MODEL ASSISTED ITERATIVE CALIBRATION OF INTERNAL COMBUSTION ENGINES

By

Anuj Pal

A DISSERTATION

Submitted to Michigan State University in partial fulfillment of the requirements for the degree of

Mechanical Engineering – Doctor of Philosophy

2021

ABSTRACT

MODEL ASSISTED ITERATIVE CALIBRATION OF INTERNAL COMBUSTION ENGINES

By

Anuj Pal

Recent automotive technological advancements mainly focus on improving fuel economy with satisfactory emission levels, leading to a significant increment of engine system complexity, especially diesel engines. This increases the number of engine control parameters, making the engine calibration process challenging and time-consuming using the conventional map-based approach. Note that engine calibration is a crucial step in achieving optimal engine performance with satisfactory emissions, and it is an expensive process in general. With the advancement and widespread adoption of machine learning methods for control applications, it is now possible to use a black-box model with intelligence to efficiently calibrate nonlinear systems without detailed knowledge of system dynamics. The surrogate-assisted optimization approach is an attractive way to reduce the total computational budget for obtaining optimal solutions. This makes it special for its application to practical optimization problems requiring a large number of expensive evaluations.

The current research work focuses on the problem of performing engine calibration using the surrogate-assisted optimization approach. The objective is to find the trade-off curve between engine efficiency in terms of brake specific fuel consumption (BSFC) and its NO_x emissions by efficiently optimizing various control parameters. The complete study is divided into three parts. The first part deals with modifying the original algorithm for efficiently handling the practical system with measurement noise. A new constrained handling algorithm is proposed for lower confidence bound (LCB) criteria that showed good performance for both deterministic and stochastic systems. Furthermore, two extensions based on the expected improvement (EI) criterion are proposed for handling stochastic multi-objective problems.

After the methodology development for handling stochastic systems, the second part validates their efficacy for performing the engine calibration in a simulation setting. All three algorithms are compared to identify the best approach for its implementation on the actual engine experimental setup. Three control parameters, namely variable geometry turbocharger (VGT) vane position, exhaust-gas-recirculating (EGR) valve position, and the start of injection (SOI), are calibrated to obtain the trade-off between engine fuel efficiency performance (BSFC) and NO_x emissions within the constrained design space. The simulation study identifies the lower confidence bound (LCB) criteria with the proposed constraint handling approach to work well in the stochastic setting, compared with the other two extensions. Therefore, this approach is used for the experimental evaluation of the proposed surrogate-assisted optimization for engine calibration.

Finally, the third part is the experimental validation. It is the first step towards automating the entire engine calibration process. Experimental evaluations are performed on a 6.7L Ford diesel engine to validate the algorithm's efficacy. Problems with different complexity are formulated and evaluated using the proposed approach. Initially, a simpler problem with two control variables is formulated to get the confidence to perform the experiments using the proposed algorithm. Two variables: EGR valve position and VGT vane positions, are calibrated to obtain a trade-off between engine efficiency (BSFC) and NO_x emissions. After observing promising results, the study is concluded with a more complicated three control variable problem. An external electrically assisted boosting device (eBoost) is added to the engine system to perform calibration. Results showed improved engine performance using the eBoost with a significant reduction in calibration effort in terms of the number of experimental evaluations. The study successfully demonstrated the application of the surrogate-assisted optimization approach to a practical engine system and opened the door to automate the engine calibration process with reduced calibration efforts.

Copyright by ANUJ PAL 2021 This work is dedicated to my parents and my brother for their constant love and support throughout this journey

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank all the people who have encouraged, motivated, and supported me throughout my Ph.D. journey. It has been a journey filled with lots of learning and adventures.

Firstly, I would like to express my gratitude to my advisor, Professor Guoming (George) Zhu, for his inspiring guidance and encouragement that helped me in shaping the research and culminating in this Ph.D. thesis. I am grateful to him for providing me the freedom to work on this topic. Apart from the research work, he has been a great mentor and motivator. It is my good fortune to work under his guidance. I would also like to thank the committee members, Professor Kalyanmoy Deb, Professor Ranjan Mukherjee, and Professor Zhaojian Li, for their valuable feedback in the research work that has helped improve the overall quality of this work.

I also want to acknowledge the financial support from Ford Motor Company. A special thanks to Dr. Yan Wang and Dr. Ling Zhu for providing support throughout the project. I benefited a lot from their technical knowledge and their overall experience of looking at things from a different perspective. I would also like to thank Dr. Yifan Men for his help and valuable discussion about the experimental setup. I thank my fellow labmates for their support and all the fun we have had in the last few years. I would especially like to thank Kevin Moran and Tom Stuecken for helping in setting up the experimental test bench during the pandemic.

I am also grateful to all my friends who have made my life at MSU enjoyable and memorable. There have been times when failures have disheartened me, but it was these friends who always managed to bring back a smile to my face and kept me going.

Finally, I would like to thank my father and mother for their sacrifices, guidance, and immense love and support, without which I could not have achieved anything. I shall remain forever indebted to them. I am also thankful to my brother for his constant support and encouragement.

TABLE OF CONTENTS

| LIST OF | TABLES | х |
|---------|---|-----------------|
| LIST OF | FFIGURES | xi |
| KEY TO | OABBREVIATIONS | xvi |
| CHAPT | ER 1 INTRODUCTION | 1 |
| 1.1 | Motivation | 1 |
| 1.2 | Existing Approaches for Engine Calibration | 2 |
| | 1.2.1 Full Factorial DOE Method for Engine Calibration | 3 |
| | 1.2.2 Model-Based Approach for Control Calibration | 3 |
| | 1.2.3 Model-Free Approach for Control Calibration | 6 |
| 1.3 | Research Overview | 8 |
| | 1.3.1 Selecting the Algorithm for Experimental Evaluation | 8 |
| | 1.3.2 Stochastic Extensions of Proposed Algorithm for Practical Systems | 9 |
| | 1.3.3 Algorithm Implementation to Engine Dynamometer Setup | 10 |
| 1.4 | Research Contributions | 12 |
| 1.5 | Dissertation outline | 13 |
| CHAPT | ER 2 SURROGATE ASSISTED OPTIMIZATION FOR PRACTICAL SYSTEMS | 15 |
| 2.1 | Problem Formulation | 15 |
| 2.1 | 2.1.1 Constrained Multi-Objective Optimization Problem | 16 |
| 2.2 | Overall Algorithm | 17 |
| 2.3 | Model Development | 19 |
| 2.0 | 2.3.1 Kriging Model | 20 |
| | 2.3.2 Stochastic Kriging Model | $\frac{20}{22}$ |
| | 2.3.3 Deterministic vs. Stochastic Kriging Models | 24 |
| 24 | Acquisition function | 25 |
| 2 | 2.4.1 Expected Improvement | 26 |
| | 2.4.2 Expected Improvement for multi-objective problem | 27 |
| | 2.4.2.1 Using Achievement Scalarization Function | $\frac{-1}{28}$ |
| | 2.4.2.2 Using Multi-Objective Improvement Function | 28 |
| | 2.4.3 Constraint handling in Expected Improvement | 30 |
| | 2.4.4 Expected Improvement in Stochastic environment | 31 |
| | 2.4.5 Multi-objective extensions for stochastic environment | 33 |
| | 2451 Constrained Augmented Expected Improvement (CAEI) | 33 |
| | 2.4.5.2 Augmented Expected Improvement Matrix (AEIM) | 34 |
| | 24.6 Lower Confidence Bound | 35 |
| | 2.4.6.1 Proposed Constraint handling logic for LCB | 36 |
| | 247 Expected Improvement (EI) vs Lower Confidence Bound (LCB) | 39 |
| 2.5 | Optimization and selection of points for further evaluations | 44 |
| | | |

| | 2.5.1 Genetic Algorithm | 4 |
|-------|--|-----------------------|
| | 2.5.2 K-Means Clustering | 15 |
| | 2.5.3 DBSCAN | 6 |
| 2.6 | Chapter Summary | 6 |
| | | |
| CHAPT | ER 3 SIMULATION VALIDATION OF PROPOSED ALGORITHMS 4 | -8 |
| 3.1 | Problems for Validating the Algorithms | 8 |
| | 3.1.1 Numerical Test Problems | 9 |
| | 3.1.2 Engine Calibration Problem | 50 |
| | 3.1.2.1 Model Validation and its Implementation | 52 |
| | 3.1.2.2 Baseline Engine Calibration Result using GT -SUITE TM Model . 5 | 54 |
| 3.2 | Validation of Constraint Handling Algorithm | 56 |
| | 3.2.1 Validation on numerical problems | 56 |
| | 3.2.1.1 Unconstrained Problems | 57 |
| | 3.2.1.2 Constrained Problems | 59 |
| | 3.2.2 Validation on Engine Calibration Problem | 50 |
| 3.3 | Validation of Stochastic Multi-Objective Extensions | 53 |
| | 3.3.1 Heteroscedastic noise addition to the simulation | 54 |
| | 3.3.2 Validation on Numerical Problems | 56 |
| | 3 3 2 1 Unconstrained Test Problems | 57 |
| | 3 3 2 2 Constrained Test Problems 7 | 12 |
| | 3.3.3. Validation on Engine Calibration Problems | 75 |
| 34 | Chapter Summary | 20 |
| 5.7 | | ,0 |
| CHAPT | R 4 INITIAL ENGINE CALIBRATION RESULTS ON EXPERIMENTAL | |
| | SETUP | 32 |
| 4.1 | Overall Experimental Setup | 32 |
| | 4.1.1 Systems Involved in the Experimental Setup | 35 |
| | 4.1.2 INCA-MATLAB communication | 37 |
| 4.2 | Performing Experiments | 38 |
| | 4.2.1 Problem Formulation | 39 |
| | 4.2.2 Finding out the Feasible Operationol Range in Design Space 9 | 90 |
| | 4.2.3 Automated Engine Operation | 90 |
| 4.3 | Two - Variable Calibration Results |)2 |
| | 4.3.1 Results from Deterministic Approach | 94 |
| | 4.3.2 Results from Stochastic Approach | 98 |
| | 4.3.3 Results at Other Operating Conditions 10 |)1 |
| | 4 3 3 1 Operating Point: 1400 RPM - 500 Nm 10 |)1 |
| | 4.3.3.2 Operating Point: 1400 RPM - 175 Nm 10 |)2 |
| | 4 3 3 3 Operating Point: 1800 RPM - 400 Nm 10 |)3 |
| | 4 3 3 4 Operating Point: 1000 RPM - 325 Nm 10 |)4 |
| | 4.3.4 Problem Formulation with Increased FGR Levels 10 | ,- -)6 |
| | 4.3.5 Results for Revised Problem 10 | טי דו |
| ΛΛ | $T_{10,0} = \text{Results for Revised From interval} + 10000 \text{ mm} + 100000 \text{ mm} + 100000 \text{ mm} + 100000 \text{ mm} + 1000000 \text{ mm} + 10000000 \text{ mm} + 100000000000000000000000000000000000$ | , /)8 |
| 4.4 | Chapter Summary | 10 |

| CHAPTE | ER 5 | CONTROL CALIBRATION FOR THE SYSTEM WITH EBOOST | • | • | ••• | . 111 |
|--|--------|--|------|----|-----|-------|
| 5.1 | Proble | m Formulation | | • | | . 113 |
| | 5.1.1 | Defining Parameter Ranges | | • | | . 113 |
| 5.2 | Perfor | ming eBoost Experiments | | • | | . 117 |
| 5.3 | Exper | imental Results with eBoost | | • | | . 119 |
| | 5.3.1 | Baseline Result | | • | | . 119 |
| | 5.3.2 | Results using Proposed Approach | | • | | . 120 |
| 5.4 | Comp | arison of Experimental Evaluations | | • | | . 128 |
| 5.5 | Chapte | er Summary | • | • | | . 129 |
| CHAPTE | ER 6 | CONCLUSIONS AND FUTURE WORKS | • | • | | . 131 |
| 6.1 | Conclu | usions | | • | | . 131 |
| 6.2 | Future | Work | • | • | | . 134 |
| APPENDICES | | | | | | |
| | | A STOCHASTIC RESULTS ON NUMERICAL TEST PROBLE | 1IV. | 15 | • | . 13/ |
| APPENDIA D STOCHASTIC RESULTS FOR ENGINE CALIBRATION 150 | | | | | | |
| BIBLIO | GRAPI | НҮ | • | • | ••• | . 158 |

LIST OF TABLES

| Table 3.1: | Numerical test problems | 49 |
|------------|--|-----|
| Table 3.2: | IGD values for both unconstrained and constrained test problems | 58 |
| Table 3.3: | IGD values for LCB, EIM, and M3-1 | 60 |
| Table 3.4: | IGD values for all test problems using algorithms #1 : <i>CAEI</i> , #2 : <i>AEIM</i> and #3 : <i>CLCB</i> | 70 |
| Table 3.5: | IGD values for engine calibration problem | 79 |
| Table 4.1: | Diesel engine specifications | 85 |
| Table 4.2: | Different iterations using DBSCAN for Deterministic and Stochastic | 94 |
| Table A.1: | Numerical test problems for stochastic evaluations | 137 |

LIST OF FIGURES

| Figure 1.1: | Extraction of Speed-Load points from the drive cycle | 2 |
|-------------|--|----|
| Figure 1.2: | Control calibration strategy with physics-based model | 4 |
| Figure 1.3: | Control calibration strategy with data-driven model | 5 |
| Figure 1.4: | Structure for model-free control calibration strategy | 7 |
| Figure 1.5: | Comparison of experimental evaluations between existing and proposed approaches | 12 |
| Figure 2.1: | Overall architecture of the surrogate-assisted optimization process | 18 |
| Figure 2.2: | Model fitting using deterministic Kriging model for a function without mea- surement noise | 24 |
| Figure 2.3: | Model fitting using both deterministic and stochastic Kriging models for a function with measurement noise | 25 |
| Figure 2.4: | Explaination of constraint handling logic using test problem | 37 |
| Figure 2.5: | Identification of different region for \mathbf{P}_c | 38 |
| Figure 2.6: | 1^{st} Iteration with Expected Improvement acquisition function | 41 |
| Figure 2.7: | 2^{nd} Iteration with Expected Improvement acquisition function | 42 |
| Figure 2.8: | 1^{st} Iteration with Lower Confidence Bound acquisition function | 43 |
| Figure 2.9: | 2^{nd} Iteration with Lower Confidence Bound acquisition function | 43 |
| Figure 3.1: | GT-Power and emission model validation using experimental data | 53 |
| Figure 3.2: | Algorithm implementation for engine calibration | 53 |
| Figure 3.3: | Trade-off curve between BSFC and NO_x from the baseline study | 54 |
| Figure 3.4: | Optimal control parameter values from the baseline study | 55 |
| Figure 3.5: | Results of the lower confidence bound (LCB) based optimization algorithm on unconstrained test problems | 58 |

| Figure 3.6: | Results of the lower confidence bound (LCB) based optimization algorithm with proposed constraint handling approach on constrained test problems | 59 |
|--------------|--|----|
| Figure 3.7: | Engine calibration optimization using proposed algorithm | 61 |
| Figure 3.8: | Engine calibration optimization using EIM method | 62 |
| Figure 3.9: | Engine calibration optimization using M3-1 method | 63 |
| Figure 3.10: | Distribution of BSFC and NO_x measurements | 65 |
| Figure 3.11: | Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM and c) CLCB on ZDT1 and ZDT2 with 10% noise | 68 |
| Figure 3.12: | Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM and c) CLCB on ZDT3 with 10% noise | 69 |
| Figure 3.13: | Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM and c) CLCB on BNH and SRN with 10% noise | 73 |
| Figure 3.14: | Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM and c) CLCB on TNK and OSY with 10% noise | 74 |
| Figure 3.15: | Algorithm implementation for engine calibration | 75 |
| Figure 3.16: | Performance of constrained Augmented Expected Improvement (CAEI) al- gorithm on engine calibration problem at 10% noise level | 76 |
| Figure 3.17: | Performance of Augmented Expected Improvement Matrix (AEIM) algorithm on engine calibration problem at 10% noise level | 77 |
| Figure 3.18: | Performance of constrained Lower Confidence Algorithm (CLCB) on engine calibration problem at 10% noise level | 78 |
| Figure 4.1: | Complete experimental setup for performing surrogate-assisted optimization process | 83 |
| Figure 4.2: | Complete experimental setup for performing surrogate-assisted optimization process | 84 |
| Figure 4.3: | Setup for automating the engine calibration process | 91 |
| Figure 4.4: | Operating points for performing experiments | 93 |
| Figure 4.5: | Pareto front in objective and design spaces using deterministic Kriging model . | 95 |

| Figure 4.6: | Response surface model for BSFC and NOx using deterministic Kriging model 96 |
|--------------|---|
| Figure 4.7: | Convergence at different scenario using deterministic and stochastic Kriging model |
| Figure 4.8: | Response surface model for BSFC and NOx using stochastic Kriging model 99 |
| Figure 4.9: | Pareto front in objective space and design space using deterministic and stochastic Kriging models |
| Figure 4.10: | Engine calibration results at 1400 RPM - 500 Nm |
| Figure 4.11: | Engine calibration results at 1400 RPM - 175 Nm |
| Figure 4.12: | Engine calibration results at 1800 RPM - 400 Nm |
| Figure 4.13: | Engine calibration results at 1000 RPM - 325 Nm |
| Figure 4.14: | Pareto fronts in objective (the 1^{st} row) and design (the 2^{nd} row) spaces using deterministic and stochastic Kriging model |
| Figure 4.15: | COV variation in the design space |
| Figure 5.1: | Schematic of the eBoost configuration |
| Figure 5.2: | Variation of VGT vane and EGR valve positions with eBoost speed |
| Figure 5.3: | Control parameter variations for three-parameter calibration problem 116 |
| Figure 5.4: | Nominal BSFC and NO_x values |
| Figure 5.5: | Initial DOE points using constrained latin hypercube sampling (CLHS) in design space |
| Figure 5.6: | Output values corresponding to initial DOE inputs |
| Figure 5.7: | Total experimental evaluations in the Objective Space |
| Figure 5.8: | Experimental evaluation points in the design space |
| Figure 5.9: | Pareto front obtained from the surrogate model after all the evaluations 125 |
| Figure 5.10: | Points corresponding to the Pareto front from the surrogate model in the design space |

| Figure 5.11: | Error (%) between actual data and estimates from the Surrogate Model 127 |
|--------------|---|
| Figure 5.12: | Comparison of experimental evaluations using existing and proposed approaches 129 |
| Figure A.1: | Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on ZDT1 and ZDT2 with 5% noise |
| Figure A.2: | Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on ZDT3 with 5% noise |
| Figure A.3: | Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on BNH and SRN with 5% noise |
| Figure A.4: | Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on TNK and OSY with 5% noise |
| Figure A.5: | Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on ZDT1 and ZDT2 with 20% noise |
| Figure A.6: | Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on ZDT3 with 20% noise |
| Figure A.7: | Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on BNH and SRN with 20% noise |
| Figure A.8: | Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on TNK and OSY with 20% noise |
| Figure A.9: | Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on ZDT1 and ZDT2 with 40% noise |
| Figure A.10 | Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on ZDT3 with 40% noise |
| Figure A.11 | Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on BNH and SRN with 40% noise |
| Figure A.12 | Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on TNK and OSY with 40% noise |
| Figure B.1: | Performance of constrained Augmented Expected Improvement (CAEI) al- gorithm on engine calibration problem at 5% noise level |
| Figure B.2: | Performance of Augmented Expected Improvement Matrix (AEIM) algorithm on engine calibration problem at 5% noise level |

| Figure B.3: | Performance of constrained Lower Confidence Algorithm (CLCB) on engine calibration problem at 5% noise level |
|-------------|--|
| Figure B.4: | Performance of constrained Augmented Expected Improvement (CAEI) al- gorithm on engine calibration problem at 20% noise level |
| Figure B.5: | Performance of Augmented Expected Improvement Matrix (AEIM) algorithm on engine calibration problem at 20% noise level |
| Figure B.6: | Performance of constrained Lower Confidence Algorithm (CLCB) on engine calibration problem at 20% noise level |

KEY TO ABBREVIATIONS

- **BSFC** Brake Specific Fuel Consumption
- **VGT** Variable Geometry Turbocharger
- EGR Exhaust Gas Recirculation
- SOI Start of Injection
- **BMEP** Brake Mean Effective Pressure
- COV Coefficient of Variation
- IMEP Indicated Mean Effective Pressure
- **bTDC** Before Top Dead Center
- EI Expected Improvement
- LCB Lower Confidence Bound
- AASF Augmented Achievement Scalarization Function
- **EIM** Expected Improvement Matrix
- AEI Augmented Expected Improvement
- CAEI Constrained Augmented Expected Improvement
- AEIM Augmented Expected Improvement Matrix
- CLCB Constrained Lower Confidence Bound
- **GD** Generational Distance
- **IGD** Inverted Generational Distance
- **ZDT** Zitzler-Deb-Thiele Test Suit
- BNH Binh and korn Test Problem
- SRN Srinivas and Deb Test Problem
- TNK Tanaka Test Problem
- OSY Osyczka and Kundu Test Problem
- **DOE** Design of Experiments
- **GA** Genetic Algorithm

RGA Real Parameter Genetic Algorithm

NSGA-II Non-Dominated Sorting Algorithm II

SBX Simulated Binary Crossover

PSO Particle Swarm Optimization

EP Evolutionary Programming

DBSCAN Density-Based Spatial Clustering of Application with Noise

LHS Latin Hypercube Sampling

cLHS Constrained Latin Hypercube Sampling

CAS Combustion Analysis System

DAQ Data Acquisition

ECM Engine Control Module

ECU Engine Control Unit

CAN Controller Area Network

CAC Charge Air Cooler

CHAPTER 1

INTRODUCTION

1.1 Motivation

Internal combustion (IC) engines have been used for transportation since their inception. It still dominates the current world market, despite the availability of electric vehicles, and it is projected to have a significant market share in the future [31]. The major disadvantage of IC engines is their greenhouse emissions. Due to its large market share, this also contributes a high amount of harmful emissions, thus causing the adverse effects on climate change [29]. With the rising awareness, stricter emission norms have been imposed to maintain the emission levels below specific standard values, and these norms are getting stringent with time.

Several technical enhancements have been made to original engine systems to meet the emission standards with maintained or enhanced engine performance. This is motivated by the push to make vehicles cleaner with reduced greenhouse emissions. Several technological enhancements have been proposed and used for both Diesel and Gasoline engines to meet the desired requirements. It includes engine downsizing using boosting mechanism [11, 59], exhaust gas recirculation (EGR) for reducing NO_x emissions [25], electronic throttle for accurate torque control of Gasoline engines [76], and multiple fuel injections for Diesel engines [27, 23] for reducing NO_x and CO_2 emissions. Advanced combustion techniques are also adopted, such as turbulent jet ignition (TJI) [66], and partial premixed combustion (PPC) [73]. All these technological modifications to meet the demand for enhanced fuel economy with satisfactory emissions are making the engine system complicated. These technological enhancements work well in achieving the desired engine performance but lead to a significant increment in the number of control parameters with non-linear interactions.

Engine calibration is defined as a process of optimizing the control parameters to achieve optimal performance. The process is divided into **Static calibration** and **Transient calibration**.

This research work only deals with static engine calibration, where the calibration is done at a fixed speed-load point, called the engine operating point. These operating points, used to perform engine calibration, are extracted from a given driving cycle based on the information of time duration spent near these cluster points, as shown in Figure 1.1.



Figure 1.1: Extraction of Speed-Load points from the drive cycle

For each operating point, a set of control variables are calibrated to achieve optimal engine performance in terms of fuel economy and emissions. The calibration process or control optimization is straightforward if the system is simple with one or two control variables. However, with the increased system complexity of up to ten calibration parameters, the control optimization process becomes challenging in multi-dimensional parameter space to achieve optimal engine performance. This research work proposes a method to reduce the engine calibration cost (total experimental evaluations) needed to perform the control parameter calibration (optimization) using the machine learning based approach.

1.2 Existing Approaches for Engine Calibration

The engine control calibration problem has a long history, motivated by the desire to enhance the engine performance output with reduced emissions. Over time, engine systems became very complicated due to many enhancements to improve their efficiency with reduced harmful emissions. These enhancements lead to a significant increment in the number of control parameters, making the system complicated to calibrate. This section provides details of various approaches being used over time for performing engine control calibration. The automotive research community has implemented several innovative strategies to expedite the overall engine calibration process.

1.2.1 Full Factorial DOE Method for Engine Calibration

The classical approach to perform the engine calibration is using the space-filling full-factorial design of experiments (DOE) method to generate a set of control inputs, which are then implemented to the engine experimental system to obtain the corresponding outputs. The design of experiments (DOE) is a way to systematically develop test points for finding the optimal solutions. In the full-factorial method, each control parameter is divided into multiple points distributed uniformly over the variation range. If there are total 'p' control parameters and each control parameter is divided into 'n' points, the total number of design points (N) with the full-factorial design is given by

$$N = n^p \tag{1.1}$$

This method is simple to implement and works well when the number of control parameters to calibrate is small (e.g., one or two). The major drawback is with a system that has a large number of control parameters (e.g., ten). This method becomes intractable as the number of control variables increases. Also, for precise optimal values, the total number of split n needed to be increased, leading to a significant increment of total test points N. This approach has been shown to require a large number of function evaluations for performing the diesel engine calibration [42]. Unfortunately, engine calibration is an expensive evaluation process and normally does not afford the required exhaustive search budget to obtain optimal calibration values due to both cost and time.

1.2.2 Model-Based Approach for Control Calibration

The availability of high-performing computers has revolutionized the field of engineering. It became possible to replicate the behavior of an actual physical system using a sophisticated (high fidelity) system model, which decreases the complete dependency on using a physical system to test the developed control strategy. This also helped the automotive industries shift the engine



Figure 1.2: Control calibration strategy with physics-based model

calibration work to use a combination of model-based and experiment-based methods to perform engine calibration and curb the experimental cost.

