Fall 2019

A Data Mining Methodology for Vehicle Crashworthiness Design

Xianping Du

A DATA MINING METHODOLOGY FOR VEHICLE CRASHWORTHINESS DESIGN

By

Xianping Du

A Dissertation Submitted to the

Department of Mechanical Engineering in the College of Engineering

in Partial Fulfillment of the Requirements

for the Degree of

Doctoral of Philosophy

in

Mechanical Engineering

Embry-Riddle Aeronautical University

Daytona Beach, Florida

Fall 2019

A DATA MINING METHODOLOGY FOR VEHICLE CRASHWORTHINESS DESIGN

By

Xianping Du

This dissertation was prepared under the direction of the candidate's dissertation Committee Chair, Dr. Feng Zhu, Assistant Professor, Daytona Beach Campus, and Dissertation Committee Members Dr. Fady Barsoum, Professor, Daytona Beach Campus, Dr. Sathya Gangadharan, Professor, Daytona Beach Campus, Dr. Hong Liu, Professor, Daytona Beach Campus and Dr. Hongyi Xu, Assistant Professor, University of Connecticut, and has been approved by the members of the Dissertation Committee. It was submitted to the Department of Mechanical Engineering in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Mechanical Engineering.

Dissertation Review Committee:

Feng Zhu, Ph.D.
Committee Chair

Fady Barsoum, Ph.D.
Committee Member

Sathya Gangadharan, Ph.D.
Committee Member

Hong Liu, Ph.D.
Committee Member

Hongyi Xu, Ph.D.
Committee Member

Eric Coyle, Ph.D.
Ph.D. Program Coordinator,
Mechanical Engineering

Eduardo Divo, Ph.D.
Department Chair,
Mechanical Engineering

Maj Mirmirani, Ph.D.
Dean, College of Engineering

Lon Moeller, ~~Ph.D.~~ J.D.
Senior Vice President for Academic Affairs and Provost

Date: 12/05/2019

Dedication

I would like to dedicate this dissertation to my parents for their selfless love.

Acknowledgments

I would like to take this opportunity to thank my advisor, Dr. F. Zhu, for his invaluable direction during the course of my study at ERAU and in the preparation of my dissertation. He not only gave me precious feedback on my research but also helped with my personal life during the past four years. His rigorous attitude and high standards for academic research shaped my style. I also thank Dr. H.Y. Xu, one of my committee members, from the University of Connecticut for offering me a precious summer internship opportunity, which inspired me and enriched my knowledge.

I would also like to thank the other dissertation committee members, Dr. F. Barsoum, Dr. S. Gangadharan and Dr. H. Liu, for their time and comments. I want to thank Dr. E. Divo, the department chair, as well. He has been so nice as to give any help he can. Thanks are also given to my current program coordinator, Dr. E. Coyle, for his help with the dissertation defense and graduation and our department office assistant, Mrs. J. Gibbs, for her help on the paperwork related to the study. I extend thanks to my friends as well for your company and encouragement. Particularly, I thank my girlfriend, Yi, for her bright smile, understanding and support. Embry-Riddle Aeronautical University, the Federal Aviation Administration, MIT and the Ford Motor Company are also appreciated for their financial support.

Last, but most importantly, my parents are worthy of the greatest thanks. They have given me the freedom and selfless love to pursue my dream. At this moment, they must be proud of their only son. He is brave and strong enough to face future life. He also hopes they are happy always.

Table of Contents

List of Figures

.

List of Tables

List of Symbols

Chapter 3

avgstiff	Average stiffness of components
A	A split point for decision tree induction
D	Current dataset
D_j	The *j*th sub-dataset of a split
G	Information gain
GR	Gain ratio
Index	A split point selection criterion
$Info$	Information or entropy
$Info_A$	Expected information required to partition using attribute A
m	Number of classes
N_v	Number of resulting partitions
p_i	The portion of *i*th class in the current dataset
SpliL	Split points list
$Splitf_A$	Split information
v	Number of partitions
$\|\cdot\|$	Dataset size

Chapter 4

AN	Slope segment angle of the dimension-based S-beam
CL	Cross-section length of the dimension-based S-beam
C and P	Two strain rate dependent constants
DR	Bottom segment radius of the dimension-based S-beam
E	Young's modulus
E_{int}	Total internal energy absorbed
E_T	Tangent modulus

H	Height of the dimension-based S-beam
M	Mass of the S-beam
SEA	Specific energy absorption
T	The thickness of dimension-based S-beam
UL	Upper segment length of the dimension-based S-beam
UR	Upper segment radius of the dimension-based S-beam
X	Design variable value before normalization
$\overline{X}$	Design variable value after normalization
X_{max}	Upper bound of design variable values
X_{min}	Lower bound of design variable values
$\dot{\varepsilon}$	Effective strain rate
P	Density of S-beam material
σ_0	Quasi-static yield stress
σ_{Yd}	Dynamic yield stress
Y	Poisson's ratio

Chapter 5

$c_{\bullet i}$	The *i*th centralized vector among different samples
$\overline{c}_{\bullet i}$	Mean of the *i*th dimension of different samples in centralized dataset
$\mathrm{cov}(\bullet,\bullet)$	Element of covariance matrix
$\mathbf{C}_{n\times m}$	Centralized matrix
CFE	Crush force efficiency
$\mathbf{COV}(\bullet)$	Covariance matrix
$d_{i\bullet}$	The *i*th element of geometry dataset
$\mathbf{D}_{n\times m}$	Geometry dataset with n samples and m dimensions
$\overline{\mathbf{D}}_{n\times m}$	Mean matrix of dataset $D_{n\times m}$
$\mathbf{D}^{*}_{n\times m}$	Approximation of the initial geometry dataset

