

MANAGEMENT OF UNCERTAINTY FOR FLEXIBLE PAVEMENT DESIGN
UTILIZING ANALYTICAL AND PROBABILISTIC METHODS

By

Jennifer Queen Retherford

Dissertation

Submitted to the Faculty of the
Graduate School of Vanderbilt University
in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

In

Civil Engineering

August, 2012

Nashville, Tennessee

Approved:

Professor Sankaran Mahadevan

Professor Mark McDonald

Professor P.K. Basu

Professor Mark Ellingham

Copyright © 2012 by Jennifer Queen Retherford

All Rights Reserved

DEDICATION

Mom, thanks for reminding me to jump.

Ron, thanks for reminding me to relax.

Toby, thanks for reminding me to be crazy.

Granny, Jackie, Kaydee, and Lily, thanks for reminding me to be a woman.

Kenzie, Quita, and Oakee, thanks for reminding me to be happy.

Much thanks to my softball families...it's the little things.

Many special thanks to all my professional mentors and academic mentors, too many to
be named but all are so important to me.

ACKNOWLEDGEMENTS

The work completed in this dissertation would not have been possible without the guidance of my wonderful committee members. Dr. McDonald, thank you for your support throughout my Vanderbilt experience. Your intelligence in this field is immense and you have been an excellent resource throughout my research. Dr. Basu, you have been my strongest advocate throughout my entire graduate experience and that was invaluable. Thanks for supporting me through this journey. Dr. Mahadevan, I have studied your professionalism and have learned more than academic knowledge in your presence. You have been a great mentor, through both direct and indirect ways. Dr. Ellingham, you too have been an example of excellence in teaching, knowledgeable researcher, and dear advisor. I appreciated your assistance both in and outside of your lectures and enjoyed your input in my graduate studies.

The Department of Civil and Environmental Engineering at Vanderbilt provided financial support during my dissertation, but more importantly, allowed me the opportunity gain valuable experience as a professor. Thank you for your support, both financially and in my professional development.

I would also like to thank the many graduate students and post-doctoral students who assisted me in my graduate work. Thank you to the Vanderbilt Risk and Reliability research group for your input on my research throughout my career at Vanderbilt. Special thanks to Dr. Vadiraj Hombal and Dr. Sirisha Rangavajhala for your direct assistance with my research. You two are incredible individuals!

TABLE OF CONTENTS

| | |
|--|-----|
| DEDICATION | iii |
| ACKNOWLEDGEMENTS | iv |
| LIST OF TABLES | ix |
| LIST OF FIGURES | xi |
| LIST OF ABBREVIATIONS..... | xii |
| I. INTRODUCTION..... | 1 |
| I.1 Motivation..... | 1 |
| I.2 Summary | 13 |
| I.3 Calibration, Selection, and Uncertainty Quantification for Permanent Deformation Predictive Models for Flexible Pavement..... | 14 |
| I.4 Surrogate Model Construction..... | 15 |
| I.5 Uncertainty Propagation for Mechanistic Empirical Pavement Design | 15 |
| I.6 Risk-Based Design for M-E Design of Flexible Pavement Design..... | 16 |
| I.7 Organization of Dissertation | 16 |
| II. BACKGROUND..... | 19 |
| II.1 AASHTO MEPDG | 20 |
| II.1.1 Terminal International Roughness Index (IRI) (Smoothness)..... | 23 |
| II.1.2 Asphalt Concrete Layer Fatigue Cracking (Alligator Cracking)..... | 25 |
| II.1.3 Permanent Deformation | 28 |
| II.2 Accelerated Pavement Testing (APT) & WesTrack..... | 29 |
| II.3 Methods for Surrogate Modeling..... | 33 |
| II.4 Analytical Reliability Methods for Pavement Design | 39 |
| II.5 Reliability Based Design Optimization | 49 |
| II.5.1 Optimization Problem Definitions..... | 49 |
| II.5.2 RBDO Problem Formulations..... | 51 |
| II.6 Discussion..... | 53 |
| III. CALIBRATION, SELECTION, AND UNCERTAINTY QUANTIFICATION FOR PERMANENT DEFORMATION PREDICTIVE MODELS FOR FLEXIBLE PAVEMENT..... | 55 |

| | | |
|---------|---|-----|
| III.1 | Introduction..... | 55 |
| III.2 | Background..... | 57 |
| III.3 | Construction of Predictive Models | 62 |
| III.3.1 | MEPDG Rutting Models..... | 63 |
| III.3.2 | Shear Theory Model | 66 |
| III.3.3 | Weighted Models | 67 |
| III.4 | Model Validation and Comparison..... | 69 |
| III.5 | Model Validation and Comparison Results | 73 |
| III.6 | Conclusion | 77 |
| IV. | SURROGATE MODEL INITIALIZATION: VARIABLE SELECTION PROCESS | 80 |
| IV.1 | Introduction..... | 80 |
| IV.2 | Surrogate Model Construction: Initialization | 81 |
| IV.2.1 | Quantity of Training Point Parameters (N_D)..... | 82 |
| IV.2.2 | Location of Training Points for Evaluation of Selection Processes | 87 |
| IV.3 | Selection Process Methods | 87 |
| IV.3.1 | ANOVA | 88 |
| IV.3.2 | Correlation Matrix | 90 |
| IV.3.3 | Gaussian Process Model Length-Scale Factors | 90 |
| IV.4 | Selection Process Comparison..... | 90 |
| IV.5 | Results..... | 91 |
| IV.5.1 | Sensitivity Analysis: Quantity of Training Point Parameters | 93 |
| IV.5.2 | Method for Selection of Training Point Parameters | 93 |
| IV.6 | Conclusion | 95 |
| V. | UNCERTAINTY PROPAGATION WITH SURROGATE MODELS | 97 |
| V.1 | Introduction..... | 97 |
| V.2 | Sources of Uncertainty..... | 100 |
| V.2.1 | Input Parameter Variability..... | 100 |
| V.2.2 | GP Model Predictive Uncertainty..... | 100 |
| V.2.3 | MEPDG Predictive Uncertainty | 101 |
| V.3 | Uncertainty Propagation and Sensitivity Analysis | 103 |

| | |
|---|-----|
| V.3.1 Uncertainty Propagation Method..... | 103 |
| V.3.2 Numerical Experiment..... | 106 |
| V.4 Conclusion..... | 120 |
| VI. ANALYTICAL RELIABILITY METHODS FOR MECHANISTIC-EMPIRICAL FLEXIBLE PAVEMENT DESIGN..... | 123 |
| VI.1 Introduction..... | 123 |
| VI.2 Distress Models for M-E Pavement Design..... | 124 |
| VI.3 Distributions of the Random Variables..... | 127 |
| VI.4 Numerical Results..... | 129 |
| VI.5 Implementation with the AASHTO MEPDG..... | 132 |
| VI.5.1 AASHTO MEPDG Prediction Equations..... | 132 |
| VI.6 Conclusion..... | 139 |
| VII. LRFD AND CORRECTION FACTORS FOR ROUTINE RELIABILITY ANALYSIS WITH THE MEPDG..... | 141 |
| VII.1 Introduction..... | 141 |
| VII.2 Inverse FORM..... | 142 |
| VII.3 Calculation of Load and Resistance Factors..... | 144 |
| VII.3.1 Distributions of the Random Variables..... | 145 |
| VII.3.2 GP Model Construction and Verification..... | 146 |
| VII.3.3 Calculation and Discussion of Load and Resistance Factors..... | 149 |
| VII.4 Correction Factors for AMV..... | 153 |
| VII.5 Conclusion..... | 154 |
| VIII. RISK-BASED DESIGN OPTIMIZATION METHOD UTILIZING M-E DESIGN EQUATIONS..... | 156 |
| VIII.1 Introduction..... | 156 |
| VIII.2 Selection of a Surrogate Model Type..... | 158 |
| VIII.3 Surrogate Model Construction: Adaptive Training Point Selection Process... 160 | |
| VIII.3.1 Quantity of Training Points (N_{TP})..... | 161 |
| VIII.4 Verification of the Surrogate Model..... | 164 |
| VIII.5 Results..... | 165 |
| VIII.5.1 GP Construction and Verification..... | 166 |

| | | |
|-------------|---|-----|
| VIII.5.2 | RBDO Solution | 169 |
| VIII.6 | Conclusion | 171 |
| IX. | CONCLUDING REMARKS | 172 |
| X. | FUTURE WORK | 176 |
| X.1 | Pay Factors and Performance Related Specifications..... | 176 |
| X.2 | Genetic Algorithms for GP Parameter Selection Process..... | 177 |
| X.3 | Additional MEPDG Distress Models & Various Pavement Structures..... | 177 |
| X.4 | Optimization Routine Improvement for Model Calibration | 178 |
| APPENDIX A: | GP Training Data Generation..... | 180 |
| APPENDIX B: | Matlab Numerical Computation Sample Codes..... | 214 |
| REFERENCES | | 271 |

LIST OF TABLES

| | |
|--|-----|
| TABLE II.1: Original Test Section Designations at WesTrack Experiment..... | 33 |
| TABLE III.1: Calibration Factors for MEPDG Predictive Models..... | 66 |
| TABLE III.2: Weight Coefficients for Weighted Models..... | 68 |
| TABLE III.3: Model Validation and Comparison Results..... | 74 |
| TABLE IV.1: MEPDG Input Parameter Ranges..... | 84 |
| TABLE IV.2: Training Point Parameters Using Correlation Matrix..... | 94 |
| TABLE V.1: Input Parameter Ranges..... | 107 |
| TABLE V.2: Probability Distributions of Random Variables..... | 108 |
| TABLE V.3: MEPDG and GP Model Uncertainty Distribution Parameters..... | 110 |
| TABLE V.4: Percent Contributions to Overall Variance..... | 114 |
| TABLE V.5: Probability of Failure of Distress Modes Including Sources of Uncertainty | 119 |
| TABLE VI.1: Probability Distributions of Random Variables..... | 127 |
| TABLE VI.2: Probability of Failure of Distress Modes and System according to various Reliability Methods..... | 129 |
| TABLE VI.3: Computation Effort of Reliability Methods Comparison..... | 131 |
| TABLE VI.4: Probability Distributions of Random Variables..... | 137 |
| TABLE VII.1: Probability Distributions of Random Variables..... | 146 |
| TABLE VII.2: Verification of Predictive Capability of GP Models..... | 148 |
| TABLE VII.3: Distribution of Bayes Factors for GP Models..... | 149 |

| | |
|---|-----|
| TABLE VII.4: Load and Resistance Factors and Parameter Offset Values for Terminal IRI Distress Model | 150 |
| TABLE VII.5: Load and Resistance Factors and Parameter Offset Values for AC Surface-Down Model..... | 150 |
| TABLE VII.6: Load and Resistance Factors and Parameter Offset Values for AC Bottom-Up Distress Model..... | 151 |
| TABLE VII.7: Load and Resistance Factors and Parameter Offset Values for AC Permanent Deformation Distress Model..... | 151 |
| TABLE VII.8: Load and Resistance Factors and Parameter Offset Values for Total Permanent Deformation Distress Model..... | 152 |
| TABLE VIII.1: GP Verification Results | 167 |
| TABLE VIII.2: Random Variable Statistics for Design Optimization Problem | 170 |
| TABLE VIII.3: Design Optimization Results | 170 |
| TABLE VIII.4: Design Optimization Results: Distress Modes..... | 170 |

LIST OF FIGURES

| | |
|---|-----|
| FIGURE I.1: Axial Strain Theory for Deformation of Asphalt Concrete (14)..... | 6 |
| FIGURE I.2: Shear Theory Deformation of Asphalt Concrete (15) (16)..... | 6 |
| FIGURE I.3: Flow Chart of Current MEPDG Design and Reliability Procedure..... | 9 |
| FIGURE I.4: MEPDG Method for Reliability Analysis for IRI Distress Mode | 10 |
| FIGURE I.5: M-E Design Procedure and Proposed Improvements (6) | 18 |
| FIGURE II.2: WesTrack Test Loop Layout | 31 |
| FIGURE II.3: WesTrack Truck Configuration..... | 32 |
| FIGURE III.1 NCHRP 455 Regression Analysis Procedure..... | 60 |
| FIGURE III.2: Probability Density Functions for Model Residuals | 75 |
| FIGURE III.3: Statistics for Model Validation..... | 77 |
| FIGURE IV.1: Adjusted R-Squared Values for the Variable Selection Processes for each Prediction Model..... | 92 |
| FIGURE V.1: Proposed Method of Design to Incorporate All Sources of Uncertainty... | 99 |
| FIGURE V.2: Family of CDFs for Distress Modes..... | 112 |
| FIGURE V.3: GP Prediction Contour Plots Evaluated at Means of Other Parameters.. | 116 |
| FIGURE VI.1: CDF Plots for AMV and MCS Results | 138 |
| FIGURE VII.1: Geometric Interpretation of Inverse FORM | 144 |
| FIGURE VIII.1: Improvement in Average GP Variance for Verification Points | 168 |

LIST OF ABBREVIATIONS

| | |
|---------|--|
| AADT: | Average Annual Daily Traffic |
| AADTT: | Average Annual Daily Truck Traffic |
| AASHO: | American Association of State Highway Officials |
| AASHTO: | American Association of State Highway and Transportation Officials |
| AC: | Asphalt Concrete |
| AMV: | Advanced Mean Value |
| ANOVA: | One-way Analysis of Variance |
| APT: | Accelerated Pavement Testing |
| AV: | Air Voids |
| CDF: | Cumulative Distribution Function |
| EBC: | Effective Binder Content |
| ESALs: | Equivalent Single Axle Loads |
| FHWA: | Federal Highway Administration |
| FORM: | First Order Reliability Method |
| GB: | Granular Base |
| GP: | Gaussian Process surrogate model |
| HMA: | Hot-mix Asphalt |
| IRI: | International Roughness Index |
| KKT: | Karush-Kuhn-Tucker |
| LHC: | Latin Hypercube |
| LSF: | Length Scale Factor |

| | |
|----------|--|
| LTPP: | Long Term Pavement Performance program |
| MCS: | Monte Carlo Simulation |
| M-E: | Mechanistic Empirical |
| MEPDG: | Mechanistic Empirical Pavement Design Guide |
| MPP: | Most Probable Point of Failure |
| MVFOSM: | Mean value First Order Second Moment |
| NCHRP: | National Cooperative Highway Research Program |
| PC: | Polynomial Chaos |
| PDF: | Probability Density Function |
| QC/QA: | Quality Control/Quality Assurance |
| RAP: | Recycled Asphalt Pavement |
| RBDO: | Risk-based Design Optimization |
| RBF: | Radial Basis Function |
| RD: | Rut Depth (Rutting Depth; Permanent Deformation) |
| RSS: | Residual Sum of Squares |
| RSST-CH: | Repeated Simple Shear Test at Constant Height |
| SSE: | Sum Squared Error |

CHAPTER I

INTRODUCTION

I.1 Motivation

America's highway infrastructure consists of approximately 8.5 million lane-miles of public roads and highways (1) and is approximately a 200 billion dollar investment per year for the U.S. government. Many states in the U.S., such as California, Illinois, Texas and Pennsylvania, have annual highway construction budgets exceeding one billion dollars. Monetary needs for improvement, maintenance, and expansion of the current highway system near the trillion dollar mark. In addition to the investment of capital by the government, the quality of roads nationally has a billion dollar impact on drivers. Lost time due to traffic congestion impacts the local and national economy and poor road conditions cost drivers in terms of repairs due to wear and road condition-related accidents. (2)

Pavement design has a critical impact on the performance of this billion dollar investment, yet typical design and construction processes often neglect impacts of uncertainty on predicted pavement performance. Pavement design is heavily influenced by variability in material properties, construction tolerances, and traffic and weather conditions, yet current design procedures are deterministic. Without appropriately accounting for uncertainty in pavement design, pavement design life and reliability level predictions can be inaccurate. Accurate predictions are necessary for appropriate

inclusion of initial and maintenance construction costs in local and federal budgets. Early failure requiring repair and replacement prior to the desired design life strain these agencies and can impact the budgets of other sponsored projects. Improvements in management of uncertainty in design, such as those proposed by this dissertation, will have significant impact on the billion dollar financial investment, both public and private, by providing designers tools for decision making. Optimal pavement design that incorporates uncertainty will result in more confident predictions of pavement life spans and maintenance schedules, reducing unexpected maintenance costs due to early failures.

The objective of this dissertation is to propose, demonstrate, and verify methods for management of uncertainty for pavement design utilizing analytical and probabilistic methods. Specifically, this dissertation develops a systematic and comprehensive approach to management of uncertainty in pavement design by quantifying model uncertainty for the permanent deformation predictive model, addressing computational cost of Mechanistic-Empirical (M-E) design by construction of a surrogate model, performing uncertainty propagation, and demonstrating risk-based design for flexible pavements under warranty.

In response to the ever growing need for accurate prediction of pavement performance, pavement design procedures are progressing from original design methods that relied solely on empirical data to M-E methods. Current pavement design practice, by numerous states, continues to be based on the AASHTO 1993 (3) empirical method; however, interest in M-E design procedures is increasing and many states have adopted or are in the process of adopting this method. The Asphalt Institute (4) and Shell Methods (5) were some of the earliest M-E methods introduced to the pavement design

community. More recently, the National Cooperative Highway Research Program (NCHRP) released the Guide for Mechanistic-Empirical Design (MEPDG) (6).

Implementation of the M-E design process is necessary in providing reliable designs. M-E design methods improve accuracy of performance predictions by incorporating mechanistic theory and are capable of predicting performance for new, novel mix designs. The AASHTO 1993, although a major historical milestone in understanding the behavior of highway pavements, is insufficient for the needs of pavement design today. The design equations are completely empirical, are based on only one type of sub-grade and specific pavement materials, and do not appropriately account for environmental effects on pavement performance. Small and Winston (7) and Madanat, Prozzi, and Han (8) have also shown that the equations are impacted by censoring bias. Furthermore, these design equations have been extrapolated to design for inputs far beyond those considered in the road test. As a result of these limitations, many pavement sections fail prematurely while other sections far outlive their design lives. M-E methods address the model error that exists in purely empirical design procedures by relating observed distresses to stresses and strains developed in the pavement structure. M-E design is being widely pursued because, with appropriate calibrations, M-E methods improve the accuracy of predicted pavement performance. This improvement is due to the detailed computational models which more realistically capture the physical processes through the mechanistic portion of the pavement analysis.

M-E design methods, although an improvement to purely empirical methods, do not eliminate the uncertainty in predicted pavement behavior models. Pavement analysis is impacted by uncertainty from a number of sources and this uncertainty must be

managed in design. Significant sources of uncertainty such as field variables, uncertainty in the predicted behavior models, and errors in these models are not currently accounted for, in the pavement design process, in an efficient and comprehensive way. Management of uncertainty due to the predictive models is necessary for properly understanding propagation of uncertainty and for performing sensitivity analysis, which leads to better pavement design decisions and cost effective designs. M-E pavement design considers structural and serviceability thresholds limits. The AASHTO M-E design method considers six distress modes relating to fatigue cracking, permanent deformation, thermal fracture, and smoothness. Calibration and validation of all of these models has been performed extensively by researchers. (6)

While many of the performance models have reached a level of accuracy appropriate for design, the permanent deformation models continue to inaccurately predict actual performance in the field. These inaccuracies are the result of a complex mechanistic behavior of the pavement system as well as the sensitivity of these models to variability in loading and climate. Permanent deformation is a highly researched topic; yet, the uncertainty of common permanent deformation models has not been quantified.

Permanent deformation significantly influences maintenance costs and schedules, is an easily measured quantity, and significant to the serviceability of a pavement section; yet, it has been difficult to accurately predict because the behavior phenomena are not extremely well understood. Permanent deformation is a critical distress model and the physical behavior of asphalt concrete pavements is very complex, governed by materials, climate, and traffic loads. Research focused on capturing the impact of these parameters on permanent deformation predictive models includes work by: Deacon et al. (9), Ali et

al. (10), El-Basyouny et al. (11), and many others; however, the work has resulted in a variety of permanent deformation models which have not converged to a single predictive model. The uncertainty in prediction arises from inadequacy of analysis models and lack of fit of distress prediction models. Investigation of the impact of model uncertainty on the predicted behavior of flexible pavements is necessary to develop a single predictive model capable of confidently predicting permanent deformation.

To date, research has resulted in the development of two primary prediction models for permanent deformation based on two different mechanistic behavioral theories. One theory, presented by the AASHTO Mechanistic Empirical Pavement Design Guide (MEPDG), for permanent deformation focuses on plastic vertical axial strains in the asphalt concrete (AC) layer and assumes that deformation is the result of axial compression of the AC layer. (6) (11) FIGURE I.1 demonstrates this model of deformation. California's Department of Transportation M-E procedure, CalME, follows a second theory, as illustrated in FIGURE I.2, which assumes deformation is the result of shear stresses in the asphalt layers. (12) (13)



FIGURE I.1: Axial Strain Theory for Deformation of Asphalt Concrete (14)

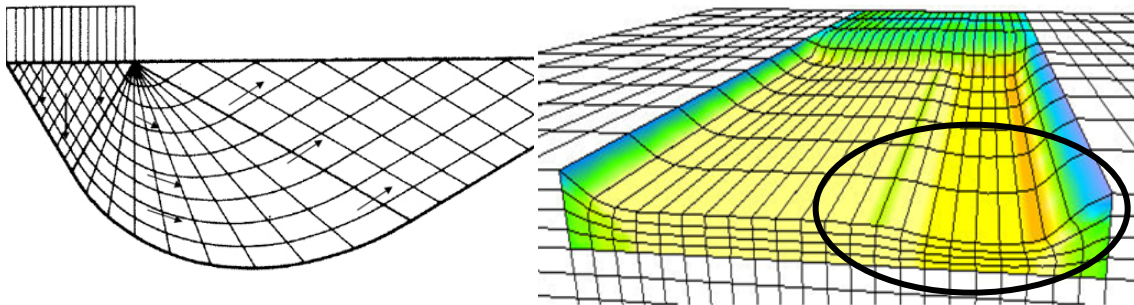


FIGURE I.2: Shear Theory Deformation of Asphalt Concrete (15) (16)

The M-E design methods offer engineers the opportunity to overcome the deficiencies in empirical design methods; still, these models rely heavily on empirical data and regression analysis to predict pavement performance. For example, the MEPDG prediction equation is based on a non-linear regression analysis of field data obtained through the Superpave Models Task C project (17), and the form of the regression equations is based on models previously developed by Kaloush (18) and Leahy (19). Although regression based on measured pavement behavior is a commonly accepted method for predicting model performance, understanding of the underlying physics

relating to pavement failure is important to developing accurate predictions of pavement performance. Historically, regression-based models based on empirical data provided acceptable accuracy in predicting pavement behavior; however, these models may no longer be suitable for use with novel mix designs and materials, such as those that incorporate polymer modified asphalts. Quantification of uncertainty in the predictive capability of these two models of permanent deformation is necessary to demonstrate the significance of model uncertainty on predicted pavement performance. Second, quantification of model uncertainty for these vastly different predictive models is necessary to state overall confidence in prediction of performance.

Permanent deformation is likely described by a model that incorporates both mechanistic theories; however little research exists in this area. A simplified approach to developing a model that incorporates both theories is through linear regression. Parameters describing the resistance of materials and structure to shear stresses can be related to calibration factors within the model that defines deformation by axial strain. Although this method may improve predictive capability, a more detailed model is likely required that incorporates the physics, through mechanistic equations, of each of these models. Models incorporating both mechanistic theories can be easily derived as weighted averages, in which the weights for each model are determined through analysis of the residuals of the independent models with experimental data. Model validation of each individual mechanistic theory and the combined models must be performed to determine the confidence in predictive capability of the models.

While empirically calibrated M-E design procedures reduce model error, their detailed models are computationally expensive to evaluate. A single, typical flexible

pavement analysis utilizing the MEPDG software requires approximately 30 minutes on a typical laptop computer. The computational expense is due to the structural analysis utilizing underlying multilayer analytical models and the well developed, yet cumbersome, climatic information. The iterative structural analysis is performed in hourly increments over the design life until the design satisfies defined threshold limits for all failure modes, which is very computationally intensive. Analyses requiring large numbers of M-E evaluations become impractical due to the computational expense. Recent research focused on highly iterative analyses such as sensitivity analysis and reliability analysis focus on implementation of sampling methods such as jack-knifing (20), Monte Carlo Simulation (MCS) (21) (22) (23), or Latin Hypercube (24) and ignore the computational expense associated with these methods. Use of macros, replacing M-E design software with highly complex computations often evaluated by super computers, are common methods for disregarding the impact of these methods, but this is not a suitable solution for practical applications. Alternatively, simplified prediction models or modified sampling techniques have been developed for reduction in computational expense; however these methods also result in reduction in model confidence. Simplified prediction models introduce approximation errors that can be difficult to quantify and reduced sampling techniques do not guarantee accuracy over the entire design space. The development of a well-trained surrogate model will address the computational expense of the M-E design procedure without reducing model confidence. Accurately trained and validated surrogate models provide high quality data because the sampling techniques, such as MCS, can be performed in a computationally efficient manner.

Propagation of uncertainty through probabilistic methods for pavement design within the M-E design procedure has been a widely researched topic; yet the computational expense has been a constant hindrance. Uncertainty propagation is necessary to overall quantification and management of uncertainty in pavement design. Initial recommendations for reliability analysis within the MEPDG prescribed MCS as the most appropriate method; however, a closed-form method was chosen due to the computational cost (6). Briefly, the MEPDG design process considers only deterministic input parameters and calculates a mean distress prediction each failure mode, for each month, for the entire design life of the pavement. (FIGURE I.3) The user defines a threshold limit for each distress mode, and the reliability is calculated assuming normal distribution, based on the calculated mean and variance derived from empirical data (See also FIGURE I.4).

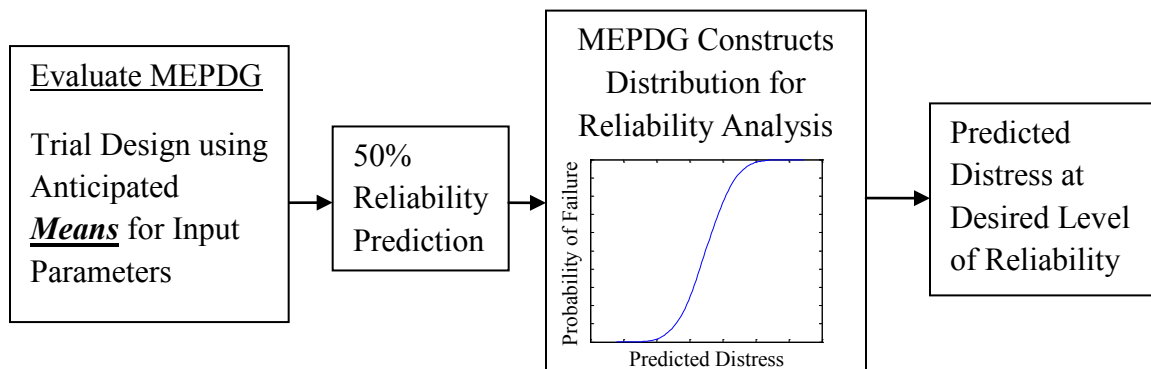


FIGURE I.3: Flow Chart of Current MEPDG Design and Reliability Procedure

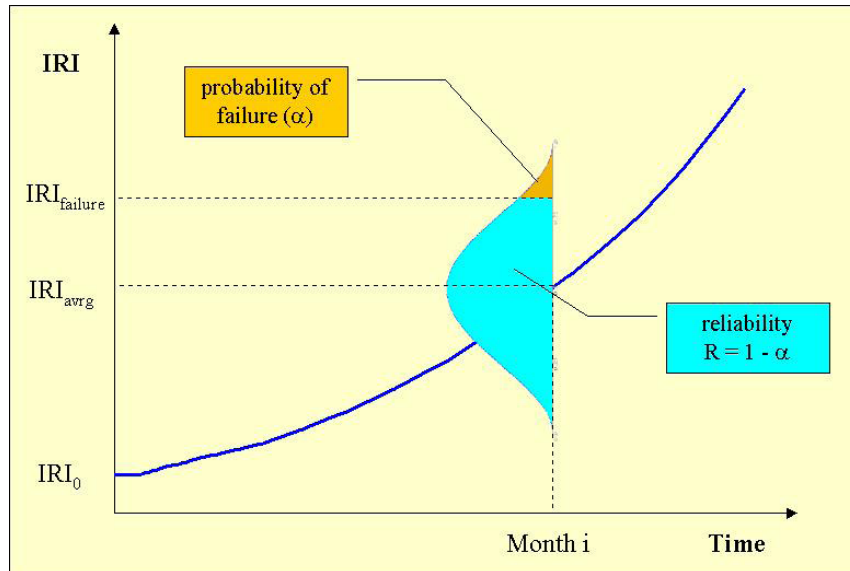


FIGURE I.4: MEPDG Method for Reliability Analysis for IRI Distress Mode

Several researchers have criticized the current method for reliability analysis in M-E design and research continues in search of a more robust method. (23) (25) (26) Darter, et al, (23) discuss the impracticality of simulation due to the large number of variables required for analysis in the MEPDG and similar M-E procedures. They discuss a Monte Carlo Simulation based technique as a less computationally expensive approach to reliability analysis for rigid pavement. Darter et al. state that the significant source of computational expense is the incremental design procedure that requires structural analysis producing “hundreds of thousands of stress and deflection calculations to compute monthly damage”. To reduce computational expense, they developed a neural network to perform the structural analysis for rigid pavements, but were unsuccessful in development of a similar neural network for flexible pavements. Although their technique is shown to save computational time for rigid pavements, they indicate that this method is

still quite time consuming and allude to future computer hardware improvements as the ultimate key in full implementation of their method with flexible pavements.

Implementation of MCS with a well-trained surrogate model addresses the computational expense associated with simulation based methods. Construction of a “cheap to evaluate” surrogate model improves computational speed of simulation methods, but introduces a model error that must be addressed. An alternative to simulation based techniques for performing reliability analysis is analytical approximation techniques such as: Mean value First Order Second Moment (MVFOSM) (27), First Order Reliability Methods (FORM) (28) (29), Rosenblueth (30), and Advanced Mean Value (AMV) (31). These more advanced statistical methods allow reliability analysis to be efficiently performed directly in the M-E design procedures, but may lose accuracy due to approximations in the limit state equation when searching for the most probable point of failure.

Accurate and efficient methods for reliability analysis are necessary for management of uncertainty in M-E pavement design. Comparison of the computational expense and accuracy of reliability analysis utilizing a surrogate model with simulation and M-E design procedures with analytical methods is necessary to determine the computational trade-off between these options. Although a surrogate model will reduce computational time, simulation based methods for reliability analysis may still be significantly expensive. Similarly, analytical methods are significantly less computationally expensive than simulations, yet the M-E design procedure is time consuming. Accuracy must also be verified for each method, as the surrogate model or the analytical reliability methods are sources of uncertainty.

Management of uncertainty is necessary for optimal pavement design and life cycle cost analyses. (32) Design optimization is specifically important to quality control and assurance (QC/QA) efforts by contractors, which is increasingly important as many new construction projects require contractors to provide extended warranties for pavement projects. Optimization over an expected design life provides the information necessary for contractors to determine initial construction design and maintenance schedules. Several optimization methods for flexible pavement design can be found in literature; however, many of the methods are based on empirical analysis methods. Prozzi et al. (8) introduce an optimization method; however, they develop performance models based on the empirical data obtained from the AASHO Road Test data. This empirical approach does not incorporate all input parameters that contribute to pavement performance, such as climate and material strengths. As previously discussed, M-E design procedures have been developed and improve the accuracy of performance predictions and are therefore more appropriate for use in optimal design.

Optimization of pavement designs, however, is not common practice as it requires additional computational expense and requires designers to perform the optimization routine outside the framework of M-E procedures. Most literature discussing optimization procedures for pavement design focus on the optimization routine and incorporate simplified pavement analysis methods. Mamlouk, et al., (33) introduce a method for optimization of flexible pavements utilizing dynamic programming in conjunction with two pavement design models for use in project-level pavement management. The design models were incorporated within the computer program specifically developed for optimization and required rewriting a commonly utilized

multilayer elastic system model. Implementation of this method utilizing the MEPDG outside the framework of the optimization routine would be computationally expensive, and rewriting this code would be prohibitive for practical users. Grivas et al. (34) completely exclude formal pavement distress models in their optimization method and incorporate a simplified approach to determining distress in pavement based on only three input parameters: pavement type, traffic volume, and distress measures. Further, currently recommended procedures, such as those by (33) (34) (8), do not incorporate reliability in the optimization process. Exploiting a well trained surrogate model makes design optimization utilizing M-E procedures computationally affordable and provides the framework for reliability analysis.

I.2 Summary

Ultimately, quantification and management of all uncertainty implemented within the M-E design process is necessary for optimal flexible pavement design. Quantification of uncertainty through MCS or other simulation techniques is difficult, primarily due to the computational expense of the M-E method. This computational expense must be addressed to perform uncertainty quantification utilizing simulation-based techniques, or analytical methods must be verified for application with the M-E procedure. Uncertainty quantification is necessary in investigating and addressing the sources of uncertainty. Further, at the present time, no comprehensive approach to uncertainty management has been proposed for M-E pavement design. Past research efforts in reliability analysis have been dedicated to only one type of uncertainty or another.

Management of uncertainty for M-E pavement design utilizing analytical and probabilistic methods requires quantification of model uncertainty, must address the computational expense associated with M-E design, and should incorporate reliability based design optimization. The permanent deformation model is significantly susceptible to errors in predicted performance due to model uncertainty. In addition to model uncertainty, input variability has a critical impact on pavement performance. A logical approach to incorporating both sources of uncertainty is presented in this dissertation. A framework for risk-based design is developed integrating the uncertainty propagation and impact of model uncertainty on predicted pavement performance.

This dissertation develops a systematic and comprehensive approach to management of uncertainty by accomplishing four major objectives: address model uncertainty for the permanent deformation model, develop a method to reduce computational expense, design a framework for incorporation of uncertainty in pavement design, and demonstrate a framework for risk-based M-E pavement design. Implementation of these four major objectives within the context of M-E pavement design is outlined in FIGURE I.5 and further discussed in the following sections.

I.3 Calibration, Selection, and Uncertainty Quantification for Permanent Deformation Predictive Models for Flexible Pavement

The proposed framework for management of uncertainty in flexible pavement design begins with investigation of model form error for the performance prediction models. The permanent deformation models are particularly susceptible to model form error due to the

complexities in mechanistic behavior of the layered pavement structure and complications in measurement of deformation for sub-grade and base layers. Chapter III of this dissertation investigates both the plastic axial strain theory and shear theory for permanent deformation and quantifies the model uncertainty of each of these models. Model averaging and model calibration is used to develop models that incorporate both theories to determine the impact of modeling permanent deformation utilizing both theories.

I.4 Surrogate Model Construction

Development of a surrogate model allows for computationally efficient probabilistic design for flexible pavements, but requires training and verification with respect to the model that is it replacing. The AASHTO MEPDG is the most widely utilized M-E design procedure in the U.S. and incorporates extensive climatic and empirical performance data. Inclusion of such extensive data hinders the computational efficiency of design with the MEPDG and likely includes design parameters of little significance to pavement performance. Chapter IV includes investigation of the required quantity of input parameters necessary to accurately imitate the MEPDG design procedure. Construction and verification of the surrogate model is discussed in detail in Chapters V and VIII.

I.5 Uncertainty Propagation for Mechanistic Empirical Pavement Design

Management of uncertainty from all significant sources is necessary to accurately predict pavement performance. Sources of uncertainty such as model form error and input

variability turn a deterministic design process into a stochastic design process. A systematic approach to uncertainty propagation is lacking in current design procedures and is necessary for accurate reliability predictions. Uncertainty propagation in Chapter V of this dissertation is performed utilizing two approaches: a surrogate model with a simulation based reliability analysis method and M-E predictive models with analytical reliability methods. These methods are compared for accuracy and computational effort. Additional concepts describing practice-ready procedures are presented in Chapter VII.

I.6 Risk-Based Design for M-E Design of Flexible Pavement Design

Ultimately, designers are in need of a risk-based design procedure that implements the uncertainty management concepts demonstrated by the first major objectives of this dissertation. Demonstration of a reliability-based design optimization routine is presented in this dissertation in Chapter VIII. This investigation also extends the discussion of surrogate model construction by presenting a method for efficiently and effectively choosing the quantity and location of training points for construction of the surrogate model.

I.7 Organization of Dissertation

The organization of the dissertation is as follows. The first major objective, addressing uncertainty in the permanent deformation models, is presented in Chapter III. A thorough calibration process is performed on three prevalent models: a model incorporating shear theory, an axial strain model, and a model combining both mechanistic theories. Model

validation and comparison is performed to determine the accuracy of these models compared to experimental results from the WesTrack experiment. Understanding the uncertainty associated with these permanent deformation models is necessary to accurately predict pavement performance, but current models are computationally expensive. A surrogate model is constructed in Chapters IV, V, and VIII that accurately emulates the AASHTO MEPDG. Chapter IV discusses the selection process for training data required for construction of surrogate models specific to M-E pavement design. The third major objective develops a logical and efficient process for incorporating uncertainty into M-E pavement design. A systematic method for uncertainty propagation is presented in Chapter V, analytical reliability methods are presented in Chapter VI, and a method for developing load and resistance factors for design is presented in Chapter VII. Propagation of uncertainty from input variability, the MEPDG prediction models, and the surrogate model is demonstrated and a sensitivity analysis is performed. Analysis of methods for the selection of the quantity and location of training points for the surrogate model is presented. Reliability analysis is performed utilizing probabilistic and analytical methods. A method for developing load and resistance factors is presented as a practice-ready option for reliable pavement design. The final major objective is addressed in Chapter VIII which presents a framework for risk-based design in the context of M-E pavement design. Through these four major objectives, this dissertation presents a comprehensive framework for management of uncertainty in flexible pavement design.

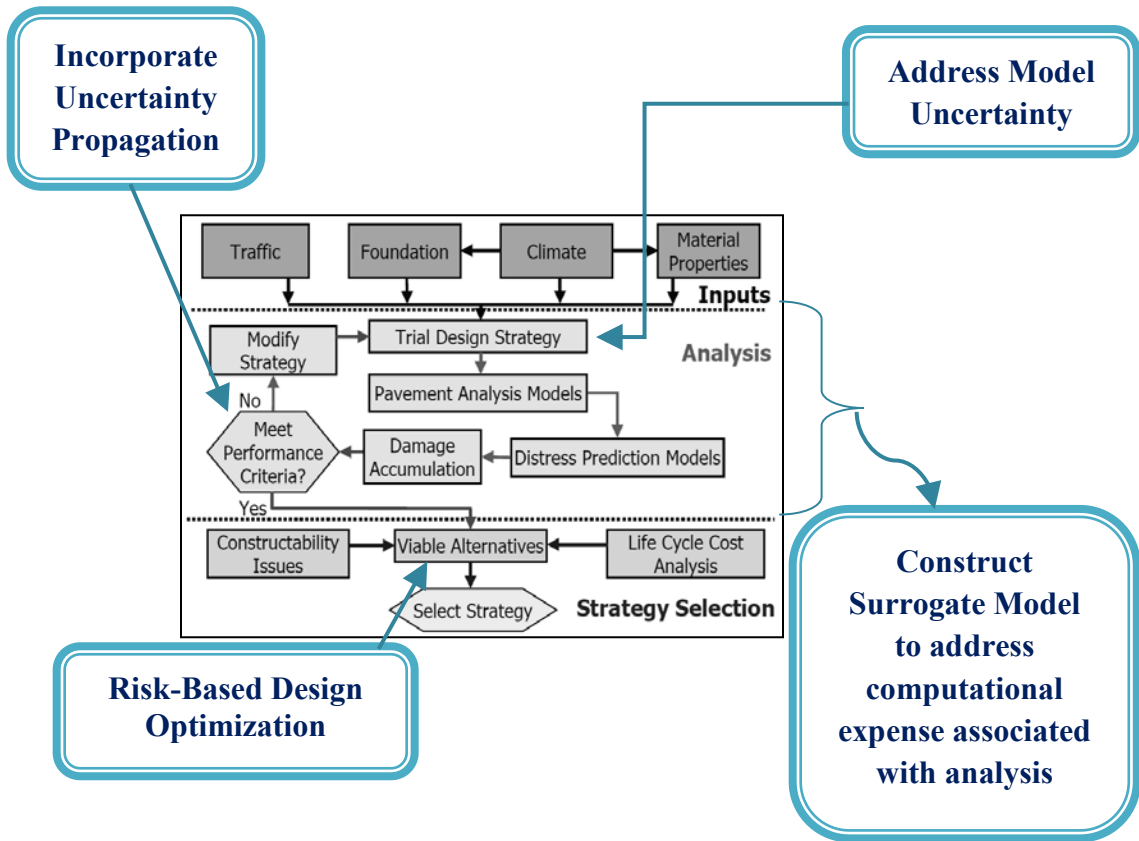


FIGURE I.5: M-E Design Procedure and Proposed Improvements (6)

CHAPTER II

BACKGROUND

Management of uncertainty for flexible pavement design requires an understanding of basic concepts related to pavement design, reliability methods, and design optimization. The research presented in this dissertation leverages past work in mechanistic-empirical pavement design and accelerated pavement testing. In particular, the American Association of State Highway and Transportation Officials' (AASHTO) Mechanistic Empirical Pavement Design Guide (MEPDG) and data from the WesTrack experiment, a specific accelerated pavement testing experiment conducted in the late 1990s, are utilized in this research. Analytical and probabilistic reliability methods including Mean Value First Order Second Moment (MVFOSM), First Order Reliability Methods (FORM), Rosenblueth, Advanced Mean Value (AMV), and Monte Carlo Simulation (MCS) are utilized and brief summaries of the theory are presented. Ultimately, the dissertation presents a method for incorporating design optimization into the pavement design process and background information regarding concepts related to problem formulation and reliability based design optimization methods are discussed.

II.1 AASHTO MEPDG

The MEPDG is the most comprehensive implementation of the mechanistic-empirical pavement design procedure to date. AASHTO released the most recent version of the MEPDG document in 2004 and, at the same time, released a software package implementing the design methods presented in the documentation. The combination of the design guide and software provides an excellent guide to design of flexible pavements in accordance with many of the nationally accepted procedures and practices. The procedure presented in the MEPDG is a mechanistic-empirical design procedure that produces predictions in the performance of a pavement according to standardized performance criteria.

Design inputs for the MEPDG include traffic data, material properties, and climatic data. The process has been developed according to three levels of design, each of varying levels of refinement relating to the input information. The Level 1 analysis is defined specifically for each input family and represents the most thorough understanding of the site characteristics. The Level 1 definition for traffic inputs is: “There is a very good knowledge of past and future traffic characteristics.” Level 2 analyses represent modest knowledge of the characteristics of the input parameters and are defined for the material input as those that are “estimated through correlations with other material properties that are measured in the laboratory or field”. Level 3 analysis represents the level with least confidence in accuracy and is utilized when significant estimation of design parameters is necessary. Design for pavements with poor knowledge of the traffic conditions or material characterization without sufficient testing constitutes a Level 3 design level. (6)

Significant traffic data is required regardless of the design level selected. Traffic data for design includes yearly truck-traffic volume, traffic speed, truck-traffic directional and lane factors, vehicle class distributions, axle load and tire information, and traffic growth projections. Although databases such as the Long Term Pavement Performance (LTPP) database include extensive traffic data, the resulting traffic characterizations may not accurately represent the current or future projections for new roads. Additionally, many state departments do not have the resources to acquire more current data or data in locations not registered in the LTPP. Nationally developed standards for traffic inputs have been presented in the MEPDG, but designers are penalized in the reliability analysis by the selection of the Level 3 designation which are likely less accurate than Level 1 or 2 analyses.

Material inputs for the MEPDG vary between bound and unbound material types, but generally include layer thickness, unit weight of the material, tensile and/or compressive strength parameters, thermal properties, shrinkage, and, when applicable, gradation information. The material characterization in the MEPDG has been specialized to allow designers to utilize nationally calibrated models or regionally calibrated models for parameters such as the dynamic modulus and viscosity for hot-mix asphalt (HMA). The design software provided by AASHTO also includes all information regarding Superpave mix designs, one of the most common national standards for mix designs.

The third major category of MEPDG input parameters is that which describes the environmental and climatic data for the pavement site. Climate inputs are detailed by month or hour and include: temperature, rainfall, wind, and conditions such as sun and freeze. Development of software integrating FHWA's comprehensive database allows for

inclusion of site-specific climatic data in a manner easy to end-users of the MEPDG design software. Integration of this database contributes significant amounts of information to the design process, improving the accuracy in performance predictions through freeze and thaw cycles over the design life of the pavement.

The MEPDG design process utilizes material properties, traffic data, and climatic information to quantify performance in terms of stress, strains, and displacements within the pavement. The most critical mechanistic properties include the horizontal tensile strain in the HMA layer, compressive vertical stresses and strains at mid-height of all layers, and compressive vertical strains and stresses at the top of the sub-grade. These values correspond to HMA fatigue cracking, HMA and total depth rutting, and sub-grade rutting, respectively. The mechanistic properties are incrementally calculated and accumulated over the design life of the pavement. At each increment, the distress models are evaluated to determine the performance criteria.

The design process is an iterative process that requires selection and refinement of design parameters until specified performance criteria are met. AASHTO has included six significant performance criteria in the design process: permanent deformation (rutting) of both the top layer and the entire pavement structure, both bottom-up and top-down fatigue cracking of the asphalt concrete layer, thermal cracking, and smoothness (IRI). These performance criteria have been chosen to best represent both the structural and serviceability requirements necessary for acceptable performance of a flexible pavement system.

The design process requires satisfying all individual performance criteria, or distress modes, as a series system, where failure of any criterion is a failure of the pavement design. The design process requires the engineer to select both a threshold value and reliability level for each of these six performance criteria over a design life. Recommended threshold values are given in the AASHTO Guide; however, little guidance is given on the required level of reliability. The decision of a predicted design life is another very significant decision by the designer that significantly impacts the criteria for acceptable pavement performance.

Development of the distress models for the MEPDG was performed by AASHTO utilizing national LTPP data and laboratory experimental results, constituting the empirical basis of the models, and mechanistic theory, typically assuming linear elastic behavior of the materials. To be concise, a brief description of the distress models follow; additional discussion of the derivation of the distress models is available in the MEPDG.

II.1.1 Terminal International Roughness Index (IRI) (Smoothness)

The Terminal IRI prediction model quantifies smoothness and is a serviceability requirement dependent on the initial as-built profile of the pavement section and the progression of structural distresses over the design life. Smoothness of a pavement structure is critical not only to driver comfort, but to operating costs and travel times. The model uses the predictions from the rutting, bottom-up, and top-down models, as well as site parameters (climate, sub-grade properties, etc.) to predict the smoothness over time.

The predictive model for Terminal IRI for new AC pavements over unbound aggregate bases is shown in Equation II.1. The empirical model is the result of several research studies and incorporates many of the structural distress models.

$$IRI = IRI_0 + 0.0463 \left[SF \left(e^{\frac{age}{20}} - 1 \right) \right] + 0.00119(TC_L)_T + 0.1834(COV_{RD}) + 0.00384(FC)_T + 0.00736(BC)_T + 0.00115(LC_{SNWP})_{MH} \quad (II.1)$$

Where:

IRI_0 : Initial IRI, m/km.

SF : Site Factor (function of site climatic and sub-grade information).

$e^{\frac{age}{20}} - 1$: Age Term, (where age is expressed in years).

COV_{RD} : Coefficient of variation of the rut depths, percent.

$(TC_L)_T$: Total length of transverse cracks, m/km.

$(FC)_T$: Fatigue cracking in wheel path, percent total lane area.

$(BC)_T$: Area of block cracking as a percent of total lane area.

$(LC_{SNWP})_{MH}$: Length of moderate and high severity sealed longitudinal cracks outside wheel path, m/km.

II.1.2 Asphalt Concrete Layer Fatigue Cracking (Alligator Cracking)

Fatigue cracking in the AC layer of the pavement structure is estimated by the bottom-up (alligator cracking) and surface-down (longitudinal cracking) fatigue distress models. Both models follow similar form in the MEPDG, defining fatigue as a function of tensile strain and mix stiffness (modulus). The bottom-up model evaluates the distress considering a critical location at the bottom of the AC layer, resulting in a crack that propagates from the bottom towards the top of the layer. The surface-down model considers a crack that develops at the surface of the AC layer and grows down through the layer.

Prediction of the fatigue cracking is performed in the MEPDG according to Miner's Law which defines fatigue as an accumulation of damage due to traffic repetitions over a design period. Calculation of the fatigue damage according to Miner's law requires prediction of the number of load repetitions to failure, N_f . The common form for this calculation is shown in two equivalent formulations, the Asphalt Institute model (Eq. II.2) and a nationally calibrated equation (Eq. II.3).

$$N_f = C k_1 \left(\frac{1}{\varepsilon_t} \right)^{k_2} \left(\frac{1}{E} \right)^{k_3} \quad (\text{II.2})$$

$$N_f = \beta_{f1} k_1 (\varepsilon_t)^{-\beta_{f2} k_2} (E)^{-\beta_{f3} k_3} \quad (\text{II.3})$$

Where:

N_f : Number of repetitions to fatigue cracking.

ε_t : Tensile strain at the critical location. (in./in.)

E : Stiffness of the material. (psi)

k_1, k_2, k_3 : Laboratory regression coefficients.

$\beta_{f1}, \beta_{f2}, \beta_{f3}$: Calibration parameters.

C : Laboratory to field adjustment factor.

Implementation for the MEPDG incorporates results from the Asphalt Institute model, in which the laboratory regression coefficients, k_1, k_2, k_3 , are taken as 0.00432, 3.291, and 0.854, respectively. The adjustment factor is defined as a function of asphalt binder content (V_b) and air voids (V_a) as given in Equation II.4.

$$C = 10^M \quad (\text{II.4})$$

Where:

$$M = 4.84 \left(\frac{V_b}{V_a + V_b} - 0.69 \right) \quad (\text{II.5})$$

The nationally calibrated model is described with parameters specific to either fatigue model as described in Equations II.6 through II.8.

$$N_f = 0.00432k'_1C \left(\frac{1}{\varepsilon_t}\right)^{3.9492} \left(\frac{1}{E}\right)^{1.281} \quad (\text{II.6})$$

$$\text{Bottom-Up Cracking:} \quad k'_1 = \frac{1}{0.000398 + \frac{0.003602}{1 + e^{(11.02 - 3.49h_{ac})}}} \quad (\text{II.7})$$

$$\text{Top-Down Cracking:} \quad k'_1 = \frac{1}{0.01 + \frac{12}{1 + e^{(15.676 - 2.8186h_{ac})}}} \quad (\text{II.8})$$

Where h_{ac} is the total thickness of the asphalt layer(s) measured in inches.

The fatigue cracking models utilized in the MEPDG are given in Equations II.9 and II.10.

$$FC_{bottom} = \left(\frac{6000}{1 + e^{(C_1 C'_{11} + C_2 C'_{22} \log_{10}(D * 100))}} \right) * \left(\frac{1}{60} \right) \quad (\text{II.9})$$

$$FC_{top} = \left(\frac{1000}{1 + e^{(7.0 + 3.5 \log_{10}(D * 100))}} \right) * (10.56) \quad (\text{II.10})$$

Where:

FC_{bottom} : Bottom-Up fatigue cracking, percent lane area

D : Bottom-Up fatigue damage by Miner's Law: $D = \sum_{i=1}^{DesignLife} \left(\frac{n_i}{N_i} \right)$

n_i : Actual traffic for time period i .

C_1 : 1.0

C'_1 : $-2C'_2$

C_2 : 1.0

C'_2 : $-2.40874 - 39.748 * (1 + h_{ac})^{-2.856}$

II.1.3 Permanent Deformation

The MEPDG predicts permanent deformation for the asphalt concrete layer and for the total pavement section as the sum of the product of plastic strains and the layer height Equation II.11. The equation for plastic strain is calculated as shown in Eq. II.12.

$$RD = \sum_{i=1}^{n_{sublayers}} \varepsilon_p^i h^i \quad (II.11)$$

$$\frac{\varepsilon_p}{\varepsilon_r} = k_z \beta_{r1} 10^{k_1 T^{k_1} \beta_{r2} N^{k_3} \beta_{r3}} \quad (II.12)$$

Equation II.11 expresses the permanent deformation (RD) as a function of plastic strain and layer height where strains accumulate across all sub-layers of the pavement structure. In Equation II.12, ε_p and ε_r are plastic and resilient strain respectively, and the k and β values are the regression coefficients and calibration factors. The regression coefficients are derived from non-linear regression based on the NCHRP 9-19 Superpave Experiment and the national calibration factors are derived from LTPP sections located in 28 different states. (6)

II.2 Accelerated Pavement Testing (APT) & WesTrack

Extensive data regarding material properties, traffic loadings, climate impacts, and measurements of performance are necessary for statistical analysis investigating the accuracy of prediction models to actual pavement performance. Current flexible design procedures rely on accurate empirical data to predict performance. Coupled with mechanistic theory, M-E design procedures must be validated to determine the accuracy of predictions through the life of the pavement. Typical pavement design life spans decades, but experimental data to support the performance of pavement over this length of time is difficult. Research facilities rarely have the resources available to evaluate test sections for such lengths of time. Further, inaccuracies exist in measurements due to changes in equipment and personnel. To address this issue, researchers have developed accelerated pavement testing (APT) procedures to expedite the acquisition of performance data. Experiments such as WesTrack include extensive data with traffic load repetitions nearing those expected over entire design life spans for typical pavement structures.