With the availability of engine modeling software, the existing calibration approach in the automotive industry is to use a physics-based engine model to simulate the engine behavior without actually performing engine experiments. GT-SUITE^{TM} based one-dimensional engine model is widely used in assisting the engine calibration process. It works well in mimicking the engine behavior. One critical aspect of getting superior performance out of the GT-SUITE^{TM} model is using the modeled combustion dynamics to predict engine behavior. A simple engine model lacks in capturing the detailed combustion mechanism, which predominantly affects the control parameter calibration. A GT-SUITE^{TM} DI-PULSE model was developed by Gamma Technologies to capture the combustion phenomenon to bridge this gap and is shown to work well in capturing the combustion phenomenon to bridge this gap and is shown to work well in capturing the combustion phenomenon under different engine speeds and load conditions [54]. Recent work [36, 45, 65] also discusses the application of a reaction-based combustion model to predict engine performance. Performance from all these methods is dependent on how well the model is calibrated using experimental data. One of the disadvantages of GT-SUITE^{TM} models is their complexity with relatively high computational load, slowing down the model-based calibration process.



Figure 1.3: Control calibration strategy with data-driven model

Figure 1.2 shows the overall structure of the engine calibration process using a physics-based model. The process works by using the model to estimate the optimal control parameters and then evaluating it on the actual experimental setup. It is challenging to make the model predict engine performance accurately under all engine speed and load conditions. Therefore, the results from the model are validated on the physical system to identify the modeling error, and the model can be improved iteratively. To obtain the optimized parameters from the engine model, the full-factorial DOE method explained above is used. The entire process gets repeated until a satisfactory performance is achieved. Even though the calibration is performed on the model, further reduction in time to obtain the optimized parameters can be achieved using the evolutionary optimization approach, which has shown to reduce the computational time significantly [69, 22, 55]. A practical engine calibration using the mentioned model-assisted approach could take roughly one year for a diesel engine with more than ten parameters even with all the advanced methodologies. It is becoming crucial to optimize or calibrate the engine system quickly without exhausting experimental evaluations in a world with rapidly changing technology. This allows making changes to the system without worrying about its high calibration cost and time.

Recently, there is a rise in using machine learning-based approaches for control applications. In place of a physics-based model, a data-driven model, called the black-box model, is generated and

used. The model uses input-output data sampled from the actual system to model the interaction between inputs and outputs without worrying about the physics behind these interactions. Figure 1.3 shows the overall process of machine learning, where the physics-based model previously used is replaced by the data-driven model. One key difference here is the model update. If the data-driven model represents the physical system well, there is no need to update the model. However, suppose the model, to begin with, is formed with limited input-output data points. In that case, the initial model will not represent the system well, and the model can be improved iteratively based on new test data.

Various approaches are presented in the Machine Learning literature for model development: 1) Polynomial Regression; 2) Neural Network; 3) Kriging (also known as Gaussian Process Regression); 4) Support Vector Machine etc. The approach of using a data-driven model in the context of engine calibration has been demonstrated before. The work in [7, 70, 20] uses the Neural Network-based models for engine calibration. Berger et al. [3] did a comparative study of using Gaussian Process models for engine modeling. Gutjahr et al. [19] performed engine calibration using a Gaussian process model formed based on 500 data points. Dellino et al. [10] performed design optimization of a CNG (compressed natural gas) injection system using Kriging meta-models. They evaluated 11 different model management strategies with a focus on model accuracy and computational efficiency. They showed that all 11 methods were able to provide accurate optimal predictions, but not all the schemes were computationally efficient.

1.2.3 Model-Free Approach for Control Calibration

The subsection above discussed the existing model-based control calibration strategy, which requires model development before devising the control calibration. Calibration of the control parameter depends on how well the model represents the actual system. As mentioned before, the developed model, both physics-based and data-driven, requires experimental data for model calibration, which is an essential step to improve these models but costs the experimental budget. Because this project aims to reduce the experimental evaluations, a model-free control calibration approach is proposed,



Figure 1.4: Structure for model-free control calibration strategy

which does not explicitly require system model development. Figure 1.4 shows the architecture of the proposed process.

In the Model-Free strategy, the control strategy inherently uses the black-box model to perform the optimal control calibration. The approach is motivated because of the need to fill the gap of performing engine calibration with a reduced number of total function evaluations without developing high-fidelity models. The current work explores the option of implementing the modelfree control calibration, known as Surrogate-Assisted Optimization. This approach has been proven to reduce the total evaluation budget when the system is complex, expensive, or time-consuming to evaluate.

Various researchers have implemented this approach to optimize system performance. Zhou et al. [77] tried calibrating parameters for groundwater reactive transport models. They were able to reduce the computational burden from 45000 evaluations to 500 evaluations using the surrogate-assisted method. Another application was sizing optimization of analog /RF circuits by Lyu et al. [41]. For surrogate-assisted optimization, they proposed to use lower confidence bound (LCB) criteria that do not require any information from the previously evaluated points (explained in Chapter 2). However, in their work, they did not consider constraints in their optimization problem. Also, they did a comparative study between existing methods (namely, MOEA/D [38] and GPMOOG [37] algorithm) and found that their method works well compared with the existing surrogate-based optimization techniques. Liu et al. [39] optimized actuator design

using this approach with a newly proposed optimization technique, called PAGHO (Parallel adjoint sensitivity and Gaussian process assisted hybrid optimization technique), based on the gradient information to achieve convergence. This algorithm was proposed for parallel computing using multiple processors, significantly reducing required computational time and obtaining optimized points efficiently. Another exciting application was optimizing chemical reactor design [52] by Park et al. They were able to get the Pareto solution within 100 high-fidelity evaluations. Ali et al. [1] performed optimization of a natural gas liquefaction plant.

1.3 Research Overview

After finalizing the proposed approach, the goal is to identify and develop the proposed algorithm capable of handling generic engine calibration problems well and is implementable to an actual engine experimental setup. The ultimate goal of this study is to demonstrate the approach on an actual Ford diesel engine installed on a dynamometer, available at the Energy and Automotive Research Lab of the Michigan State University. The proposed research work is therefore progressed in three different stages, as explained next.

1.3.1 Selecting the Algorithm for Experimental Evaluation

As will be explained in the next chapter (Chapter 2), the surrogate-assisted optimization approach requires forming an acquisition function for efficient exploration and exploitation in the design space. Various approaches are available in the literature. Expected Improvement (EI) and Lower Confidence Bound (LCB) are two widely used acquisition functions to perform this operation. EI criteria work well without external user input but extending it to a multi-objective or many-objective problem is not straightforward. LCB is much simpler to formulate in such scenarios. However, there is no efficient constraint handling method available for LCB. Note that a practical engine calibration problem is a multi-objective problem with constraints, and therefore, it becomes crucial to address the issue of handling constraints with LCB.

To bridge the gap of handling constraints with the Lower Confidence Bound (LCB) based al-

gorithm, a new constraint handling method was proposed, which works well in different scenarios. Details of the proposed algorithm will be discussed in Chapter 2. The proposed approach with its application to the engine calibration problem was published in the IEEE/ASME Transactions on Mechatronics [49]. In that paper, the proposed method was first evaluated on several numerical test problems before implementing it for the engine calibration study. For the performance comparison, the modified algorithm was compared with the state-of-the-art expected improvement (EI) algorithms. The entire analysis was done in a deterministic setting without considering measurement noise. However, for a practical system, this is never true. The actual system is always corrupted with measurement noise. For practical implementation, an algorithm should be able to handle the measurement noise without hampering its performance. This is a natural step next towards algorithm development for practical applications, that is, development for its stochastic extension.

1.3.2 Stochastic Extensions of Proposed Algorithm for Practical Systems

The engine calibration problem is usually a problem of optimizing various control parameters to obtain the best possible objectives with satisfactory constraints. There could be multiple objectives to handle with several constraints. Taking practicality into consideration, the goal of the proposed algorithm should be to solve a multi- or many-objective constrained problem with measurement noise. There were gaps in the literature for performing surrogate-assisted optimization in the stochastic environment. The extension of the surrogate-assisted optimization algorithm to handle problems with measurement noise is available in numerous literature, but most of them are for single-objective problems. For solving our multi-objective problem, two important things were considered: 1) the algorithm should be able to work for constrained multi-objective problems, and 2) it should be able to handle non-uniform (Heteroscedastic) noise. It is highly unlikely for a practical system to have uniform noise. Because the end goal of this study is to develop an algorithm for a practical system, these steps become crucial before moving towards experimental validation.

All the applications discussed in Section 1.2.3 were simulation-based studies. A new set of

literature was explored to get motivated for the stochastic extension of the proposed algorithm. Various extensions of the proposed surrogate-assisted optimization algorithm were available in the literature. Reference [57] summarizes the extensions for single-objective unconstrained problems with homoscedastic (uniform) noise variance. The work by Jalali et al. [26] compared a few different stochastic methods with heteroscedastic noise variance. Most of the work available extended the original approach for handling single objective problems with only a few taking into account the multi-objective constrained problems [18, 75].

This gap is bridged in this research by proposing three different extensions of the available algorithms. Two of the proposed algorithms are based on Expected Improvement (EI) criteria, while the third one is using the Lower Confidence Bound (LCB) along with our proposed constrained handling algorithm. In order to identify the best possible algorithm to be implemented for performing the experimental study, a comparative study is done using the engine simulation model. To make the model mimic the actual engine behavior, external noise is added to each measured output. A preliminary version of the work was presented at the 2020 American Control Conference [51], and the work showing the detailed results with a comparative study among all three methods at different noise levels was published in ASME Journal of Dynamic Systems, Measurement, and Control [50]. The detailed explanations and results of the developed algorithms will be discussed in Chapters 2 and 3.

1.3.3 Algorithm Implementation to Engine Dynamometer Setup

The final hurdle is the implementation of the developed algorithm into an engine dynamometer setup. The engine calibration process involves a manual intervention to detect if the parameter optimization process is performing well. It needs various checks to ensure the safe operation of the engine dynamometer. To the best of our knowledge, this is the first time implementing the stochastic surrogate-assisted optimization algorithm for performing the engine calibration on an actual experimental test bench. Since the approach automates the calibration process and there is no manual interference, the challenge is to ensure the safe operation of the developed algorithm with

all possible checks. Usually, the actual test setup is a highly complex system, with several systems communicating with each other to run the engine at the desired condition. The challenge is to develop a system that will replace human control with a host PC running the developed algorithm and ensuring a safe operating range while simultaneously evaluating the developed algorithm's performance.

The experimental setup is developed with all the safety considerations, where the host PC communicates among different systems to capture the desired measurements. The developed algorithm is implemented in the host PC. Based on the obtained measurements, a new set of control inputs are pushed to the engine control module (ECM) to change the control parameters. Chapter 4 discusses the entire experimental setup in detail, with some exciting results observed after implementing this approach. We initially formulated a simple problem to get the confidence of running the algorithm on the dynamometer. After obtaining reasonable results, different problem formulations are tried to see the potential of the developed algorithm. Finally, we demonstrated the calibration of a complex system with an electrical boosting mechanism (eBoost). Problem formulation contained three control parameters to calibrate with two objectives and seven constraints. Results demonstrated the advantage of using eBoost for obtaining the best engine efficiency possible while maintaining the emissions below the acceptable limit. Details of all the experimental results will be discussed in Chapter 4 and Chapter 5.

It is the first and important step in demonstrating the practicality of the proposed approach in reducing the manual interference and the model dependence from the calibration process. The entire engine calibration process is automated and showed a significant reduction in the experimental evaluations for obtaining the optimized control parameters. A comparison of the total number of experimental evaluations required between the DOE-based approach and the proposed algorithm is shown in Figure 1.5. This result is observed based on actual experimental tests. As can be seen, increasing the number of control variables causes an exponential increment in the number of experimental evaluations required for performing the engine calibration using the existing method. However, this does not get reciprocated in the proposed surrogate-assisted optimization approach.



Figure 1.5: Comparison of experimental evaluations between existing and proposed approaches

There is a significant difference in the experimental evaluation with just three control parameters. This difference might increase as the number of control parameters increases, which is usually the case with engine calibration problems.

1.4 Research Contributions

The main contribution of this research work resides in the development and implementation of the surrogate-assisted optimization approach for performing the engine calibration. The major contributions of this dissertation can be summarized as below:

 A new constraint handling method is proposed for Lower Confidence Bound (LCB) based acquisition function. This method only requires information of model mean and uncertainty estimation at unknown points. The proposed method has been shown to work well for various constrained optimization test problems and engine calibration problems.

- 2. Three different extensions are made to perform stochastic surrogate-assisted optimization in the presence of heteroscedastic (non-uniform) measurement noise to implement the algorithm on practical systems. These algorithms are evaluated by different multi-objective constrained optimization problems to show the efficacy of these proposed extensions. The results are promising from the Lower Confidence Bound-based constrained optimization. Because of its ease of implementation, the constrained lower confidence bound (CLCB) method is used to perform the experimental study.
- 3. Experimental studies have been done to validate the efficacy of this approach. Initial experimental work is performed with only two variables to understand the approach. Lower confidence bound with proposed constraint handling is shown to work well in identifying the near-optimal front. A comparative study is done between the stochastic and deterministic surrogate-assisted optimization approaches. The stochastic approach is shown to work better than the deterministic approach. The latter one would get caught up exploring unwanted areas due to the creation of multiple local bins. This is attributed to the overfitting of the input-output data during model development. Later, the calibration results of a complex engine calibration problem with three control parameters are shown. The entire engine calibration process is automated and showed a significant reduction in the experimental evaluations for obtaining the optimized control parameters. This is the first and important step in demonstrating the practicality of the proposed approach in reducing manual interference and model dependence during the calibration process.

1.5 Dissertation outline

The complete dissertation is organized as follows. Chapter 2 provides a detailed discussion about the surrogate-assisted optimization approach for both deterministic and stochastic systems. It describes the model development process and necessary modifications for handling measurement noise, discussion of various acquisition functions and their modifications to handle multi-objective constrained optimization problems, and the test point selection for further evaluations. Validation

of the proposed algorithms and their extensions are shown in Chapter 3. The entire chapter is dedicated to the study of the Engine calibration problem in a simulation environment. The engine calibration problem in Chapter 3 is performed using a high-fidelity GT-SUITE^{TM} model. After observing the satisfactory results from the simulation study, Chapter 4 addresses the initial engine calibration results by implementing the algorithm onto an actual engine test bench. The chapter initially discusses the modifications made to the original setup to implement the proposed algorithm, followed by the initial experimental results. Chapter 5 discusses a more complicated problem of performing engine calibration with an external boosting device (eBoost). Chapter 6 concludes the study with recommendations for future work.

CHAPTER 2

SURROGATE ASSISTED OPTIMIZATION FOR PRACTICAL SYSTEMS

Surrogate-assisted optimization is a process of finding the global optimal region with the help of data-driven models, also known as surrogates. The approach performs an efficient search operation without evaluating the entire model. The algorithm intelligently processes the information available from the model to direct the search towards the global optimal region. This, in turn, reduces the evaluation budget needed to find the global optima since evaluations away from the optima are not required. Note that the main goal of this study is to reduce the experimental evaluations for performing the engine control calibration. This chapter provides a detailed explanation of the approach in the context of dealing with an expensive multi-objective constrained optimization problem with measurement noise. Most of the existing work on implementing this approach to practical systems is simulation-based, without considering noise in the system measurements. Because the goal here is to develop an algorithm applicable to practical systems, extensions are made to the algorithm to handle stochastic measurements. This chapter discusses each step involved in the proposed algorithm and modifications made to make it applicable to practical systems.

The first step in the engine calibration study is the problem formulation. It is an important step that defines the objectives we are trying to optimize and the control variables we are interested in changing. The actual algorithms start after problem definition. The next section discusses the problem formulation, followed by a detailed explanation of the proposed algorithm.

2.1 **Problem Formulation**

Modifications to the engine system architecture are driven by the goal of enhancing its performance. The two major performance indices for any engine calibration problem are maximizing the engine efficiency while maintaining the emissions below the desired threshold value. Engine efficiency can be measured in terms of brake-specific fuel consumption (BSFC), indicating the amount of fuel consumed by the engine to generate one kilowatt power output. Note that the smaller the brake-specific fuel consumption is, the higher the engine operating efficiency. The second and the most crucial parameter is engine output missions. Several harmful emissions are observed for both Gasoline and Diesel engines, namely Nitrous oxide (NO_x), Hydrocarbon (HC), soot (PM), carbon monoxide (CO), and carbon dioxide (CO_2) [63]. All these emissions are harmful to the environment, and human health [68, 43] and are required to meet regulative emission standards. For the current work, engine efficiency in terms of brake-specific fuel consumption (BSFC) and the NO_x emissions are considered as the desired system outputs to optimize.

It is well known that optimizing the in-cylinder combustion process leads to improved engine efficiency. This, in turn, adversely affects the NO_x emissions. The high in-cylinder combustion temperature causes the nitrogen and air in the combustion mixture to turn to nitrous oxide, thus increasing the emission level, especially for the optimized combustion process. On the contrary, reducing the NO_x emissions adversely affects the combustion efficiency. Both objectives considered here are inversely related to each other. Therefore, a multi-objective optimization problem could be formulated to perform the engine calibration.

2.1.1 Constrained Multi-Objective Optimization Problem

A general constrained multiple objective optimization problem can be described as follows.

Minimize
$$y_1(\mathbf{x}), y_2(\mathbf{x}), ..., y_m(\mathbf{x})$$

subject to $g_1(\mathbf{x}), g_2(\mathbf{x}), ..., g_n(\mathbf{x}) \ge 0$
 $\mathbf{x} = [x_1, x_2, ..., x_p]^T \in \mathbb{R}^p$ (2.1)
 $y_i \in \mathbb{R}, \ i = 1, 2, ..., m$
 $g_j \in \mathbb{R}, \ j = 1, 2, ..., n$

where y_i (i = 1, 2, ..., m) are objective functions and m is total number of objective functions; g_j (j = 1, 2, ..., n) are inequality constraint functions and n is total number of constraints; and \mathbf{x} is a variable vector representing all control parameters.

As discussed above, the current study only deals with two objectives. Constraints are added to make the operation run within the desired boundary. An upper limit on the emissions could be added to maintain the emission level below the required threshold value. Another constraint on boosting pressure can be added to run the engine in a safe operating region. Constraints even on the control variable range can be imposed to avoid engine running in the infeasible region, thus maintaining the safe operation. Additional constraints can be added to the problem, depending on the situation and requirements.

Note that \mathbf{x} is a vector of control parameters, and space spanned by the control variable is called *Design Space*. As mentioned at the beginning, modern combustion engines consist of multiple control parameters. Therefore, depending on the problem, variable vector \mathbf{x} can be adjusted. With this problem formulation in mind, we will devise the strategy for performing the calibration.

2.2 Overall Algorithm

Figure 2.1 shows a complete architecture of the optimization process. It starts by creating the initial model for the learning algorithm. Based on this initial model, the algorithm tries to look for the possible optimal region and tells the actual system to perform the high-fidelity evaluation. Therefore, it becomes crucial to get a good initial model representing the model well in the entire design space.

For the model development, we need a set of input-output data. Design of experiments (DOE) is a way to generate a set of input points that span the design space well. Design space here is defined as the control parameter variation region. Latin Hypercube Sampling (LHS) is a widely used DOE method for creating uniformly distributed points throughout the design space. If the control variables have constraints, constrained Latin Hypercube Sampling (cLHS) [53] could be used. Initial points are then evaluated using a high fidelity system (through physical tests or simulations) to get a set of data points for initial surrogate model development. Now, one can use many points to generate a better initial model, but that will take a significant part of the total evaluation budget. The advantage is that the surrogate-assisted optimization algorithm will get a better initial model to work with, but this will give less chance to the learning algorithm to update the model near the optimal region. On the contrary, using a smaller number of points for initial model development



Figure 2.1: Overall architecture of the surrogate-assisted optimization process

would not cover the design space well. Therefore, it could mislead the learning algorithm to look for points in the not-so-good region. This affects the convergence rate and could lead to the learning algorithm wander a lot before convergence. This is the trade-off, and it is recommended in [30] to use minimum initial points of 10p, where p is the total number of control parameters, to develop the initial surrogate model.

With the set of input-output data, the next step is model development. This is an important step as it assists the algorithm in searching for possible optimal regions. For the surrogate-assisted optimization algorithm, Kriging is the most popular method used for model development. It not only provides mean value at any unknown point like other models but also provides the uncertainty bounds. Any unknown test point is associated with mean and uncertainty estimates from the model. This information is crucial in searching the design space efficiently, as explained later (Section 2.4).

Since the developed model would not accurately represent the actual surface, performing direct

optimization on the response surface model could mislead the search direction. An acquisition function is an intelligent way to combine the model estimates to direct the search towards a possible optimal location in the design space. It is developed to balance the exploration and exploitation of design space intelligently while searching for the possible global optimal region. Acquisition function is developed for objective functions. If the constraints are also expensive to evaluate, it becomes imperative to devise a strategy to perform the exploration and exploitation of the constraint functions intelligently. Therefore, the complete problem formulation involves acquisition function development with a constraint handling mechanism.

The final step involves optimizing the acquisition function to get points in the possible global optimal region. These points could further be selected for high fidelity evaluations. With these points selected from the selection pool, the high-fidelity function is evaluated next to get the set of input-output data points. The new input-output data set is then archived with the old data, which helps to update the belief of the model for the next iteration. This iterative loop keeps going until the evaluation budget is exhausted or the desired accuracy is achieved.

This is the general discussion of the overall algorithm, and different steps can be modified as per problem requirements. The following sections will discuss each step in detail. Each section will explain the original approach followed by the modifications made to handle the current problem.

2.3 Model Development

Various methods are available in the existing literature for surrogate model development, namely, polynomial regression, Kriging, radial basis function (RBF), support vector machines (SVM), neural network, etc. [48, 62]. All these models can be characterized as parametric and non-parametric models. For the parametric models, the structure for model development is defined beforehand. Therefore, these approaches work well if the system structure is known beforehand. Non-parametric models work without considering any structure, which is more beneficial for our application because the structure of the engine system is unknown.

Out of all the methods available, Kriging, also known as Gaussian Process Regression (GPR), is
the most popular method used for performing surrogate-assisted optimization. It is a non-parametric model. Therefore it does not assume the structure of the model beforehand and keeps updating the structure as more points are added to the data set. One of the important characteristics of the Kriging model is that it estimates both mean and uncertainty values at any unknown point. These values are helpful in efficiently exploring and exploiting the search for the optimal region in the design space, as will be explained in Section 2.4. This section will provide the mathematical details of the model development for both deterministic and stochastic Kriging models.

2.3.1 Kriging Model

Kriging is a probabilistic modeling method to fit the data points. It starts by assuming a prior mean function and a stochastic function, and their posterior distribution is calculated based on the observed data points to represent the model structure. The mean function is used to capture the behavior of system dynamics, whereas the stochastic function helps to fit the model better. The original Kriging model does not consider measurement noise during the modeling process. Therefore, it performs the interpolation rather than regression, meaning the model passes through all the known points.

The following is an explanation of the entire Kriging model development process. The expression for a scalar output measurement $y(\mathbf{x})$ at any input vector \mathbf{x} with no measurement noise can be represented by

$$y(\mathbf{x}) = \mathbf{f}^T(\mathbf{x})\boldsymbol{\beta} + z(\mathbf{x})$$
(2.2)

where $\mathbf{x} \in \mathbb{R}^p$ is the input vector of dimension $p, \beta \in \mathbb{R}^q$ is a vector of regression coefficients, $\mathbf{f} \in \mathbb{R}^q$ is a regression function vector, and $y(\mathbf{x})$ is a scalar output. The first part represents the regression term. It is added to capture the trend of the system under consideration. It could be constant ($\mathbf{f}(\mathbf{x}) = 1$) or a general polynomial function ($\mathbf{f}(\mathbf{x}) \in \mathbb{R}^q$). Considering a second order function $\mathbf{f}(\mathbf{x})$ with the input vector dimension p = 2. The regression term will become:

$$\mathbf{f}^{T}(\mathbf{x})\boldsymbol{\beta} = (1) \cdot \beta_{1} + (x_{1}) \cdot \beta_{2} + (x_{2}) \cdot \beta_{3} + (x_{1}^{2}) \cdot \beta_{4} + (x_{2}^{2}) \cdot \beta_{5} + (x_{1}x_{2}) \cdot \beta_{6}$$

Therefore, in this case, $\mathbf{f}(\mathbf{x}) = [1, x_1, x_2, x_1^2, x_2^2, x_1x_2]^T$ and q equals to 6.

The second term is a stochastic term, also known as extrinsic uncertainty. It is added to fit the model better. It makes the model pass through all the points, thus making Kriging an interpolation model. It is usually modeled as a zero-mean function with a covariance function defined between points in the design space. Here, $z(\mathbf{x})$ can be thought of as a sample mapping from \mathbb{R}^p to \mathbb{R} . The expression for the covariance function between any two points (\mathbf{x}^i and \mathbf{x}^j) in the design space can be expressed as

$$Cov[\mathbf{x}^{i}, \mathbf{x}^{j}] = k(\mathbf{x}^{i}, \mathbf{x}^{j}) = \sigma^{2} \operatorname{Corr}[\mathbf{x}^{i}, \mathbf{x}^{j}]$$
(2.3)

where σ^2 in equation (2.3) represents process variance with correlation function Corr. Various correlation functions are available in reference [61]. For this work, Gaussian correlation function is used for model development and it is defined by

$$\operatorname{Corr}[\mathbf{x}^{i}, \mathbf{x}^{j}] = exp\left(-\sum_{l=1}^{p} \theta_{l} |x_{l}^{i} - x_{l}^{j}|^{2})\right)$$
(2.4)

where θ_l , x_l^i and x_l^j are the l^{th} element of Gaussian distribution width vector $\boldsymbol{\theta}$, design space vectors \mathbf{x}^i and \mathbf{x}^j , respectively.