E_{int}	Internal energy absorbed by components
$F_{\max}$	Maximum impact force
F_{mean}	Mean impact force
k	Number of selected principal components
m	Number of sample dimensions
n	Number of samples
p_i	The ith element of principal component matrix
$\mathbf{P}_{m\times m}$	Principal component matrix
$\mathbf{P}_{k\times m}$	Principal component matrix with selected k elements
s_{ij}	Element of principal component score matrix in ith row and jth colume
S_E	Maximum crush distance
$\mathbf{S}_{n\times m}$	PCS matrix
$\mathbf{S}_{n\times k}$	PCS matrix with selected k elements

Chapter 6

b	Offset constant of support vector machine
b_j	Offset of in jth hidden neuron
$\boldsymbol{C}$	Weight of error item in the support vector training objective
d	Displacement of the ten-bar planer truss system
deg	Degree of polynomial dot kernel
$D_d(\bullet)$	Output of dth decision tree
f	Real response
$\tilde{f}$	Surrogate predicted response
$\mathcal{GP}$	Gaussian distribution
$k(\bullet)$	Covariance function
$\mathbf{K}$	Kernel function
LCB	Lower confidence bound

MXAE	Maximum absolute error
n_h	Number of hidden neurons
N	Number of training or test data points
$\mathrm{N_R}$	Number of decision trees in random forest
offset	Offset added to the scaled dot product
$\mathbf{R}^m$	Response m-dimensional space
$\mathbf{R}^n$	Design variable n-dimensional space
RMSE	Root means square error
$\hat{s}$	Posterior standard deviation
T	Training computational time
u_j	Output of *j*th hidden neuron
w_{ij}	Weight of *i*th variable on *j*th hidden neuron
w_{jo}	Weight of *j*th hidden neuron in output neuron
$\boldsymbol{X}$	Input features/ design variables
$\boldsymbol{X}$	Input features of training dataset
$\boldsymbol{X}^*$	Input features of test dataset
$\boldsymbol{Y}$	Output features/ responses
α , α^*	Lagrange multipliers
δ	Random noisy variable
ε	The distance of bounds to in support vector machine
ξ , ξ^*	Slack variables for points out of bounds in support vector machine
σ	Weight of nodes with different distances to the current node
$\mu(\bullet)$	Mapping function of support vector machine
$\hat{\mu}$	Posterior mean
$\boldsymbol{\omega}^T$	Linear coefficients of support vector machine
φ	weight of standard deviation in lower confidence bound calculation
$<\bullet>$	Dot product
$\|\bullet\|$	Euclidean distance

Chapter 7

A_{ij}	A precise value in traditional dataset without uncertainty
A_{ij}^{CL}	Lower bound of an interval for the *j*th feature's *i*th sample
A_{ij}^{CU}	Upper bound of an interval for the *j*th feature's *i*th sample
A_{ij}^{m}	Central value of an interval
ACC	Classification accuracy
D_{sp}	The *sp*th subset due to a split point
n_c	Number of tuples in the current dataset
n_l	Number of leaf nodes
n_{Lt}	Number of tuples with L_t label in the current dataset
n_m	Size of training dataset
n_{sp}	Number of tuples in the subset due to a split point
P_g	Predicted probability of an tuple to be good design
P_i^c	Summation of correctly classified samples tuple probability
p_i^t	Tuple probability of *i*th sample
$p_i^{t_{Lt}}$	Tuple probability of *i*th tuple with label L_t in the current dataset
$p_i^{t_{sp}}$	Tuple probability of *i*th tuple in the subset due to a split point
$\overrightarrow{p_i^L}$	Label probability vector in *i*th leaf node
$\overrightarrow{p_L}$	Uncertain decision tree predicted label probability
p_L^t	The tuple probability of a sample assigned to the left branch
P^{Lt}	Label probability in the current dataset
P_m	Predicted probability of an tuple to be intermediate design
P_p	Predicted probability of an tuple to be poor design
p_R^t	The tuple probability of a sample assigned to the right branch
P_s^t	Tuple probability of an observation assigned to *s*th leaf in validation
$\boldsymbol{R}$	A ratio for defining the half of an interval by multiplying a constant
$\boldsymbol{S_j}$	A selected split point of *j*th input feature

List of Abbreviations

ANN	Artificial neural network
ATD	Anthropomorphic test dummy
CAE	Computer aided engineering
CART	Classification and regression tree
CFE	Crush force efficiency
CPI	Critical parameter identification
DDR	Design domain reduction
DM	Data mining
DMM	Data mining method
DOE	Design of experiment
DT	Decision tree
DTUD	Decision tree for uncertain datasets
EAS	Energy absorbing structures
EI	Expected improvement
FE	Finite element
FMVSS	Federal motor vehicle safety standard
GA	Genetic algorithm
GP	Gaussian process
GPR	Gaussian process regression
IIHS	Insurance Institute for highway safety
KBD	Knowledge-based design
LCB	Lower confidence bound
LHM	Latin hypercube method
LP	Label probability
LS-SVM	Least square-support vector machine
Max TN	Maximum numbers of terminal nodes
MCM	Monte Carlo method
Min TS	Minimum terminal node size
ML	Machine learning

MLA	Machine learning algorithm
MXAE	Maximum absolute error
NCAC	National crash analysis center
NCAP	New car assessment program
NF	Number of randomly selected features for each split of decision tree
NHTSA	National highway transportation safety administration
NURBS	Non-uniform rational basis spline
OMcT	Thin-walled octagonal multi-cell tube
PC	Principal component
PCA	Principal component analysis
PCS	Principal component score
PDF	Probability density function
PRSM	Polynomial response surface model
RBF	Radial basis function
RBNN	Radial basis neural network
RFR	Random forest regression
RMSE	Root mean square error
RSM	Response surface method
SD	Standard deviation
SEA	Specific Energy absorption
ShB	Thin-walled S-shaped beam
SM	Surrogate model
SMBO	Sequential model-based optimization
SSE	Sum of squares for error
SVM	Support vector machine
SVR	Support vector regression
TbPT	Ten-bar planer truss
TP	Tuple probability
TqA	Torque arm

