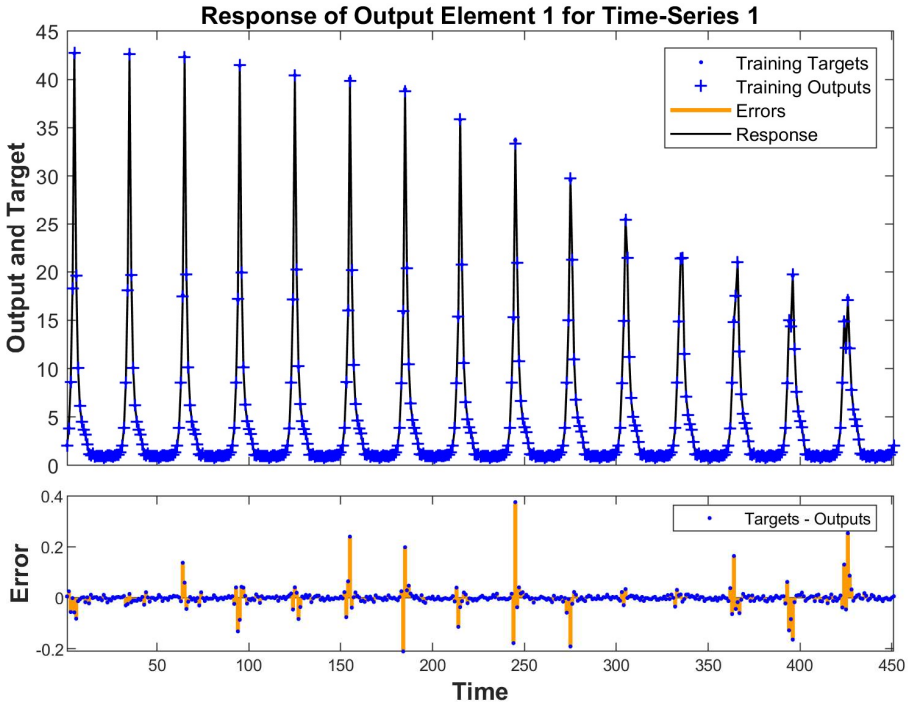

Figure 4.18: Response of output element 1 for time series 1

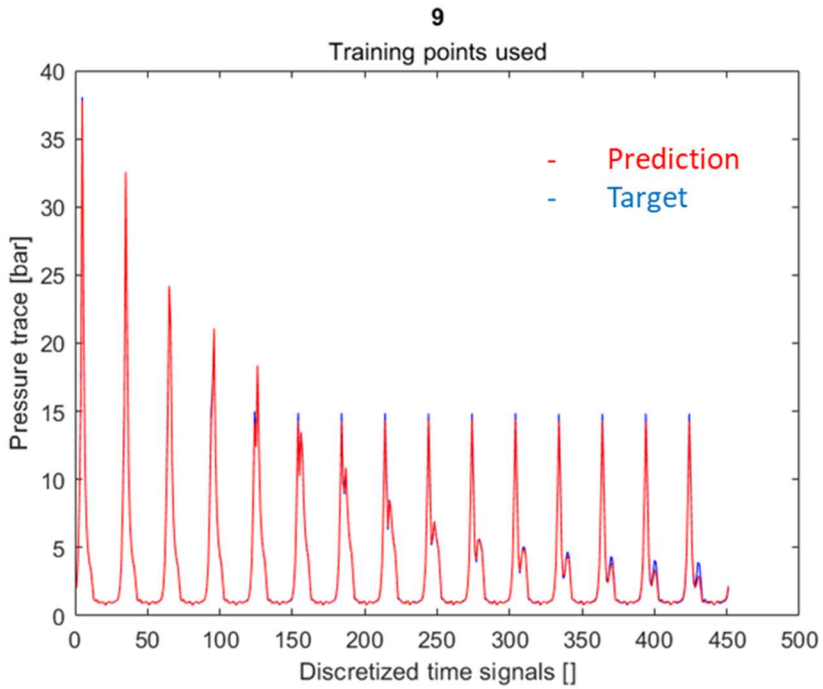

Figure 4.19: Comparison between target and predicted cylinder pressure profiles

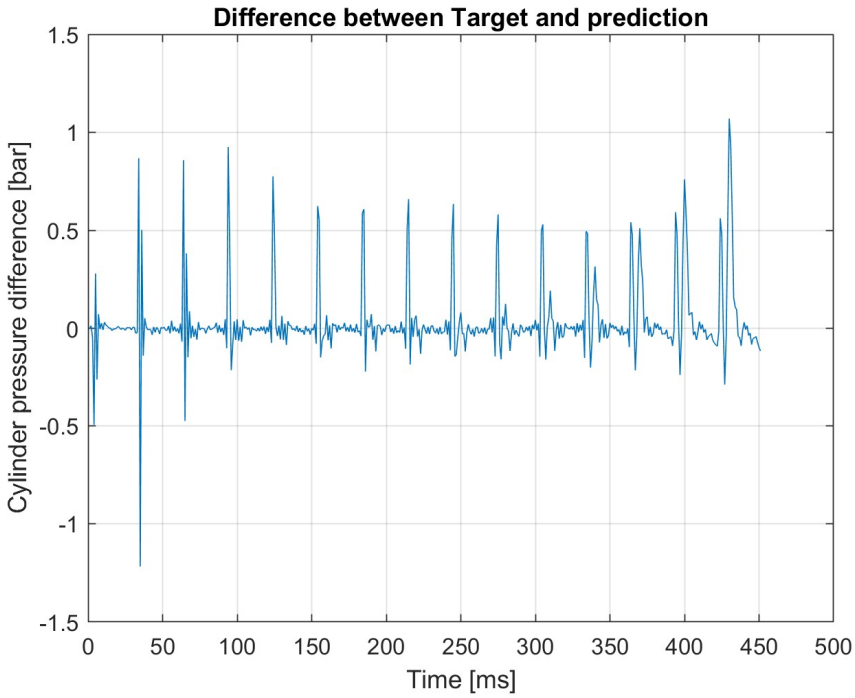

Figure 4.20: Difference between target and predicted cylinder pressure

CHAPTER FIVE

CONTRIBUTIONS AND FUTURE WORK

5.1 Dissertation Contributions

The contributions of this research relate to developing analytical capabilities using time-independent and time-dependent metamodeling to obtain first I.C. engine fuel maps and second the required spark timing profile to achieve a desired engine torque profile during a tip-out maneuver. The developed metamodels use only a small number of expensive engine simulation runs making them very attractive and useful in engine development. Below are specific contributions.

- A new process was introduced to generate an engine fuel map for full load and part load conditions with less computational effort while maintaining accuracy. The developed approach reduces the computational time to generate an I.C. engine fuel map by a factor of 14 compared to heuristically optimized conventional methods used by the industry. At full load conditions, a small number of DOE points (121 points) was enough to generate the wide-open throttle (WOT) curve. At part load conditions, only 110 DOE points were used to train the developed metamodels compared to almost 3000 points existing approaches require.
- Two new iterative methods were introduced to determine the spark timing profile for a spark-ignition engine to achieve a desired engine torque profile during a tip-out maneuver using an iterative sequence. Both methods use time-dependent metamodeling. The first method is based on SVD and Kriging. The developed SVD-Kriging metamodel converged with acceptable accuracy after only 32 simulation runs

(training profiles). In the second method the spark timing profile corresponding to a desired torque profile was predicted using only 13 training profiles.

- A NARX-NN metamodel was also developed to obtain the spark timing profile for a spark-ignition engine to achieve a desired engine torque profile during a tip-out maneuver. This approach outperformed the SVD-Kriging method in terms of accuracy and efficiency. Additionally, the NARX-NN is able to predict the highly nonlinear periodic cylinder pressure profile corresponding to a desired torque profile.

5.2 Future Work

- Apply the developed time-dependent metamodeling approaches to different engine configurations to further demonstrate their accuracy, efficiency and practical usefulness.
- Demonstrate the value of the developed time-dependent metamodeling approaches in engine calibration to avoid the current costly and time consuming trial and error methods.