With all these, the model has three unknown parameters σ , β , and θ , also known as hyperparameters. Based on all the observed points (total *N* points), the values of these hyperparameters are optimized to obtain the posterior distribution of the model. Their optimal values are determined by maximizing the marginal likelihood function (*L*) below. The expression for the likelihood function after taking natural logarithm becomes

$$h(\sigma, \boldsymbol{\beta}, \boldsymbol{\theta}) = ln(L) = -\frac{1}{2} \Big[Nln(2\pi) + ln(det(\mathbf{K})) + (\mathbf{y} - \mathbf{F}\boldsymbol{\beta})^T \mathbf{K}^{-1} (\mathbf{y} - \mathbf{F}\boldsymbol{\beta}) \Big]$$
(2.5)

where *h* represents the log-likelihood function, $\mathbf{x}^i \in \mathbb{R}^p$ (*i* = 1, 2, ..., *N*) are *N* observed points and

$$\mathbf{y} = [y(\mathbf{x}^1), y(\mathbf{x}^2), ..., y(\mathbf{x}^N)]^T \in \mathbb{R}^N$$
$$\boldsymbol{\beta} = [\beta_1, \beta_2, ..., \beta_q]^T \in \mathbb{R}^q$$
$$\mathbf{F} = [\mathbf{f}(\mathbf{x}^1), \mathbf{f}(\mathbf{x}^2), ..., \mathbf{f}(\mathbf{x}^N)]^T \in \mathbb{R}^{q \times N}$$
$$\mathbf{K} = \text{Covariance Matrix} = [k(\mathbf{x}^i, \mathbf{x}^j)_{1 \le i, j \le N}] \in \mathbb{R}^{N \times N}$$

Using $\mathbf{K} = \sigma^2 \mathbf{R}$, where **R** is a correlation matrix, and rearranging terms in equation (2.5) yields

$$h(\sigma, \boldsymbol{\beta}, \boldsymbol{\theta}) = \ln(L) = -\frac{1}{2} \Big[N \ln(2\pi\sigma^2) + \ln(\det(\mathbf{R})) + (\mathbf{y} - \mathbf{F}\boldsymbol{\beta})^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{F}\boldsymbol{\beta}) / \sigma^2 \Big]$$
(2.6)

Note that equation (2.6) depends on three parameters (σ , β , θ) and their optimum values can be obtained by simultaneously solving following three equations:

$$\frac{\partial l(\sigma, \boldsymbol{\beta}, \boldsymbol{\theta})}{\partial \sigma} = 0 \qquad \frac{\partial l(\sigma, \boldsymbol{\beta}, \boldsymbol{\theta})}{\partial \boldsymbol{\beta}} = 0 \qquad \frac{\partial l(\sigma, \boldsymbol{\beta}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = 0$$
(2.7)

Using the first two conditions from equation (2.7), optimal values of σ and β can be obtain as follows.

$$\sigma = \frac{1}{N} (\mathbf{y} - \mathbf{F}\boldsymbol{\beta})^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{F}\boldsymbol{\beta})$$
(2.8)

$$\boldsymbol{\beta} = (\mathbf{F}^T R^{-1} \mathbf{F})^{-1} \mathbf{F}^T R^{-1} \boldsymbol{y}$$
(2.9)

After substituting equations (2.8) and (2.9) into equation (2.6) and using the last expression in equation (2.7), we can obtain the optimal θ . Using these optimal values of the hyperparameters, the expressions for scalar output mean ($\mu(\mathbf{x})$) and variance ($s^2(\mathbf{x})$) at any test point \mathbf{x} can be estimated as

Mean

$$\mu(\mathbf{x}) = \mathbf{f}^{T}(\mathbf{x})\boldsymbol{\beta} + r^{T}(\mathbf{x})\mathbf{R}^{-1}(\boldsymbol{y} - \mathbf{F}\boldsymbol{\beta})$$
Uncertainty
$$s^{2}(\mathbf{x}) = \sigma^{2}[1 + u^{T}(\mathbf{x})(\mathbf{F}^{T}\mathbf{R}^{-1}\mathbf{F})^{-1}u(\boldsymbol{x}) - r^{T}(\mathbf{x})\mathbf{R}^{-1}r(\mathbf{x})]$$

$$\Sigma^{T}\mathbf{R}^{-1}(\boldsymbol{x}) = c^{T}(\boldsymbol{x}) - c^{T}(\mathbf{x})\mathbf{R}^{-1}r(\mathbf{x})]$$
(2.10)

where $u(\mathbf{x}) = \mathbf{f}(\mathbf{x}) - \mathbf{F}^T \mathbf{R}^{-1} r(\mathbf{x})$ and $r^T(\mathbf{x}) = [R(\mathbf{x}, \mathbf{x}^1), ..., R(\mathbf{x}, \mathbf{x}^N)].$

The detailed mathematical derivation can be found in [15, 61]. For the current work, Design and Analysis of Computer Experiments (DACE) MATLAB toolbox [46] by Nielsen et al. is used for model development.

2.3.2 Stochastic Kriging Model

The original Kriging model needs modification to handle stochastic measurements with measurement noises as explained in [2]. If the output measurement is noisy, the expression for equation (2.2) becomes

$$y(\mathbf{x}) = \mathbf{f}^T(\mathbf{x})\boldsymbol{\beta} + z(\mathbf{x}) + \boldsymbol{\epsilon}(\mathbf{x})$$
(2.11)

where $\epsilon(\mathbf{x})$ represents the measurement noise of output y at **x**. At any input point **x**, the noise structure is assumed to be zero mean Gaussian with non-uniform variance $\tau^2(\mathbf{x})$ ($\epsilon(\mathbf{x}) = \mathcal{N}(0, \tau^2(\mathbf{x}))$). The distribution changes with input point **x**. The Gaussian assumption of the measurement noise makes the calculation easy. Any non-Gaussian distribution can easily be transformed to Gaussian distribution. If the noise is considered homoscedastic (uniform), the expression for $\tau^2(\mathbf{x})$ becomes $\tau^2(\mathbf{x}) = \sigma_{\epsilon}^2$, where σ_{ϵ}^2 is the uniform noise variance present in the measurements. For this work, we are considering non-uniform noise variance, but the setup could easily be adjusted for uniform noise variance.

With this additional term in the expression, the expression for the log-likelihood function can be written as before:

$$h(\sigma, \boldsymbol{\beta}, \boldsymbol{\theta}) = \ln(L) = -\frac{1}{2} \Big[N \ln(2\pi) + \ln(\det(\mathbf{K} + \mathbf{T})) + (\mathbf{y} - \mathbf{F}\boldsymbol{\beta})^T (\mathbf{K} + \mathbf{T})^{-1} (\mathbf{y} - \mathbf{F}\boldsymbol{\beta}) \Big] \quad (2.12)$$

where $\mathbf{T} = diag[\tau_1^2, \tau_2^2, ..., \tau_N^2]$ is the noise covariance matrix for N observed points. Like the deterministic case, partial derivatives in (2.7) can be used to obtained the optimal unknown hyperparameters (σ, β, θ). Unfortunately, due to the addition of noise term (\mathbf{T}) in the expression, the closed form expressions for σ, β, θ are not possible. To get the optimized hyperparameter values, *pattern search* algorithm is used. With the observed hyperparameter values, the expressions for mean ($\mu(\mathbf{x})$) and uncertainty ($s(\mathbf{x})$) estimations at any test point \mathbf{x} can be given as

Mean

$$\mu(\mathbf{x}) = \mathbf{f}^{T}(\mathbf{x})\boldsymbol{\beta} + \mathbf{k}^{T}(\mathbf{x})\mathbf{K}_{n}^{-1}(\mathbf{y} - \mathbf{F}\boldsymbol{\beta})$$
Uncertainty
(2.13)

$$s^{2}(\mathbf{x}) = \sigma^{2} + u^{T}(\mathbf{x})(\mathbf{F}^{T}\mathbf{K}_{n}^{-1}\mathbf{F})^{-1}u(x) - \mathbf{k}^{T}(\mathbf{x})\mathbf{K}_{n}^{-1}\mathbf{k}(\mathbf{x})$$

where $\mathbf{K}_n = \mathbf{K} + \mathbf{T}$, $u(\mathbf{x}) = \mathbf{f}(\mathbf{x}) - \mathbf{F}^T \mathbf{K}_n^{-1} \mathbf{k}(\mathbf{x})$, and $\mathbf{k}^T(\mathbf{x}) = [k(\mathbf{x}, \mathbf{x}^1), ..., k(\mathbf{x}, \mathbf{x}^N)]$

Literature [2, 26] can be referred for detailed mathematical discussions on stochastic Kriging modeling.

2.3.3 Deterministic vs. Stochastic Kriging Models

As mentioned before, Kriging is an interpolation-based model. It passes through all the points used for model development. If the system is deterministic, the model will try to go through the input points to predict the best mean and uncertainty estimates. However, the interpolation property will tend to overfit the model if the output from the system has measurement noise. In order to show the possible issue, a simple test problem with a few input points can be used. The test problem considered here is given by



$$y = 5 + (6x - 2)^{2} sin(12x - 4)$$
(2.14)

Figure 2.2: Model fitting using deterministic Kriging model for a function without measurement noise

Figure 2.2 shows the plot for this simple test problem without measurement noise. It shows that the mean estimation from the model matches very well with the actual curve with only a few input points. Now, certain measurement noise is added to these test points to make them stochastic. Figure 2.3 shows the comparison of mean estimates from both deterministic and stochastic Kriging models, where the red curve shows the output from the deterministic model, and the blue curve



Figure 2.3: Model fitting using both deterministic and stochastic Kriging models for a function with measurement noise

shows the stochastic one. An interesting observation from the plot is that multiple optimal regions are created by fitting the deterministic model using the stochastic test data. A stochastic model performs regression rather than doing the interpolation. Therefore it is much smoother. This shows that with high dimensional complex practical problems with measurement noise, using the deterministic Kriging model might lead to exploration of unnecessary locations, thus wasting the evaluation budget in exploring regions other than possible global optimal regions.

2.4 Acquisition function

Using the model alone cannot efficiently guide the algorithm to look for the possible global optimal region. Because the Kriging model provides both mean and uncertainty estimates at an unknown point, it can be suitably utilized for search purposes. Acquisition function is a way to intelligently combine mean and uncertainty estimates from the model to direct the search towards a possible optimal location in the design space. It helps balance exploration (high uncertainty region)

and exploitation (best region found by so far) in the design space to identify the possible optimal region. When the problem is expensive to evaluate constraints, it is another challenge, and the acquisition function is modified accordingly (to be discussed later). Efficient constraint handling helps identify the feasible region to restrict the search in a smaller domain.

Several methods are available in the literature to form the acquisition function, namely: Probability of Improvement (PI), Expected Improvement (EI), Upper Confidence Bound (UCB), and Lower Confidence Bound (LCB) [72, 28]. Expected Improvement (EI) [30] is the most common method used because it does not require any user input and efficiently balances the exploration and exploitation. Extension of the EI-based acquisition function to handle constraints is also essential and addressed in the literature ([17, 16, 14]). Expected Improvement is simple to implement for a single objective problem, but it becomes computationally challenging when implementing it in a multi-objective scenario. Lower Confidence Bound (LCB) is another approach requiring external user input. However, it is straightforward to extend to multi-objective problems as it does not require information of previous best values.

This work uses both acquisition functions, Expected Improvement (EI) and Lower confidence bound (LCB), to explore the possible option of their implementation for a practical engine calibration problem. Because we have to deal with stochastic measurements, extensions have been proposed to handle multi-objective constrained optimization problems. Further sections will first provide details of these methods and then their extension to stochastic measurements.

2.4.1 Expected Improvement

Considering a single objective problem, the expression for output estimate $y(\mathbf{x})$ from the Kriging model at any unknown point \mathbf{x} can be written as:

$$y(\mathbf{x}) = \mathcal{N}(\mu(\mathbf{x}), s^2(\mathbf{x})) \tag{2.15}$$

where $\mu(\mathbf{x})$ and $s(\mathbf{x})$ corresponds to the mean and uncertainty estimation from the model at the unknown point \mathbf{x} , respectively. Considering the minimization problem, the improvement function

is defined as:

$$I(\mathbf{x}) = max(y_{min} - y(\mathbf{x}), 0) \tag{2.16}$$

where y_{min} is the current best objective value obtained from the set of observed data. The improvement function is defined to identify the point that will give the maximum improvement based on the model estimates. Equation (2.16) will be maximized at the location where $y(\mathbf{x})$ will be minimum. This is what we want since the optimization problem considered is a minimization problem. Therefore, this expression needs to be maximized to find point \mathbf{x} . Taking the expectation of the improvement function and performing mathematical computation, the closed-form expression for the Expected Improvement can be written as below.

$$EI(\mathbf{x}) = (y_{min} - \mu(\mathbf{x}))\Phi\left(\frac{y_{min} - \mu(\mathbf{x})}{s(\mathbf{x})}\right) + s(\mathbf{x})\phi\left(\frac{y_{min} - \mu(\mathbf{x})}{s(\mathbf{x})}\right)$$
(2.17)

The goal is to maximize the expression of expected improvement to get the possible global optimal solution. It can be observed from equation (2.17) that the first term is mean value dominant, and the second term depends more on the uncertainty estimation from the model. This means that the first term is responsible for performing the exploitation in the design space, meaning that it will push the algorithm to look for regions with minimum mean value estimation from the model; while the second term will try to find the region with the highest uncertainty estimates, meaning that it will explore the design space. This expression creates a balance between exploration and exploitation of the design space without requiring any external user input. This is why it is a widely used approach.

The discussion here is for applying the EI to a single-objective optimization problem. The following section extends this approach to handle multi-objective problems.

2.4.2 Expected Improvement for multi-objective problem

It is typical for a practical system to have multiple objectives to be optimized. In such scenarios, it becomes essential to define the formulation of Expected Improvement (EI) based acquisition

function in the context of multi-objective problems. Its extension can be categorized into two different categories, as explained next.

2.4.2.1 Using Achievement Scalarization Function

For performing surrogate-assisted optimization of multi-objective problems, the classical way is to combine objective functions using different weight vectors [34] and makes it multiple single-objective optimization problems. This is a well-known way to perform multi-objective optimization. There are various approaches available to combine the objective functions. For this work, the very similar approach proposed in [9], named M3-1, is used. Rather than using the weighted Tchebycheff method as shown in [34], M3-1 uses the augmented achievement scalarization function (AASF). The expression for AASF at input **x** is given by

$$AASF(\mathbf{x}) = \max_{i=1,2,\dots,m} \left(\frac{y_i(\mathbf{x}) - r_i}{w_i} \right) + \rho \sum_{i=1}^m \left(\frac{y_i(\mathbf{x}) - r_i}{w_i} \right)$$
(2.18)

where r_i and w_i are the *i*th elements of reference and weight vector (**r** and **w**), respectively, y_i is the *i*th element of objective function vector **y** (see equation (5.1)), and *m* is the total number of objectives. A set of reference vector **r** is generated using the method proposed by Das and Dennis in [6] with reference direction **w** making an equal angle to all objective axes. For each reference vector **r**, the AASF function is evaluated and then used to form the surrogate model and perform the optimization. For each surrogate model formulated using the AASF function, expected improvement, as explained before, is used for optimization.

2.4.2.2 Using Multi-Objective Improvement Function

Rather than using the classical way to perform multi-objective optimization using expected improvement, one can even define the improvement function in higher dimensions. Three different multi-objective improvement functions have been proposed in the literature: Euclidean distance, Maximin, and Hypervolume-based approaches. Rather than using the current best point (y_{min}) , these improvement functions use the current Pareto front (*P*) obtained from the observed points. The Pareto front is the optimal trade-off curve between different objectives y_i (i = 1, 2, ..., m). Assuming that the total number of points in the Pareto front is k, the improvement function in terms of Euclidean distance [33] can be written as

$$I_e(\mathbf{x}) = \min_{j=1,2,...,k} \sqrt{\sum_{i=1}^m (P_i^j - y_i(\mathbf{x}))}$$
(2.19)

where P_i^j is the *i*th objective value of the *j*th point in the Pareto front *P*. This expression is trying to find a point **x** with minimum euclidean distance from the Pareto front in the objective space.

Improvement function based on Maximin approach [67] is given by

$$I_M(\mathbf{x}) = -\max_{j=1,2,\dots,k} \left[\min_{i=1,2,\dots,m} (y_i(\mathbf{x}) - P_i^j) \right]$$
(2.20)

Rather than using the euclidean-based distance, this approach uses axis-wise distance to calculate the improvement function value. Finally, using Hypervolume criteria [12]. Hypervolume indicator H(P) calculates the volume of the region obtained by Pareto front P bounded by reference point r_z^* . The reference point is user-defined and dominated by all the points in the Pareto front. The expression for Hypervolume indicator is

$$H(P) = Volume\left\{ \mathbf{y} \in \mathbf{R}^m \mid P < \mathbf{y} < r_z^* \right\}$$
(2.21)

The improvement function using the Hypervolume indicator can be written as:

$$I_h(\mathbf{x}) = H(P \cup \mathbf{x}_{new}) - H(P) \tag{2.22}$$

where \mathbf{x}_{new} is a new point added to the current Pareto front. This expression is trying to find the improvement after adding a new point to the Pareto front.

Based on these improvement functions, the multi-objective expected improvement value can be calculated in any non-dominating region D using the following expression:

$$EI(\mathbf{x}) = \int_{y \in D} I(\mathbf{x}) \prod_{i=1}^{m} \frac{1}{s_i} \phi\left(\frac{y_i - \mu_i}{s_i}\right) dy_i$$
(2.23)

For this work, a hypervolume-based expected improvement approach is used for performing multi-objective optimization. The only drawback with this approach is its difficulty in computing

the expectation value. Zhan et al. [74] proposed a matrix-based fast method to approximately calculate the expected improvement for the multi-objective problem for all three above approaches. The closed-form expression for hypervolume based multi-objective EI is proposed as

$$EIM_{h}(\mathbf{x}) = \min_{j=1,2,\dots,k} \left[\prod_{i=1}^{m} (r_{z}^{*}(i) + EI_{i}^{j}(\mathbf{x}) - P_{i}^{j}) - \prod_{i=1}^{m} (r_{z}^{*}(i) - P_{i}^{j}) \right]$$
(2.24)

where $r_z^*(i)$ is the *i*th element of reference point vector \mathbf{r}_z^* , and *k* is the total number of points in the Pareto front.

The above discussions are all about finding the best point in the objective space. However, if the optimization problem is expensive to evaluate constraints, it becomes challenging to explore the design space efficiently. Without the efficient constraint handling mechanism, the algorithm will keep looking for optimal points in infeasible regions, thus wasting the evaluation budget. This will, in turn, use more evaluation budget than what ideally should. To deal with such a situation, the following subsection talks about constraint handling specifically for expected improvement.

2.4.3 Constraint handling in Expected Improvement

Any constraint function g(x) is given to be either greater than (\geq) or less than (\leq) a certain threshold value $(g_{threshold})$. It can easily be normalized to $\bar{g}(x) \geq 0$ to be used in the algorithm. After normalization, a feasible and infeasible constraint can be defined as follows:

$$Constraints = \begin{cases} Feasible, & \text{if } \bar{g}(\mathbf{x}) \ge 0\\ Infeasible, & \text{otherwise} \end{cases}$$

The normalized constrained function can then be used to form the response surface model. To handle constraints for the Expected Improvement (EI) based function, logic proposed by [16, 21] is used by defining the probability of feasibility as below

$$\mathbf{P}_{c}(\mathbf{x}) = \prod_{i=1}^{n} \Phi(\frac{\mu_{i}(\mathbf{x}) - 0}{s_{i}(\mathbf{x})})$$
(2.25)

where n is the total number of constraint functions. This expression is then multiplied with the acquisition function, and the constrained optimization problem is converted into an unconstrained

one. The final expression becomes

$$EI_c(\mathbf{x}) = EI(\mathbf{x}) \times \mathbf{P}_c(\mathbf{x})$$
(2.26)

and for multi-objective expected improvement matrix, the expression becomes

$$EIM_{c}(\mathbf{x}) = EIM(\mathbf{x}) \times \mathbf{P}_{c}(\mathbf{x})$$
(2.27)

Note that for these points violating constraints (i.e., $\bar{g}(\mathbf{x}) < 0$), the value of the expression \mathbf{P}_c will be small. And after multiplying it with the expression of expected improvement, it will make the overall expression small. During the maximization process, the smaller values of EI_c or EIM_c will eventually get eliminated. Thus these points violating constraint boundary will get handled efficiently.

Everything discussed above regarding the expected improvement is for the deterministic system. Implementing the approach for a stochastic system needs some modification, as explained next.

2.4.4 Expected Improvement in Stochastic environment

Both model development method and problem formulation discussed above are developed for problems with deterministic output measurements. To apply it to real systems, it is necessary to incorporate a noise handling mechanism in the system. Apart from modifying the model (discussed in Section 2.3.2), the acquisition function also needs modifications. Several extensions for the acquisition function have been proposed in [47, 24, 56, 57] for handling measurement noises.

The concept of extending the Expected Improvement approach for the stochastic environment is adapted from the work by Huang et al. . The proposed approach is called Augmented Expected Improvement (AEI), explained next. As can be seen from equation (2.17), this method tries to find the solution better than the current best value y_{min} . However, under the stochastic environment, the exact value of f_{min} is unknown. To deal with this situation, it is suggested to use mean value $\mu(\mathbf{x}^{**})$ from the model in place of y_{min} , where \mathbf{x}^{**} is known as the "effective best solution" determined by

$$\mathbf{x}^{**} = argmax[-\mu(\mathbf{x}) - cs(\mathbf{x})]$$
(2.28)

where c = 1 is proposed in [24]. This expression depends only on the model estimates. Therefore, to avoid additional computational effort, equation (2.28) is evaluated at points that are already evaluated. After including all the modifications, a new expression for Expected Improvement for stochastic system becomes

$$EI_{n}(\mathbf{x}) = (\mu(\mathbf{x}^{**}) - \mu(\mathbf{x}))\Phi\left(\frac{\mu(\mathbf{x}^{**}) - \mu(\mathbf{x})}{s(\mathbf{x})}\right) + s(\mathbf{x})\phi\left(\frac{\mu(\mathbf{x}^{**}) - \mu(\mathbf{x})}{s(\mathbf{x})}\right)$$
(2.29)

Because this is a stochastic process, repetition of function evaluation at the same point is allowed. This will enhance the model prediction at that point. However, with high noise variances, the algorithm can get stuck in a local optimum without searching for other regions. To counter this issue, reference [24] suggested a multiplication factor, which discourages the repetition of points after several function evaluations. With the multiplication factor, the final expression for Expected Improvement, known as Augmented Expected Improvement, can be written as

$$AEI(\mathbf{x}) = EI_n(\mathbf{x}).(1 - \frac{\tau(\mathbf{x})}{\sqrt{s^2(\mathbf{x}) + \tau^2(\mathbf{x})}})$$
(2.30)

Here, the second term in the expression is the multiplication term, where $\tau^2(\mathbf{x})$ is the estimation of noise variance at any unknown test point \mathbf{x} and $\mathbf{s}(\mathbf{x})$ is the model uncertainty estimate from the Kriging model. As repetition at a particular test point increases,

$$s^2(\mathbf{x}) \downarrow \implies$$
 multiplying factor $\downarrow \implies AEI(\mathbf{x}) \downarrow$

Since we are trying to maximize the augmented expected improvement (AEI) function, selection of the same point will get discouraged while performing optimization. Augmented Expected Improvement based approach was proposed for single-objective problems. But it can easily be extended for constrained multi-objective problems. Based on this analysis, two extensions are discussed in the next section.

Before moving further, one crucial thing to note here is the use of noise variance in evaluating the expression for $AEI(\mathbf{x})$. A system with a uniform noise variance all over the objective space will not have an issue implementing this strategy. Evaluating few points will give the value of noise variance, which can easily be implemented into the overall stochastic optimization approach.

However, in general, any practical system will have non-uniform (heteroscedastic) noise variance all over the design space, and getting the information at any unknown points is difficult. A way to overcome this hurdle is to develop a noise model separately and use that to estimate the noise variance at unknown points as explained in [2]. The intent to highlight this point is to discuss that for implementing the surrogate-assisted optimization approach to a practical problem, it is important to look for options where the information about noise variance is not needed beforehand so that the algorithm can efficiently deal with the practical system.

2.4.5 Multi-objective extensions for stochastic environment

Most of the work about extending the approach to handle stochastic systems is done for single objective unconstrained problems. Only a handful of work is done to perform multi-objective constrained optimization. The current work proposes two different extensions, motivated by the extension of expected improvement (EI) criterion for handling the multi-objective problem, as explained in Section 2.4.2.

2.4.5.1 Constrained Augmented Expected Improvement (CAEI)

As explained before, an intuitive way to extend the classical multi-objective expected improvement approach for stochastic measurements is to convert them to 'a' single-objective problems and then use the Augmented Expected Improvement (AEI) approach to solve the problem.

$$Multi - Objective \implies 'a'$$
 Single Objective

In this dissertation, the approach of using augmented achievement scalarization function (AASF) for combining the objective functions is used. After converting the problem, the obtained single objective problem is then solved using Augmented Expected Improvement (AEI) method discussed above.

The stochastic Kriging model considers the noise information while fitting the model and accordingly estimates the mean and uncertainty values at unknown points. Looking at the expression

for \mathbf{P}_c , it contains both mean and uncertainty estimates from the model. These estimates will contain the noise information. Therefore, for handing expensive constraint function, we do not need to change the expression of the probability of feasibility \mathbf{P}_c (equation 2.25). The expression of AEI(\mathbf{x}) with constraint handling becomes:

$$CAEI(\mathbf{x}) = AEI(\mathbf{x}) \times \mathbf{P}_{c}(\mathbf{x})$$
(2.31)

This expression handles constraints as well. Using this expression and evaluating the best point for each reference vector, the overall multi-objective problem is solved.

For stochastic problems, a similar approach is proposed in [18], where authors have used augmented Tchebycheff. In [75], an ϵ -constrained method was implemented. Our early work [51] has extended this approach for the stochastic case but in the uniform (homoscedastic) noise environment.