Abstract

Researcher: Xianping Du

Title: A Data Mining Methodology for Vehicle Crashworthiness Design

Institution: Embry-Riddle Aeronautical University

Degree: Doctor of Philosophy in Mechanical Engineering

Year: 2019

This study develops a systematic design methodology based on data mining theory for decision-making in the development of crashworthy vehicles. The new data mining methodology allows the exploration of a large crash simulation dataset to discover the underlying relationships among vehicle crash responses and design variables at multiple levels and to derive design rules based on the whole-vehicle safety requirements to make decisions about component-level and subcomponent-level design. The method can resolve a major issue with existing design approaches related to vehicle crashworthiness: that is, limited abilities to explore information from large datasets, which may hamper decision-making in the design processes.

At the component level, two structural design approaches were implemented for detailed component design with the data mining method: namely, a dimension-based approach and a node-based approach to handle structures with regular and irregular shapes, respectively. These two approaches were used to design a thin-walled vehicular structure, the S-shaped beam, against crash loading. A large number of design alternatives were created, and their responses under loading were evaluated by finite element simulations. The design variables and computed responses formed a large design dataset. This dataset

was then mined to build a decision tree. Based on the decision tree, the interrelationships among the design parameters were revealed, and design rules were generated to produce a set of good designs. After the data mining, the critical design parameters were identified and the design space was reduced, which can simplify the design process.

To partially replace the expensive finite element simulations, a surrogate model was used to model the relationships between design variables and response. Four machine learning algorithms, which can be used for surrogate model development, were compared. Based on the results, Gaussian process regression was determined to be the most suitable technique in the present scenario, and an optimization process was developed to tune the algorithm's hyperparameters, which govern the model structure and training process.

To account for engineering uncertainty in the data mining method, a new decision tree for uncertain data was proposed based on the joint probability in uncertain spaces, and it was implemented to again design the S-beam structure. The findings show that the new decision tree can produce effective decision-making rules for engineering design under uncertainty.

To evaluate the new approaches developed in this work, a comprehensive case study was conducted by designing a vehicle system against the frontal crash. A publicly available vehicle model was simplified and validated. Using the newly developed approaches, new component designs in this vehicle were generated and integrated back into the vehicle model so their crash behavior could be simulated. Based on the simulation results, one can conclude that the designs with the new method can outperform the original design in

terms of measures of mass, intrusion and peak acceleration. Therefore, the performance of the new design methodology has been confirmed.

The current study demonstrates that the new data mining method can be used in vehicle crashworthiness design, and it has the potential to be applied to other complex engineering systems with a large amount of design data.

Chapter 1 Introduction

1.1 Motivation

Vehicle design requires complex systems engineering, where a large number of attributes must be systematically considered. Among these attributes, safety ranks as the top priority. In recent years, the auto industry has experienced greater demand than ever before from customers, governments, and media to achieve five-star safety ratings for vehicles with satisfactory crashworthiness performance. The terminology "crashworthiness" denotes a measure of plastic deformation of the vehicular structure and its maintenance of a sufficient survival space for its occupants during crashes involving reasonable deceleration loads.

To be compliant with crashworthiness design requirements, the vehicle structure must manage the crash energy effectively in various crash modes likely to be encountered in fleet service. Crash energy management means controlling, by design, the dynamic behavior of multiple subsystems in the very violent and complex environment of a collision. Many design factors, such as load path, structural deformations and component collapse sequence as well as vehicle size, weight, geometry, and more must be considered simultaneously. To design such a complicated system, efficient design methods are necessary.

The energy absorbing structures (EASs) on the vehicle are essentially designed as thin-walled beams or columns assembled to sustain plastic deformation when crushed in a crashworthy car. Traditional engineering structural analysts typically study structures using elastic analysis to design them to withstand service loads without yielding or

collapsing. Automotive structures, however, must meet all previously mentioned service load requirements in addition to deforming plastically in a short time (milliseconds) and within a limited crush space to absorb the crash energy in a controllable manner. During an impact, a large number of such structures work together as a system to dissipate the crash energy. When one component is changed in the design, the other parts must be revised or redesigned accordingly. This type of complex interaction among different components is often ***invisible*** and makes the design problem ***highly nonlinear***. Without sufficient knowledge about interrelations of different components, decision-making to achieve an effective and efficient design for this complex system would be difficult.

1.2 General vehicle crashworthiness design process

The whole-vehicle design can generally be implemented as a four-step procedure, as shown in Figure 1.1. In Step 1, the overall performance (e.g. maximum speed, dimensions and weight) is defined as the design requirements. Then, to achieve these objectives, in Step 2, the overall layout design is performed for all subsystems, such as body, electric system, engine and chassis. This is followed by the detailed design of each subsystem in Step 3. After the detailed design is completed and verified, the vehicle is manufactured and integrated.

In Step 2, a coarse design of the vehicle crashworthiness system is implemented using various methods, such as a lumped mass-spring model (Choi et al., 2019), multi-body model (Carvalho and Ambrósio, 2010, Ambrosio and Dias, 2007), vehicle frame with plastic joint model (Gui et al., 2018, Zuo and Bai, 2016) or topology optimization

(Duddeck and Volz, 2012). The design of each component in this system is carried out in Step 3 to satisfy the safety requirements, and this is the focus of the present study.