REFERENCES

[1] Pischinger, S., 2016, "Internal Combustion Engines, Volume I, II," Lecture notes, Institute for Combustion Engines, RWTH Aachen University, Aachen, Germany.

[2] Kleijnen, J.P.C., 1987, *Statistical Tools for Simulation Practitioners*, Marcel Dekker.

[3] Fang, K.-T., Li, R. and A. Sudjianto, A., 2005, *Design and Modeling for Computer Experiments*, Chapman & Hall/CRC.

[4] Heywood, J., 1988, *Internal Combustion Engine Fundamentals*, McGraw-Hill.

[5] Bosch: Automotive Handbook, 1st English edition, Robert Bosch GmbH, 1976.

[6] Rogowski, A. R., 1953, *Elements of Internal Combustion Engines*," McGraw-Hill.

[7] Taylor, C. F., 1968, *The Internal Combustion Engine in Theory and Practice*, vol. 2, MIT Press, Cambridge, Mass.

[8] Takiuawa, M., Uno, T., Oue, T., and Yura, T., 1982, "A Study of Gas Exchange Process Simulation of an Automotive Multi-Cylinder Internal Combustion Engine," SAE paper 820410, SAE, vol. 91.

[9] Ohata, A., and Ishida, Y., 1982, "Dynamic Inlet Pressure and Volumetric Efficiency of Four Cycle Four Cylinder Engine," *SAE Trans.*, SAE paper 820407, vol. 91.

[10] Chapman, M., Novak, J. M., and Stein, R. A., 1982, "Numerical Modeling of Inlet and Exhaust nom in Multi-Cylinder Internal Combustion Engines," *Flows in Internal Combustion Engines -Winter Annual Meeting*, ASME, New York.

[11] Nakajima, Y., Sugihara, K. and Takagi, Y., 1979, "Lean Mixture or EGR-Which is Better for Fuel Economy and NO, Reduction," *Proceedings of Conference on Fuel Economy and Emissions of Lean Burn Engines*, Paper C94/79, Institution of Mechanical Engineers, London.

[12] Wade, W., and Jones, C., 1984, "Current and Future Light Duty Diesel Engines and Their Fuels," SAE paper 840105, SAE Trans., vol. 93.

[13] Lavoie, G. A., and Blumberg, P. N., 1980, "A Fundamental Model for Predicting Fuel Consumption, NO- and HC Emissions of a Conventional Spark-Ignited Engine," *Combust. Sci. Technol.*, vol. 21, pp. i25-258.

[14] Rao, C. R.,1973, *Linear Statistical Inference and its Applications*, Wiley, New York.

[15] Antoniadis, A. and Fan, J.,2001, "Regularization of Wavelets Approximations," *J. Amer. Statist. Assoc.* 96, 939–967.

[16] An, J. and Owen, A. B., 2001, "Quasi-regression," *J. Complexity* 17, 588–607.

[17] Dennis, J. and Schnabel, R., 1983, *Numerical Methods of Unconstrained Optimization and Nonlinear Equations*, Englewood Cliffs: Prentice-Hall.

[18] Youn, B. D. and Choi, K. K., 2004, "Selecting Probabilistic Approaches for Reliability Based Design Optimization," *AIAA Journal*, vol. 42, no. 1, pp. 124-131.

[19] Thoft-Christensen, P. and Baker, M., 1982, *Structural Reliability Theory and its Application*, Berlin Heidelberg, New York: Springer.

[20] Du, X., Sudjianto, A. and Chen, W., 2004, "An Integrated Framework for Optimization under Uncertainty using Inverse Reliability Strategy," *ASME Mech Design*, vol. 124, no. 4, pp. 562-570, 2004.

[21] Youn, B. D. and Choi, K. K., 2004, "Selecting Probabilistic Approaches for Reliability Based Design Optimization," *AIAA Journal*, vol. 42, no. 1, pp. 124-131.