2.4.5.2 Augmented Expected Improvement Matrix (AEIM)

As for the second approach mentioned before, the hypervolume-based improvement function is selected for performing multi-objective expected improvement. For efficiently computing the multi-objective expected improvement function, the matrix method proposed in [74] is used. The authors have suggested the matrix form for three different approaches: Euclidean, maximin, and hypervolume-based. This work extends the hypervolume-based Expected Improvement Matrix (EIM) approach for handling the stochastic system with an AEI function. A similar extension could be made for the other two methods. The original expression for EIM is given below.

$$EIM_{h}(\mathbf{x}) = \min_{j=1}^{k} \left[\prod_{i=1}^{m} (r_{z}^{*}(i) + EI_{i}^{j}(\mathbf{x}) - P_{i}^{j}) - \prod_{i=1}^{m} (r_{z}^{*}(i) - P_{i}^{j}) \right]$$
(2.32)

The proposed extension for handling the stochastic system with Augmented Expected Improvement (AEI) is given by

$$AEIM_{h}(\mathbf{x}) = \min_{j=1}^{k} \left[\prod_{i=1}^{m} (r_{z}^{*}(i) + AEI_{i}^{j}(\mathbf{x}) - \mu_{i}^{j}(\mathbf{x}^{**})) - \prod_{i=1}^{m} (r_{z}^{*}(i) - \mu_{i}^{j}(\mathbf{x}^{**})) \right]$$
(2.33)

where \mathbf{r}_{z}^{*} is the user defined reference point and *k* is the total number of points in the Pareto front. It is well known that

$$AEI(\mathbf{x}) \ge 0 \implies AEIM_h(\mathbf{x}) \ge 0$$

When the algorithm gets stuck at local optimal point \mathbf{x} ,

$$AEI_i^j(\mathbf{x}) \rightarrow decreases \implies AEIM_h(\mathbf{x}) \rightarrow decreases$$

Thus, repetition of the same point would get discouraged during the optimization process to escape the local optimal region. For constraint handling, a similar approach is used as before. The expression is multiplied with the probability of feasibility (equation (2.25)), and the final expression can be written as below.

$$CAEIM_{h}(\mathbf{x}) = AEIM_{h}(\mathbf{x}) \times \mathbf{P}_{c}(\mathbf{x})$$
(2.34)

This concludes our discussion on extending the Expected Improvement-based approaches to deal with expensive constrained multi-objective optimization problems in both deterministic and stochastic environments. The next subsection discusses another acquisition function called Lower Confidence Bound (LCB) and its advantages/disadvantages in implementing it for solving practical problems.

2.4.6 Lower Confidence Bound

Current research work focuses on extending Lower Confidence Bound (LCB), which is driven by its easiness of extending the algorithm to handle multi-objective optimization problems. The LCB at any test point \mathbf{x} can be expressed as:

$$LCB(\mathbf{x}) = \mu(\mathbf{x}) - c \times s(\mathbf{x}) \tag{2.35}$$

where $\mu(\mathbf{x})$ and $s(\mathbf{x})$ represent the mean and standard deviation estimates from the model, respectively, and *c* is a constant. The mean value here is used to perform exploitation in the search space, whereas the standard deviation is used for exploration purposes. An effort on exploration is decided by the constant value *c* in the equation. The higher its value is, the more the acquisition function will try to explore. Since the LCB does not require the previous best value for performing search operation, it could be easily extended to handle multi-objective optimization problems. However, an important aspect here is constraint handling. It is crucial to identify the feasibility boundary to direct the search towards the optimal region in the feasible region. Otherwise, the algorithm would consume a significant search budget in infeasible regions for optimal solutions. An efficient constraint handling method is proposed for the LCB approach, as discussed next.

2.4.6.1 Proposed Constraint handling logic for LCB

Not much research is conducted in defining constraint handling functions for LCB based acquisition function. Since LCB could easily be extended for a multi-objective problem, defining an efficient constraint handling function would make the algorithm appealing for practical problems. For this research, a new constraint handling approach is proposed for optimization with the LCB function.

As explained before, the constraints $g_i(\mathbf{x})$ (i = 1, 2, ..., n) are normalized to $\bar{g}_i(\mathbf{x})$ (i = 1, 2, ..., n) before modeling, and any positive value is considered feasible and vice-versa. Since the constraint function is also modeled using Kriging, it provides uncertainty estimations. One can intelligently design a mechanism to explore the design space well and try to identify the feasible region as the algorithm progresses efficiently. Since the model is updated continuously as the algorithm proceeds, getting few infeasible points at the beginning could help developing constraint boundaries quickly. Therefore, by considering the exploration for the constraint space, a new constraint handling mechanism has been devised for this work.

Figure 2.4 explains the constraint handling logic. A one-dimensional test problem is plotted in Figure 2.4 to make it easy to understand, where horizontal *x*-axis represents input variation (design variable) and vertical *y*-axis represents the corresponding output (constraint) value. In the plot, blue curve represents mean value ($\mu(\mathbf{x})$) for the model and red curve for $\mu(\mathbf{x}) + s(\mathbf{x})$. If only the mean value were used for determining the infeasible region, these would be the regions marked with *a* and *b*. Note that this will be the case where standard deviation information available from



Figure 2.4: Explaination of constraint handling logic using test problem

constraint models is not utilized. If the standard deviation information is used, the region with $\mu(\mathbf{x}) + s(\mathbf{x}) \ge 0$ should be explored. This will expand the design space available for searching optimal solution, and the revised infeasible region becomes a_o and b_o . This is the logic used for constraint handling in this study. Note that for exploration, current work uses only one standard deviation $(s(\mathbf{x}))$ value. It can be modified based on the evaluation budget available.

For writing the proposed method mathematically, the logic used for handling constraints in Expected Improvement (EI) is used. As discussed in the literature, constraints are handled by defining feasibility function as

$$F(\mathbf{x}) = P_c(\bar{g}_i(\mathbf{x}) \ge 0) = \Phi(\frac{\mu_i(\mathbf{x}) - 0}{s_i(\mathbf{x})})$$
(2.36)

where Φ is the standard normal cumulative distribution function. Since constraint models are assumed to be independent, the net probability for *n* constraint functions can be written as

$$\mathbf{P}_{c}(\mathbf{x}) = \prod_{i=1}^{n} \Phi(\frac{\mu_{i}(\mathbf{x}) - 0}{s_{i}(\mathbf{x})})$$
(2.37)

The expression of \mathbf{P}_c is always greater than zero. Therefore, to differentiate the feasible and infeasible zones, one needs to subtract it with a fixed or adaptively varying quantity. Otherwise, all points will become feasible. The logic for choosing that quantity is explained as follows. Considering the case of n = 1, Figure 2.5 shows three different regions for \mathbf{P}_c .



Figure 2.5: Identification of different region for P_c

Region I: $\mu(\mathbf{x}) \ge 0$, $s(\mathbf{x}) \ge 0$ and $\mu(\mathbf{x}) + s(\mathbf{x}) > 0$

In this region, the constraint is always inside the feasible boundary since the mean value is positive.

Region II: $\mu(\mathbf{x}) < 0$, $s(\mathbf{x}) \ge 0$ and $\mu(\mathbf{x}) + s(\mathbf{x}) \ge 0$

This is the region of exploration, where the mean value is negative, but adding uncertainty to the mean could make this expression positive, meaning that if this point is explored, there is a chance of getting this point into the feasible region. Now the worst-case scenario would be $\mu(\mathbf{x}) + s(\mathbf{x}) = 0$ and in this case, expression for \mathbf{P}_c equals to $\Phi(-1)$. Overall, the value of \mathbf{P}_c is always greater than or equal to $\Phi(-1)$.

Region III: $\mu(\mathbf{x}) < 0$, $s(\mathbf{x}) \ge 0$ and $\mu(\mathbf{x}) + s(\mathbf{x}) < 0$

In this region, even utilizing the variance estimation could not make the given point feasible. This is clearly the case of point in the infeasible region. \mathbf{P}_c obtained in this case is less than $\Phi(-1)$.

The above discussions show that Regions I and II are feasible for the optimization algorithm, and Region III is not. After analyzing the complete system case-by-case, one can find the threshold value between feasible and infeasible regions equal to $\Phi(-1)$. After subtracting the threshold value from equation (2.37), the model for constraint handling with provision for exploration can be established. The final expression for the constraint function becomes

$$\mathbf{g}_{\mathcal{C}}(\mathbf{x}) = \prod_{i=1}^{n} \Phi(\frac{\mu_i(\mathbf{x})}{s_i(\mathbf{x})}) - (\Phi(-1))^n$$
(2.38)

Note the difference in handling the constraints for both expected improvement (EI) and lower confidence bound (LCB) criteria. For EI, the constrained problem becomes an unconstrained problem after multiplying the *probability of feasibility* function with the acquisition function, while for LCB, the constraints are handled separately. The problem stays as a constrained optimization problem.

2.4.7 Expected Improvement (EI) vs Lower Confidence Bound (LCB)

Because of the easiness of the Expected Improvement-based approach and no required user input for performing the exploration and exploitation, it became a popular approach in literature. However, it is myopic in nature, meaning it tends to perform the greedy search. Reference [60] has shown that the order of points in the sub-optimal region allocated by the expected improvement approach is of the order of $O(\log n)$, where *n* is the number of observations. This means that *n* should be sufficiently large to have points from the sub-optimal region, which is important for exploring design space.

In order to demonstrate this phenomenon, the same test problem defined before (2.14) is used but with different initial test points. Figure 2.6 shows the model developed using the initial test points, and it also shows the expected improvement curve (at the bottom), calculated throughout the design space to select the next evaluation point. Based on the expected improvement criteria, the algorithm will pick a point $\in [0.1 \ 0.2]$, as it has the maximum expected improvement value. The black dashed line represents the next evaluation point observed by maximizing the expected improvement acquisition function. It can be seen that the actual optimal region lies between input $\in [0.6 \ 0.9]$ but based on the expected improvement calculation, the algorithm is not suggesting the next evaluation point in that region. After evaluating the point suggested by expected improvement calculation for choosing the next point. The 2nd plot suggests selecting a point very close to the point from the previous iteration. This shows the greedy nature of the approach with less focus on exploration.

Therefore, to have a better exploration in the design space, the Lower Confidence Bound (LCB) approach is used to form the acquisition function, which is a non-myopic approach. It, however, requires external user input for setting the exploration constant (explained in Subsection 2.4.6). Figures 2.8 and 2.9 shows the points selected and model improvement using lower confidence bound acquisition function. Figure 2.8 shows the same initial model as before, but at this time, the next evaluation point suggested by the acquisition function is in the optimal region. This is the exploration part of the algorithm. Figure 2.9 shows the model after adding the observation suggested by LCB. As can be seen, the estimated model is converging towards the actual curve, and the black dashed line, showing the next evaluation point, is converging towards the optimal region.

The greedy nature of the expected improvement criterion is exposed due to the poor initial model generated. Referring back to the figure, it can be seen that the initial model did not have points between $x \in [0.4, 1]$. Therefore, the model could not capture the system behavior between these points, and the greedy nature of the expected improvement criteria did not suggest exploring the unknown region. However, we try to generate points uniformly throughout the design space for the overall surrogate-assisted optimization algorithm. This, therefore, helps to capture the model characteristics well in the entire design space. Therefore, for the deterministic system, expected improvement criteria have shown their capability in identifying the optimal region. However,



Figure 2.6: 1st Iteration with Expected Improvement acquisition function

with the stochastic system, the initial model could get affected due to noise, which might mislead the search for the optimal region. Therefore, for practical systems, it becomes crucial for the optimization algorithm to perform a non-myopic search to avoid local convergence.



Figure 2.7: 2nd Iteration with Expected Improvement acquisition function

Another major advantage of LCB based approach is noise handling. It does not explicitly require noise information feedback. This makes it attractive for its implementation for practical systems as the noise information is usually unknown beforehand. The EI-based criteria, on the



Figure 2.8: 1st Iteration with Lower Confidence Bound acquisition function



Figure 2.9: 2nd Iteration with Lower Confidence Bound acquisition function

other hand, required noise information beforehand for an effective search in the optimal region. If the system has uniform noise variance, it is easy to find out the noise level using few initial experiments, but this is generally not true. This study has performed an exhausting comparative study between lower confidence bound and expected improvement-based criteria for obtaining the optimal region of stochastic systems, which is to be discussed in the next chapter.

2.5 Optimization and selection of points for further evaluations

After formulating the acquisition function with a constraint handling mechanism, the next task is to perform its optimization to find a set of optimal points, which will be used for performing high-fidelity evaluations. Various methods are available to perform the optimization, from classical gradient-based approaches to evolutionary approaches.

Gradient descent is the most commonly used algorithm for performing optimization because of its ease of implementation. However, it depends on the steps size and starting point. It has the probability of getting stuck at the local optima. The evolutionary optimization approach has been shown to work well in identifying the global optimal solution. Several algorithms come under the umbrella of evolutionary optimization. The most popular ones are Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and evolutionary programming (EP). For this work, the Genetic Algorithm is used for performing the optimization of the acquisition function.

2.5.1 Genetic Algorithm

A Genetic Algorithm (GA) is a population-based approach, which starts with a set of data points called population. The algorithm then progresses with performing selection, crossover, and mutation operations at each iteration based on the function evaluations to keep the best possible solution while simultaneously exploring the design space. The approach is shown to work well in identifying the global solution for both convex and non-convex problems with fast convergence.

For both expected improvement-based approaches, constrained multi-objective problem is converted to an unconstrained single objective problem (unified acquisition function is developed). Therefore, a real parameter Genetic Algorithm *RGA* [35] is used for performing the optimization. With the lower confidence bound, the acquisition function is developed for each objective. Since the current work deals with a two-objective problem, a non-dominated sorting algorithm (NSGA-II) [32] is used for performing its optimization.

With *RGA*, every iteration for optimization gives one final solution at the end, which can then be evaluated on the high-fidelity system (either actual experimental setup or simulation). However, with multi-objective optimization (case with LCB), the NSGA-II algorithm generates a set of optimal solutions, called Pareto front. It is a trade-off curve among different objectives, the size of which depends on the population size of the optimization algorithm. All points from the Pareto front can be evaluated using a high-fidelity system, but doing so will exhaust the total evaluation budget and reduce the iterations for the learning algorithm. Using fewer points will not update the model well and might delay the convergence. One of the possible ways to deal with this trade-off situation is to perform clustering and select the centroid of each cluster for further evaluations to improve the model accuracy. Various methods are available for cluster formation, namely divided into supervised and unsupervised categories. For this work, two specific methods are implemented as discussed, namely, *k-means clustering* and *dbscan*.

2.5.2 K-Means Clustering

K-Means clustering algorithm divides the total number of points into 'K' different clusters, based on the euclidean distance between the centroid of clusters and observation points [40]. The only user input required for implementing this approach is the number of clusters 'K'. This approach has both advantages and disadvantages. It is easy to implement but struggles in the presence of outliers. It also requires the total cluster number beforehand, which is usually not known.

For this work, this approach is implemented in the objective space to identify a few points from the Pareto front observed after performing optimization of the acquisition function. The uniform spread of points from the Pareto front will ensure the distribution of the optimal solution.

2.5.3 DBSCAN

Density-based spatial clustering of application with noise or, in short, DBSCAN is a density-based clustering approach [13]. As the name suggests, it forms different clusters based on the density of the spread of points. This approach does not require defining the total number of clusters beforehand. It, however, requires other user inputs such as maximum distance between two points (r_{max}) to be considered as neighbors and Minimum Number of Points (MNP) required to form a dense region. r_{max} is a crucial parameter to identify the cluster numbers correctly. For the current work, an adaptive strategy is designed to calculate the value of r_{max} for each iteration of the learning algorithm.

For the minimum number of points (MNP), it is recommended that the value should be $\ge N+1$, where N is the total number of input variables. Another advantage of DBSCAN is handling the outliers, and it provides a set of points termed noise, which can be excluded for further evaluation.

2.6 Chapter Summary

This chapter explained the entire surrogate-assisted optimization approach for solving the engine calibration problem along with modifications required to implement it on to the practical systems. The following conclusions can be made out of this chapter:

- 1) Different steps involved in performing the surrogate-assisted optimization approach are explained with the emphasis on the steps required to modify for its practical implementation.
- Both deterministic and stochastic Kriging models are discussed and showed the advantage of using stochastic Kriging model for a system with measurement noise.
- 3) The current work uses two widely used acquisition functions: Expected Improvement (EI) and Lower Confidence Bound (LCB), for performing the surrogate-assisted optimization. This chapter also briefly discussed a possible greedy nature of the EI-based approach and how it could affect the convergence if implemented to a system with measurement noise.

- 4) Since the lower confidence bound (LCB) approach did not have an efficient constraint handling mechanism, a new constrained handling algorithm is proposed.
- 5) In order to implement the approach for performing optimization of a practical system, three different approaches are proposed for handling constrained multi-objective problems with measurement noise. Specifically, non-uniform (Heteroscedastic) noise addition is considered to make the approach implementable for any practical system.

CHAPTER 3

SIMULATION VALIDATION OF PROPOSED ALGORITHMS

Based on all the methods discussed in the previous chapter for handling the constrained multiobjective problem both in deterministic and stochastic environments, this chapter validates these results for the engine calibration problem. Before moving to the actual experimental study, the first necessary step is to validate the proposed approaches in the simulation environment. Therefore, a high-fidelity engine model developed using GT-SUITETM is used to perform the engine calibration study. Based on the results obtained from this chapter, the experimental work is planned and will be discussed in detail in the next two chapters.

This chapter is divided into two major sections. The first section discusses the validation of the constraint handling algorithm for the Lower Confidence Bound (LCB) criterion. Because the Expected Improvement criterion with constrained handling has already been validated extensively in the literature, it is not discussed here. After the constrained handling validation, the proposed algorithm extensions for stochastic systems are validated with different noise levels. For this part, both multi-objective Expected Improvement and Lower confidence bound-based criteria are demonstrated. Algorithm validation is first performed on the standard numerical test problems, where the Pareto front is well known. This is done to check the algorithm's efficacy in detecting the points in the global optimal region. Later, it is implemented for the engine calibration problem. For each section mentioned before, validation on both multiple numerical test problems and the actual engine calibration problem is shown for comprehensive testing of the proposed algorithm. Before showing the results, this chapter will explain the problem setup used for performing surrogateassisted optimization.

3.1 Problems for Validating the Algorithms

Any practical problem, in general, contains various objectives to be optimized. Moreover, with the increased system complexity, there are multiple control parameters needed to be calibrated. The

| Problem | No. of Objectives | No. of Constraints | No. of Design Variables |
|---------|-------------------|--------------------|-------------------------|
| ZDT1 | 2 | 0 | 8 |
| ZDT2 | 2 | 0 | 8 |
| ZDT3 | 2 | 0 | 8 |
| BNH | 2 | 2 | 2 |
| SRN | 2 | 2 | 2 |
| TNK | 2 | 2 | 2 |
| OSY | 2 | 6 | 6 |

Table 3.1: Numerical test problems

current work is, therefore, validated on several constrained multi-objective optimization problems. Initially, the algorithms are tested on several constrained and unconstrained multi-objective numerical test problems explicitly defined for validating the efficacy of newly developed algorithms. Later, a constrained multi-objective engine calibration problem is formulated.

3.1.1 Numerical Test Problems

Several standard multi-objective numerical test problems, both unconstrained and constrained, with different types of optimal Pareto fronts, are available in evolutionary optimization-related literature [8]. The advantage of using these standard test problems is that the actual optimal solution is known, which could be used to check if the proposed algorithm converges to the global optima within a reasonable number of iterations.

Out of the entire set, seven different problems, shown in Table 3.1, are used for the validation purpose. The first three problems in Table 3.1 are unconstrained multi-objective optimization problems. These problems are not useful for validating the constraint handling approach. However, they are useful to detect the convergence for unconstrained problems using the stochastic extension of the multi-objective approach. The last four problems in the table are multi-objective problems with constraints. These problems are useful for validating both the proposed constraint handling approach and the algorithms with stochastic extension.

After implementing the algorithm for these test problems, it is important to assess its performance of converging to the true Pareto front. For multi-objective problems, the performance is measured in terms of how close the identified optimal curve is to the true front (**Convergence**) and how good is the spread of points covering the actual Pareto front. Several performance assessments are available in literature [8] including Generational Distance (*GD*) [71], Inverted Generational Distance (*IGD*) [5], Hypervolume [78], to name a few. *IGD* is the most commonly used assessment criterion as it not only identifies the closeness of obtained optimal front from the actual Pareto front but also shows the distribution over the actual Pareto front.

Calculation of IGD is the same as that of GD, but with a reversed role of the actual Pareto front and the near optimal front obtained by the proposed algorithm. With P and Q defining the actual Pareto front and the obtained optimal curve, respectively, the expression for IGD is defined by

$$IGD(P,Q) = \frac{1}{|P|} \Big(\sum_{i=1}^{|P|} d_i^a \Big)^{1/a}$$
(3.1)

where d_i is the minimum distance between the i^{th} point in the actual Pareto front and the front obtained by the algorithm (Q). Using a = 2, distance d_i is euclidean distance that can be calculated by

$$d_{i} = min_{k=1}^{|Q|} \sqrt{\sum_{j=1}^{m} ((y*)_{j}^{i} - (y)_{j}^{k})^{2}}$$

where *m* is the dimension of objective function, $(y*)_j^i$ is the j^{th} objective function value of the i^{th} point from the actual Pareto front *P* and $(y)_j^k$ is the j^{th} objective function value of the k^{th} point from the obtained optimal curve (*Q*) for calculating the distance.

3.1.2 Engine Calibration Problem

Fuel economy and emissions are two major criteria for analyzing key engine performances. The goal is always to maximize the performance while simultaneously achieving satisfactory emissions. A generic engine calibration problem could have multiple objectives and constraints with several control parameters for calibration. Therefore, with the target of optimizing both the engine fuel

economy and emissions, a multi-objective engine calibration problem is formulated as below.

Minimize
$$y_1(\mathbf{x}) = BSFC(\mathbf{x}), \ y_2(\mathbf{x}) = NO_x(\mathbf{x})$$
subject to $g_1(\mathbf{x}) = Boost(\mathbf{x}) \le 2 \ bar$, $g_2(\mathbf{x}) = Load(\mathbf{x}) \ge 9.95 \ Nm$,Control Variables $\mathbf{x} = [VGT, \ EGR, \ SOI]^T$ $x_1 = VGT \in [0\ 1]\ \%$, $x_2 = EGR \in [0\ 1]\ degree$, $x_3 = SOI \in [0\ 1]\ bTDC$

The main objective of the engine calibration problem is to find the optimal trade-off between Brake Specific Fuel Consumption (BSFC) and NO_x emissions, which provides a typical curve between engine efficiency and its emissions. Therefore, this is a two-objective optimization problem (m = 2) with $y_1(x) = BSFC$ and $y_2(x) = NO_x$. Reducing fuel consumption (i.e., improving engine efficiency) is known to increase NO_X emissions and vice-versa. Therefore, the optimal setting for these two objective functions provides a trade-off relationship known as the Pareto front. For performing engine calibration, some constraints are applied to the engine boost pressure $(g_1(x))$ and load $(g_2(x))$ to operate the engine within the desired boundary. Hence, n = 2 here. Three control parameters are selected to perform calibration, namely VGT vane position (x_1) , EGR valve position (x_2), and fuel injection timing (SOI) (x_3). Therefore, with p = 3 and **x** is a three-element vector. The problem formulation shows the normalized variation range for these control parameters. All three control parameters affect the combustion characteristics, which, in turn, affect both BSFC and NO_x emissions. Increasing the EGR amount reduces NO_x emissions but adversely increases the BSFC. Optimizing engine boost by regulating VGT vane position for improved BSFC could lead to increased NO_x emissions. Injection timing also alters both combustion efficiency and NO_x emissions. This work is to find the optimal combination of all three control parameters.

This study uses a simulation model of Ford 6.7L 8-cylinder turbocharged Scorpion diesel engine to perform the calibration study. For the diesel engine, a high fidelity GT-SUITE^{TM} model integrated with a Simulink-based emission model provided by Ford Motor Company is used. The

GT-SUITETM model could run either at a constant speed or load mode. For this study, the model is run at a constant speed mode. For the current work, the steady-state optimization is performed at a fixed engine speed of 2000 RPM and a load of 10 bar. Changing the control variables affects the engine performance. Therefore, a constraint is added to ensure that the load condition is satisfied. Since the original constraint is an equality, the limit is relaxed by 0.5%, that is, BMEP (Brake Mean Effective Pressure) constraint is greater than or equal to 9.95 bar for a target at 10 bar. The closed-loop control strategy in the high fidelity GT-SUITETM model guarantees that the BMEP should always be less than 10 bar.

3.1.2.1 Model Validation and its Implementation

The engine behavior modeled by the GT-SUITETM model is assumed to be accurate, i.e., there is no discrepancy between the experimental and simulation results at the desired engine operational condition. To validate this hypothesis, steady-state model validation is performed at the desired engine speed of 2000 rpm. Figure 3.1 shows these plots for BSFC and NO_x emissions. As can be seen from Figure 3.1, both GT-SUITETM and the emission model shows a good correlation with the experimental data at low load (BMEP ≤ 10 bar). The difference becomes significant at high engine load conditions. Therefore, the current work considers the 10 bar load condition.

The surrogate-assisted optimization algorithm is programmed in MATLAB. Therefore, the engine model is integrated with the MATLAB interface to perform the simulation study. Figure 3.2 shows the implementation of the proposed optimization strategy for the engine calibration problem in a simulation environment. Because we are using the model here, no measurement noise is present. Here, the control block runs the proposed optimization strategy and provides a set of possible calibration variables to be evaluated next. These are then passed on to the plant model, a combination of the GT-SUITETM and Simulink model, to get all desired measurement variables. Before performing the optimization, all the desired variables are set in the control strategy block according to the problem formulation discussed before.



Figure 3.1: GT-Power and emission model validation using experimental data



Figure 3.2: Algorithm implementation for engine calibration

3.1.2.2 Baseline Engine Calibration Result using GT-SUITETM Model

The optimal solutions for the numerical test problems are well defined beforehand. However, for the current engine calibration problem, the actual optimal solution is not known. It is required to have the knowledge of the Pareto front to compare the performance of different algorithms. Therefore, a baseline study is done using the available engine model, where the high-fidelity model is directly integrated with any optimization approach to get the near-optimal Pareto front.



Baseline Result: BSFC vs NO_x

Figure 3.3: Trade-off curve between BSFC and NO_x from the baseline study

For the baseline study, the engine model (GT-SUITETM for performance and Simulink model for emission measurement) was integrated with the evolutionary optimization approach (NSGA-II) for performing the calibration. Parameter values used for the NSGA-II algorithm are 40 for the population size, 100 for the total generation number, 0.9 for the crossover probability, 1/3 for the mutation probability, 15 for the distribution index used for the SBX operator, and 20 for polynomial mutation. A total of 4000 high fidelity evaluations were performed in order to obtain the near-optimal Pareto front. Figure 3.3 shows the near-optimal Pareto front from the high fidelity GT-SUITETM model evaluation. This result is used as a reference optimal Pareto front to check the goodness of proposed algorithms.