APT procedures allow for investigation of pavement performance after significant load repetitions in a concise time frame. Numerous facilities exist across the U.S. capable of performing APT experiments. Facilities such as those at Caltrans and Kansas State University perform experiments on a small scale utilizing specialized simulation equipment to imitate traffic loads. Other APT facilities, such as those at MnROADS, NCAT, and WesTrack, have constructed full scale pavement test tracks and perform experiments utilizing actual vehicles. Each style of APT has its advantages and disadvantages. The controlled environment in the simulation facilities reduces impacts of climate and improves (reduces) measurement errors. Conversely, test tracks reduce model error that may exist due to simulation of traffic and/or climatic effects.

The WesTrack accelerated pavement testing experiment is one of the most well known and well documented pavement performance experiments. Funded by the Federal Highway Administration (FHWA), the experiment was performed at a test track constructed in Nevada. The 2.9 km oval track (FIGURE II.1) completed constructed in 1995 and APT was performed from March 1996 to February 1999. The objectives for the experiment at this facility were to continue development of performance-related specifications for HMA pavements and to provide performance data on Superpave mix design procedures. The comprehensive report includes well documented materials, traffic, and climate parameters as well as performance data making this experiment an excellent source for research related to pavement performance prediction equations. Most of the individual testing activities at WesTrack were funded through the National Cooperative Highway Research Program and are readily available in published NCHRP Reports.

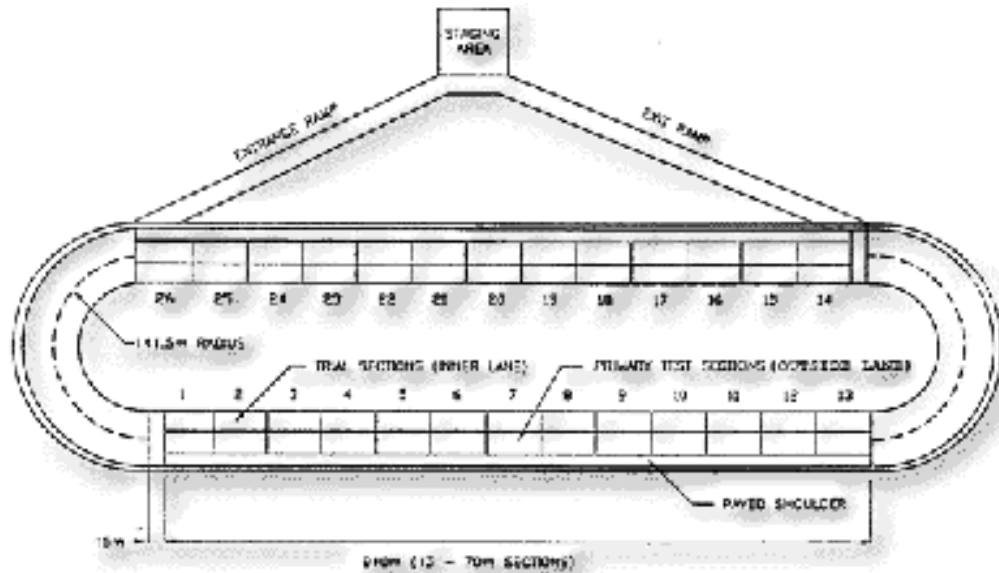


FIGURE II.1: WesTrack Test Loop Layout

Trafficking experiments were performed at WesTrack utilizing four triple-trailer combinations (FIGURE II.2) utilizing driver-less vehicle technology which provided consistency in speed and driving performance as well as job site safety. The trucks operated at a speed of 64 kph daily for up to 22 hours per day. Approximately 5 million equivalent single-axle loads (ESALs) were measured over the entire experiment. Performance monitoring for permanent deformation, fatigue cracking, and smoothness was performed throughout the trafficking experiment. Rut depth measurements were performed bi-weekly with the “Dipstick” and a laser device developed by NATC. Fatigue cracking was recorded by visual inspection surveys every two weeks, typically, and more frequently when traffic loading was increased or when rapid development of fatigue

cracking was witnessed. Profile measurements were performed to detect distresses such as longitudinal cracking and differences in rutting across the width of the track. During trafficking, weather data was recorded and post mortem sampling and testing was performed after traffic loading ceased.

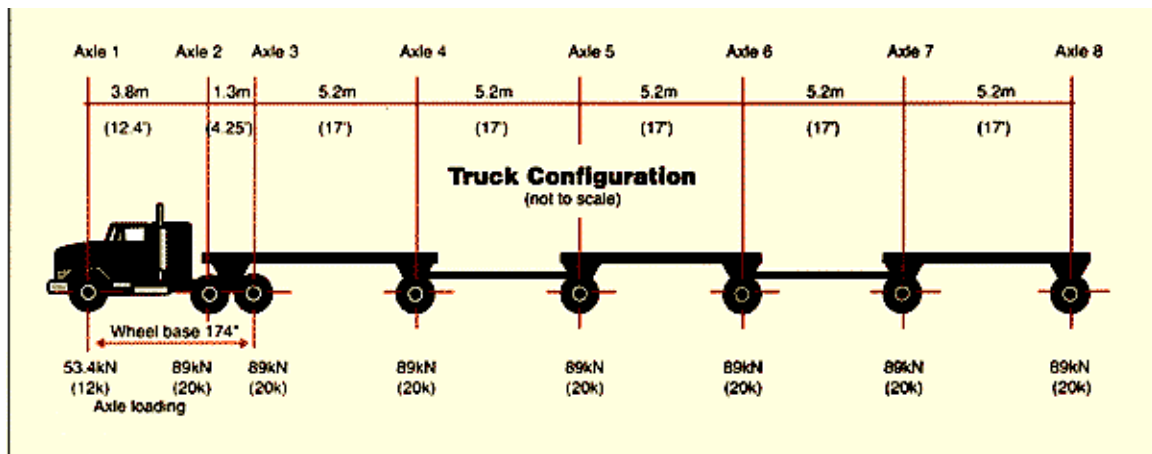


FIGURE II.2: WesTrack Truck Configuration

During the WesTrack experiment, detailed information was obtained regarding material properties. Site exploration and laboratory experiments document material data for the site soils and sub-grade conditions. Aggregate gradation information for the granular base layers and the asphalt concrete mixes was obtained over the entire construction and maintenance process. Investigation of various mix designs, and the impact on performance of asphalt binder content, aggregate properties, air void content, and layer thickness, was performed by developing 26 mix designs for the original experiment (FIGURE II.1). Material testing on the various AC mix designs was

performed prior and during construction to ensure consistency for accurate experiment results. The test sections were designated with respect to these material properties as shown in TABLE II.1. The aggregate gradation designations are based on Superpave mix design specifications of the same names. The designations of asphalt content refer to $\pm 0.7\%$ from optimal binder content based on Superpave volumetrics, which includes the aggregate classification. (35)

TABLE II.1: Original Test Section Designations at WesTrack Experiment

| Design Air Void Content | Aggregate Gradation Designation | | | | | | | | |
|----------------------------|---------------------------------|-------|------|-----------|-------|-------|--------|-------|------|
| | Fine | | | Fine Plus | | | Coarse | | |
| | Asphalt Content (%) | | | | | | | | |
| | Low | Opt. | High | Low | Opt. | High | Low | Opt. | High |
| Low: 4% | | 04 | 18 | | 12 | 09/21 | | 23 | 25 |
| Medium: 8% | 02 | 01/15 | 14 | 22 | 11/19 | 13 | 08 | 05/24 | 07 |
| High: 12% | 03/16 | 17 | | 10 | 20 | | 23 | 06 | |

II.3 Methods for Surrogate Modeling

Many engineering design problems require expensive computation incorporating a large quantity of input data and/or complex mathematical models. These applications have recently benefited from surrogate modeling techniques which allow engineers to evaluate computationally inexpensive models rather than the computationally expensive models from which the surrogates are derived. Construction, validation, and verification are all required to demonstrate that the surrogate model well represents the model function, but once these steps are completed, the surrogate model can be utilized in many design applications such as sensitivity analysis, design optimization, and reliability analyses.

A number of surrogate modeling approaches are available, each with advantages and disadvantages relating to their success to produce an accurate predictive model. Surrogate modeling methodologies vary in complexity in computations which can influence the accuracy of the model compared to the true function.

The simplest surrogate model is a regression model where an output variable (Y) is described with respect to input parameters (X) and regression coefficients (β). The regression model requires definition of a model form, such as linear or quadratic, where the regression coefficients are determined through an analysis of the residuals. Typical methods, such as least squared error, define the regression coefficients as those that minimize the error in the predictions of the model compared to true output values. Although regression models are simple to construct, inaccuracy arises when the model form is not close to the true function form. Additionally, functions with many parameters can become complex and the improvement in computational speed of the model compared to the true function reduces.

More advanced methods include polynomial chaos (PC) and radial basis functions (RBF) improve on simple regression models. PC models improve on regression models by replacing the input parameters (x) with Hermite polynomials. An example of a PC model can be expressed as:

$$Y(x) = \beta_0 + \sum_{i=1}^n \beta_i [\Gamma_p(\xi_i)] \quad (\text{II.13})$$

The input parameters (x) are expressed in the Hermite terms in the standard normal space and are represented as (ξ). The Hermite polynomial term (Γ) can include any order (p), for example:

$$\begin{aligned}\Gamma_1 &= \xi \\ \Gamma_2 &= \xi^2 - 1\end{aligned}\tag{II.14}$$

The methods of solving for the regression coefficients (β) are similar to those for the simpler regression models, but the inclusion of the Hermite polynomials allows for improved model performance. The PC method is highly effective for second and third order models with as many as ten input parameters, but the computation expense beyond this can be prohibitive.

Another advanced technique, an RBF model, is expressed mathematically as:

$$Y(x) = \sum_{i=1}^{n_c} w_i \psi \|x - c^{(i)}\| \tag{II.15}$$

The model output (Y) in an RBF is a weighted sum of the n_c basis functions evaluated at the Euclidean distances between the input parameter (x) and the centers of the basis functions ($c^{(i)}$). Basis functions (ψ) provide a simplification to the complex model by dividing the model into a family of simpler models. Common applications include multivariable polynomial models and periodic functions such as Fourier models. The

benefit of RBF models is that the estimation of the weights (w_i) are computationally cheap and yet the model is capable of emulating highly non-linear functions.

The Gaussian Process (GP) surrogate model is a special form of an RBF model and has been shown to be a very powerful surrogate modeling technique for many engineering applications. GP models are shown to be capable of fitting data for high dimensional problems, on the order of 30-50 input parameters, and are an interpolation method that does not follow a specific functional form. GP models are suitable for approximating any smooth, continuous function, common in many engineering applications.

Construction of a GP model requires decisions for a correlation function and a mean function. The squared-exponential is a commonly utilized correlation function. This form utilizes the following equation:

$$c(x^j, x^k) = \exp\left[-\sum_{i=1}^n \xi_i (x_i^j - x_i^k)^2\right] \quad (\text{II.16})$$

Where ξ_i is a scale factor that must be estimated, x_i^j represents the j^{th} training point at the i^{th} dimension, and x_i^k represents the new prediction point at the i^{th} dimension. The terms are summed over the number of training points, n . The correlation function is utilized to construct a correlation matrix, R :

$$R = \begin{bmatrix} c(x_1^1, x_1^k) & \cdots & c(x_1^j, x_n^k) \\ \vdots & \ddots & \vdots \\ c(x_n^1, x_1^k) & \cdots & c(x_n^j, x_n^k) \end{bmatrix} \quad (\text{II.17})$$

The covariance function, indicating the covariance between the observed model response values of the training data, $Y(x^j)$, and the predicted responses, $Y(x^k)$, is represented as a function of the correlation matrix, R , and variance as shown here:

$$\text{Cov}(Y(x^j), Y(x^k)) = \sigma^2 R \quad (\text{II.18})$$

The variance term in Eq. II.18 is another parameter of the GP model that must be estimated. A mean function is also required for construction of the surrogate model. A common constant function form is shown in Equation II.19.

$$\mu(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots \quad (\text{II.19})$$

The vector, β , is the final parameter that must be estimated to complete the construction process. Once the model form has been selected, the model parameters (mean μ , variance σ^2 , and correlation length-scale factors ξ) must be estimated. The process of parameter

estimation is commonly performed utilizing a maximum likelihood estimation method. The procedure takes the form of an optimization problem. To avoid common complications due to ill-conditioned matrices, the optimization problem is often modified to a minimization of the negative log-likelihood function, $-\log[L(\cdot)]$, of the form:

$$\begin{array}{l} \text{Minimize} \\ (\beta, \sigma^2, \xi) \end{array} \quad -\log[L(\cdot)] = n * \log(\sigma^2) + \log|R| + \frac{(Y - \mu)^T R^{-1} (Y - \mu)}{\sigma^2} \quad (\text{II.20})$$

Model verification is required prior to use in design applications. Model verification is often based on prediction testing. The values for the prediction points are calculated as the mean value of the distribution:

$$E[Y(x^k) | Y] = \mu + r^T R^{-1} (Y - \mu) \quad (\text{II.21})$$

Where r represents a vector of correlations as represented by:

$$r = \begin{bmatrix} c(x_1^j, x_1^k) \\ \vdots \\ c(x_n^j, x_n^k) \end{bmatrix} \quad (\text{II.22})$$

II.4 Analytical Reliability Methods for Pavement Design

The early concepts for reliability analysis in the context of pavement design have been summarized by Hudson (36) and Huang (37). Prior to 1965, the safety factor method was applied in the design of Portland cement concrete pavements. The safety factor method, however failed to properly account for different magnitudes of uncertainties associated with the design and load parameters, which can significantly affect the reliability of the pavement. Later, Lemer and Moavenzadeh (38) employed the Monte Carlo Simulation technique to compute this reliability. In the MCS technique, the uncertainties in the random variables are described by appropriate probability distributions. However, a large number of iterations requiring very large amounts of computer time were required, rendering the technique infeasible for all but the simplest problems to obtain a result with a small variance. The approach never gained widespread application until very recently, and now only for simplified approaches to pavement design.

Darter and Hudson (39) characterized the pavement design problem by two random variables: N_F , the number of allowable axle load applications to failure, and N_A , the number of actual load applications. The condition of the pavement can then be described by the limit state function shown in Equation II.23.

$$g = \log(N_F) - \log(N_A) \quad (\text{II.23})$$

The condition of the pavement is considered to have deteriorated below acceptable limits when N_A exceeds N_F , or equivalently, when $g \leq 0$. By assuming lognormal distributions for N_F and N_A the probability of failure is obtainable as:

$$P_F = \Phi(-\beta_c) \quad (\text{II.24})$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal random variable and:

$$\beta_c = \frac{E[g]}{\sigma[g]}$$

where $E[g] = g[M]$

$$\text{and } \sigma[g] = \nabla g^T [C] \nabla g \Big|_{\bar{X}=\bar{M}} \quad (\text{II.25})$$

where β_c represents the reliability index. The variables $E[g]$ and $\sigma[g]$ are the mean and standard deviation of g , respectively. These moments are calculated by finding the moments of the first-order Taylor expansion of the limit state equation, g . The Taylor series expansion is truncated at the linear terms, providing the first order approximation of the mean and variance of the limit state equation. The result is that the mean is calculated by evaluating the limit state function at the mean values of the random, dependent variables. Similarly, the variance involves the covariance and mean values of the variables, as represented in Equation II.25. This method of using the mean and covariance of the random, dependent variables to determine the reliability of the limit

state function is known as the Mean Value First Order Second Moment method. Given the computational simplicity of this method, subsequent work in probabilistic pavement design adopted a similar approach in the procedure used in the determination of the moments of N_F (40–43). All of these papers apply second-moment reliability methods, particularly the MVFOSM method (27, 44).

The Rosenblueth method (30) is another well known method and is similar to the FOSM method in that it calculates reliability from mean and variance of $g(x)$. However, these moments are calculated by evaluating $g(\cdot)$ at all 2^n combinations of the n random inputs, each taken at one standard deviation above and below the mean. The mean of the performance function is given by:

$$E[g] = \frac{1}{2^n} * \sum g \quad (\text{II.26})$$

where n is the number of variables.

The variance is calculated as:

$$V[g] = E[g^2] - (E[g])^2 \quad (\text{II.27})$$

The probability of failure is determined in the same manner as the FOSM method, by calculating the reliability index and evaluating the cumulative distribution function.

While the MVFOSM and Rosenblueth methods have enjoyed popularity in pavement engineering, they have several important limitations. Firstly, more information beyond the first and second moments is typically available to the design engineer. In practical problems the researcher will likely have data from which higher order moments and full probability distributions can be determined. This renders second moment methods biased, as a reliability analyst must consider all information available. Furthermore, the assumption of a normal distribution for the distribution of the limit state function evaluated in the space of the original random variables is not necessarily valid. But the most important limitation of the MVFOSM and Rosenblueth methods is that of the lack of invariance with respect to equivalent formulations of the limit state equation, first explained by Ditlevsen (45). The reliability estimates resulting from different but equivalent expressions of the limit state function can be different using these methods. Although limit states can be expressed in mechanically equivalent terms, such as stress or strength, the statistical results for this method will not be mathematically equivalent. Not only is invariance a problem, it makes it impossible to quantify accurate correlations among failure modes. For that reason, MVFOSM is not used in system reliability calculations.

The FORM methods are an improvement to the FOSM method, but require additional computation. The FOSM method has a number of deficiencies, one of which is the absence of the probabilistic distribution properties of the random variables. The FORM method utilizes the variable properties, transforms all the variables into equivalent

normal variates, and ultimately determines the reliability index by solving for the limit state defined by the performance function. There are multiple methods of solving FORM, one method, FORM I, (28) requires an iterative approach and a FORM II method (29) incorporates an algorithm to solve for the reliability index. A third method, FORM III, utilizes a generalized reduced gradient search algorithm and can be implemented with the solver function in Microsoft Excel.

In FORM, a limit state function $g(\mathbf{x})$ is used to characterize the state of the system as failed or safe, and failure and safety domains are characterized as:

$$\begin{aligned} \{F\} &= \{\mathbf{x} : g(\mathbf{x}) \leq 0\} \\ \{S\} &= \{\mathbf{x} : g(\mathbf{x}) > 0\} \end{aligned} \tag{II.28}$$

where $\{F\}$ and $\{S\}$ define the failure and safety sets, respectively, and limit state function $g(\mathbf{x})$ defines the limit-state or failure surface $L_x = \{\mathbf{x} : g(\mathbf{x}) = 0\}$ that divides the entire \mathbf{x} space into the above distinct sets. The limit state functions are derived from the individual distresses. The probability of “failure” of the pavement section is defined as:

$$P_F = P\{g(\mathbf{x}) \leq 0\} = \int_{g(\mathbf{x}) \leq 0} f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \tag{II.29}$$

where $f_{\mathbf{x}}(\mathbf{x})$ is the joint probability density of variables x_1, x_2, \dots, x_n . The reliability is then the probability that the design criteria is not exceeded, or $1 - P_F$. An analytical evaluation of the integral in Equation II.29 is possible in only a few special cases, and hence numerical integration is necessary. However, the limits of integration become intractable whenever the number of random variables exceeds two or three.

In FORM, there are four important steps in the calculation of the probability of failure for an individual component distress mode. These are:

1. Definition of a transformation from the original \mathbf{x} space to the standard uncorrelated normal \mathbf{u} space. In the case of uncorrelated variables the transform is given by

$$\mathbf{u} = \Phi^{-1}(F(\mathbf{x})) \quad (\text{II.30})$$

Convenient transformations are defined in Liu and Der Kiureghian (46) for the general case of correlated variables with prescribed marginal distributions.

2. Calculation of the most probable point of failure (MPP), the solution to the constrained optimization problem:

$$\mathbf{u}^* = \arg \min(\|\mathbf{u}\| \mid g(\mathbf{x}) = 0) \quad (\text{II.31})$$

3. Calculation of the reliability index β . β is in general equal to $\alpha \mathbf{u}^*$, in which α is the negative normalized gradient row vector of the limit state surface in the \mathbf{u} space, pointing toward the failure domain. For most practical problems β is greater than zero, in which case β is also equal to $\|\mathbf{u}^*\|$. The magnitude of the elements of the α vector gives information about the sensitivity (relative contribution to the variance of the limit state function). (47)
4. Calculation of the probability of failure. In FORM, the limit state surface is approximated by the hyperplane $\beta - \alpha \mathbf{u} = 0$ to simplify the integration boundary. The probability of failure is approximated as $P_{F1} = \Phi(-\beta)$.

The results of the individual component reliabilities in FORM can also be used to estimate system reliability. A pavement is best represented as a serial system of components defined by individual limit state functions, for the pavement is considered failed if any one of the individual component distresses is exceeded. For pavements in general, the serial system failure probability is:

$$P_{F,SYS} = P\left\{\bigcap_i g_i(\mathbf{x}) \leq 0\right\} \quad (\text{II.32})$$

Let \mathbf{B} be the vector of reliability indices for each of the limit states and the elements of the matrix \mathbf{R} be the dot products of the corresponding α vectors for each distress mode. Then for a series system, the system failure probability is given by $1 - \Phi(B, R)$, where

$\Phi(B, R)$ is the standard normal multivariate CDF with correlation matrix \mathbf{R} . For the bivariate case it can be shown that

$$\Phi(\beta_1, \beta_2, \rho_{1,2}) = \Phi(\beta_1)\Phi(\beta_2) + \int_0^{\rho_{1,2}} \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left[-\frac{\beta_1^2 + \beta_2^2 - 2\rho\beta_1\beta_2}{2(1-\rho^2)}\right] d\rho \quad (\text{II.33})$$

If more than two limit states are considered, then one may elect to use bounding formulae such as those in Ditlevsen (48) or evaluate the multi-normal CDF using a numerical scheme.

The FORM I method performs the previously outlined procedure using an algorithm introduced by Rackwitz (28). Specifically, the algorithm begins by defining the limit state equation, assuming an initial value for the reliability index, and assuming initial values for the random variables. The mean and standard deviations of the equivalent normal distribution for all the random variables are calculated and used to evaluate the partial derivatives of the performance function at each random variable. The evaluated partial derivatives and standard deviations of the normal equivalents are used to determine the direction cosines. Once these calculations have been performed, a new design point can be evaluated for each random variable by the following equation:

$$x_i^* = \mu_{X_i}^N - \alpha_i \beta \sigma_{X_i}^N \quad (\text{II.34})$$

Where x_i^* represents the new design point, $\mu_{X_i}^N$ and $\sigma_{X_i}^N$ represent the mean and standard deviation in the equivalent normal space, respectively, α_i is the direction cosine, and β is the reliability index. These steps are repeated in this method until the direction cosines converge to a pre-determined tolerance. Once the direction cosines converge, a new value of β can be calculated by forcing the performance function to zero by treating β as the unknown variable, and solving for β . This last step is repeated until the reliability index converges. Once the reliability index is determined, the final step is to determine the probability of failure by evaluating the cumulative distribution function at the reliability index.

The FORM II method is a modification to the FORM I method which can be cumbersome or impossible if the reliability index cannot be obtained by evaluating the performance function equal to zero. This method implements an algorithm that linearizes the performance function and performs iterations based on the partial derivatives of the performance function. The initial procedure is the same as that for the FORM I method. The partial derivatives are calculated and then the partial derivatives in the equivalent normal space are evaluated. These partial derivatives represent the components of the gradient vector of the performance function in the equivalent standard normal space (47) and are calculated by the equation:

$$\left(\frac{\partial g}{\partial x_i} \right)^* = \left(\frac{\partial g}{\partial x_i} \right) \sigma_{X_i}^N \quad (\text{II.35})$$

Where $\left(\frac{\partial g}{\partial x_i}\right)$ and $\left(\frac{\partial g}{\partial x'_i}\right)^*$ represent the partial derivatives in the original and equivalent normal spaces, respectively, and $\sigma_{x_i}^N$ is the standard deviation in the standard normal space. New design points are determined in the equivalent standard normal space, utilizing the Rackwitz-Fiessler formula:

$$x_{k+1}^* = \frac{1}{|\nabla g(x_k^*)|^2} * [\nabla g(x_k^*)^t * x_k^* - g(x_k^*)] * \nabla g(x_k^*) \quad (\text{II.36})$$

where $\nabla g(x_k^*)$ represents the gradient vector of the performance function. The reliability index can then be calculated as the root sum of the squares of the design variables. The new values of the design points should be used to repeat the process until the reliability index converges. The probability of failure is determined similar to the method described for the FORM I method.

A third FORM method, FORM III, determines the probability of failure by calculating the cumulative distribution function at the reliability index. The reliability index can be evaluated by minimizing β , subject to the performance function equal to zero, by modifying all random variables. The reliability index is calculated as the square root of the sum product of the equivalent normal values of the design variables, and the probability is calculated by evaluating the cumulative distribution function at β .

One final reliability method to be studied in the context of M-E pavement design is the Advanced Mean Value Method. This method is similar to the FORM method, but the AMV method makes one simplifying assumption. The AMV method assumes that when the limit state function approaches zero, that point represents the most probable point. Therefore, the limiting function can be forced to zero by changing the β value. This method has an advantage over the second moment method in accuracy because it, like FORM, uses computation in the rotationally symmetric standard uncorrelated normal space. However, while AMV in general is not as accurate as FORM due to the imprecise calculation of the MPP, it only needs to evaluate the gradients of the limit state function once. Because the \mathbf{u} -space gradients, evaluated at the origin in \mathbf{u} -space, are used to approximate the \mathbf{a} vector, system reliability analysis can be performed.

II.5 Reliability Based Design Optimization

II.5.1 Optimization Problem Definitions

Reliability based design optimization problems are commonly categorized into three problems sets: \mathbf{P}_1 , \mathbf{P}_2 , and \mathbf{P}_3 . Additionally, each of these categories is applicable to either component or system-level design problems.

The \mathbf{P}_1 optimization problem considers a problem where the objective is to minimize the cost of a design subject to the constraint that each component in the design maintains a safe reliability level. This can be expressed as follows:

$$P_1 = \min_d \{c_0(d) | p_k(d, x) \leq \hat{p}_k\} \quad (\text{II.37})$$

The objective function is a cost function (c_0) of stochastic variables (d) and the constraint requires evaluation of the probability of failure with respect to both the stochastic and deterministic variables (x) for each k constraint function.

The \mathbf{P}_2 problem seeks to minimize the failure probability of the component (k) with the largest probability of failure, termed the critical component. This formulation minimizes the maximum probability of failure, but does not guarantee that the probability of failure meet a required threshold. Again, the probability of failure is with respect to both stochastic and deterministic variables.

$$P_2 = \min_d \{\max_k p_k(d, x)\} \quad (\text{II.38})$$

The third formulation (\mathbf{P}_3) differs slightly from the \mathbf{P}_1 problem in that the cost function is written with respect to the probability of failure of the components.

$$P_3 = \min_d \{c_0(d) + \sum_{k=1}^K c_k(d) p_k(d, x) | p_k(d, x) \leq \hat{p}_k\} \quad (\text{II.39})$$

These three formulations are easily re-written for system optimization problems.

II.5.2 RBDO Problem Formulations

Various formulations of RBDO methods have been developed and applied to numerous engineering applications (49). Solution techniques utilizing First Order Reliability Methods have been shown effective for both component and system RBDO problems. RBDO using Efficient Global reliability Analysis (EGRA) provides another practical design process that has been shown to be accurate and efficient.

Two popular formulations for the FORM based optimization method are common: single loop direct FORM (also known as the reliability index approach) and inverse FORM (also known as performance measure approach). The single loop direct FORM based model can be described mathematically as:

$$\min_{\mathbf{d}, \mathbf{x}} f(\mathbf{d}) \quad (\text{II.40})$$

$$\text{subject to: } g(\mathbf{d}, \mathbf{x}) = 0 \quad (\text{II.41})$$

$$\frac{\mathbf{u}^*}{\|\mathbf{u}^*\|} = -\frac{\nabla_{\mathbf{u}}g(\mathbf{d}, \mathbf{x})}{\|\nabla_{\mathbf{u}}g(\mathbf{d}, \mathbf{x})\|} \quad (\text{II.42})$$

$$\|\mathbf{u}^*\| = \beta_{\text{required}} \quad (\text{II.43})$$

Where the objective function, Eq. II.40, is minimized with respect to design parameters (\mathbf{d}) and random variable input parameters (\mathbf{x}). Equations II.41 and II.42 are the constraints required to satisfy the Karush-Kuhn-Tucker (KKT) conditions, and Eq. II.43 is the

reliability constraint required by FORM. The reliability index ($\beta_{required}$) is defined as the norm of the vector of input parameters transformed to the equivalent standard normal space (u). The vector of variables in the transformed space is related to the gradient of the limit state equation (g) and its norm.

The single loop inverse FORM model can be described as:

$$\min_{\mathbf{d}, \mathbf{x}} f(\mathbf{d}) \quad (\text{II.44})$$

$$\text{subject to: } g(\mathbf{d}, \mathbf{x}) \geq 0 \quad (\text{II.45})$$

$$\frac{\mathbf{u}^*}{\|\mathbf{u}^*\|} = -\frac{\nabla_{\mathbf{u}}g(\mathbf{d}, \mathbf{x})}{\|\nabla_{\mathbf{u}}g(\mathbf{d}, \mathbf{x})\|} \quad (\text{II.46})$$

$$\|\mathbf{u}^*\| = \beta_{required} \quad (\text{II.47})$$

Here, the objective function is similar to the direct FORM method, Eq.s II.46 and II.47 satisfy the KKT conditions, and Eq. II.45 satisfies the inverse FORM optimality condition. The inverse FORM method has numerous advantages, as described by McDonald and Mahadevan in (49).

FORM methods are efficient with respect to required function evaluations, but approximations in these methods can cause them to fail to find the MPP. Methods such as the direct FORM method can result in inaccuracies if the approximation to the shape of the limit state is poor. The EGRA process is an alternative to these FORM based methods that improves accuracy and maintains efficiency, primarily through the use of surrogate

modeling. EGRA evaluates the function at a small number of samples, constructs a surrogate model for the function, and solves an auxiliary optimization problem finding the point of maximum expected feasibility. This point of feasibility is determined through an expected feasibility function which searches for potential training points near the limit state, the area where accuracy is most important. The process iterates by selecting this point as a new training point and re-training the surrogate repeatedly until the expected feasibility converges. The final surrogate model can then be used to make predictions of reliability for the true function.

The EGRA RBDO method can be formulated as a nested, single-loop, or sequential optimization problem. The nested loop formulation is the most computationally expensive process as each iteration requires full reliability analyses and no information from these analyses are shared in later iterations. A single-loop method improves the efficiency of the nested method through use of a surrogate model that evaluates the reliability across a domain rather than at individual candidate points. The potential for model error is introduced with the inclusion of the surrogate, but results are easily verified after convergence of the EGRA analysis. Sequential formulation improves on the single-loop process by intermittently improving the accuracy of the surrogate model to incorporate verification into the iterative process. (50)

II.6 Discussion

Current flexible pavement design in the United States requires improvements in accuracy in pavement predictive models, reduction in computational inefficiency, and practical

design optimization methods. The AASHTO MEPDG utilizes an extensive amount of experimental data, but many researchers have found inaccuracies in the predictive models. Model form error contributes to the inaccuracy, compounded by neglect of input parameter variability in the current, deterministic design procedure. Specifically, the permanent deformation models have been shown to not perform well compared to field measurements. Chapter III of this dissertation investigates multiple permanent deformation models and discusses the predictive capability of these models compared to experimental data. The current M-E predictive models are computationally expensive to evaluate when performing multiple design iterations. Surrogate modeling has been demonstrated in Chapters IV,V, and VIII as a solution to reduce the computational expense of these models without loss of accuracy. A framework for uncertainty propagation is presented in Chapter V, and the two chapters immediately following propose methods for analytical and probabilistic reliability analysis within this framework. Improvement to the current design procedure is necessary to provide designers computationally efficient tools for incorporating initial construction and recurring maintenance costs into the design and construction decision process. The risk-based design optimization framework presented in Chapter VIII is computationally efficient, through the use of a well-trained surrogate model, and incorporates all sources of uncertainty in pavement design for a more reliable predicted performance.

CHAPTER III

CALIBRATION, SELECTION, AND UNCERTAINTY QUANTIFICATION FOR PERMANENT DEFORMATION PREDICTIVE MODELS FOR FLEXIBLE PAVEMENT

III.1 Introduction

M-E design methods, although an improvement to purely empirical methods, do not eliminate the uncertainty in predicted pavement behavior models. One uncertainty that currently exists is model form error, which occurs due to a lack of fit between the predicted and actual behavior of pavement. The source of this error is primarily due to inadequacy of the model in incorporating all the mechanistic properties of the pavement behavior. One specific pavement distress model which is highly susceptible to model form error is the permanent deformation model for flexible pavements. This Chapter investigates the model uncertainty for the permanent deformation distress mode with respect to three differing predictive theories. Model calibration is performed and the models are validated with experimental data to determine the uncertainty that exists in the predicted and actual performance. Management of uncertainty due to the predictive models is necessary for properly understanding propagation of uncertainty and for performing sensitivity analysis, necessary for better pavement design decisions and cost effective designs. Quantification of this model uncertainty will increase reliability of predicted performance which is critical to design processes such as design optimization.

To date, research has resulted in the development of two primary prediction models for the permanent deformation distress mode based on two different mechanistic behavioral theories: behavior due to axial strain and shear theory behavior. A third theory introduced in this dissertation assumes that permanent deformation is best described by a model that combines both shear and axial theories. Accurate predictions for permanent deformation require reduction in model form error. Six permanent deformation predictive models are investigated in this dissertation to determine the most accurate predictive model for use in flexible pavement design. Validation of the models presented within this dissertation is achieved utilizing empirical data from the NCHRP WesTrack Project.

Permanent deformation is chosen as the distress model for illustrative purposes for several reasons. First, permanent deformation is a failure mode that impacts driver safety and significantly contributes to maintenance cost over the design life. Second, permanent deformation in the bound layers can be repaired by resurfacing. (51) Resurfacing alone accounted for \$3.8 billion dollars in federal funds in 2009, approximately 12% of the total obligation of federal funds on the National Highway System for all improvement types. (52) Deep structural rutting requires reconstruction of the entire pavement. Most important, many researchers have shown that the current models have been shown inaccurate in predicting true asphalt concrete behavior. (10) (53) (54) For example, the current models are not capable of accurately predicting pavement performance for asphalt concrete with polymer-modified binders. The behavior of such binders may increase pavement life spans, which can be contrary to the predicted behavior with current permanent deformation models. (55) Lastly, deformation is an easy

quantity to measure, therefore it is hypothesized that significant uncertainty comes from the predictive models rather than errors in field measurements.

III.2 Background

The two primary permanent deformation predictive models investigated in this dissertation are the MEPDG predictive model and the NCHRP Report 455 WesTrack Level 1-B predictive model. Each model assumes that permanent deformation in a pavement system is the accumulation (sum) of the permanent deformation through all layers of the pavement system. Different models are necessary for describing the permanent deformation in bound and unbound layers. Deformation in the unbound layers is assumed to be a function of vertical compressive strain by both the MEPDG and WesTrack models, though different forms of the mechanistic equations were utilized by the two predictive models. Further, the models for the unbound layers were calibrated with different empirical data. The MEPDG investigated numerous mechanistic models and selected a model derived by El-Basyouny and Witczak. (6) The MEPDG utilized LTPP data to calibrate this model. The WesTrack model utilized the Asphalt Institute equation and empirical data from the WesTrack project.(4)

The significant distinction between these two predictive models comes in the predictive model for the bound layers which differs due to the underlying assumptions of the mechanistic behavior that causes permanent deformation in the asphalt concrete layer. The MEPDG model assumes that permanent deformation is the result of axial strain and the WesTrack model assumes that shear deformation is the cause of deformation.

The MEPDG predictive model for predicting permanent deformation for asphalt pavements is derived from empirical data obtained from LTPP data and linear elastic analysis of the asphalt layer(s). The model form in the asphalt layer is based on a constitutive relationship initially derived from laboratory repeated load permanent deformation tests:

$$\frac{\varepsilon_p}{\varepsilon_r} = aT^b N^c \quad (\text{III.1})$$

Where the plastic strain (ε_p) is expressed as a function of N load repetitions, a pavement temperature T , the resilient strain ε_r , and regression coefficients a , b , and c . This model form comes with the assumption that the permanent deformation is a function of vertical plastic deformations and not a function of plastic shear deformations. The MEPDG discusses three major stages of pavement rutting and concludes that the primary and secondary stages describe most practical applications. Previous research indicates that these two stages are predominantly impacted by vertical strains and it is only the tertiary stage at which shear deformation must be considered for predicted performance. (6) The mechanistic model for asphalt deformation is modified with the inclusion of calibration regression coefficients (β) (Equation III.2) which are calibrated with LTPP data.

$$\frac{\varepsilon_p}{\varepsilon_r} = \beta_{r1} a T^{\beta_{r2} b} N^{\beta_{r3} c} \quad (\text{III.2})$$

The MEPDG comments that the form of this predictive model is quite simple as the permanent strain is determined by evaluating the resilient strain. The resilient strain is defined by a simple equation, assuming elastic behavior, including only the material's elastic modulus, Poisson's ratio and the state of stress due to the applied traffic loading.

The calibration process for the model of permanent deformation in the asphalt layer is performed by minimizing the error between actual and predicted performance, utilizing Equation III.1 where a layered elastic analysis program is used to determine the resilient strain. The regression coefficients are derived from non-linear regression based on the NCHRP 9-19 Superpave Experiment and the calibration factors are derived from LTPP sections located in 28 different states. (6)

The NCHRP Report 455 develops a number of permanent deformation models by investigating direct regression analyses as well as regression based on mechanistic-empirical analyses utilizing the data from the WesTrack project. Permanent deformation in the asphalt concrete layer for these models is based on the assumption that shear deformation governs deformation. One WesTrack formulation, based on M-E analysis, is a least squares regression between predicted total permanent deformation and the WesTrack rutting data. The regression equation is developed by estimating the rut depth of all layers through the procedure shown in FIGURE III.1. The process requires evaluating the impact of RSST-CH data on temperature and moduli of elasticity. Stresses and strains in the pavement structure are calculated by elastic analysis at key locations including: 2 inches below the surface at wheel edges and at the top of the sub-grade layer.

The accumulation of these strains is used to estimate rut depths and the regression process iterates until the M-E model regression coefficients converge. Regression utilizing equivalent single axle loads (ESALs) and mix parameters is then performed between the calibrated M-E model and the empirical data from the WesTrack experiment.

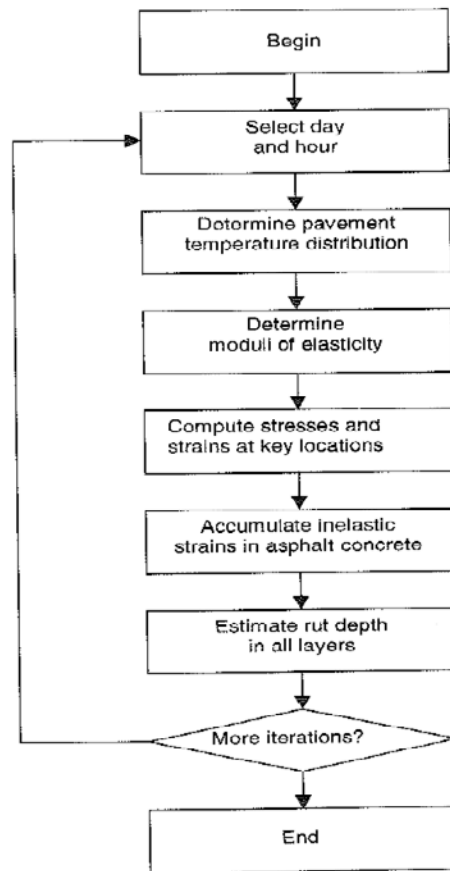


FIGURE III.1 NCHRP 455 Regression Analysis Procedure

The WesTrack Level 1-B equation for permanent deformation (rd_{HMA}), derived by the regression procedure previously described, is defined as the product of a regression coefficient (κ) and the permanent (inelastic) shear strain (γ^i).

$$rd_{HMA} = \kappa \gamma^i \quad (III.3)$$

Where:

$$\gamma^i = a * \exp(b\tau\gamma^e n^c) \quad (III.4)$$

Permanent (inelastic) shear strain (γ^i) is defined as a function of elastic shear stress (γ^e) and shear strain (τ), the number of axle load repetitions (n), and regression coefficients a , b , and c . The regression coefficient is determined empirically outside the scope of the NCHRP project and is defined as a function of HMA thickness. Similar to the MEPDG model, layered elastic behavior is assumed and is necessary in calculation of the elastic shear stress and corresponding shear strain values in the WesTrack Level 1-B model. The elastic analysis utilizes the moduli of elasticity as determined empirically through the RSST-CH laboratory results.

Once the NCHRP M-E model is calibrated, a final regression model is derived relating the M-E model to mix parameters. One recommended regression model presented by the NCHRP report is shown in Equation III.5 and includes mix parameters: percent of asphalt content (P_{asp}), percent of air void content (V_{air}), percent of aggregate finer than a No. 200 sieve (P_{200}), and ESALs. The terms *fine plus* and *coarse* take the

value of unity when the mix is equivalent to the corresponding WesTrack mix design and zero otherwise. This equation is the formulation chosen for analysis and comparison in this dissertation.

$$\begin{aligned} \ln(rd) = & -6.1651 + 0.309941 \ln(ESAL) + 0.00294305V_{air}^2 + 0.0688276P_{asp}^2 - \\ & 0.0657803P_{asp}P_{200} + 0.600498(fine\ plus) - 1.59167(coarse) + \\ & 0.21327 \ln(ESAL) (coarse) \end{aligned} \quad (III.5)$$

III.3 Construction of Predictive Models

Six permanent deformation prediction models are considered in this chapter. The nationally calibrated MEPDG model, herein referred to as the “national” model, is used to represent a model based purely on plastic axial strain. Two additional models are derived from the MEPDG design method. The first method, also purely based on the plastic axial strain, is a locally calibrated model (locally calibrated), and a second, combined model modifies the calibration factors through a regression analysis (parameter calibrated). A shear theory model utilized in this study is derived from NCHRP Report 455 (NCHRP). Two final, combined models are constructed as weighted averages between the NCHRP and the calibrated axial strain models.

III.3.1 MEPDG Rutting Models

The national, locally calibrated, and parameter calibrated models are constructed within the MEPDG design procedure. The MEPDG utilizes calibration factors in the rutting performance prediction equation that can be altered from a national average to a local or project-specific value. The equation for plastic strain is calculated as shown in Equation III.6.

$$\frac{\varepsilon_p}{\varepsilon_r} = k_z \beta_{r1} 10^{k_1 T^{k_2} N^{k_3} \beta_{r3}} \quad (\text{III.6})$$

where ε_p and ε_r are plastic and resilient strain respectively, and the k and β values are the regression coefficients and calibration factors.

For this study, the regression coefficients (k_1, k_2, k_3) are kept at the national values of (-3.35412, 1.5606, 0.4791). The calibration factors ($\beta_1, \beta_2, \beta_3$) are modified for the MEPDG based predictive models as described in TABLE III.1. The national model utilizes the nationally derived factors presented in the MEPDG.

The locally calibrated factors are derived utilizing the performance data obtained at the WesTrack experiment, including climatic data specific to the site, traffic input that best represents the actual traffic loadings, and mix properties for the numerous experimental pavement test sections. The local calibration process is performed by minimizing the sum of the squared residuals between the measured WesTrack permanent deformation values and the predicted values from the MEPDG for 17 WesTrack test

sections: 1, 4, 7, 9, 11 through 15, and 18 through 25. These specific sections are chosen to construct a locally calibrated model that can be compared to the parameter calibrated model, which is dependent on experimental results taken from these select test sections.

The calibration process is performed utilizing the Hooke-Jeeves optimization method and is implemented manually through both the MEPDG and the numerical computation software program, MATLAB. This method is a pattern search method that systematically searches in orthogonal directions for a local minimum. This method of optimization is specifically advantageous for this application for a number of reasons. First, this optimization method does not require knowledge of the form of the optimization problem's objective function. Additionally, because of compatibility issues between the MEPDG and software capable of optimization, the optimization routine cannot be automated. The Hooke-Jeeves method is easily adapted to the manual procedure required.

The model calibration problem is stated:

$$\min_{\beta_1, \beta_2, \beta_3} \sum_{n=1}^N \sum_{i=1}^{12} Residuals_i^2 \quad \forall N \text{ Sections} \quad (\text{III.7})$$

In Equation III.7, optimal calibration factors are found to minimize the sum of the squared residuals for the first 12 months of the WesTrack experiment, for all N test sections. To reduce computational effort, two convergence limits are imposed on the optimization routine. A solution to the optimization routine is considered converged if the

difference between the sums of the squared residuals between iterations is below 0.01. The smallest step size for the design variables is 0.125. The parameters of the locally calibrated model are given in TABLE III.1.

The parameter calibration model is considered to include both axial and shear theories through modification of the MEPDG calibration factors as a function of material properties that describe shear behavior. By modifying the calibration factors with regard to shear based parameters, the predictive model will contain both shear and axial parameters, incorporating both physics philosophies. Specifically, the shear based parameters utilized for this study are two measured values obtained through the Repeated Simple Shear Test at Constant Height (RSST-CH) test: repetitions of the test to 5% strain (*Reps5%*) and the resilient (complex) shear modulus (G^*). The parameters are related to the calibration factors through a linear relationship described as:

$$\beta_i = \beta_i(\text{LocallyCalibratedModel}) + m_{i,1}(\text{Reps5\%}(n)) + m_{i,2}(G^*(n))$$

$$i \in 1,2,3$$

$$\forall n \in N \quad \text{(III.8)}$$

where each of the three calibration factors ($\beta_1, \beta_2, \beta_3$) is a function of the *Reps5%* and G^* parameters for each of the pavement sections and of the calibration factors derived from the locally calibrated model ($\beta_1(LCM), \beta_2(LCM), \beta_3(LCM)$). The coefficients, or slope terms, in the linear relationship (m_1, m_2) are derived through a least squared optimization

routine, similar to the locally calibrated model, and are described in TABLE III.1. The optimization is considered converged if the difference between the sums of the squared residuals between iterations is below 0.01. The smallest step size for the design variable is 0.025.

TABLE III.1: Calibration Factors for MEPDG Predictive Models

| Predictive Model | β_1 | β_2 | β_3 |
|----------------------|---|--|---------------------------------------|
| National | 1.0 | 1.0 | 1.0 |
| Locally Calibrated | 2.875 | 1.0 | 1.0 |
| Parameter Calibrated | $2.875 + 0.15\text{Reps5\%} + 0.175(G^*)$ | $1.0 - 0.075\text{Reps5\%} - 0.1(G^*)$ | $1.0 + 0.1\text{Reps5\%} + 0.05(G^*)$ |

III.3.2 Shear Theory Model

A model that relates permanent deformation to shear behavior of the asphalt concrete layer(s) is considered, utilizing the “Level 1-B” analysis presented in the NCHRP Report 455 (35). The derivation of this model is based on both a regression analysis from WesTrack pavement performance and M-E analysis as described in the NCHRP Report and in Chapter III.2. This model can be utilized for prediction for pavement mixes similar to those used at WesTrack, but requires re-calibration for alternative mix designs. Calibration may be required for alternative climatic regions and traffic patterns, but the concepts presented in the NCHRP indicate that the model should be applicable to all other traffic and environmental conditions. Additional experimental validation of this model with data other than the WesTrack data is beyond the scope of this dissertation.

The prediction model chosen is of the form:

$$\begin{aligned} \ln(rd) = & -6.1651 + 0.309941 \ln(ESAL) + 0.00294305V_{air}^2 + 0.0688276P_{asp}^2 - \\ & 0.0657803P_{asp}P_{200} + 0.600498(fine\ plus) - 1.59167(coarse) + \\ & 0.21327 \ln(ESAL) (coarse) \end{aligned} \quad (III.9)$$

where the permanent deformation (rd) is a function of equivalent single axle loads ($ESAL$), the percent of aggregates finer than the No. 200 sieve (P_{200}), air void content by percent (V_{air}) and the percent asphalt (P_{asp}). The terms *fine plus* and *coarse* take the value of unity when the mix is equivalent to the corresponding WesTrack mix design and zero otherwise.

III.3.3 Weighted Models

Two weighted models are constructed combining the NCHRP model and the calibrated models. The weighted average models predict permanent deformation as described by Eq. III.10.

$$PermanentDeformation = (w)(NCHRP) + (1 - w)(CalibratedModel) \quad (III.10)$$

where the permanent deformation prediction is calculated by weighting the prediction from each of the models considered. The weights for each model are established through a least squares optimization routine utilizing WesTrack data and are described in TABLE III.2.

TABLE III.2: Weight Coefficients for Weighted Models

| Model | Weights (w) | |
|--------------------------------|-------------|------------------|
| | NCHRP | Calibrated Model |
| NCHRP and Locally Calibrated | 0.61 | 0.39 |
| NCHRP and Parameter Calibrated | 0.48 | 0.52 |

It is important to note that the results from the least squares optimization provide information regarding the sensitivity of the predictive model to each of the model types. The combination of the NCHRP and locally calibrated models indicates that both models contribute information to the rutting prediction. The second model, combining the NCHRP and parameter calibrated model demonstrates a nearly equal contribution from either model to the sensitivity of prediction of rutting. Further, because the NCHRP model does not dominate the weighted average, neither the axial strain or shear theories are capturing a dominate amount of predictive power.

The weighted average coefficients show deformation to be more sensitive to the parameter calibrated model than the calibrated model. This is expected because the locally calibrated model is a subset of the parameter based model. Recall, the calibration factors utilized in the parameter calibrated model are functions of both material shear

behavior and the calibration factors derived for the locally calibrated model. Interestingly, the weighted coefficients are near equal for the NCHRP and parameter calibrated model, indicating that the inclusion of shear related materials parameters in the axial strain model improves the significance of this model with respect to predicted performance. Permanent deformation for this model is nearly equally sensitive to either the shear or axial theory models.

III.4 Model Validation and Comparison

Model validation is performed for each prediction model and then models are compared to determine the most effective predictive model. Model validation herein includes both classical and Bayesian techniques. Model validation metrics, including mean squared error and the adjusted coefficient of determination, are calculated for each model and the results are compared. Additional metrics including the Bayes Factor and the F-test are evaluated for direct model comparisons. A brief discussion of the computation required for these techniques follows.

The data utilized for the model validation and comparison calculations is from the WesTrack experiment described in NCHRP Report 455. The limitations of this model validation are restricted to the shear theory models which are derived utilizing this same data. The NCHRP Report 455 provides detailed commentary related to the applicability of this model to data outside the WesTrack experiment and it is assumed that the bias towards this data is minimal. Further model validation studies are necessary and should

include data outside the WesTrack experiment to quantify and address any bias in the validation results presented in this dissertation.

The mean squared error is a classical method of model validation, which incorporates the actual field measurements obtained through the WesTrack project and the predicted performance by each model. This metric requires the calculation of the residual between the actual and predicted pavement performance. The mean squared error is evaluated as the expected value, or mean, of the residuals squared. An additional classical metric, the coefficient of determination (R-squared value) relates a correlation between the predicted and actual field performance. A value close to one indicates that the two values are closely correlated and this is the objective of the calibrated models.

A Bayes factor is another model validation metric and is used to compare the two models' ability to describe experimental data. The Bayes factor is calculated as a ratio of the likelihood of observing the validation data conditioned upon the null and alternative hypotheses, as shown in Eq. III.11. (21) (56)

$$B = \frac{L(data|H_0)}{L(data|H_1)} \quad (III.11)$$

The null hypothesis states that the behavior of the experimental data is well represented by the predictive model in the numerator. The alternative does not support the model in the numerator as a good predictor of the experimental data in comparison with the model

represented in the denominator. Because the hypotheses are continuous functions, the likelihood of each hypothesis is proportional to the product of the probability densities of all the validation output. Calculation of the Bayes factor for this application can be reduced to:

$$B(x_0) = \frac{\prod_{i=1}^N \phi((y_i - \hat{y}_i), \text{model prediction}, \text{model error})}{\prod_{i=1}^N \phi((y_i - \hat{y}_i), \text{model prediction}, \text{model error})} \quad (\text{III.12})$$

where the numerator is the product of the evaluation of the standard normal probability density function (PDF) evaluated at the residual error between the predictive model and the WesTrack data. The denominator in Eq. III.12 is the product of the PDF evaluated in the same manner for a competing model. The mean and standard deviations for the normal PDFs are assumed to be functions of the models under investigation. The mean and standard deviations for the PDFs are calculated as the mean and standard deviations of the model error for each model.