[22] Kreyszig, E., 1983, *Advanced Engineering Mathematics*, New York, Wiley.

[23] Wahba, G., 1990, *Spline Models for Observational Data*, SIAM, Philadelphia.

[24] Stone, C. J., Hansen, M. H., Kooperberg, C. and Truong, Y. K., 1997, "Polynomial Splines and their Tensor Products in Extended Linear Modeling," *Ann. Statist.* 25, 1371–1470.

[25] Eilers, P. H. C. and Marx, B. D., 1996 "Flexible Smoothing with b-splines and Penalties," *Statist. Sci.* 11, 89–121.

[26] Ruppert, D. and Carroll, R. J., 2000, "Spatially-adaptive Penalties for Spline Fitting," *Austral. & New Zealand J. Statist.* 42, 205–223.

[27] Friedman, J. H., 1991, "Multivariate Adaptive Regression Splines," *The Annals of Statistics*, 19(1), 1-67.

[28] Powell, M. J. D., 1987, *Radial Basis Functions for Multivariable Interpolation: A Review - Algorithms for Approximation*, Clarendon Press, USA, 143–167.

[29] Moody, J. Darken, C. J., 1989, "Fast Learning in Networks of Locally-Tuned Processing Units," *Neural Computation*, 1, 281-294.

[30] Bishop, C. M., 1995, *Neural Networks for Pattern Recognition*, Oxford University Press.

[31] Putko, M., Newman, P., Taylor, A. I. and Green, L., 2002, "Approach for Uncertainty Propagation and Robust Design in CFD using Sensitivity Derivatives," *ASME Fluids Eng*, vol. 124, no. 1, pp. 60-69.

[32] Krige, D. G., 1951, "A Statistical Approach to Some Mine Valuation and Allied Problems on the Witwatersrand," M.Sc. Thesis in Engineering, University of Witwatersrand.

[33] Sacks, J., Welch, W.J., Mitchell, T.J. and Wynn, H.P., 1989. "Design and analysis of computer experiments," *Statistical science, 4*(4), pp.409-423.

[34] Cressie, Noel A. C., 1993, *Statistics for Spatial Data,* John Wiley & Sons, Inc.

[35] Currin, C., 1991, "Bayesian Prediction of Deterministic Functions with Applications to the Design and Analysis of Computer Experiments," *Journal of the American Statistical Association*, vol. 86, no. 416, pp. 953–963.

[36] Lophaven, S. N., Nielsen, H. B., and Søndergaard, J., 2002, "DACE-A Matlab Kriging toolbox, version 2.0."

[37] Goodfellow, I., Bengio, Y., Courville, A., 2016, *Deep Learning*, MIT Press.

[38] Hassoun, M. H., 1995, *Fundamentals of Artificial Neural Networks*, The MIT Press, Cambridge, Massachusetts.

[39] Bishop, C., 1995, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford.

[40] Haykin, S. S., 1998, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, Upper Saddle River, NJ.

[41] Hagan, M. T., Demuth, H. B. and Beale, M. H., 1996, *Neural Network Design*, PWS Publishing, Boston.

[42] McCulloch, W. S., Pitts, W. A., 1943, "Logical Calculus of the Ideas Immanent in Nervous Activity,' *Bulletin of Mathematical Biophysics* 5, 115–133.

[43] Rosenblatt, F., 1958, "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," *Psychological review*, 65(6), p.386.

[44] Widrow, B. and Hoff, M. E., 1960, "Adaptive Switching Circuits," Stanford Univ Ca Stanford Electronics Labs.

[45] Minsky, M. L. and Papert, S. A., 1988, *Perceptrons: Expanded Edition*, The MIT Press.

[46] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. and Jackel, L. D., 1989, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural computation*, 1(4), pp.541-551.

[47] Hochreiter, S. and Schmidhuber, J., 1997, "Long Short-term Memory," *Neural Computation,* 9(8), pp.1735-1780.

[48] Hinton, G. E. and Salakhutdinov, R. R., 2006, "Reducing the Dimensionality of Data with Neural Networks," *Science*, 313(5786), pp.504-507.