Figure 3.4: Optimal control parameter values from the baseline study
As expected, there is a trade-off relationship between engine performance (BSFC) and NO_x emissions. The optimization results in terms of the Pareto front provide important information. As shown in Figure 3.3, the Pareto front is divided into three different regions; blue region (Region 1) for small change in BSFC, black region (Region 3) for small change in NO_x , and red region (Region 2) provides significant trade-off. Both blue and black regions are highly sensitive. A small change in BSFC causes a significant change in NO_x emissions and vice versa. This indicates a narrow zone for calibration parameters to obtain the optimal solution. The rest of the Pareto front (red region) is not very sensitive. The corresponding points in the design space are shown in Figure 3.4.

The surrogate-assisted optimization approach works by directing the search algorithm to look for optimal solutions based on the developed model. It is difficult to have a good model with limited data points. Thus, these narrower optimal regions could not get enough attention as the focus is mainly on balancing exploration and exploitation. Based on this hypothesis, it is expected that the proposed algorithm would most likely find multiple optimal solutions in the red region, and getting solutions in the other two regions would be difficult. The convergence might get even worse when testing the stochastic algorithms, where the measurement noise will be added externally to the model outputs to simulate the actual engine behavior.

3.2 Validation of Constraint Handling Algorithm

This section focuses on validating the constraint handling algorithm. Initially, the algorithm is validated on numerical test problems to check its efficacy and later in the simulation environment for the engine calibration problem.

3.2.1 Validation on numerical problems

For the numerical test problems, NSGA-II was used to optimize the acquisition function with a population size of 100 and 200 generations. Crossover probability equals to 0.9 and mutation probability equals to 1/N, where N is *total number of variables*. The distribution index for crossover (SBX) operator equals to 15; polynomial mutation equals to 20, and k = 10 clusters were used for the *k-mean clustering* algorithm. Inverted generational distance (IGD) is computed for comparing the actual Pareto set with the optimal solution obtained from surrogate-assisted optimization. 20 points from the true Pareto set is used to calculate IGD for both test problem and actual engine calibration problem.

All the test problems, shown in Table 3.1, are used to validate the proposed constrained handling approach for Lower Confidence Bound (LCB). LCB was used as an acquisition function for every test problem with a constant value of c = 2 for all objectives. Kriging models were initially developed using 100 data points with a second-order polynomial as a regression function. For each test problem, the total high fidelity computational budget was fixed at 300. Each problem was run 30 times to validate the repeatability of the proposed algorithm. All these parameters were the same for both constrained and unconstrained test problems. With these algorithm parameters, surrogate-assisted optimization was performed.

3.2.1.1 Unconstrained Problems

In order to perform tests on unconstrained problems, three standard unconstrained test problems were selected, namely ZDT1, ZDT2, and ZDT3. Original problems have a total of 30 design variables. Since the Kriging model suffers from the curse of dimensionality, the design variables for all unconstrained test problems were reduced from 30 to 8.

Optimization results for the three problems are shown in Figure 3.5. It can be seen that within 300 high fidelity evaluations, the proposed algorithm is able to find multiple points near the actual Pareto optimal set. Note that a conventional evolutionary optimization algorithm would take thousands of iterations to find the optimal solution. Also, the proposed algorithm is not dependent on the type of Pareto set. Even though the Pareto set for ZDT3 is discontinuous and Pareto sets for ZDT1 and ZDT2 are convex and concave, respectively, the algorithm was able to get the near-optimal solution within a limited budget. Table 3.2 shows both means and standard deviations of the IGD values indicating a convergent optimization process. It can be observed from Table 3.2 that the standard deviations of all unconstrained problems are small, indicating good



Figure 3.5: Results of the lower confidence bound (LCB) based optimization algorithm on unconstrained test problems

convergence. Since these are unconstrained problems, it was expected to achieve similar results since the acquisition function used here is pretty standard. The key is to test the proposed algorithm for constrained problems discussed next.

Table 3.2: IGD values for both unconstrained and constrained test problems

| | ZDT1 | ZDT2 | ZDT3 | BNH | SRN | TNK | OSY |
|---------|--------|--------|--------|--------|--------|--------|--------|
| Average | 0.0121 | 0.0091 | 0.0184 | 0.4599 | 1.0486 | 0.0311 | 0.8962 |
| SD | 0.0019 | 0.0014 | 0.0038 | 0.0719 | 0.229 | 0.0169 | 0.1476 |

3.2.1.2 Constrained Problems

One of the main contributions of this work is the proposed constraint handling method. Four different test problems were selected, namely BNH, SRN, TNK, and OSY, to study the effectiveness of the proposed algorithm. True Pareto fronts of these problems are either continuous or discontinuous. Total high fidelity evaluations are kept the same as before at 300. Figure 3.6 shows the optimization results. Figure 3.6 shows that with only 300 true evaluations, the proposed constraint handling method was able to find multiple points near the true Pareto optimal front.



Figure 3.6: Results of the lower confidence bound (LCB) based optimization algorithm with proposed constraint handling approach on constrained test problems

Table 3.2 shows the IGD values for these problems. As can be seen, the average values are pretty low with small standard deviations, showing good repeatability. With the promising results obtained

from the proposed algorithm for test problems, the next section discusses its implementation for the engine calibration problem.

3.2.2 Validation on Engine Calibration Problem

To develop the initial model for performing the surrogate-assisted optimization, 100 data points are generated in the design space using the latin hypercube sampling (LHS) based DOE (Design of Experiment) approach. A total of 500 high fidelity evaluations are set as the computational budget. In the LCB acquisition function, a constant of c = 2 is used, and the revised constraint function is developed according to the proposed constraint handling method. Figure 3.7 compares results between baseline Pareto front and approximate Pareto front obtained from the proposed algorithm. To perform a comparative study with the existing state-of-the-art methods in the literature, the proposed method is compared with two other methods: Multi-objective Expected improvement approach using expected improvement matrix (EIM) and M3-1 approach. Figures 3.8 and 3.9 shows the results from EIM and M3-1, respectively.

From Figures 3.7, 3.8 and 3.9, one can observe that all three surrogate-assisted optimization algorithms work well with a proper distribution of optimal points near the actual optimal Pareto front. With a limited budget, all three algorithms improve performance with more than 85% reduction in the computational budget compared with the evaluation needed to perform the baseline study. In order to study the spread of points, IGD values are calculated and presented in Table 3.3. For the repeatability study, all three algorithms were run six times, and Table 3.3 shows the mean and standard deviation of IGD values from these simulations.

Table 3.3: IGD values for LCB, EIM, and M3-1

| | LCB | EIM | M3-1 |
|---------|------|------|------|
| Average | 5.11 | 3.66 | 6.95 |
| SD | 1.23 | 0.85 | 0.88 |



Figure 3.7: Engine calibration optimization using proposed algorithm

With a total of 500 evaluations, all three methods performed well with M3-1, the worst one based on the IGD values shown. The multi-objective expected improvement (EIM) approach seems to be the best, with a good distribution of points throughout the Pareto front. Our proposed method (LCB) also demonstrates a good distribution of points over the Pareto front. An interesting observation from Figures 3.7 and 3.8 and 3.9 is the spread of Pareto front points. As can be seen, the majority of these points lie in Region II, with very few in Regions I and III, which justifies the hypothesis mentioned before. For all three algorithms, the majority of points lie in Region II. The overall spread of optimal points is better for LCB and EIM and the worst for M3-1. Our proposed constrained LCB algorithm has a tuning parameter ($\Phi(-1)$) in the constraint handling function, which controls the exploration in design space. Exploring the design space to identify a few infeasible points helps in defining the feasible boundary. However, towards the end of the evaluation budget, exploration is unnecessary, and we could set the limit of tuning parameters to



Figure 3.8: Engine calibration optimization using EIM method

be more conservative in identifying new points. An adaptive constraint handling mechanism could be developed for enhanced convergence, but the current study does not consider it.

A good spread of points throughout the Pareto front is vital to obtain rough estimates of the calibration parameter range, which will give optimal engine performance. It gives the flexibility to choose the particular parameter setting, depending on optimization priority, to reduce either BSFC or NO_x emissions at any operating condition. This is particularly important in the engine calibration problem requiring minimal BSFC with satisfactory NO_x emission constraint. The primary motivation of this section is to validate the hypothesis that the proposed algorithm (Constrained LCB) is able to obtain the global optimal solution for the engine calibration problem with reduced cost in terms of computational effort. As can be observed, with the proposed algorithm, it is possible to get the near-optimal solutions with just 500 function evaluations. Compared with 4000 high-fidelity evaluations using an evolutionary optimization approach, this reduces the computational cost by



Figure 3.9: Engine calibration optimization using M3-1 method

more than 85%.

The main advantage of the proposed optimization approach (with constrained Lower Confidence Bound) is the ease of extending the algorithm for handling multiple objective values with reduced computational effort. Also, for implementing the algorithm in a stochastic environment, the current method does not require any modification as it only works by using the model information, making it less sensitive to the stochastic environment. Because the EI approach uses the current best value for search, it needs to be modified to perform optimization with measurement noise (discussed in Chapter 2). This next section addresses the stochastic results.

3.3 Validation of Stochastic Multi-Objective Extensions

This section initially demonstrates the performance of proposed algorithms for test problems, and the second part discusses its application to the engine calibration problem to demonstrate its applicability for practical systems. The problem formulation for EI-based approaches converts constrained multi-objective optimization problems to unconstrained single-objective problems. Therefore, to optimize the acquisition function, a real variable genetic algorithm (rGA) is used. For LCB based approach, NSGA-II is used. For both optimization methods (rGA and NSGA-II), the population size of 100 and a total of 200 generations are used for both test problems and the engine calibration problem. Other parameters of the algorithm: crossover probability, mutation probability, distribution index for crossover (SBX) operator, and polynomial mutation are also fixed at 0.9, $\frac{1}{N}$ (N = total number of variables), 15, and 20, respectively. For the NSGA-II approach (used in CLCB), *k-mean clustering* algorithm with k = 10 is used to select points for further high fidelity evaluations.

All three algorithms are run until the total evaluation budget is exhausted. After that, nondominated sorting is performed on all high fidelity evaluated points. For comparing the performance of these final solutions obtained, the IGD value is calculated using 20 points from the actual Pareto front.

The output generated from the simulation model is deterministic. Therefore, it is required to add external noise to the measurement to simulate the behavior of a practical system. The stochastic Kriging model explained in Chapter 2 assumes Gaussian noise addition to make the calculation simpler. Note that any distribution can be transformed to a Gaussian distribution if its structure is known. Because this approach is extended for the engine calibration problem, the experiments are performed to identify the noise distribution for the measurement signals. Figure 3.10 shows the distribution plots of BSFC and NO_x measurements using 200 data points. As can be seen from the figure, both distributions are Gaussian. Therefore, a Gaussian noise can be added for the engine model to make it stochastic to represent the actual engine measurements.

3.3.1 Heteroscedastic noise addition to the simulation

In general, at fixed control input values, the engine output always fluctuates within a range depending on the engine operating condition. The further the engine runs from its optimal zone, the higher



Figure 3.10: Distribution of BSFC and NO_x measurements

the combustion instability and the measurement variances. Engine performance becomes very stable near its optimal region. With this behavior in mind, total function variation at any point is

considered to be linearly dependent as:

$$n = a \times f(x) + b \tag{3.2}$$

where *a* and *b* are constants selected based on the problem, and f(x) is the function mean value. In the optimal region, f(x) will be minimum and thus with low fluctuations, and vice-versa for other regions. Considering that the output variation follows Gaussian distribution with $\pm 3\sigma$ from its mean value f(x), the standard deviation of noise becomes

$$\tau = \frac{n}{6} \tag{3.3}$$

For the current application, b = 0 and a = 5%, 10%, 20%, and 40% for different noise levels. Note that under any actual engine operational condition, it is almost impossible to have variations in *BSFC* and *NO_x* beyond 20%, and therefore, 40% noise level is only considered for performance assessment of proposed algorithms on numerical test problems, shown in Table 3.1. This is done to check the algorithm's applicability to handle problems with similar structures but higher noise levels.

Usually, for stochastic problems, it is required to perform multiple function evaluations at a given point to get both mean and noise variance. However, with an engine operated at a given condition, the measurement variable fluctuates each engine cycle. Therefore, a batch of data could be collected to get mean and variance estimations in a single function evaluation. To create a similar scenario, the output value at any location is first corrupted with external noise, and a total of 100 samples are generated. Using these 100 random samples under that operating point, the mean and standard deviation values are calculated. This eliminates the need to perform *replication step* discussed in the literature of stochastic optimization.

3.3.2 Validation on Numerical Problems

This section deals with different numerical test problems to check the efficacy of proposed algorithms. Both unconstrained and constrained problems with continuous and discontinuous Pareto front are selected. For all three algorithms, the Kriging metamodel was initially developed using a 100-point training data set. The total evaluation budget is kept fixed at 500 for all the problems. For EI-based approaches (CAEI and AEIM), external user input is not required, but for CLCB, c = 2 is used in the expression of the acquisition function. Each problem is run 10 times to test the repeatability of algorithms. All these parameters were kept the same for both constrained and unconstrained test problems.

3.3.2.1 Unconstrained Test Problems

As mentioned previously, the total number of design variables for all three problems (ZDT1, ZDT2, and ZDT3) are reduced from 30 down to 8. For all three algorithms, simulations are performed at four different noise levels: 5%, 10%, 20%, and 40%. Figures 3.11 and 3.12 only show the results at 10% noise level, where Table 3.4 presents algorithm convergence property in terms of IGD at all noise levels. Results at all other noise levels are shown in Appendix A.

Both Figures 3.11 and 3.12 show plots for the best results out of all the evaluations for each problem. For all the problems, the CLCB algorithm is able to get a well-distributed solution near the actual Pareto front. This can also be observed from Table 3.4. For ZDT1, all three algorithms performed well, with LCB outperforming the other two by a small margin. This gap becomes prominent for the other two problems, ZDT2 and ZDT3. These algorithm performances show similar trends at all noise levels. One can also observe the robustness of algorithms from Table 3.4 with a smaller deviation for all noise levels.



Figure 3.11: Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM and c) CLCB on ZDT1 and ZDT2 with 10% noise



Figure 3.12: Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM and c) CLCB on ZDT3 with 10% noise

| | | Noise = 5% |) | | Noise = 10% | lo | | Noise $= 20\%$ | lo | Ì | Noise = 40° | lo | |
|---------------|---------|------------|---------|---------|----------------|---------|---------|----------------|---------|---------|--------------------|---------|------|
| Algorithm | #1 | # 2 | #3 | #1 | # 2 | #3 | #1 | # 2 | #3 | #1 | # 2 | #3 | • |
| Unconstrained | | | | | | | | | | | | | |
| 7071 | 0.0259 | 0.0215 | 0.0126 | 0.0249 | 0.0201 | 0.0104 | 0.0265 | 0.0238 | 0.0127 | 0.0299 | 0.0206 | 0.0126 | mean |
| | 0.0049 | 0.0040 | 0.0034 | 0.0027 | 0.0039 | 0.0023 | 0.0044 | 0.0088 | 0.0043 | 0.0042 | 0.0072 | 0.0041 | std. |
| 7072 | 0.0328 | 0.0430 | 0.0119 | 0.0318 | 0.0343 | 0.0127 | 0.0253 | 0.0519 | 0.0132 | 0.0611 | 0.0483 | 0.0162 | mean |
| ZDI 2 | 0.0069 | 0.0095 | 0.0018 | 0.0037 | 0.0102 | 0.0018 | 0.0042 | 0.0117 | 0.0032 | 0.0103 | 0.0051 | 0.0039 | std. |
| 7.0.72 | 0.0618 | 0.0467 | 0.0166 | 0.0746 | 0.0284 | 0.0144 | 0.0789 | 0.0207 | 0.0187 | 0.0791 | 0.0185 | 0.0182 | mean |
| ZDI 3 | 0.0077 | 0.0068 | 0.0043 | 0.0125 | 0.0035 | 0.0022 | 0.0085 | 0.0043 | 0.0029 | 0.0082 | 0.0065 | 0.0022 | std. |
| Constrained | | | | | | | | | | | | | |
| RNH | 1.5227 | 0.1439 | 0.3647 | 1.4495 | 0.1850 | 0.4189 | 1.3365 | 0.2436 | 0.4831 | 1.1818 | 0.3450 | 0.5486 | mean |
| Divil | 0.2141 | 0.0358 | 0.0901 | 0.0912 | 0.0769 | 0.0799 | 0.2123 | 0.1217 | 0.1219 | 0.2062 | 0.1161 | 0.1453 | std. |
| OSV | 10.0901 | 14.1855 | 18.1762 | 11.8761 | 22.0622 | 19.0537 | 12.5085 | 20.3393 | 19.8792 | 15.1228 | 23.0618 | 22.7355 | mean |
| 031 | 1.5400 | 17.3693 | 1.5659 | 2.8290 | 20.1817 | 1.1517 | 2.7517 | 13.2482 | 1.5108 | 2.9568 | 11.6228 | 2.0024 | std. |
| CDN | 7.8818 | 1.7984 | 0.7541 | 9.3570 | 2.8186 | 0.8199 | 8.0319 | 2.4688 | 0.8988 | 9.3222 | 3.1552 | 1.1973 | mean |
| SKN | 1.6377 | 1.9449 | 0.1006 | 1.6972 | 2.4666 | 0.1854 | 1.1338 | 2.4228 | 0.1711 | 1.3132 | 3.2074 | 0.2122 | std. |
| TNF | 0.0754 | 0.0137 | 0.0403 | 0.1089 | 0.0155 | 0.0477 | 0.1219 | 0.0620 | 0.0536 | 0.1477 | 0.0718 | 0.0589 | mean |
| 1 1 V K | 0.0184 | 0.0029 | 0.0074 | 0.0184 | 0.0051 | 0.0079 | 0.0291 | 0.0067 | 0.0089 | 0.0325 | 0.0113 | 0.0103 | std. |

Table 3.4: IGD values for all test problems using algorithms #1 : *CAEI*, #2 : *AEIM* and #3 : *CLCB*

As evident from the simulation results, EI-based algorithms performed worse than LCB based ones. It is well known that the search from EI based approach is myopic in nature [58, 64, 60]. LCB based acquisition function, however, focuses more on exploration and performs a non-myopic search.

The myopic nature of the EI-based algorithms is not an issue with the deterministic system as the model development could represent the actual response surface well. However, in the presence of measurement noise, the response surface generated by the stochastic Kriging method could mislead the search. Therefore, the exploratory nature of the algorithm helps explore the design space better and, thus, captures the response surface well. This makes the LCB based algorithm perform better exploration over the entire design space, therefore, causing to look for optimal points within a fixed number of evaluations compared with EI based approach. For a fixed evaluation budget, good exploration might lead to better convergence for complex problems. This phenomenon can be clearly seen from Table 3.4 at any fixed noise level.

To improve the convergence of EI-based algorithms, either the exploration ability of the algorithm needs to be improved, or the total evaluation budget needs to be increased. Exploration can be enhanced by increasing the uncertainty estimation of the physical system. Note that the stochastic model incorporates the noise level in it. Therefore, high noise levels would lead to large uncertainty estimates, thus improving the exploration of EI-based algorithms at the high noise level. However, with a high noise level, the stochastic model will suffer in accurately capturing the behavior of the response surface. This could mislead the search, even with the high exploration capabilities. Therefore, if the developed model initially captures the system behavior well, the high noise level could have better convergence. If the initial surface is not good, the high noise level could easily mislead the search for the optimal region. This ambiguous behavior can be justified from the results in Table 3.4. As the noise level increases from 5% to 40%, the algorithm performance suffers most of the time, but few instances show better average IGD values. For unconstrained problems, this behavior could be observed more frequently. It does not matter where the algorithm is searching for optimal points, as the entire design space is feasible. High exploration, combined with a good initial model, could help in identifying the better optimal region. However, it might lead to a higher number of infeasible points for constrained problems, thus leading to slow convergence, as shown next.

3.3.2.2 Constrained Test Problems

Four different test problems were selected, namely BNH, SRN, TNK, and OSY, to test the algorithm performance. True Pareto fronts of these problems are either continuous or discontinuous. BNH is a relatively easy problem among all. Therefore, for BNH, the total evaluation budget is kept at 300, while for the rest of the three problems, it is increased to 500 to achieve convergence in a limited budget. Figures 3.13 and 3.14 show plots for the best result out of all evaluations from all three algorithms for these four constrained problems at a 10% noise level, and Table 3.4 contains the results at all noise levels. Results at all other noise levels are shown in Appendix A.

Because BNH is the easiest problem, all three methods performed well to get the solution near-optimal Pareto front, with CAEI being slightly worst among them. For TNK, Algorithms 2 (AEIM) performed well at low noise levels, while algorithm 3 (CLCB) showed better performance at high noise levels. In the case of SRN, both AEIM and CLCB performed well in identifying the distribution of points near the actual Pareto front, whereas the CAEI approach suffered. OSY is a relatively complicated problem to solve. The results from all three approaches suffered, with AEIM and CAEI performing relatively better compared with CLCB. It is clear from both Figures 3.13 and 3.14 that for all the problems, CLCB can find the near-optimal solution with good distribution over the actual Pareto front. Only the distribution of optimal solutions in OSY is not good compared to the AEIM method. This could be due to the constant exploration term ($\Phi(-1)$) in the constraint handling function, controlling the exploration part. It is kept fixed for all the problems with all noise levels. It could be optimized to get a better distribution of points, which is not the goal here. The intent of this dissertation is to demonstrate the feasibility of these algorithms for stochastic multi-objective problems.

As can be seen from Table 3.4, the convergence of all the algorithms decreases with an increase



Figure 3.13: Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM and c) CLCB on BNH and SRN with 10% noise

in both noise level and the problem complexity. It makes sense because, with complex constrained problems, it becomes difficult to get into the feasible zone within a limited search. And with added noises, it becomes more challenging. As explained before, a high noise level makes the algorithm



Figure 3.14: Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM and c) CLCB on TNK and OSY with 10% noise

explore more, thus making it difficult to converge for constrained problems. It is difficult to converge as a high noise level would make these algorithms focus on a search operation over the entire design space rather than only on the feasible region. The higher noise level also makes it difficult for the model to capture the original response surface better, adversely affecting the convergence of the proposed algorithms. This hypothesis is supported by results in Table 3.4. The effect of noises could be seen for all three algorithms. With the increase in noise level, the performance of all three approaches deteriorates.

3.3.3 Validation on Engine Calibration Problems

Because these models are deterministic, external noises are added to their outputs to make the process stochastic. Figure 3.15 shows the structure implemented to perform the engine calibration study with external noises added to the system measurements. For this work, the Gaussian noises are considered with zero means and linearly varying standard deviations (explained before).



Figure 3.15: Algorithm implementation for engine calibration

For all three proposed methods, the Kriging model is developed initially using 100 points. Parameters used for optimization algorithms (both RGA and NSGA-II) are mentioned at the beginning of this section. The total evaluation budget is fixed at 500 for all three methodologies. Figures 3.16, 3.17, and 3.18 show the calibration results for all three proposed algorithms at 10% noise level, and Table 3.5 shows complete results with the IGD values at all noise levels. Appendix B shows the results at other noise levels.



Figure 3.16: Performance of constrained Augmented Expected Improvement (CAEI) algorithm on engine calibration problem at 10% noise level



Figure 3.17: Performance of Augmented Expected Improvement Matrix (AEIM) algorithm on engine calibration problem at 10% noise level



Figure 3.18: Performance of constrained Lower Confidence Algorithm (CLCB) on engine calibration problem at 10% noise level

| Noise | # 1: CAEI | # 2: AEIM | # 3: CLCB | |
|-------|-----------|-----------|-----------|------|
| 501 | 13.53 | 11.83 | 12.93 | Mean |
| 3% | 4.62 | 4.37 | 6.51 | Std. |
| 10% | 15.19 | 14.72 | 13.64 | Mean |
| | 3.04 | 2.88 | 2.33 | Std. |
| 20% | 19.66 | 91.09 | 16.49 | Mean |
| | 7.87 | 60.75 | 0.85 | Std. |

Table 3.5: IGD values for engine calibration problem

Figures 3.16, 3.17, and 3.18 shows both Pareto fronts and their corresponding points in the design space, which are actual positions of different calibration parameters. Blue dots show the data from the baseline study, while the red ones show the points obtained from the proposed surrogate-assisted methodologies. Comparing all three methods, CLCB performs the best while performances of CAEI and AEIM are very close. This can also be seen from the subplots representing actuator positions, and this trend can be observed at all noise levels in Table 3.5. For all three methods, as the noise level increases, their performance for identifying the optimal region also suffers. This is attributed to the difficulty in capturing the response surface behavior properly with the addition of high noise levels to the system. This, in turn, misleads the optimization algorithm to search for sub-optimal regions and thus, affecting the convergence within the limited evaluation budget.

From the design space results, it can be observed that all three algorithms are able to find points in the center zone, where finding the optimal actuator position is the easiest. For the other two zones that are narrow and difficult to achieve, all three methods cannot get the solution there. This could also be seen on the Pareto front as it lacks the spread of points over the actual front. The distribution of points for the CLCB method could be further improved by adaptively adjusting the exploration part of the constraint handling algorithm. The constant value used in the acquisition function could also be changed.

Comparing the results with these from the baseline study, all the proposed stochastic surrogateassisted methods are able to get the near-optimal solutions with more than 85% reduction in the computational budget. This is a significant cost reduction for performing optimization. Out of all three approaches, constraint lower confidence bound (CLCB) performs the best. This approach does not explicitly require noise variance input to perform the optimization, while EI-based approaches (AEIM and CAEI) do. This makes the CLCB algorithm suitable for practical problems when the noise structure is not known beforehand. Also, it is easy to extend this approach to high-dimensional multi-objective problems with multiple constraints.

Generally, the information of system noises is not known beforehand. Therefore, for performing the engine calibration, the algorithm should automatically incorporate all the system characteristics. The drawback of expected improvement-based approaches shown here is that they require noise information beforehand to work. One way to deal with the situation is making a separate noise model to estimate noise variance at any unknown test point, but this could not be accurate. Also, EI suffers from the myopic nature of the search process. The LCB-based approach has shown good performance on both numerical and engine calibration problems. It is also easy to implement as it does not require any information about the system. The only downside of this approach is that it has few user-defined inputs that need adjustments to have fast convergence. These user-defined inputs can be easily determined through offline study similar to what has been done to a GT-SUITETM model. Moving forward, for performing the experimental study, the CLCB-based surrogate-assisted optimization algorithm is used.