The regression models' Bayes factor can be compared to each other, as well as to the national or NCHRP model Bayes factors, to determine the most appropriate predictive model. Jeffreys (57) provides a standard scale that is commonly used for interpretation of Bayes factors. A Bayes factor of 3 gives a substantial measure of support for the model in the numerator with respect to the model in the denominator, 10 a strong measure of support, 30 a very strong measure of support, and 100 a decisive measure of support.

The F-test is utilized to evaluate the benefit of incorporating additional model parameters. The F-test is a likelihood ratio test, expressed as:

$$F = \frac{\left(\frac{RSS_1 - RSS_2}{p_2 - p_1}\right)}{\left(\frac{RSS_2}{n - p_2}\right)} \quad (\text{III.13})$$

where the numerator is the ratio of the difference of residual sum of squares (RSS) for the two models divided by the difference in the number of parameters (p). The denominator is the ratio of the residual sum of squares for the second model divided by the difference between the number of validation points (n) and the number of parameters for the second model. This F value will be compared to a critical F , F_{crit} , to test the null hypothesis against an alternative. The critical F value is calculated as shown in Eq. III.14.

$$F_{crit} = F_{\alpha}(p_2 - p_1, n - p_2) \quad (\text{III.14})$$

For the formulation shown in Eq.s III.13 and III.14, the null hypothesis states that the second model does not provide a better fit to the data than the first model. An additional requirement for the test is that the model with fewer parameters be nested within the second model for an appropriate interpretation of results. The null hypothesis will be rejected if $F \geq F_{crit}$. (58)

III.5 Model Validation and Comparison Results

Model validation and comparison metrics for the six predictive models are presented in TABLE III.3. The F-test is performed only for the parameter calibrated and weighted models, with comparison to the locally calibrated model, as these are the only models of which the locally calibrated model is nested. The null hypothesis states that the parameter calibrated does not provide a better fit to the data than the locally calibrated model and a similar hypothesis is taken for the weighted models. The null hypothesis is rejected when the F value calculated for the two models under comparison is greater than the critical value. Further, if the F value is greater than the critical F value, the model with additional parameters provides significantly more information and the inclusion of these additional parameters is supported. Similarly, the Bayes factor is only calculated in comparison to the national model. The Bayes factor is not dependent on the number of model parameters, unlike the F-test, and therefore model comparisons can be made between all the models when calculated with respect to only the national model.

TABLE III.3: Model Validation and Comparison Results

| Model | Mean Squared Error | Adjusted Coefficient of Determination (R ²) | Bayes Factor (Compared to National Model) | F-Test (Compared to Locally Calibrated Model) | Critical F Value ($\alpha=1\%$) |
|--|--------------------|---|---|---|-----------------------------------|
| National | 0.132 | 0.350 | - | - | - |
| Locally Calibrated | 0.085 | 0.369 | 7.75E+05 | - | - |
| Parameter Calibrated | 0.072 | 0.471 | 3.20E+13 | 6.02 | 2.90 |
| NCHRP (Shear Theory) | 0.075 | 0.578 | 5.48E+21 | - | - |
| Weighted: Locally Calibrated and NCHRP | 0.068 | 0.579 | 1.57E+20 | 12.61 | 3.42 |
| Weighted: Parameter Calibrated and NCHRP | 0.058 | 0.613 | 4.31E+24 | 8.76 | 2.41 |

Note: Dash indicates data is not applicable.

The locally calibrated model has a lower mean squared error and a higher coefficient of determination than the national (MEPDG) model. The Bayes factor for the locally calibrated model indicates a decisive measure of support for the model over the national model. This clearly demonstrates the need for local calibration with regards to climatic data and traffic loading patterns; a fact that has been demonstrated repeatedly by other researchers (59) (60).

The weighted models show significant predictive power with the smallest MSE values and largest R² values. These two models reduce the average residual across all validation points as well as the standard deviation for the residuals, as shown in FIGURE III.2. The parameter weighted model is shown to reduce the error, with a mean nearer to zero than the national model. The PDF for the parameter weighted model also demonstrates the reduction in variance with a slope steeper than the other models. The

incorporation of both mechanistic theories with local calibration is clearly critical in optimizing accuracy of rutting predictions.

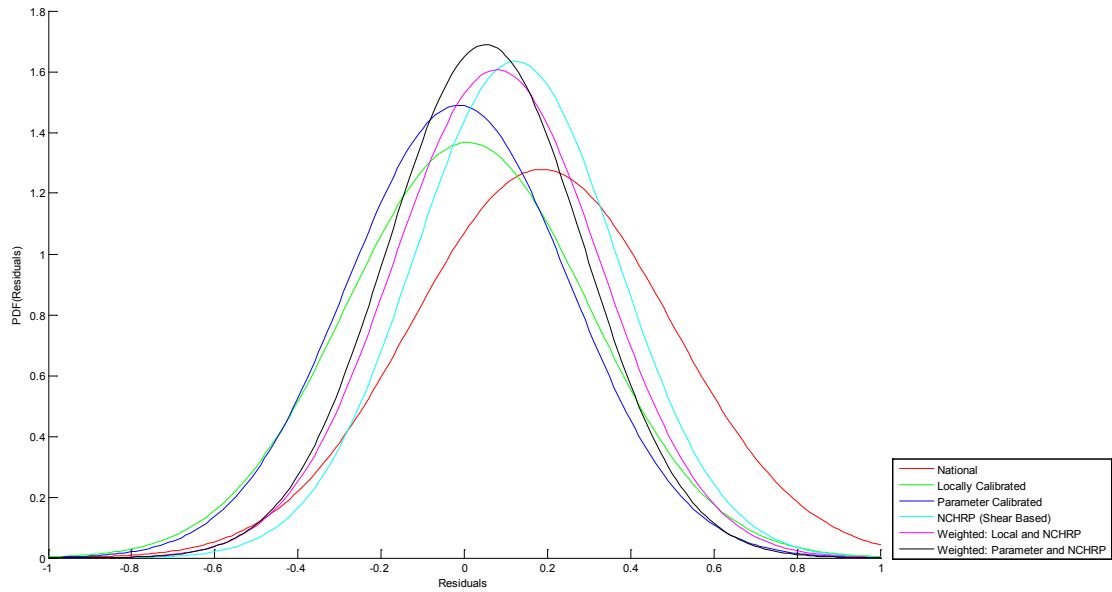


FIGURE III.2: Probability Density Functions for Model Residuals

The validation metrics clearly support the models which combine both shear and axial theories. The parameter calibrated model, again, is a model which incorporates the locally calibrated model and shear material parameters to create a predictive model which includes both theories for the mechanistic behavior governing permanent deformation. All validation results decisively support this model over the national and locally calibrated models. This indicates that calibration incorporating additional mix properties, specifically properties that describe the behavior of the mix with respect to shear, further improves predictive power. The Bayes factor for the parameter calibrated model indicates decisive support of this model over the national and local models. Additionally, the

results of the F-test indicate that the inclusion of additional model parameters significantly improves accuracy in predicted performance compared to the locally calibrated model.

FIGURE III.3 provides a visual comparison of the mean squared error and adjusted coefficient of determination for all the predictive models. It is clear that the performance of the axial strain models improve by calibration processes and the weighted models further this improvement. The weighted average models both minimize the error and are supported by larger adjusted R^2 values. The weights associated with each weighted model (shown in TABLE III.2) in conjunction with the metrics presented in FIGURE III.3 lead to the conclusion that the axial strain model and shear model are both contributing significantly to the accuracy in predictions. Although the adjusted coefficients of determination are quite low, with a best performing model only achieving a 0.613 value, the improvement from the national model value of 0.350 is significant. Investigation of an alternative weighted model, such as a quadratic or a form including an interaction term may see additional improvement in predictive power.

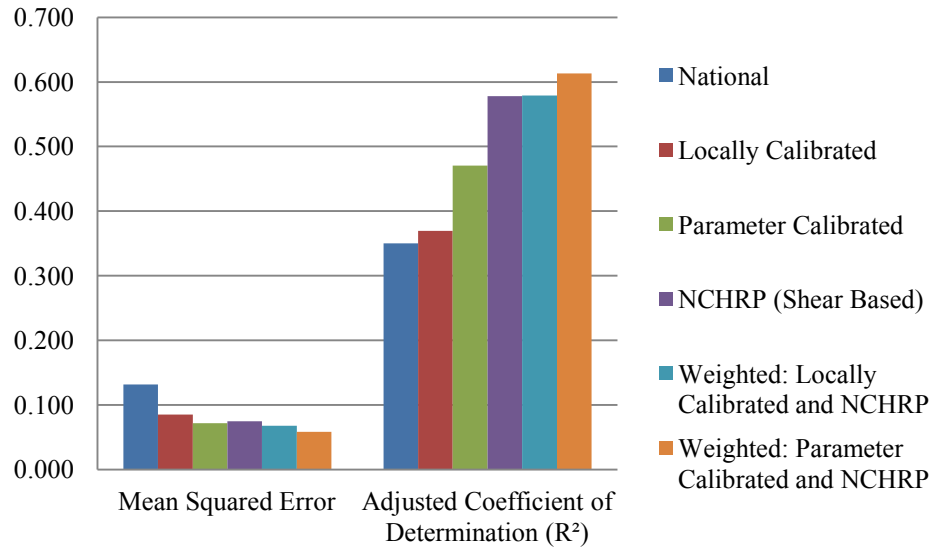


FIGURE III.3: Statistics for Model Validation

III.6 Conclusion

The MEPDG is a powerful tool for pavement design, but model error, which impacts predicted reliability levels for pavement performance, is neglected. The development of the MEPDG and the NCHRP design methods have been shown to significantly improve the accuracy of prediction of pavement performance over purely empirical methods, but, the development of M-E design methods has not eliminated model form error. The evolution of two contrasting mechanistic theories defining the behavior of flexible pavement with respect to the permanent deformation distress mode clearly indicates the need for an improved predictive model. Although models based on each theory have significant predictive power, it is clear that neither model fully captures the mechanistic behavior of flexible pavements with respect to the permanent deformation failure mode. The hypothesis presented herein states that a combined model will better capture the

mechanistic behavior and reduce model form error. The results presented here indicate that a model which combines both theories does reduce model form error and improves accuracy in predictions of permanent deformation. The weights for the weighted models indicate that both models contribute to improved model performance.

Reliability analysis for pavement structures requires accurate predictions in pavement performance and therefore requires accurate predictive models. The models presented indicate that calibration of the MEPDG model to incorporate local factors, site specific factors, *and* mix parameters is a critical step in accurate pavement predictions. Although the locally calibrated model improved accuracy in model predictions in comparison with the national or NCHRP models, additional improvement is found when the model incorporates both axial strain and shear theory. The mix parameters describing shear strength included in the parameter calibrated model are not currently included as design input parameters in axial strain M-E design models. Presented here is a procedure to incorporate those parameters through the calibration factors that already exist in the MEPDG software, resulting in a model that begins to capture the behavior of the pavement with respect to both mechanistic theories. These models improve predictive capability for pavement design, which is critical to providing reliable and cost effective designs.

Reduction of model error in pavement prediction models is necessary, but not sufficient, to assume reliable and cost effective pavement designs. The AASHTO MEPDG is a comprehensive design procedure, based on the theory that deformation is a function of axial strain, that can be enhanced through local and parameter calibration, but the computationally expensive model must be replaced if it is to become an efficient tool

for design engineers. A well-trained surrogate model can accurately imitate the MEPDG design equations and improves the computational time required for single design evaluations. The surrogate model combined with the already efficient regression model incorporating shear parameters will provide an accurate predictive model that greatly improves computational speed.

CHAPTER IV

SURROGATE MODEL INITIALIZATION: VARIABLE SELECTION PROCESS

IV.1 Introduction

The AASHTO MEPDG is the most current and comprehensive implementation of M-E design in the U.S, but it is computationally expensive to evaluate. The MEPDG design process enables engineers to choose a method based on the level of knowledge about the input parameters which impacts the computational effort required to design a pavement section. Evaluation of the MEPDG at the Level 1 design input level improves accuracy of design predictions by incorporating detailed information for input variables believed to be most closely linked to pavement performance, but practitioners are faced with a complicated data acquisition problem. Complex prediction models and extensive climatic data in conjunction with Level 1 input parameters result in a design process that is computationally expensive to run for highly iterative analyses such as design optimization or sensitivity analyses. Practitioners need a computationally efficient and accurate method for performing flexible pavement design. While Level 1 analyses are assumed to be more accurate, sensitivity analysis is necessary to determine the true impact of these inputs on predicted performance.

A surrogate model that accurately emulates the MEPDG Level 1 analysis will significantly improve computationally efficiency for analyses that require numerous iterations, but the efficiency of the surrogate model is dependent on the construction

process. Surrogate model construction requires selection of the quantity of training points (N_{TP}), the quantity of parameters for each training point (N_D), and selection of the location of the training points. For clarity, the quantity of training points refers to the selection of a point in the design domain space that will be utilized as an input in the construction of the surrogate model. The quantity of parameters for each training point refers to the dimensions of the model's inputs. For example, construction of a surrogate model can be performed by selecting N_{TP} training points, each of which describes a point in the N_D dimensions of the design domain. The surrogate model is constructed utilizing the $[N_{TP} \times N_D]$ matrix of input training points. Construction of the surrogate model also requires training values provided as a matrix of size $[N_{TP} \times N_Y]$, where each training point input has a paired training value (output) for each prediction model (Y).

The objective of this chapter is to develop an efficient selection process to determine the optimal quantity of parameters (N_D) for a surrogate model emulating the Level 1 MEPDG design procedure. Three selection processes are investigated: a correlation matrix method, ANOVA method, and a GP length-scale factor method. Sensitivity analysis will provide insight to the most significant design information for Level 1 analyses.

IV.2 Surrogate Model Construction: Initialization

Surrogate model construction requires selection of the quantity of parameters for each training point (N_D). In selecting the training data, one must consider the limitations of surrogate models and choose to vary only the most important input parameters, as

increasing the number of inputs will require larger amounts of training data to estimate an accurate surrogate model. Further, evaluation of the actual function is computationally expensive. Utilizing a surrogate model that includes all the parameters for the true function may not improve the speed to evaluate the function, negating the purpose of construction of the surrogate. The number of input parameters in the surrogate model should be reduced to only those variables that contribute significantly to the function output, thereby greatly improving speed with a minimal sacrifice in accuracy. Due to the large quantity of parameters utilized by the MEPDG, the selection of the variables for the MEPDG surrogate model requires determination of the most critical design parameters. It is assumed that although the MEPDG utilizes thousands of input parameters, there is a limited few with greater influence on the pavement design process than others. Experience with pavement analysis and design would suggest that the most important parameters would be layer thickness, material properties, and traffic volume. Sensitivity studies such as the one undertaken by Ayyala, et al. (61) will allow for identification of the most critical design parameters.

IV.2.1 Quantity of Training Point Parameters (N_D)

Fifty three parameters are chosen as candidate input parameters for the surrogate model variable selection process. These parameters, excluding the binder viscosity, were chosen to vary within specific, typical ranges, partially derived from statistical information summarized by Huang (37), Darter et al. (39), and Rada et al. (62), outlined in TABLE IV.1.

These input parameter values and their ranges were chosen to represent a potential pavement design and incorporate variability due to sources such as construction tolerances, measurement errors, or variation in traffic. The construction of the surrogate model is tailored to a specific design and will be trained and verified according to that design application. For each differing application, it is the designer's responsibility to select training points and perform model verification according to the input parameters of interest to that application, and this can be performed according to the framework presented here.

TABLE IV.1: MEPDG Input Parameter Ranges

| Parameter Name | Minimum Value | Maximum Value |
|-----------------------|----------------------|----------------------|
| AADT | 1300 | 1700 |
| LDF | 80 | 90 |
| OpSpeed | 60 | 70 |
| Class4 | 0.1 | 10.7 |
| Class5 | 2.7 | 20.1 |
| Class6 | 0.2 | 12.8 |
| Class7 | 0 | 10 |
| Class8 | 0.1 | 15.9 |
| Class9 | 55.8 | 68.2 |
| Class10 | 0 | 14.2 |
| Class11 | 0.2 | 12.4 |
| Class12 | 0.1 | 11.4 |
| TrafficGrowth | 3 | 5 |
| MeanWheel | 16.2 | 19.8 |
| Wander | 9 | 11 |
| LaneWidth | 10.8 | 13 |
| AxleSpTand | 46.4 | 56.8 |
| AxleSpTri | 44.3 | 54.1 |
| AxleSpQuad | 44.3 | 54.1 |
| AxleAveSpShort | 12 | 15 |
| AxleAveSpMed | 15 | 18 |
| AxleAveSpLong | 18 | 21 |
| HMAThick | 6 | 10 |
| EBC | 5.3 | 7 |
| AV | 3 | 5.5 |
| UnitWt | 135 | 165 |
| %Ret34 | 98 | 100 |
| %Ret38 | 80 | 93 |
| %Ret#4 | 54 | 76 |
| %Pass#200 | 0 | 6.2 |
| ThermalCond | 0.6 | 0.74 |
| HeatCap | 0.21 | 0.25 |
| Gstar40 | 4514250 | 32985750 |
| Gstar55 | 3300600 | 4499400 |
| Gstar70 | 234605 | 3298465 |
| Gstar85 | 200015 | 229985 |
| Gstar100 | 25088 | 199913 |
| Gstar115 | 14006 | 24995 |
| Gstar130 | 3505 | 13995 |
| delta40 | 47 | 55 |
| delta55 | 55 | 57 |
| delta70 | 57 | 63 |
| delta85 | 63 | 67 |
| delta100 | 67 | 70 |

TABLE IV.1, continued

| | | |
|----------|-------|-------|
| delta115 | 70 | 73 |
| delta130 | 73 | 82 |
| GBThick | 7 | 9 |
| GBMod | 35502 | 39998 |
| GBpois | 0.1 | 0.4 |
| GBlat | 0.5 | 0.6 |
| SubMod | 12006 | 23994 |
| Subpois | 0.2 | 0.3 |
| Sublat | 0.6 | 0.7 |

Where:

AADT: Average Annual Daily Truck Traffic.

LDF: Lane Distribution Factor. (Percent Trucks in the Design Lane.)

OpSpeed: Operational Speed (mph).

Class4 – Class12: AADTT Distribution by Vehicle Class.

TrafficGrowth: Traffic Growth Rate.

MeanWheel: Mean Wheel Location (inches from the lane marking).

Wander: Traffic Wander Standard Deviation (in.).

LaneWidth: Design Lane Width (ft.).

AxleSpTand: Average Axle Spacing for Tandem Axle Trucks.

AxleSpTri: Average Axle Spacing for Tridem Axle Trucks.

AxleSpQuad: Average Axle Spacing for Quad Axle Trucks.

| | |
|-----------------------------------|--|
| <i>AxleAveSpShort, Med, Long:</i> | Average Axle Spacing Wheelbase Distribution. |
| <i>HMAThick:</i> | HMA layer thickness (in.). |
| <i>EBC:</i> | HMA Effective Binder Content (%). |
| <i>AV:</i> | Percent Air Voids in HMA. |
| <i>UnitWt:</i> | HMA Total Unit Weight (pcf). |
| <i>%Ret34:</i> | Percent of aggregates in AC passing the $\frac{3}{4}$ " sieve. |
| <i>%Ret38:</i> | Percent of aggregates in AC passing the $\frac{3}{8}$ " sieve. |
| <i>%Ret#4:</i> | Percent of aggregates in AC passing the #4 sieve. |
| <i>%Pass#200:</i> | Percent of aggregates in AC passing the #200 sieve. |
| <i>ThermalCond:</i> | Thermal conductivity of asphalt (BTU/hr-ft-F°) |
| <i>HeatCap:</i> | Heat capacity of asphalt (BTU/lb-F°) |
| <i>Gstar40 – Gstar130:</i> | Binder Complex Shear Modulus (Pa) at each tested temperature. |
| <i>delta40 – delta130:</i> | Binder Phase Angle (°) at each tested temperature. |
| <i>GBThick:</i> | Granular Base Layer Thickness (in.). |
| <i>GBMod:</i> | Granular Base Layer Modulus (psi.). |
| <i>GBpois:</i> | Granular Base Layer Poisson's ratio. |
| <i>GBlat:</i> | Granular Base Layer Coefficient of Lateral Pressure (Ko). |

| | |
|-----------------|---|
| <i>SubMod:</i> | Sub-grade Layer Modulus. |
| <i>Subpois:</i> | Sub-grade Layer Poisson's ratio. |
| <i>Sublat:</i> | Sub-grade Layer Coefficient of Lateral Pressure (Ko). |

IV.2.2 Location of Training Points for Evaluation of Selection Processes

Training points are chosen for the selection process according to a Latin Hypercube sampling plan, based on the input parameter ranges, as previously discussed. The distribution of the probability over the range was assumed to be uniform for all variables, and 1,000 training points were chosen in total.

IV.3 Selection Process Methods

The variable selection process is important in developing an accurate and efficient surrogate model, but investigation of the significance of a sample of nearly fifty of the Level 1 input variables is a very large “0-1” optimization problem. Evaluation of all 2^{50} possible combinations is extremely expensive. The design of experiments for the surrogate model requires a more efficient process for variable selection.

Several classic, heuristic methods such as a correlation matrix or one-way analysis of variance (ANOVA) can be utilized as methods to determine the variables to be included in the surrogate model. A surrogate modeling parameter, the length-scale factor, can also be utilized as a metric for determining the most significant variables to

include in the model. These three heuristics do not guarantee construction of the most accurate surrogate model, but provide a means for finding a “good” model.

Each of these three methods will be investigated in terms of computational effort and accuracy in the model selection problem. A description of each selection process method follows.

IV.3.1 ANOVA

One way analysis of variance (ANOVA) is a traditional statistical method that can be used to compare two populations of data, describing the variability between the two populations. For construction of the surrogate model, selection of the input parameter is based on improvement in the accuracy of the prediction of the model compared to the actual MEPDG model. For the variable selection process, the F-test statistic is utilized as a metric to rank the input parameters according to their significance. The ANOVA process requires the evaluation of null and alternative hypotheses:

H_0 : Knowledge of the input parameter gives no information about the value of the output value (Y). Therefore, the inclusion of the parameter does not significantly improve the accuracy of the GP model.

H_{alt} : Knowledge of the input parameter gives some information about the value of the output value (Y). Therefore, the inclusion of the parameter does improve the accuracy of the GP model.

The formulation of the ANOVA process also requires definition of the f statistic. The F test utilizes the F distribution and describes the rejection criteria for the null hypothesis. The F statistic can be represented mathematically as:

$$f = \frac{SSR}{\left[\frac{SSE}{(N-2)} \right]} \quad (IV.1)$$

The SSR is the sum of the squared residuals between the input parameters and the MEPDG output which describes the explained variance for all N training points. The SSE term is the sum of the errors squared which describes the unexplained variance. These two terms are defined as:

$$SSR = \sum_{i=1}^I \sum_{n=1}^N (\bar{x}_i - \bar{x}_{..})^2 \quad (IV.2)$$

$$SSE = \sum_{i=1}^I \sum_{n=1}^N (x_{in} - \bar{x}_i)^2 \quad (IV.3)$$

The f statistic provides a ranking system where the parameters with the largest f values more strongly reject the null hypothesis and are, therefore, more significant to the model under investigation.

IV.3.2 Correlation Matrix

The Correlation process utilizes a pair-wise, linear correlation between all parameters and the model output as a ranking method for determining the most significant parameters. The parameters with the greatest correlation with the MEPDG output are assumed to contribute more significantly to the predictive power of the GP model.

IV.3.3 Gaussian Process Model Length-Scale Factors

The Length-Scale Factor (LSF) process requires construction of a Gaussian Process (GP) model and selects variables for inclusion in the surrogate model with regard to the length-scale factor values. The length-scale factor (ξ) is estimated through the GP construction process as described in Chapter VIII.2. Each factor is an indication of the correlation between the variable and output value. Inclusion of the parameters with the largest length-scale factors should, therefore, provide the most significant parameters for inclusion in the surrogate model.

IV.4 Selection Process Comparison

Implementation of each method requires investigation of improvement as a function of computational cost to determine the most efficient selection process. The LSF method, for example, ranks each input parameter according to significance. Selection of the optimal quantity of input variables requires investigation of the improvement in accuracy with the addition of another variable. The adjusted R-squared value has been chosen here as a verification metric that indicates an improvement in GP accuracy adjusted for the

quantity of model parameters (N_D). An improvement in the adjusted R-squared (R^2_{Adj}) value indicates that the addition of the parameter improves predictive capability and should be included in the model.

Comparison of the effectiveness of these models is quantified by an R^2_{Adj} evaluated between the GP model and the true MEPDG predictions. Evaluation of 1,000 MEPDG design sites was performed and this data was used in construction and verification of the GP model. For each GP construction, 100 randomly selected points were defined as verification points and were not utilized as training points. The construction of GP models with $[900 \times N_D]$ was performed for N_D equal to one through fifty-three parameter dimensions. Model verification is evaluated on the remaining, random 100 points.

IV.5 Results

The three variable selection processes must be compared both computationally and by accuracy, described by the R^2_{Adj} statistic. The R^2_{Adj} for each model as a function of N_D is shown in FIGURE IV.1. All variable selection processes demonstrate an ability to select additional parameters in a positive order of significance, consistently improving R^2_{Adj} . The correlation matrix method chooses the most significant parameters for all MEPDG models in a manner that quickly achieves a large R^2_{Adj} value. The correlation matrix ranking procedure is based on a linear relationship between the input parameters and the MEPDG prediction outputs. The performance of this method indicates that the behavior of the trend of all of the prediction models is likely described well by a linear model.

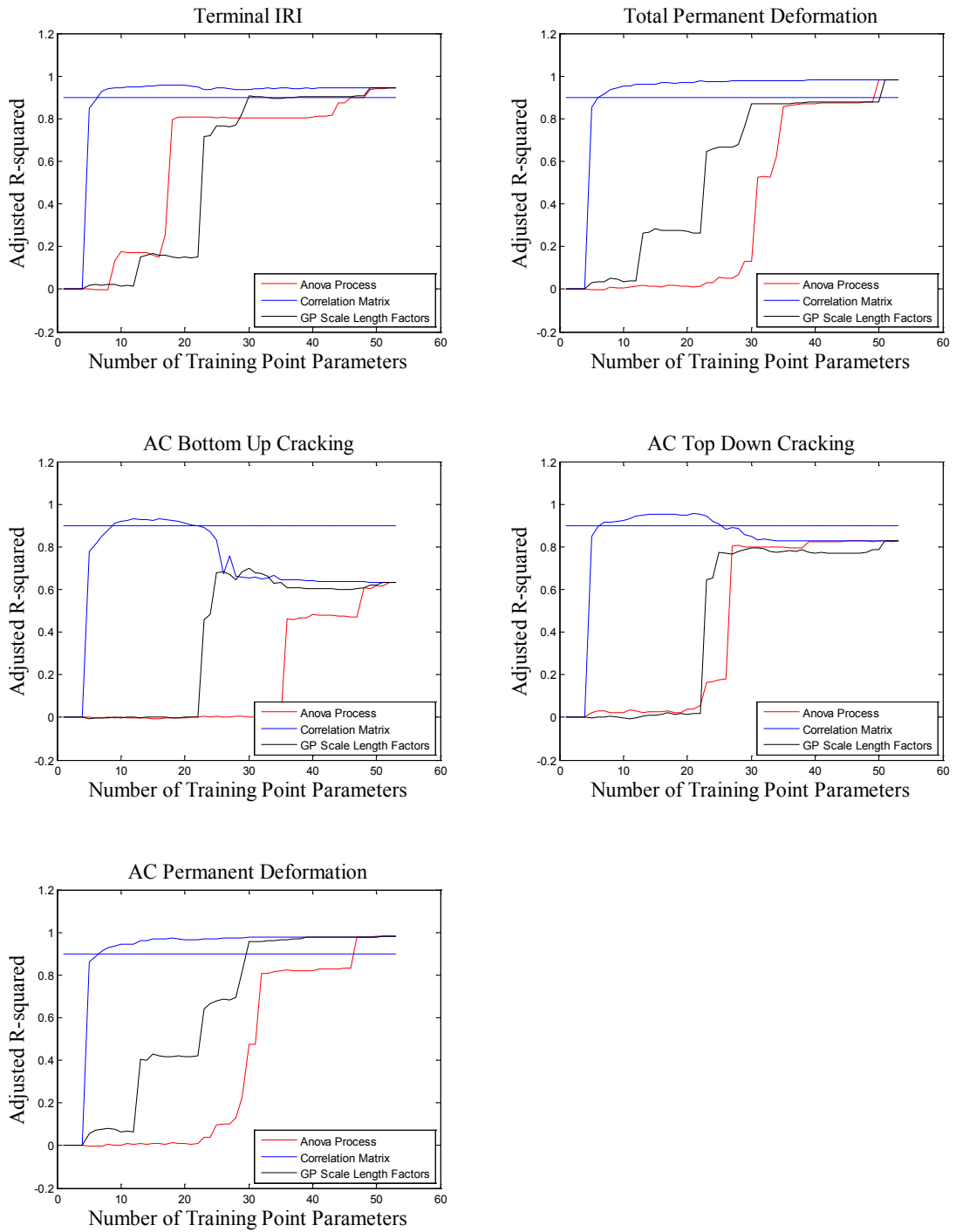


FIGURE IV.1: Adjusted R-Squared Values for the Variable Selection Processes for each Prediction Model

IV.5.1 Sensitivity Analysis: Quantity of Training Point Parameters

In addition to the comparison between selection methods, FIGURE IV.1 can be used to investigate the sensitivity of the accuracy in predictions for each distress model to the quantity of training point parameters. Considering first the Correlation method, each of the five distress modes are shown to be highly sensitive to the quantity of parameters for approximately the first ten parameters. The R^2_{Adj} values for the models that include more than about ten parameters are nearly equal, demonstrating that the addition of parameters beyond this quantity provide a minimal amount of improved accuracy in predicted performance. The Anova and LSF methods do not perform in the same manner as the Correlation method and do not achieve the plateau, or convergence, in the R^2_{Adj} values. The lack of convergence in the R^2_{Adj} value indicates that these processes achieve better accuracy with the addition of parameters and would require a larger number of parameters if a minimum R^2_{Adj} value was required. Further, this lack of convergence in the Anova and LSF methods indicates that the Correlation method is selecting the most significant parameters in a more efficient manner.

IV.5.2 Method for Selection of Training Point Parameters

Selection of the minimum quantity of training points for a surrogate can be performed with respect to a minimum R^2_{Adj} requirement. Considering a minimum requirement that the model achieve an R^2_{Adj} greater than or equal to 0.9, selection of the most significant parameters can be made for each distress model. TABLE IV.2 outlines the quantity and parameters required to achieve this standard, selected through the correlation matrix

method. This method is chosen because, as shown in FIGURE IV.1, this method consistently chooses the minimum quantity of training point parameters to quickly achieve large R^2_{Adj} values.

TABLE IV.2: Training Point Parameters Using Correlation Matrix

| | Terminal IRI | Total Permanent Deformation | AC Bottom-Up Cracking | AC Top-Down Cracking | AC Permanent Deformation |
|----------------|---------------|-----------------------------|-----------------------|----------------------|--------------------------|
| Parameter Name | HMAThick | HMAThick | HMAThick | HMAThick | TrafficGrowth |
| | TrafficGrowth | TrafficGrowth | AV | AV | HMAThick |
| | %Pass#200 | %Pass#200 | %Pass#200 | SubMod | %Pass#200 |
| | AV | SubMod | TrafficGrowth | TrafficGrowth | %Ret#4 |
| | %Ret#4 | %Ret#4 | SubMod | %Pass#200 | AADTT |
| | SubMod | AADTT | EBC | AxleSpTand | Wander |
| | AADTT | Wander | GBMod | EBC | EBC |
| | | | %Ret#4 | | |

The results in TABLE IV.2 provide insight into the sensitivity of these distress modes to the input parameters. Only twelve unique design parameters are necessary to adequately model all five pavement distress modes. Parameters describing the asphalt layer and material strength for all layers are shown to significantly impact pavement performance. The thickness of the HMA layer (*HMAThick*) is significant to all distress modes, which is not unexpected. Improved performance for the permanent deformation models relies heavily on the thickness of the HMA layer. The deformation is calculated as a sum of the product of strains and thicknesses for each layer in the structure, so modifications in the thickness of the HMA layer is significant, especially in the two layer pavement system evaluated here. Additional properties such as asphalt air voids (*AV*) and effective binder content (*EBC*) also are significant to most distress models. The fatigue

cracking models are evaluated as a function of strains and stresses in the asphalt layer, directly impacted by the asphalt thickness and these material properties. The sub-grade modulus is another parameter that is shown to be significant in the distress models. Again, the strength of this layer impacts the stresses and strains utilized in all the distress functions. The impact of traffic growth is demonstrated to significantly impact all five distress modes. The likely cause of this significance is the impact of this projected growth on the accumulation of stresses and strains over time. Greater projected traffic growth would be expected to increase the rate of accumulation of strains in the pavement.

IV.6 Conclusion

M-E design methods are computationally expensive to evaluate and hinder highly iterative design processes such as design optimization and sensitivity analyses. The construction of a surrogate model alleviates this computational expense, but its approximation to the true distress models introduces model form error. Minimization of this model form error is achieved through appropriate selection of training points for the surrogate model.

Construction of a well-trained surrogate model requires appropriate selection of the quantity of training point dimensions (N_D) in an efficient manner. The surrogate must accurately imitate the true functions, but should incorporate only the most significant design parameters. A method for selecting N_D should not add significant expense to the construction process and should require a minimum number of true model evaluations. The computational expense for the ANOVA and Correlation methods are very similar.

The standard statistical techniques require few function evaluations and the programming is available in most common computational software. The evaluation of the length-scale factors is more expensive than the alternate methods as it requires construction of the full 53-parameter GP model. The LSF method may be more appropriate for sensitivity analyses, where the objective is to understand the impact of all parameters under investigation, but is not likely computationally effective for construction of GPs for practical use. Further, the LSF method is a back-solved problem, where construction of the less-parameter GP requires prior construction of larger models.

Once selection of the quantity of dimensions for the surrogate model is performed, the location and quantity of training points must be determined. Efficiency in construction of the surrogate is necessary for practical implementation of surrogate modeling for M-E pavement design. Current implementations of the M-E procedure are computationally expensive, but a surrogate model can alleviate this expense and provides a powerful tool for advanced design processes.

Construction of a surrogate model that accurately emulates the M-E design procedure allows for computationally efficient evaluation of highly iterative analyses, but accurate evaluation must also incorporate uncertainty that impacts flexible pavement design. Reliability analysis, design optimization, and sensitivity analyses are necessary to provide accurate and reliable performance predictions, but these processes are impacted by uncertainty due to model form errors and input parameter variability. A comprehensive approach to management of uncertainty from these sources is necessary for accurate analyses.

CHAPTER V

UNCERTAINTY PROPAGATION WITH SURROGATE MODELS

V.1 Introduction

Flexible pavement design is significantly impacted by uncertainty due to input parameter variability and model form error. Reduction of model form error and improvements in computational speed of M-E design procedures greatly improves the accuracy in prediction of flexible pavement performance; however no comprehensive approach to uncertainty management has been proposed for the AASHTO MEPDG. This is largely due to the computational expense associated with evaluating the MEPDG. Introduction of a well-trained surrogate model eliminates this issue and allows for robust methods, such as Monte Carlo Simulation (MCS) for incorporating model uncertainty into reliability analyses. Calibration of predictive models as demonstrated in Chapter III will reduce, but not eliminate, model prediction error. The process of calibration quantifies the model form error which can be included in design and analysis processes.

The development of a comprehensive approach for uncertainty management in pavement design is necessary because without accounting for all sources of uncertainty, reliability will be incorrectly estimated. The uncertainty in prediction of the model will lead to uncertainty in the design life of the pavement, but the variability in the design inputs will be a second, additive source of uncertainty in the design life of the pavement. As a result, the MEPDG design process will incorrectly state the reliability level of the

pavement. Methods of reliability analysis based only on input variability also are subject to the same overstatement (or understatement) of reliability because they do not take into account a lack of fit of distress prediction equations. Understanding the significance of the impact of uncertainty due to these various sources is critical in verification of the MEPDG predictive distress models and for reliable pavement design.

In this chapter, quantification of significant sources of uncertainty is performed, a method for uncertainty propagation is demonstrated, sensitivity of the predicted performance models to the sources of uncertainty is included, and a reliability analysis is demonstrated and discussed. Uncertainty due to the MEPDG predictive model, uncertainty introduced through use of a surrogate model, and variability due to the stochastic nature of the pavement design parameters all contribute to uncertainty in overall pavement design. A method for propagation of these sources of uncertainty is outlined in FIGURE V.1. Sensitivity analysis evaluates the sources of uncertainty and their impact on accuracy of predicted behavior. The sensitivity analysis presents the contribution to overall variance in the predicted values from each source of uncertainty. Lastly, results of a simulation-based reliability analysis are presented and the impact on the predicted reliability is discussed.

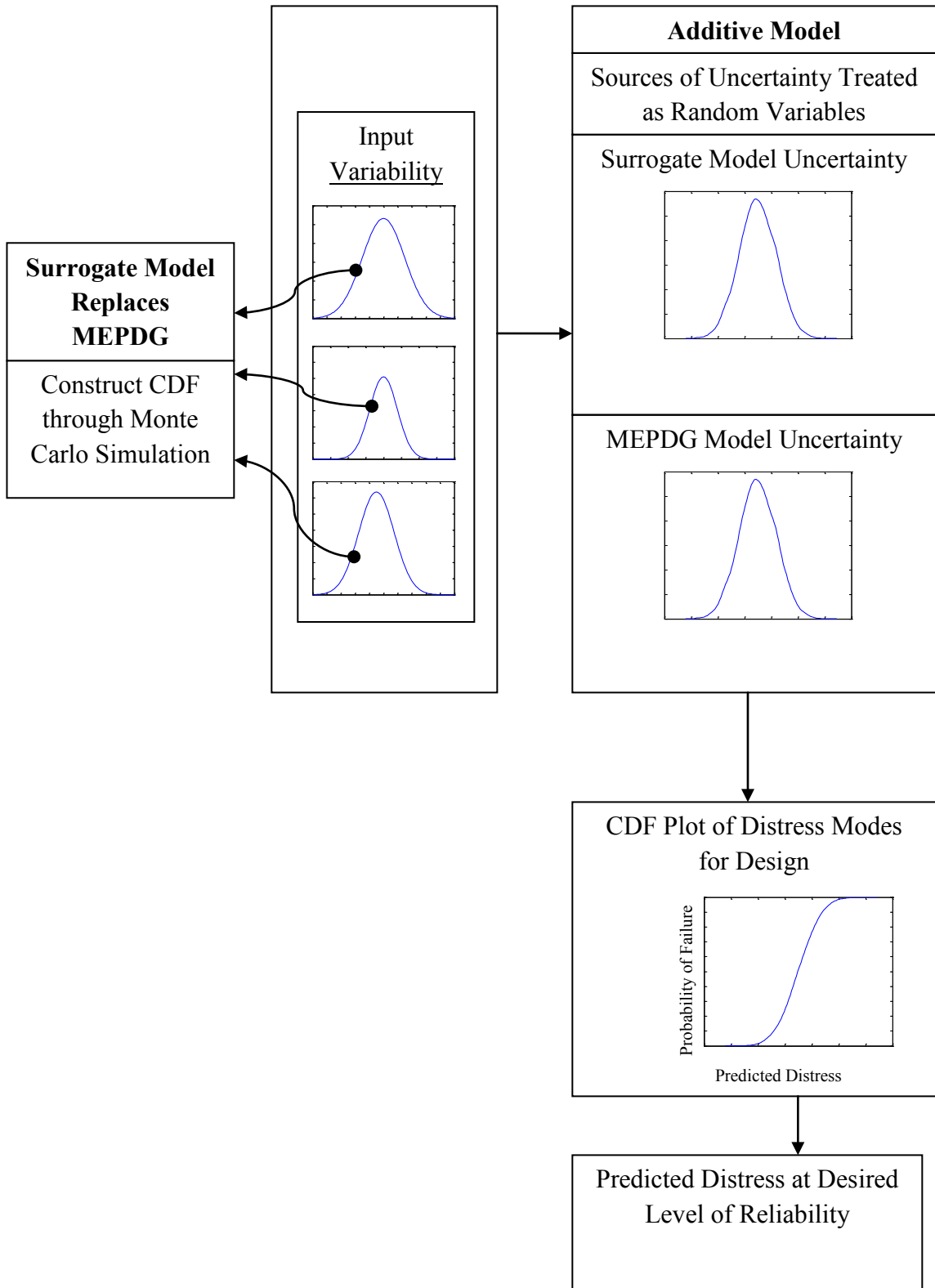


FIGURE V.1: Proposed Method of Design to Incorporate All Sources of Uncertainty

V.2 Sources of Uncertainty

Three sources of uncertainty are considered: Input parameter variability, MEPDG model uncertainty, and surrogate model uncertainty. The statistical properties for each are discussed in the following sections.

V.2.1 Input Parameter Variability

Input parameters utilized by the MEPDG have inherent variability. Numerous databases of empirical data capable of providing statistical properties regarding such input parameters as material thicknesses, material strength properties, and traffic loadings are available (6). In instances where these distributions are not known, testing should be performed or expert opinion should be used.

V.2.2 GP Model Predictive Uncertainty

Construction of a well-trained surrogate model contributes to predictive uncertainty through approximation errors between the surrogate and the true performance function. Construction of a surrogate model defines a design domain from which GP predictions of pavement performance can be made. The GP function is defined as a Gaussian conditional distribution dependent on the training data and the correlations between the data and any new point. Design points selected within this domain, but not utilized in the training of the surrogate, can be evaluated by the GP. These GP predictions, conditioned on the training data, estimate the best expected performance value (GP mean) and a corresponding GP variance. Uncertainty from the surrogate model can be quantified

using the residual between the GP mean, or predicted GP value, and value of the true function, as a random variable characterized by the evaluations of numerous non-training points. The standard deviation can be calculated based on the standard deviation of the residuals between the true and surrogate functions. The probability distribution is treated as Gaussian, making the assumption that the error follows the same Gaussian form as the GP prediction function.

V.2.3 MEPDG Predictive Uncertainty

MEPDG uncertainty arises primarily from model form error that exists in the M-E performance prediction models. MEPDG model predictions evaluate a design site at a mean value and modify the design prediction utilizing statistical data related to LTPP empirical data. The MEPDG “reliability analysis” is similar to a traditional model confidence calculation in which empirical data is compared to model predictions and the mean and standard deviations of the residuals are used to define the confidence that the model is accurately predicting the true behavior. Where the word *reliability* is used in the MEPDG, this dissertation will use the terminology *model confidence*. The word *reliability* will refer to the probability that the distress prediction is given at a specified level of *model confidence*. The method of incorporating model confidence in the MEPDG is implemented by evaluation of a design site at a mean value and modifying the predicted performance as a function such as that described by Equation V.1.

$$Prediction_p = Prediction_{mean} + STD_{PredModel} * Z_p \quad (V.1)$$

The prediction of the performance at a specified “reliability level” (p) is expressed as the prediction evaluated at the mean plus the standard deviation multiplied by a standardized normal variate corresponding to the reliability level. The standard deviation of the prediction model is obtained by regression analysis between measured and predicted values utilized in calibration of the MEPDG prediction functions. Predictions are therefore penalized as they move away from prediction means where model confidence is higher.

Uncertainty due to the MEPDG is back-solved from the model confidence formulation to determine the statistics of the model form error for the specific data set chosen. The distribution of the uncertainty is treated as a normal distribution with zero mean in accordance with the method developed in the MEPDG. The standard deviation of the MEPDG is calculated as a ratio between a margin of safety and the inverse of the standard normal cumulative distribution function (CDF) evaluated at the MEPDG-calculated probability of failure. The procedure for a single distress mode is outlined in the following steps:

1. Calculate the MEPDG margin of safety (MS) as the difference between the limiting acceptable value for the distress mode, or threshold value, and the predicted MEPDG output for the given design input.
2. Evaluate the inverse of the standard normal distribution at the probability of failure.

$$z = \Phi^{-1}(P_f) \quad (V.2)$$

3. Calculate the standard deviation of the distress mode.

$$\sigma_{MEPDG} = \left| \frac{MS}{z} \right| \quad (V.3)$$

V.3 Uncertainty Propagation and Sensitivity Analysis

Uncertainty propagation incorporating all significant sources of uncertainty is necessary for flexible pavement design utilizing M-E design procedures. The following numerical experiment demonstrates a procedure for uncertainty propagation utilizing the MEPDG. The proposed methodology is implemented to demonstrate the importance of each source of uncertainty.

V.3.1 Uncertainty Propagation Method

An additive model for uncertainty propagation incorporating input variability and model form error is considered for application with the MEPDG flexible pavement design procedure. Input variability, MEPDG model form error, and surrogate model uncertainty are considered the most significant sources of uncertainty. Treating these three sources of uncertainty as random variables, MCS can be performed to determine the impact of the combined uncertainties on each distress mode, as shown in FIGURE V.1. It is assumed that the uncertainty in the terminal distresses is represented by the random variable $D_i(x)$ such that:

$$D_t(x) = \tilde{D}_t(x) + u_{MEPDG} + u_{surrogate} \quad (V.4)$$

where x is a vector of random design inputs, $\tilde{D}_t(x)$ the GP prediction of the distress, u_{MEPDG} is a random variable representing the MEPDG predictive uncertainty, and u_{GP} is a random variable representing the discrepancy between the MEPDG and GP predictions. Each MCS realization of $D_t(x)$ is generated by sampling the random design inputs.

Four CDF's can be constructed for each distress mode to understand the uncertainty introduced by each source of variability as:

1. Model input variability only, where:

$$D_t(x) = \tilde{D}_t(x) \quad (V.5)$$

2. Model input variability and MEPDG predictive uncertainty, where:

$$D_t(x) = \tilde{D}_t(x) + u_{MEPDG} \quad (V.6)$$

3. Model input variability, MEPDG predictive uncertainty, and GP uncertainty, where:

$$D_t(x) = \tilde{D}_t(x) + u_{MEPDG} + u_{GP} \quad (V.7)$$

4. The current MEPDG uncertainty estimate:

$$D_t(x) = \tilde{D}_t(\bar{x}) + u_{MEPDG} \quad (\text{V.8})$$

where \bar{x} is the mean or nominal value of x .

The resulting family of CDF's can be used to perform sensitivity analysis for each distress mode. Sensitivity analysis provides important information regarding the impact of the sources of uncertainty both independently and in combination. The relative importance of input variability, MEPDG predictive uncertainty, and GP uncertainty can be found by comparing the variances of $\tilde{D}_t(x)$, u_{MEPDG} , and u_{GP} . Quantifying the relative contributions of these uncertainties allow a designer to understand whether the uncertainty in the final prediction is reducible through stricter quality control and/or gaining more information about the random variables, whether the uncertainty is due to a lack of fit of the MEPDG and/or measurement error, or whether the uncertainty could be significantly reduced by refining the surrogate model.

It should be noted that the MEPDG reliability estimate may not be conservative. By neglecting the variability in the input parameters, the distribution of the design life may have a smaller variance and a distribution with lighter tails than the distribution most representative of the true state of uncertainty. Therefore, the MEPDG estimate of the failure probability will be less than what is appropriate, given the overall state of uncertainty. The proposed methodology, by contrast, is always conservative. Even if part of the reason for the lack of fit of the MEPDG is variability in critical design inputs, the

estimate of reliability given by the proposed methodology will be a conservative estimate as this uncertainty would be doubly counted.

V.3.2 Numerical Experiment

In the following sections, a method for uncertainty propagation is demonstrated, sensitivity of the predicted performance models to the sources of uncertainty is included, and a reliability analysis is demonstrated and discussed.

V.3.2.a Surrogate Model Construction

Eight variables were chosen to construct the GP necessary to demonstrate this approach to uncertainty propagation. The eight variables chosen were: annual average daily truck traffic (AADTT), hot-mix asphalt (HMA) thickness, granular base (GB) thickness, effective binder content (EBC) of the asphalt layer, air void ratio (AV) of the asphalt layer, modulus of subgrade (E_{subgrade}), modulus of GB (K_{GB}), and one parameter representing the binder viscosity, A. These parameters, excluding the binder viscosity, were chosen to vary within specific ranges as outlined in TABLE V.1. The potential values for the binder viscosity term, for a MEPDG Level 3 analysis, cannot be selected in the same manner, but must be selected from a finite set of values associated with one of three viscosity grade families. The binder viscosity parameter, A, was chosen randomly from a set of six potential values within the “conventional viscosity grade” sub-set. This viscosity grading system provides default MEPDG design inputs describing the

relationship between viscosity and temperature for the asphalt concrete and is based on AASHTO M226.

The distribution of the probability over the range was assumed to be uniform for all eight variables, and 150 training points were chosen in total. An additional 10 points were generated randomly for use in verification of the surrogate model.

TABLE V.1: Input Parameter Ranges

| Parameter Name | Minimum Value | Maximum Value |
|-----------------------------|---------------|---------------|
| AADTT | 1300 | 1700 |
| HMA Thickness (in.) | 6 | 10 |
| GB Thickness (in.) | 4 | 12 |
| EBC | 5% | 15% |
| AV | 6.5% | 10.5% |
| E_{subgrade} (psi) | 13,000 | 16,000 |
| K_{GB} (psi) | 38,000 | 42,000 |
| AADT | 1300 | 1700 |

V.3.2.b Uncertainty Propagation

Uncertainty analysis performed in this chapter includes investigation of the impact of three major sources of uncertainty, and their effects individually and in combination. Monte Carlo Simulation (MCS) is performed and the results are utilized to obtain cumulative distribution functions of the various distress modes, incorporating the various sources of uncertainty. The impact of uncertainties can be analyzed visually.

Uncertainty associated with the input parameters is calculated through MCS of the GP model, choosing 10,000 samples. Samples of each input parameter are generated randomly, from a normal distribution as defined in TABLE V.2, and the GP model is

evaluated at all samples. MEPDG model uncertainty is sampled according to the normal distribution with zero mean and a standard deviation calculated based on the experimental data. The uncertainty due to the GP model is sampled as a random variable, similar to the MEPDG uncertainty, with the appropriate distribution and distribution parameters.

TABLE V.2: Probability Distributions of Random Variables

| Random Variable | Probability Distribution | Mean | Standard Deviation |
|-----------------------------|--------------------------|------------|--------------------|
| AADTT | Normal Random Variable | 1500 | 150 |
| HMA Thickness (in.) | Normal Random Variable | 8 | 0.78 |
| GB Thickness (in.) | Normal Random Variable | 8 | 1.25 |
| EBC | Normal Random Variable | 10% | 1% |
| AV | Normal Random Variable | 8.5% | 0.85% |
| E_{subgrade} (psi) | Normal Random Variable | 14,500 psi | 1250 psi |
| K_{GB} (psi) | Normal Random Variable | 40,000 psi | 1750 psi |
| A | Constant | 10.7709 | |

The uncertainty in the predicted pavement performance due to the MEPDG and GP is shown in TABLE V.3 for all distress modes. The MEPDG uncertainty was calculated by the method previously described, utilizing the MEPDG output values for 170 design sites. The same 170 design input sets were used in the training and calibration of the GP model. To determine the contribution of uncertainty due to a GP model, 160 training points were used in construction of the GP and the remaining 10 MEPDG

evaluations, or verification sets, were utilized to calculate the mean and standard deviation of the residual between the MEPDG and the GP. To avoid biases in training and verification data, the 160 training points were selected randomly and the entire construction process was repeated 10,000 times. The uncertainty in TABLE V.3 represents the mean and standard deviation values from all 10,000 iterations.

The data shown supports the use of a GP in uncertainty propagation for these prediction performance models. Contrarily, MEPDG model uncertainty significantly impacts the uncertainty in predicted behavior. Comparison of the means of the GP and MEPDG uncertainties to the commonly accepted threshold values clarifies the importance of incorporating these uncertainties into pavement design. For example, the Terminal IRI standard threshold value is 200 in./mi., which can be significantly impacted by an uncertainty due to model form error of 34 in./mi. The uncertainty due to the GP model error is negligible at less than 1.

TABLE V.3: MEPDG and GP Model Uncertainty Distribution Parameters

| Distress Mode | MEPDG | GP Bias | |
|--------------------------------|-----------|-------------|-----------|
| | Std. Dev. | Mean | Std. Dev. |
| Terminal IRI | 34.0220 | -0.3081 | 0.8445 |
| AC Surface Down Cracking | 1821.3 | 20.6545 | 105.6707 |
| AC Bottom Up Cracking | 9.4563 | 0.0371 | 0.1769 |
| AC Permanent Deformation | 0.0990 | -7.7039e-04 | 0.0013 |
| Total Permanent Deformation | 0.1223 | -5.6077e-04 | 0.0028 |

The family of CDF's for all the distress models are shown in FIGURE V.2. The Figure visually demonstrates the impact of uncertainty on the predicted performance and reliability level for the models. For all models, the CDF of all three sources of uncertainty is very similar to that without the GP uncertainty. This confirms that the GP model's uncertainty is negligible compared to that of the MEPDG uncertainty. It is necessary to note that the results from the numerical example shown in FIGURE V.2 demonstrate a model form error that must be corrected for practical implementation of this method. The MCS evaluation results in predicted pavement performance outside the feasible range of output values for the two cracking models. This is clearly evident in the AC Surface Down Cracking model, for which the CDF plots extend well beyond the minimum physically feasible value of zero. The error in these predictions can be attributed to a number of sources, but is most likely due to MEPDG model form error. To eliminate this

error, the input parameter variability must be verified to exist within the empirical data utilized to derive the MEPDG model, accurate field measurements must be utilized to calibrate the models at these input parameters, and the GP must be shown to accurately emulate this model.