The vehicle safety requirements, such as Federal Motor Vehicle Safety Standards (FMVSS) and those from the New Car Assessment Program (NCAP) and the Insurance Institute for Highway Safety (IIHS) have defined many specific performance indices for vehicle crash safety (NHTSA, 1997, IIHS, 2002, NHTSA, 2008). These indices are related to the crash peak force and acceleration, passenger compartment intrusion and Anthropomorphic Test Dummy (ATD) responses. To comply with these criteria, the vehicle energy absorption structures must be designed in a top-down sequence at (1) the main EAS system level to meet the overall safety requirements and then (2) the component level with the proper boundary conditions. At both levels, large design datasets are generated, since a large number of crash simulations or tests are carried out to verify the design. Although existing Computer-Aided Engineering (CAE) based design methodologies can significantly speed up product development cycles and reduce the cost, they are not suited to achieve a fast, efficient and accurate design for complex systems due to their limited ability to explore information from large design datasets to link different levels. A detailed literature review and an analysis of the existing methods are presented in Chapter 2.

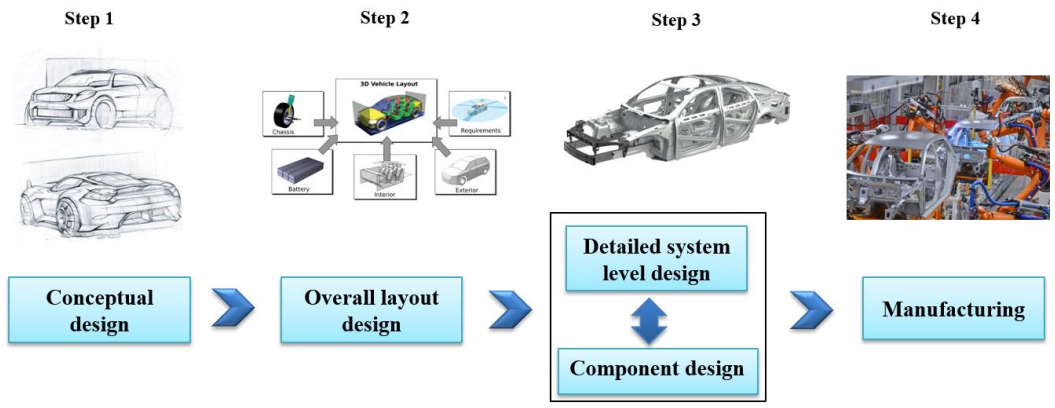


Figure 1.1. General procedure of vehicle design.

1.3 Research tasks

The objectives of this research are as follows: (a) to provide a systematic design methodology by exploring large crash simulation datasets to discover the underlying complicated relationships between response and design variables at multiple levels (primary energy-absorbing features at system, components, and geometric levels) and to derive design rules based on the whole-vehicle body safety requirements to make decisions about the component and subcomponent-level design and (b) to demonstrate the feasibility of this method by applying it to a numerical or CAE passenger car model. Uncertainty is also considered and quantified in the design process.

1.4 Dissertation outline

The organization of this dissertation is illustrated in Figure 1.2. The remaining eight chapters are outlined briefly below.

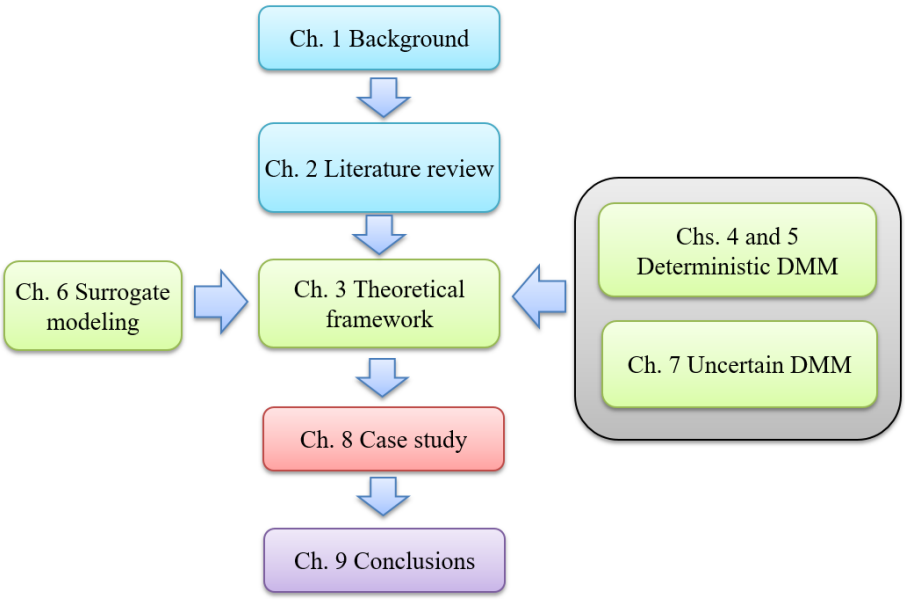


Figure 1.2. Organization of the dissertation.

In Chapter 2, a literature review is made on the existing design methodologies for vehicle crashworthiness. A detailed comparison and analysis are presented.

To overcome the limitations identified in Chapter 2, Chapter 3 proposes a new methodology based on data mining (DM) theory that can design a multilevel, complicated mechanical system such as a crashworthy vehicle.

The new method is implemented for the detailed design of critical energy-absorbing structures in a vehicle with a regular and irregular geometries implemented in Chapters 4 and 5, respectively. The modeling procedures in both cases are described in detail.

Chapter 6 is a study of an engineering method known as surrogate modeling, which with high accuracy can partially replace expensive numerical simulations. A detailed

parametric study is performed to identify the most suitable surrogate model for the present structural design problem and determine its parameters.

In Chapter 7, a new algorithm is developed to quantify the uncertainty in the design process. The basic algorithm is validated and integrated into conventional data mining.

In Chapter 8, the new methodology developed in this study is applied in the crashworthiness design of a typical passenger car to verify its rationality and performance.

The whole study is summarized in Chapter 9, and future studies are suggested.