[49] Bengio, Y., Lamblin, P., Popovici, D. and Larochelle, H., 2007, "Greedy Layer-wise Training of Deep Networks," *In Advances in Neural Information Processing Systems,* pp. 153-160.

[50] Ranzato, M., Poultney, C., Chopra, S. and LeCun, Y., 2007 "Efficient Learning of Sparse Representations with an Energy-based Model," *Advances in Neural Information Processing Systems*, 19, p.1137.

[51] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G. and Petersen, S., 2015, "Human-level Control Through Deep Reinforcement Learning," *Nature*, 518(7540), pp.529-533.

[52] Finn, E. S., Shen, X., Scheinost, D., Rosenberg, M. D., Huang, J., Chun, M. M., Papademetris, X. and Constable, R. T., 2015, "Functional Connectome Fingerprinting: Identifying Individuals using Patterns of Brain Connectivity," *Nature Neuroscience*, 18(11), pp.1664-1671.

[53] Brunton, S. L., and Kutz, J. N., 2019, *Data-driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, Cambridge University Press.

[54] Rumelhart, D., Hinton, G. and Williams, R., 1986, "Learning Internal Representations by Error Propagation," in D. E. Rumelhart, J.L. Mcclelland, and the PDP Research Group, eds, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, MIT Press, Cambridge, MA.

[55] Chen, S., Billings, S. A., Grant, P. M., 1990, "Nonlinear System Identification using Neural Networks," *Int. .I. Conir*., vol. 51, no. 6, pp. 1191-1214.

[56] Su, H.-T., McAvoy, T. J., 1991, "Identification of Chemical Processes using Recurrent Networks," *Proc. Amer. Conrr. Conj*., vol. 3 pp. 2314-2319.

[57] Connor, J., Atlas, L. E., Martin, D. R., 1992, "Recurrent Networks and NARMA Modeling," *Advances in Neural Information Processing Systems 4, J. E. Moody*, S. J. Hanson, and R. P. Lippmann, Eds. San Mateo, CA: Morgan Kaufmann, pp. 301-308.

[58] Graši˘c, B., Mlakar, P., Božnar, M., 2006," Ozone Prediction based on Neural Networks and Gaussian Processes," *Nuovo Cimento Soc. Ital. Fis., C Geophys. Space Phys*. 29(6), 651–661.

[59] Grancharova, A., Kocijan, J., Krastev, A., Hristova, H., 2010," High-order Gaussian Process Models for Prediction of Ozone Concentration in the Air," *Proceedings of the 7th EUROSIM Congress on Modelling and Simulation*, EUROSIM'10.

[60] Grancharova, A., Nedialkov, D., Kocijan, J., Hristova, H., Krastev, A., 2009, "Application of Gaussian Processes to the Prediction of Ozone Concentration in the Air of Burgas," *Proceedings of International Conference on Automatics and Informatics*, pp. IV-17-IV-20.

[61] Pisoni, E., Farina, M., Carnevale, C., Piroddi, L., 2009," Forecasting Peak Air Pollution Levels using NARX models," *Eng. Appl. Artif. Intell*. 22, 593–602.

[62] Al-Alawi, S. M., Abdul-Wahab, S .A. Bakheit, C. S., 2008, "Combining Principal Component Regression and Artificial Neural-networks for More Accurate Predictions of Ground-level Ozone," *Environ. Modell. Softw.* 23, 396–403.

[63] Solaiman,T. A., Coulibaly, P. and Kanaroglou, P., 2008,"Ground-level Ozone Forecasting using Data-driven Methods," *Air Qual. Atmos. Health* 1, 179–193.

[64] Beale, M. H., Hagan, M.T. and Demuth, H. B., 2021, "Deep Learning Toolbox User's Guide," *The MathWorks, Inc*. 1 Apple Hill Drive Natick, MA 01760-2098.

[65] Geroulas, V., Mourelatos, Z. P., Tsianika, V., and Baseski, I., 2017, "Reliability Analysis of Nonlinear Vibratory Systems Under Non-Gaussian Loads," ASME. *J. Mech. Des.* February 2018; 140(2): 021404.