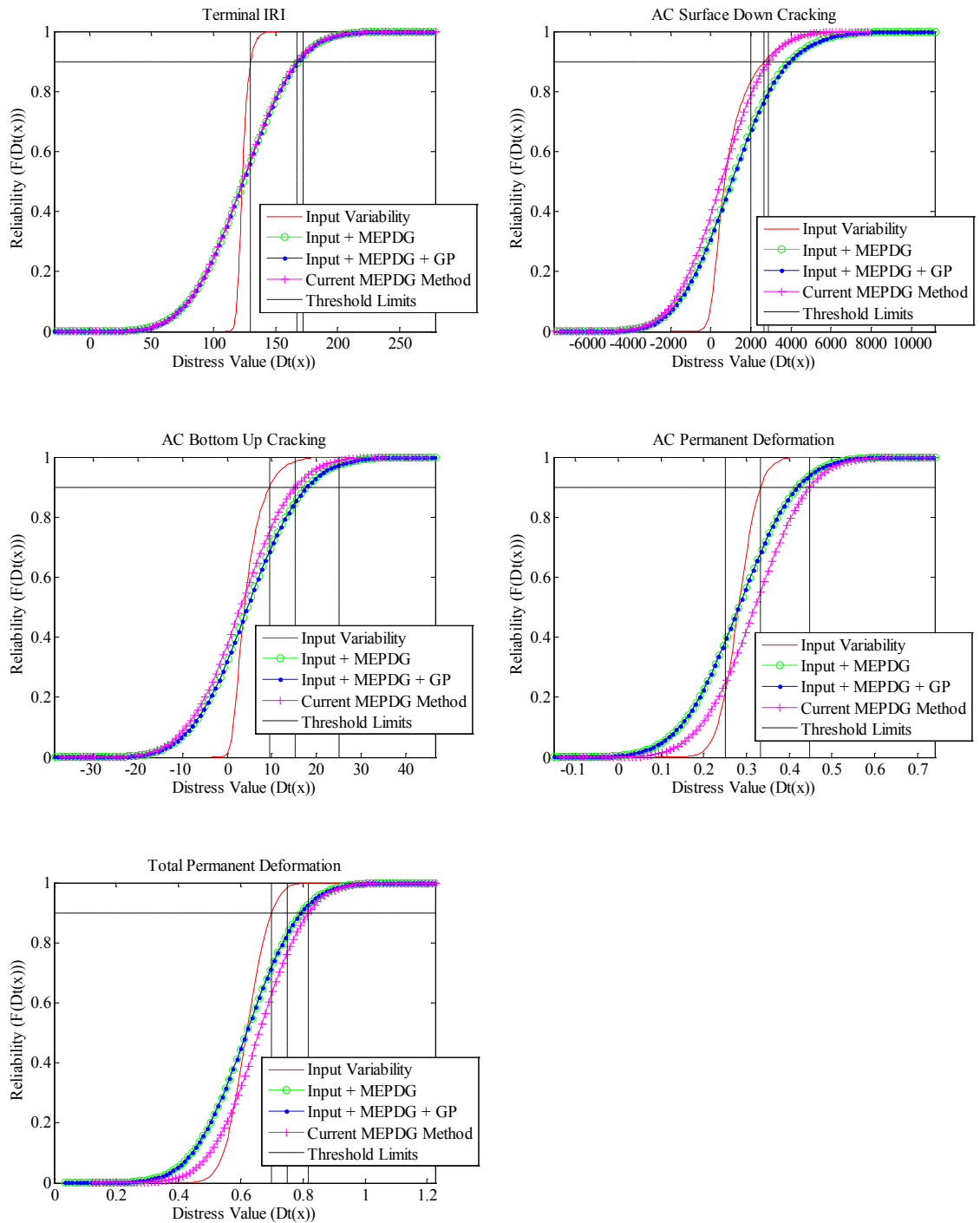


FIGURE V.2: Family of CDFs for Distress Modes

FIGURE V.2 describes the impact of uncertainty on the predicted pavement performance. The impact of uncertainty on the final predicted performance and reliability level is demonstrated in the range of predicted distress values between the CDF plots at a specified reliability level. While the impact varies across distress modes, concern exists when these bands are large relative to the magnitude of the distress. Further, the total permanent deformation model presented in this numerical example demonstrates the potential of accepting a design that exceeds the threshold value. The band of uncertainty overlaps the threshold value at a specified reliability level. A prediction that incorporates only input parameter uncertainty is neglecting model form error which, when included, indicates that the design does not meet the required threshold limit. A complimentary comparison can be made in terms of the required threshold level and required level of reliability. A commonly accepted threshold limit for the AC surface-down cracking distress mode is 2000 ft./mi. . Disregard of model uncertainty can ultimately result in a design that does not perform to the required level of reliability. The prediction for AC surface-down cracking disregarding model form error satisfies a higher reliability level compared to the prediction incorporating model form error.

V.3.2.c Relative Importance of Sources of Uncertainty

The three major sources of uncertainty investigated impact pavement design to varying degrees. TABLE V.4 shows the percent contribution to overall variance for each distress mode. These results indicate that the uncertainty of the input parameters and GP are less significant compared to the predictive uncertainty in the MEPDG. The minimal contribution to variance by the GP model is expected as the model requires thorough

verification during construction. A more robust GP with additional training points and/or dimensions would be necessary to reduce this form of epistemic model error.

Input variability cannot be neglected if accurate reliability estimation is to be achieved in M-E design. The contribution of input variability, specifically with the AC surface down cracking model, indicates the need to accurately measure and model the design parameters and their variability. Quality control and construction processes can reduce the variability in the input parameters which will likely reduce the contribution of this type of uncertainty to this distress model.

TABLE V.4: Percent Contributions to Overall Variance

| Distress Mode | Input Parameter | MEPDG | GP |
|-----------------------------|-----------------|--------|-------|
| Terminal IRI | 1.74% | 98.20% | 0.06% |
| AC Surface Down Cracking | 28.20% | 71.56% | 0.24% |
| AC Bottom Up Cracking | 11.00% | 88.97% | 0.03% |
| AC Permanent Deformation | 14.35% | 85.63% | 0.01% |
| Total Permanent Deformation | 19.36% | 80.60% | 0.04% |

Sensitivity of the model prediction to these sources of uncertainty can be visualized with the contour plots presented in FIGURE V.3. These plots are developed utilizing the surrogate models constructed in Chapter III. One thousand MCS points were sampled,

holding all parameters at their mean values except AADTT and HMA thickness. The plots demonstrate the behavior of the distress modes over the range of parameter values and are useful in comparing the impact of the sources of uncertainty on the predicted performance of flexible pavement.

Model uncertainty due to the MEPDG is another source of uncertainty that significantly impacts predicted performance. Pavement design utilizing the MEPDG incorporates model uncertainty after evaluation of the distress functions at the means of the input parameters. The method for incorporating this model confidence in the M-E design functions makes a very significant assumption: that the expectation of the function is equal to the function evaluated at the means of the input parameters. Typically, for highly non-linear functions, this assumption is incorrect. The contour plots for the total permanent deformation and AC bottom up cracking clearly demonstrate that these prediction functions are not linear; therefore demonstrating that this assumption is incorrect for the distress functions utilized in M-E pavement design.

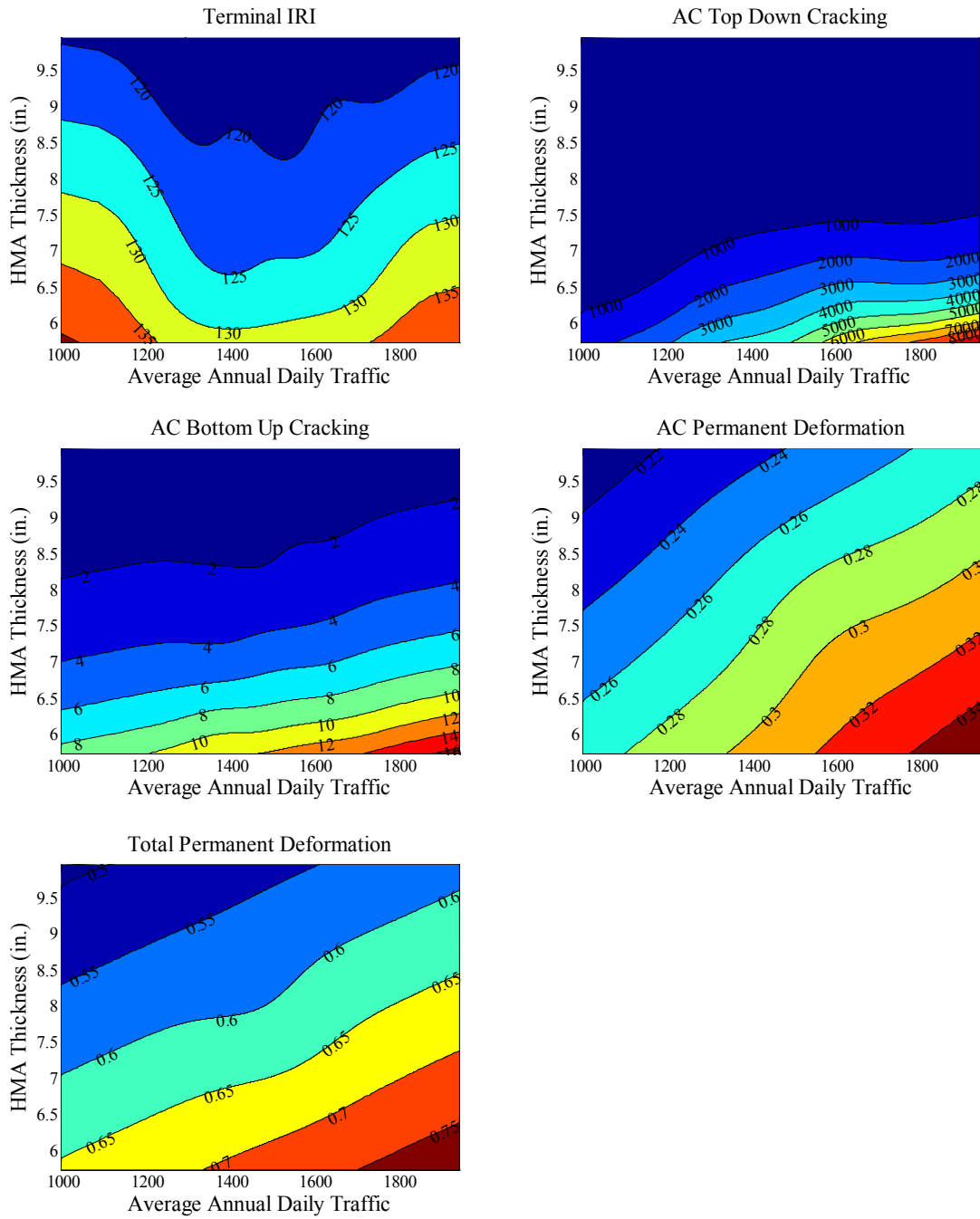


FIGURE V.3: GP Prediction Contour Plots Evaluated at Means of Other Parameters

V.3.2.d Reliability Analysis

Reliability analysis has been performed by MCS with 10,000 samples for the numerical example under consideration. Both component and system reliability analyses were performed and are reported in TABLE V.5. Component reliability analysis was performed by comparing the predicted distress with the threshold values in Eqn. (V.9). For the system reliability analysis, the pavement was considered to fail if any of the distresses exceeded their threshold values.

$$\text{Terminal IRI :} \quad D_i(\bar{x}) \leq 172 \text{ in. / mi.}$$

$$\text{AC Surface Down Cracking :} \quad D_i(\bar{x}) \leq 2000 \text{ ft. / mi.}$$

$$\text{AC Bottom Up Cracking :} \quad D_i(\bar{x}) \leq 25 \text{ percent lane area}$$

$$\text{AC Thermal Cracking :} \quad D_i(\bar{x}) \leq 1000 \text{ ft. / mi.}$$

$$\text{AC Permanent Deformation :} \quad D_i(\bar{x}) \leq 0.25 \text{ in.}$$

$$\text{Total Permanent Deformation :} \quad D_i(\bar{x}) \leq 0.75 \text{ in.} \quad (\text{V.9})$$

Analysis of the results shown in TABLE V.5 indicate that there are significant differences in reliability estimates obtained from methods using primarily input variability and methods focused on primarily predictive uncertainty. When only input variability is considered, the failure probability estimates may be fairly low, as in the first

column. However, neglecting model uncertainty can lead to a very significant understatement of the failure probability, and the failure probabilities in the second column are much higher as model predictive uncertainty is taken into account. It should be recognized that some uncertainty is introduced through the use of surrogate models. When this uncertainty is accounted for, there is a slight increase in the failure probability estimate. However, this discrepancy is very slight. This result reinforces the point that the uncertainty introduced through use of the surrogate models is negligible in comparison with the other major sources of uncertainty. The use of GP models has very little influence in the results of reliability analysis.

The fourth column in TABLE V.5 presents reliability results using a method most similar to that implemented by the MEPDG. In some cases, this failure probability estimate is significantly lower than those obtained in column two, but in others it is higher. This is a somewhat counterintuitive result. It would be expected that the MEPDG would systematically understate the failure probability, but there is a reason why this is not always observed in practice. *For nonlinear functions of random variables, the expectation of the function of random variables is not equal to the value of the function evaluated at the expectations of the random variables.* The results of the MEPDG are very nonlinear in the inputs, but the reliability analysis procedure in MEPDG has made the assumption that the expectation of its output is the output at the expectations of the input. As can be seen from the MCS results in FIGURE V.2, the error in this estimate of the mean value of the distress function can be large, and the resulting reliability results can be very inaccurate.

TABLE V.5: Probability of Failure of Distress Modes Including Sources of Uncertainty

| Distress Mode | Probability of Failure | | | |
|--|------------------------|-----------------------------|----------------------------------|----------------------------|
| | Input Variability Only | Input and MEPDG Uncertainty | Input, MEPDG, and GP Uncertainty | MEPDG Uncertainty Estimate |
| Terminal IRI | 0.00% | 8.09% | 7.88% | 7.64% |
| AC Surface Down Cracking (Long. Cracking) | 16.77% | 31.83% | 32.20% | 20.49% |
| AC Bottom Up Cracking (Alligator Cracking) | 0.02% | 2.39% | 2.42% | 0.97% |
| Permanent Deformation (AC Only) | 79.38% | 61.93% | 61.63% | 75.63% |
| Permanent Deformation (Total Pavement) | 1.68% | 16.90% | 16.83% | 23.26% |
| System (All Distress Modes) | 79.81% | 77.50% | 77.39% | 86.20% |

The inaccuracy in the MEPDG reliability estimates allows for the understanding of the true importance of input uncertainty propagation in estimating pavement reliability. Although input uncertainty does not account for a large percentage of the variance in the predicted design life, the underlying models in the MEPDG are nonlinear. The importance of the input uncertainty is in the shifting of the expectation of the output. The bias in the MEPDG's estimate of the mean of the distress causes large errors in the estimate of the reliability.

V.4 Conclusion

This chapter has introduced the development of an all-inclusive approach to uncertainty management for M-E pavement design in which both predictive uncertainty in the MEPDG and the uncertainty in design inputs are taken into consideration. The method includes the construction and verification of a surrogate model and uncertainty quantification resulting from three major uncertainty sources: input parameter variability, MEPDG predictive uncertainty, and surrogate model uncertainty. The numerical experiment presented illustrates the effectiveness of the proposed framework for uncertainty analysis. The use of a surrogate model to emulate the MEPDG reduces the computational expense associated with uncertainty quantification analysis, and has made input uncertainty propagation via MCS affordable.

The results show that the dominant source of uncertainty exists in the predictive uncertainty in the MEPDG. This is in no way an indictment of the models utilized in the MEPDG. This uncertainty is large because many factors are responsible for the performance of pavement sections upon which the MEPDG is calculated, including construction quality and practices; factors not easily captured in the MEPDG. Further, the data are subject to measurement errors in both design inputs and field-measured distresses. Even if the MEPDG was perfect in its predictions, the uncertainty in the calibration data represents a major source of predictive uncertainty.

Though the contribution to variance of input variability is relatively small, it is not an insignificant source of uncertainty. Because the expectation of a nonlinear function of random variables is not equal to the function evaluated at the expectation of the random variables, the MEPDG is subject to bias in its estimation of the mean value of the distress

distribution. This bias is caused by the input variability and can create significant errors in reliability estimation.

The contribution of surrogate modeling uncertainty is very small in relation to that of input uncertainty and model predictive uncertainty. It has shown to cause only small effects on reliability estimates. These results show the accuracy and validity of the use of surrogate models for pavement reliability analysis while harnessing the predictive power of the MEPDG.

By combining the effects of the three sources of uncertainty, this chapter has presented a unified approach to uncertainty analysis. By use of surrogate models, the hurdle of the computational expense of the MEPDG has been eliminated for MCS-based reliability analysis. The ability to perform this analysis corrects for biases in the MEPDG estimate of the expected value of the distress distribution and reliability estimate. Predictive uncertainty in the MEPDG has been accounted for along with the errors introduced through use of surrogate models for a comprehensive approach.

Reliability analysis is an important aspect of pavement design. Accurate prediction of pavement performance is necessary for design optimization, but the method must be practical to implement. Although the methods presented in this chapter provide a framework for simulation-based reliability analyses, many practicing engineers have not been trained in construction of surrogate models. Introduction of a method that incorporates the current M-E design equations, rather than replacing them, may be a more practical implementation for improvement to reliability analysis. Analytical reliability methods have been utilized with success in similar engineering applications and will

utilize the true M-E functions. The computational expense of the M-E design process can be offset by the efficiency of analytical methods that usually require a small number of function evaluations. Ultimately, a comparison in computational efficiency and accuracy of simulation-based methods utilizing a surrogate model and analytical methods utilizing the true M-E design functions is necessary to determine the best method.

CHAPTER VI

ANALYTICAL RELIABILITY METHODS FOR MECHANISTIC-EMPIRICAL FLEXIBLE PAVEMENT DESIGN

VI.1 Introduction

Although a trained surrogate model that accurately emulates M-E design equations saves computational expense, the construction and verification process is no trivial task. Hundreds of evaluations of the true functions may be necessary for the construction and verification of an accurate model and model form error, though often minimized, may not be eliminated. Surrogate modeling enables designers to use robust simulation-based reliability methods, but analytical reliability methods have been shown useful in similar engineering applications. Analytical reliability methods reduce the computational expense of the reliability analysis, typically by approximations in the behavior of the limit state function near the most probably point of failure. These methods often converge to a final solution with a relatively small number of function evaluations.

The purpose of this chapter is to determine the most efficient analytical reliability method that incorporates input parameter statistics and provides the most accurate probability of failure for flexible pavement design. Once the probabilities of failure are evaluated, the accuracy of each reliability method is determined considering the Monte Carlo Simulation technique as a baseline index, best representing the probability of failure. A simulation process was chosen as a baseline because these processes ultimately

produce the statistical properties of a performance function essentially by brute force. The performance function is evaluated at many randomly generated simulation points, the results of which are used to represent the performance function's distribution. A large number of simulation points must be evaluated to obtain a true representation of the performance function. This necessitates the use of a surrogate model to evaluate probability of failure, necessary to determine the accuracy of the proposed reliability methods.

Investigation of analytical reliability methods with the M-E design procedure for a conventional flexible pavement structure is performed. Four reliability methods (MVFOSM, Rosenblueth, FORM, and AMV) are applied to these two distress models to determine the probability of failure of these components. Then, these components are considered as a system, and reliability analyses for the series system are performed.

VI.2 Distress Models for M-E Pavement Design

Investigation of these analytical reliability methods requires an understanding of the distress models utilized in flexible pavement design. Generally, distress models, or transfer functions, are used to calculate the expected number of load repetitions that will fail the pavement section. Many distress models have been introduced by various entities. The models all follow a general formula, but the difference is introduced in the constants. Transfer functions for fatigue cracking generally take the form:

$$N_f = f_{1*}(\varepsilon_t)^{f_2}(E_1)^{f_3} \quad (\text{VI.1})$$

where N_f is the number of load repetitions until failure by fatigue cracking, ε_t is the tensile strain at the bottom of the hot mix asphalt, E_1 is the asphalt concrete modulus of elasticity, and f_1, f_2 , and f_3 are empirically determined constants. Because the magnitude of f_2 is generally much larger than that of f_3 , the effect of the modulus of elasticity is negligible, and the expression becomes:

$$N_f = f_{1*}(\varepsilon_t)^{f_2} \quad (\text{VI.2})$$

For rutting, transfer functions typically take the form

$$N_r = f_{4*}(\varepsilon_v)^{-f_5} \quad (\text{VI.3})$$

where N_r is the number of load repetitions until failure by rutting, ε_v is the vertical compressive strain on the top of the subgrade layer, and f_4 and f_5 are empirically determined constants.

The equations utilized for this study incorporate the constants derived by the Illinois Department of Transportation (63) for the fatigue model and the Asphalt Institute (4) for the rutting model.

$$N_f = (5 * 10^{-6}) * (\epsilon_t)^{-3} \quad (\text{VI.4})$$

$$N_r = (1.365 * 10^{-9}) * (\epsilon_v)^{-4.4477} \quad (\text{VI.5})$$

The asphalt concrete tensile strain and the subgrade compressive strain equations used are calculated according to algorithms developed by Thompson and Elliott (64). The equations were determined through the use of ILLI-PAVE, a computer program developed in 1980. The computer program was utilized to run 168 pavement configurations and the resulting algorithms are as follows:

$$\log(\epsilon_t) = 2.9496 + 0.1289 * h_1 - \frac{0.1595}{h_1} * \log(h_2) - 0.0807 * h_1 * \log(E_1) - 0.0408 * \log(K_1) \quad (\text{VI.6})$$

$$\log(\epsilon_c) = 4.5040 - 0.0738 * h_1 - 0.0334 * h_2 - 0.3267 * \log(E_1) - 0.0231 * K_1 \quad (\text{VI.7})$$

Where h_1 represents the HMA thickness, h_2 is the base thickness, E_1 is the HMA modulus, and the K_1 is the breakpoint resilient modulus of the subgrade.

VI.3 Distributions of the Random Variables

The asphalt concrete tensile strain and subgrade compressive strain design equations incorporate four design variables and the statistical properties of these variables are required for the reliability analysis. The variables are represented by statistical means and standard deviations from various sources. TABLE VI.1 summarizes the values chosen for this investigation.

TABLE VI.1: Probability Distributions of Random Variables

| Random Variable | Probability Distribution | Mean | Standard Deviation |
|-----------------|--------------------------|-----------|--------------------|
| h1 | Normal Random Variable | 3.1 in. | 0.48 in. |
| h2 | Normal Random Variable | 12.5 in. | 1.25 in. |
| E1 | Normal Random Variable | 1,600 ksi | 100 ksi |
| K1 | Normal Random Variable | 7.21 ksi | 1 ksi |

The HMA thickness and subgrade properties are from results presented by Darter et al. (39). The resilient modulus for the subgrade is obtained from work by Rada and Witczak (62) and represents properties of a crushed stone granular material. The HMA modulus mean and standard deviation are both obtained from Shell Nomographs and equations from the Asphalt Institute as summarized by Huang (37). The value used for experiment here is applicable for a temperature of 70°F and a load frequency of 4 Hz.

The performance functions for fatigue cracking and rutting compare the calculated number of load repetitions to an assumed number of load repetitions. This assumed number of load repetitions is treated as a constant, however further analysis could be performed to incorporate the variance of this term as well. The assumed number of load repetitions per year is calculated according to Eq. VI.8 which represents the number of equivalent single-axle loads per year. (37)

$$N^{18} / Y = (ADT)_o * T * T_f * G * D * L * (365) \quad (VI.8)$$

Where ADT_o represents the average daily traffic, T is the percentage of trucks in the average daily traffic, T_f is the number of 18-kips single-axle load applications per truck, G is a growth factor, D is the directional distribution factor, L represents the lane distribution, and Y is the design period. The value for N^{18} used for analysis assumes an average daily traffic count of 2,000 vehicles, 15% of the traffic classified as truck traffic, 0.2 load applications per truck, a growth factor of 2, distribution factor of 0.5, and a lane distribution equal to 1. The resulting number of equivalent single-axle loads per year is 21,900.

VI.4 Numerical Results

The results of the reliability analyses for the fatigue cracking distress and rutting distress components, and the system reliability results for the M-E transfer functions are presented in TABLE VI.2.

TABLE VI.2: Probability of Failure of Distress Modes and System according to various Reliability Methods

| Method | Probability of Failure | | Error | | Probability of Failure | Error |
|-------------|------------------------|---------|---------|---------|------------------------|--------|
| | Fatigue | Rutting | Fatigue | Rutting | System | System |
| FOSM | 0.2678 | 0.1307 | 4.46% | 10.26% | -- | NA |
| FORM I | 0.2304 | 0.0274 | 0.72% | 0.07% | 0.2314 | 3.66% |
| FORM II | 0.2304 | 0.0274 | 0.72% | 0.07% | 0.2314 | 3.66% |
| FORM III | 0.2304 | 0.0274 | 0.72% | 0.07% | 0.2314 | 3.66% |
| Rosenblueth | 0.2036 | 0.1158 | 1.96% | 8.77% | 0.4992 | 47.25% |
| AMV | 0.2294 | 0.0274 | 0.62% | 0.07% | 0.2314 | 3.66% |
| Monte Carlo | 0.2232 | 0.0280 | | | 0.2232 | |

The probability of failure of the components indicates relatively consistent results regardless of the reliability method applied. The error, calculated by the root sum of squares method, compares the results of each reliability method to the Monte Carlo Simulation method. The results indicate that the error for all methods is less than 11%, but more impressive, the FORM methods all produce an error less than 1%. Although the FOSM method is one of the simplest processes to implement, the deficiencies seem to be indicated by the decrease in accuracy. The importance of the minimum error produced by a single method should not be overshadowed by the consistency of that error. The Rosenblueth method produced an error of only 1.96% for the fatigue cracking component, but an error of 8.77% for the cracking component. Consistency of results,

along with a reasonably accurate result, indicates that the FORM or AMV methods are well suited for pavement design regarding fatigue cracking and subgrade rutting.

Component reliability is important, but the typical pavement design procedure calls for an understanding of the performance of a complete system. System reliability analysis was performed and is represented here as the combination of the Failure and Rutting components. The FOSM method is inaccurate in quantifying correlations among failure modes due to the method's invariance, and is therefore not included in the system based calculations. The analysis of the system, similar to the component analysis, produces results that also favor FORM and AMV as suitable reliability methods for pavement design. The Rosenblueth method produces a large error for the system analysis, but further, the method again seems to be inconsistent in comparison with the component Rosenblueth analysis, rendering it less appealing as a reliable method.

In addition to accuracy, it is of interest to investigate the effort required to perform these analytical probability methods compared to the MCS method. The original disregard of the MCS method as a method of performing reliability analysis was due, in strong part, to the computational effort required to perform the simulation. As previously discussed, the FORM I method is an iterative process, but the remainder of methods used are closed form. Therefore, computational effort is minor. TABLE VI.3 outlines a comparison in computation in terms of computational time and function count. Gradients were evaluated using finite differencing with $n + 1$ function evaluations required to obtain the final solution.

TABLE VI.3: Computation Effort of Reliability Methods Comparison

| Method | Function Count | | |
|-------------|----------------|---------|---------|
| | Fatigue | Rutting | System |
| FOSM | 5 | 5 | NA |
| FORM I | 50 | 30 | 1 |
| FORM II | 20 | 25 | 1 |
| FORM III | 35 | 45 | 1 |
| Rosenblueth | 5 | 5 | 5 |
| AMV | 10 | 15 | 1 |
| Monte Carlo | 100,000 | 100,000 | 100,000 |

TABLE VI.3 verifies the relatively cheap computational cost of all reliability methods, in comparison to simulation methods. As anticipated, the FOSM and Rosenblueth methods provide the cheapest computational effort. The FORM methods range in required power depending on the method implemented. AMV required very little computational effort. The FORM methods are more expensive due to the number of iterations required to perform the analysis. Although the function counts seem relatively reasonable from this experiment, an increase in the number of variables will significantly increase the computational effort required because gradient evaluations are more expensive and more iterations will be required to achieve convergence, though this is likely to be small for most problems in comparison to the effort required for MCS.

VI.5 Implementation with the AASHTO MEPDG

The numerical experiment for the simplified pavement performance transfer functions justifies the use of analytical reliability methods for M-E pavement design. Although these transfer functions have been well calibrated, the AASHTO MEPDG design equations incorporate more extensive climatic, material, and traffic data. Implementation of these analytical reliability methods is demonstrated and discussed.

VI.5.1 AASHTO MEPDG Prediction Equations

The AASHTO MEPDG is one of the most comprehensive M-E design methods available to pavement engineers. The MEPDG predicts pavement performance as a function of six major distress modes. Total and AC permanent deformation models describe the structural performance of the pavement structure as a function of stresses and strains. Two fatigue cracking models complete the structural assessment of pavement performance and the Terminal IRI metric describes the serviceability of the pavement.

The predictive model for Terminal IRI for new AC pavements over unbound aggregate bases is a function of: an initial IRI due to inconsistencies in initial construction, site factors, and fatigue and cracking distress quantities (Eq. VI.9); see also Introduction Section).

$$IRI = IRI_0 + 0.0463 \left[SF \left(e^{\frac{age}{20}} - 1 \right) \right] + 0.00119(TC_L)_T + 0.1834(COV_{RD}) + 0.00384(FC)_T + 0.00736(BC)_T + 0.00115(LC_{SNWP})_{MH} \quad (VI.9)$$

The MEPDG fatigue cracking models are based on Miner's Law and are a function of calibration factors and traffic loading (Eq.s VI.10 and VI.11).

$$FC_{bottom} = \left(\frac{6000}{1 + e^{(C_1 C'_1 + C_2 C'_2 \log_{10}(D * 100))}} \right) * \left(\frac{1}{60} \right) \quad (VI.10)$$

$$FC_{top} = \left(\frac{1000}{1 + e^{(7.0 + 3.5 \log_{10}(D * 100))}} \right) * (10.56) \quad (VI.11)$$

The MEPDG predicts permanent deformation for the AC layer and for the total pavement section as the sum of the product of plastic strains and the layer height (Eq. VI.12). The equation for plastic strain is calculated as shown in Eq. VI.13.

$$RD = \sum_{i=1}^{n_{sublayers}} \varepsilon_p^i h^i \quad (VI.12)$$

$$\frac{\varepsilon_p}{\varepsilon_r} = k_z \beta_{r1} 10^{k_1 T^{k_1} \beta_{r2} N^{k_3} \beta_{r3}} \quad (VI.13)$$

VI.5.1.a Reliability Analysis

The AMV method performed well with the simplified M-E design equations and is therefore selected as the reliability analysis method for investigation with the MEPDG design process. The accuracy of the AMV method is evaluated by comparing the AMV results to a Monte Carlo Simulation, where the MCS evaluation is considered the best representation of the actual reliability level.

While the AMV method is computationally efficient with the MEPDG, the computational expense of the MCS evaluation of the MEPDG must be reduced for appropriate comparison between methods. Even with the latest release of the MEPDG software, Darwin-M-E, which requires approximately 10 minutes for a single pavement evaluation, a MCS analysis for 1 million evaluations would require nearly 20 years to complete. To reduce the computational expense, the well-trained, accurate surrogate model constructed in Chapter IV is utilized for the MCS analysis, replacing the actual MEPDG software.

The AMV analysis is also evaluated with the surrogate model to appropriately compare the accuracy of the AMV method to the MCS analysis. The following analysis investigates the performance of the AMV method across the range of three standard deviations above and below the mean to demonstrate the accuracy of the AMV method across the majority of reliability levels. Eight design parameters were considered stochastic with the characteristics provided in TABLE VI.4.

Accurate prediction of reliability requires inclusion of all sources of uncertainty: input variability, surrogate model approximation errors, and model form error. An

additive model for uncertainty propagation is considered in this analysis, similar to that presented in Chapter V.3. It is assumed that the uncertainty in the terminal distresses is represented by the random variable $D_t(x)$ such that:

$$D_t(x) = \tilde{D}_t(x) + u_{MEPDG} + u_{surrogate} \quad (\text{VI.14})$$

where x is a vector of random design inputs, $\tilde{D}_t(x)$ the GP prediction of the distress, u_{MEPDG} is a random variable representing the MEPDG predictive uncertainty, and u_{GP} is a random variable representing the discrepancy between the MEPDG and GP predictions.

The contribution of error from the surrogate and MEPDG is the same for both the MCS and AMV reliability methods. Each MCS realization of $\tilde{D}_t(x)$ in the MCS analysis is generated by sampling the random design inputs. The AMV realizations of $\tilde{D}_t(x)$ are evaluated at design inputs that are determined as a function of a specified reliability index, β as shown in Equation VI.15.

$$\tilde{D}_t(x) = \tilde{D}_t(\mu_i^N - \alpha_i \beta \sigma_i^N) \quad (\text{VI.15})$$

The direction cosines (α_i) in Equation VI.15 are evaluated at the means of the input parameters.

Two CDF's can be constructed for each distress mode to understand the uncertainty introduced by each source of variability as:

1. Model input variability, MEPDG predictive uncertainty, and GP uncertainty, where:

$$D_t(x) = \tilde{D}_t(x) + u_{MEPDG} + u_{GP} \quad (\text{VI.16})$$

2. AMV prediction, MEPDG predictive uncertainty, and GP uncertainty, where:

$$D_t(x) = \tilde{D}_t(\mu_i^N - \alpha_i \beta \sigma_i^N) + u_{MEPDG} + u_{GP} \quad (\text{VI.17})$$

To construct the AMV CDF, evaluation of Equation VI.15 was performed for six reliability indices: -3, -2, -1, 0, 1, 2, and 3. The evaluation at these reliability levels allows for construction of a CDF three standard deviations above and below the mean which will well represent the true CDF.

TABLE VI.4: Probability Distributions of Random Variables

| Random Variable | Mean | Standard Deviation |
|----------------------|-------|--------------------|
| AADTT | 1500 | 150 |
| HMA Thickness (in) | 8 | 0.8 |
| GB Thickness (in) | 8 | 0.8 |
| EBC | 0.10 | 0.01 |
| AV | 0.08 | 0.008 |
| Esubgrade (psi) | 14500 | 1450 |
| Egb (psi) | 40000 | 4000 |
| A (binder viscosity) | 11.15 | 0.1115 |

Cumulative distribution functions for the AMV and MCS analyses are presented in FIGURE VI.1 . The CDFs demonstrate the impact of probability integration errors using this analytical reliability method. The total permanent deformation model and top down cracking models are significantly impacted by probability integration errors; however, AMV performed well for the remaining three predictive models. The lateral shift in the CDFs for the two poor performing functions is likely due to an error in the assumption that the mean of the function occurs at a reliability index of zero. A lateral shift of the AMV results significantly improves the predictions in both models. Additional discussion follows in Chapter VII.4.

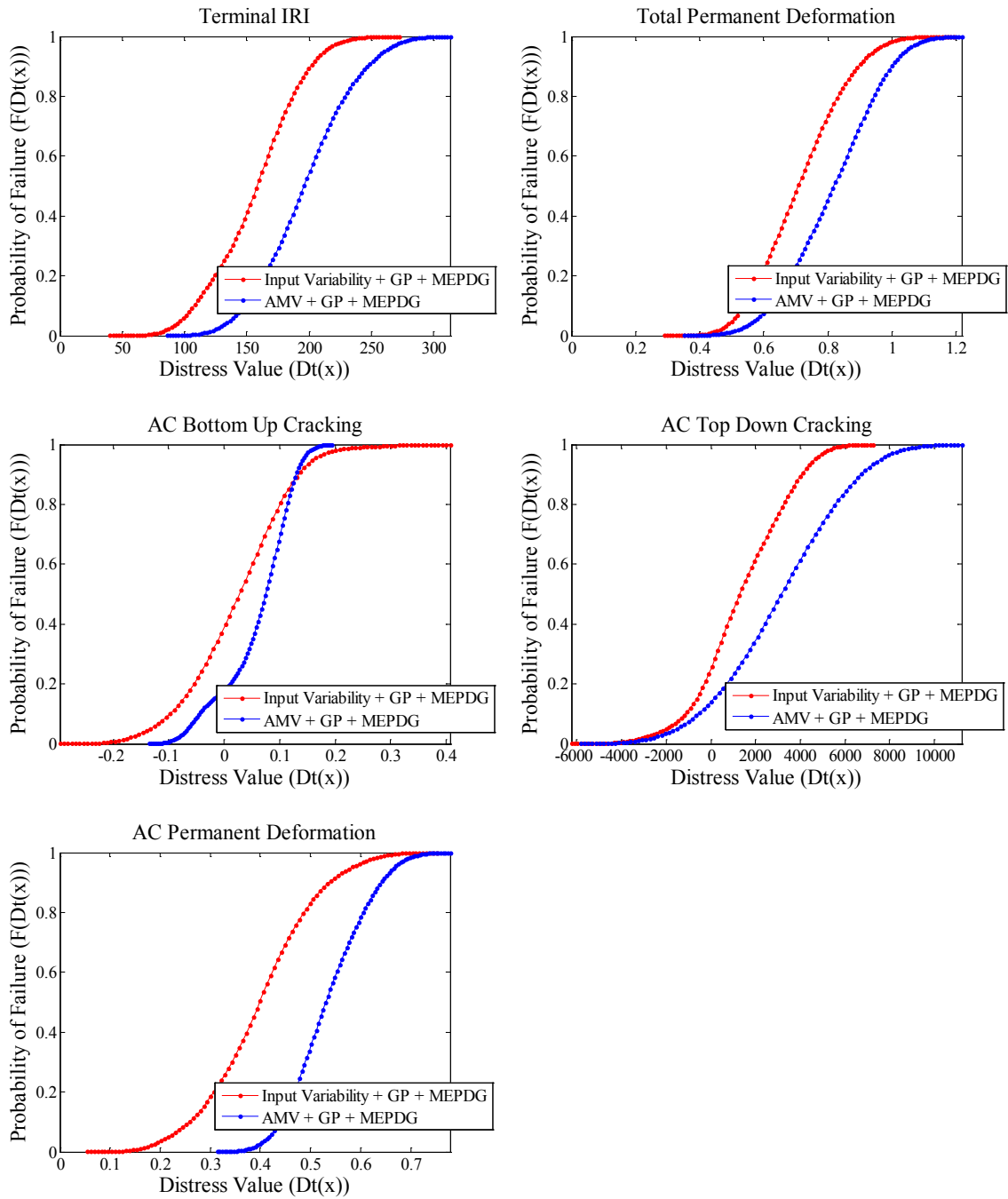


FIGURE VI.1: CDF Plots for AMV and MCS Results

VI.6 Conclusion

The application of reliability methods based on probabilistic uncertainty propagation is under-utilized in pavement design, but it can be of great benefit. Previous codes and design guides have depended heavily on empirical data over established reliability methods in an attempt to avoid perceived computational costs. However, in exchange for cheap computation, pavement designs have been historically over-designed, but more importantly, inconsistently designed. The application of mechanistic design procedures has increased efficiency of design, but even the most current pavement design procedures have forgone use of probabilistic methods due to their perceived high computational expense. This is not necessary as advances have been made in both the reliability methods and the computational power of design engineers.

Reliability methods such as those presented here provide reliability-based design that is capable of incorporating both the variability of the parameters of the pavement design and uncertainty due to model form error. These reliability methods have been shown to be efficient methods of design that require a minimal amount of computational time or cost. The FOSM and Rosenblueth methods both prove to be efficient methods that significantly sacrifice accuracy of results. The FOSM and Rosenblueth methods are also limited to normal and lognormal distributions for the random variables. These methods are also less accurate than FORM and AMV. FORM is a reasonably accurate method for evaluating the component and system reliability for M-E pavement design. However, the FORM method will tend to increase in computational effort as the number of variables increases. Further, convergence issues may arise. In particular, the FORM I method requests that the designer to perform numerous iterations to verify convergence

of the direction cosines and the reliability index. Under certain circumstances, the original input parameters, such as the initial reliability index, can cause oscillations, and the algorithm will not converge. AMV appears to be the best of all methods studied with regard to combined accuracy and efficiency.

Ultimately, the reliability methods presented here, in particular the AMV method, can be used efficiently to perform component and system reliability. The computational effort required for all these methods is reasonable and obtainable by a majority of design engineers. The reliability methods all provided reasonably accurate solutions and avoid the intensive computation time required to perform simulation techniques, such as MCS. Because these methods are based on the use of distributional information about the random variables, these methods can help quantify the benefits of quality control and management in the field and can help provide a rigorous justification for pay factors for contractors meeting quality control targets. The use of the most accurate probabilistic data as input for the design calculations will tend to produce solutions that accurately represent the construction conditions. Implementation of either the analytical or simulation based reliability analysis processes presented provide designers the ability to accurately and efficiently perform design optimization. As liability for pavement performance tends to lie towards that of the contractor on many state and federal road projects, contractors require appropriate tools for evaluation of pavement design over the desired life span. Consideration of construction costs over the life of the pavement allows for adequate financial preparation by both the governing body and the construction partner.

CHAPTER VII

LRFD AND CORRECTION FACTORS FOR ROUTINE RELIABILITY ANALYSIS WITH THE MEPDG

VII.1 Introduction

The previous chapter demonstrates the effectiveness of implementing analytical reliability methods. Because the majority of engineers designing flexible pavements do not have advanced training in structural reliability theory and/or probabilistic methods of engineering analysis, a simplified approach to design is necessary.

This chapter develops methods for calculating load and resistance factors and parameter offsets to use in routine design. These factors and offsets will allow designers to make reasonably conservative assumptions for values of the design inputs. The proposed methodology includes four primary steps. First, training data must be collected through use of a design of experiments and evaluation of the MEPDG. Next, Gaussian Process surrogate models are estimated to emulate the response of the MEPDG. The GP surrogate models must then be verified to assure that they accurately replicate the predictions of the MEPDG. Finally, design offsets and load and resistance factors are calculated through the use of first-order reliability methods, particularly inverse FORM.

This chapter also develops correction factors for the analytical reliability analysis method presented in Chapter VI.5, required for accurate reliability analysis in routine use. These correction factors improve accuracy in reliability predictions by reducing the bias

due to numerical integration errors when applying the AMV method with the MEPDG design procedure.

VII.2 Inverse FORM

Inverse FORM has been chosen as the method for calculation of load and resistance factors for the MEPDG because of some key features. An important by-product of FORM utilized in the inverse FORM method is the vector of probabilistic sensitivities, defined in step 3 (previously discussed in Chapter II.4).

$$\boldsymbol{\alpha}_i = -\frac{\nabla_u G(\mathbf{u})_i}{\|\nabla_u G(\mathbf{u})\|} \quad (\text{VII.1})$$

$$\text{where,} \quad \frac{\partial G(\mathbf{u})}{\partial u_i} = \frac{\partial g(\mathbf{x})}{\partial x_i} \sigma_{x_i} \quad (\text{VII.2})$$

The alpha vector is the negative normalized gradient row vector of the limit state function in the transformed space. Also, at optimality in the FORM problem, it is important to note that the alpha vector is collinear with the MPP vector. The alpha vector can help analysts determine which uncertain parameters are the most important so that information gathering efforts are focused on these variables. Random variables with alpha values of low magnitude can often be modeled as deterministic at the mean.

In this chapter, it is desired to design a structural system to perform for a single, worst case point that will guarantee that a specified level of reliability is attained. For such problems, the inverse FORM approach is often used to determine this point for design synthesis purposes. An example of this approach, to include an existence proof for reliably optimal solutions, can be found in (65). With inverse FORM, a trial design is proposed, and then the following problem is solved to determine design checking points:

$$\mathbf{u}^* = \operatorname{argmin}(G(\mathbf{u}) \mid \|\mathbf{u}\| = \beta^t) \quad (\text{VII.3})$$

FIGURE VII.1 depicts the inverse FORM approach as a diagram drawn in the standard uncorrelated normal space \mathbf{u} . Contours of constant probability density are shown as concentric circles. In higher dimensions, they are concentric hyperspheres. In the inverse FORM problem, an optimizer searches for the point that minimizes the value of $G(\mathbf{u})$ over a sphere with radius β^t . At this point, the limit state contour and the sphere (with radius equal to β) share a common tangent. The vector $\boldsymbol{\alpha}$ is collinear with the MPP \mathbf{u}^* . Therefore, the relationships $\mathbf{u}^* = \boldsymbol{\alpha}^* \beta$ and $\beta = \boldsymbol{\alpha}^* \cdot \mathbf{u}^*$ hold at MPP. Design offsets are determined from the vector \mathbf{u}^* . The vector \mathbf{u}^* for an inverse FORM problem is interpreted to be the number of standard deviations above or below the mean value of a random variable at which the design must be checked in order to assure a reliability level of $\Phi(\beta^t)$. The \mathbf{u}^* vector provides progressively more conservative values for design checking points as β^t increases, as illustrated in FIGURE VII.1.

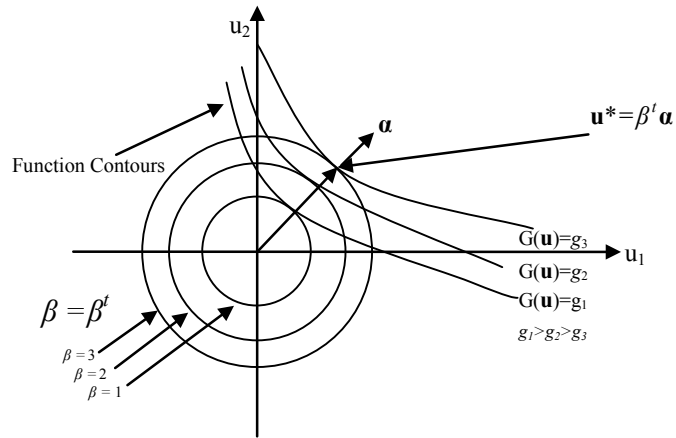


FIGURE VII.1: Geometric Interpretation of Inverse FORM

VII.3 Calculation of Load and Resistance Factors

Calculation of load and resistance factors improves routine evaluation of predicted reliability for pavement design. Though application of reliability methods is straightforward for engineers with advanced training in probability and statistical methods, relatively few in the highway community have the advanced training required to implement these methods in design. Additionally, it is not always desirable to incur the expense of using these methods every time a routine pavement section is designed. One important benefit of the use of analytical reliability methods in the context of the MEPDG is that these methods can provide a rigorous and justifiable basis for finding design values for the random variables that can allow design using the MEPDG without the end user having to be experienced in reliability methods.

The objective in this chapter is to offset the random variables from their means so that design is done on the basis of only one point at which the pavement is designed and computational efforts in reliability analysis are minimized. Equivalently, the designer can

determine load or resistance factors by which the mean of an uncertain variable can be multiplied in order to determine a design variable. The latter is the uncertainty management approach taken in the AISC Steel Design Manual (66) and in the ACI 318 Concrete Design Manual, (67) where the loads and resistances are factored and the structural element is designed to be safe given the factored loads and resistances. Design values for the variables can be calculated by solving the inverse FORM problem to find \mathbf{u}^* , the offset for the random variable in terms of standard deviations. Once the \mathbf{u}^* point is determined, it is then transformed to the \mathbf{x} space to find the design values of the random variables.

VII.3.1 Distributions of the Random Variables

In the calculation of the load and resistance factors and design offsets, the random variable probability distributions are assumed to be normal and independent with means and standard deviations as shown in TABLE VII.1.

TABLE VII.1: Probability Distributions of Random Variables

| Random Variable | Mean | Standard Deviation |
|----------------------|-------|--------------------|
| AADTT | 1500 | 150 |
| HMA Thickness (in) | 8 | 0.8 |
| GB Thickness (in) | 8 | 0.8 |
| EBC | 0.10 | 0.01 |
| AV | 0.08 | 0.008 |
| Esubgrade (psi) | 14500 | 1450 |
| Egb (psi) | 40000 | 4000 |
| A (binder viscosity) | 11.15 | 0.1115 |

VII.3.2 GP Model Construction and Verification

A GP model is utilized in the derivation of load and resistance factors. The GP model for this chapter was constructed with the mathematical platform MATLAB, with a Kriging toolbox (68) and model verification was performed. Training data consists of 110 evaluations of the MEPDG performed at values of the first seven random variables in TABLE VII.1, selected randomly from intervals bounded by the means of the random variables plus or minus three standard deviations. The values for A were selected randomly from the finite set of default AC binder grades from the MEPDG level 3 data. The MEPDG was evaluated for a new flexible pavement section with the input parameters based primarily on those found in the “New-HMA.dgp” file available through

the MEPDG software (6). The design utilized climatic data for Nashville, TN and a desired design life of 20 years.

Model verification was performed on the surrogate model to verify accuracy with respect to the true MEPDG output. The purpose of model verification herein is not to investigate potential verification metrics, but to show that the surrogate model constructed is acceptable for use in this specific application. For that purpose, verification of the surrogate will involve verification of the predictive capability of the GP model at points within the domain from which the training points have been selected.

Two statistical parameters for model verification, predictive coefficient of determination (predictive R-square) and Bayes factor, were performed for five major distress modes. These metrics were calculated by selecting ten points randomly from the total quantity of training points and designating them as verification points. A surrogate model was constructed utilizing the remaining training points and the model was evaluated at the verification points. The output from the surrogate model at those verification points is compared to the true MEPDG output values as verification of the GP accuracy. To avoid potential bias due to the selection of the training points, the entire verification process was repeated 10,000 times. The mean and variance values for the predictive R-square are reported in TABLE VII.2. The mean of the R-square values for each distress mode is high (with two near unity), indicating that the GP prediction is closely correlated with the MEPDG. This statistical method of model verification confirms that the GP is suitable for accurately representing the MEPDG.

TABLE VII.2: Verification of Predictive Capability of GP Models

| Distress Mode | Predictive Coefficient of Determination (R^2) | | |
|-----------------------------|---|------------|--------|
| | Mean | Variance | COV |
| Terminal IRI | 0.7442 | 0.0665 | 0.3465 |
| AC Surface Down Cracking | 0.8886 | 0.0225 | 0.1690 |
| AC Bottom Up Cracking | 0.8200 | 0.0321 | 0.2183 |
| AC Permanent Deformation | 0.9881 | 6.7893e-05 | 0.0083 |
| Total Permanent Deformation | 0.9871 | 0.0002 | 0.0156 |

The 10,000 samples were also used to calculate Bayes Factors as a second form of model verification. The probability that the Bayes Factor is less than a specific “threshold” value corresponding to the level of support for the model was calculated for each distress mode and the results are shown in TABLE VII.3. The results indicate that the selection of the verification points will significantly impact the level of support for the model. The surrogate model is strongly supported when the Bayes Factor exceeds 100. All models were considered strongly supported for at least half of the 10,000 samples. When all 110 training data points are used, the model is certainly valid.