3.4 Chapter Summary

This chapter demonstrates the performance for these algorithms proposed in Chapter 2 that are first tested in a simulation environment before finalizing their implementation for the actual experimental setup. Following are the conclusions of this chapter.

- An engine calibration problem is formulated to validate the proposed algorithm. For validation purposes, a high-fidelity GT-SUITETM model is used. Experimental validation is planned based on the results observed from this study.
- 2) The proposed constraint handling algorithm for lower confidence bound (LCB) is first val-

idated on numerical test problems and engine calibration problems without considering measurement noise. The proposed approach demonstrated good performance compared with the existing state-of-the-art methods available in the literature.

- 3) The entire simulation setup is modified, and noises are added externally to these measurement outputs to match these from practical engine systems. Measurements are corrupted with different noise levels to check the efficacy of all three proposed multi-objective algorithms.
- 4) With measurement noises presented, the constraint lower confidence bound (CLCB) based algorithm showed better performance than expected improvement-based approaches. Comparing the results from proposed stochastic algorithms, CLCB is more practically implementable. The approach does not require the knowledge of the system noise information beforehand and shows better performance compared with the other two proposed extensions.

CHAPTER 4

INITIAL ENGINE CALIBRATION RESULTS ON EXPERIMENTAL SETUP

Motivated by the positive results obtained from the proposed stochastic algorithms in a simulation environment, the next step is its implementation onto an actual experimental setup. As discussed in Chapter 3, the constrained lower confidence bound (CLCB) based approach is feasible to implement onto a practical system as it does not require any system noise level information beforehand. Therefore, for the experimental work, only the CLCB based algorithm is implemented. The engine system available at the Energy and Automotive Research Lab at the Michigan State University is modified to perform the experimental study, which will be discussed in detail in this chapter.

This chapter discusses the overall experimental setup and the system modifications required to implement the algorithm, followed by the experimental results on problems with different complexity.

4.1 Overall Experimental Setup

To perform the experimental study, an 8-cylinder diesel engine provided by Ford Motor Company is used. The engine is installed on a dynamometer located at the Energy and Automotive Research Lab of the Michigan State University. Figure 4.1 shows the actual experimental setup with Figure 4.2 showing the communication between different instruments involved in the experimental setup.

As can be seen from both figures, several systems interact with the engine system to obtain the desired measurements. All the measurement hardware is installed in the engine test cell with the measurement instruments in the control room, as shown in Figure 4.2. Details of each system involved are explained next, followed by the procedure for performing the experiments.



Figure 4.1: Complete experimental setup for performing surrogate-assisted optimization process



Figure 4.2: Complete experimental setup for performing surrogate-assisted optimization process

4.1.1 Systems Involved in the Experimental Setup

The main component of the entire setup is the Ford 6.7L turbocharged diesel engine, whose specifications are mentioned in Table 4.1. Several additional sensors and actuators are installed on the existing engine to perform the experimental study. To spin the engine, the test setup uses an alternating current (AC) dynamometer (dyno), coupled with the engine crankshaft through a 1:2.75 gearbox and an inline Himmelstein torque sensor. It is used to maintain the desired engine speed by providing load or spinning torque to the engine. The AC dyno installed is capable of holding the engine speed up to 3200 RPM.

| Item | Value |
|-------------------|--|
| Engine Type | 4-Stroke Diesel with variable geometry turbo |
| Configuration | V8 |
| Displacement | 6.7 L |
| Bore | 99 mm |
| Stroke | 108 mm |
| Compression Ratio | 15.7:1 |
| Fuel System | High Pressure Common Rail Direct Injection |

Table 4.1: Diesel engine specifications

On the measurement side, cylinder pressures are measured by 8 piezo-electric pressure transducers installed in each cylinder head. The charge signals generated by the transducers are amplified by two AVL charge amplifiers and then fed to an A&D Combustion Analysis System (CAS) for data visualization and recording. Regular pressure signals such as manifold absolute pressure (MAP), compressor inlet pressure, turbine outlet pressure, etc., are measured by Druck pressure sensors and are also fed into CAS. All the temperature signals, such as ambient temperature, intake manifold temperature, coolant temperature, exhaust temperature, etc., are measured by thermocouples, and the amplified signals are recorded by the dyno NI Data Acquisition (DAQ) system. Two additional sensors for measuring brake torque and NO_x emissions are installed in the system. For brake torque measurement, a Himmelstein wireless digital torque meter with a stator-rotor assembly is installed. The rotor is installed on the driveshaft with the stator placed closed to the rotor to record the torque measurement. For the NO_x emission measurement, a separate NO_x sensor, provided by the Ford Motor Company, is installed in the exhaust pipe. The sensor is then connected to the Ford engine control module (ECM), which directly samples and broadcast the NO_x measurement.

The engine is controlled by a production Ford engine control module (ECM) using a preinstalled calibration map. The control and measurement variables available in the Ford ECM are accessed from the host computer running the ETAS-INCA software through the ES600.1 interface. In addition to the ECM, an open engine control unit (ECU), MotoTron, is used for sending the control signals to peripheral devices, such as a cooling tower, fuel pump, and bypass valve. MotoTron module is also responsible for sending the commanded pedal position signal to the ECM to set the desired engine load. CAN (controller area network) communication is used to communicate among Ford ECM, ETAS-INCA software, MotoTron, and NI Data Acquisition system. The communication is used to sync and record the data across multiple devices. In this way, data can be shared on the CAN bus, and INCA software can be used for data recording. Using this communication, desired control parameters, such as fuel injection quantity, injection timing, variable geometry turbocharger (VGT) vane position, exhaust gas recirculation (EGR) valve position, etc., can be commanded to the engine ECM through the INCA software.

The torque sensor also broadcasts the brake torque measurement through CAN, which is then communicated to the host computer (running ETAS-INCA software) to perform the learning algorithm. An exhaust valve is also installed to maintain the desired backpressure at the exhaust side of the engine to simulate the backpressure caused by the engine aftertreatment system. A closed-loop feedback control strategy is developed that adjusts the valve position to maintain the commended backpressure. Each operating condition requires a reference backpressure value to run the engine at its optimal condition. This reference value is commanded through the MotoTron according to the table provided by Ford Motor Company.

4.1.2 INCA-MATLAB communication

This is a crucial step in defining the supervisory control system for running the engine. As mentioned before, ETAS-INCA software can access all the measurement and calibration variables from the engine ECM. A set of these measurement and calibration variables are needed by the surrogate-assisted optimization approach to perform the calibration operation. Since the entire algorithm is coded in MATLAB, therefore, in the computer running the ETAS-INCA software, communication between MATLAB and the INCA software must be established to transfer the desired parameters between the optimization algorithm and engine ECM. In this case, the ETAS-INCA software acts as an interface that will capture the measurement signals for the algorithm and then pass the desired control parameters from the algorithm to the engine ECM. To establish the communication between these two software packages, the INCA-SIP add-on software from ETAS is used. Implementing the communication provides flexibility to control the calibration variables in the INCA directly from MATLAB. As can be seen from Figure 4.2, the MATLAB script communicates only to the ETAS INCA software for control parameter tuning.

After establishing the communication with INCA, the MATLAB script defines all the measurement and calibration variables in the INCA window. The measurement window is used to measure the desired objective functions, and the calibration window is used to change these variables to desired values. After defining the problem settings, the program overrides the calibration values in INCA software and measures the desired objective functions. These measurements are done when the engine system is stabilized at a new test point. A stopping criterion is generally put in the code to determine when to terminate the calibration algorithm. It could either be the maximum experimental budget available or any other convergence criterion. For the current work, no convergence criterion is set in place. The complete calibration process takes place until the maximum evaluation budget gets exhausted. Convergence criterion could be put in place to do early stoppage and save some evaluation budget.

4.2 Performing Experiments

The entire experimental setup installed at the Energy and Automotive Research lab at the Michigan State University has many operating constraints. The system could be run at several operating conditions, but precise control of all external parameters affecting the engine performance is challenging compared with a professional test bench. Running the engine on different days at a fixed point could give different output values due to changing environmental conditions, such as air humidity, ambient pressure, etc. One way to overcome this situation is to run the entire test within the same day. The assumption here is that the test cell condition would not change throughout the day. It is a valid assumption as the environmental condition would not significantly differ from morning till evening. With this assumption, the first step is the problem formulation with an intention to finish the complete experiments within a day. This limitation is due to the test cell capabilities, not with the algorithm. In an industrial test cell, where the environmental conditions could be monitored and controlled precisely, the test could be split into multiple days, but we do not have this flexibility.

Another issue with the current engine setup is the identification of non-operational points. The process for calibration involves exploring the unknown region in the design space of the control variables to detect the best possible objective functions. Not all the points in the design space are feasible, and even some are a little risky to run the engine as it may damage the engine, for example, due to engine knock or irregular combustion. Early detection of the non-operational point is crucial to avoid engine damage. It can easily be detected in industrial settings even before running the engine at that operating point completely. This, however, is not possible for our test setup. Therefore, after formulating the problem, the first step in performing the experiments is to identify a feasible region in the design space by manually checking the feasibility range for each design variable, as described in later sections.

4.2.1 Problem Formulation

The optimization problem formulated for the experimental work is slightly different from the one used for performing the simulation study. The main reason behind this is safety consideration and the limitation of our experimental setup, as mentioned before. Because this is the first practical implementation of this approach for engine calibration, the experimental work started with a simple multi-objective constrained problem. The intent here is to demonstrate the feasibility of implementing this algorithm to an actual test setup. After building the confidence of running the algorithm on the experimental setup, we will increase the problem complexity. The problem could be well extended to higher-dimensional complex problems in an industrial setting that would require multiple days to run the calibration test using the proposed approach.

This initial setup considers only two calibration variables. Increasing the number of control variables, objectives, and/or constraints will be decided based on this initial outcome. The following problem is formulated to begin the experimental validation of the calibration approach.

Minimize $y_1 = BSFC(\mathbf{x}), y_2 = NO_x(\mathbf{x})$ subject to $g_1 = NO_x(\mathbf{x}) \le NO_x$ limit Control vars. $\mathbf{x} = [VGT, EGR]^T \in \mathbb{R}^2$

The objective is to find an optimum trade-off between engine efficiency in terms of brake specific fuel consumption (BSFC) (y_1) and engine NO_x emissions (y_2) by varying two control variables: Exhaust Gas Recirculation (EGR) valve and Variable Geometry Turbocharger (VGT) vane positions. A change in the EGR valve position changes the amount of EGR going to the combustion chamber, which helps reduce the NO_x emissions but adversely affects the combustion characteristics of the engine, hampering its efficiency. VGT vane position controls the engine boosting. Closing the VGT vane causes an increase in the engine boost pressure, thus causing changes in the engine performance and emissions. In addition to these objectives, an additional constraint on NO_x emissions (g_1) is put to find the optimal operational region with NO_x levels below a certain threshold value ($NO_x \ limit$).

4.2.2 Finding out the Feasible Operationol Range in Design Space

Before running the optimization algorithm, it is desired to give the ranges of variations for the control variables to the learning algorithm. The current engine configuration cannot reject infeasible points. Therefore, to make the engine operate within a safe envelope, the safe operating ranges for the control variables are identified manually. Both control variables are changed so that the feasible VGT position should not choke the turbo, and the feasible EGR valve position should maintain engine combustion stability. A parameter called the coefficient of variation (COV) of in-cylinder pressure is used to measure combustion stability and is an indication of signal fluctuations. Usually, the COV is calculated for the pressure traces obtained directly from the engine. We calculated the COV of the indicated mean effective pressure (IMEP) directly from the in-cylinder pressure traces for this work. To ensure combustion stability, the COV value limit shall be maintained below the desired number. We have set the limit to be 1.5%, which is a reasonable limit for diesel engines.

For finding the safe range of operation, the engine is first run at the desired speed and load condition, and the actual values of the VGT and EGR observed from the engine ECM are taken as the baseline values. A sweep test is then performed around the baseline values to check the feasibility of these selected ranges for these calibration parameters. All these manual changes have been made directly using the ETAS-INCA software. A conservative range selection is made to maintain the safe engine operating region. Aggressive selection could be made in the future, depending on the results of the current operation. The set of these activities are repeated for all the operating points.

4.2.3 Automated Engine Operation

Figure 4.3 shows the entire setup for performing the experiments. The first block shows the steps needed to establish the setup for running the learning algorithm, while the second block shows the

overall automated engine calibration process. As shown here, the entire system runs automatically to perform engine calibration in real-time.



Figure 4.3: Setup for automating the engine calibration process

The overall experimental process starts with problem formulation, including the identification of a safe operating region. It is followed by establishing communication between MATLAB and INCA software, which creates a set of measurement and calibration variable windows for the variables, measurements, and calibrations declared in the MATLAB script. The engine is then run and stabilized at the desired operating condition (speed-load point) before performing the optimization process. With the finalized range of variation for the control parameters, the code is executed, generating a set of input points for evaluation. The algorithm overwrites the commanded
VGT and ERG positions and waits for the system to stabilize before performing measurements. From the experimental work, it was observed that small changes of EGR and VGT positions do not drastically change system characteristics, and letting the system run for 3 minutes is enough to achieve a steady state. Once the system achieves steady-state, the algorithm checks the coolant temperature and the charge air cooler (CAC) temperature to ensure that they are within 88-92°C and 43-47°C, respectively, before recording the data.

Before recording the data, the script clears the ring buffer, which is used to store data from INCA and lets it add the newest data set. The script then takes the last 100 data points and calculates the associated mean and standard deviation values to be used by the optimization algorithm. Until the next evaluation point is generated based on the new measurements, the engine remains operational at the previous evaluation point. The whole process keeps repeating until the evaluation budget is exhausted. All these operations happen in real-time while the engine runs with no or minimal manual interference to implement this calibration strategy.

4.3 Two - Variable Calibration Results

With the initial problem setup mentioned above, the experimental evaluation of the surrogateassisted optimization approach is made at five different operating conditions as shown in Figure 4.4. We will discuss the results from these experiments performed at the engine operating condition of 1400 RPM and 390 Nm load in detail. For other operating conditions, only the final results are shown.

For this particular operating condition (1400 RPM - 390 Nm), the constraint on NO_x emissions is set to NO_x limit = 0.6. Due to confidentiality, we cannot show the actual BSFC and NOx measurements and their corresponding EGR and VGT valve positions. Therefore, we scaled both objective and design space parameters, where EGR valve and VGT vane positions are scaled to be between [0 1]. For the EGR valve, 0 represents the closing position with minimum EGR flow and 1 the EGR valve position with maximum EGR flow; for the VGT position, 0 represents the maximum allowed open position for the VGT vane and 1 the maximum allowed closing position.



Figure 4.4: Operating points for performing experiments

The final optimization problem setup for running the engine is shown below. With the scaled control variables and constraints, the formulated optimization problem becomes

Minimize $y_1 = BSFC(\mathbf{x}), y_2 = NO_x(\mathbf{x})$ subject to, $g_1 = NO_x(\mathbf{x}) \le 0.6$ Control vars. $x_1 = VGT \in [0 \ 1], x_2 = EGR \in [0 \ 1]$

Because this is a two-variable problem, the total number of points for initial model development is fixed at 25 with a total evaluation budget of 45. As mentioned at the beginning of this chapter, the entire experimental evaluations are done using the Lower Confidence Bound (LCB) criterion. Therefore, the NSGA-II algorithm is used to optimize the acquisition function with a population size of 60 and 100 generations. Crossover probability is set to 0.9, and mutation probability equals 1/N, where N is *total number of variables*. The distribution index for crossover (SBX) operator = 15, and polynomial mutation = 20. As mentioned before, lower confidence bound (LCB) is used for acquisition function development with constant c = 2. For clustering, *dbscan*

| Туре | Total Iterations | Points Addition |
|---------------|-------------------------|------------------------|
| Deterministic | 4 algorithm iteration | (5,7,5,10) |
| Stochastic | 7 algorithm iteration | (5.1,4,1,5,3,1) |

 Table 4.2: Different iterations using DBSCAN for Deterministic and Stochastic

command from MATLAB is used with *minpoints* of 5. This work initially performs a case study using both deterministic and stochastic surrogate-assisted optimization approaches for performing engine calibration on an actual test bench. Based on the results, the approach for the rest of the experimental work is decided.

4.3.1 Results from Deterministic Approach

Firstly, the original algorithm with the deterministic Kriging model is implemented to the engine for evaluating its advantages and issues. The mean values obtained from the measurements are used in the algorithm for response surface development. Figure 4.5 shows the optimal trade-off curves obtained at different iterations with corresponding optimal points in the design space. Two different points are plotted. The actual points (red) are obtained from the actual experimental setup. The blue curve is obtained from the surrogate model formulated using the experimental points. Plots on the left-hand side show the trade-off curve obtained between the BSFC and NO_x emissions, whereas the right-hand side plots show the corresponding control parameter values. Iteration 0 corresponds to the model developed with 25 initial points, and further iterations correspond to the points added for model improvement. For this approach, all the evaluation budget gets exhausted after four iterations. The first row of Table 4.2 shows the total iterations and points added in each iteration during the model development.

As can be observed from Figure 4.5, the near-optimal curve obtained from the surrogate model is discontinuous and not smooth. Also, the design space shows optimal control configurations in multiple groups. This behavior could be well explained from the response surface developed from the surrogate model for both BSFC and NO_x emissions. Based on the observation, BSFC and NO_x



Figure 4.5: Pareto front in objective and design spaces using deterministic Kriging model

surface is expected to have multiple optimal locations. This observation is validated by plotting the response surface model obtained after fitting the original Kriging model.

Figure 4.6 shows the response surface model obtained at different iterations. As can be observed, with additional points for model enhancement, the response surface is becoming even



Figure 4.6: Response surface model for BSFC and NOx using deterministic Kriging model

more complex with multiple local optimal regions. This could be attributed to both measurement noises and interpolating nature of Kriging modeling. The Kriging model tries to pass through all the points while fitting the model because of its interpolation nature. In the presence of measurement



Figure 4.7: Convergence at different scenario using deterministic and stochastic Kriging model

noises, the surface would not be smooth, eventually leading to model overfitting. This is the main reason of developing several optimal regions, thus pushing the algorithm to look for evaluation points in the undesired region and affect algorithm convergence.

The hypothesis of convergence deterioration in the presence of measurement noises is further validated using a numerical test problem BNH. It is a two-objective and two-constraint problem with two design parameters. Figure 4.7 shows the convergence of the optimization algorithm under different conditions after the fixed initial 25 points and the total evaluation budget of 50. Figure 4.7 shows three different configurations, where Configuration 1 only has the measurement noises on objectives; Configuration 2 adds measurement noise on constraint 1, and Configuration 3 has measurement noises on both objectives and constraints. From Figure 4.7, it can be observed that for the deterministic Kriging model case, Configuration 1 (line with dot marker) converges faster than Configurations 2 (line with asterisk marker) and 3 (line with triangle marker). It is intuitive to understand the trend. With additional measurement noises on constraints and the deterministic Kriging model, the algorithm has difficulty finding the feasible region and converging towards

the optimal zone. Therefore, within the limited evaluation budget, the algorithm will suffer from convergence. However, the convergence is fast using the stochastic Kriging model with Scenario 3 (line with square marker). This is because of the smooth model generated during the process. Thus, the algorithm gets a chance to properly explore the design space and choose the points accordingly for further evaluation, saving the evaluation budget to explore in good regions. This provides good motivation to evaluate the stochastic surrogate-assisted optimization approach for the experimental work.

4.3.2 Results from Stochastic Approach

With the drawback in evaluating the deterministic approach on the experimental setup, the next step is to evaluate the same problem using the stochastic approach. Here, both the mean and standard deviation observed from the measurements are used for model development. After performing the experiments on the same problem setup as before, Figure 4.8 shows the progression of response surface of brake specific fuel consumption (BSFC) and NO_x emissions developed using a stochastic model. Table 4.2 (the 2nd row) shows the total iteration number required to exhaust the evaluation budget with points added to improve the model in each iteration. A total of seven iterations were performed to exhaust the budget.

As can be observed from the figure, the response surface obtained using the stochastic model is pretty smooth throughout the iteration process. Getting a smooth surface is essential in implementing an efficient exploration and exploitation strategy. The whole logic of the proposed approach is to explore areas with possible optimal regions based on the response surface. If the response surface has multiple local bins, the algorithm would require a higher number of iterations to converge to the actual optimal region. Comparing it with the response surface shown in Figure 4.6, it can be inferred that the optimization process would converge faster without wasting the evaluation budget on unwanted regions.

Using the data set obtained from implementing the stochastic optimization approach, both stochastic and deterministic models are developed, and their Pareto fronts are observed and shown



Figure 4.8: Response surface model for BSFC and NOx using stochastic Kriging model

in Figure 4.9. Iteration 0 corresponds to the initial data set and additional points as added as per Table 4.2 (2^{nd} row). The Pareto front observed from the stochastic model is continuous and smooth compared with the one obtained from the deterministic model. The observation becomes



Figure 4.9: Pareto front in objective space and design space using deterministic and stochastic Kriging models

interesting in the design space. With the stochastic approach, the algorithm quickly converges to the possible optimal region, which is at the boundary for this operating condition. However, even after iteration 4 (addition of 10 points for model improvement), the deterministic model is still creating clusters in other regions. This validates our hypothesis of convergence deterioration using the deterministic approach for problems with measurement noises.

4.3.3 Results at Other Operating Conditions

After identifying the appropriate approach for experimental evaluations, this section shows results observed by performing the stochastic surrogate-assisted optimization at the remaining four operating conditions. Different operating conditions are selected based on the combinations of speed-load points. Three different engine speeds are picked to represent **low-speed**: 1000 RPM, **mid-speed**: 1400, and **high-speed**: 1800 RPM case. Different loads are selected at these engine speeds.

Unlike the previous operating condition, the surrogate-assisted optimization algorithm is implemented for each operating condition with 20 initial points and a 45 evaluation budget. The rest of the parameters used for running the algorithms remain the same as before. Similar steps are followed for performing these experiments at each operating condition. Because each test condition will generate different results, a different problem is formulated for each operating condition, with the feasible operating range identified manually.

4.3.3.1 Operating Point: 1400 RPM - 500 Nm

A different problem is formulated for running the engine at this operating point as below.

Minimize
$$y_1 = BSFC(\mathbf{x}), y_2 =: NO_x(\mathbf{x})$$

Control vars.
$$x_1 = VGT \in [0 \ 1], x_2 = EGR \in [0 \ 1]$$

Note that the problem does not limit the normalized NO_x value to get the entire optimal front. This gives the end-user freedom to choose the control parameter specific to the requirements. Figure 4.10 shows the optimal curve along with the response surface obtained from the surrogate model at the end of the evaluation. The optimal solution obtained lies at the boundary of the VGT vane position. The optimal location could change depending on the range of variation of control variables, which is manually defined for each problem to maintain safe operation.



Figure 4.10: Engine calibration results at 1400 RPM - 500 Nm

4.3.3.2 Operating Point: 1400 RPM - 175 Nm

This is a low load condition. It is selected to validate the performance of the learning algorithm when the response surfaces are relatively flat. Problem formulation for this operating point is shown below.

Minimize $y_1 = BSFC(\mathbf{x}), y_2 = NO_x(\mathbf{x})$ subject to, $g_1 = NO_x(\mathbf{x}) \le 0.6$ Control vars. $x_1 = VGT \in [0\ 1], x_2 = EGR \in [0\ 1]$

Figure 4.11 shows the optimal results at this test point, which lies at the EGR boundary. The behavior of the result is very similar to the one observed for the 1400 RPM-390 Nm condition. Variation of the EGR valve could be enhanced to obtain the optimal solution within the design

space.



Figure 4.11: Engine calibration results at 1400 RPM - 175 Nm

4.3.3.3 Operating Point: 1800 RPM - 400 Nm

The problem formulated at this operating condition is as below.

Minimize $y_1 = BSFC(\mathbf{x}), y_2 = NO_x(\mathbf{x})$ subject to, $g_1 = NO_x(\mathbf{x}) \le 0.8$ Control vars. $x_1 = VGT \in [0\ 1], x_2 = EGR \in [0\ 1]$

Optimal results at this point are shown in Figure 4.12. This is a high-speed point with a moderate load, where changing the VGT and EGR positions could significantly affect engine behavior.

Therefore, after the initial DOE, the learning algorithm only focused on the optimal region, which is with low EGR values. That is why the response surface for BSFC is not smooth. More points are necessary for the unexplored region to get a good surface, which is not the goal here.



Figure 4.12: Engine calibration results at 1800 RPM - 400 Nm

4.3.3.4 Operating Point: 1000 RPM - 325 Nm

The following problem is formulated at the current operating point.

Minimize $y_1 = BSFC(\mathbf{x}), y_2 = NO_x(\mathbf{x})$ subject to, $g_1 = NO_x(\mathbf{x}) \le 0.5$ Control vars. $x_1 = VGT \in [0\ 1], x_2 = EGR \in [0\ 1]$ Figure 4.13 shows the optimal solution obtained at this low-speed mid-load operating condition. Again, the optimal solution lies at the boundary of design space.



Figure 4.13: Engine calibration results at 1000 RPM - 325 Nm

From the experimental results of all these operating conditions, it can be concluded that the proposed learning algorithm worked well in performing the engine calibration study under different engine speed-load conditions. For each operating condition, the parameter variation range is decided manually, which is not the ideal scenario but is limited to safety reasons. This can be further improved in a controlled environment.

One of the key observations from the majority of optimization results shown here is that the optimal solution obtained at the end of the algorithm lies at the boundary of the design space. Due to the limitations of performing engine tests under safe engine operational conditions, a conservative range for both VGT vane position and EGR valve opening was initially used. With the confidence

of successfully running the engine using the proposed calibration strategy, the obvious next step is to relax the limit and re-run the engine.

To perform the extension, the operating point **1400 RPM - 390 Nm** is again used. Using this point, the next section will discuss the revised problem formulation and its results.