Chapter 2 Literature review

2.1 Introduction

In this chapter, three main methods for vehicle crashworthiness design are reviewed: namely, iteration, population and knowledge-based approaches. Their design strategy, application and performance are compared.

2.2 Iteration-based design

This method is a traditional mechanical design approach that includes several trial-and-error loops (Zhu et al., 2012, Weng et al., 2010) as presented in Figure 2.1. The process starts with design variables, which are usually geometric features or material properties of vehicular components. Within the specified range of design variables (termed the design space), a design alternative is generated. A vehicle crash analysis is then performed (frequently by means of finite element (FE) simulations) and the predicted crush responses of the whole vehicle, such as energy absorption, intrusion and peak force and acceleration, are compared with the design requirements. If the performance is satisfactory, the current design can be accepted as final. Otherwise, the values of design variables are changed to form a new design alternative, and the analysis is run again. This iterative cycle is repeated until a satisfactory design is identified.

It is often the case that only a small number of design variables can be modified in each iteration, since this method depends on intensive human intervention. The final result is highly dependent on the initial design, which increases the reliance on the designer's intuition, experience and skill. Therefore, such a design approach with trial-and-error

loops is of low efficiency when designing a complex system such as a vehicle and is not likely to lead to the best possible design.

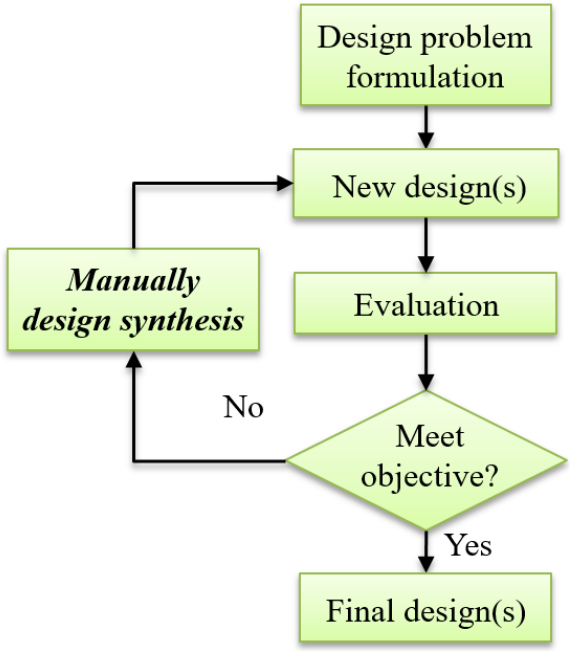


Figure 2.1. The general procedure of the trial-and-error method.

2.3 Population-based design

Automatic population-based approaches are widely used for vehicle structure optimization. Depending on the nature of the computational model used, these can be categorized into simulation-based and surrogate-based approaches.

2.3.1 Simulation-based optimization

Simulation-based optimal design is realized by integrating numerical simulations, such as the FE model, with an automatic heuristic searching algorithm. The workflow is illustrated in Figure 2.2. With this method, a large number of design alternatives are

created within the design space, and each one is termed a design of experiments (DOE). These DOEs are computed using FE analysis subject to the predefined constraints. Based on the evaluation results, the heuristic algorithm was used to determine the population of the next generation based on current and previous generations until the final optimum or the termination criterion is reached. Many heuristic algorithms can be used in this step, such as evolutional, genetic, simulated annealing and particle swarm algorithms.

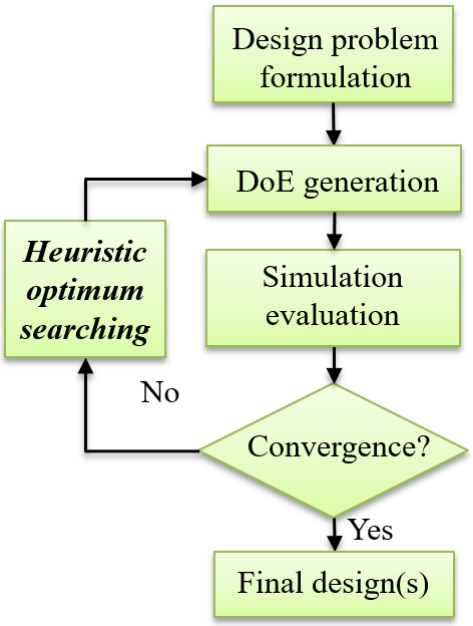


Figure 2.2. The general procedure of the simulation-based optimization approach.

Unlike the traditional trial-and-error method, this method automatically realizes an optimal or ultimate design search process under the control of a heuristic algorithm. However, due to the high computational cost of FE modeling of vehicle crashes, it is time-consuming to generate a large number of design alternatives and perform the simulations. In addition, for highly nonlinear problems, it is difficult to realize the optimization process. Applications of this method to crashworthiness design have been

reported in the literature (Yildiz and Solanki, 2012, Hou et al., 2012, Liao et al., 2008, Fang et al., 2005, Fu and Sahin, 2004, Sobieszczanski-Sobieski et al., 2001).

2.3.2 Surrogate-based optimization

To reduce the computational cost of FE simulations, surrogate models (SMs) can be used to supplement or partially replace the simulations. The general procedure for this is illustrated in Figure 2.3. The surrogate model is a mathematical model that learns the relationship between the design variables and the response from a dataset and then predicts the responses of new designs. A wide variety of surrogate modeling techniques have been reported in the literature, such as the polynomial response surface model (PRSM) (Shimoyama et al., 2009), Kriging (Qian et al., 2019, Fan et al., 2019), radial basis functions (RBFs) (Kitayama et al., 2013), artificial neural networks (ANNs) (Roshanian et al., 2018) and support vector regression (SVR) (Prado et al., 2018). By replacing the FE simulation with the appropriate surrogate model, the computational cost of the design can be greatly reduced. Using this approach, medium- to high-dimensional design problems can also be brought to converge more efficiently and effectively by increasing the number of heuristic generations.