[66] Jacobs, T., Assanis, D. and Filipi, Z., 2003, "The Impact of Exhaust Gas Recirculation on Performance and Emissions of a Heavy-Duty Diesel Engine," *SAE World Congress*, Paper# 2003-01-1068.

[67] Schubiger, R., Bertola, A. and Boulouchos, K., 2001, "Influence of EGR on Combustion and Exhaust Emissions of Heavy Duty DI-Diesel Engines Equipped with Common-Rail Injection Systems," *SAE Fuels and Lubricants Meeting and Exposition*, Paper# 2001-01-3497.

[68] Gray, C., 1988, "A Review of Variable Engine Valve Timing," *SAE World Congress*, Paper# 880386.

[69] Bohac, S. and Assanis, D., 2004, "Effect of Exhaust Valve Timing on Gasoline Engine Performance and Hydrocarbon Emissions," *SAE World Congress*, Paper# 2004-01-3058.

[70] Lenz, U. and Schroeder, D., 1997, "Transient Air-Fuel Ratio Control Using Artificial Intelligence," *SAE International Congress and Exposition*, Paper# 970618.

[71] Fu, H., Chen, X., Mustafa, E., Trigui, N., Richardson, S. and Shiling, I., 2004, "Analytical Investigation of Cam Strategies for SI Engine Part Load Operation," *SAE World Congress*, Paper# 2004-01-0997.

[72] Wu, B., Filipi, Z., Assanis, D., Kramer, D., Ohl, G., Prucka, M. and DiValentin, E., 2004, "Using Artificial Neural Networks for Representing the Air Flow Rate through a 2.4 Liter VVT Engine," *SAE Powertrain and Fluid Systems Conference and Exhibition*, Paper# 2004-01-3054.

[73] Fang, K.-T., Ma, C. and Winker, P., 2002, "Centered L_2-Discrepancy of Random Sampling and Latin Hypercube Design and Construction Of Uniform Designs," *Mathematics of Computation*, 71: (275-296).

[74] McKay, M. D., Beckman, R. J. and Conover, W. J., 1979, "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, 21(2), 239-245.

[75] Owen, A.B., 1992, "Orthogonal Arrays for Computer Experiments, Integration and Visualization," *Statistica Sinica*, 2(2), p. 439-52.

[76] Johnson, M. E., Moore, L. M. and Ylvisaker, D., 1990, "Minimax and Maximin Distance Designs," *Journal of Statistical Planning and Inference*, 26(2), p. 131-48.

[77] Kennard, R. W. and Stone, L. A., 1969, "Computer Aided Design of Experiments," *Technometrics*, 11(1), p. 137-48.

[78] Gutmann, H.-M., 2001, "A Radial Basis Function Method for Global Optimization," *Journal of Global Optimization*, 19(3), p. 201-27.

[79] Booker, A., 1998, "Using Metamodels for Engineering Design," INFORMS Seattle Fall 1998 Meeting, Seattle, WA, INFORMS, October 25-28, 1998.

[80] Panagiotopoulos, D., Iqbal, O., Mourelatos, Z.P. and Papadimitriou, D., 2018, "Optimal Water Jacket Flow Distribution Using a New Group-Based Space-Filling Design of Experiments Algorithm," *SAE World Congress*, Paper# 2008-01-1017.

[81] Li, Q., Yang, D., Hu, L. and Sheng, X., 2014, "Optimization of Turbocharger Compressor Stages using DOE for Vehicle Engine Application," *SAE World Congress*, Paper# 2014-01-0406.

[82] Theodossiades, S., Gnanakumarr, M., and Rahnejat, H., 2005, "Root Cause Identification and Physics of Impact-induced Driveline Noise in Vehicular Powertrain Systems", Proceedings of the IMECHE Part D *Journal of Automobile Engineering*, Number D11, pp. 1303-1319.