TABLE VII.3: Distribution of Bayes Factors for GP Models

| Distress Mode | Probability that Bayes Factor is Less than: | | | |
|-----------------------------|---|--------|--------|--------|
| | 3 | 10 | 30 | 100 |
| Terminal IRI | 25.2% | 32.32% | 39.27% | 46.57% |
| AC Surface Down Cracking | 12.84% | 17.28% | 22.64% | 29.50% |
| AC Bottom Up Cracking | 21.29% | 27.31% | 33.80% | 42.22% |
| AC Permanent Deformation | 0.02% | 0.02% | 0.04% | 0.04% |
| Total Permanent Deformation | 0.00% | 0.01% | 0.02% | 0.05% |

VII.3.3 Calculation and Discussion of Load and Resistance Factors

Inverse FORM was performed for the five distress modes commonly encountered in Tennessee and the computed load and resistance factors, as well as the design offsets, are shown in TABLE VII.4 through TABLE VII.8. In order to compute design values to use for analysis with MEPDG, design engineers can use either of the following two equivalent equations:

$$x_{design} = \mu_x + k\sigma_x \quad (VII.4)$$

$$x_{design} = \phi\mu_x \quad (VII.5)$$

TABLE VII.4: Load and Resistance Factors and Parameter Offset Values for Terminal IRI Distress Model

| Variable | 80% Reliability ($\beta=0.85$) | | 90% Reliability ($\beta=1.3$) | | 97.5% Reliability ($\beta=2$) | |
|---------------|-------------------------------------|--------|---------------------------------|--------|---------------------------------|--------|
| | k | ϕ | k | ϕ | k | ϕ |
| AADTT | 0.0294 | 1.0029 | 0.0449 | 1.0045 | 0.0691 | 1.0069 |
| HMA Thickness | -0.0135 | 0.9987 | -0.0206 | 0.9979 | -0.0317 | 0.9968 |
| GB Thickness | -0.0036 | 0.9996 | -0.0055 | 0.9995 | -0.0084 | 0.9992 |
| EBC | 0.0050 | 1.0005 | 0.0077 | 1.0008 | 0.0119 | 1.0012 |
| AV | 0.0146 | 1.0015 | 0.0223 | 1.0022 | 0.0343 | 1.0034 |
| Esubgrade | 0.0457 | 1.0046 | 0.0699 | 1.0070 | 0.1075 | 1.0107 |
| Egb | 0.0213 | 1.0021 | 0.0326 | 1.0033 | 0.0502 | 1.0050 |
| A | -0.8477 | 0.9915 | -1.2965 | 0.9870 | -1.9947 | 0.9801 |

TABLE VII.5: Load and Resistance Factors and Parameter Offset Values for AC Surface-Down Model

| Variable | 80% Reliability ($\beta=0.85$) | | 90% Reliability ($\beta=1.3$) | | 97.5% Reliability ($\beta=2$) | |
|---------------|-------------------------------------|--------|---------------------------------|--------|---------------------------------|--------|
| | k | ϕ | k | ϕ | k | ϕ |
| AADTT | 0.0926 | 1.0093 | 0.1417 | 1.0142 | 0.2180 | 1.0218 |
| HMA Thickness | -0.7976 | 0.9202 | -1.2198 | 0.8780 | -1.8767 | 0.8123 |
| GB Thickness | -0.1888 | 0.9811 | -0.2887 | 0.9711 | -0.4442 | 0.9556 |
| EBC | -0.0678 | 0.9932 | -0.1036 | 0.9896 | -0.1594 | 0.9841 |
| AV | 0.1904 | 1.0190 | 0.2912 | 1.0291 | 0.4480 | 1.0448 |
| Esubgrade | 0.0246 | 1.0025 | 0.0376 | 1.0038 | 0.0579 | 1.0058 |
| Egb | 0.0080 | 1.0008 | 0.0122 | 1.0012 | 0.0188 | 1.0019 |
| A | -0.0251 | 0.9997 | -0.0384 | 0.9996 | -0.0591 | 0.9994 |

TABLE VII.6: Load and Resistance Factors and Parameter Offset Values for AC Bottom-Up Distress Model

| Variable | 80% Reliability ($\beta=0.85$) | | 90% Reliability ($\beta=1.3$) | | 97.5% Reliability ($\beta=2$) | |
|---------------|----------------------------------|--------|---------------------------------|--------|---------------------------------|--------|
| | k | ϕ | k | ϕ | k | ϕ |
| AADTT | 0.0928 | 1.0093 | 0.1419 | 1.0142 | 0.2183 | 1.0218 |
| HMA Thickness | -0.4450 | 0.9555 | -0.6805 | 0.9319 | -1.0470 | 0.8953 |
| GB Thickness | -0.0980 | 0.9902 | -0.1499 | 0.9850 | -0.2306 | 0.9769 |
| EBC | -0.0139 | 0.9986 | -0.0212 | 0.9979 | -0.0327 | 0.9967 |
| AV | 0.7097 | 1.0710 | 1.0855 | 1.1085 | 1.6699 | 1.1670 |
| Esubgrade | 0.0246 | 1.0025 | 0.0376 | 1.0038 | 0.0579 | 1.0058 |
| Egb | 0.0080 | 1.0008 | 0.0122 | 1.0012 | 0.0188 | 1.0019 |
| A | -0.0415 | 0.9996 | -0.0634 | 0.9994 | -0.0975 | 0.9990 |

TABLE VII.7: Load and Resistance Factors and Parameter Offset Values for AC Permanent Deformation Distress Model

| Variable | 80% Reliability ($\beta=0.85$) | | 90% Reliability ($\beta=1.3$) | | 97.5% Reliability ($\beta=2$) | |
|---------------|----------------------------------|--------|---------------------------------|--------|---------------------------------|--------|
| | k | ϕ | k | ϕ | k | ϕ |
| AADTT | 0.0929 | 1.0093 | 0.1421 | 1.0142 | 0.2186 | 1.0219 |
| HMA Thickness | 0.5718 | 1.0572 | 0.8745 | 1.0874 | 1.3453 | 1.1345 |
| GB Thickness | 0.2874 | 1.0287 | 0.4395 | 1.0440 | 0.6762 | 1.0676 |
| EBC | 0.1780 | 1.0178 | 0.2723 | 1.0272 | 0.4189 | 1.0419 |
| AV | 0.2975 | 1.0298 | 0.4551 | 1.0455 | 0.7001 | 1.0700 |
| Esubgrade | 0.0246 | 1.0025 | 0.0376 | 1.0038 | 0.0579 | 1.0058 |
| Egb | 0.0080 | 1.0008 | 0.0122 | 1.0012 | 0.0188 | 1.0019 |
| A | 0.4283 | 1.0043 | 0.6551 | 1.0066 | 1.0079 | 1.0101 |

TABLE VII.8: Load and Resistance Factors and Parameter Offset Values for Total Permanent Deformation Distress Model

| Variable | 80% Reliability ($\beta=0.85$) | | 90% Reliability ($\beta=1.3$) | | 97.5% Reliability ($\beta=2$) | |
|---------------|-------------------------------------|--------|---------------------------------|--------|---------------------------------|--------|
| | k | ϕ | k | ϕ | k | ϕ |
| AADTT | 0.0929 | 1.0093 | 0.1422 | 1.0142 | 0.2187 | 1.0219 |
| HMA Thickness | 0.6102 | 1.0610 | 0.9333 | 1.0933 | 1.4358 | 1.1436 |
| GB Thickness | 0.3134 | 1.0313 | 0.4793 | 1.0479 | 0.7373 | 1.0737 |
| EBC | 0.1960 | 1.0196 | 0.2998 | 1.0300 | 0.4612 | 1.0461 |
| AV | 0.3275 | 1.0328 | 0.5009 | 1.0501 | 0.7706 | 1.0771 |
| Esubgrade | 0.0246 | 1.0025 | 0.0376 | 1.0038 | 0.0579 | 1.0058 |
| Egb | 0.0080 | 1.0008 | 0.0122 | 1.0012 | 0.0188 | 1.0019 |
| A | 0.3114 | 1.0031 | 0.4762 | 1.0048 | 0.7326 | 1.0073 |

Intuitive results for load and resistance factors were observed. Notice that the design offsets k become larger in magnitude with higher levels of reliability. IRI predictions were found to be highly sensitive to the binder stiffness, and pavements with harder binders were more susceptible to developing roughness. Therefore the recommended design offsets require that the MEPDG analysis be performed with the A parameter set to a value approximately two standard deviations below the mean if 97.5% reliability is required. The uncertainty in the asphalt concrete surface down cracking failure mode was dominated by the asphalt concrete layer thicknesses. Analysis with the MEPDG should be undertaken with a layer thickness of approximately 92 percent of the nominal (mean) layer thickness if 97.5% reliability is required. Air voids were also found to be detrimental to cracking, and the analysis should be performed with an elevated value of

air voids. Similar results were observed for bottom up fatigue cracking. Thicker layers were found to be more susceptible to permanent deformation. This is likely because the MEPDG accumulates plastic strain through the entire depth of the pavement section. For the permanent deformation limit states, soft binders and higher percentages of air voids tend to make the pavement more susceptible to deformation. For instance, to design against permanent deformation in the asphalt concrete layer, the layer thickness should be offset by 1.34 standard deviations above the mean, the A parameter should be offset by 1.0079 standard deviations above the mean, and the air voids should be offset by 0.7001 standard deviations above the mean if 97.5% reliability is required.

VII.4 Correction Factors for AMV

A process similar to the LRFD procedure previously discussed can be implemented to correct for errors in the reliability predictions with the MEPDG AMV reliability procedure. The results presented in Chapter VI.5 indicate probability integration errors and a bias due to approximation errors in the AMV procedure. One method for correction of these errors in practical application includes a lateral shift and a correction factor multiplier.

The bias in the AMV predictions compared to the MCS evaluations is attributed to an incorrect assumption that the function evaluated at the means of the input parameters has a reliability index equal to zero. Predicted reliability is improved by shifting the 50% reliability prediction from the AMV procedure to the value of the MCS evaluation at 50% reliability. This lateral shift corrects for the error in the assumption of

the value of the reliability index at the mean. The AMV procedure can be repeated at this new “checking point” to create a new CDF centered on the MCS CDF.

In addition to correcting for bias in the AMV predictions, a correction factor is necessary to correct for probability integration errors. A multiplicative factor can be applied to the AMV predictions to correct for this type of error. These correction factors must be derived for specific reliability levels as the integration error varies at differing probabilities of failure.

VII.5 Conclusion

This chapter has shown the feasibility of implementing an approach to the management of uncertainty, similar to that used in LRFD structural design codes, specifically outlined for pavement engineering. A method for deriving load and resistance factors and design parameter offsets for the MEPDG inputs has been developed for the purpose of assuring, to a high level of probability, that the MEPDG predicted distress at any level of model confidence does not exceed a given threshold. One salient feature of this approach is that the two most significant sources of uncertainty in pavement design, input variability and model prediction error, are handled separately. The proposed methodology involves four major steps: (1) experimental design, (2) surrogate model estimation, (3) model verification, and (4) calculation of load and resistance factors and design offsets through the inverse first order reliability method.

Though the results were intuitive, the contribution of this chapter is the quantification of parameter offsets for routine evaluation of a typical flexible pavement

design. Since high computational demand of flexible MEPDG makes the use of Monte Carlo impractical, the proposed technique may prove useful. Utilization of these load and resistance factors, or parameter offsets, in the context of an analytical reliability analysis provides an alternative to a surrogate with a simulation-based process, either of which is necessary in performing risk-based design for flexible pavements utilizing M-E design procedures. Liability for performance of pavements over the entire design life is increasingly shifting to the agency required to construct and maintain the pavement. Responsibility for initial construction and reconstruction costs necessitates accurate design optimization routines.

CHAPTER VIII

RISK-BASED DESIGN OPTIMIZATION METHOD UTILIZING M-E DESIGN EQUATIONS

VIII.1 Introduction

Optimization of flexible pavement design incorporating uncertainty is dependent on reliable pavement design and construction. Accuracy in the predictive models is necessary in achieving target reliability levels for the performance of the pavement over a specified design life. A practical application where accurate design optimization is necessary is warranty-based construction in which contractors are required to perform initial construction and provide maintenance over a specified design life for the pavement. Risk-based design optimization, incorporating all sources of uncertainty, is critical to bidding and budgeting for pavements designed for these contracts. Decision-making tools that include the cost as a decision variable aid in the design process for contractors who seek to design pavements systems that maximize profit and are reliable over the life of the warranty.

The current implementation of the MEPDG provides a *descriptive* design process that can be utilized to define, or describe, a pavement design that meets a specific threshold reliability level within a specified design life, but it is computationally ineffective for use in design optimization problems. Uncertainty propagation for reliability and sensitivity analyses becomes computationally efficient utilizing simulation based methods with an accurate surrogate model replacing the more expensive M-E

design equations. A well trained surrogate model is a powerful decision-making tool in terms of accuracy of predictions of performance over the design life of the pavement. Further, construction of a surrogate provides the designer a tool that can be utilized as a *prescriptive* design tool which can be utilized to consider the optimal solution including the design life as a decision variable. Construction of a surrogate model requires a design of experiments that determines the quantity of training points required to accurately mimic the performance function. Many sampling techniques exist, typically choosing to either investigate the entire design space or explore a specific target region of the design space. For the flexible pavement design problem, a method that incorporates each of these concepts will provide a model that is well trained across the domain space, but refined in a region of interest related to overall construction cost.

To develop a framework for risk-based design optimization for flexible pavements, it is necessary to construct a computationally efficient surrogate model that emulates the MEPDG pavement prediction models. Estimation with a surrogate model requires selection of training data and construction. Model verification of the surrogate is also required. This Chapter presents a selection process for determining the quantity of training points (N_{TP}) by incorporating an adaptive sampling technique. This method simultaneously builds the surrogate model and provides an optimization tool for designers; reducing the overall computational expense required for optimization of design.

VIII.2 Selection of a Surrogate Model Type

There exist a number of surrogate modeling approaches, each with advantages and disadvantages relating to their success to produce an accurate predictive model given certain properties regarding the data. The Gaussian Process (GP) surrogate model has been chosen for the MEPDG design software for several reasons. First, GP models are shown to be capable of fitting data for high dimensional problems, on the order of 30-50 input parameters, which is appropriate for the MEPDG's large number of significant input parameters. Second, the GP model is an interpolation method that does not follow a specific functional form. GP models are suitable for approximating any smooth, continuous function.

Construction of a GP surrogate model requires selection of a correlation function and a mean function. The squared-exponential form has been selected as the correlation function. This form utilizes the following equation:

$$c(x^j, x^k) = \exp\left[-\sum_{i=1}^n \xi_i (x_i^j - x_i^k)^2\right] \quad (\text{VIII.1})$$

Where ξ_i is a scale factor that must be estimated, x_i^j represents the j^{th} training point at the i^{th} dimension, and x_i^k represents the new prediction point at the i^{th} dimension. The terms are summed over the number of training points, n . The correlation function is utilized to construct a correlation matrix, R :

$$R = \begin{bmatrix} c(x_1^1, x_1^k) & \cdots & c(x_1^j, x_n^k) \\ \vdots & \ddots & \vdots \\ c(x_n^1, x_1^k) & \cdots & c(x_n^j, x_n^k) \end{bmatrix} \quad (\text{VIII.2})$$

The covariance function, indicating the covariance between the observed MEPDG response values of the training data, $Y(x^j)$, and the predicted responses, $Y(x^k)$, is represented as a function of the correlation matrix, R , and variance as shown here:

$$\text{Cov}(Y(x^j), Y(x^k)) = \sigma^2 R \quad (\text{VIII.3})$$

The variance term in Eq. VIII.3 is another parameter of the GP model that must be estimated. A mean function is also required for construction of the surrogate model. For this application, a constant function form is utilized:

$$\mu(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots \quad (\text{VIII.4})$$

The vector, β , is the final parameter that must be estimated to complete the construction process.

Once the model form has been selected, the model parameters (mean μ , variance σ^2 , and correlation length-scale factors ξ) must be estimated. The process of parameter estimation is commonly performed utilizing a maximum likelihood estimation method. The procedure takes the form of an optimization problem. To avoid common complications due to ill-conditioned matrices, the optimization problem is modified to a minimization of the negative log-likelihood function, $-\log[L(\cdot)]$, of the form:

$$\text{Minimize}_{(\beta, \sigma^2, \xi)} -\log[L(\cdot)] = n * \log(\sigma^2) + \log|R| + \frac{(Y - \mu)^T R^{-1} (Y - \mu)}{\sigma^2} \quad (\text{VIII.5})$$

VIII.3 Surrogate Model Construction: Adaptive Training Point Selection

Process

To efficiently construct a surrogate model that accurately predicts pavement performance across the entire domain space, an adaptive selection technique is presented. The location of training points for the surrogate model is determined through an optimization routine, combining both an exploration and exploitation optimization process. The exploration process improves the predictive accuracy of the GP across the entire domain space and guarantees that the GP is accurate within a specified tolerance across the space. The exploitation routine refines the GP model around a local optimum to provide greater accuracy in a specific area of interest.

VIII.3.1 Quantity of Training Points (N_{TP})

A Latin Hypercube (LHC) sampling routine is utilized to generate potential training points. The development of the potential points is computationally inexpensive. Training values (outputs) for the surrogate model are required only when selection of a training point has been made and are found utilizing the MEPDG design software, therefore selection of the training points does not require evaluation of the MEPDG functions.

The surrogate model is initialized with randomly chosen points, a sub-set selected from the full set of potential training points, and exploration and exploitation routines are performed (in parallel) until convergence criteria is reached for both methods. The training points not selected in the initialization routine are considered as candidate points which can become training points through the exploration and exploitation routines. This process of pre-selecting candidate points by the LHC sampling method is not required. New training points could be selected as any feasible solution in the domain space. The LHC process was utilized here to reduce the computational cost associated with the exploration routine.

VIII.3.1.a Exploration Routine

The Exploration routine explores the design domain and selects additional training points that will most significantly improve the accuracy of the model predictions across the entire design space. Improvement in accuracy is defined in this routine as a reduction in the GP variance. This algorithm selects a new training point in a region of the domain

where the GP variance is a maximum, otherwise stated as the maximum distance from all other training points.

This exploration routine chooses the next potential training point based on the average GP variance for that candidate point across all distress models. The improvement of the GP through the exploration process is quantified by the variance of a set of verification points randomly chosen across the domain space. The verification points are not utilized as training points or candidate points, therefore maintaining consistency throughout the construction process. Additional training points, selected from a pre-defined candidate pool, are added to the surrogate model at each iteration of the exploration routine and are chosen as the points that minimize the average variance for the candidate points across all MEPDG distress modes.

VIII.3.1.b Exploitation Routine

The exploitation routine chooses additional training points for the surrogate model utilizing a construction cost function. This process provides model refinement in the region of the design space where a local minimum, and potentially a global minimum, exists.

The selection of the next training point for the surrogate can be performed with a cost function which includes an initial construction cost and an additive maintenance cost, similar to that shown in Equation VIII.6.

$$\text{Minimize} \quad \text{Cost}_{\text{PerLaneMile}} = \$20,000 * HMA_{\text{thick}} + \$7,500 * GB_{\text{thick}} + \$125,000 p_{f,Ave}$$

(VIII.6)

Equation VIII.6 defines the initial construction cost per lane mile or road as a function of two significant material properties: asphalt and granular base layer thicknesses. The maintenance cost is treated as a function of the average probability of failure across all distress modes.

VIII.3.1.c Stopping Criteria

The minimum required quantity of training points (N_{TP}) for the surrogate model is determined by the stopping criteria for the exploration and exploitation routines. The exploration routine stopping criteria is best defined when the addition of a new training point does not significantly improve the accuracy of the surrogate model across the domain. The selection routine from the pool of candidate points will not always reduce the average GP variance for the remaining candidate points. Although the point of greatest GP variance is removed from the candidate points, the mean of the GP variance is impacted by the change in quantity. Further, the GP model is retrained at each iteration, so the GP variance for each candidate point is likely to change based on the updated GP parameters. Therefore, improvement is defined as a significant reduction in GP variance for the set of verification points which remains constant through the construction process. The verification points will quantify the performance of the model across the domain, independent of the location and quantity of the training points.

The stopping criterion for the exploitation routine is dependent on the cost function utilized. The cost function utilized in this dissertation is a function of two design variables and the probability of failure for the design. Additional constraints could limit the feasible solutions and provide a stopping criterion for this routine. Stopping criteria could include a budgetary constraint, which for this formulation, would also require a minimum reliability level. The unconstrained problem in the exploitation routine does not restrict the probability of failure for a pavement, which may not be acceptable to some agencies. However, the increased use of warranty contracts for pavement construction can use this routine as a financial decision-making process.

For the analysis here, the exploitation routine is left unconstrained, allowing for a better investigation into the performance of the exploration routine and impact on accuracy in predictions by the GP. The constraints on the exploitation routine will always reduce the number of training points, as a function of feasible cost and performance requirements, which is an important aspect to the purpose of RBDO.

VIII.4 Verification of the Surrogate Model

In addition to the exploration and exploitation routines, model verification is required prior to use in risk-based design optimization applications. Model verification for this application is based on prediction testing. The values for the prediction points are calculated as the mean value of the distribution:

$$E[Y(x^k) | Y] = \mu + r^T R^{-1} (Y - \mu) \quad (\text{VIII.7})$$

Where r represents a vector of correlations as represented by:

$$r = \begin{bmatrix} c(x_1^j, x_1^k) \\ \vdots \\ c(x_n^j, x_n^k) \end{bmatrix} \quad (\text{VIII.8})$$

The predictions for the GP can be compared to the results from the MEPDG to determine the validity of the surrogate model at points other than the training data (i.e. verification points). One classic verification metric, the adjusted R-squared (R^2_{Adj}) value, can be computed, to compare the GP predictions to the actual MEPDG predictions. Values near one indicate that the GP is accurately emulating the MEPDG design functions.

VIII.5 Results

The process developed in the previous discussion is demonstrated for a numerical example. A surrogate model is constructed utilizing the exploration and exploitation routines to minimize the quantity of training points required to construct the GP while simultaneously searching for a feasible solution to a cost optimization problem. Verification of the GP is presented, followed by the solution to the RBDO example.

VIII.5.1 GP Construction and Verification

The exploitation and exploration routines implemented in the surrogate model construction demonstrate an effective method for minimizing the number of required training points while simultaneously solving the design optimization problem. Implementation of the construction process requires the selection of a stopping criterion to define the level of accuracy for the model. The GP variance at a set of verification points has been selected as the ‘statistic’ for stopping criteria. The variance of each individual verification point will approach a minimum value of zero with the addition of training points, but improvement of the GP variance will likely plateau at an optimal quantity of training points.

A thorough investigation of 1,000 training points was performed for a numerical example and the average GP variance for all verification points was calculated for each distress mode.

FIGURE VIII.1 demonstrates a convergence of the model as the number of training points approaches 1,000. The improvement with the additional training points demonstrates the effectiveness of the method, but also provides a tool to determine the minimum number of training points required to capture most of the behavior of the design functions. Visual inspection of all the models and an approximation for the system prediction indicates that improvement in the reduction of the GP variance converges to a near constant value. For the numerical example presented here, the visual inspection is performed after evaluating all 1,000 training points, but this is not required in practical applications. Convergence criteria can be defined when a minimum change in improvement is achieved. It is recommended that this criterion be met considering

improvement over a range of training points, rather than a single-step memory system to avoid local minima.

Although the visual inspection of the GP variance plots implies a potential minimum number of training points, verification of the GP compared to the actual function is also necessary. For application herein, the GP model is considered to accurately emulate the MEPDG when the minimum of the R^2_{Adj} values for a verification dataset compared to true MEPDG data is greater than 0.8. Construction with 500 training points, chosen by the GP variance plots, was shown to be acceptably accurate. The GP and MEPDG are well correlated with R^2_{Adj} values presented in TABLE VIII.1. All models achieve the specified minimum value for the R^2_{Adj} statistic. Additional improvements could be made to the GP by choosing additional training points in the same systematic way until the verification for each model reaches the approved minimum value.

TABLE VIII.1: GP Verification Results

| Distress Mode | Adjusted R ² |
|------------------------------------|-------------------------|
| Terminal IRI (in./mi.) | 0.837 |
| Total Permanent Deformation (in.) | 0.857 |
| AC Bottom Up Cracking | 0.821 |
| AC Surface Down Cracking (ft./mi.) | 0.884 |
| AC Permanent Deformation (in.) | 0.820 |

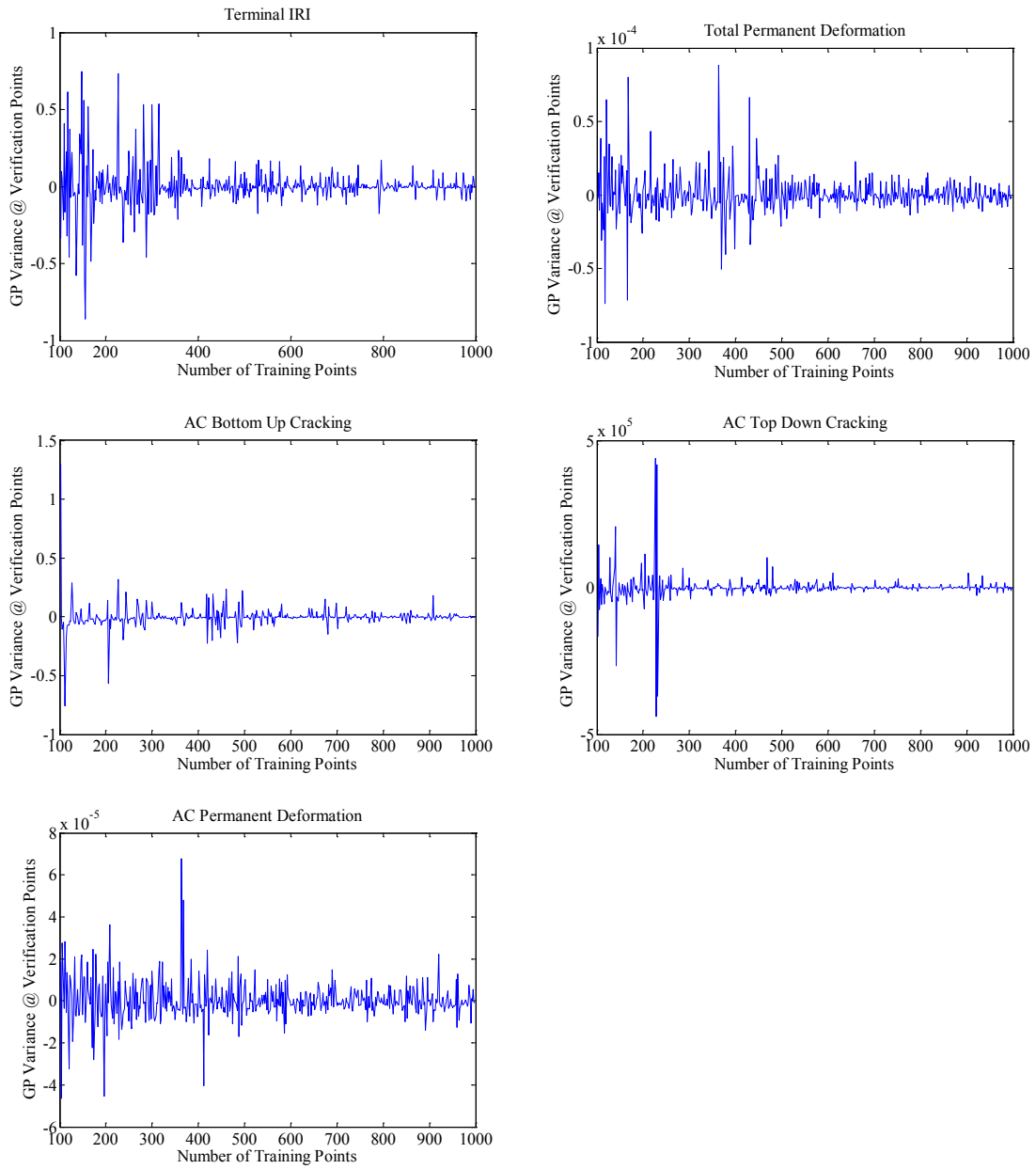


FIGURE VIII.1: Improvement in Average GP Variance for Verification Points

VIII.5.2 RBDO Solution

The optimal solution for the numerical experiment presented here occurs with mean design parameters presented in TABLE VIII.3. The means of the random variables utilized in the GP model are presented in TABLE VIII.2. Analysis incorporating uncertainty from the GP and MEPDG, and input parameter uncertainty results in a design that will cost approximately \$211,200. The optimal pavement meets a minimum reliability level of 70% which occurs in the AC permanent deformation distress mode (excluding the AC top down cracking model). (TABLE VIII.4) The AC top down cracking model is significantly impacted by model uncertainty and the solution presented in this numerical experiment results in a very high probability of failure for this distress model. It is assumed that use of an improved model would increase the reliability level for this distress mode without significantly impacting the optimal solution.

From the results of this numerical example, it is clear that the reliability level achieved is strongly influenced by the reconstruction cost term in the objective function. For appropriate life-cycle cost assessments, design engineers should perform cost optimization over a design life to determine the cost over the entire life of the pavement, incorporating yearly maintenance budgets as a function of the probability that the pavement does not meet a specified threshold value. The MEPDG is a powerful tool, but the current design process described in the Design Guide merely defines the performance of a pavement design as a function of a deterministic design life and target reliability. Implementation of the GP construction and optimization process presented here improves the MEPDG design process and provides a powerful infrastructure management tool. Design engineers implementing this procedure can make decisions for acceptable

reliability levels based on the cost over the life-time of the system which matches the practical maintenance process.

TABLE VIII.2: Random Variable Statistics for Design Optimization Problem

| Parameter Name | Mean | Standard Deviation |
|-----------------------------|-------|--------------------|
| AADTT | 1500 | 115.53 |
| Traffic Growth Rate | 4.0 | 0.58 |
| Percent Retained (#4) | 65.00 | 6.35 |
| Percent Passing (#200) | 3.10 | 1.79 |
| E _{subgrade} (psi) | 18000 | 3466 |

TABLE VIII.3: Design Optimization Results

| Design Parameter Name | Optimal Value |
|-----------------------|---------------|
| HMA Thickness (in.) | 6.03 |
| EBC (%) | 5.61 |
| AV (%) | 3.34 |
| GB Thickness (in.) | 7.11 |

TABLE VIII.4: Design Optimization Results: Distress Modes

| Distress Mode | Threshold Value | Reliability Achieved @ Threshold Value |
|------------------------------------|-----------------|--|
| Terminal IRI (in./mi.) | 275 | 79.13% |
| Total Permanent Deformation (in.) | 1.25 | 86.43% |
| AC Bottom Up Cracking | 25% | 87.3% |
| AC Surface Down Cracking (ft./mi.) | 2000 | 16.24% |
| AC Permanent Deformation (in.) | 0.75 | 69.58% |

VIII.6 Conclusion

Risk-based design optimization utilizing the framework presented here is necessary for accurate and reliable pavement design and construction. The construction of a surrogate model with the routines presented provides a computationally efficient method for evaluation of RBDO applications. The method presented can be adapted to consider alternative optimization problem formulations such as those described in Chapter II.5, the surrogate model can be trained utilizing additional parameters, and the method could be utilized with alternate M-E design procedures.

CHAPTER IX

CONCLUDING REMARKS

This dissertation has presented methods for management of uncertainty utilizing analytical and probabilistic methods in the context of M-E pavement design. A systematic and comprehensive approach to management of uncertainty in pavement design has been presented, incorporating uncertainty from input parameters, surrogate models, and the MEPDG prediction models.

A systematic and comprehensive approach to management of uncertainty by has been achieved by accomplishing four major objectives:

1. Address model uncertainty for the permanent deformation model
2. Develop a method to reduce computational expense.
3. Design a framework for incorporation of uncertainty in pavement design
4. Demonstrate a framework for risk-based M-E pavement design.

The methods presented demonstrate a comprehensive framework for performing accurate and reliable pavement performance predictions in a practical and computationally efficient way. Current M-E design procedures are computationally inefficient due to the inclusion of extensive quantities of design input parameters. Although these models are robust, surrogate modeling has been demonstrated to accurately emulate the M-E design equations while reducing computational expense. Surrogate modeling for the M-E

procedure increases computational speed allowing designers to perform highly iterative analyses that are otherwise too time-consuming in practice. Design optimization and reliability analysis can be performed in a fraction of the time, without significant loss of accuracy.

Quantification of model uncertainty for the permanent deformation predictive model has been presented and analysis performed to determine the most accurate model form. A predictive model that incorporates parameters that describe the pavements ability to resist shear and axial deformations has been demonstrated to improve accuracy in predictions over the more commonly utilized models that do not simultaneously consider these mechanistic behaviors. The weighted average models provide a computationally efficient means for developing predictive performance models without the computationally expensive evaluation of more advanced mechanistic concepts. Advanced theoretical developments are necessary, but require highly complicated non-linear evaluations of non-homogenous materials. Though these methods would improve theoretical knowledge, the M-E design equations presented are shown to achieve highly accurate predictive capability.

The construction and verification of a surrogate model accurately emulating the MEPDG flexible pavement design process was performed to reduce computational expense of current M-E design procedures. Specifically, a GP model was shown to accurately emulate M-E pavement design models and minimize computational expense. The GP model is a powerful tool that allows for additional investigation of the M-E prediction models. This model was exploited and sensitivity analyses were performed to determine the impact on predicted performance by Level 1 input parameters, quantity of

training data, and location of training points. A framework for selection of training points utilizing a correlation matrix between input parameters and predicted performance was shown to be an efficient method for selection of the quantity of training point parameters for the GP model.

A design framework for M-E flexible pavement design has been presented, incorporating all sources of uncertainty, to provide a design procedure that is accurate and computationally efficient. Reliability analysis is a critical design procedure impacted by the computational burden of M-E design procedures. Surrogate modeling improves computational speed allowing for robust reliability methods such as Monte Carlo Simulation. Analytical reliability methods have also been shown to provide accurate reliability estimates in a computationally efficient way. The GP model developed in this dissertation was shown to contribute only a minimal amount of uncertainty to predicted performance relative to MEPDG uncertainty and input parameter variability. Analytical reliability methods, specifically FORM and AMV were shown to be powerful reliability methods capable of accurate and computationally efficient evaluations. In addition, FORM and AMV provide a basis for development of LRFD factors and correction factors for routine reliability-based design optimization.

An exploration and exploitation GP construction process was demonstrated as an efficient algorithm for performing risk-based design optimization for flexible pavements. Design optimization is a critical design step that cannot be implemented efficiently in current M-E procedures. Contractors and design engineers need computationally efficient tools to perform design optimization within the constraints of rapid construction schedules and restricted budgets. To satisfy the final objective of the dissertation, it was

necessary to construct a surrogate model that emulates the MEPDG pavement prediction. Estimation with a surrogate model requires selection of training data and construction. Model verification of the surrogate is also required. This dissertation presented a selection process for determining the number of training points, N_{TP} , for construction of an accurate surrogate model by an adaptive sampling technique. The method simultaneously provides an optimization tool for designers reducing the overall computational expense required for optimization of design.

The methods presented here are critical to accuracy in predicted pavement performance, reliability analysis for flexible pavements, sensitivity analysis regarding design parameters and their significance to the design equations, and pavement design optimization.

CHAPTER X

FUTURE WORK

Although the accomplishments in this dissertation include a comprehensive method for management of uncertainty in flexible pavement design, research beyond the topics presented will further improve pavement design and analysis with M-E methods. Suggested research includes expansion of this work to alternative types of pavement structures such as rigid pavements and inverted pavements. Model form error should be quantified for M-E distress models in addition to the permanent deformation models investigated in this dissertation. Improvements to practical implementation of M-E design procedures, inclusion of additional empirical data, and additional verification and validation of continually evolving M-E models are all necessary for appropriate routine use by design engineers.

Four specific research topics related to the work presented in this dissertation are discussed in the following sections. These topics are not listed in any priority and additional research is not limited to the topics discussed.

X.1 Pay Factors and Performance Related Specifications

Further work related to this dissertation is necessary to develop a framework for implementation of these methods into computation of pay factors for contractors,

performance-related specifications for design agencies, and QC/QA guidelines to improve construction practices. Reliability analysis based on probabilistic or analytical methods as presented in this dissertation would provide significant benefits to the highway community. It would aid highway agencies by providing a basis for quantifying the benefits of quality control and quality assurance and providing a technically sound basis for computation of pay factors to be awarded to contractors for meeting certain quality control standards. The combination of these methods also enables the design engineer to account for uncertainty in the design parameters and to design pavements accordingly.

X.2 Genetic Algorithms for GP Parameter Selection Process

While the selection process methods for construction of the surrogate models performed well in Chapter IV of this dissertation, genetic algorithms may also be investigated as a selection process. The more robust optimization procedure may provide insight into the impact of the Level 1 input parameters on predicted pavement performance and may prove to be a more efficient method for selecting the quantity of parameters to accurately predict performance.

X.3 Additional MEPDG Distress Models & Various Pavement Structures

In addition to the permanent deformation model, investigation into all distress models is necessary to accurately evaluate reliability for pavement performance at a system level. The procedures presented in this dissertation can be applied to any of the distress models.

Rigid pavements can also be evaluated with the procedures demonstrated in this dissertation. Similar to flexible pavements, the MEPDG design procedure requires local calibration and is susceptible to model uncertainty.

New pavement types such as inverted pavement systems and flexible pavement using recycled asphalt pavement (RAP) require development of design equations that accurately predict performance. The process of model calibration presented in this dissertation may be applicable to RAP pavement design. The analytical reliability methods presented in this dissertation should be evaluated for inverted pavement systems.

X.4 Optimization Routine Improvement for Model Calibration

Future work is recommended to improve the optimization routine for deriving the calibrated performance models for the MEPDG. One possibility is investigation for, or development of, software that would be capable of automating the optimization routine with the MEPDG software. A second possibility is the use of a surrogate model. A well trained surrogate model, such as demonstrated by Retherford and McDonald (69), can accurately approximate the results of the MEPDG and can be implemented utilizing software capable of highly efficient optimization methods. For this application, the surrogate model must be trained including the calibration factors, in addition to all other significant design parameters. Future work also includes sensitivity analysis of the slope terms included in the parameter calibrated model to investigate the impact of the shear-based mix properties. A quadratic model could also be constructed and included in the sensitivity analysis to examine the impact of the interaction terms and higher order terms.

A sensitivity analysis of this kind would provide important information influencing mix design. Validation utilizing experimental data such as the results of test tracks at MnROADS and ALF would provide additional support for the approaches for permanent deformation prediction models presented.

Implementation of any of the aforementioned recommendations could improve accuracy in the prediction of permanent deformation performance in flexible pavement structures. Improved accuracy in predictions leads to optimal performance and reliable design life.

APPENDIX A: GP Training Data Generation

The training data utilized to construct the GP for the MEPDG was generated by Latin Hypercube Sampling. The sample Matlab code herein describes the method of generating this data.

```
clear all; clc;

%-----
----
%% Notes for Robust GP Procedure/m-files
%-----
----
% Run 'LatinHypercubeSamplingPlan' to obtain LHC training points for
most
% GP parameters;
%
% output = 'RobustGPInputs.mat'; matrix of 54 parameters
%
% Run 'TensileandCreep' to obtain remaining GP parameters; these are
the
% ave. indirect tensile strength and creep compliance calculations that
% MEPDG/Darwin-ME do not calculate for Level 1 or 2 Binder inputs
%
% output = 'RobustGPInputsFull.mat'; matrix of 76 parameters
%
% Run 'RGPDarwinFiles' to create xml files for use in Darwin-ME
%
% output = xml files located in 'Tempfiles' folder
%
% Import into Darwin-ME and evaluate;

%-----
----
%-----
----
%% Define Input Parameter Ranges
n = 1000; %Define number of LHC samples
p =45; %Define number of Input Parameters

LHCperms = lhsdesign(n,p, 'smooth', 'off');

%% Traffic Category

%-----
----
% Traffic
%-----
----
AADTT      = (400 * LHCperms(:,1)) + 1300; %Uniform Range [1300 1700]
LDF        = (10 * LHCperms(:,2)) + 80; %Uniform Range [0.8 0.9]
OpSpeed    = (10 * LHCperms(:,3)) + 60; %Uniform Range [60 70]
```

```

Traffic = [round(AADTT), round(LDF * 10) / 10, round(OpSpeed * 10) /
10];
%-----
----
% Traffic Volume
%-----
----
DistVehClass3ave = [0.9, 11.6, 3.6, 0.2, 6.7, 62, 4.8, 2.6, 1.4, 6.2];
DVC3min = 0.9 * DistVehClass3ave;
DVC3max = 1.1 * DistVehClass3ave;
m = 10; %Used for rounding for use in mepdg
for j = 1:n
    for i = 1:size(DistVehClass3ave,2)
        RandNumDVH(1,i) = LHCperms(j,4);
    end
    for i = 1:size(DistVehClass3ave,2)
        DVC3RandO(1,i) = ((DVC3max(1,i) - DVC3min(1,i)) *
RandNumDVH(1,i))...
            + DVC3min(1,i);
    end

    DVCCheck1(j,1) = sum(round(DVC3RandO * m)) / m;
    DVCresid(j,1) = 100 - DVCCheck1(j,1);

    %Adjust one value to force sum = 100; do not allow adjustment of
    %Class 9 vehicle -> maintain LHC for this vehicle class
    RandDVCadjust = round(random('uniform',1,10));
    while (RandDVCadjust > 6 && RandDVCadjust < 6)
        RandDVCadjust = round(random('uniform',1,10));
    end
    DVC3Rand = DVC3RandO;
    DVC3Rand(1,RandDVCadjust) = DVC3RandO(1,RandDVCadjust) +
DVCresid(j,1);

    DVCCheck1(j,1) = sum(round(DVC3Rand * m)) / m;
    DVCresid(j,1) = 100 - DVCCheck1(j,1);
    DVCmin(j,1) = min(DVC3Rand(1,:));

    while (DVCmin(j,1) < 0)
        DVCCheck1(j,1) = sum(round(DVC3RandO * m)) / m;
        DVCresid(j,1) = 100.0 - DVCCheck1(j,1);

        RandDVCadjust = round(random('uniform',1,10));
        while (RandDVCadjust > 6 && RandDVCadjust < 6)
            RandDVCadjust = round(random('uniform',1,10));
        end
        DVC3Rand = DVC3RandO;
        DVC3Rand(1,RandDVCadjust) = DVC3RandO(1,RandDVCadjust)...
            + DVCresid(j,1);

        DVCCheck1(j,1) = sum(round(DVC3Rand * m)) / m;
        DVCresid(j,1) = 100 - DVCCheck1(j,1);
        DVCmin(j,1) = min(DVC3Rand(1,:));
    end
end

```

```

    DVC(j,:) = DVC3Rand;
end
DVCmepdg = round(DVC * m) / m;
%Check that sum(DVCCheck1) = n * 100; Check should = 0;
%Check that minimum value is positive; Check should be >= 0;
%Check that rounded values for mepdg = 100; Check should = 100;
%Check that min and max for mepdg = same value; Check should = 0;
Check(1,1) = sum(DVCCheck1) - (n * 100);
Check(2,1) = min(min(DVC));
Check(3,1) = sum(sum(DVCmepdg,2))/n;
Check(4,1) = max(sum(DVCmepdg,2)) - min(sum(DVCmepdg,2)); Check

TrGrowth    = (2 * LHCperms(:,5)) + 3; %Uniform [3% 5%]

TrafficVol = [DVCmepdg, round(TrGrowth * 10)/10];
%-----
----
% General Traffic
%-----
----
MeanWheel    = (3.6 * LHCperms(:,6)) + 16.2; %Uniform [16.2 19.8]
WanderSD     = (2 * LHCperms(:,7)) + 9; %Uniform [9 11]
LaneWidth    = (2.2 * LHCperms(:,8)) + 10.8; %Uniform [10.8 13]
% TirePress  = (12 * LHCperms(:,18)) + 114; %Uniform [114 126]
AxleTand     = (10.4 * LHCperms(:,9)) + 46.4; %Uniform [46.4 56.8]
AxleTri      = (9.8 * LHCperms(:,10)) + 44.3; %Uniform [44.3 54.1]
AxleQuad     = (9.8 * LHCperms(:,11)) + 44.3; %Uniform [44.3 54.1]
AveAxleShort = (3 * LHCperms(:,12)) + 12; %Uniform [12 15]
AveAxleMed   = (3 * LHCperms(:,13)) + 15; %Uniform [15 18]
AveAxleLong  = (3 * LHCperms(:,14)) + 18; %Uniform [18 22]

GenTraffic = [round(MeanWheel*10)/10, round(WanderSD*10)/10,...
    round(LaneWidth*10)/10, round(AxleTand*10)/10,
    round(AxleTri*10)/10,...
    round(AxleQuad*10)/10, round(AveAxleShort*10)/10,...
    round(AveAxleMed*10)/10, round(AveAxleLong*10)/10];
%-----
----
%-----
----
AllTraffic = [Traffic, TrafficVol, GenTraffic];
%-----
----
%% Material Parameters
%-----
----
%Asphalt Layer
%-----
----
HMAtthick   = (4 * LHCperms(:,15)) + 6;
EBC         = (1.7 * LHCperms(:,16)) + 5.3;
AV          = (2.5 * LHCperms(:,17)) + 3;
UnitWt      = (30 * LHCperms(:,18)) + 135;

```

```

%Percent Passing: (modified 10142011 to reflect passing, not retained)
AggGrad34 = 100 - (2 * LHCperms(:,19)); %modify for Darwin-ME
AggGrad38 = 100 - ((13 * LHCperms(:,20)) + 7);
AggGrad4 = 100 - ((22 * LHCperms(:,21)) + 24);
%Percent Passing:
AggGrad200 = (6.2 * LHCperms(:,22));

ThermCond = (0.14 * LHCperms(:,23)) + 0.6;
HeatCap = (0.04 * LHCperms(:,24)) + 0.21;

for i = 1:n
    Gstar130(i,1) = ((14-3.5) * LHCperms(i,25) + 3.5) * 1000;
    %[3.5 14] kPa
    Gstar115(i,1) = ((25-14) * LHCperms(i,26) + 14) * 1000; %[14
25]
    Gstar100(i,1) = ((200-25) * LHCperms(i,27) + 25) * 1000; %[25
200]
    Gstar85(i,1) = ((230-200) * LHCperms(i,28) + 200) * 1000;
    %[200 230]
    Gstar70(i,1) = ((3300-230) * LHCperms(i,29) + 230) * 1000;
    %[230 3300]
    Gstar55(i,1) = ((4500-3300) * LHCperms(i,30) + 3300) * 1000;
    %[3300 4500]
    Gstar40(i,1) = ((33000-4500) * LHCperms(i,31) + 4500) * 1000;
    %[4500 33000]
end

%Let deltas vary between [50 85]; increasing with decreasing temps.
%REVISED: see excel file; decreasing with decreasing temp
delta130 = 9 * LHCperms(:,32) + 73; %[73 82]
delta115 = 3 * LHCperms(:,33) + 70; %[70 73]
delta100 = 3 * LHCperms(:,34) + 67; %[67 70]
delta85 = 4 * LHCperms(:,35) + 63; %[63 67]
delta70 = 6 * LHCperms(:,36) + 57; %[57 63]
delta55 = 2 * LHCperms(:,37) + 55; %[55 57]
delta40 = 8 * LHCperms(:,38) + 47; %[47 55]

%Verify Superpave requirements are met
for i = 1:n
    SP130(i,1) = Gstar130(i,1) / sind(delta130(i,1));
    SP115(i,1) = Gstar115(i,1) / sind(delta115(i,1));
    SP100(i,1) = Gstar100(i,1) / sind(delta100(i,1));
    SP85(i,1) = Gstar85(i,1) / sind(delta85(i,1));
    SP70(i,1) = Gstar70(i,1) / sind(delta70(i,1));
    SP55(i,1) = Gstar55(i,1) / sind(delta55(i,1));
    SP40(i,1) = Gstar40(i,1) / sind(delta40(i,1));
end

SPmin130 = min(SP130);
SPmin115 = min(SP115);
SPmin100 = min(SP100);
SPmin85 = min(SP85);
SPmin70 = min(SP70);
SPmin55 = min(SP55);

```

```

SPmin40 = min(SP40);

SPmin = min([SPmin130, SPmin115, SPmin100, SPmin85, SPmin70,...
    SPmin55, SPmin40]);

%Superpave Requirement: Gstar/sin(delta) > 1 kPa (1000 Pa)

CheckSP = SPmin - 1e3 %If CheckSP > 0 then values of G*, delta OK

AspLayer = [round(HMATHick*10)/10, round(EBC*10)/10,
    round(AV*10)/10,...
    round(UnitWt), round(AggGrad34*10)/10, round(AggGrad38*10)/10,...
    round(AggGrad4*10)/10, round(AggGrad200*10)/10,...
    round(ThermCond*100)/100, round(HeatCap*100)/100,
    round(Gstar40),...
    round(Gstar55), round(Gstar70), round(Gstar85), round(Gstar100),...
    round(Gstar115), round(Gstar130), round(delta40*10)/10,...
    round(delta55*10)/10, round(delta70*10)/10,
    round(delta85*10)/10,...
    round(delta100*10)/10, round(delta115*10)/10,
    round(delta130*10)/10];
%-----
----
%Granular Base Layer
%-----
----
GBthick    = (2      * LHCperms(:,39)) + 7;
Kgb        = (4500 * LHCperms(:,40)) + 35500;
GBPois     = (0.3   * LHCperms(:,41)) + 0.1;
GBKo       = (0.1   * LHCperms(:,42)) + 0.5;

GBLayer = [round(GBthick*10)/10, round(Kgb), round(GBPois*100)/100,...
    round(GBKo*1000)/1000];
%-----
----
%Unbounded Subgrade Layer
%-----
----
Esub       = (12000 * LHCperms(:,43)) + 12000;
SubPois    = (0.1   * LHCperms(:,44)) + 0.2;
SubKo      = (0.1   * LHCperms(:,45)) + 0.6;

SubLayer = [round(Esub), round(SubPois*100)/100,
    round(SubKo*1000)/1000];
%-----
----
%-----
----
Materials = [AspLayer, GBLayer, SubLayer];
%-----
----
%% Gaussian Process Surrogate Model Input Parameters

gpInputs = [AllTraffic, Materials];
save('RobustGPInputs.mat', 'gpInputs');

```

```

%-----
----
%% Various Plots to Show Scatter of Training Points
%-----
----
% %Choose Input Parameters to View
% xx = 1; yy = 14; zz = 23;
%
% %2-D Plot
% plot(gpInputs(:,xx), gpInputs(:,yy),
'LineStyle','none','Marker','.');
% axis([min(gpInputs(:,xx)) max(gpInputs(:,xx))...
%       min(gpInputs(:,yy)) max(gpInputs(:,yy))]);
% grid 'on'
%
% %3-D Plot
% figure
% plot3(gpInputs(:,xx), gpInputs(:,yy), gpInputs(:,zz),...
%       'LineStyle','none','Marker','.');
% axis([min(gpInputs(:,xx)) max(gpInputs(:,xx))...
%       min(gpInputs(:,yy)) max(gpInputs(:,yy))...
%       min(gpInputs(:,zz)) max(gpInputs(:,zz))]);
% grid 'on'


---


clear all; clc;

%Define Ave. Tensile Strength (St) and Creep Compliance (t) for given
%values of Gstar and delta; For import into RobustGPInputs matrix

%Import RobustGPInputs
%=====
====
%Modify for Climate Region
GPInputs = importdata('RobustGPInputsSpokane.mat');
%=====
====

Gstar = GPInputs(:,34:40);
delta = GPInputs(:,41:47);
Vbeff = GPInputs(:,25); %EBC
Va     = GPInputs(:,26); %AV

Temp = [40, 55, 70, 85, 100, 115, 130];

%-----
----
%% Find A and VTS
%-----
----
%Calculate Log(Temp R)
for i = 1:size(Temp,2)
    logTempR(1,i) = log10(Temp(1,i) + 459.67);
end

%Calculate Log(Log(Viscosity))

```

```

for i = 1:size(Gstar,1)
    for j = 1:size(Gstar,2)
        loglogvis(i,j) =
log10(log10((Gstar(i,j)/10)*((1/(sin(delta(i,j)*pi/180)))^4.8628)*1000)
);
        end
    end

%Perform Linear Regression to Obtain A and VTS
for i = 1:size(loglogvis,1)
    [r(i), VTS(i,1), A(i,1)] = regression(logTempR(1,:),
loglogvis(i,:));
end
clear r;
%-----
----
%% Calculate Average Tensile Strength (St)
%-----
----
%Regression Parameters

for i = 1:size(Va,1)
    Vasqd(i,1) = Va(i,1) * Va(i,1);
    VFA(i,1) = 100 * Vbeff(i,1) / (Vbeff(i,1) + Va(i,1));
    VFAsqd(i,1) = VFA(i,1) * VFA(i,1);
    Pen77(i,1) = 10^(290.5013-
sqrt(81177.288+257.0694*(10^(A(i,1)+2.72973*VTS(i,1)))));
end

StRegPs = [ones(size(Va,1),1), Va, Vasqd, VFA, VFAsqd, log10(Pen77),
log10(A)];

StRegCoeffs = [4976.34, -42.49, -2.73, -80.61, 0.465, 174.35, -
1217.54];

for i = 1:size(Va,1)
    St(i,1) = StRegCoeffs * StRegPs(i,:);
end

%-----
----
%% Calculate Creep Compliance
%-----
----
%Regression Parameters

% T = - 20 C
D120Cs = [-11.9254, 1.52206, 4.49876, -3.8132];
m20Cs = [-1.75987, 1.78187, 0.00089];

for i = 1:size(Va,1)
    logVa(i,1) = log10(Va(i,1));
    logVFA(i,1) = log10(VFA(i,1));
    logA(i,1) = log10(A(i,1));