These surrogate modeling techniques and their variations, for example, the sequential surrogate model (Kitayama et al., 2013, Hutter et al., 2011), the surrogate ensemble (Ferreira and Serpa, 2017, Goel et al., 2007) and surrogate model automatic selection (Mehmani et al., 2017, Bagheri et al., 2016), have been successfully applied to crashworthiness design (Ozcanan and Atahan, 2019, Fan et al., 2019, Roshanian et al., 2018). Although surrogate models can significantly speed up the product development

process, their accuracy is sometimes a concern, especially in highly nonlinear crash problems.

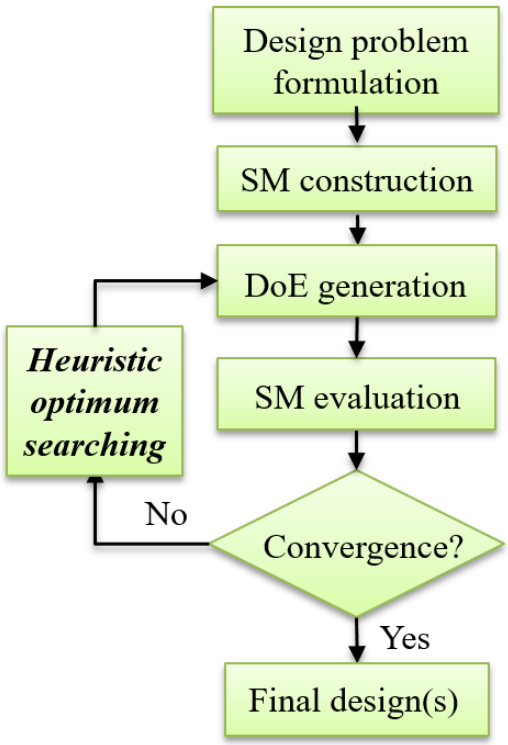


Figure 2.3. The general procedure for the surrogate-based optimization approach.

From the point of view of system design, the population-based approaches described above have numerous limitations:

(a) Lack of capability in revealing intrinsic interrelations or coupling effects among components or design variables:

These methods cannot explain the interrelationship or coupling effect among various components implied in design datasets. Since in complex systems such as a vehicle, the components are integrated by complex joints, they have a coupling effect and cannot be deemed independent. Some components are “parents” whose behaviors

can influence “children” components and then the overall response of the vehicle. In design practice, the parent components must be determined first, followed by the children components (Huang, 2002). Such intrinsic coupling effects between components or design variables are complicated and are usually implicit and hidden in the vast amount of simulation data. Without the knowledge of these effects, however, it is challenging to generate suitable designs and decision-making rules to tune components in the right sequence.

(b) Difficulty linking the detailed geometric variations of each component to the overall vehicle response:

In all of the studies reviewed in Section 2.3, the design variable is limited to the wall thickness of each beam or column, which by itself is not able to fully describe the detailed profile of a component. In other words, the relationship between subcomponent-level geometric variations and vehicle level response is not established. Since a vast number of variables are needed to describe the geometry of a component accurately, the inclusion of all these variables will significantly increase the number of DOEs and generally make the computational cost unaffordable.

(c) Intensive human intervention if a bottom-up design strategy is used:

Some attempts have been made to address limitation (b) above by decomposing a complex structure into several substructures and then designing these substructures separately. For example, Kim et al. (2001), Chase et al. (2012) and Gandikota et al. (2015) proposed applying a multi-hierarchical strategy to design complex systems (Gandikota et al., 2014, Kim and Wierzbicki, 2001, Chase et al., 2012). Specifically,

the vehicle is first decomposed into several functional systems and these are decomposed further into subsystems. The design starts from the subsystem level in and progresses in a bottom-up sequence. Once a lower-level design is completed, the resulting design information is passed to the higher levels, then up to the top level. However, to achieve the final design, a large number of iterations between different levels are needed, which means intensive intervention by the designer and hence low efficiency (Kim et al., 2003). Again, in population-based approaches the coupling effects among components and design variables cannot be identified, thus making it extremely difficult to establish efficient design rules to determine the sequence of component design. As a result, it is very costly to establish a decision-making workflow that moves from the vehicle-level functional requirements to the detailed component design.

2.4 Knowledge-based design (KBD)

As indicated above, the existing design methodologies have limitations that hamper the efficient development of a complicated engineering product such as a crashworthy vehicle. All three limitations identified in Section 2.3 are associated with a lack of ability to mine information from large datasets. To overcome these limitations, a new design approach is required that is able to discover the desired relationships by exploring a large amount of design data. With this knowledge obtained, design rules can be generated to make appropriate decisions in a top-down sequence without intensive human intervention, as demonstrated in this study.

As one of the most promising solutions to overcoming the limitations of the two traditional methods, knowledge-based methods have been developed that can extract useful knowledge from design datasets and apply it to improve the design process for complex systems. This type of approach enables designers to condense a large amount of data, discover hidden patterns, reveal new relations and patterns and extract anticipative and useful information implicit in large datasets. Many knowledge-based design (KBD) methodologies have been used to help the design of numerous products, such as airfoils (Mosavi, 2014a, Mosavi, 2014b), vehicle side frames (Kim and Ding, 2005), bridges (Burrows et al., 2011), rock tunnel boring machines (Tao et al., 2009), ships (Yang et al., 2012) and vehicle restraint systems (Zhao et al., 2010). These methods are used not only for response prediction but also for information exploration and decision-making for understanding the problem. It has been widely accepted that KBD is superior traditional design optimization approaches, which are focused on prediction only (Huang, 2006). KBD can potentially have a great impact on the development of complex systems. However, to the best of our knowledge, KBD has not yet been implemented in the systematic design of a complex system such as a vehicle.