4.3.4 Problem Formulation with Increased EGR Levels

It is essential to realize the interaction between the EGR level and desired objectives in this study. Higher EGR levels help to reduce the engine NO_x emissions. However, an increment of EGR level in the combustion chamber affects the combustion characteristics for a given fuel-air mixture, leading to a higher coefficient of variation (COV) of indicated mean effective pressure (IMEP), a measure of combustion stability. Usually, the higher the COV value, the worse the combustion stability. Therefore, an additional constraint on the COV value is added to ensure increment in EGR will not significantly affect the combustion stability. The revised problem formulation now relaxes the EGR variation range with an additional COV constraint. The rest of the problem formulation remains the same. The revised problem for performing the surrogate-assisted optimization now becomes

Minimize
$$y_1 = BSFC(\mathbf{x}), y_2 = NO_x(\mathbf{x})$$

subject to, $g_1 = NO_x(\mathbf{x}) \le 0.6$,

$$g_2: COV_{IMEP}(\mathbf{x}) \le 1.2\%$$

Control vars. $\mathbf{x} = VGT \in [0 \ 1], EGR \in [0 \ 1.625]$

where the COV_{IMEP} is calculated using the following expression.

$$COV_{IMEP}(\%) = \left(\frac{Satudard Deviation}{Mean}\right)_{IMEP} \times 100$$
(4.1)

The above problem formulation also shows an increased EGR valve range compared with the previously evaluated problem. The total evaluation budget is kept at 45, but the total number of

points required to construct the initial model is reduced to 20. This increases the number of search points for a given total budget. It gives the algorithm a chance to perform more exploration to now a complex problem. The rest of the optimization algorithm structure remains unchanged. The parameters for the NSGA-II algorithm for performing the optimization remain the same as before.

4.3.5 Results for Revised Problem

Based on the revised problem formulated, experiments are performed and corresponding optimal results are observed; see Figure 4.14.

Figure 4.14 shows the Pareto front of BSFC and NO_x (the 1st column) emissions and corresponding design space variables, EGR and VGT positions (in the 2nd column). It is well known that the addition of EGR in the combustion chamber helps reduce the overall heat coefficient of the charger-air mixture. This, in turn, lowers the in-cylinder temperature and thus helps reduce the NO_x emissions in the engine exhaust. However, the addition of EGR displaces fresh air going to the combustion chamber. This adversely affects the engine efficiency and, therefore, increases the BSFC value from the engine operation. With less engine boosting (low VGT value), high EGR levels will inadvertently deteriorate engine efficiency. Therefore, the points on the Pareto front do not have a region with a low VGT vane and high EGR valve position. A better picture of the feasible design space can be observed from the COV plot shown in Figure 4.15.

Figure 4.15 is obtained from interpolating the COV data based on 45 experimental points to show feasible points (dot marker) and infeasible points (square marker). It indicates that increasing the VGT closing percentage leads to lower COV values due to increased engine boosting and vice versa. From the Pareto front and its variables in the design space, it can be seen that with enlarged design space and an extra constraint on the COV, the near-optimal trade-off curve can be found without saturating the EGR valve position.



Figure 4.14: Pareto fronts in objective (the 1^{st} row) and design (the 2^{nd} row) spaces using deterministic and stochastic Kriging model

4.4 Chapter Summary

For this work, we initially performed a case study of implementing both the deterministic and stochastic surrogate-assisted optimization approaches for performing the engine calibration. It is



Figure 4.15: COV variation in the design space

well known that the deterministic Kriging modeling is an interpolation process. If the system has measurement noises, the model will try to go through all the points in a process, leading to overfitting and causing possible multiple local optimal solutions. In contrast, stochastic Kriging modeling is a regression process, and it will not try to get through all the points, leading to a better fit for the overall data set using both mean and variance values. After performing the experiments, it is observed that the algorithm quickly identifies the optimal region for the stochastic case, comparing it with the deterministic case. Therefore, for all further experimental work, the stochastic approach is implemented.

The initial experimental results for performing a simple engine calibration problem showed that the optimal solution lies at the boundary of the feasible EGR valve range. Therefore, a revised two-objective problem is formulated with an increased EGR valve feasible range and an additional constraint on the COV to ensure engine combustion stability. The algorithm was able to handle the additional constraint well and showed improved performance in identifying the feasible optimal region within the design space. Based on the promising results, the next chapter discusses a more complicated engine calibration problem with the addition of an electric-assisted boosting system (eBoost).

CHAPTER 5

CONTROL CALIBRATION FOR THE SYSTEM WITH EBOOST

The current diesel engine system is equipped with a variable geometry turbocharger to enhance its performance. A *turbocharger* is a device that utilized the waste energy from the engine exhaust to boost the intake air pressure, which helps enhance the engine power output with improved efficiency. However, the turbocharger suffers from the issue of *turbo-lag*, a delay in engine power output response when the demand is input to the engine system. This is especially predominant when a large step power demand is required since the turbocharger takes a certain time to speed up to the target speed. Turbo-lag mainly affects the transient response of the system. To handle the gap, an external electric-assisted boosting system (eBoost) can be used in addition to the turbocharger. eBoost uses external electrical power to operate, which can be controlled independently to the engine operational condition. An eBoost has been shown to be able to enhance the transient engine performance [44]. However, since it requires external energy, the power consumed by the eBoost needs to be considered while calculating BSFC, which adversely affects the engine BSFC.

An interesting observation using the eBoost can be found in [44]. The author showed that low to moderate eBoost speed range helps to improve the engine efficiency in terms of BSFC up to a particular level under certain operating conditions, especially at low engine speed. Upon further increasing the eBoost speed, the BSFC benefit gets overpowered by the amount of energy needed to spin the eBoost. This behavior can be explained by studying the system schematics shown in Figure 5.1 first.

Figure 5.1 shows the configuration of the air handling system used in the experimental setup at our lab, with turbocharger and eBoost connected in series, where eBoost is installed upstream of the turbocharger compressor. Air enters the system from the left and goes through two paths: bypass valve and eBoost compressor. The bypass valve is used to adjust the effect of eBoost on the system. Fully opening the valve will nullify the eBoost effect, and fully closing will make the system run through eBoost only. A feedback control logic is developed using a PI-controller for adjusting the



Figure 5.1: Schematic of the eBoost configuration

opening and closing of the bypass valve. For active eBoost operation, the bypass valve remains at the fully closed position to gain the maximum benefit. The charge air goes through the eBoost to the inlet of the compressor driven by turbo, which further performs the boosting operating based on the VGT vane position. The boosted air is then mixed with the exhaust gas downstream of the turbocharger, controlled by the EGR valve. Air goes into the combustion chamber, and the engine exhaust gas is then used to run the turbo. Some of the exhaust gas from the exhaust manifold is used for diluting the air charge going into the combustion chamber, where the EGR is cooled using an EGR cooler. The exhaust gas is then pushed out of the system after spinning the turbocharger. This explains the interaction between different components in the air-management system. However, How the addition of eBoost affects the BSFC is explained next.

As mentioned, the Bypass valve at the inlet side is fully closed while performing the eBoost experiments. At a particular operating point, the engine ECM adjusts the control parameter to maintain the boost pressure and the EGR flow rate to the engine. Increasing the eBoost speed increases the air pressure at the inlet of the turbocharger compressor. To maintain the same boost pressure at the intake manifold, the turbo speed needs to be reduced. This is achieved by opening the VGT vanes. Opening the VGT vane position, in turn, decreases the pressure in the exhaust manifold. The reduced exhaust pressure helps reduce the pumping losses, which contributes to

improved engine efficiency. Apart from the engine efficiency, with the reduced exhaust pressure, the EGR flow to the intake manifold after going through the EGR cooler will decrease. Therefore, to maintain the desired EGR flow to the combustion chamber, the EGR valve opens more. These observations were experimentally validated in Ph.D. dissertation [44]. The addition of eBoost to the system provides another parameter contributing to the performance enhancement, which needs to be calibrated to achieve better performance than that of the original system configuration.

This chapter discusses the experimental results obtained by calibrating the three engine control parameters: VGT vane position, EGR valve position, and eBoost speed. First, the optimization problem is formulated considering the dependency of two other control parameters (VGT and EGR) on the eBoost speed. Proper design parameter ranges are defined to operate the engine safely. It is followed by a quick discussion on performing the experiments with eBoost, and finally, a discussion on the observed experimental results.

5.1 **Problem Formulation**

A crucial step in formulating the problem is defining safe operating ranges for parameters in the design space. As discusses above, both VGT vane and EGR valve positions depend on the eBoost speed. This behavior is quantified experimentally; see Figure 5.2. The normalized values of VGT vane closing, EGR valve opening, and eBoost speed are plotted.

As expected, increasing the eBoost speed opens the VGT vane more (decrease in the VGT vane closing) to maintain the boost pressure and also increases the EGR valve opening. Looking at the EGR valve opening plot, beyond a certain eBoost speed, the EGR valve position is saturated at the maximum level, indicating that it is impossible to maintain the EGR % at the desired level beyond that eBoost speed.

5.1.1 Defining Parameter Ranges

Considering the behavior of both VGT vane and EGR valve positions, upper and lower bounds are determined for both control parameters, which change as a function of eBoost speed. The



Figure 5.2: Variation of VGT vane and EGR valve positions with eBoost speed

constrained design space is shown in Figure 5.3. To ensure that the search point should lie between the two bounds, additional constraints on VGT vane and EGR valve positions are imposed in the problem formulation.

Figure 5.3 also shows a dashed vertical line. It splits the entire design space into two parts: eBoost inactive and eBoost active regions. The region to the left is assumed to be the eBoost inactive case, and the right side is the eBoost active case. As mentioned before, for performing the experiments, the bypass valve remains closed all the time. Therefore, eBoost spins all the time with the lowest speed that develops ambient pressure (0.99 bar) at the air inlet. A small eBoost speed variation is allowed in the eBoost inactive region. Within the allowed variation, the eBoost does not affect the boost level.

The rest of the problem formulation remains the same as before. The objectives are the same, that is, to obtain the trade-off curve between engine efficiency in terms of effective brake specific fuel consumption ($y_1 = BSFC_{eff}$) and engine emissions ($y_2 = NO_x$). The current setup involves three control parameters: VGT vane position, EGR valve position, and eBoost speed. Like before, there are constraints on NO_x and COV values. An additional constraint is added for the power consumption used by eBoost. It is recommended that the power drawn by the eBoost should be less than or equal to its power limit of 6 kW. With the setup described, the optimization problem



Figure 5.3: Control parameter variations for three-parameter calibration problem

can be mathematically written as below.

Minimize
$$y_1 = BSFC_{eff}(\mathbf{x}), y_2 = NO_x(\mathbf{x})$$

subject to, $g_1 = NO_x(\mathbf{x}) \le 0.83,$
 $g_2 = COV(\mathbf{x}) \le 1.2\%,$
 $g_3 = ePower(\mathbf{x}) \le 6 \ kW,$
 $g_4 = -VGT(\mathbf{x}) \le -VGT_{low}$ and $g_5 = VGT(\mathbf{x}) \le VGT_{high},$
 $g_6 = -EGR(\mathbf{x}) \le -EGR_{low}$ and $g_7 = EGR(\mathbf{x}) \le EGR_{high}$

$$(5.1)$$

Control Vars $\mathbf{x} = [VGT, EGR, eSpeed]^T \in [0, 1]^3$

Because eBoost uses the external electrical energy to operate, that is accounted for during calculating the effective brake specific fuel consumption $(BSFC_{eff})$. The expression for effective BSFC $(BSFC_{eff})$ is defined below.

$$BSFC_{eff} = \frac{BSFC \times \dot{W}_b}{\dot{W}_b - \frac{1}{\eta_m \eta_e} \dot{W}_m}$$
(5.2)

where $BSFC_{eff}$ is the effective BSFC value; \dot{W}_b is the brake power; η_m is the motor characteristic efficiency; η_e is the battery energy conversion efficiency. For this work, we assumed $\eta_e \eta_m = 0.9$, which is a reasonable assumption.

Because four of the constraints are on control variables, they are not expensive to evaluate. Therefore, the optimization algorithm uses the NSGA-II method directly to handle these constraints.

5.2 Performing eBoost Experiments

For performing the eBoost experiments, the bypass valve is fully closed to avoid air leakage from the intake side. Therefore, after closing the bypass valve, the eBoost needs to run at all times to maintain the conventional compressor inlet pressure at close to ambient pressure to avoid engine damage. The minimum eBoost speed should be able to generate inlet pressure equal to the ambient pressure, which is usually around 0.99 bar. A threshold value is set for the eBoost speed $(eSpeed_{Th})$. Anything above this speed is considered as eBoost active case, and anything below that is eBoost inactive case. This threshold limit is set at 500 RPM more than the minimum eBoost speed required to maintain the desired ambient pressure.

The current experimental work with eBoost is performed only at low engine speed due to the possible efficiency improvement benefit. The threshold eBoost speed required to maintain the desired ambient pressure varies mainly with engine speed. High engine speed requires high eBoost speed, which, in turn, uses more power from the battery, leading to the early discharge of the battery bank. The current experimental setup has two sets of battery banks to maintain the desired minimum 48V to operate the eBoost, and only one bank is used at a time. Both banks are charged using a 56V DC supply. However, simultaneous engine operation while charging the bank is avoided due to the electrical noise from the DC power supply to the engine setup. Therefore, the way we operated the eboost is that we first drain both battery banks while performing desired experiments. Once a set of experiments are done, the algorithm is paused, and both battery banks are charged. Once the banks are fully charged, they are again used to continue performing tests.

The rest of the experimental procedure remains the same, as explained in Chapter 4. After the problem formulation, the connection is established between INCA-MATLAB to create the measurement and calibration window in the ETAS-INCA. After the diesel engine gets stabilized at the desired operating condition, eBoost is run before closing the bypass valve to avoid any chance of engine choking. The learning algorithm is then implemented to perform the engine calibration. All the measurements from the eBoost are communicated through the MotoTron via CAN communication, which can be visualized through NI-Veristand in a separate computer. Therefore, the eBoost parameters are manually entered in the main algorithm, running in the ETAS-INCA laptop. These variables can be communicated to the INCA via CAN communication to make this process fully automated.

5.3 Experimental Results with eBoost

Experiments were performed at a fixed engine operating condition of 1000 RPM and 325 Nm. Total 35 points are taken for the initial model development, and the evaluation budget is fixed at 70 points. NSGA-II is used for optimizing the acquisition function with a population size of 100 and a total of 200 generations. Other parameters for NSGA-II include crossover probability of 0.9, mutation probability of 1/3 (where 3 is for the total number of variables), the distribution index for crossover (SBX) operator of 15, and polynomial mutation of 20. Lower confidence bound (LCB) is used for acquisition function development with constant c = 2. For clustering, *dbscan* command from MATLAB is used.

5.3.1 Baseline Result

It is important to identify baseline results from the engine at 1000 RPM and 325 Nm load condition. The baseline result is obtained by performing the experiments without running the eBoost, where the control parameters take the optimal values predefined in the engine ECM. Figure 5.4 shows the normalized BSFC and NO_x values at this operating condition.

The baseline result will be used as a reference to decide if the surrogate-assisted optimization approach is able to identify a better region of operation with the eBoost or not. Another point to mention here is the NO_x limit. For the learning algorithm, the constraint limit on the NO_x emissions is relaxed to obtain the entire optimal front with regions of improved BSFC without worrying about the NO_x emissions. However, in general, the limit on NO_x emissions is decided based on the drive-cycle constraints and other emission levels since the addition of EGR to the combustion chamber affects hydrocarbon and soot emissions as well. Therefore, we will initially show the entire optimal curve to demonstrate the efficacy of the proposed algorithm but will eventually be interested in the settings for obtaining the NO_x emissions close to the threshold limit.



Figure 5.4: Nominal BSFC and NO_x values

5.3.2 Results using Proposed Approach

Before implementing the learning algorithm, the first step is creating the initial set of points for model development. As mentioned before, both VGT vane and EGR valve positions are constrained. Therefore, to create points within the defined bounds, the constrained Latin Hypercube Sampling (CLHS) toolbox [4] is used. A total of 35 points are used to generate the initial model to better capture the behavior of the response surface. Figure 5.5 shows the initial DOE test points, and Figure 5.6 shows the corresponding output values in the objective space.

The red points in both figures are when the eBoost is active, and the blue ones are when the eBoost is inactive. As can be observed from Figure 5.6, most of the DOE points are with high NO_x values with few points showing improvement in $BSFC_{eff}$ and/or NO_x . We can expect the learning algorithm to push the search in the design space for improved $BSFC_{eff}$ and NO_x values.

Figure 5.7 shows all the experimental points after complete these 73 tests. It also identifies



Figure 5.5: Initial DOE points using constrained latin hypercube sampling (CLHS) in design space



Figure 5.6: Output values corresponding to initial DOE inputs

the Pareto front obtained from the actual experiments with the hexagonal sign. Compared to the points initially evaluated using the DOE (Figure 5.6), the learning algorithm is able to search for points in the region with better $BSFC_{eff}$ and NO_x values than the initial ones. The constraint on the NO_x emissions used for this problem is relaxed to get the entire optimal front. This gives more flexibility to the end-users to decide the trade-off between two objectives at this operating condition. The trade-off curve obtained has points for both eBoost active and inactive cases. With certain configurations, the eBoost inactive case showed better BSFC but suffered from increased NO_x emissions at those points. There are also points with both improved $BSFC_{eff}$ and NO_x emissions compared with the baseline results.

If we strictly follow the NO_x limit from the baseline results, as shown by the dashed line in Figure 5.7, we can observe that points obtained with the eBoost active case have both improved $BSFC_{eff}$ and NO_x emission values. This confirms the benefit, measured by engine performance, of using the eBoost. The improvement in the BSFC value using the same setup has been validated



Figure 5.7: Total experimental evaluations in the Objective Space

in Ph.D. dessertation [44]. With the changes in both VGT and EGR positions upon increasing the eBoost speed, the performance improves. This enhanced performance is observed after considering the power drawn by the eBoost from the battery. With a further increase in the eBoost speed, the power that is drawn by the eBoost increases, which reduces the $BSFC_{eff}$. There exists a sweet spot for running the eBoost in order to get the optimal performance from the engine.

Figure 5.8 shows the experimental points in the design space. These points corresponding to the trade-off curve are also shown in the figure. The optimum eBoost speed range to obtain improved engine performance is between 0.2 and 0.4. Operating eBoost in this range uses less battery power. The energy used by the eBoost is less than the benefit from the engine efficiency improvement. In addition, the optimal solution lies around the boundaries of the design space.

With all the experimental evaluations performed, the final surrogate model is obtained. Based on the developed model, the near-optimal front is identified by evaluating the unknown points in the design space. Since the model is cheap to evaluate, a set of 20,000 points in the design space is



Figure 5.8: Experimental evaluation points in the design space

evaluated to obtain the near-optimal Pareto front. Figure 5.9 shows the Pareto front obtained from the surrogate model, while Figure 5.10 shows these points in the design space. When evaluating the Pareto front, the constraint on the NO_x is set to 0.4 (that was 0.83 before). This is done to clearly represent the advantage of using eBoost to improve performance while maintaining the NO_x emissions to the desired level.



BSFC vs NO_x : Pareto Front from Surrogate Model

Figure 5.9: Pareto front obtained from the surrogate model after all the evaluations

A generic diesel engine calibration problem accounts for optimizing multiple emission levels such as NO_x , Hydrocarbon (HC), soot emissions. Since, for our study, we are only considering the NO_x emissions, we are setting the NO_x emission limit close to the one defined in the engine ECM to make sure other emission levels are within a certain feasible range. To more accurately validate this hypothesis, these emissions should be measured and controlled. This is not considered here, but the problem could be implemented in a well-controlled engine test cell.



Figure 5.10: Points corresponding to the Pareto front from the surrogate model in the design space

These results correlate these observations using the actual experimental data, as discussed above. Within the eBoost speed range of [0.2 - 0.4], the results obtained for $BSFC_{eff}$ and NO_x emissions are better than the baseline ones. There is also a region in the eBoost inactive case, where the NO_x emissions are superior, but it suffers from a high $BSFC_{eff}$ value.

In order to validate the accuracy of these estimates from the surrogate model, four different points are identified from the achieved Pareto front of the surrogate model. All these points are evaluated using the experiential setup to identify the corresponding actual values. Figure 5.11 shows the error between actual versus the estimated values from the surrogate model for both $BSFC_{eff}$ and NO_x emissions.



Figure 5.11: Error (%) between actual data and estimates from the Surrogate Model

As can be observed, the error estimation for $BSFC_{eff}$ is less than 2%, which is acceptable for the experimental setup. However, the NO_x emissions have a relatively high error percentage. We
measured the NO_x emissions in ppm (parts per million) for the current work, and it is common to have higher fluctuations in terms of ppm. For example, if the model estimates the NO_x emissions to be 90 ppm and through the experiments, we obtain an average of 80 ppm, then the estimation error is 12.5 %. The fluctuation itself is normal, but the error percentage makes it look worse in error estimation. This concludes our experimental work with the addition of eBoost to the diesel engine system.

5.4 Comparison of Experimental Evaluations

This section compares the total number of experimental evaluations required for performing the engine calibration process using the conventional and proposed surrogate-assisted optimization approaches. Industries currently use a high-fidelity GT-SUITETM model to identify the optimal parameter range and then perform the full-factorial search. Assuming a total of 8 points for each control variable for performing the full-factorial calibration, Figure 5.12 shows the comparison of experimental evaluations corresponding to the number of control parameters using both approaches.

In Figure 5.12, x-axis shows the number of control variables, and y-axis shows the number of experimental evaluations corresponding to these control variables on a log-scale. As can be observed, the total number of evaluations using the full-factorial approach increases exponentially as the number of design variables increases. For a more complex problem with more than 10 control variables, this becomes intractable. Whereas, comparing the evaluations using the surrogate-assisted optimization approach, the increase in the number of experimental evaluations is not high.

Comparing the ratio of the experimental evaluations using these two approaches, for a twovariable calibration problem, the ratio is $\frac{45}{64} = 0.7$. This ratio reduces significantly for a threevariable calibration problem. With three control parameters, the ratio is $\frac{70}{64} = 0.13$, which is a drastic reduction in the experimental evaluation. We expected this approach to show an even smaller ratio for the optimization problem with more than three variables. The dashed line in Figure 5.12 is our expected total number of evaluations for the four variable control problems. This is observed



Figure 5.12: Comparison of experimental evaluations using existing and proposed approaches

by assuming the total experimental evaluation budget of 140 for four variable calibration problems. With this number, the estimated ratio would be $\frac{140}{4096} = 0.035$, which is even smaller. However, this is only our hypothesis. We observed significant improvement in reducing the experimental evaluations based on the two/three-variable calibration problems. It will be exciting to validate this approach for performing engine calibration for engine calibration problems with dimensions higher than three.

5.5 Chapter Summary

This chapter discusses the problem of performing engine calibration with three control variables. An external boosting device (eBoost) is added to the diesel engine system to improve both engine transient and steady-state performance, where this study focuses only on steady-state calibration. Before performing the experiments, a revised problem is formulated for calibrating a complex engine architecture based on the interactions of VGT vane and EGR valve positions with the eBoost speed. Based on obtained behavior, a three-variable control calibration problem is formulated with two objectives and seven constraints. The calibration results demonstrated improved engine performance in terms of both effective BSFC and NO_x emissions using the eBoost, compared with the baseline results using only VGT and EGR. Also, significant savings in experimental evaluations are observed compared to the traditional full-factorial approach. This study provides confidence in implementing the proposed approach to solve a high-dimensional calibration problem with a significant reduction in the experimental evaluations.

CHAPTER 6

CONCLUSIONS AND FUTURE WORKS

This chapter presents the concluding remarks of the research work discussed in this dissertation and provides future research directions to advance the data-driven approaches for performing engine calibration.

6.1 Conclusions

This dissertation demonstrates the use of a surrogate-assisted optimization approach, through both simulation and experimental studies, to efficiently perform the engine calibration with a goal to reduce the experimental evaluations. This work is a first step towards automating the engine calibration process with little to no manual interactions to the engine system. The research work begins with extending the existing algorithms to make them applicable to practical physical systems. The work only focuses on the two widely used approaches: Expected Improvement (EI) and Lower Confidence Bound (LCB). The lower confidence bound (LCB) based optimization approach extends well for solving multi-objective problems and is easy to implement onto any engine system. However, there was no efficiency constraint handling algorithm available. Therefore, a simple constraint handling approach is proposed for lower confidence bound (LCB). It is initially validated on various constrained multi-objective numerical test problems and later implemented to perform the engine calibration using a high fidelity GT-SUITETM engine model. Simulation studies aim at calibrating the exhaust gas recirculation (EGR), variable geometry turbocharger (VGT) valve, and Start of Injection (SOI) to achieve a trade-off between engine efficiency in terms of brake specific fuel consumption (BSFC) and NO_x emissions. Constraints on engine load condition (brake mean effective pressure) and boost pressure are used to make the engine operation feasible. The efficacy of the proposed constrained lower confidence bound (CLCB) approach was compared with existing state-of-the-art expected improvement (EI) based algorithms, and the performance is shown to be on par.

In order to implement this approach to a practical system, it is crucial to extend the algorithm to handle the system with measurement noises. Heteroscedastic (non-uniform) noise variance is assumed for the extension to make it more realistic for real-world implementation. Three different extensions are proposed. Two of them are based on expected improvement (EI): Constrained Augmented Expected Improvement (CAEI) and Augmented Expected Improvement Matrix (AEIM). The third is the proposed constrained lower confidence bound (CLCB) approach, which does not require any modifications to handle the measurement noise. All three extensions are validated using the high-fidelity GT-SUITETM engine model corrupted with measurement noises at different noise levels. Based on the results, the CLCB approach has shown to work well in a stochastic environment compared with two other expected improvement-based extensions. One major disadvantage of the expected improvement-based approach is that it requires noise information beforehand to handle the stochastic system. However, the proposed CLCB approach does not require any modification. Therefore, for the experimental evaluation, only the lower confidence bound-based approach is implemented.