```

```

    Va0203(i,1) = (Va(i,1))^0.0203;
    Pen7796(i,1) = (Pen77(i,1))^0.9687;
end

D120Ps = [ones(size(Va,1),1), logVa, logVFA, logA];
m20Ps = [ones(size(Va,1),1), Va0203, Pen7796];

for i = 1:size(Va,1)
    D120(i,1) = 10^(D120Cs * D120Ps(i,:));
    m20(i,1) = m20Cs * m20Ps(i,:);
end

% T = - 10 C
D110Cs = [-10.7656, 1.5196, 3.49983, -2.9987];
m10Cs = [-1.8269, 1.94218, 0.00098];

for i = 1:size(Va,1)
    Va016(i,1) = (Va(i,1))^0.016;
    Pen77969(i,1) = (Pen77(i,1))^0.96857;
end

D110Ps = D120Ps;
m10Ps = [ones(size(Va,1),1), Va016, Pen77969];

for i = 1:size(Va,1)
    D110(i,1) = 10^(D110Cs * D110Ps(i,:));
    m10(i,1) = m10Cs * m10Ps(i,:);
end

% T = 0 C
D10Cs = [-9.80627, 1.50845, 2.99, -2.90157];
m0Cs = [-2.41043, 2.59093, 0.00199];

for i = 1:size(Va,1)
    Va0155(i,1) = (Va(i,1))^0.01547;
    Pen7797(i,1) = (Pen77(i,1))^0.97247;
end

D10Ps = D120Ps;
m0Ps = [ones(size(Va,1),1), Va0155, Pen7797];

for i = 1:size(Va,1)
    D10(i,1) = 10^(D10Cs * D10Ps(i,:));
    m0(i,1) = m0Cs * m0Ps(i,:);
end

Dvalues = [D120, D110, D10];
mvalues = [m20, m10, m0];

for i = 1:size(Va,1)
    % Loading time = 1
    creep120(i,1) = Dvalues(i,1) * (1 ^ mvalues(i,1));
    creep110(i,1) = Dvalues(i,2) * (1 ^ mvalues(i,2));
    creep10(i,1) = Dvalues(i,3) * (1 ^ mvalues(i,3));

```



```

% Loading time = 2
creep220(i,1) = Dvalues(i,1) * (2 ^ mvalues(i,1));
creep210(i,1) = Dvalues(i,2) * (2 ^ mvalues(i,2));
creep20(i,1)  = Dvalues(i,3) * (2 ^ mvalues(i,3));
% Loading time = 5
creep520(i,1) = Dvalues(i,1) * (5 ^ mvalues(i,1));
creep510(i,1) = Dvalues(i,2) * (5 ^ mvalues(i,2));
creep50(i,1)  = Dvalues(i,3) * (5 ^ mvalues(i,3));
% Loadint time = 10
creep1020(i,1) = Dvalues(i,1) * (10 ^ mvalues(i,1));
creep1010(i,1) = Dvalues(i,2) * (10 ^ mvalues(i,2));
creep100(i,1)  = Dvalues(i,3) * (10 ^ mvalues(i,3));
% Loading time = 20
creep2020(i,1) = Dvalues(i,1) * (20 ^ mvalues(i,1));
creep2010(i,1) = Dvalues(i,2) * (20 ^ mvalues(i,2));
creep200(i,1)  = Dvalues(i,3) * (20 ^ mvalues(i,3));
% Loading time = 50
creep5020(i,1) = Dvalues(i,1) * (50 ^ mvalues(i,1));
creep5010(i,1) = Dvalues(i,2) * (50 ^ mvalues(i,2));
creep500(i,1)  = Dvalues(i,3) * (50 ^ mvalues(i,3));
% Loading time = 100
creep10020(i,1) = Dvalues(i,1) * (100 ^ mvalues(i,1));
creep10010(i,1) = Dvalues(i,2) * (100 ^ mvalues(i,2));
creep1000(i,1)  = Dvalues(i,3) * (100 ^ mvalues(i,3));
end

%Matrix of Creep Compliance Values
CC = [creep120, creep110, creep10, creep220, creep210, creep20,...
      creep520, creep510, creep50, creep1020, creep1010, creep100,
      creep2020,...
      creep2010, creep200, creep5020, creep5010, creep500, creep10020,...
      creep10010, creep1000];

%check compliance for first training point
CC1 = [CC(1,1:3); CC(1,4:6); CC(1,7:9); CC(1,10:12); CC(1,13:15);...
      CC(1,16:18); CC(1,19:21)];
%-----
----
%% Save Results
%-----
----
gpInputs = [GPInputs, St, CC];
save('RobustGPInputsFullSpokane.mat', 'gpInputs');
clear all; clc; fclose('all');
tin = tic;
GPInputs = importdata('RobustGPInputsFull.mat');

%define number of output files to be generated
numOF = 999;

for i = 100:numOF %***adjust loop***adjust outputname zeros in strcount
    i
    %open base file
    fin = fopen('RGPBaseModel.xml', 'r');

```

```

%create file to write to
%-----*****-----
----
strcount = num2str(i);
strcount = strcat('0', strcount);
%-----*****-----
----
outputname = 'C:\Users\Jenny\Documents\Retherford-
Vanderbilt\Fall2011\RobustSurrogateModel\Matlab\RGPTTrainingPoints\RGP';
outputname = strcat(outputname, strcount, '.xml');

fout = fopen(outputname, 'w');

%define counter to see position in execution
position = ftell(fin) + 1;
%-----*****-----
----
%% Initial Lines
%-----*****-----
----
for position = 1:216566
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
for position = 216567
    GPIval = strcat('RGP', strcount);
    fline = fgetl(fin);
    tline = strcat('    <displayName>',GPIval, '</displayName>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
for position = 216568:216752
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----*****-----
----
%% Asphalt
%-----*****-----
----
% 24 thickness 1
for position =216753
    GPIval = num2str(GPInputs(i,24));
    fline = fgetl(fin);
    tline = strcat('    <thickness>',GPIval, '</thickness>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----*****-----
----
for position = 216754:216803
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);

```

```

        position = position + 1;
end
%-----
-----
% 31 pass #200 2
for position = 216804
    GPIval = num2str(GPInputs(i,31));
    fline = fgetl(fin);
    tline = strcat('                <p200>',GPIval, '</p200>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end

% Agg. Gradation
% 28 pass 3/4" 3
for position = 216805
    GPIval = num2str(GPInputs(i,28));
    fline = fgetl(fin);
    tline = strcat('                <p3_4>',GPIval, '</p3_4>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
% 29 pass 3/8" 4
for position = 216806
    GPIval = num2str(GPInputs(i,29));
    fline = fgetl(fin);
    tline = strcat('                <p3_8>',GPIval, '</p3_8>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
% 30 pass #4 5
for position = 216807
    GPIval = num2str(GPInputs(i,30));
    fline = fgetl(fin);
    tline = strcat('                <p4>',GPIval, '</p4>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
for position = 216808:216824
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
% 26 AV 6
for position = 216825
    GPIval = num2str(GPInputs(i,26));
    fline = fgetl(fin);
    tline = strcat('                <airVoids>',GPIval, '</airVoids>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end

```

```

% 27 Unit Wt. 7
for position = 216826
    GPIval = num2str(GPInputs(i,27));
    fline = fgetl(fin);
    tline = strcat('          <totalWeight>',GPIval,
'</totalWeight>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end

% 32 Thermal Cond. 8
for position = 216827
    GPIval = num2str(GPInputs(i,32));
    fline = fgetl(fin);
    tline = strcat('          <thermalConductivity>',GPIval,
'</thermalConductivity>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end

% 33 Heat Cap. 9
for position = 216828
    GPIval = num2str(GPInputs(i,33));
    fline = fgetl(fin);
    tline = strcat('          <heatCapacity>',GPIval,
'</heatCapacity>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end

% 25 EBC 10
for position = 216829
    GPIval = num2str(GPInputs(i,25));
    fline = fgetl(fin);
    tline = strcat('          <binderContent>',GPIval,
'</binderContent>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end

%-----
-----
for position = 216830:217003
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----

% Superpave Binder Info (Gstar, delta, @ T's) 11:24
% T = 40
for position = 217004
    GPIval = num2str(GPInputs(i,34));
    fline = fgetl(fin);
    tline = strcat('          <gStar>',GPIval, '</gStar>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end

```

```

end
% T = 40
for position = 217005
    GPIval = num2str(GPInputs(i,41));
    fline = fgetl(fin);
    tline = strcat('          <delta>',GPIval, '</delta>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
for position = 217006:217008
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
% T = 55
for position = 217009
    GPIval = num2str(GPInputs(i,35));
    fline = fgetl(fin);
    tline = strcat('          <gStar>',GPIval, '</gStar>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
% T = 55
for position = 217010
    GPIval = num2str(GPInputs(i,42));
    fline = fgetl(fin);
    tline = strcat('          <delta>',GPIval, '</delta>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
for position = 217011:217013
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
%T = 70
for position = 217014
    GPIval = num2str(GPInputs(i,36));
    fline = fgetl(fin);
    tline = strcat('          <gStar>',GPIval, '</gStar>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%T = 70
for position = 217015
    GPIval = num2str(GPInputs(i,43));
    fline = fgetl(fin);
    tline = strcat('          <delta>',GPIval, '</delta>');

```

```

        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 217016:217018
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
% T = 85
    for position = 217019
        GPIval = num2str(GPInputs(i,37));
        fline = fgetl(fin);
        tline = strcat('                <gStar>',GPIval, '</gStar>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
% T = 85
    for position = 217020
        GPIval = num2str(GPInputs(i,44));
        fline = fgetl(fin);
        tline = strcat('                <delta>',GPIval, '</delta>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 217021:217023
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
%T = 100
    for position = 217024
        GPIval = num2str(GPInputs(i,38));
        fline = fgetl(fin);
        tline = strcat('                <gStar>',GPIval, '</gStar>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%T = 100
    for position = 217025
        GPIval = num2str(GPInputs(i,45));
        fline = fgetl(fin);
        tline = strcat('                <delta>',GPIval, '</delta>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 217026:217028

```

```

        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
    %-----
-----
    % T = 115
    for position = 217029
        GPIval = num2str(GPInputs(i,39));
        fline = fgetl(fin);
        tline = strcat('                <gStar>',GPIval, '</gStar>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
    % T = 115
    for position = 217030
        GPIval = num2str(GPInputs(i,46));
        fline = fgetl(fin);
        tline = strcat('                <delta>',GPIval, '</delta>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
    %-----
-----
    for position = 217031:217033
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
    %-----
-----
    % T = 130
    for position = 217034
        GPIval = num2str(GPInputs(i,40));
        fline = fgetl(fin);
        tline = strcat('                <gStar>',GPIval, '</gStar>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
    % T = 130
    for position = 217035
        GPIval = num2str(GPInputs(i,47));
        fline = fgetl(fin);
        tline = strcat('                <delta>',GPIval, '</delta>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
    %-----
-----
    %-----
-----
    for position = 217036:217039
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
end

```

```

-----
%-----
-----
%55 Indirect Ave. Tensile Strength 25
for position = 217040
    GPIval = num2str(GPInputs(i,55));
    fline = fgetl(fin);
    tline = strcat('                <tensileStrength>',GPIval,
'</tensileStrength>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end

%-----
-----
for position = 217041:217051
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end

%-----
-----
%56: 76 CreepCompliance 26:46
for position = 217052
    GPIval = num2str(GPInputs(i,56));
    fline = fgetl(fin);
    tline = strcat('                <creepCompliance>',GPIval,
'</creepCompliance>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end

%-----
-----
for position = 217053:217056
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end

%-----
-----
for position = 217057
    GPIval = num2str(GPInputs(i,57));
    fline = fgetl(fin);
    tline = strcat('                <creepCompliance>',GPIval,
'</creepCompliance>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end

%-----
-----
for position = 217058:217061
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end

%-----
-----

```



```

    for position = 217062
        GPIval = num2str(GPInputs(i,58));
        fline = fgetl(fin);
        tline = strcat('                <creepCompliance>',GPIval,
'</creepCompliance>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
-----
    for position = 217063:217066
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
-----
    for position = 217067
        GPIval = num2str(GPInputs(i,59));
        fline = fgetl(fin);
        tline = strcat('                <creepCompliance>',GPIval,
'</creepCompliance>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
-----
    for position = 217068:217071
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
-----
    for position = 217072
        GPIval = num2str(GPInputs(i,60));
        fline = fgetl(fin);
        tline = strcat('                <creepCompliance>',GPIval,
'</creepCompliance>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
-----
    for position = 217073:217076
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
-----
    for position = 217077
        GPIval = num2str(GPInputs(i,61));
        fline = fgetl(fin);
        tline = strcat('                <creepCompliance>',GPIval,
'</creepCompliance>');

```

```

        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 217078:217081
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 217082
        GPIval = num2str(GPInputs(i,62));
        fline = fgetl(fin);
        tline = strcat('                <creepCompliance>', GPIval,
'</creepCompliance>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 217083:217086
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 217087
        GPIval = num2str(GPInputs(i,63));
        fline = fgetl(fin);
        tline = strcat('                <creepCompliance>', GPIval,
'</creepCompliance>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 217088:217091
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 217092
        GPIval = num2str(GPInputs(i,64));
        fline = fgetl(fin);
        tline = strcat('                <creepCompliance>', GPIval,
'</creepCompliance>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----

```

```

for position = 217093:217096
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
----
for position = 217097
    GPIval = num2str(GPInputs(i,65));
    fline = fgetl(fin);
    tline = strcat('                <creepCompliance>', GPIval,
'</creepCompliance>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
----
for position = 217098:217101
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
----
for position = 217102
    GPIval = num2str(GPInputs(i,66));
    fline = fgetl(fin);
    tline = strcat('                <creepCompliance>', GPIval,
'</creepCompliance>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
----
for position = 217103:217106
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
----
for position = 217107
    GPIval = num2str(GPInputs(i,67));
    fline = fgetl(fin);
    tline = strcat('                <creepCompliance>', GPIval,
'</creepCompliance>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
----
for position = 217108:217111
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end

```

```

-----
%-----
-----
for position = 217112
    GPIval = num2str(GPInputs(i,68));
    fline = fgetl(fin);
    tline = strcat('                <creepCompliance>',GPIval,
'</creepCompliance>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
for position = 217113:217116
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
for position = 217117
    GPIval = num2str(GPInputs(i,69));
    fline = fgetl(fin);
    tline = strcat('                <creepCompliance>',GPIval,
'</creepCompliance>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
for position = 217118:217121
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
for position = 217122
    GPIval = num2str(GPInputs(i,70));
    fline = fgetl(fin);
    tline = strcat('                <creepCompliance>',GPIval,
'</creepCompliance>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
for position = 217123:217126
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
for position = 217127
    GPIval = num2str(GPInputs(i,71));
    fline = fgetl(fin);

```

```

        tline = strcat('                <creepCompliance>', GPIval,
'</creepCompliance>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 217128:217131
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 217132
        GPIval = num2str(GPInputs(i,72));
        fline = fgetl(fin);
        tline = strcat('                <creepCompliance>', GPIval,
'</creepCompliance>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 217133:217136
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 217137
        GPIval = num2str(GPInputs(i,73));
        fline = fgetl(fin);
        tline = strcat('                <creepCompliance>', GPIval,
'</creepCompliance>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 217138:217141
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 217142
        GPIval = num2str(GPInputs(i,74));
        fline = fgetl(fin);
        tline = strcat('                <creepCompliance>', GPIval,
'</creepCompliance>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
end

```

```

-----
%-----
for position = 217143:217146
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
for position = 217147
    GPIval = num2str(GPInputs(i,75));
    fline = fgetl(fin);
    tline = strcat('                <creepCompliance>', GPIval,
'</creepCompliance>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
for position = 217148:217151
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
for position = 217152
    GPIval = num2str(GPInputs(i,76));
    fline = fgetl(fin);
    tline = strcat('                <creepCompliance>', GPIval,
'</creepCompliance>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
for position = 217153:217158
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
%-----
%% Granular Base
%-----
-----
% 48 Thickness 47
for position = 217159
    GPIval = num2str(GPInputs(i,48));
    fline = fgetl(fin);
    tline = strcat('                <thickness>', GPIval, '</thickness>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;

```

```

end
%-----
-----
for position = 217160:217199
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
% 50 Poisson 48
for position = 217200
    GPIval = num2str(GPInputs(i,50));
    fline = fgetl(fin);
    tline = strcat('          <poisson>',GPIval, '</poisson>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end

% 51 Coeff. Lat. 49
for position = 217201
    GPIval = num2str(GPInputs(i,51));
    fline = fgetl(fin);
    tline = strcat('          <k0>',GPIval, '</k0>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
for position = 217202:217275
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
% 49 Modulus 50
for position = 217276
    GPIval = num2str(GPInputs(i,49));
    fline = fgetl(fin);
    tline = strcat('          <unbValue>',GPIval,
'</unbValue>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
for position = 217277:217470
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----

```

```

-----
%% Sub-Base
-----

% 53 Poisson 51
for position = 217471
    GPIval = num2str(GPInputs(i,53));
    fline = fgetl(fin);
    tline = strcat('          <poisson>',GPIval, '</poisson>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end

% 54 Coeff. Lat. 52
for position = 217472
    GPIval = num2str(GPInputs(i,54));
    fline = fgetl(fin);
    tline = strcat('          <k0>',GPIval, '</k0>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
-----

for position = 217473:217546
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
-----

% 52 Modulus 53
for position = 217547
    GPIval = num2str(GPInputs(i,52));
    fline = fgetl(fin);
    tline = strcat('          <unbValue>',GPIval,
'</unbValue>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
-----

for position = 217548:218121
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
-----

-----

%% TRAFFIC
-----

```



```

% 1:aadt 54
for position = 218122
    GPIval = num2str(GPInputs(i,1));
    fline = fgetl(fin);
    tline = strcat('    <aadt>', GPIval, '</aadt>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
for position = 218123:218124
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----

% 2 lane distribution 55
for position = 218125
    GPIval = num2str(GPInputs(i,2));
    fline = fgetl(fin);
    tline = strcat('    <percentTrucksDesignLane>', GPIval,
'</percentTrucksDesignLane>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end

% 3 operating speed 56
for position = 218126
    GPIval = num2str(GPInputs(i,3));
    fline = fgetl(fin);
    tline = strcat('    <trafficSpeed>',GPIval, '</trafficSpeed>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
for position = 218127
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----

% 15 mean wheel 57
for position = 218128
    GPIval = num2str(GPInputs(i,15));
    fline = fgetl(fin);
    tline = strcat('    <meanWheelLocation>', GPIval,
'</meanWheelLocation>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end

```

```

% 16 wander st dev 58
for position = 218129
    GPIval = num2str(GPInputs(i,16));
    fline = fgetl(fin);
    tline = strcat('    <trafficWander>', GPIval,
'</trafficWander>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end

% 17 lane width 59
for position = 218130
    GPIval = num2str(GPInputs(i,17));
    fline = fgetl(fin);
    tline = strcat('    <laneWidth>',GPIval, '</laneWidth>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
for position = 218131:218132
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
% 14 traffic growth 60
for position = 218133
    GPIval = num2str(GPInputs(i,14));
    fline = fgetl(fin);
    tline = strcat('
<allClassGrowthrate>',GPIval,'</allClassGrowthrate>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
for position = 218134:218137
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
% 18 axle sp. - tandem 61
for position = 218138
    GPIval = num2str(GPInputs(i,18));
    fline = fgetl(fin);
    tline = strcat('    <dualAxleSpacing>',GPIval,
'</dualAxleSpacing>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end

% 19 axle sp. - tridem 62

```

```

    for position = 218139
        GPIval = num2str(GPInputs(i,19));
        fline = fgetl(fin);
        tline = strcat('    <tripleAxleSpacing>',GPIval,
'</tripleAxleSpacing>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end

    % 20 axle sp. - quad. 63
    for position = 218140
        GPIval = num2str(GPInputs(i,20));
        fline = fgetl(fin);
        tline = strcat('    <quadAxleSpacing>',GPIval,
'</quadAxleSpacing>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end

    % 21 ave. axle sp. - short 64
    for position = 218141
        GPIval = num2str(GPInputs(i,21));
        fline = fgetl(fin);
        tline = strcat('    <shortAxleSpacing>',GPIval,
'</shortAxleSpacing>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end

    % 22 ave. axle sp. - med 65
    for position = 218142
        GPIval = num2str(GPInputs(i,22));
        fline = fgetl(fin);
        tline = strcat('    <mediumAxleSpacing>',GPIval,
'</mediumAxleSpacing>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end

    % 23 ave. axle sp. - long 66
    for position = 218143
        GPIval = num2str(GPInputs(i,23));
        fline = fgetl(fin);
        tline = strcat('    <longAxleSpacing>',GPIval,
'</longAxleSpacing>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end

    end

    %-----
    -----
    for position = 218144:218182
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
end

```

```

-----
%-----
-----

% 4:13 distribution of vehicle classes 67:76
for position = 218183
    GPIval = num2str(GPInputs(i,4));
    fline = fgetl(fin);
    tline = strcat('          <percentTrucksPerClass>', GPIval,
'</percentTrucksPerClass>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----

for position = 218184:218186
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----

for position = 218187
    GPIval = num2str(GPInputs(i,5));
    fline = fgetl(fin);
    tline = strcat('          <percentTrucksPerClass>', GPIval,
'</percentTrucksPerClass>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----

for position = 218188:218190
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----

for position = 218191
    GPIval = num2str(GPInputs(i,6));
    fline = fgetl(fin);
    tline = strcat('          <percentTrucksPerClass>', GPIval,
'</percentTrucksPerClass>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----

for position = 218192:218194
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----

for position = 218195

```

```

        GPIval = num2str(GPInputs(i,7));
        fline = fgetl(fin);
        tline = strcat('          <percentTrucksPerClass>', GPIval,
'</percentTrucksPerClass>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
    %-----
-----
    for position = 218196:218198
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
    %-----
-----
    for position = 218199
        GPIval = num2str(GPInputs(i,8));
        fline = fgetl(fin);
        tline = strcat('          <percentTrucksPerClass>', GPIval,
'</percentTrucksPerClass>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
    %-----
-----
    for position = 218200:218202
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
    %-----
-----
    for position = 218203
        GPIval = num2str(GPInputs(i,9));
        fline = fgetl(fin);
        tline = strcat('          <percentTrucksPerClass>', GPIval,
'</percentTrucksPerClass>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
    %-----
-----
    for position = 218204:218206
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
    %-----
-----
    for position = 218207
        GPIval = num2str(GPInputs(i,10));
        fline = fgetl(fin);
        tline = strcat('          <percentTrucksPerClass>', GPIval,
'</percentTrucksPerClass>');
        fprintf(fout, '%s\n', tline);

```

```

        position = position + 1;
    end
%-----
----
    for position = 218208:218210
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 218211
        GPIval = num2str(GPInputs(i,11));
        fline = fgetl(fin);
        tline = strcat('          <percentTrucksPerClass>', GPIval,
'</percentTrucksPerClass>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 218212:218214
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 218215
        GPIval = num2str(GPInputs(i,12));
        fline = fgetl(fin);
        tline = strcat('          <percentTrucksPerClass>', GPIval,
'</percentTrucksPerClass>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 218216:218218
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 218219
        GPIval = num2str(GPInputs(i,13));
        fline = fgetl(fin);
        tline = strcat('          <percentTrucksPerClass>', GPIval,
'</percentTrucksPerClass>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 218220:218388

```

```

        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
    %-----
----

    for position = 218389 %77:86
        GPIval = num2str(GPInputs(i,14));
        fline = fgetl(fin);
        tline = strcat('
<percentGrowthRate>', GPIval, '</percentGrowthRate>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
    %-----
----

    for position = 218390:218393
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
    %-----
----

    for position = 218394
        GPIval = num2str(GPInputs(i,14));
        fline = fgetl(fin);
        tline = strcat('
<percentGrowthRate>', GPIval, '</percentGrowthRate>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
    %-----
----

    for position = 218395:218398
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
    %-----
----

    for position = 218399
        GPIval = num2str(GPInputs(i,14));
        fline = fgetl(fin);
        tline = strcat('
<percentGrowthRate>', GPIval, '</percentGrowthRate>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
    %-----
----

    for position = 218400:218403
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
end

```

```

-----
%-----
-----
for position = 218404
    GPIval = num2str(GPInputs(i,14));
    fline = fgetl(fin);
    tline = strcat('
<percentGrowthRate>',GPIval,'</percentGrowthRate>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
for position = 218405:218408
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
for position = 218409
    GPIval = num2str(GPInputs(i,14));
    fline = fgetl(fin);
    tline = strcat('
<percentGrowthRate>',GPIval,'</percentGrowthRate>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
for position = 218410:218413
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
for position = 218414
    GPIval = num2str(GPInputs(i,14));
    fline = fgetl(fin);
    tline = strcat('
<percentGrowthRate>',GPIval,'</percentGrowthRate>');
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
for position = 218415:218418
    tline = fgetl(fin);
    fprintf(fout, '%s\n', tline);
    position = position + 1;
end
%-----
-----
for position = 218419
    GPIval = num2str(GPInputs(i,14));
    fline = fgetl(fin);

```



```

        tline = strcat('
<percentGrowthRate>', GPIval, '</percentGrowthRate>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 218420:218423
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 218424
        GPIval = num2str(GPInputs(i,14));
        fline = fgetl(fin);
        tline = strcat('
<percentGrowthRate>', GPIval, '</percentGrowthRate>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 218425:218428
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 218429
        GPIval = num2str(GPInputs(i,14));
        fline = fgetl(fin);
        tline = strcat('
<percentGrowthRate>', GPIval, '</percentGrowthRate>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 218430:218433
        tline = fgetl(fin);
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
%-----
----
    for position = 218434
        GPIval = num2str(GPInputs(i,14));
        fline = fgetl(fin);
        tline = strcat('
<percentGrowthRate>', GPIval, '</percentGrowthRate>');
        fprintf(fout, '%s\n', tline);
        position = position + 1;
    end
end

```

```
-----  
%-----  
-----  
%% Remainder of inputs  
%-----  
-----  
for position = 218435:219317  
    tline = fgetl(fin);  
    fprintf(fout, '%s\n', tline);  
    position = position + 1;  
end  
fclose('all');  
end  
tout = toc(tin);
```

APPENDIX B: Matlab Numerical Computation Sample Codes

Sample Matlab codes demonstrating the computational work presented in this dissertation are included herein.

Chapter III

Calculation of Validation Metrics

```
clear all; clc;
RegressionResults = importdata('C:\Users\Jenny\Documents\Retherford-
Vanderbilt\Fall2010\WestTrack\EXCELwestrack\TRB91FinalModels.xlsx');

%-----
%-----
%%   Westrack Results
%-----
%-----
%Build Matrix of Westrack Measurements
WestrackResultsAll = RegressionResults.data.Westrack;
%Remove columns from Westrack that do not have Shear parameter results
WestrackResults = RegressionResults.data.Westrack;
WestrackResults(:, [2:3, 5:6, 8, 10, 16, 17, 26]) = [];
%Modify matrix to single column vector for validation analyses
WestrackResults = tocol(WestrackResults);

%-----
%-----
%%   National Model
%-----
%-----
National = RegressionResults.data.National;
National = National(1:12, 1:26);
National(:, [2:3, 5:6, 8, 10, 16, 17, 26]) = [];
National = tocol(National);

%-----
%-----
%%   NCHRP Model
%-----
%-----
NCHRP = RegressionResults.data.NCHRPRegEqn;
NCHRP = NCHRP(1:12, 1:26);
NCHRP(:, [2:3, 5:6, 8, 10, 16, 17, 26]) = [];
```

```

NCHRP = tocol(NCHRP);

%-----
%-----
%%  Locally Calibrated      (Optimal where Betas = [2.875, 1, 1])
%-----
%-----
LocallyCalibrated = RegressionResults.data.LocalAll;
LocallyCalibrated = LocallyCalibrated(1:12,1:26);
LocallyCalibrated(:, [2:3,5:6,8,10,16,17,26]) = [];
LocallyCalibrated = tocol(LocallyCalibrated);

%-----
%-----
%%  Parameter Calibrated    (Linear Regression from Locally Calibrated)
%-----
%-----
ParamCalibrated = RegressionResults.data.LocalRegBased;
ParamCalibrated = ParamCalibrated(1:12,1:26);
ParamCalibrated(:, [2:3,5:6,8,10,16,17,26]) = [];
ParamCalibrated = tocol(ParamCalibrated);

%-----
%-----
%%  WeightedON              (Weighted NCHRP with Locally Calibrated)
%-----
%-----
WeightedON = RegressionResults.data.WeightedON;
WeightedON = WeightedON(1:12,1:26);
WeightedON(:, [2:3,5:6,8,10,16,17,26]) = [];
WeightedON = tocol(WeightedON);

%-----
%-----
%%  WeightedRN              (Weighted NCHRP with Parameter Calibrated)
%-----
%-----
WeightedRN = RegressionResults.data.WeightedRN;
WeightedRN = WeightedRN(1:12,1:26);
WeightedRN(:, [2:3,5:6,8,10,16,17,26]) = [];
WeightedRN = tocol(WeightedRN);

%-----
%-----
%%  Mean Square Error
%-----
%-----
for i = 1:size(WestrackResults,1)
    ErrorNat(i,1) = WestrackResults(i,1) - National(i,1);
    ErrorLoc(i,1) = WestrackResults(i,1) - LocallyCalibrated(i,1);
    ErrorParam(i,1) = WestrackResults(i,1) - ParamCalibrated(i,1);
    ErrorNCHRP(i,1) = WestrackResults(i,1) - NCHRP(i,1);

```

```

        ErrorWON(i,1) = WestrackResults(i,1) - WeightedON(i,1);
        ErrorWRN(i,1) = WestrackResults(i,1) - WeightedRN(i,1);
    end
    ModelError = ...
        [ErrorNat, ErrorLoc, ErrorParam, ErrorNCHRP, ErrorWON, ErrorWRN];

%-----
----
%Scatter plot to show residuals

figure
xplot = (1:1:204);
scatter(xplot, abs(ErrorNat),'.r');
ylabel('Abs. Value of Residuals (in.)');
xlabel('Test Sections');
title('National Model Residuals');

figure
hold on
scatter(xplot, abs(ErrorNat),'.r');
scatter(xplot, abs(LocallyCalibrated),'.g');
legend('National','Local');
ylabel('Abs. Value of Residuals (in.)');
xlabel('Test Sections');
title('National and Local Model Residuals');
hold off

figure
hold on
scatter(xplot, abs(ErrorNat),'.r');
scatter(xplot, abs(LocallyCalibrated),'.g');
scatter(xplot, abs(ErrorWRN),'.b');
legend('National','Local','WeightedParam');
ylabel('Abs. Value of Residuals (in.)');
xlabel('Test Sections');
title('National, Local, and Parameter Weighted Residuals');
hold off
% figure
% hold on
% scatter(WestrackResults, National, '.');
% scatter(WestrackResults, LocallyCalibrated, '.');
% scatter(WestrackResults, ParamCalibrated, '.');
% scatter(WestrackResults, NCHRP, '.');
% scatter(WestrackResults, WeightedON, '.');
% scatter(WestrackResults, WeightedRN, '.');
%
legend('National','Local','Parameter','NCHRP','WeightedLocal','Weighted
Param');
% hold off

for i = 1:size(WestrackResults,1)
    SqErNat(i,1) = (ErrorNat(i,1))^2;
    SqErLoc(i,1) = (ErrorLoc(i,1))^2;
    SqErParam(i,1) = (ErrorParam(i,1))^2;

```

```

    SqErNCHRP(i,1) = (ErrorNCHRP(i,1))^2;
    SqErWON(i,1) = (ErrorWON(i,1))^2;
    SqErWRN(i,1) = (ErrorWRN(i,1))^2;
end

MSENat      = mean(SqErNat,1);
MSELoc      = mean(SqErLoc,1);
MSEParam    = mean(SqErParam,1);
MSENCHRP    = mean(SqErNCHRP,1);
MSEWON      = mean(SqErWON,1);
MSEWRN      = mean(SqErWRN,1);

MSE = [MSENat; MSELoc; MSEParam; MSENCHRP; MSEWON; MSEWRN];

MSENatStd   = std(SqErNat,1);
MSELocStd   = std(SqErLoc,1);
MSEParamStd = std(SqErParam,1);
MSENCHRPStd = std(SqErNCHRP,1);
MSEWONStd   = std(SqErWON,1);
MSEWRNStd   = std(SqErWRN,1);
MSEstd = [MSENatStd; MSELocStd; MSEParamStd; MSENCHRPStd; MSEWONStd;
MSEWRNStd];

clear f xi
%mean = E(Residuals)
%stdev = Std(Residuals)
for i = 1:size(ModelError,2)
%    [f(:,i), xi(:,i)] = ksdensity(ModelError(:,i));
xi(:,i) = (-1:0.01:1);
f(:,i) = normpdf(xi(:,i), mean(ModelError(:,i)), std(ModelError(:,i)));
end

for i = 1:size(ModelError,2)
    figure
    plot(xi(:,i), f(:,i));
end

%-----
----
%PDF plot to show residuals: Figure III.2
figure
hold on
for i = 1:size(ModelError,2)
    if i == 1
        Line = 'r';
    elseif i == 2
        Line = 'g';
    elseif i == 3
        Line = 'b';
    elseif i == 4
        Line = 'c';
    elseif i == 5
        Line = 'm';
    else Line = 'k';
    end
end

```

```

plot(xi(:,i),f(:,i), Line);

if i == size(MSE,1)
    clear xlabel ylabel
    xlabel('Residuals');
    ylabel('PDF(Residuals)');
    legend('National','Locally Calibrated','Parameter
Calibrated','NCHRP (Shear Based)','Weighted: Local and
NCHRP','Weighted: Parameter and NCHRP', 'Location',
'SouthEastOutside');
end
end
hold off

%-----
----
%% Coefficient of Determination (R^2)
%-----
----
MeanWestrack = mean(WestrackResults,1); %This is ybar

for i = 1:size(WestrackResults,1)
    SSWestrack(i,1) = (WestrackResults(i,1) - MeanWestrack)^2;%These
are yi
end
SStot = sum(SSWestrack);

for i = 1:size(WestrackResults,1)
    SSerrN(i,1) = (WestrackResults(i,1) - National(i,1))^2;
    SSerrL(i,1) = (WestrackResults(i,1) - LocallyCalibrated(i,1))^2;
    SSerrP(i,1) = (WestrackResults(i,1) - ParamCalibrated(i,1))^2;
    SSerrNC(i,1) = (WestrackResults(i,1) - NCHRP(i,1))^2;
    SSerrWO(i,1) = (WestrackResults(i,1) - WeightedON(i,1))^2;
    SSerrWR(i,1) = (WestrackResults(i,1) - WeightedRN(i,1))^2;
end
SSerrNat = sum(SSerrN,1);
SSerrLoc = sum(SSerrL,1);
SSerrParam = sum(SSerrP,1);
SSerrNCHRP = sum(SSerrNC,1);
SSerrWON = sum(SSerrWO,1);
SSerrWRN = sum(SSerrWR,1);

RsqdNat = 1 - (SSerrNat / SStot);
RsqdLoc = 1 - (SSerrLoc / SStot);
RsqdParam = 1 - (SSerrParam / SStot);
RsqdNCHRP = 1 - (SSerrNCHRP / SStot);
RsqdWON = 1 - (SSerrWON / SStot);
RsqdWRN = 1 - (SSerrWRN / SStot);

Rsquared = [RsqdNat; RsqdLoc; RsqdParam; RsqdNCHRP; RsqdWON; RsqdWRN];
%-----
----
%% Adjusted R squared

```

```

Natregstats = regstats(WestrackResults, National, 'linear',
{'rsquare','adjrsquare'});
Locregstats = regstats(WestrackResults, LocallyCalibrated, 'linear',
{'rsquare','adjrsquare'});
Paramregstats = regstats(WestrackResults, ParamCalibrated, 'linear',
{'rsquare','adjrsquare'});
NCHRPregstats = regstats(WestrackResults, NCHRP, 'linear',
{'rsquare','adjrsquare'});
WONregstats = regstats(WestrackResults, WeightedON, 'linear',
{'rsquare','adjrsquare'});
WRNregstats = regstats(WestrackResults, WeightedRN, 'linear',
{'rsquare','adjrsquare'});

AdjRsquared = [Natregstats.adjrsquare; Locregstats.adjrsquare;...
Paramregstats.adjrsquare; NCHRPregstats.adjrsquare; ...
WONregstats.adjrsquare; WRNregstats.adjrsquare];
%-----
----
%% Bayes Factor
%-----
----

%Evaluate residuals for normal pdf with
%mean = mean of the model error; stdev = stdev of the model error
MeanNat = mean(ErrorNat,1);
MeanLoc = mean(ErrorLoc,1);
MeanParam = mean(ErrorParam,1);
MeanNCHRP = mean(ErrorNCHRP,1);
MeanWON = mean(ErrorWON,1);
MeanWRN = mean(ErrorWRN,1);

Means = [MeanNat, MeanLoc, MeanParam, MeanNCHRP, MeanWON, MeanWRN];

StDevNat = std(ErrorNat,1);
StDevLoc = std(ErrorLoc,1);
StDevParam = std(ErrorParam,1);
StDevNCHRP = std(ErrorNCHRP,1);
StDevWON = std(ErrorWON,1);
StDevWRN = std(ErrorWRN,1);

Stdevs = [StDevNat, StDevLoc, StDevParam, StDevNCHRP, StDevWON,
StDevWRN];

for i = 1:size(ModelError,1)
    for j = 1:size(ModelError,2)
        EvalPDF(i,j) = normpdf(ModelError(i,j),Means(1,j),Stdevs(1,j));
    end
end
Products = prod(EvalPDF,1);

%Calculate Bayes Factor all compared to National Model
for i = 1:size(Products,2);
    Bayes(i,1) = Products(1,i) / Products(1,1);
end

```



```

%-----
----
%% F-Test
%-----
----
%Define Number of Model Parameters
pNat = 3;
pLoc = 3;
pParam = 9;
pNCHRP = 4;
pWON = pNCHRP + pLoc;
pWRN = pNCHRP + pParam;

pAll = [pNat; pLoc; pParam; pNCHRP; pWON; pWRN];

nTestPoints = size(WestrackResults,1);

%Create vector of Sum Sqd Residuals for use in F-test
Rss = [SSerrNat; SSerrLoc; SSerrParam; SSerrNCHRP; SSerrWON; SSerrWRN];

%Calculate F-Test with respect to Local Model
for i = 1:size(pAll,1)
    Ftest(i,1) = ((Rss(2,1) - Rss(i,1)) / (pAll(i,1) - pAll(2,1))) /
    ...
    (Rss(i,1) / (nTestPoints - pAll(i,1)));
end

%Find Critical F
Falpha = 0.01; fP = 1 - Falpha;
%Second and third terms for finv
for i = 1:size(pAll,1)
    v1(1,i) = pAll(i,1) - pAll(2,1);
    v2(1,i) = nTestPoints - pAll(i,1);
end

for i = 1:size(pAll,1)
    Fcrit(i,1) = finv(fP,v1(1,i), v2(1,i));
end

%-----
----
%% Validation Metrics Summary Matrix
%-----
----
ValidationMatrix = [MSE, AdjRsquared, Bayes, Ftest, Fcrit];

```

Chapter IV

Evaluation for Adjusted R-Square Values for Variable Selection Method Comparisons

```
clear all; clc;
addpath('C:\Users\Jenny\Documents\Retherford-
Vanderbilt\Fall2011\RobustSurrogateModel\Matlab\accre\accreInput');
addpath('C:\Users\Jenny\Documents\Retherford-
Vanderbilt\Fall2011\RobustSurrogateModel\Matlab\accre\accreOutput');

% Import Training Points
% pnts = inputs
% vals = output from Darwin-ME
%%
=====
=
% Import Training Data
%
=====
=
InitRGP = RGPinitializerOrig;
RGP_vals = InitRGP{1,1};
RGP_pnts = InitRGP{1,2};
RGP_pntnames = InitRGP{1,3};

%Import Data: GPDiscrepancy values for each selection process
%Number of Models
num_Models = 5;
for MEPDGmodel = 1:num_Models
    Modelnum = num2str(MEPDGmodel);
    DiscNameA = strcat('RGPanovaAllmodel_', Modelnum, '.txt');
    NameAixs = strcat('RGPanovaAllmodel_', Modelnum, 'ParamsIX.txt');
    DiscNameF = strcat('RGPforcorrAllmodel_', Modelnum, '.txt');
    NameFixs = strcat('RGPforcorrAllmodel_', Modelnum,
'ParamsIX.txt');
    DiscNameS = strcat('RGPslfAllmodel_', Modelnum, '.txt');
    NameSixs = strcat('RGPslfAllmodel_', Modelnum, 'ParamsIX.txt');

    AnovaResults(:, :, MEPDGmodel) = importdata(DiscNameA);
    AnovaPmIXs (MEPDGmodel, :) = importdata (NameAixs);

    clear Forcorrholder;
    Forcorrholder = importdata(DiscNameF);
    ForcorrResults(:, :, MEPDGmodel) = Forcorrholder;
    clear Forcorrholder;
    Forcorrholder = importdata (NameFixs);
    ForcorrPmIXs (MEPDGmodel, :) = Forcorrholder (MEPDGmodel, :);

    clear SLFholder;
    SLFholder = importdata (DiscNameS);
    SLFsResults(:, :, MEPDGmodel) = SLFholder;
    clear SLFholder;
    SLFholder = importdata (NameSixs);
```

```

        SLFPmIXs(MEPDGmodel,:) = SLFholder(MEPDGmodel,:);
end

%Report Parameter Names according to Rankings
clear ForcorrResultsParams SLFResultsParams AnovaResultsParams
for i = 1:size(ForcorrPmIXs,1)
    for j = 1:size(ForcorrPmIXs,2)
        AnovaResultsParams(i,j) = RGP_pntnames(1, AnovaPmIXs(i,j));
        ForcorrResultsParams(i,j) = RGP_pntnames(1, ForcorrPmIXs(i,j));
        SLFResultsParams(i,j) = RGP_pntnames(1, SLFPmIXs(i,j));
    end
end
end
%%
=====
=
%Choose Best Parameters for each model
% close all; clear YpF;
% Xp = (1:size(AnovaResults,1)); Xp(1) = [];
% for MEPDGmodel = 1:num_Models
%     YpF(:,MEPDGmodel) = diff(ForcorrResults(:,2,MEPDGmodel));
%     Opt_Nd(:,MEPDGmodel) = interp1(Xp, YpF(:,MEPDGmodel), 32);
% end
% Opt_Nd
% zz(1:33.9,1) = 4; zz(34:52,1) = -16;
% figure
% plot([YpF, zz])
% title('Reduction in Ave % Error vs. Nd')

%Plot Only Adj. Rsquared for All Modes
Xn = (1:size(AnovaResults,1));
figure
for MEPDGmodel = 1:num_Models

    subplot(2,3,MEPDGmodel)
    plot(Xn, AnovaResults(:,5,MEPDGmodel), 'r', Xn,
ForcorrResults(:,5,MEPDGmodel), 'b', Xn, SLFsResults(:,5,MEPDGmodel),
'k');
    %[Discrepancy Term, Ave %Error, Stdev %Error, Rsqd, Adj Rsqd];

    suptitle('Adjusted Rsqd vs Quantity of Training Point Parameters');

    if MEPDGmodel <= 1
        title('Terminal IRI')

    elseif MEPDGmodel <= 2 && MEPDGmodel > 1
        title('Total Permanent Deformation')

    elseif MEPDGmodel <= 3 && MEPDGmodel > 2
        title('AC Bottom Up Cracking')

    elseif MEPDGmodel <= 4 && MEPDGmodel > 3
        title('AC Top Down Cracking')

    elseif MEPDGmodel >= 5
        title('AC Permanent Deformation')

```

```

        xlabel('Number of Training Point Parameters')
        legend('Anova Process', 'Correlation Matrix', 'GP Scale Length
Factors', 'Location', 'EastOutside');
    end
end

%locate point where AR2 > 0.9
for MEPDGmodel = 1:num_Models
    ARsANOVA(:,MEPDGmodel) = AnovaResults(:,5,MEPDGmodel);
    ARsCorr(:,MEPDGmodel) = ForccorrResults(:,5,MEPDGmodel);
    ARsSLFs(:,MEPDGmodel) = SLFsResults(:,5,MEPDGmodel);
end
for MEPDGmodel = 1:num_Models
    for i = 1:max(Xn)
        if ARsANOVA(i,MEPDGmodel) < 0.9
            ARsANOVA(i,MEPDGmodel) = 0;
        end
        if ARsCorr(i,MEPDGmodel) < 0.9
            ARsCorr(i,MEPDGmodel) = 0;
        end
        if ARsSLFs(i,MEPDGmodel) < 0.9
            ARsSLFs(i,MEPDGmodel) = 0;
        end
    end
end

%Search only in Correlation Matrix Method
BestQTPind = ones(1,num_Models);
for MEPDGmodel = 1:num_Models
    BestQTPind(1,MEPDGmodel) = find(ARsCorr(:,MEPDGmodel),1,'first')
end
clear BestQTP1corr BestQTP2corr BestQTP3corr BestQTP4corr BestQTP5corr
BestQTP1corr = ForccorrResultsParams(1,(1:BestQTPind(1,1)))';
BestQTP2corr = ForccorrResultsParams(2,(1:BestQTPind(1,2)))';
BestQTP3corr = ForccorrResultsParams(3,(1:BestQTPind(1,3)))';
BestQTP4corr = ForccorrResultsParams(4,(1:BestQTPind(1,4)))';
BestQTP5corr = ForccorrResultsParams(5,(1:BestQTPind(1,5)))';

BestQTPcorrUnique = unique([BestQTP1corr; BestQTP2corr; BestQTP3corr;
BestQTP4corr; BestQTP5corr]);

for MEPDGmodel = 1:num_Models
    figure

        plot(Xn, AnovaResults(:,5,MEPDGmodel), 'r', Xn,
ForccorrResults(:,5,MEPDGmodel), 'b', Xn, SLFsResults(:,5,MEPDGmodel),
'k', [min(Xn), max(Xn)], [0.9, 0.9], '--');
        %[Discrepancy Term, Ave %Error, Stdev %Error, Rsqd, Adj Rsqd];
        xlabel('Number of Training Point Parameters', 'FontSize',
18, 'FontName', 'Times New Roman')
        ylabel('Adjusted R-squared', 'FontSize', 18, 'FontName', 'Times New
Roman')
        legend('Anova Process', 'Correlation Matrix', 'GP Scale Length
Factors', 'Location', 'SouthEast');
        if MEPDGmodel <= 1

```

```

        title('Terminal IRI', 'FontSize', 18,'FontName','Times New
Roman')

        elseif MEPDGmodel <= 2 && MEPDGmodel > 1
            title('Total Permanent Deformation', 'FontSize',
18,'FontName','Times New Roman')

        elseif MEPDGmodel <= 3 && MEPDGmodel > 2
            title('AC Bottom Up Cracking', 'FontSize', 18,'FontName','Times
New Roman')

        elseif MEPDGmodel <= 4 && MEPDGmodel > 3
            title('AC Top Down Cracking', 'FontSize', 18,'FontName','Times
New Roman')

        elseif MEPDGmodel >= 5
            title('AC Permanent Deformation', 'FontSize',
18,'FontName','Times New Roman')
        end
    end
end

```

```

%%
=====
=
%Plot Results: Discrepancy(SSR)
for MEPDGmodel = 1:num_Models
    figure
    for i = 1:size(AnovaResults,2)
        Xn = (1:size(AnovaResults,1));
        subplot(2,3,i)
        plot(Xn, AnovaResults(:,i,MEPDGmodel), 'red', Xn,
ForcorrResults(:,i,MEPDGmodel), 'green', Xn,
SLFsResults(:,i,MEPDGmodel), 'blue');
        %[Discrepancy Term, Ave %Error, Stdev %Error, Rsqd, Adj Rsqd];
        if i == 1
            title('SSR: Want Minimum');
        elseif i == 2
            title('Ave. % Error: Want Minimum');
        elseif i == 3
            title('% Error Std Dev: Want = 0');
        elseif i == 4
            title('Ave Rsqd: Want = 1');
        else
            title('Adjusted Rsqd: Want = 1');
            legend('Anova Process', 'Correlation Matrix', 'GP Scale Length
Factors', 'Location', 'EastOutside');
        end

        if MEPDGmodel <= 1
            suptitle('Terminal IRI')

            elseif MEPDGmodel <= 2 && MEPDGmodel > 1
                suptitle('Total Permanent Deformation')

```

```
elseif MEPDGmodel <= 3 && MEPDGmodel > 2
    suptitle('AC Bottom Up Cracking')

elseif MEPDGmodel <= 4 && MEPDGmodel > 3
    suptitle('AC Top Down Cracking')

elseif MEPDGmodel >= 5
    suptitle('AC Permanent Deformation')
end
end
end
```

Chapter V

Calculation of Uncertainty Distribution Parameters for MEPDG and GP Models

Evaluation for Family of CDF Plots

Calculation of Contribution to Overall Variance

Evaluation for Contour Plots

Reliability Analysis Calculation

```
clear all; clc; close all;

training_pointsCOMPLETE;
r=size(train_pntsCOMPLETE);
s=size(train_valsCOMPLETE);
MEPDGPoints = train_valsCOMPLETE;

%Run only some MEPDG models
num_Models = 5;

%Evaluate each training point as a potential test point to produce
%"Optimal" GP model
for num_point = 1:r(1)
    num_point
    num_test = 1;
    num_train = r(1) - num_test;

    %Define Training Point Matricies

    %zscore returns "centered and scaled" version of the training
points
    %(inputs), as well as the mean and std. var.
    [train_pntsFULL,train_mean,train_std]=zscore(train_pntsCOMPLETE);

    train_pnts = removerows(train_pntsFULL,num_point);
    train_vals = removerows(train_valsCOMPLETE,num_point);

    %Normalize training values (outputs)
    yoffset = mean(train_vals);
    for i=1:size(train_vals,1)
        for j=1:size(train_vals,2)
            train_vals0(i,j) = train_vals(i,j) - yoffset(1,j);
        end
    end
end
```

```

%Define Test Point Matricies
test_pnts = train_pntsFULL(num_point,:);
test_vals = train_valsCOMPLETE(num_point,:);

%Construct and Evaluate GP
for MEPDGmodel = 1:num_Models

    %Train Model
    nsams = size(train_pnts,1);
    ndims = size(train_pnts,2);
    theta0 = ones(1,ndims); lob = 0.01*ones(1,ndims); upb =
10*ones(1,ndims);
    [GPmodel(MEPDGmodel), GPModelInfo(MEPDGmodel)] =
dacefit(train_pnts, train_vals0(:,MEPDGmodel), @regpoly1, @corrGauss,
theta0, lob,upb);

    %Evaluate GP model for GP Uncertainty Quantification
    gptest(:,MEPDGmodel) =
predictor(test_pnts,GPmodel(MEPDGmodel));
    end

    for i=1:size(gptest,1)
        for MEPDGmodel=1:num_Models
            GPtest(i,MEPDGmodel) = gptest(i,MEPDGmodel) +
yoffset(1,MEPDGmodel);
        end
    end

    for i=1:size(gptest,1)
        for MEPDGmodel = 1:num_Models
            Residuals(num_point,MEPDGmodel) = (test_vals(i,MEPDGmodel)-
GPtest(i,MEPDGmodel));
            ResidualsSquared(num_point,MEPDGmodel) =
Residuals(i,MEPDGmodel)^2;
        end
    end
end

clear train_pnts train_vals test_pnts test_vals
clear yoffset train_vals0
clear GPmodel GPModelInfo gptest GPtest

%Utilize Residuals to determine "Best" GP; This method will not provide
%the optimal GP, but is a quick solution to finding one of the best
GP's.
%A better GP is possible; however, for this application, we only
require
%that the GP introduce minimum uncertainty
%Later, will verify that this GP contributes only a small amount of
%uncertainty compared to MEPDG and Input Parameters

%Sort Residuals

```



```

for j=1:num_Models
    [ResSort(:,j),ResInd(:,j)] = sort(abs(Residuals(:,j)), 'descend');
end

%Construct GP utilizing Residuals information to choose test/train
points
for MEPDGmodel=1:num_Models
    %Take bottom 10 (lowest residuals) as test points;
    %The remainder will be used as training points
    train_pnts =
removerows(train_pntsFULL, [ResInd(161:170,MEPDGmodel)]);
    train_vals =
removerows(train_valsCOMPLETE, [ResInd(161:170,MEPDGmodel)]);

    %Normalize training points
    yoffset = mean(train_vals);
    for i=1:size(train_vals,1)
        for j=1:size(train_vals,2)
            train_vals0(i,j) = train_vals(i,j) - yoffset(1,j);
        end
    end

    %Define Test Point Matricies
    test_pnts = removerows(train_pntsFULL, [ResInd(1:160,MEPDGmodel)]);
    test_vals =
removerows(train_valsCOMPLETE, [ResInd(1:160,MEPDGmodel)]);

    %Construct and Evaluate GP

    %Train Model
    nsams = size(train_pnts,1);
    ndims = size(train_pnts,2);
    theta0 = ones(1,ndims); lob = 0.01*ones(1,ndims); upb =
10*ones(1,ndims);
    [GPmodel(MEPDGmodel), GPModelInfo(MEPDGmodel)] =
dacefit(train_pnts, train_vals0(:,MEPDGmodel), @regpoly1, @corr Gauss,
theta0, lob,upb);

    %Evaluate GP model for GP Verification Process
    gpctest(:,MEPDGmodel) = predictor(test_pnts,GPmodel(MEPDGmodel));

    for i=1:size(gpctest,1)
        GPctest(i,MEPDGmodel) = gpctest(i,MEPDGmodel) +
yoffset(1,MEPDGmodel);
    end

    %Calculate Residuals for "Best" GP trained w/160 points:
Residuals160
    %This will be used for Uncertainty Quantification
    for i=1:size(gpctest,1)
        Residuals160(i,MEPDGmodel)=(test_vals(i,MEPDGmodel)-
GPctest(i,MEPDGmodel));

ResidualsSquared160(i,MEPDGmodel)=Residuals(i,MEPDGmodel)^2;