2.5 Summary

In this chapter, three methods that have been used for vehicle crashworthiness design were reviewed and compared. The iteration-based method, which features a large number of trial-and-error iterations, is unable to deal with highly complex vehicle crashworthiness design problems due to its low efficiency and effectiveness. The population-based methods generate a group of evaluated designs by means of either numerical simulations or surrogate models. Such methods can automate the design

process and achieve an optimal design by applying a wide range of optimization algorithms. However, because they cannot mine large datasets, they are unable to discover the implicit complex interrelationships of components and design variables that would allow them to derive design rules for a complex multilevel system.

Data-driven KBD approaches can be used to overcome the limitations related to the aforementioned methods and to discover the desired relationships by exploring a large amount of design data. This study proposes and demonstrates a knowledge-discovery method from which design rules can be generated to make appropriate decisions in a top-down sequence without intensive human intervention.

Chapter 3 Vehicle crashworthiness design based on data mining method (DMM) – The theoretical framework

3.1 Introduction

As indicated in the literature review, the existing optimal design methodologies have limitations that hamper the efficient development of a crashworthy vehicle. There is a great need for a new knowledge-based systematic design method that can discover the desired relationships by exploring a large amount of design data. With this knowledge obtained, design rules can be generated to make correct decisions in a top-down sequence without extensive human intervention. Using these functional requirements, the proposed design methodology is based on a *data mining method* (DMM) (Han et al., 2011) that has been developed to explore and analyze, by automatic or semiautomatic means, large quantities of data to discover meaningful patterns and rules. The data mining tools enable designers to filter a large amount of data, discover hidden data, reveal new relations and patterns and extract anticipative and useful information implicit in large datasets. Section 3.2 briefly introduces the basic theory of DMM. Then, in Section 3.3, the workflow of this method as applied in vehicle crashworthiness design is outlined step by step.

3.2 Theoretical introduction

Data mining is used to explore hidden patterns and rules in data. These rules are often implicit but are critical for decision-making. For multilevel system design, these rules can also reveal the hidden interrelations and can bridge the gaps between different levels. In this work, a rule-generation data mining algorithm, **decision tree**, is used to achieve these objectives.

The decision tree algorithm is characterized by a tree-like model with a number of decision-making rules, as demonstrated by a simple credit evaluation process in Figure 3.1. Generally, a decision tree is composed of leaf and non-leaf nodes and their connections. The top non-leaf node is called the root node, which is also the origin of each branch. By following the path from the top root node to a bottom leaf node linked by connections, the implicit decision-making rules are interpreted explicitly in an if-then manner. In Figure 3.1, by following the leftmost branch, the salary of a loan applicant is evaluated first. If the applicant's salary is not more than $100k per year and if the applicant's profession is teacher, the credit can be labeled as "good." The other rules corresponding to the labels in their leaf nodes can be generated by following each branch in a similar fashion.

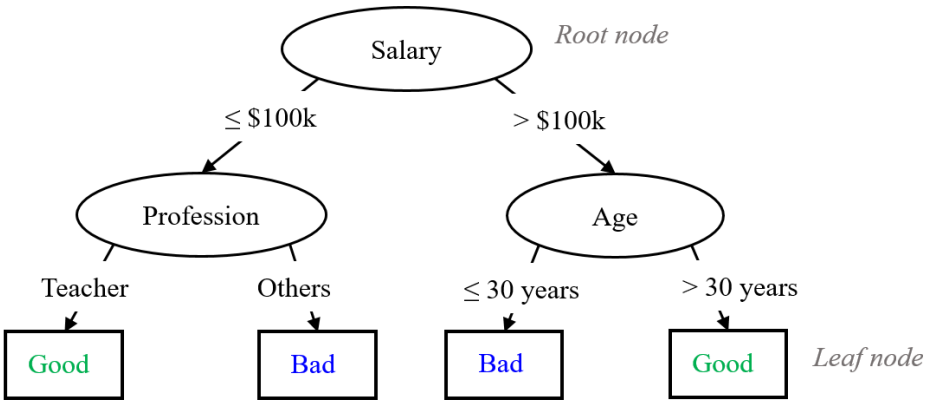


Figure 3.1. A typical decision tree for an individual's credit evaluation.

In a similar manner, using engineering data, the interrelations or patterns in design attributes may be identified and used to efficiently determine the performance of a new design. With these attributes, decision-making rules can be generated using a similar tree-like structure to design components in a top-down sequence.

To induct a decision tree from data, many algorithms are available, including Iterative Dichotomiser 3 (ID3), C4.5 (extension of ID3), Classification and Regression Tree (CART), Naive Bayes Tree (NBTree), Random Forest and Regression tree representative (REPTree) (Han et al., 2011, Witten et al., 2011). In this study, the C4.5 algorithm is used because it has superior performance and is widely applied in engineering design. Decision tree training is a process to determine the splitting points of each non-leaf node recursively. The goal is to split the current dataset and make its subsets as pure as possible. Several indices have been developed to quantify the purity of a dataset label, such as the information gain (used in ID3), information gain ratio (used with C4.5) and Gini index (used with CART). The information gain ratio *GR(A)* is used in C4.5 and can be calculated using the following equation:

$$GR(A) = \frac{G(A)}{Splitf_A(D)} \ , \qquad (3.1)$$

where *G*(*A*), defined in Eq. 3.2, is the information gain to split the current dataset (*D*) by the attribute or splitting point *A*. $Splitf_A(D)$ is the "split information," which is a measure of the potential information generated from the splitting of current dataset *D* into two subsets. It can be calculated using Eq. 3.3.

$$G(A) = Info(D) - Info_A(D) \ , \qquad (3.2)$$

$$Splitf_A(\mathrm{D}) = -\sum_{j=1}^{N_v} \frac{|D_j|}{|D|} \times \log_2(\frac{|D_j|}{|D|}) \ , \qquad (3.3)$$

where D_j is the *j*th sub-dataset of dataset *D* corresponding to a split. |·| represents the dataset size. N_v (2 in this study) is the number of resulting partitions when the set is split based on the selected attribute or splitting point. *Info(D)* and $Info_A(D)$, as defined in Eq.