Finally, the algorithm is implemented onto an actual engine setup installed at the Energy and Automotive Research Laboratory of the Michigan State University. The entire experimental setup is modified, and an automated calibration strategy is devised to run the engine in a safe operating region at all times. Initially, a simple problem is formulated to perform the control calibration of two variables: variable geometry turbocharger (VGT) vane and exhaust gas recirculation (EGR) valve positions. The goal is to obtain the optimal trade-off between brake specific fuel consumption (BSFC) and NO_x emissions with a constraint on NO_x emissions. Both deterministic and stochastic surrogate-assisted optimization approaches are implemented to check their efficacy on the actual engine experimental setup. It is observed that the stochastic approach provides a better evolution of the learning optimization algorithm, comparing with the deterministic one. This is due to the regression-based modeling by stochastic approach compared with the interpolation one by deterministic algorithm, that leads to overfitting of the response surface in the presence of measurement noise. Therefore, the rest of the experiential work is performed using only the stochastic approach.

After getting comfortable using the proposed approach for engine calibration, the original engine system is expanded with an external electrically-assisted boosting device called *eBoost*, and a new calibration problem is formulated for calibrating this new engine architecture. The revised problem becomes a three-parameter calibration problem (VGT, EGR, and eBoost) with the same two objectives (BSFC and NO_x emissions) along with several constraints on both measurements and control variables.

The overall experimental demonstration of the proposed approach leads to optimized engine performance with a significant reduction in the number of experimental evaluations required for performing the engine calibration. This study has opened the door to explore different challenges in control calibration problems for practical systems. This work's true potential can be observed in a controlled environment, usually achievable in an industrial setting. Because the current work is a part of the proof of concept, all the tests are performed in our dyno cell at the Michigan State University. We do not have the resources to perform extensive tests with high-dimensional complex problems. The following is a list summarizing conclusions from this study.

- This study successfully proposed a simple constraint handling method for Lower Confidence Bound (LCB) based acquisition function. The proposed multi-objective constraint handling approach is shown to work well for both numerical problems and engine calibration problems.
- To deal with the practical system, three extensions of the surrogate-assisted optimization algorithm are proposed to handle the constrained multi-objective optimization problem with measurement noises. Heteroscedastic (non-uniform) noises are considered to make the approach more realistic for real-world applications.
- Lower confidence bound with the proposed constraint handling approach has shown to work well in the stochastic environment compared with two other expected-improvement (EI) based extensions. Also, since EI-based extensions require noise information beforehand to perform the optimization in a stochastic system, which is not a viable option for their implementation to the practical systems.

- For initial experimental evaluation, two control variables, variable geometry turbocharger (VGT) vane and exhaust gas recirculation (EGR) valve positions, are calibrated to identify the optimal front between engine efficiency (BSFC) and NO_x emissions. The stochastic surrogate-assisted optimization with the proposed constrained lower confidence bound approach provides better convergence compared with the deterministic approach. It is due to the regression-based modeling approach over the interpolation one that causes overfitting of the response surface model in the presence of measurement noises. Since the measurements of practical systems are always polluted with noises, the stochastic surrogate-assisted optimization approach is considered for the rest of the experimental work.
- The experimental evaluation started with a simple two-variable problem with one constraint. Based on the positive results, the calibration problem is extended to a three-variable control problem with multiple constraints. All the experiments are conducted by operating the engine within a feasible operational region. A significant reduction in experimental evaluations is observed compared with the existing approach used in the industries.
- The true advantage of the proposed algorithm would be evident for high-dimensional design space problems. With the study on both two and three variable calibration problems, it is projected that the proposed approach would significantly reduce the number of engine evaluations required for the calibration process.

6.2 Future Work

For future work, one of the modifications that could enhance the algorithm's convergence is to make the exploration limit of the constraint handling function adaptive. A fixed exploration limit is used throughout simulations and experimental studies. The exploration of constrained design space is required initially to identify feasible boundaries. However, as we get close to exhausting the evaluation budget, a more conservative approach could be implemented. A convergence analysis shall be done in order to update the exploration limit effectively.

Note that the demonstrated experimental work is for performing the static engine calibration. However, the overall calibration process is a combination of both static and transient engine calibrations. For transient engine calibration, a dynamic system model is required. Therefore, one of the future works could be performing the transient engine calibration using the data-driven approach. A data-driven system identification could be performed. It will be interesting to look for an approach requiring fewer points to identify the system dynamics capturing all necessary system modes for control strategy development. Another exciting direction is performing the *dynamic system identification*. It is well known that system dynamics changes due to system aging, different operational environments, and conditions. Therefore a strategy for dynamic update of the data-driven model could be developed. This will help in updating the control strategy effectively without putting extra effort into the system modeling.

One of the observations from the experimental results is the response surface behavior. For a particular objective function, the behavior remains the same at different operating conditions, only the function value changes. However, for each operating condition, the entire set of experiments were repeated. The knowledge of the response surface behavior could be used to reduce the experimental evaluations further. The concept of *Transfer Learning* can be implemented in such scenarios, where at any new operating condition, the entire set of experiments is not needed to be repeated. Only a few evaluations are required, and the response surface generated at the previous operating condition can be updated to the new operating condition. Implementing this approach will further reduce the number of experimental evaluations for performing the overall static engine calibration.

APPENDICES

APPENDIX A

STOCHASTIC RESULTS ON NUMERICAL TEST PROBLEMS

This appendix shows the results observed on the numerical test problems using the proposed stochastic extension for the surrogate-assisted optimization approach. All three extensions are applied to both unconstrained and constrained multi-objective numerical test problems to evaluate their efficacy.

As mentioned in Chapter 3, the following test problems are used for validation purposes, as shown below.

| Problem | No. of Objectives | No. of Constraints | No. of Design Variables |
|---------|-------------------|--------------------|-------------------------|
| ZDT1 | 2 | 0 | 8 |
| ZDT2 | 2 | 0 | 8 |
| ZDT3 | 2 | 0 | 8 |
| BNH | 2 | 2 | 2 |
| SRN | 2 | 2 | 2 |
| TNK | 2 | 2 | 2 |
| OSY | 2 | 6 | 6 |

Table A.1: Numerical test problems for stochastic evaluations

The outputs from these test problems are deterministic. Therefore, both objective and constraints are corrupted with the noise of different variances. Four different noise levels: 5%, 10%, 20%, and 40%, are implemented to test the performance of the proposed algorithms. These results for 10% noise level are shown in Chapter 3. This appendix shows these results at other noise levels.

A.1 5% Noise variance

The current section shows the results by adding 5% noise to both objectives and constraints.

A.1.1 Unconstrained

This section shows the results for problems without constraints. Three different problems come under this category from the list of all the test problems used.



Figure A.1: Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on ZDT1 and ZDT2 with 5% noise



Figure A.2: Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on ZDT3 with 5% noise

A.1.2 Constrained

This section is dedicated to problems with constraints. Problems with different complexity are used. Since the noise level is small, we could expect better convergence from the algorithm within 139

the limited evaluation budget.



Figure A.3: Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on BNH and SRN with 5% noise



Figure A.4: Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on TNK and OSY with 5% noise

A.2 20% Noise variance

The section here shows the results observed by corrupting the outputs of the test problems with a 20% noise.

A.2.1 Unconstrained

With a higher noise level, we expect convergence of the algorithms to suffer. The results showed better performance with the lower confidence bound (LCB) based optimization approach.



Figure A.5: Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on ZDT1 and ZDT2 with 20% noise



Figure A.6: Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on ZDT3 with 20% noise

A.2.2 Constrained

Because of the higher noise level, it becomes difficult for the algorithm to identify feasible boundaries and converge towards the optimal region. Compared to the low noise case, the results with higher noise are worse.



Figure A.7: Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on BNH and SRN with 20% noise



Figure A.8: Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on TNK and OSY with 20% noise

A.3 40% Noise variance

This section shows results with addition of 40% noise level. The 40% noise level is added to test the efficacy of the algorithm in handling high noise level.

A.3.1 Unconstrained



The subsection here shows the results for unconstrained problems.

Figure A.9: Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on ZDT1 and ZDT2 with 40% noise



Figure A.10: Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on ZDT3 with 40% noise

A.3.2 Constrained

For the constrained problems, it is a challenging task to identify feasible regions easily since the high noise level affects the response surface behavior drastically. This subsection shows the convergence 147

in such scenarios.



Figure A.11: Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on BNH and SRN with 40% noise



Figure A.12: Stochastic surrogate-assisted optimization results using a) CAEI b) AEIM, and c) CLCB on TNK and OSY with 40% noise

APPENDIX B

STOCHASTIC RESULTS FOR ENGINE CALIBRATION

This appendix here shows the simulation results obtained after performing the stochastic-surrogate assisted optimization for the engine calibration problem at 5% and 10% noise levels. Chapter 3 only shows the results for 10% noise level.

As mentioned before, the following problem is formulated for performing the engine calibration using a high-fidelity GT-SUITE^{TM} engine model.

Minimize
$$y_1(\mathbf{x}) = BSFC(\mathbf{x}), y_2(\mathbf{x}) = NO_x(\mathbf{x})$$

subject to $g_1(\mathbf{x}) = Boost(\mathbf{x}) \le 2 \ bar$,
 $g_2(\mathbf{x}) = Load(\mathbf{x}) \ge 9.95 \ Nm$,
Control Variables $\mathbf{x} = [VGT, EGR, SOI]^T$
 $x_1 = VGT \in [0\ 1]\ \%$,
 $x_2 = EGR \in [0\ 1]\ degree$,
 $x_3 = SOI \in [0\ 1]\ bTDC$

The goal here is to get the optimal trade-off between engine efficiency (BSFC) and its emission (NO_x) by optimally controlling the engine control parameters: Variable Geometry Turbocharger (VGT) vane position, Exhaust Gas Recirculation (EGR) valve position, and Start of Injection (SOI). To make the process stochastic, the outputs from the simulations are corrupted with non-uniform noise variance and then given to the learning algorithm.

B.1 5% noise level

The current section shows the results by corrupting the measurements with a 5% noise level. The addition of system noises will change the response surface behavior. Since the noise level is small, we could expect the results not to get affected much in terms of convergence to the baseline Pareto front.



Figure B.1: Performance of constrained Augmented Expected Improvement (CAEI) algorithm on engine calibration problem at 5% noise level



Figure B.2: Performance of Augmented Expected Improvement Matrix (AEIM) algorithm on engine calibration problem at 5% noise level



Figure B.3: Performance of constrained Lower Confidence Algorithm (CLCB) on engine calibration problem at 5% noise level

B.2 20% noise level

This section presents the case when the output measurements are corrupted with 20% noise. Both objectives and constraints have 20% noise. The addition of high noise level will cause two issues for the learning algorithm. It will first affect the constraint surfaces. This means that the algorithm might have difficulty identifying the feasible region while performing the search operation. The addition of a 20% noise level would also significantly affect the behavior of the objective response surface. This, therefore, could lead to poor convergence performance from the learning algorithm. In such a scenario, it becomes crucial to perform the exploration.

The results show good performance with the proposed constrained lower confidence bound (CLCB) based optimization algorithm. As demonstrated before, the expected improvement (EI) based approach could potentially lead to a greedy search on the design space. The addition of noise could make it even worse.



Figure B.4: Performance of constrained Augmented Expected Improvement (CAEI) algorithm on engine calibration problem at 20% noise level



Figure B.5: Performance of Augmented Expected Improvement Matrix (AEIM) algorithm on engine calibration problem at 20% noise level



Figure B.6: Performance of constrained Lower Confidence Algorithm (CLCB) on engine calibration problem at 20% noise level

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] Wahid Ali, Mohd Shariq Khan, Muhammad Abdul Qyyum, and Moonyong Lee. Surrogateassisted modeling and optimization of a natural-gas liquefaction plant. *Computers & Chemical Engineering*, 118:132–142, 2018.
- [2] Bruce Ankenman, Barry L Nelson, and Jeremy Staum. Stochastic kriging for simulation metamodeling. *Operations research*, 58(2):371–382, 2010.
- [3] Benjamin Berger, Florian Rauscher, and Boris Lohmann. Analysing gaussian processes for stationary black-box combustion engine modelling. *IFAC Proceedings Volumes*, 44(1):10633– 10640, 2011.
- [4] Rik Blok. *lhsdesigncon*. (https://github.com/rikblok/matlab-lhsdesigncon), Retrieved July 13, 2021.
- [5] Carlos A Coello Coello and Margarita Reyes Sierra. A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In *Mexican international conference on artificial intelligence*, pages 688–697. Springer, 2004.
- [6] Indraneel Das and John E Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.
- [7] Francesco de Nola, Giovanni Giardiello, Alfredo Gimelli, Andrea Molteni, Massimiliano Muccillo, Roberto Picariello, and Diego Tornese. Reduction of the experimental effort in engine calibration by using neural networks and 1d engine simulation. *Energy Procedia*, 148:344–351, 2018.
- [8] Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001.
- [9] Kalyanmoy Deb, Rayan Hussein, Proteek Chandan Roy, and Gregorio Toscano-Pulido. A taxonomy for metamodeling frameworks for evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 23(1):104–116, 2018.
- [10] Gabriella Dellino, Paolo Lino, Carlo Meloni, and Alessandro Rizzo. Kriging metamodel management in the design optimization of a cng injection system. *Mathematics and Computers in Simulation*, 79(8):2345–2360, 2009.
- [11] Hermann-Josef Ecker, Markus Schwaderlapp, and Dave K Gill. Downsizing of diesel engines: 3-cylinder/4-cylinder. Technical report, SAE Technical Paper, 2000.
- [12] Michael TM Emmerich, Kyriakos C Giannakoglou, and Boris Naujoks. Single-and multiobjective evolutionary optimization assisted by gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, 2006.

- [13] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.
- [14] Paul Feliot, Julien Bect, and Emmanuel Vazquez. A bayesian approach to constrained singleand multi-objective optimization. *Journal of Global Optimization*, 67(1-2):97–133, 2017.
- [15] Alexander Forrester, Andras Sobester, and Andy Keane. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.
- [16] Jacob R Gardner, Matt J Kusner, Zhixiang Eddie Xu, Kilian Q Weinberger, and John P Cunningham. Bayesian optimization with inequality constraints. In *ICML*, pages 937–945, 2014.
- [17] Michael A Gelbart, Jasper Snoek, and Ryan P Adams. Bayesian optimization with unknown constraints. *arXiv preprint arXiv:1403.5607*, 2014.
- [18] Sebastian Rojas Gonzalez, Hamed Jalali, and Inneke Van Nieuwenhuyse. A multiobjective stochastic simulation optimization algorithm. *European Journal of Operational Research*, 2019.
- [19] Tobias Gutjahr, Thomas Kruse, and Thorsten Huber. Advanced modeling and optimization for virtual calibration of internal combustion engines. In *NDIA Ground Vehicle Systems Engineering and Technology Symposium*, 2017.
- [20] M Hafner, M Schüler, O Nelles, and R Isermann. Fast neural networks for diesel engine control design. *Control Engineering Practice*, 8(11):1211–1221, 2000.
- [21] José Miguel Hernández-Lobato, Michael A Gelbart, Ryan P Adams, Matthew W Hoffman, and Zoubin Ghahramani. A general framework for constrained bayesian optimization using information-based search. *The Journal of Machine Learning Research*, 17(1):5549–5601, 2016.
- [22] Hiro Hiroyasu, Haiyan Miao, Tomo Hiroyasu, Mitunori Miki, Jiro Kamiura, and Sinya Watanabe. Genetic algorithms optimization of diesel engine emissions and fuel efficiency with air swirl, egr, injection timing and multiple injections. SAE transactions on JSTOR, 112:1353– 1364, 2003.
- [23] Yoshihiro Hotta, Minaji Inayoshi, Kiyomi Nakakita, Kiyoshi Fujiwara, and Ichiro Sakata. Achieving lower exhaust emissions and better performance in an hsdi diesel engine with multiple injection. Technical report, SAE Technical Paper, 2005.
- [24] Deng Huang, Theodore T Allen, William I Notz, and Ning Zeng. Global optimization of stochastic black-box systems via sequential kriging meta-models. *Journal of global optimization*, 34(3):441–466, 2006.
- [25] Timothy Jacobs, Dennis N Assanis, and Zoran Filipi. The impact of exhaust gas recirculation on performance and emissions of a heavy-duty diesel engine. Technical report, SAE Technical paper, 2003.

- [26] Hamed Jalali, Inneke Van Nieuwenhuyse, and Victor Picheny. Comparison of krigingbased algorithms for simulation optimization with heterogeneous noise. *European Journal of Operational Research*, 261(1):279–301, 2017.
- [27] B Jayashankara and V Ganesan. Effect of fuel injection timing and intake pressure on the performance of a di diesel engine–a parametric study using cfd. *Energy Conversion and Management*, 51(10):1835–1848, 2010.
- [28] Yaochu Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. Swarm and Evolutionary Computation, 1(2):61–70, 2011.
- [29] Timothy V Johnson. Review of co2 emissions and technologies in the road transportation sector. *SAE International Journal of Engines*, 3(1):1079–1098, 2010.
- [30] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.
- [31] Gautam Kalghatgi. Is it really the end of internal combustion engines and petroleum in transport? *Applied energy*, 225:965–974, 2018.
- [32] Deb Kalyanmoy, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on evolutionary computation*, 6(2):182–197, 2002.
- [33] Andy J Keane. Statistical improvement criteria for use in multiobjective design optimization. *AIAA journal*, 44(4):879–891, 2006.
- [34] Joshua Knowles. Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.
- [35] KDA Kumar. Real-coded genetic algorithms with simulated binary crossover: Studies on multimodal and multiobjective problems. *Complex syst*, 9:431–454, 1995.
- [36] Ruixue C Li and Guoming G Zhu. A real-time pressure wave model for knock prediction and control. *International Journal of Engine Research*, page 1468087419869161, 2019.
- [37] Bo Liu, Hadi Aliakbarian, Soheil Radiom, Guy AE Vandenbosch, and Georges Gielen. Efficient multi-objective synthesis for microwave components based on computational intelligence techniques. In *Proceedings of the 49th annual design automation conference*, pages 542–548. ACM, 2012.
- [38] Bo Liu, Francisco V Fernández, Qingfu Zhang, Murat Pak, Suha Sipahi, and Georges Gielen. An enhanced moea/d-de and its application to multiobjective analog cell sizing. In *IEEE Congress on Evolutionary Computation*, pages 1–7. IEEE, 2010.
- [39] Bo Liu, Vic Grout, and Anna Nikolaeva. Efficient global optimization of actuator based on a surrogate model assisted hybrid algorithm. *IEEE Transactions on Industrial Electronics*, 65(7):5712–5721, 2018.

- [40] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.
- [41] Wenlong Lyu, Fan Yang, Changhao Yan, Dian Zhou, and Xuan Zeng. Multi-objective bayesian optimization for analog/rf circuit synthesis. In *Proceedings of the 55th Annual Design Automation Conference*, page 11. ACM, 2018.
- [42] Fabio Mallamo, M Badami, and F Millo. Application of the design of experiments and objective functions for the optimization of multiple injection strategies for low emissions in cr diesel engines. SAE transactions on JSTOR, pages 208–222, 2004.
- [43] Jacob D McDonald, Matthew D Reed, Matthew J Campen, Edward G Barrett, JeanClare Seagrave, and Joe L Mauderly. Health effects of inhaled gasoline engine emissions. *Inhalation toxicology*, 19(sup1):107–116, 2007.
- [44] Yifan Men. *Reaction-based modeling and control of an electrically boosted diesel engine*. PhD thesis, Michigan State University, 2019.
- [45] Yifan Men, Ibrahim Haskara, and Guoming Zhu. Multi-zone reaction-based modeling of combustion for multiple-injection diesel engines. *International Journal of Engine Research*, page 1468087418788488, 2018.
- [46] H. B. Nielsen, S. N. Lophaven, and J. Søndergaard. DACE a matlab kriging toolbox, 2002.
- [47] Michael A Osborne, Roman Garnett, and Stephen J Roberts. Gaussian processes for global optimization. In 3rd international conference on learning and intelligent optimization (LION3), volume 2009, 2009.
- [48] Torben Østergård, Rasmus Lund Jensen, and Steffen Enersen Maagaard. A comparison of six metamodeling techniques applied to building performance simulations. *Applied Energy*, 211:89–103, 2018.
- [49] A. Pal, L. Zhu, Y. Wang, and G. Zhu. Constrained surrogate-based engine calibration using lower confidence bound. *IEEE/ASME Transactions on Mechatronics*, pages 1–1, 2021.
- [50] Anuj Pal, Yan Wang, Ling Zhu, and Guoming G Zhu. Multi-objective surrogate-assisted stochastic optimization for engine calibration. *Journal of Dynamic Systems, Measurement, and Control*, 143(10):101004, 2021.
- [51] Anuj Pal, Ling Zhu, Yan Wang, and Guoming G Zhu. Multi-objective stochastic bayesian optimization for iterative engine calibration. In *2020 American Control Conference (ACC)*, pages 4893–4898. IEEE, 2020.
- [52] Seongeon Park, Jonggeol Na, Minjun Kim, and Jong Min Lee. Multi-objective bayesian optimization of chemical reactor design using computational fluid dynamics. *Computers & Chemical Engineering*, 119:25–37, 2018.
- [53] Matthieu Petelet, Bertrand Iooss, Olivier Asserin, and Alexandre Loredo. Latin hypercube sampling with inequality constraints. *AStA Advances in Statistical Analysis*, 94(4):325–339, 2010.

- [54] Andrea Piano, Federico Millo, Giulio Boccardo, Mahsa Rafigh, Alessandro Gallone, and Marcello Rimondi. Assessment of the predictive capabilities of a combustion model for a modern common rail automotive diesel engine. Technical Report 2016-01-0547, SAE Technical Paper, 2016.
- [55] Andrea Piano, Federico Millo, Francesco Sapio, and Francesco Concetto Pesce. Multiobjective optimization of fuel injection pattern for a light-duty diesel engine through numerical simulation. *SAE International Journal of Engines*, 11(6):1093–1108, 2018.
- [56] Victor Picheny, David Ginsbourger, Yann Richet, and Gregory Caplin. Quantile-based optimization of noisy computer experiments with tunable precision. *Technometrics*, 55(1):2–13, 2013.
- [57] Victor Picheny, Tobias Wagner, and David Ginsbourger. A benchmark of kriging-based infill criteria for noisy optimization. *Structural and Multidisciplinary Optimization*, 48(3):607–626, 2013.
- [58] Chao Qin, Diego Klabjan, and Daniel Russo. Improving the expected improvement algorithm. In *Advances in Neural Information Processing Systems*, pages 5381–5391, 2017.
- [59] Martinez-Botas Ricardo, Pesiridis Apostolos, and MingYang Yang. Overview of boosting options for future downsized engines. *Science China Technological Sciences*, 54(2):318–331, 2011.
- [60] Ilya O Ryzhov. On the convergence rates of expected improvement methods. *Operations Research*, 64(6):1515–1528, 2016.
- [61] Jerome Sacks, William J Welch, Toby J Mitchell, and Henry P Wynn. Design and analysis of computer experiments. *Statistical science*, pages 409–423, 1989.
- [62] L Shi and K Rasheed. A survey of fitness approximation methods applied in evolutionary algorithms. In *Computational intelligence in expensive optimization problems*, pages 3–28. Springer, 2010.
- [63] Akhilendra Pratap Singh, Anuj Pal, and Avinash Kumar Agarwal. Comparative particulate characteristics of hydrogen, cng, hcng, gasoline and diesel fueled engines. *Fuel*, 185:491–499, 2016.
- [64] András Sóbester, Stephen J Leary, and Andy J Keane. On the design of optimization strategies based on global response surface approximation models. *Journal of Global Optimization*, 33(1):31–59, 2005.
- [65] Ruitao Song, Gerald Gentz, Guoming Zhu, Elisa Toulson, and Harald Schock. A controloriented model of turbulent jet ignition combustion in a rapid compression machine. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 231(10):1315–1325, 2017.
- [66] Ruitao Song, Ravi Teja Vedula, Guoming Zhu, and Harold Schock. Optimal combustion phasing modeling and control of a turbulent jet ignition engine. *IEEE/ASME Transactions on Mechatronics*, 23(4):1811–1822, 2018.
- [67] Joshua Svenson and Thomas Santner. Multiobjective optimization of expensive-to-evaluate deterministic computer simulator models. *Computational Statistics & Data Analysis*, 94:250– 264, 2016.
- [68] A Sydbom, Anders Blomberg, S Parnia, Nikolai Stenfors, Thomas Sandström, and SE Dahlen. Health effects of diesel exhaust emissions. *European Respiratory Journal*, 17(4):733–746, 2001.
- [69] Matthew P Thiel, Adam E Klingbeil, and Rolf D Reitz. Experimental optimization of a heavyduty diesel engine using automated genetic algorithms. Technical Report 2002-01-0960, SAE Technical Paper, 2002.
- [70] Richard Fiifi Turkson, Fuwu Yan, Mohamed Kamal Ahmed Ali, and Jie Hu. Artificial neural network applications in the calibration of spark-ignition engines: An overview. *Engineering science and technology, an international journal*, 19(3):1346–1359, 2016.
- [71] David A Van Veldhuizen and Gary B Lamont. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary computation*, 8(2):125–147, 2000.
- [72] G Gary Wang and Songqing Shan. Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical design*, 129(4):370–380, 2007.
- [73] Lianhao Yin, Gabriel Turesson, Per Tunestål, and Rolf Johansson. Model predictive control of an advanced multiple cylinder engine with partially premixed combustion concept. *IEEE/ASME Transactions on Mechatronics*, 25(2):804–814, 2020.
- [74] Dawei Zhan, Yuansheng Cheng, and Jun Liu. Expected improvement matrix-based infill criteria for expensive multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 21(6):956–975, 2017.
- [75] Jianxia Zhang, Yizhong Ma, Taho Yang, and Lijun Liu. Estimation of the pareto front in stochastic simulation through stochastic kriging. *Simulation Modelling Practice and Theory*, 79:69–86, 2017.
- [76] Shupeng Zhang, Jie J Yang, and Guoming G Zhu. Lpv modeling and mixed constrained H_2/H_{∞} control of an electronic throttle. *IEEE/ASME Transactions on Mechatronics*, 20(5):2120–2132, 2014.
- [77] Jun Zhou, Xiaosi Su, and Geng Cui. An adaptive kriging surrogate method for efficient joint estimation of hydraulic and biochemical parameters in reactive transport modeling. *Journal of contaminant hydrology*, 216:50–57, 2018.
- [78] Eckart Zitzler and Lothar Thiele. Multiobjective optimization using evolutionary algorithms—a comparative case study. In *International conference on parallel problem solving from nature*, pages 292–301. Springer, 1998.