```

```

end
end

%% Uncertainty Quantification: MEPDG and GP
%Calculate Uncertainty due to "Best" GP
mean_Res = mean(Residuals160);
std_Res = std(Residuals160);
var_Res = var(Residuals160);

%Threshold Limits
ThresholdLimits = [172 2000 25 0.25 0.75];
%Calculate Uncertainty due to MEPDG
%Calculations involves MEPDG output and threshold values
for MEPDGmodel = 1:num_Models
    for i=1:size(train_valsCOMPLETE,1)
        MS(i,MEPDGmodel) = ThresholdLimits(1,MEPDGmodel) -
train_valsCOMPLETE(i,MEPDGmodel);
        Z(i,MEPDGmodel) = norminv(1-
(PercentFailuresMEPDG(i,MEPDGmodel)/100));
    end

    for i=1:size(train_valsCOMPLETE,1)
        Sigma(i,MEPDGmodel) = MS(i,MEPDGmodel)/Z(i,MEPDGmodel);
    end
    %Umepdg = Standard Deviation of MEPDG Uncertainty
    Umepdg(1,MEPDGmodel) = abs(mean(Sigma(:,MEPDGmodel)));
end
UncertaintyResults = [Umepdg',mean_Res',std_Res'];

%% Family of CDF Plots
%Evaluate GP for many points to incorporate input uncertainty

%MCS: Generate sample input vectors
N=10000;
AADTx = random('norm',1500,150,N,1);
HMATHICKX = random('norm',8,0.78,N,1);
GBTHICKX = random('norm',8,1.25,N,1);
EBCX = random('norm',0.1,0.01,N,1);
AVX = random('norm',0.085,0.0085,N,1);
ESUBGRADEX = random('norm',14500,1250,N,1);
KGBX = random('norm',40000,1750,N,1);
Ax(1:N,1) = 10.7709;

for i=1:N

Xs(i,:)=[AADTx(i),HMATHICKX(i),GBTHICKX(i),EBCX(i),AVX(i),ESUBGRADEX(i)
,KGBX(i),Ax(i)];
end

for i=1:N
    for j=1:size(Xs,2)
        XsS(i,j)=((Xs(i,j)-train_mean(1,j))./train_std(1,j));
    end
end
end

```

```

for MEPDGmodel = 1:num_Models
    %Evaluate GP model for Uncertainty Analysis
    gp_eval(:,MEPDGmodel) = predictor(XsS,GPmodel(MEPDGmodel));
end

for i=1:N
    for MEPDGmodel=1:num_Models
        GPeval(i,MEPDGmodel) = gp_eval(i,MEPDGmodel) +
yoffset(1,MEPDGmodel);
    end
end

for MEPDGmodel=1:num_Models

    %Input Parameter Only
    [cdf_G1(:,MEPDGmodel),xi_G1(:,MEPDGmodel)] =
ksdensity(GPeval(:,MEPDGmodel),'function','cdf');
    Reliability90G1(MEPDGmodel) =
interplq(cdf_G1(:,MEPDGmodel),xi_G1(:,MEPDGmodel),0.9);

    %Input Parameters + MEPDG
    for i=1:N
        R(i,MEPDGmodel) = random('norm',0,Umepdg(1,MEPDGmodel));
        G2(i,MEPDGmodel) = GPeval(i,MEPDGmodel) + R(i,MEPDGmodel);
    end

[cdf_G2(:,MEPDGmodel),xi_G2(:,MEPDGmodel)]=ksdensity(G2(:,MEPDGmodel),'
function','cdf');

Reliability90G2(MEPDGmodel)=interplq(cdf_G2(:,MEPDGmodel),xi_G2(:,MEPDG
model),0.9);

    %Input Parameters + MEPDG + GP
    for i=1:N
        R2(i,MEPDGmodel) =
random('norm',mean_Res(1,MEPDGmodel),std_Res(1,MEPDGmodel));
        G3(i,MEPDGmodel) = GPeval(i,MEPDGmodel) + R(i,MEPDGmodel) +
R2(i,MEPDGmodel);
    end

[cdf_G3(:,MEPDGmodel),xi_G3(:,MEPDGmodel)]=ksdensity(G3(:,MEPDGmodel),'
function','cdf');

Reliability90G3(MEPDGmodel)=interplq(cdf_G3(:,MEPDGmodel),xi_G3(:,MEPDG
model),0.9);

    %Plot Means + MEPDG + GP
    X_means = [1500 8 8 0.1 0.085 14500 40000 10.96142];

    for j=1:size(train_mean,2)
        X_meansS(1,j)=(X_means(1,j)-train_mean(1,j))./train_std(1,j);
    end

    for j=MEPDGmodel

```

```

        gp_meansS(j) = predictor(X_meansS,GPmodel(j));
    end

    GPmeansS(1,MEPDGmodel) = gp_meansS(1,MEPDGmodel) +
yoffset(MEPDGmodel);

    for i=1:N
        G4(i,MEPDGmodel) = GPmeansS(1,MEPDGmodel) + R(i,MEPDGmodel);
    end

[cdf_G4(:,MEPDGmodel),xi_G4(:,MEPDGmodel)]=ksdensity(G4(:,MEPDGmodel),'
function','cdf');

Reliability90G4(MEPDGmodel)=interp1q(cdf_G4(:,MEPDGmodel),xi_G4(:,MEPDG
model),0.9);

end
%Plot CDFs
for MEPDGmodel=1:num_Models
    figure

        clear MinX MaxX MinY MaxY
        MinX(MEPDGmodel) =
min(min(min(xi_G4(:,MEPDGmodel)),min(min(xi_G3(:,MEPDGmodel)),min(min(x
i_G2(:,MEPDGmodel)),min(xi_G1(:,MEPDGmodel))))),0);
        MaxX(MEPDGmodel) =
max(max(xi_G4(:,MEPDGmodel)),max(max(xi_G3(:,MEPDGmodel)),max(max(xi_G2
(:,MEPDGmodel)),max(xi_G1(:,MEPDGmodel)))));
        MinY =
min(min(min(cdf_G4(:,MEPDGmodel)),min(min(cdf_G3(:,MEPDGmodel)),min(min
(cdf_G2(:,MEPDGmodel)),min(cdf_G1(:,MEPDGmodel))))),0);
        MaxY =
max(max(cdf_G4(:,MEPDGmodel)),max(max(cdf_G3(:,MEPDGmodel)),max(max(cdf
_G2(:,MEPDGmodel)),max(cdf_G1(:,MEPDGmodel)))));

plot(xi_G1(:,MEPDGmodel),cdf_G1(:,MEPDGmodel),'r',xi_G2(:,MEPDGmodel),c
df_G2(:,MEPDGmodel),'o--g',...
        xi_G3(:,MEPDGmodel),cdf_G3(:,MEPDGmodel),'.-b',
xi_G4(:,MEPDGmodel),cdf_G4(:,MEPDGmodel),'+-m',...
        [MinX(MEPDGmodel) MaxX(MEPDGmodel)],[0.9 0.9],'--
k',[ThresholdLimits(MEPDGmodel) ThresholdLimits(MEPDGmodel)],[MinY
MaxY],'--k',...
        [Reliability90G1(MEPDGmodel) Reliability90G1(MEPDGmodel)],[MinY
MaxY],':k',[Reliability90G4(MEPDGmodel) Reliability90G4(MEPDGmodel)],
[MinY MaxY],':k');
        axis([MinX(MEPDGmodel) MaxX(MEPDGmodel) 0 1])
        set(gca,'FontSize',14,'FontName','Times New Roman');
        ylabel('Reliability (F(Dt(x)))','FontSize',18,'FontName','Times
New Roman');
        xlabel('Distress Value (Dt(x))','FontSize',18,'FontName','Times
New Roman');
        legend('Input Variability','Input + MEPDG','Input + MEPDG +
GP','Current MEPDG Method','Threshold Limits')
        if MEPDGmodel <= 1

```

```

        title('Terminal IRI', 'FontSize', 18, 'FontName', 'Times New
Roman')

        elseif MEPDGmodel <= 2 && MEPDGmodel > 1
            title('AC Surface Down Cracking', 'FontSize',
18, 'FontName', 'Times New Roman')

        elseif MEPDGmodel <= 3 && MEPDGmodel > 2
            title('AC Bottom Up Cracking', 'FontSize', 18, 'FontName', 'Times
New Roman')

        elseif MEPDGmodel <= 4 && MEPDGmodel > 3
            title('AC Permanent Deformation', 'FontSize',
18, 'FontName', 'Times New Roman')

        elseif MEPDGmodel >= 5
            title('Total Permanent Deformation', 'FontSize',
18, 'FontName', 'Times New Roman')
        end
    end
end

%% Contributions to Variance
for i=1:num_Models
    VarXs(1,i) = var(GPeval(:,i));
    VarMEPDG(1,i) = Umepdg(1,i) * Umepdg(1,i);
    VarGP(1,i) = var_Res(1,i);
end

for i=1:size(VarXs,2)
    TotalVariance(1,i) = VarXs(1,i) + VarMEPDG(1,i) + VarGP(1,i);
    InputParamVar(1,i) = VarXs(1,i) / TotalVariance(1,i);
    MEPDGVar(1,i) = VarMEPDG(1,i) / TotalVariance(1,i);
    GPVar(1,i) = VarGP(1,i) / TotalVariance(1,i);
end

%Values are given as percentages
PercentContributions = [InputParamVar'*100,MEPDGVar'*100,GPVar'*100];

%% Reliability Analysis

for i=1:N
    %Calculate Number of failures: Input Parameter Uncertainty Only
    %(GPeval)
    for j=1:size(GPeval,2)
        if GPeval(i,j)>ThresholdLimits(1,j);
            failure1(i,j) = 1;
        else failure1(i,j) = 0;
        end
    end
    %System Reliability: Input Parameter Uncertainty Only
    if sum(failure1(i,:))>0
        SystemFailure1(i,1) = 1;
    else SystemFailure1(i,1) = 0;
    end
end

```

```

%Calculate Number of failures: Input Parameters + MEPDG (G2)
for j=1:size(GPeval,2)
    if G2(i,j)>ThresholdLimits(1,j);
        failure2(i,j) = 1;
    else failure2(i,j) = 0;
    end
end
%System Reliability
if sum(failure2(i,:))>0
    SystemFailure2(i,1) = 1;
else SystemFailure2(i,1) = 0;
end

(G3) %Calculate Number of failures: Input Parameters + MEPDG + GP
for j=1:size(GPeval,2)
    if G3(i,j)>ThresholdLimits(1,j);
        failure3(i,j) = 1;
    else failure3(i,j) = 0;
    end
end
%System Reliability
if sum(failure3(i,:))>0
    SystemFailure3(i,1) = 1;
else SystemFailure3(i,1) = 0;
end

%Calculate Number of failures: MEPDG Estimate (G4)
for j=1:size(GPeval,2)
    if G4(i,j)>ThresholdLimits(1,j);
        failure4(i,j) = 1;
    else failure4(i,j) = 0;
    end
end
%System Reliability
if sum(failure4(i,:))>0
    SystemFailure4(i,1) = 1;
else SystemFailure4(i,1) = 0;
end

end

ProbabilityOfFailure1=[(sum(failure1))/N,(sum(SystemFailure1))/N];
ProbabilityOfFailure2=[(sum(failure2))/N,(sum(SystemFailure2))/N];
ProbabilityOfFailure3=[(sum(failure3))/N,(sum(SystemFailure3))/N];
ProbabilityOfFailure4=[(sum(failure4))/N,(sum(SystemFailure4))/N];
%Values given as percentages
POF_Table=[ProbabilityOfFailure1'*100,ProbabilityOfFailure2'*100,ProbabilityOfFailure3'*100,ProbabilityOfFailure4'*100];
%-----
----
%-----
----
%% Output for Plot for pdf and cdf plots (Response to Reviewer
comments)

```

```

clear pdf_MEPDG xi_MEPDG cdf_MEPDG xicdf_MEPDG pdf_Inputs xi_Inputs
pdf_GP xi_GP xi_InputsN xi_MEPDGN xi_GPN
[pdf_MEPDG(:,1),xi_MEPDG(:,1)] = ksdensity(R(:,1),'function','pdf');
figure
plot(xi_MEPDG,pdf_MEPDG); set(gca,'yticklabel',[]);
set(gca,'xticklabel',[]);

[cdf_MEPDG(:,1),xicdf_MEPDG(:,1)] = ksdensity(R(:,1),'function','cdf');
figure
plot(xicdf_MEPDG,cdf_MEPDG); set(gca,'yticklabel',[]);
set(gca,'xticklabel',[]);
%% Create Plots for Variance of Sources (Response to Reviewers
Comments)
[pdf_Inputs(:,1),xi_Inputs(:,1)] =
ksdensity(GPeval(:,1),'function','pdf');
[pdf_GP(:,1),xi_GP(:,1)]=ksdensity(R2(:,1),'function','pdf');
xi_InputsN = zscore(xi_Inputs);
xi_MEPDGN = zscore(xi_MEPDG);
xi_GPN = zscore(xi_GP);
pdf_InputsN = zscore(pdf_Inputs);
pdf_MEPDGN = zscore(pdf_MEPDG);
pdf_GPN = zscore(pdf_GP);
figure
plot(xi_InputsN(:,1), pdf_Inputs(:,1), 'r', xi_MEPDG(:,1),
pdf_MEPDG(:,1), '--b', xi_GP(:,1), pdf_GP(:,1), '--g');
%set(gca,'yticklabel',[]); set(gca,'xticklabel',[]);
legend('Inputs','MEPDG','GP')

```

```

clear all; clc; close all;
addpath('C:\Users\Jenny\Documents\Retherford-
Vanderbilt\Summer2010\TRB90\Matlab\dace');

```

```

training_pointsCOMPLETE;
num_Models = size(train_valsCOMPLETE,2);

```

```

r=size(train_pntsCOMPLETE);
s=size(train_valsCOMPLETE);
num_test = 10;
yy = randperm(r(1));
tstidx = sort(yy(1:num_test));
trnidx = setdiff([1:r],tstidx);

```

```

train_pnts=train_pntsCOMPLETE(trnidx,:);
train_vals=log(train_valsCOMPLETE(trnidx,:));

```

```

%Train Model
ndims = size(train_pnts,2);
theta0 = 1*ones(1,ndims);
lob = 0.01*ones(1,ndims);
upb = 100*ones(1,ndims);
clear GPmodel GPModelInfo
for MEPDGmodel = 1:num_Models
    MEPDGmodel
    [GPmodel(1, MEPDGmodel), GPModelInfo(1, MEPDGmodel)] =...
    dacefit(train_pnts, train_vals(:,MEPDGmodel),...

```

```

    @regpoly1, @corrgauss, theta0, lob, upb);
end

%% Predict with the GP models
%MCS: Generate 100 sample input vectors
N=500;
clear AADTsa HMATHICKSA GBTHICKSA EBCSA AVSA ESubgradesa Kgbsa
AADTsa = random('norm',1500,150,N,1);
HMATHICKSA = random('norm',8,0.78,N,1);
GBTHICKSA = random('norm',8,1.25,N,1);
EBCSA = random('norm',0.1,0.01,N,1);
AVSA = random('norm',0.085,0.0085,N,1);
ESubgradesa = random('norm',14500,1250,N,1);
Kgbsa = random('norm',40000,1750,N,1);

InputMeans = [1500, 8, 8, 0.1, 0.085, 14500, 40000, 10.7709];

%MEPDG evaluation at means: fx(mux)
MEPDG_valsMeans = [121.7, 334, 2.7, 0.28, 0.62];
Asa(1:N,1) = 10.7709;

for i=1:N

Xsa(i,:)=[AADTsa(i),HMATHICKSA(i),GBTHICKSA(1),EBCSA(1),AVSA(1),ESubgradesa(1),Kgbsa(1),Asa(1)];
end

close all; clear XsaS
for MEPDGmodel = 1:num_Models
    MEPDGmodel
    %Evaluate all Combinations of AADT and HMATHICKNESS
    clear GPsa_response
    XsaS(:,1) = sort(AADTsa(:,1));
    XsaS(:,2) = sort(HMATHICKSA(:,1));
    for i = 1:N
        for j = 1:N
            GPsa_response(j,i) =
predictor([XsaS(i,1),XsaS(j,2),InputMeans(1,3:8)],GPmodel(1,MEPDGmodel)
);
            end
        end
    end
    GPsa_response = exp(GPsa_response);
    %Create contour plots of GP
    clear C1 h1

    figure
    hold on
    [C1,h1]=contourf(XsaS(:,1),XsaS(:,2),GPsa_response());
    clabel(C1,h1,'FontSize',14,'FontName','Times New Roman');
    set(gca,'FontSize',14,'FontName','Times New Roman');
    % plot([InputMeans(1,1), InputMeans(1,1)], [min(Xsa(:,2)),
max(Xsa(:,2))], 'k', [min(Xsa(:,1)), max(Xsa(:,1))], [InputMeans(1,2),
InputMeans(1,2)], 'k','LineWidth',2);

```



```

        xlabel('Average Annual Daily Traffic', 'FontSize',
18, 'FontName', 'Times New Roman')
        ylabel('HMA Thickness (in.)', 'FontSize', 18, 'FontName', 'Times New
Roman')
        if MEPDGmodel <= 1
            title('Terminal IRI', 'FontSize', 18, 'FontName', 'Times New
Roman')

            elseif MEPDGmodel <= 2 && MEPDGmodel > 1
                title('AC Top Down Cracking', 'FontSize', 18, 'FontName', 'Times
New Roman')

            elseif MEPDGmodel <= 3 && MEPDGmodel > 2
                title('AC Bottom Up Cracking', 'FontSize', 18, 'FontName', 'Times
New Roman')

            elseif MEPDGmodel <= 4 && MEPDGmodel > 3
                title('AC Permanent Deformation', 'FontSize',
18, 'FontName', 'Times New Roman')

            elseif MEPDGmodel >= 5
                title('Total Permanent Deformation', 'FontSize',
18, 'FontName', 'Times New Roman')
            end
            hold off
        end
end

%-----
----
clear XsaS;
for MEPDGmodel = 1:num_Models
    MEPDGmodel
    %Evaluate all Combinations of HMAtickness and GBthicksa
    clear GPsa_response
    XsaS(:,1) = sort(HMAticknessa(:,1));
    XsaS(:,2) = sort(GBthicksa(:,1));
    for i = 1:N
        for j = 1:N
            GPsa_response(j,i) =
predictor([InputMeans(1,1), XsaS(i,1), XsaS(j,2), InputMeans(1,4:8)], GPmod
el(1,MEPDGmodel));
        end
    end
    GPsa_response = exp(GPsa_response);
    %Create contour plots of GP
    clear C1 h1

    figure
    [C1,h1]=contourf(XsaS(:,1),XsaS(:,2),GPsa_response());
    clabel(C1,h1);

    xlabel('HMA Thickness (in.)', 'FontSize', 18, 'FontName', 'Times New
Roman')

```

```

        ylabel('GB Thickness (in.)', 'FontSize', 18,'FontName','Times New
Roman')
        if MEPDGmodel <= 1
            title('Terminal IRI', 'FontSize', 18,'FontName','Times New
Roman')

            elseif MEPDGmodel <= 2 && MEPDGmodel > 1
                title('AC Top Down Cracking', 'FontSize', 18,'FontName','Times
New Roman')

            elseif MEPDGmodel <= 3 && MEPDGmodel > 2
                title('AC Bottom Up Cracking', 'FontSize', 18,'FontName','Times
New Roman')

            elseif MEPDGmodel <= 4 && MEPDGmodel > 3
                title('AC Permanent Deformation', 'FontSize',
18,'FontName','Times New Roman')

            elseif MEPDGmodel >= 5
                title('Total Permanent Deformation', 'FontSize',
18,'FontName','Times New Roman')
            end
        end

end

%-----
----
clear XsaS;
for MEPDGmodel = 1:num_Models
    MEPDGmodel
    %Evaluate all Combinations of HMAtickness and EBCsa
    clear GPsa_response
    XsaS(:,1) = sort(HMAticknessa(:,1));
    XsaS(:,2) = sort(EBCsa(:,1));
    for i = 1:N
        for j = 1:N
            GPsa_response(j,i) =
predictor([InputMeans(1,1),XsaS(i,1),InputMeans(1,3),XsaS(j,2),InputMea
ns(1,5:8)],GPmodel(1,MEPDGmodel));
        end
    end
    GPsa_response = exp(GPsa_response);
    %Create contour plots of GP
    clear C1 h1

    figure
    [C1,h1]=contourf(XsaS(:,1),XsaS(:,2),GPsa_response());
    clabel(C1,h1);
    xlabel('HMA Thickness (in.)', 'FontSize', 18,'FontName','Times New
Roman')
    ylabel('Effective Binder Content (in.)', 'FontSize',
18,'FontName','Times New Roman')
    if MEPDGmodel <= 1
        title('Terminal IRI', 'FontSize', 18,'FontName','Times New
Roman')
    end
end

```

```

elseif MEPDGmodel <= 2 && MEPDGmodel > 1
    title('AC Top Down Cracking', 'FontSize', 18, 'FontName', 'Times
New Roman')

elseif MEPDGmodel <= 3 && MEPDGmodel > 2
    title('AC Bottom Up Cracking', 'FontSize', 18, 'FontName', 'Times
New Roman')

elseif MEPDGmodel <= 4 && MEPDGmodel > 3
    title('AC Permanent Deformation', 'FontSize',
18, 'FontName', 'Times New Roman')

elseif MEPDGmodel >= 5
    title('Total Permanent Deformation', 'FontSize',
18, 'FontName', 'Times New Roman')
end
end

%-----
----
clear XsaS;
for MEPDGmodel = 1:num_Models
    MEPDGmodel
    %Evaluate all Combinations of HMAthickness and AVsa
    clear GPsa_response
    XsaS(:,1) = sort(HMAthickness(:,1));
    XsaS(:,2) = sort(AVsa(:,1));
    for i = 1:N
        for j = 1:N
            GPsa_response(j,i) =
predictor([InputMeans(1,1), XsaS(i,1), InputMeans(1,3:4), XsaS(j,2), InputM
eans(1,6:8)], GPmodel(1,MEPDGmodel));
            end
        end
        GPsa_response = exp(GPsa_response);
        %Create contour plots of GP
        clear C1 h1

        figure
        [C1,h1]=contourf(XsaS(:,1),XsaS(:,2),GPsa_response());
        xlabel('HMA Thickness (in.)', 'FontSize', 18, 'FontName', 'Times New
Roman')
        ylabel('Air Voids (in.)', 'FontSize', 18, 'FontName', 'Times New
Roman')
        if MEPDGmodel <= 1
            title('Terminal IRI', 'FontSize', 18, 'FontName', 'Times New
Roman')

        elseif MEPDGmodel <= 2 && MEPDGmodel > 1
            title('AC Top Down Cracking', 'FontSize', 18, 'FontName', 'Times
New Roman')

```

```

elseif MEPDGmodel <= 3 && MEPDGmodel > 2
    title('AC Bottom Up Cracking', 'FontSize', 18,'FontName','Times
New Roman')

elseif MEPDGmodel <= 4 && MEPDGmodel > 3
    title('AC Permanent Deformation', 'FontSize',
18,'FontName','Times New Roman')

elseif MEPDGmodel >= 5
    title('Total Permanent Deformation', 'FontSize',
18,'FontName','Times New Roman')
end
end

%-----
----
clear XsaS;
for MEPDGmodel = 1:num_Models
    MEPDGmodel
    %Evaluate all Combinations of HMAtickness and Esubgrade
clear GPsa_response
XsaS(:,1) = sort(HMAticknessa(:,1));
XsaS(:,2) = sort(Esubgradesa(:,1));
for i = 1:N
    for j = 1:N
        GPsa_response(j,i) =
predictor([InputMeans(1,1),XsaS(i,1),InputMeans(1,3:5),XsaS(j,2),InputM
eans(1,7:8)],GPmodel(1,MEPDGmodel));
        end
        i
    end
    GPsa_response = exp(GPsa_response);
    %Create contour plots of GP
clear C1 h1

figure
[C1,h1]=contourf(XsaS(:,1),XsaS(:,2),GPsa_response());
clabel(C1,h1);
xlabel('HMA Thickness (in.)', 'FontSize', 18,'FontName','Times New
Roman')
ylabel('Subgrade Modulus(in.)', 'FontSize', 18,'FontName','Times
New Roman')
if MEPDGmodel <= 1
    title('Terminal IRI', 'FontSize', 18,'FontName','Times New
Roman')

elseif MEPDGmodel <= 2 && MEPDGmodel > 1
    title('AC Top Down Cracking', 'FontSize', 18,'FontName','Times
New Roman')

elseif MEPDGmodel <= 3 && MEPDGmodel > 2
    title('AC Bottom Up Cracking', 'FontSize', 18,'FontName','Times
New Roman')

elseif MEPDGmodel <= 4 && MEPDGmodel > 3

```

```

        title('AC Permanent Deformation', 'FontSize',
18,'FontName','Times New Roman')

        elseif MEPDGmodel >= 5
            title('Total Permanent Deformation', 'FontSize',
18,'FontName','Times New Roman')
        end
    end
end

%-----
----
clear XsaS;
for MEPDGmodel = 1:num_Models
    MEPDGmodel
    %Evaluate all Combinations of HMAtickness and Kgbsa
    clear GPsa_response
    XsaS(:,1) = sort(HMAticknessa(:,1));
    XsaS(:,2) = sort(Kgbsa(:,1));
    for i = 1:N
        for j = 1:N
            GPsa_response(j,i) =
predictor([InputMeans(1,1),XsaS(i,1),InputMeans(1,3:6),XsaS(j,2),InputM
eans(1,8)],GPmodel(1,MEPDGmodel));
            end
        i
    end
    GPsa_response = exp(GPsa_response);
    %Create contour plots of GP
    clear C1 h1

    figure
    [C1,h1]=contourf(XsaS(:,1),XsaS(:,2),GPsa_response());
    clabel(C1,h1);
    xlabel('HMA Thickness (in.)', 'FontSize', 18,'FontName','Times New
Roman')
    ylabel('GB Modulus (in.)', 'FontSize', 18,'FontName','Times New
Roman')
    if MEPDGmodel <= 1
        title('Terminal IRI', 'FontSize', 18,'FontName','Times New
Roman')

        elseif MEPDGmodel <= 2 && MEPDGmodel > 1
            title('AC Top Down Cracking', 'FontSize', 18,'FontName','Times
New Roman')

            elseif MEPDGmodel <= 3 && MEPDGmodel > 2
                title('AC Bottom Up Cracking', 'FontSize', 18,'FontName','Times
New Roman')

            elseif MEPDGmodel <= 4 && MEPDGmodel > 3
                title('AC Permanent Deformation', 'FontSize',
18,'FontName','Times New Roman')

            elseif MEPDGmodel >= 5

```

```
        title('Total Permanent Deformation', 'FontSize',  
18, 'FontName', 'Times New Roman')  
    end  
end
```

Chapter VI

CDF Plots for AMV and MCS Results

```
clear all; clc; close all;
addpath('C:\Users\Jenny\Documents\Retherford-
Vanderbilt\Spring2012\OptimizationProblems\Matlab\accre\accreInput');
addpath('C:\Users\Jenny\Documents\Retherford-
Vanderbilt\Spring2012\OptimizationProblems\Matlab\accre\accreOutput');

%% Import Data
Inputs = importdata('Year20SetUp.mat');
FF = 'Rev20Full';
%-----
----
%Initializing Data

ParamNamesTrain = Inputs.ParamNamesTrain;
RGP_pntsYear = Inputs.RGP_pntsYear;
RGP_valsYear = exp(Inputs.RGP_valsYear);

%Stats
num_Models = size(RGP_valsYear,2);

%-----
----
%Data From ACCRE

% Name1 = strcat(FF, 'VV_GPsize.txt');
% Name2 = strcat(FF, 'VV_varModels.txt');
% Name3 = strcat(FF, 'VV_holder.txt');
% Name4 = strcat(FF, 'MaxCandVar.txt');
% Name5 = strcat(FF, 'Init_pntsTrain.txt');
% Name6 = strcat(FF, 'Init_valsTrain.txt');
% Name7 = strcat(FF, 'RGP_pntsTrain.txt');
% Name8 = strcat(FF, 'RGP_valsTrain.txt');

% VV_GPsize      = importdata(Name1);
% VV_varModels   = importdata(Name2);
% VV_holder      = importdata(Name3);
% MaxCandVar     = importdata(Name4);
% Init_pntsTrain = importdata(Name5);
% Init_valsTrain = importdata(Name6);
% RGP_pntsTrain  = importdata(Name7);
% RGP_valsTrain  = importdata(Name8);

%-----
----

mean_Res = [-0.3081, -5.6077e-4, 0.000371, 20.6545, -7.7039e-4];
std_Res  = [0.8445, 0.0028, 0.0001769, 105.6707, 0.0013];
Umepdgd = [34.022, 0.1223, 0.094563, 1821.3, 0.099];
```

```

%Threshold Limits
% DistressModes =
% ['TermIRI', 'RuttingTotal', 'ACBottomUp', 'ACTopDown', 'RuttingAC'];
% ThresholdLimits = [250, 0.75, 0.25, 2000, 0.50];

%% Build GP
%From plots, typically need only Nopt training points to have 'good' GP
%which TPs:
Ind = randperm(size(RGP_pntsYear,1));
Nopt = 500;
gp_pntsTrain = RGP_pntsYear(Ind(1:Nopt), [1:4, 17, 18, 20]);
gp_NamesTrain = ParamNamesTrain(1, [1:4, 17, 18, 20]);
gp_valsTrain = RGP_valsYear(Ind(1:Nopt), :);

TPmeans = mean(gp_pntsTrain);
TPstdev = std(gp_pntsTrain);
TPlb = min(gp_pntsTrain);
TPub = max(gp_pntsTrain);

TPstats = [TPmeans; TPstdev; TPlb; TPub];

ndims = size(gp_pntsTrain,2);
clear GPmodel GPModelInfo
theta0 = 1*ones(1,ndims);
lob = 0.1*ones(1,ndims);
upb = 100*ones(1,ndims);
for MEPDGmodel = 1:num_Models
    MEPDGmodel
    [GPmodel(1, MEPDGmodel), GPModelInfo(1, MEPDGmodel)] =...
    dacefit(gp_pntsTrain, gp_valsTrain(:,MEPDGmodel),...
    @regpoly1, @corrGauss, theta0, lob, upb);
end

%Evaluate Mean
N_opt = 100;
Opt_pntsMeans = TPmeans;
clear MCSpointsOpt
for i = 1:ndims
    MCSpointsOpt(:,i) = random('norm', Opt_pntsMeans(1,i),
    TPstdev(1,i), [N_opt,1]);
end

clear gp_evalMCS
for MEPDGmodel = 1:num_Models
    %Evaluate GP model for Uncertainty Analysis
    gp_evalMCS(:,MEPDGmodel) =
    predictor(MCSpointsOpt,GPmodel(MEPDGmodel));
end

%Transform from log space
% gp_evalMCS = exp(gp_evalMCS);

for MEPDGmodel=1:num_Models
    %Input Variability Only

```



```

[cdf_G1(:,MEPDGmodel),xi_G1(:,MEPDGmodel)]=ksdensity(gp_evalMCS(:,MEPDG
model),'function','cdf');
    %MEPDG Uncertainty
    R(:,MEPDGmodel) = random('norm',0,Umepdg(1,MEPDGmodel), [N_opt,1]);
    %GP Uncertainty
    R2(:,MEPDGmodel) =
random('norm',mean_Res(1,MEPDGmodel),std_Res(1,MEPDGmodel), [N_opt,1]);

    %Input Parameters + GP
    for i=1:N_opt

        G3(i,MEPDGmodel) = gp_evalMCS(i,MEPDGmodel) + R2(i,MEPDGmodel)
+ R(i,MEPDGmodel);
        end
        [cdf_MCS(:,MEPDGmodel),xi_MCS(:,MEPDGmodel)] =
ksdensity(gp_evalMCS(:,MEPDGmodel),'function','cdf');
        [cdf_G3(:,MEPDGmodel),xi_G3(:,MEPDGmodel)] =
ksdensity(G3(:,MEPDGmodel),'function','cdf');
    end

%%=====
====
%% Darwin-ME output
BetaVector = [1, 2, 3, -1, -2, -3];

Probf = normcdf(BetaVector);

% Direction Cosines
Alphas = [
-0.2263 -0.1567 -0.1878 -0.2201 -0.2412 -0.1753 -0.1915
-0.1943 0.2914 0.0000 -0.1943 -0.1943 0.1943 0.0000
-0.0208 0.1508 0.0312 0.0416 -0.0624 0.0156 0.0416
-0.0450 0.4239 0.1095 0.0727 -0.1866 -0.1117 0.1222
-0.1843 0.1229 -0.0614 -0.1843 -0.2457 0.0000 -0.0614
];

AMVstats = [
1500 8 8 0.1 0.08 14500 40000
150 0.8 0.8 0.01 0.008 1450 4000];

for i = 1:6
    for j = 1:7
        Darwin_pntsTermIRI(i,j) = AMVstats(1,j) - Alphas(1,j) *
BetaVector(1,i) * AMVstats(2,j);
        Darwin_pntsTotPD(i,j) = AMVstats(1,j) - Alphas(2,j) *
BetaVector(1,i) * AMVstats(2,j);
        Darwin_pntsACBU(i,j) = AMVstats(1,j) - Alphas(3,j) *
BetaVector(1,i) * AMVstats(2,j);
        Darwin_pntsACTD(i,j) = AMVstats(1,j) - Alphas(4,j) *
BetaVector(1,i) * AMVstats(2,j);
        Darwin_pntsACPD(i,j) = AMVstats(1,j) - Alphas(5,j) *
BetaVector(1,i) * AMVstats(2,j);
    end
end

```

```

end
end

gp_evalDarwin(:,1) = predictor(Darwin_pntsTermIRI(:,1:7),GPmodel(1,1));
gp_evalDarwin(:,2) = predictor(Darwin_pntsTotPD(:,1:7),GPmodel(1,2));
gp_evalDarwin(:,3) = predictor(Darwin_pntsACBU(:,1:7),GPmodel(1,3));
gp_evalDarwin(:,4) = predictor(Darwin_pntsACTD(:,1:7),GPmodel(1,4));
gp_evalDarwin(:,5) = predictor(Darwin_pntsACPD(:,1:7),GPmodel(1,5));

% gp_evalDarwin = exp(gp_evalDarwin);

for MEPDGmodel = 1:num_Models
    for i=1:6
        G4(i,MEPDGmodel) = gp_evalDarwin(i,MEPDGmodel) +
R2(i,MEPDGmodel) + R(i,MEPDGmodel);
    end
    [cdf_AMV(:,MEPDGmodel),xi_AMV(:,MEPDGmodel)] =
ksdensity(gp_evalDarwin(:,MEPDGmodel),'function','cdf');
    [cdf_G4(:,MEPDGmodel),xi_G4(:,MEPDGmodel)] =
ksdensity(G4(:,MEPDGmodel),'function','cdf');
end
%%=====
====
%% Plot CDFs
MinRel = 0.9; %Target threshold; not included in opt. routine
close all;
for MEPDGmodel = 1:num_Models
    figure

    clear MinX MaxX MinY MaxY
    MinX(MEPDGmodel) = min(min(min(min(xi_G1(:,MEPDGmodel)),
min(min(xi_G3(:,MEPDGmodel)), min(xi_G4(:,MEPDGmodel))))), 0);
    MaxX(MEPDGmodel) = max(max(max(max(xi_G1(:,MEPDGmodel)),max(
max(xi_G3(:,MEPDGmodel)), max(xi_G4(:,MEPDGmodel))))), 0);
    MinY = 0;
    MaxY = 1;

    plot(xi_G3(:,MEPDGmodel),cdf_G3(:,MEPDGmodel),'.-r',... %CDF Input
Variability + GP + MEPDG
        xi_G4(:,MEPDGmodel),cdf_G4(:,MEPDGmodel),'.-b'); %AMV

    axis([MinX(MEPDGmodel) MaxX(MEPDGmodel) 0 1])
    set(gca,'FontSize', 14,'FontName','Times New Roman');
    ylabel('Probability of Failure (F(Dt(x)))','FontSize',
18,'FontName','Times New Roman');
    xlabel('Distress Value (Dt(x))','FontSize', 18,'FontName','Times
New Roman');
    legend('Input Variability + GP + MEPDG','AMV + GP + MEPDG')
    % DistressModes = ['TermIRI', 'RuttingTotal', 'ACBottomUp',
'ACTopDown', 'RuttingAC'];
    if MEPDGmodel <= 1
        title('Terminal IRI','FontSize', 18,'FontName','Times New
Roman')

```

```

elseif MEPDGmodel <= 2 && MEPDGmodel > 1
    title('Total Permanent Deformation','FontSize',
18,'FontName','Times New Roman')

elseif MEPDGmodel <= 3 && MEPDGmodel > 2
    title('AC Bottom Up Cracking','FontSize', 18,'FontName','Times
New Roman')

elseif MEPDGmodel <= 4 && MEPDGmodel > 3
    title('AC Top Down Cracking','FontSize', 18,'FontName','Times
New Roman')

elseif MEPDGmodel >= 5
    title('AC Permanent Deformation','FontSize',
18,'FontName','Times New Roman')
end
end

% for MEPDGmodel = 1:num_Models
%     figure
%
%     clear MinX MaxX MinY MaxY
%     MinX(MEPDGmodel) = min(min(min(min(xi_G1(:,MEPDGmodel)),
min(min(xi_G3(:,MEPDGmodel)), min(xi_G4(:,MEPDGmodel))))) , 0);
%     MaxX(MEPDGmodel) = max(max(max(max(xi_G1(:,MEPDGmodel)),max(
max(xi_G3(:,MEPDGmodel)), max(xi_G4(:,MEPDGmodel))))) , 0);
%     MinY = 0;
%     MaxY = 1;
%
%
%
%     plot(xi_MCS(:,MEPDGmodel),cdf_MCS(:,MEPDGmodel),'.-r',... %CDF
Input Variability + GP + MEPDG
%         xi_AMV(:,MEPDGmodel),cdf_AMV(:,MEPDGmodel),'.-b'); %AMV
%
%     axis([MinX(MEPDGmodel) MaxX(MEPDGmodel) 0 1])
%     ylabel('Probability of Failure (F(Dt(x)))');
%     xlabel('Distress Value (Dt(x))');
%     legend('Input Variability + GP + MEPDG','AMV + GP + MEPDG')
%     % DistressModes = ['TermIRI', 'RuttingTotal', 'ACBottomUp',
'ACTopDown', 'RuttingAC'];
%     if MEPDGmodel <= 1
%         title('Terminal IRI')
%
%     elseif MEPDGmodel <= 2 && MEPDGmodel > 1
%         title('Total Permanent Deformation')
%
%     elseif MEPDGmodel <= 3 && MEPDGmodel > 2
%         title('AC Bottom Up Cracking')
%
%     elseif MEPDGmodel <= 4 && MEPDGmodel > 3
%         title('AC Top Down Cracking')
%
%     elseif MEPDGmodel >= 5
%         title('AC Permanent Deformation')
%     end
% end

```

```

%% %% System Reliability
%%
%% %generate random points from AMV cdf
%% Nmcs = 100;
%% yy = randperm(100);
%% for i = 1:Nmcs
%%     for MEPDGmodel = 1:num_Models
%%         AMV_relpnts(i,MEPDGmodel) =
random('unif',min(xi_AMV(:,MEPDGmodel)), max(xi_AMV(:,MEPDGmodel)));
%%         end
%%     end
%%
%% %generate MCSpoints
%% for MEPDGmodel=1:num_Models
%%     clear mcsIndsA mcsIndsR
%%     mcsIndsR =
(random('normal',mean(cdf_G3(:,MEPDGmodel)), std(cdf_G3(:,MEPDGmodel)),
[Nmcs,1]));
%%     mcsIndsA =
(random('normal',mean(cdf_G4(:,MEPDGmodel)), std(cdf_G4(:,MEPDGmodel)),
[Nmcs,1]));
%%     for i = 1:Nmcs
%%         if mcsIndsR(i,1) < 0;
%%             mcsIndsR(i,1) = min(cdf_G3(i,MEPDGmodel));
%%         end
%%         if mcsIndsR(i,1) > 1;
%%             mcsIndsR(i,1) = max(cdf_G3(i,MEPDGmodel));
%%         end
%%         if mcsIndsA(i,1) < 0;
%%             mcsIndsA(i,1) = min(cdf_G4(i,MEPDGmodel));
%%         end
%%         if mcsIndsA(i,1) > 1;
%%             mcsIndsA(i,1) = max(cdf_G4(i,MEPDGmodel));
%%         end
%%     end
%%     for i=1:Nmcs
%%         RGPpreliability(i,MEPDGmodel) = interp1q(cdf_G3(:,MEPDGmodel),
xi_G3(:,MEPDGmodel), mcsIndsR(i,1));
%%         AMVreliability(i,MEPDGmodel) = interp1q(cdf_G4(:,MEPDGmodel),
xi_G4(:,MEPDGmodel), mcsIndsA(i,1));
%%     end
%% end
%%
%% %% System Reliability
%% -----
%%
%% % CASE 1
%% ThresholdLimits = [250, 0.75, 0.30, 2000, 0.50];
%% for MEPDGmodel = 1:num_Models
%%     %Calculate Reliability @ Threshold Limit
%%     ProbrGP(:,MEPDGmodel) = 1 - interp1q(xi_G3(:,MEPDGmodel),
cdf_G3(:,MEPDGmodel), ThresholdLimits(1,MEPDGmodel));
%%     if max(xi_G3(:,MEPDGmodel)) < ThresholdLimits(1,MEPDGmodel)
%%         ProbrGP(:,MEPDGmodel) = 0;
%%     end

```

```

%     if min(xi_G3(:,MEPDGmodel)) > ThresholdLimits(1,MEPDGmodel)
%         ProbrGP(:,MEPDGmodel) = 1;
%     end
%     ProbAMV(:,MEPDGmodel) = 1 - interp1q(xi_G4(:,MEPDGmodel),
cdf_G4(:,MEPDGmodel), ThresholdLimits(1,MEPDGmodel));
%     if max(xi_G4(:,MEPDGmodel)) < ThresholdLimits(1,MEPDGmodel)
%         ProbAMV(:,MEPDGmodel) = 0;
%     end
%     if min(xi_G4(:,MEPDGmodel)) > ThresholdLimits(1,MEPDGmodel)
%         ProbAMV(:,MEPDGmodel) = 1;
%     end
% end
%
%
% for MEPDGmodel = 1:num_Models
% for i = 1:size(xi_G4,1)
%     %Calculate Number of Failures
%     if AMVreliability(i,MEPDGmodel) > ThresholdLimits(1,MEPDGmodel)
%         SystemFailAMV(i,MEPDGmodel) = 1;
%     else SystemFailAMV(i,MEPDGmodel) = 0;
%     end
%     %System Reliability
%     if sum(SystemFailAMV(i,:))>0
%         SystemFailure1(i,1) = 1;
%     else SystemFailure1(i,1) = 0;
%     end
%     if RGPpreliability(i,MEPDGmodel) > ThresholdLimits(1,MEPDGmodel)
%         SystemFailRGP(i,MEPDGmodel) = 1;
%     else SystemFailRGP(i,MEPDGmodel) = 0;
%     end
%     %System Reliability
%     if sum(SystemFailRGP(i,:))>0
%         SystemFailure2(i,1) = 1;
%     else SystemFailure2(i,1) = 0;
%     end
% end
% end
%
% AMVsysRel = sum(SystemFailure1) / Nmcs;
% RGPsysRel = sum(SystemFailure2) / Nmcs;
% [ProbAMV',ProbrGP']
% [AMVsysRel', RGPsysRel']
%
%-----
-----
% % CASE 2
% ThresholdLimits = [200, 0.75, 0.30, 2000, 0.50];
% for MEPDGmodel = 1:num_Models
%     %Calculate Reliability @ Threshold Limit
%     ProbrGP(:,MEPDGmodel) = 1 - interp1q(xi_G3(:,MEPDGmodel),
cdf_G3(:,MEPDGmodel), ThresholdLimits(1,MEPDGmodel));
%     if max(xi_G3(:,MEPDGmodel)) < ThresholdLimits(1,MEPDGmodel)
%         ProbrGP(:,MEPDGmodel) = 0;
%     end
%     if min(xi_G3(:,MEPDGmodel)) > ThresholdLimits(1,MEPDGmodel)
%         ProbrGP(:,MEPDGmodel) = 1;

```

```

%     end
%     ProbAMV(:,MEPDGmodel) = 1 - interp1q(xi_G4(:,MEPDGmodel),
cdf_G4(:,MEPDGmodel), ThresholdLimits(1,MEPDGmodel));
%     if max(xi_G4(:,MEPDGmodel)) < ThresholdLimits(1,MEPDGmodel)
%         ProbAMV(:,MEPDGmodel) = 0;
%     end
%     if min(xi_G4(:,MEPDGmodel)) > ThresholdLimits(1,MEPDGmodel)
%         ProbAMV(:,MEPDGmodel) = 1;
%     end
% end
%
%
% for MEPDGmodel = 1:num_Models
% for i = 1:size(xi_G4,1)
%     %Calculate Number of Failures
%     if AMVreliability(i,MEPDGmodel) > ThresholdLimits(1,MEPDGmodel)
%         SystemFailAMV(i,MEPDGmodel) = 1;
%     else SystemFailAMV(i,MEPDGmodel) = 0;
%     end
%     %System Reliability
%     if sum(SystemFailAMV(i,:))>0
%         SystemFailure1(i,1) = 1;
%     else SystemFailure1(i,1) = 0;
%     end
%     if RGPpreliability(i,MEPDGmodel) > ThresholdLimits(1,MEPDGmodel)
%         SystemFailRGP(i,MEPDGmodel) = 1;
%     else SystemFailRGP(i,MEPDGmodel) = 0;
%     end
%     %System Reliability
%     if sum(SystemFailRGP(i,:))>0
%         SystemFailure2(i,1) = 1;
%     else SystemFailure2(i,1) = 0;
%     end
% end
% end
%
% AMVsysRel = sum(SystemFailure1) / Nmcs;
% RGPsysRel = sum(SystemFailure2) / Nmcs;
% [ProbAMV', ProbRGP']
% [AMVsysRel', RGPsysRel']
%
% -----
%
% % CASE 3
% ThresholdLimits = [150, 0.75, 0.30, 2000, 0.50];
% for MEPDGmodel = 1:num_Models
%     %Calculate Reliability @ Threshold Limit
%     ProbRGP(:,MEPDGmodel) = 1 - interp1q(xi_G3(:,MEPDGmodel),
cdf_G3(:,MEPDGmodel), ThresholdLimits(1,MEPDGmodel));
%     if max(xi_G3(:,MEPDGmodel)) < ThresholdLimits(1,MEPDGmodel)
%         ProbRGP(:,MEPDGmodel) = 0;
%     end
%     if min(xi_G3(:,MEPDGmodel)) > ThresholdLimits(1,MEPDGmodel)
%         ProbRGP(:,MEPDGmodel) = 1;
%     end
% end

```

```

%   ProbAMV(:,MEPDGmodel) = 1 - interp1q(xi_G4(:,MEPDGmodel),
cdf_G4(:,MEPDGmodel), ThresholdLimits(1,MEPDGmodel));
%   if max(xi_G4(:,MEPDGmodel)) < ThresholdLimits(1,MEPDGmodel)
%       ProbAMV(:,MEPDGmodel) = 0;
%   end
%   if min(xi_G4(:,MEPDGmodel)) > ThresholdLimits(1,MEPDGmodel)
%       ProbAMV(:,MEPDGmodel) = 1;
%   end
% end
%
%
% for MEPDGmodel = 1:num_Models
% for i = 1:size(xi_G4,1)
%   %Calculate Number of Failures
%   if AMVreliability(i,MEPDGmodel) > ThresholdLimits(1,MEPDGmodel)
%       SystemFailAMV(i,MEPDGmodel) = 1;
%   else SystemFailAMV(i,MEPDGmodel) = 0;
%   end
%   %System Reliability
%   if sum(SystemFailAMV(i,:))>0
%       SystemFailure1(i,1) = 1;
%   else SystemFailure1(i,1) = 0;
%   end
%   if RGPpreliability(i,MEPDGmodel) > ThresholdLimits(1,MEPDGmodel)
%       SystemFailRGP(i,MEPDGmodel) = 1;
%   else SystemFailRGP(i,MEPDGmodel) = 0;
%   end
%   %System Reliability
%   if sum(SystemFailRGP(i,:))>0
%       SystemFailure2(i,1) = 1;
%   else SystemFailure2(i,1) = 0;
%   end
% end
% end
%
% AMVsysRel = sum(SystemFailure1) / Nmcs;
% RGPsysRel = sum(SystemFailure2) / Nmcs;
% [ProbAMV', ProbRGP']
% [AMVsysRel', RGPsysRel']

```

```

%-----

```

```

----

```

```

% Correction Factors for AMV values

```

```

%Shift mean of AMV to MCS

```

```

% for MEPDGmodel = 1:num_Models
%   xi_MCS50(1,MEPDGmodel) = interp1q(cdf_MCS(:,MEPDGmodel),
xi_MCS(:,MEPDGmodel), 0.5);
%   xi_AMV50(1,MEPDGmodel) = interp1q(cdf_AMV(:,MEPDGmodel),
xi_AMV(:,MEPDGmodel), 0.5);
%   Mean_diff(1,MEPDGmodel) = xi_MCS50(1,MEPDGmodel) -
xi_AMV50(1,MEPDGmodel);
%
%   for i = 1:size(xi_AMV,1)

```

```

%          xi_AMVmm(i,MEPDGmodel) = xi_AMV(i,MEPDGmodel) +
Mean_diff(1,MEPDGmodel);
%      end
% end
% close all;
% for MEPDGmodel = 1:num_Models
%     figure
%
%     clear MinX MaxX MinY MaxY
%     MinX(MEPDGmodel) = min(min(min(min(xi_G1(:,MEPDGmodel)),
min(min(xi_MCS(:,MEPDGmodel)), min(xi_AMV(:,MEPDGmodel))))), 0);
%     MaxX(MEPDGmodel) = max(max(max(max(xi_G1(:,MEPDGmodel)),max(
max(xi_MCS(:,MEPDGmodel)), max(xi_AMV(:,MEPDGmodel))))), 0);
%     MinY = 0;
%     MaxY = 1;
%
%
%     plot(xi_MCS(:,MEPDGmodel),cdf_MCS(:,MEPDGmodel),'-r',... %CDF
Input Variability + GP + MEPDG
%         xi_AMVmm(:,MEPDGmodel),cdf_AMV(:,MEPDGmodel),'-b',... %AMV
%         xi_AMV(:,MEPDGmodel), cdf_AMV(:,MEPDGmodel));
%
%     axis([MinX(MEPDGmodel) MaxX(MEPDGmodel) 0 1])
%     ylabel('Probability of Failure (F(Dt(x)))');
%     xlabel('Distress Value (Dt(x))');
%     legend('MCS','AMV Mean Moved')
%     % DistressModes = ['TermIRI', 'RuttingTotal', 'ACBottomUp',
'ACTopDown', 'RuttingAC'];
%     if MEPDGmodel <= 1
%         title('Terminal IRI')
%
%     elseif MEPDGmodel <= 2 && MEPDGmodel > 1
%         title('Total Permanent Deformation')
%
%     elseif MEPDGmodel <= 3 && MEPDGmodel > 2
%         title('AC Bottom Up Cracking')
%
%     elseif MEPDGmodel <= 4 && MEPDGmodel > 3
%         title('AC Top Down Cracking')
%
%     elseif MEPDGmodel >= 5
%         title('AC Permanent Deformation')
%     end
% end
%
% for MEPDGmodel = 1:num_Models
%     X_AMV85(1,MEPDGmodel) = interp1q(cdf_AMV(:,MEPDGmodel),
xi_AMVmm(:,MEPDGmodel), 0.85);
%     cdf_MCS85(1,MEPDGmodel) = interp1q(xi_MCS(:,MEPDGmodel),
cdf_MCS(:,MEPDGmodel), X_AMV85(1,MEPDGmodel));
%     if isnan(cdf_MCS85(1,MEPDGmodel)) > 0;
%         if X_AMV85(1,MEPDGmodel) > max(xi_MCS(:,MEPDGmodel))
%             cdf_MCS85(1,MEPDGmodel) = 0.999;
%         else cdf_MCS85(1,MEPDGmodel) = 0.001;
%         end
%     end
% end