[83] Govindswamy, K., Hueser, M., D'Anna, T., Diemer, P., and Roxin, C., 2003, "Study of Low-Frequency Driveline Clunk During Static Engagements", SAE Paper 2003-01-1480.

[84] Volinski, B., 1999, "Automatic Transaxle Lash Study for Park Disengagement Clunk", SAE Paper 1999-01-1765.

[85] Chae, C., Lee, Y., Won, K., and Kang, K., 2004, "Experimental and Analytical Approach for Identification of Driveline Clunk Source and Transfer Path", SAE Paper 2004-01-1231.

[86] Oh, W., and Singh, R., 2005, "Examination of Clunk Phenomena Using a Non-Linear Torsional Model of a Front-Wheel-Drive Vehicle with Manual Transmission", SAE Paper 2005-01-2291.

[87] Schumacher, T., Biermann, J. W., Jansz, N., Willey, J., and Küpper, K., 2003, "Load Change Reactions of Passenger Cars: Method of Investigation and Improvement", *Proceedings of the IMECHE Part K Journal of Multi-body Dynamics*, Vol. 217, Number 4, pp. 283-291.

[88] Krentz, R. A., 1985 "Vehicle Response to Throttle Tip-In/Tip Out", SAE Paper 850967.

[89] Crowther, A. R., Zhang, N., and Singh, R., 2005 "Development of a Clunk Simulation Model for a Rear-Wheel-Drive Vehicle with Automatic Transmission," SAE Paper 2005-01- 2292.

[90] Khan, M. A. Z., 2011, "Transient Engine Model for Calibration Using Two-stage Regression Approach", PhD. Dissertation, Loughborough University.

[91] Tafreshi, A. and Mourelatos, Z. P., 2020, "Prediction of Fuel Maps in Variable Valve Timing Spark Ignited Gasoline Engines Using Kriging Metamodels," *SAE Int. J. Adv. & Curr. Prac. in Mobility*, 2(4):1999-2010.

[92] Atkinson, C. M., Long, T. W., and Hanze-vack, E. L., 1998," Virtual Sensing: A Neural Network-based Intelligent Performance and Emissions Prediction System for On-board Diagnostics and Engine Control", *Progress in Technology*, 73:301-314.

[93] Atkinson, C., Allain, M., and Zhang, H., 2008, "Using Model-Based Rapid Transient Calibration to Reduce Fuel Consumption and Emissions in Diesel Engines," SAE Technical Paper 2008-01-1365.

[94] Atkinson, C. M., Allain, M., Kalish, Y., and Zhang, H., 2009, "Model-Based Control of Diesel Engines for Fuel Efficiency Optimization", SAE Technical Paper Series No. 2009-01-0727. 3, 116, 146, 160.

[95] Lee, T., 2009, "Optimal Calibration and Transient Control of High Degree of Freedom Internal Combustion Engines" PhD. Dissertation, University of Michigan.

[96] Panagiotopoulos, D., Iqbal, O., Mourelatos, Z. P. and Papadimitriou, D., 2018, "Optimal Water Jacket Flow Distribution Using a New Group-Based Space-Filling Design of Experiments Algorithm," *SAE World Congress*, Paper# 2008-01-1017.

[97] Obert, E. F., 1973, *Internal Combustion Engines and Air Pollution*, Chap. 2, Intext Educational Publishers, New York.

[98] SAE Standard: *Engine Test Code-Spark Ignition and Diesel*, SAE J816b, SAE Handbook.

RELATED PUBLICATIONS

- Tafreshi, A. and Mourelatos, Z. P., 2020, "Prediction of Fuel Maps in Variable Valve Timing Spark Ignited Gasoline Engines Using Kriging Metamodels," *SAE Int. J. Adv. & Curr. Prac. in Mobility* 2(4):1999-2010.

- Tafreshi, A. and Mourelatos, Z. P., 2021, "Prediction of Spark Timing to Achieve a Specified Torque Profile in Spark-Ignition Engines Using Time-Dependent Metamodeling," *SAE International Journal of Advances and Current Practices in Mobility-V130-99EJ:* 2021-01-0238.