```



```

%      CF85(1,MEPDGmodel) = norminv(cdf_MCS85(1,MEPDGmodel)) /
norminv(0.85);
%
%      X_AMV90(1,MEPDGmodel) = interp1q(cdf_AMV(:,MEPDGmodel),
xi_AMVmm(:,MEPDGmodel), 0.90);
%      cdf_MCS90(1,MEPDGmodel) = interp1q(xi_MCS(:,MEPDGmodel),
cdf_MCS(:,MEPDGmodel), X_AMV90(1,MEPDGmodel));
%      if isnan(cdf_MCS90(1,MEPDGmodel)) > 0;
%          if X_AMV90(1,MEPDGmodel) > max(xi_MCS(:,MEPDGmodel))
%              cdf_MCS90(1,MEPDGmodel) = 0.999;
%          else cdf_MCS90(1,MEPDGmodel) = 0.001;
%          end
%      end
%      CF90(1,MEPDGmodel) = norminv(cdf_MCS90(1,MEPDGmodel)) /
norminv(0.90);
%
%      X_AMV95(1,MEPDGmodel) = interp1q(cdf_AMV(:,MEPDGmodel),
xi_AMVmm(:,MEPDGmodel), 0.95);
%      cdf_MCS95(1,MEPDGmodel) = interp1q(xi_MCS(:,MEPDGmodel),
cdf_MCS(:,MEPDGmodel), X_AMV95(1,MEPDGmodel));
%      if isnan(cdf_MCS95(1,MEPDGmodel)) > 0;
%          if X_AMV95(1,MEPDGmodel) > max(xi_MCS(:,MEPDGmodel))
%              cdf_MCS95(1,MEPDGmodel) = 0.999;
%          else cdf_MCS95(1,MEPDGmodel) = 0.001;
%          end
%      end
%      CF95(1,MEPDGmodel) = norminv(cdf_MCS95(1,MEPDGmodel)) /
norminv(0.95);
%
%      X_AMV97(1,MEPDGmodel) = interp1q(cdf_AMV(:,MEPDGmodel),
xi_AMVmm(:,MEPDGmodel), 0.975);
%      cdf_MCS97(1,MEPDGmodel) = interp1q(xi_MCS(:,MEPDGmodel),
cdf_MCS(:,MEPDGmodel), X_AMV97(1,MEPDGmodel));
%      if isnan(cdf_MCS97(1,MEPDGmodel)) > 0;
%          if X_AMV97(1,MEPDGmodel) > max(xi_MCS(:,MEPDGmodel))
%              cdf_MCS97(1,MEPDGmodel) = 0.999;
%          else cdf_MCS97(1,MEPDGmodel) = 0.001;
%          end
%      end
%      CF97(1,MEPDGmodel) = norminv(cdf_MCS97(1,MEPDGmodel)) /
norminv(0.975);
%
%      X_AMV99(1,MEPDGmodel) = interp1q(cdf_AMV(:,MEPDGmodel),
xi_AMVmm(:,MEPDGmodel), 0.99);
%      cdf_MCS99(1,MEPDGmodel) = interp1q(xi_MCS(:,MEPDGmodel),
cdf_MCS(:,MEPDGmodel), X_AMV99(1,MEPDGmodel));
%      if isnan(cdf_MCS99(1,MEPDGmodel)) > 0;
%          if X_AMV99(1,MEPDGmodel) > max(xi_MCS(:,MEPDGmodel))
%              cdf_MCS99(1,MEPDGmodel) = 0.999;
%          else cdf_MCS99(1,MEPDGmodel) = 0.001;
%          end
%      end
%      CF99(1,MEPDGmodel) = norminv(cdf_MCS99(1,MEPDGmodel)) /
norminv(0.99);
% end
% CF_all = [CF85; CF90; CF95; CF97; CF99]

```

```

%
% CF_all
%
% X_AMVall = [X_AMV85; X_AMV90; X_AMV95; X_AMV97; X_AMV99];
%
%
%
%
% Betas = [norminv(0.85); norminv(0.90); norminv(0.95); norminv(0.975);
norminv(0.99)];
%
%
% for i = 1:5
%     for j = 1:7
%         CF_pntsTermIRI(i,j) = AMVstats(1,j) - Alphas(1,j) *
CF_all(i,1) * Betas(i,1) * AMVstats(2,j);
%         CF_pntsTotPD(i,j) = AMVstats(1,j) - Alphas(2,j) *
CF_all(i,2) * Betas(i,1) * AMVstats(2,j);
%         CF_pntsACBU(i,j) = AMVstats(1,j) - Alphas(3,j) *
CF_all(i,3) * Betas(i,1) * AMVstats(2,j);
%         CF_pntsACTD(i,j) = AMVstats(1,j) - Alphas(4,j) *
CF_all(i,4) * Betas(i,1) * AMVstats(2,j);
%         CF_pntsACPD(i,j) = AMVstats(1,j) - Alphas(5,j) *
CF_all(i,5) * Betas(i,1) * AMVstats(2,j);
%     end
% end
%
%
% gp_CFAMV(:,1) = predictor(CF_pntsTermIRI, GPmodel(1,1));
% gp_CFAMV(:,2) = predictor(CF_pntsTotPD, GPmodel(1,2));
% gp_CFAMV(:,3) = predictor(CF_pntsACBU, GPmodel(1,3));
% gp_CFAMV(:,4) = predictor(CF_pntsACTD, GPmodel(1,4));
% gp_CFAMV(:,5) = predictor(CF_pntsACPD, GPmodel(1,5));
%
%
%
% for MEPDGmodel = 1:num_Models
%
%     xi_CFAMV(:,MEPDGmodel) = xi_AMVmm(:,MEPDGmodel);
%
%     yy85 = find(cdf_MCS(:,MEPDGmodel) < 0.859);
%     Inds_CF(1,MEPDGmodel) = yy85(end);
%
%     yy90 = find(cdf_MCS(:,MEPDGmodel) < 0.901);
%     Inds_CF(2,MEPDGmodel) = yy90(end);
%
%     yy95 = find(cdf_MCS(:,MEPDGmodel) < 0.951);
%     Inds_CF(3,MEPDGmodel) = yy95(end);
%
%     yy97 = find(cdf_MCS(:,MEPDGmodel) < 0.976);
%     Inds_CF(4,MEPDGmodel) = yy97(end);
%
%     yy99 = find(cdf_MCS(:,MEPDGmodel) < 0.999);
%     Inds_CF(5,MEPDGmodel) = yy99(end);
% end
%
% for MEPDGmodel = 1:num_Models
%     for i = 1:5

```

```

%           pred_CFAMV(i,MEPDGmodel) = gp_CFAMV(i,MEPDGmodel)+
Mean_diff(1,MEPDGmodel);
%       end
% end
%
%
% %Predict from CDFs to include uncertainty in prediction
% for MEPDGmodel = 1:num_Models
% %     pred85(1,MEPDGmodel) = interp1q(cdf_AMV(:,MEPDGmodel),
xi_CFAMV(:,MEPDGmodel), 0.85);
% %     pred90(1,MEPDGmodel) = interp1q(cdf_AMV(:,MEPDGmodel),
xi_CFAMV(:,MEPDGmodel), 0.90);
% %     pred95(1,MEPDGmodel) = interp1q(cdf_AMV(:,MEPDGmodel),
xi_CFAMV(:,MEPDGmodel), 0.95);
% %     pred97(1,MEPDGmodel) = interp1q(cdf_AMV(:,MEPDGmodel),
xi_CFAMV(:,MEPDGmodel), 0.975);
% %     pred99(1,MEPDGmodel) = interp1q(cdf_AMV(:,MEPDGmodel),
xi_CFAMV(:,MEPDGmodel), 0.99);
%
%     mcs_vals85(1,MEPDGmodel) = interp1q(cdf_MCS(:,MEPDGmodel),
xi_MCS(:,MEPDGmodel), 0.85);
%     mcs_vals90(1,MEPDGmodel) = interp1q(cdf_MCS(:,MEPDGmodel),
xi_MCS(:,MEPDGmodel), 0.90);
%     mcs_vals95(1,MEPDGmodel) = interp1q(cdf_MCS(:,MEPDGmodel),
xi_MCS(:,MEPDGmodel), 0.95);
%     mcs_vals97(1,MEPDGmodel) = interp1q(cdf_MCS(:,MEPDGmodel),
xi_MCS(:,MEPDGmodel), 0.975);
%     mcs_vals99(1,MEPDGmodel) = interp1q(cdf_MCS(:,MEPDGmodel),
xi_MCS(:,MEPDGmodel), 0.99);
% end
%
% pred_CFAMVall = pred_CFAMV%[pred85; pred90; pred95; pred97; pred99];
% mcs_valsAll   = [mcs_vals85; mcs_vals90; mcs_vals95; mcs_vals97;
mcs_vals99];
%
% clear DissResultsTable DissTableFormat
% for MEPDGmodel = 1:num_Models
%     DissResultsTable(:, :, MEPDGmodel) = [X_AMVall(:,MEPDGmodel), ...
%         CF_all(:,MEPDGmodel), ...
%         pred_CFAMVall(:,MEPDGmodel)...
%         mcs_valsAll(:,MEPDGmodel)];
% end
%
% DissTableFormat = [DissResultsTable(:, :, 1), ...
%     DissResultsTable(:, :, 2), ...
%     DissResultsTable(:, :, 3), ...
%     DissResultsTable(:, :, 4), ...
%     DissResultsTable(:, :, 5)];
%
% for MEPDGmodel = 1:num_Models
%     figure
%
%     clear MinX MaxX MinY MaxY
%     MinX(MEPDGmodel) = min(min(min(min(xi_CFAMV(:,MEPDGmodel)),
min(min(xi_MCS(:,MEPDGmodel)), min(xi_AMV(:,MEPDGmodel))))), 0);

```

```

%     MaxX(MEPDGmodel) = max(max(max(max(xi_CFAMV(:,MEPDGmodel)),max(
max(xi_MCS(:,MEPDGmodel)), max(xi_AMV(:,MEPDGmodel))))), 0);
%     MinY = 0;
%     MaxY = 1;
%
%
%     plot(xi_MCS(:,MEPDGmodel),cdf_MCS(:,MEPDGmodel),'.-r',... %CDF
Input Variability + GP + MEPDG
%         xi_CFAMV(:,MEPDGmodel),cdf_AMV(:,MEPDGmodel),'.-g'); %AMV
%
%     axis([MinX(MEPDGmodel) MaxX(MEPDGmodel) 0 1])
%     ylabel('Probability of Failure (F(Dt(x)))');
%     xlabel('Distress Value (Dt(x))');
%     legend('Input Variability + GP + MEPDG','AMV + GP + MEPDG',
'Corrected AMV')
%     % DistressModes = ['TermIRI', 'RuttingTotal', 'ACBottomUp',
'ACTopDown', 'RuttingAC'];
%     if MEPDGmodel <= 1
%         title('Terminal IRI')
%
%     elseif MEPDGmodel <= 2 && MEPDGmodel > 1
%         title('Total Permanent Deformation')
%
%     elseif MEPDGmodel <= 3 && MEPDGmodel > 2
%         title('AC Bottom Up Cracking')
%
%     elseif MEPDGmodel <= 4 && MEPDGmodel > 3
%         title('AC Top Down Cracking')
%
%     elseif MEPDGmodel >= 5
%         title('AC Permanent Deformation')
%     end
% end
%
%
% close all;

%%-----
----
%% Contour Plots for ACTD

% %Generate a bunch of points
% N=1000;
% clear Xsa
% Xsa = ones(N,7);
% for i = 1:N
%     for j = 1:7
%         Xsa(i,j) = Xsa(i,j) * AMVstats(1,j);
%     end
% end
%
% AADTsa = sort(random('norm',1500,150,N,1));
% HMATHICKsa = sort(random('norm',8,0.8,N,1));
%
% Xsa(:,1) = AADTsa;
% Xsa(:,2) = HMATHICKsa;

```

```

%
% for MEPDGmodel = 4
%   MEPDGmodel
%   %Evaluate all Combinations of AADT and HMAthickness
%   clear gp_eval
%   for i=1:N
%     for j=1:N
%       gp_eval(i,j) =
predictor([Xsa(i,1),Xsa(j,2),Xsa(1,3:7)],GPmodel(MEPDGmodel));
%       end
%     end
%   %Transform from log space
%   %   gp_eval = exp(gp_eval);
%
%   %Create contour plots of GP
%   clear C1 h1
%   figure
%   [C1,h1]=contourf(AADTsa,HMAthicknessa,gp_eval());
%   clabel(C1,h1);
%   xlabel('Average Annual Daily Traffic')
%   ylabel('HMA Thickness (in.)')
%   if MEPDGmodel <= 1
%     title('Terminal IRI')
%
%   elseif MEPDGmodel <= 2 && MEPDGmodel > 1
%     title('Total Permanent Deformation')
%
%   elseif MEPDGmodel <= 3 && MEPDGmodel > 2
%     title('AC Bottom Up Cracking')
%
%   elseif MEPDGmodel <= 4 && MEPDGmodel > 3
%     title('AC Top Down Cracking')
%
%   elseif MEPDGmodel >= 5
%     title('AC Permanent Deformation')
%   end
% end

```

Chapter VII

Verification of Predictive Capability of GP Models

Distribution of Bayes Factors for GP Models

```
clear all; clc; close all;

training_pointsCOMPLETE;
r=size(train_pntsCOMPLETE);
s=size(train_valsCOMPLETE);

%Run numerous iterations
num_iters = 10;
%Run specific number of MEPDG models;
%need to revise BayesFactor if eval only some MEPDGmodels (Lines 178-
182)
num_Models = 5;

%Perform Verification: MSE, Rsquared, and Bayes with chosen number of
%training points

for y=1:num_iters
Y
    num_test = 10;
    num_train = r(1) - num_test;
    %Randomly generate which terms are selected as training vs. testing
    m(:,y)=(randperm(r(1)))';
%    m(:,y) = (1:r(1))';

    %Define Training Point Matricies
    [train_pntsFULL,train_mean,train_std]=zscore(train_pntsCOMPLETE);
    for i=1:num_train
        train_pnts(i,:)=train_pntsFULL(m(i,y),:);
        train_vals(i,:)=train_valsCOMPLETE(m(i,y),:);
    end

    yoffset = mean(train_vals);
    for i=1:size(train_vals,1)
        for j=1:size(train_vals,2)
            train_vals0(i,j) = train_vals(i,j) - yoffset(1,j);
        end
    end

    %Define Test Point Matricies
    for i=1:num_test
        test_pnts(i,:)=train_pntsFULL(m(num_train+i,y),:);
        test_vals(i,:)=train_valsCOMPLETE(m(num_train+i,y),:);
    end

    %Construct and Evaluate GP
    for MEPDGmodel = 1:num_Models
```

```

%Train Model
nsams = size(train_pnts,1);
ndims = size(train_pnts,2);
theta0 = ones(1,ndims); lob = 0.01*ones(1,ndims); upb =
10*ones(1,ndims);
[GPmodel(MEPDGmodel), GPModelInfo(MEPDGmodel)] =
dacefit(train_pnts, train_vals0(:,MEPDGmodel), @regpoly1, @corr Gauss,
theta0, lob,upb);

%Evaluate GP model for Verification Process
gptest(:,MEPDGmodel) =
predictor(test_pnts,GPmodel(MEPDGmodel));
end

for i=1:num_test
for MEPDGmodel=1:num_Models
GPtest(i,MEPDGmodel) = gptest(i,MEPDGmodel) +
yoffset(1,MEPDGmodel);
end
end

for i=1:num_test
for j=1:num_Models
Residuals(i,j)=(test_vals(i,j)-GPtest(i,j));
%Calculate R^2 value using Matlab function; linear
regression WITH
%intercept; first beta value will be the y-intercept =>
representing the
%systematic bias of the underlying (MEPDG) function
stats(y,j) =
regstats(test_vals(:,j),GPtest(:,j),'linear',{'rsquare', 'beta'});
%Calculate the slope of a generalized linear model
%desired slope = 1 want X = y (MEPDG = GP)
%glmslope(y,j) =

%glmfit(GPtest(:,j),test_vals(:,j),'normal','link','identity','constant
','off');
end
end

for i=1:size(Residuals,2)
ResidualsStdDev(1,i)=std(Residuals(:,i));
end

%=====
%Caclulate Predictive Coefficient of Determination (R^2)
%=====
clear meanTestVals diff diff2 SStotal SSerror
for i=1:num_test
for j=1:num_Models
RsquaredMTLB(y,j) = stats(y,j).rsquare;
end
end
end

```

```

%=====
%Calculate Bayes Factor
%=====
%Number of Testing Points
    q=size(test_pnts,1);
    for i=1:q
        for j=1:num_Models

posteriorPDF(i,j)=normpdf(Residuals(i,j),0,ResidualsStdDev(1,j));
        end
    end

        %Construct PDF's for each GP Evaluated at Test Values
        for j=1:num_Models

[pdfGPtest(:,j),xiGPtest(:,j)]=ksdensity(GPtest(:,j),'function','pdf');
        end

        %Construct PDF's for each MEPDG Evaluated at Test Values
        for j=1:num_Models

[pdfMEtest(:,j),xiMEtest(:,j)]=ksdensity(test_vals(:,j),'function','pdf
');
        end

        for i=1:q
            for j=1:num_Models

pdfGP(i,j)=interp1q(xiGPtest(:,j),pdfGPtest(:,j),test_vals(i,j));
            end
        end

            P = prod(posteriorPDF);
            Prod = prod(pdfGP);

            for j=1:num_Models
                BayesFactor(y,j)=P(1,j)/Prod(1,j);
            end
end

posteriorPDF;
pdfGP;
BayesFactor;
%Remove Nan values;
%Nan values occur when pdfGP is interpolated at test points outside the
%GP results

i = find(~isnan(BayesFactor(:,1))); B1 = BayesFactor(i,1);
i = find(~isnan(BayesFactor(:,2))); B2 = BayesFactor(i,2);
i = find(~isnan(BayesFactor(:,3))); B3 = BayesFactor(i,3);
i = find(~isnan(BayesFactor(:,4))); B4 = BayesFactor(i,4);
i = find(~isnan(BayesFactor(:,5))); B5 = BayesFactor(i,5);

%Calculate Average Values for Verification Metrics
AveRsquaredMTLB = mean(RsquaredMTLB,1);

```



```

AveB = [mean(B1);mean(B2);mean(B3);mean(B4);mean(B5)];
%Aveglmslope = mean(glmslope,1);

%Calculate Variance Values for Verification Metrics
VarRsquaredMTLB = var(RsquaredMTLB,1);
VarB = [var(B1);var(B2);var(B3);var(B4);var(B5)];
%Varglmslope = var(glmslope,1);

%Calculate Coefficient of Variation (COV) for Verification Metrics
clear COVRsquared COVB
for i=1:num_Models
    COVRsquared(i,1) = sqrt(VarRsquaredMTLB(1,i)) /
AveRsquaredMTLB(1,i);
    COVB(i,1) = sqrt(VarB(i,1)) / AveB(i,1);
end
%Calculate Probability that Bayes Factor is Less than 3
for i=1:size(B1,1)
    if B1 < 3
        CountB1(i,1) = 1;
    else
        CountB1(i,1) = 0;
    end
end
ProbB1LessThan3 = sum(CountB1,1)/(size(B1,1));

for i=1:size(B2,1)
    if B2 < 3
        CountB2(i,1) = 1;
    else
        CountB2(i,1) = 0;
    end
end
ProbB2LessThan3 = sum(CountB2)/(size(B2,1));

for i=1:size(B3,1)
    if B1 < 3
        CountB3(i,1) = 1;
    else
        CountB3(i,1) = 0;
    end
end
ProbB3LessThan3 = sum(CountB3)/(size(B3,1));

for i=1:size(B4,1)
    if B4 < 3
        CountB4(i,1) = 1;
    else
        CountB4(i,1) = 0;
    end
end
ProbB4LessThan3 = sum(CountB4)/(size(B4,1));

for i=1:size(B5,1)
    if B5 < 3
        CountB5(i,1) = 1;

```

```

        else
            CountB5(i,1) = 0;
        end
    end
end
ProbB5LessThan3 = sum(CountB5)/(size(B5,1));

ProbBFLessThan3 =
[ProbB1LessThan3;ProbB2LessThan3;ProbB3LessThan3;ProbB4LessThan3;ProbB5
LessThan3];

%Columns = Metric; Rows = MEPDGmodel
ValidationResults = [AveRsquaredMTLB',VarRsquaredMTLB',COVRsquared,
AveB,VarB, COVB];
%GLMSlopeResults = [Aveglmslope',Varglmslope'];

% %% ACCRE Output Files
% %Output Verification Results
% VerificationResults = fopen('VerificationResults.txt','a');
% for k=1:size(ValidationResults,1)
%     for i=1:size(ValidationResults,2)
%         fprintf(VerificationResults,'%G\t',ValidationResults(k,i));
%     end
%     fprintf(VerificationResults,'\n',ValidationResults(k,i));
% end

BayesFactorProb = fopen('BayesFactorProb.txt','a');
for k=1:size(ProbBFLessThan3,1)
    fprintf(BayesFactorProb,'%G\t',ProbBFLessThan3(k,1));
end

```

Chapter VIII

GP Verification Results

Improvement in Average GP Variance for Verification Points

Design Optimization Results

```
clear all; clc; close all;

%%
=====
=
% Import Training Data
%
=====
=
InitRGP = RGPinitializerOrig;
RGP_valsYear = InitRGP{1,1};
RGP_pntsYear = InitRGP{1,2};
ParamNamesTrain = InitRGP{1,3};
%Initializing Data

%Choose only most significant 8 parameters
RGP_pntsYear(:, [2:12,14:22,26:28,31:46,48:50,52:53]) = [];
ParamNamesTrain(:, [2:12,14:22,26:28,31:46,48:50,52:53]) = [];

ndims = size(RGP_pntsYear,2);
num_Models = size(RGP_valsYear,2);

%Verification Points
Ver_pntsYear = importdata('RGPVerificationInputs.mat');
Ver_valsYear = importdata('RGPVer_vals.mat');
Ver_pntsYear(:,55:76)=[]; Ver_pntsYear(:,13) = [];
Ver_pntsYear(:, [2:12,14:22,26:28,31:46,48:50,52:53]) = [];
Ver_valsYear(:,4) = [];

TPmeans = mean(RGP_pntsYear);
TPstdev = std(RGP_pntsYear);
TPlb = min(RGP_pntsYear);
TPub = max(RGP_pntsYear);

TPstats = [TPmeans; TPstdev; TPlb; TPub];
%=====
=====
%% Initialize; Build GP with 'n' Training Points; Selected Randomly
```

```

%=====
=====
Note = 'Select Initial GP Training Points'
%Construct Initial GP
% Select Random TPs

num_trainInit = 100;
IXinit = randperm(size(RGP_pntsYear,1));

Init_pntsTrain = RGP_pntsYear(IXinit(1:num_trainInit),:);
Init_valsTrain = RGP_valsYear(IXinit(1:num_trainInit),:);

clear InitCand_pnts InitCand_vals
InitCand_pnts =
RGP_pntsYear(IXinit(num_trainInit+1:size(RGP_pntsYear,1)),:);
InitCand_vals =
RGP_valsYear(IXinit(num_trainInit+1:size(RGP_valsYear,1)),:);

theta0 = 1*ones(1,ndims);
lob = 0.01*ones(1,ndims);
upb = 100*ones(1,ndims);
clear GPmodel GPModelInfo
for MEPDGmodel = 1:num_Models
    MEPDGmodel
    [GPmodel(1, MEPDGmodel), GPModelInfo(1, MEPDGmodel)] =...
        dacefit(Init_pntsTrain, Init_valsTrain(:,MEPDGmodel),...
            @regpoly1, @corrgauss, theta0,lob,upb);
end

for MEPDGmodel = 1:num_Models
    [Ver_preds(:,MEPDGmodel), VV_var(:,MEPDGmodel)] =
predictor(Ver_pntsYear, GPmodel(:,MEPDGmodel));
    [fitobject, gof] =
fit(Ver_valsYear(:,MEPDGmodel),Ver_preds(:,MEPDGmodel), 'poly1');
    Ver_ARsqd(1,MEPDGmodel) = gof.adjrsquare;
end

Ver_ARsqd
%%=====
=====
%Calculate Ave GP Variance at Verification Points

VV_varModels = mean(VV_var);
VV_varMean = mean(mean(VV_var));

%=====
=====
%% Perform Design Optimization According to P3 Formulation; Use MCS
%=====
=====
Note = 'Add TPs to GP Construction Until Stopping Criterion Achieved'

%Initialize

```

```

clear RGP_pntsTrain RGP_valsTrain Cand_pnts Cand_vals
clear VV_holder VV_GPsize
RGP_pntsTrain = Init_pntsTrain;
RGP_valsTrain = Init_valsTrain;

Cand_pnts = InitCand_pnts;
Cand_vals = InitCand_vals;
VV_holder = VV_varMean;
VV_GPsize = size(RGP_pntsTrain,1);
Iter = size(RGP_pntsTrain,1);
MaxCandVar = [0];

num_trainMax = 998;
num_train = size(Init_pntsTrain,1);
while num_train < num_trainMax

    %Exploit = Cost Function
    NextTPexploit = PrimaryExploitP3(GPmodel, Cand_pnts,
ParamNamesTrain, TPstats);
    %Explore = GP Variance Search
    NextTPexplore = PrimaryExploreP3(GPmodel, Cand_pnts, Cand_vals);
    MaxCandVar = [MaxCandVar, NextTPexplore{1,2}];

    %Re-train using Opt Point at New Training Point
    New_pnts = [NextTPexplore{1,1},NextTPexploit{1,1}];
    New_pnts = unique(New_pnts); %unique function sorts in ascending
order

    clear EE_pnts EE_vals
    for i = 1:size(New_pnts,2)
        EE_pnts(i,:) = Cand_pnts(New_pnts(i),:);
        EE_vals(i,:) = Cand_vals(New_pnts(i),:);
    end
    RGP_pntsTrain = [RGP_pntsTrain; EE_pnts];
    RGP_valsTrain = [RGP_valsTrain; EE_vals];
    New_pntsDescend = sort(New_pnts, 'descend');
    for i = 1:size(New_pntsDescend,2)
        Cand_pnts(New_pntsDescend(i),:) = [];
        Cand_vals(New_pntsDescend(i),:) = [];
    end
    %Build New GP
    clear GPmodel GPModelInfo
    for MEPDGmodel = 1:num_Models
        [GPmodel(1, MEPDGmodel), GPModelInfo(1, MEPDGmodel)] =...
dacefit(RGP_pntsTrain, RGP_valsTrain(:,MEPDGmodel),...
@regpoly1, @corrgauss, theta0,lob,upb);
    end

    %Calculate Ave GP Variance at Verification Points
    clear VV_var
    for MEPDGmodel = 1:num_Models
        [Ver_preds(:,MEPDGmodel), VV_var(:,MEPDGmodel)] =
predictor(Ver_pntsYear,GPmodel(MEPDGmodel));
        [fitobject, gof] =
fit(Ver_valsYear(:,MEPDGmodel),Ver_preds(:,MEPDGmodel), 'poly1');

```

```

        RGP_ARSqd(1,MEPDGmodel) = gof.adjrsquare;
    end
    VV_varModels = [VV_varModels; mean(VV_var)];
    VV_holder = [VV_holder; mean(mean(VV_var))];
    VV_GPsize = [VV_GPsize; size(RGP_pntsTrain,1)];

    num_train = size(RGP_pntsTrain,1)
end

%=====
====
%Improvement
for MEPDGmodel = 1:num_Models
    figure

    Xplot = VV_GPsize; Xplot(1) = [];
    Yplot = diff(VV_varModels(:,MEPDGmodel));
    plot(Xplot, Yplot);
    xlabel('Number of Training Points');
    ylabel('GP Variance @ Verification Points');
    if MEPDGmodel <= 1
        title('Terminal IRI')

    elseif MEPDGmodel <= 2 && MEPDGmodel > 1
        title('Total Permanent Deformation')

    elseif MEPDGmodel <= 3 && MEPDGmodel > 2
        title('AC Bottom Up Cracking')

    elseif MEPDGmodel <= 4 && MEPDGmodel > 3
        title('AC Top Down Cracking')

    elseif MEPDGmodel >= 5
        title('AC Permanent Deformation')
    end
end

%
%=====
====
% %% Save Workspace Variables
%
%=====
====
Note = 'Saving Workspace'
ClockName = clock;
File = ClockName(1,1)*100000000000 + ClockName(1,2)*1000000000 +
ClockName(1,3)*10000000 + ClockName(1,4)*100000 + ClockName(1,5)*1000 +
round(ClockName(1,6));
FileName = num2str(File);
FileName = strcat('C:\Users\Jenny\Documents\Retherford-
Vanderbilt\Spring2012\OptimizationProblems\Matlab\ChapterVIII\',
FileName, 'GPModelVerification.mat');
save(FileName);

```

```

close all;
%=====
=====
%Optimization Problem Solution
mean_Res = [-0.3081,-5.6077e-4,0.0371,20.6545,-7.7039e-4];
std_Res = [0.8445,0.0028,0.1769,105.6707,0.0013];
Umepdg = [34.022, 0.1223, 9.4563, 1821.3, 0.0990];

%Threshold Limits
% DistressModes =
% ['TermIRI', 'RuttingTotal', 'ACBottomUp', 'ACTopDown', 'RuttingAC'];
ThresholdLimits = [275, 1.25, 25, 2000, 0.75];

%% Build GP
%From plots, typically need only Nopt training points to have 'good' GP
%which TPs:
Nopt = 500;
Optimal_pntsTrain = RGP_pntsTrain(1:Nopt,:);
Optimal_valsTrain = RGP_valsTrain(1:Nopt,:);

clear GPmodel GPModelInfo
for MEPDGmodel = 1:num_Models
    MEPDGmodel
    [GPmodelOpt(1, MEPDGmodel), GPModelInfoOpt(1, MEPDGmodel)] =...
    dacefit(Optimal_pntsTrain, Optimal_valsTrain(:,MEPDGmodel),...
    @regpoly1, @corr Gauss, theta0, lob, upb);
end
clear Xplot Yplot
Xplot = Ver_valsYear; %MEPDG
clear Ver_preds fitobject gof RGP_ARsqd RGP_sse
for MEPDGmodel = 1:num_Models
    clear Yplot
    Yplot = predictor(Ver_pntsYear, GPmodelOpt(:,MEPDGmodel)); %GP

    Ver_preds(:,MEPDGmodel) = Yplot;
    [fitobject, gof] =
    fit(Ver_valsYear(:,MEPDGmodel),Ver_preds(:,MEPDGmodel), 'poly1');
    RGP_ARsqd(1,MEPDGmodel) = gof.adjrsquare;
end

%=====
=====
%Find Optimal Soln by MCS
%Vary Design Parameters: HMAThick EBC AV GBThick [3,4,5,8]
%Keep all other variable parameters at mean
N_mcs = 10000;

OptCand_pnts = ones(N_mcs,ndims);
for MEPDGmodel = 1:ndims
    for i = 1:N_mcs
        OptCand_pnts(i,MEPDGmodel) = OptCand_pnts(i,MEPDGmodel) *
        TPmeans(1,MEPDGmodel);
    end
end
end

```

```

for i = 1:N_mcs
    OptCand_pnts(i,3) = random('unif',TPlb(1,3), TPub(1,3));
    OptCand_pnts(i,4) = random('unif',TPlb(1,4), TPub(1,4));
    OptCand_pnts(i,5) = random('unif',TPlb(1,5), TPub(1,5));
    OptCand_pnts(i,8) = random('unif',TPlb(1,8), TPub(1,8));
end
N_opt = 100;
for iter = 1:size(OptCand_pnts,1)

    Opt_pntsMeans = OptCand_pnts(iter,:);
    clear MCSpointsOpt
    for i = 1:ndims
        MCSpointsOpt(:,i) = random('norm', Opt_pntsMeans(1,i),
TPstdev(1,i), [N_opt,1]);
    end

    clear gp_evalMCS
    for MEPDGmodel = 1:num_Models
        %Evaluate GP model for Uncertainty Analysis
        gp_evalMCS(:,MEPDGmodel) =
predictor(MCSpointsOpt,GPmodelOpt(MEPDGmodel));
    end

    for MEPDGmodel=1:num_Models
        %Input Variability Only

[cdf_G1(:,MEPDGmodel),xi_G1(:,MEPDGmodel)]=ksdensity(gp_evalMCS(:,MEPDG
model),'function','cdf');
        %MEPDG Uncertainty
        R(:,MEPDGmodel) = random('norm',0,Umepdg(1,MEPDGmodel),
[N_opt,1]);
        %GP Uncertainty
        R2(:,MEPDGmodel) =
random('norm',mean_Res(1,MEPDGmodel),std_Res(1,MEPDGmodel), [N_opt,1]);

        %Input Parameters + GP
        for i=1:N_opt
            G2(i,MEPDGmodel) = gp_evalMCS(i,MEPDGmodel) +
R2(i,MEPDGmodel);
            G3(i,MEPDGmodel) = gp_evalMCS(i,MEPDGmodel) +
R2(i,MEPDGmodel) + R(i,MEPDGmodel);
        end
        [cdf_G2(:,MEPDGmodel),xi_G2(:,MEPDGmodel)] =
ksdensity(G2(:,MEPDGmodel),'function','cdf');
        [cdf_G3(:,MEPDGmodel),xi_G3(:,MEPDGmodel)] =
ksdensity(G3(:,MEPDGmodel),'function','cdf');
    end
    for MEPDGmodel = 1:num_Models
        %Calculate Reliability @ Threshold Limit
        ProbFailure(:,MEPDGmodel) = 1 - interp1q(xi_G3(:,MEPDGmodel),
cdf_G3(:,MEPDGmodel), ThresholdLimits(1,MEPDGmodel));
        if max(xi_G3(:,MEPDGmodel)) < ThresholdLimits(1,MEPDGmodel)
            ProbFailure(:,MEPDGmodel) = 0;
        end
    end
end

```



```

        end
        PfAve(iter,1) = mean(ProbFailure);
    end

%% Evaluate Cost Function for All Potential (Candidate) Training Points
indHMA = 0; indGB = 0;
TF_HMA = zeros(0,ndims); TF_GB = zeros(0,ndims);
InitConCost = zeros(size(OptCand_pnts,1),1);
ReconCost = zeros(size(OptCand_pnts,1),1);
CandCost = zeros(size(OptCand_pnts,1),1);
for i = 1:ndims
    TF_HMA(i) = strcmp(ParamNamesTrain(1,i),'HMAThick');
    indHMA = indHMA + TF_HMA(i) * i;
    TF_GB(i) = strcmp(ParamNamesTrain(1,i),'GBThick');
    indGB = indGB + TF_GB(i) * i;
end
for i = 1:size(OptCand_pnts,1)
    HMAThick = OptCand_pnts(i,indHMA);
    GBthick = OptCand_pnts(i,indGB);

    InitConCost(i,1) = 20000*HMAThick + 7500*GBthick;

    ReconCost(i,1) = 125000 * PfAve(i,1);

    CandCost(i,1) = InitConCost(i,1) + ReconCost(i,1);
end

[OptMean, OptCostInd] = min(CandCost);

OptimalCost = OptMean

%=====
%====
%Re-evaluate optimal solution
Opt_pntsMeans = OptCand_pnts(OptCostInd,:);
Opt_pntsWORD = Opt_pntsMeans(1,[3,4,5,8]);
clear MCSpointsOpt
for i = 1:ndims
    MCSpointsOpt(:,i) = random('norm', Opt_pntsMeans(1,i),
    TPstdev(1,i), [N_opt,1]);
end

clear gp_evalMCS
for MEPDGmodel = 1:num_Models
    %Evaluate GP model for Uncertainty Analysis
    gp_evalMCS(:,MEPDGmodel) =
    predictor(MCSpointsOpt,GPmodelOpt(MEPDGmodel));
end

for MEPDGmodel=1:num_Models
    %Input Variability Only

```

```

[cdf_G1(:,MEPDGmodel),xi_G1(:,MEPDGmodel)]=ksdensity(gp_evalMCS(:,MEPDG
model),'function','cdf');
    %MEPDG Uncertainty
    R(:,MEPDGmodel) = random('norm',0,Umepdg(1,MEPDGmodel), [N_opt,1]);
    %GP Uncertainty
    R2(:,MEPDGmodel) =
random('norm',mean_Res(1,MEPDGmodel),std_Res(1,MEPDGmodel), [N_opt,1]);

    %Input Parameters + GP
    for i=1:N_opt
        G2(i,MEPDGmodel) = gp_evalMCS(i,MEPDGmodel) + R2(i,MEPDGmodel);
        G3(i,MEPDGmodel) = gp_evalMCS(i,MEPDGmodel) + R2(i,MEPDGmodel)
+ R(i,MEPDGmodel);
    end
    [cdf_G2(:,MEPDGmodel),xi_G2(:,MEPDGmodel)] =
ksdensity(G2(:,MEPDGmodel),'function','cdf');
    [cdf_G3(:,MEPDGmodel),xi_G3(:,MEPDGmodel)] =
ksdensity(G3(:,MEPDGmodel),'function','cdf');
end
for MEPDGmodel = 1:num_Models
    %Calculate Reliability @ Threshold Limit
    ProbPlot(:,MEPDGmodel) = interp1q(xi_G3(:,MEPDGmodel),
cdf_G3(:,MEPDGmodel), ThresholdLimits(1,MEPDGmodel));
    if max(xi_G3(:,MEPDGmodel)) < ThresholdLimits(1,MEPDGmodel)
        ProbPlot(:,MEPDGmodel) = 1;
    end
end
end

OptimalRelThresh = ProbPlot*100
MinRel = 0.9; %Target threshold; not included in opt. routine
for MEPDGmodel = 1:num_Models
    figure

    clear MinX MaxX MinY MaxY
    MinX(MEPDGmodel) = min(min(min(min(xi_G1(:,MEPDGmodel)),
min(min(xi_G2(:,MEPDGmodel)), min(xi_G3(:,MEPDGmodel))))),
ThresholdLimits(:,MEPDGmodel)), 0);
    MaxX(MEPDGmodel) = max(max(max(max(xi_G1(:,MEPDGmodel)),max(
max(xi_G2(:,MEPDGmodel)), max(xi_G3(:,MEPDGmodel))))),
ThresholdLimits(:,MEPDGmodel)), 0);
    MinY = 0;
    MaxY = 1;

    plot(xi_G1(:,MEPDGmodel),cdf_G1(:,MEPDGmodel),'-b',... %CDF Input
Variability
        xi_G2(:,MEPDGmodel),cdf_G2(:,MEPDGmodel),'-g',... %CDF Input
Variability + GP
        xi_G3(:,MEPDGmodel),cdf_G3(:,MEPDGmodel),'-r',... %CDF Input
Variability + GP + MEPDG
    [MinX(MEPDGmodel) MaxX(MEPDGmodel)], [MinRel MinRel], '--k',...
%Reliability Constraint
    [ThresholdLimits(MEPDGmodel) ThresholdLimits(MEPDGmodel)],
[MinY MaxY], '--k',... %Threshold Constraint
    [MinX(MEPDGmodel) MaxX(MEPDGmodel)], [ProbPlot(MEPDGmodel)
ProbPlot(MEPDGmodel)], 'k'); % Reliability at Threshold

```

```

axis([MinX(MEPDGmodel) MaxX(MEPDGmodel) 0 1])
ylabel('Reliability (F(Dt(x)))');
xlabel('Distress Value (Dt(x))');
legend('Input Variability','Input Variability + GP','Input
Variability + GP + MEPDG','Threshold Limits', 'Reliability @ Threshold
Value')
    % DistressModes = ['TermIRI', 'RuttingTotal', 'ACBottomUp',
'ACTopDown', 'RuttingAC'];
    if MEPDGmodel <= 1
        title('Terminal IRI')

    elseif MEPDGmodel <= 2 && MEPDGmodel > 1
        title('Total Permanent Deformation')

    elseif MEPDGmodel <= 3 && MEPDGmodel > 2
        title('AC Bottom Up Cracking')

    elseif MEPDGmodel <= 4 && MEPDGmodel > 3
        title('AC Top Down Cracking')

    elseif MEPDGmodel >= 5
        title('AC Permanent Deformation')
    end
end

%
%=====
====
% %% Save Workspace Variables
%
%=====
====
Note = 'Saving Workspace'
ClockName = clock;
File = ClockName(1,1)*100000000000 + ClockName(1,2)*1000000000 +
ClockName(1,3)*10000000 + ClockName(1,4)*100000 + ClockName(1,5)*1000 +
round(ClockName(1,6));
FileName = num2str(File);
FileName = strcat('C:\Users\Jenny\Documents\Retherford-
Vanderbilt\Spring2012\OptimizationProblems\Matlab\ChapterVIII\',
FileName, 'GPModelVerification.mat');
save(FileName);

```

REFERENCES

1. FHWA. Highway Statistics 2009 - FHWA [Internet]. 2009 [cited 2011 Mar 15]. Available from: <http://www.fhwa.dot.gov/policyinformation/statistics/2009/>
2. RC2009_full_report.pdf [Internet]. [cited 2011 Jan 15]. Available from: http://www.infrastructurereportcard.org/sites/default/files/RC2009_full_report.pdf
3. Officials AA of SH and T. AASHTO Guide for Design of Pavement Structures. American Association of State Highway and Transportation Officials; 1986.
4. Institute A. Thickness Design: Asphalt Pavements for Highways and Streets. Asphalt Institute (Asphalt Institute Building, College Park, MD 20740); 1991.
5. Shell International Petroleum Company Ltd. Shell pavement design manual: asphalt pavements and overlays for road traffic. Shell International Petroleum Co.; 1978.
6. ARA, Inc., ERES Consultants Division. NCHRP Guide for Mechanistic-Empirical Design. National Cooperative Highway Research Program transportation Research Board National Research Council; 2004.
7. Small KA, Winston C. Optimal highway durability. The American economic review. 1988;78(3):560–9.
8. Madanat S, Prozzi JA, Han M. Effect of performance model accuracy on optimal pavement design. Computer-Aided Civil and Infrastructure Engineering. 2002;17(1):22–30.
9. Deacon JA, Harvey JT, Guada I, Popescu L, Monismith CL. Analytically based Approach to rutting prediction. Transportation Research Record. 2002;1806:9–18.
10. Ali HA, Tayabji SD. Evaluation of mechanistic-empirical performance prediction models for flexible pavements. Transportation Research Record: Journal of the Transportation Research Board. 1998;1629(-1):169–80.
11. El-Basyouny MM, Witzak MW, El-Badawy S. Verification for the Calibrated Permanent Deformation Models for the 2002 Design Guide (With Discussion). Journal of the Association of Asphalt Paving Technologists. 2005;74.
12. Monismith CL, Hicks RG, Finn FN, Sousa J, Harvey J, Weissman S, et al. Permanent deformation response of asphalt aggregate mixes. 1994.
13. Monismith CL, Deacon JA, Harvey JT. WesTrack: Performance models for permanent deformation and fatigue. Report to Nichols Consulting Engineers, Chtd.,

- Pavement Research Center, Institute of Transportation Studies, University of California, Berkeley. 2000;
14. University of Nevada Reno. Facilities | Civil and Environmental Engineering | College of Engineering [Internet]. 2011 [cited 2011 Mar 15]. Available from: <http://www.unr.edu/cee/facilities/index.html>
 15. Mechanistic-Empirical Models 3 [Internet]. [cited 2011 Mar 19]. Available from: <http://www.dynatest.com/papers-7-3.php>
 16. CAPA-3D Finite Elements System [Internet]. [cited 2011 Mar 19]. Available from: <http://capa-3d.org/projects.htm>
 17. Witczak M, Schwartz C, Von Quintus H. NCHRP Project 9-19: Superpave support and performance models management. Interim Rep., Federal Highway Administration and the National Cooperative Highway Research Program, Washington DC. 2001;
 18. Kaloush KE, Witczak MW. Development of a Permanent to Elastic Strain Ratio Model for Asphalt Mixtures. Development of the 2002 Guide for the Design of New and Rehabilitated Pavement Structures. NCHRP 1-37 A. 2000;
 19. Leahy RB. Permanent deformation characteristics of asphalt concrete. Maryland Univ., College Park, MD (USA); 1989.
 20. Kim S, Ceylan H, Heitzman M. Sensitivity study of design input parameters for two flexible pavement systems using the Mechanistic-Empirical Pavement Design Guide. Proc of the 2005 Mid-Continent Transportation Research Symposium [Internet]. 2005. p. 18–9. Available from: <http://www.intrans.iastate.edu/pubs/midcon2005/KimFlexible.pdf>
 21. Timm DH, Newcomb DE, Galambos TV. Incorporation of Reliability into Mechanistic-Empirical Pavement Design. Transportation Research Record. 2000;(1730):73–80.
 22. Alsherri A. Reliability model for pavement performance. Journal of Transportation Engineering. 1988;114:294.
 23. Darter M, Khazanovich L, Yu T, Mallela J. Reliability Analysis of Cracking and Faulting Prediction in the New Mechanistic-Empirical Pavement Design Procedure. Transportation Research Record: Journal of the Transportation Research Board. 2005;1936(-1):150–60.
 24. Thyagarajan S. Improvements to Strain Computation and Reliability Analysis in the Mechanistic-Empirical Pavement Design Guide [Internet]. Washington State University; 2009. Available from: http://www.dissertations.wsu.edu/Dissertations/Spring2009/s_thyagarajan_0042309.pdf

25. ReliabilityintheMEPDGonestatesperspective(Pierce)(1-2007).pdf [Internet]. [cited 2011 Mar 19]. Available from: [http://design.transportation.org/Documents/ReliabilityintheMEPDGonestatesperspective\(Pierce\)\(1-2007\).pdf](http://design.transportation.org/Documents/ReliabilityintheMEPDGonestatesperspective(Pierce)(1-2007).pdf)
26. Thyagarajan S, Muhunthan B, Sivaneswaran N, Petros K. Efficient Simulation Techniques for Reliability Analysis of Flexible Pavements Using the Mechanistic-Empirical Pavement Design Guide. *Journal of Transportation Engineering*. 2011 Mar;
27. Cornell CA. A Probability-Based Structural Code*. *ACI Journal Proceedings*. ACI; 1969.
28. Rackwitz R, Safety JC on S. Practical probabilistic approach to design. *Technical University of Munich, Institut für Bauingenieurwesen III*; 1976.
29. Rackwitz R, Flessler B. Structural reliability under combined random load sequences. *Computers & Structures*. 1978;9(5):489–94.
30. Rosenblueth E. Point estimates for probability moments. *Proceedings of the National Academy of Sciences of the United States of America*. 1975;72(10):3812.
31. Wu YT, Millwater HR, Cruse TA. Advanced probabilistic structural analysis method for implicit performance functions. *AIAA journal*. 1990;28(9):1663–9.
32. Small KA, Winston C, Evans CA. Road work: a new highway pricing and investment policy. *Brookings Inst Pr*; 1989.
33. Mamlouk MS, Zaniewski JP, He W. Analysis and design optimization of flexible pavement. *Journal of Transportation Engineering*. 2000;126:161.
34. Grivas DA, Ravirala V, Schultz BC. State increment optimization methodology for network-level pavement management. *Transportation research record*. 1993;1397(6):25–33.
35. Epps JA, Hand A, Seeds S, Scholz T, Alavi S, Ashmore C, et al. NCHRP Report 455: Recommended Performance-Related Specification for Hot-Mix Asphalt Construction: Results of the WesTrack Project. *Transportation Research Board of the National Academies, Washington, DC*. 2002;283–6.
36. Hudson WR, (US) S and PL, Laboratory UA EWESS and P. State-of-the-Art in Predicting Pavement Reliability from Input Variability. *Soils and Pavements Laboratory, US Army Engineer Waterways Experiment Station*; 1975.
37. Huang YH. *Pavement Analysis and Design*. 2nd ed. Upper Saddle River, NJ, 07458: Prentice Hall; 2004.

38. Lemer AC, Moavenzadeh F. Reliability of Highway Pavements. Paper presented at 50th Annual Meeting of the Highway Research Board. 1971;
39. Darter MI, Hudson WR. Probabilistic Design Concepts Applied to Flexible Pavement System Design. Austin, TX: University of Texas; 1973 May. Report No.: 1-8-69-123-18.
40. Moavenzadeh F, Brademeyer B. A Stochastic Model for Pavement Performance and Management. Proceedings from the 4th International Conference on Structural Design of Asphalt Pavements. 1977;344–65.
41. Riggins M, Lytton R, Garcia-Diaz A. Developing Stochastic Flexible Pavement Distress and Serviceability Equations. 64th Annual TRB Meeting. 1985;
42. George KP, Husain S. Thickness Design for Flexible Pavements. 65th Annual Meeting TRB. 1986;
43. Irick P, Hudson WR, McCullough BF. Application of reliability concepts to pavement design. INTERNATIONAL CONFERENCE ON THE STRUCTURAL DESIGN. 1987.
44. Madsen HO, Krenk S, Lind NC. Methods of structural safety. 1986. Prentice-Hall, Englewood Cliffs, NJ;
45. Ditlevsen O. Structural Reliability and the Invariance Problem. Research Report No. 22, Solid Mechanics Division, University of Waterloo, Waterloo, Canada; 1973.
46. Liu PL, Der Kiureghian A. Multivariate distribution models with prescribed marginals and covariances. Probabilistic Engineering Mechanics. 1986;1(2):105–12.
47. Haldar A, Mahadevan S. Probability, reliability, and statistical methods in engineering design. Wiley; 2000.
48. Ditlevsen O. Narrow reliability bounds for structural systems. Journal of structural mechanics. 1979;7(4):453–72.
49. McDonald M, Mahadevan S. Design optimization with system-level reliability constraints. Journal of Mechanical Design. 2008;130:021403.
50. Bichon BJ, Mahadevan S, Eldred MS. Reliability-based design optimization using efficient global reliability analysis. American Institute of Aeronautics and Astronautics, 1801 Alexander Bell Dr., Suite 500 Reston VA 20191-4344 USA,; 2009.
51. Washington State Department of Transportation. 9.7 Pavement Evaluation - Flexible Pavement Distress [Internet]. 2011 [cited 2011 Mar 17]. Available from: http://training.ce.washington.edu/WSDOT/Modules/09_pavement_evaluation/09-7_body.htm#rutting

52. FHWA. Table FA-6A - Highway Statistics 2009 - FHWA [Internet]. 2009 [cited 2011 Mar 17]. Available from:
<http://www.fhwa.dot.gov/policyinformation/statistics/2009/fa6a.cfm#foot3>
53. Azari H, Mohseni A, Gibson N. Verification of Rutting Predictions from Mechanistic-Empirical Pavement Design Guide by Use of Accelerated Loading Facility Data. *Transportation Research Record: Journal of the Transportation Research Board*. 2008;2057(-1):157–67.
54. ARA, Inc., ERES Consultants Division. NCHRP 09-30A Calibration of Rutting Models for HMA Structural and Mix Design [Internet]. 2011 [cited 2011 Mar 18]. Available from:
<http://144.171.11.40/cmsfeed/TRBNetProjectDisplay.asp?ProjectID=965>
55. Khattak MJ, Baladi GY. Fatigue and permanent deformation models for polymer-modified asphalt mixtures. *Transportation Research Record: Journal of the Transportation Research Board*. 2001;1767(-1):135–45.
56. Forrester AIJ, Sobester A, Keane AJ, Astronautics AI of A and. *Engineering design via surrogate modelling: a practical guide*. Wiley; 2008.
57. Jeffreys H. *Theory of Probability*. Oxford: Clarendon Press; 1961.
58. Devore JL. *Probability and Statistics for Engineering and the Sciences*. united States: Duxbury Thompson Learning; 2000.
59. Muthadi NR, Kim YR. Local calibration of mechanistic-empirical pavement design guide for flexible pavement design. *Transportation Research Record: Journal of the Transportation Research Board*. 2008;2087(-1):131–41.
60. Hoegh K, Khazanovich L, Jense M. Local Calibration of Mechanistic-Empirical Pavement Design Guide Rutting Model. *Transportation Research Record: Journal of the Transportation Research Board*. 2010;2180(-1):130–41.
61. Ayyala D, Chehab GR, Daniel JS. *Sensitivity of MEPDG Level 2 and 3 Inputs Using Statistical Analysis Techniques for New england States*. Washington, D.C.;
62. Rada G, Witzak MW. . Comprehensive Evaluation of Laboratory Resilient Moduli Results for Granular Materials. *Transportation Research Record*. 1981;810:23–33.
63. Thompson MR. ILLI-PAVE Based Full-Depth Asphalt Concrete Pavement Design Procedure. *Sixth International Conference on Structural Design of Asphalt Pavements*. 1:13–22.
64. Thompson MR, Elliott RP. ILLI-PAVE Based Response Algorithms for Design of Conventional Flexible Pavements. *Transportation Research Record; Transportation Research Board*. 1985;1043:50–7.

65. McInvale HD, McDonald MP, Mahadevan S. A System of Systems Approach for Effects-Based Operational Planning Under Uncertainty. *Military Operations Research*. 2011;16(3):31–46.
66. *Steel Construction Manual*. American Institute of Steel Construction, Inc.; 2005.
67. 318AC, Institute AC. Building code requirements for structural concrete (ACI 318-05) and commentary (ACI 318R-05). American Concrete Institute; 2005.
68. Nielsen HB, Lophaven SN, Søndergaard J. DACE [Internet]. 2007 [cited 2011 Apr 15]. Available from: <http://www2.imm.dtu.dk/~hbn/dace/>
69. Retherford JQ, McDonald M. Estimation and Validation of Gaussian Process Surrogate Models for MEPDG-Based Sensitivity Analysis and Design Optimization. *Transportation Research Record: Journal of the Transportation Research Board*. 2011;