3.4 and 3.5, respectively, are the information or entropy of the original dataset and the expected information required when partitioning by *A*:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i) \ , \qquad (3.4)$$

$$Info_A(D) = \sum_{j=1}^{N_v} \frac{|D_j|}{|D|} info(D_j) \ , \qquad (3.5)$$

where *m* is the number of label classes, and p_i is the portion of the *i*th class in the current dataset. Using the information gain ratio as the index in C4.5, a decision tree can be generated in a recursive manner. The detailed algorithm is shown in Figure 3.2.

Algorithm: Generate a decision tree
Input: *D*: Labeled dataset
SpliL: Split points list:
Index: Split point selection criterion:
Output: a decision tree model
Method:
For the current node *N*,
(1) if tuples in *D* are all of the same class *C*, then
    return *N* as a leaf node labeled with the class *C*;
(2) if *SpliL* is empty, then
    return *N* as a leaf node labeled with the majority class in *D*;
(3) else apply the *index* to find the best splitting point for a binary split;
    for each of two outcomes, let D_j be the sub-dataset in *D* satisfying outcome *j* (*j* = 1, 2);
  (31) if *Dj* is empty, attach a leaf labeled with the major class in *D* to node *N*;
  (32) else attach the child nodes returned by the splitting point to node *N*;
  End
Return *N* and its next generation.

Figure 3.2. A general procedure for a binary decision tree construction.

3.3 The data mining method for vehicle crashworthiness design

Based on the data mining theory and decision tree method introduced above, a system design methodology was developed that is capable of exploring the simulation dataset to uncover the hidden knowledge and design rules. The procedure of this approach is illustrated in Figure 3.3, where the dashed arrows indicate the information flow, and the solid arrows are the workflows. The whole design process can be divided into three steps: 1) design at the system level; 2) design at the component level; 3) design verification using the whole-vehicle model. The steps are briefly described below.

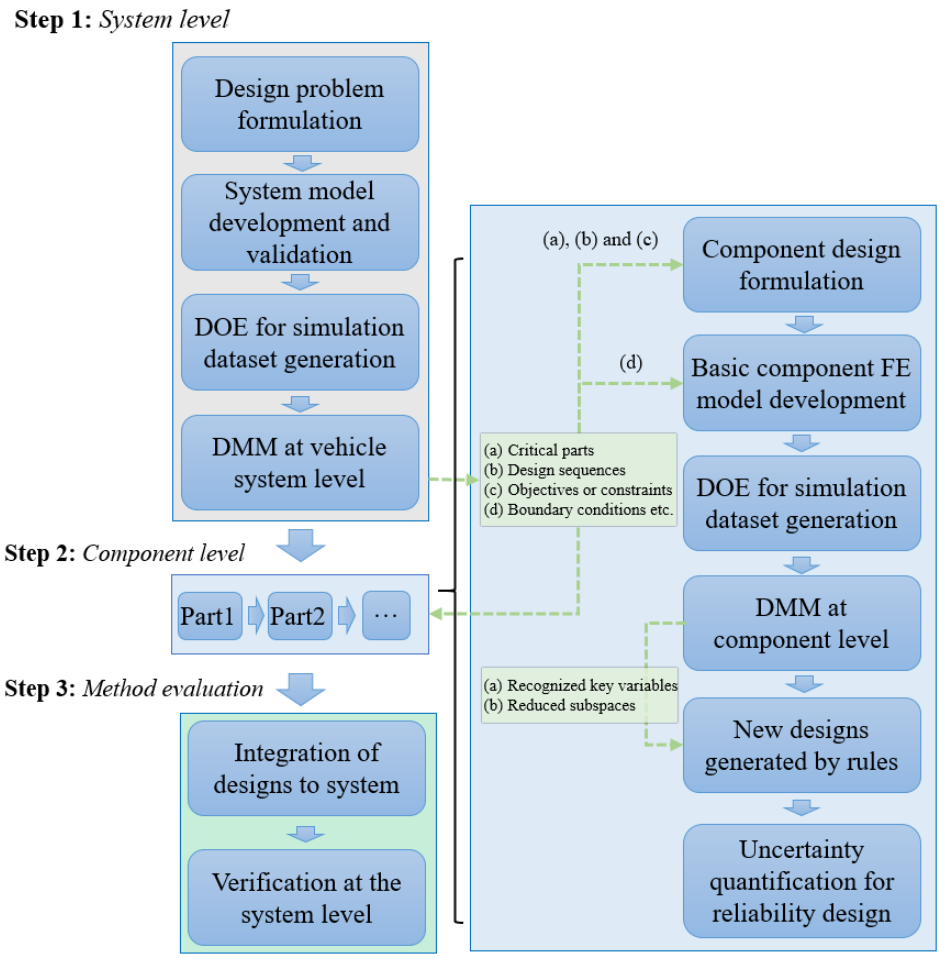


Figure 3.3. The procedure of the data mining method for vehicle crashworthiness design.

3.3.1 Step 1: Design at the system level

This step aims to establish a large simulation dataset and to mine data for the main energy-absorbing structure at the system level to identify the relationship between vehicular overall response and the behavior of critical components. Based on the information explored, decision-making rules are derived to determine the sequence of component design and calculate the boundary conditions for these components. In this step, the system and component design are linked by these rules as generated through two primary operations: the dataset generation and data mining.

A selected or developed vehicle FE model is used to generate the system-level simulation dataset. After model validation, a preliminary crash simulation is performed, and the components with high energy absorption are considered to be the main energy-absorbing components. This screening operation is a critical step to exclude noncritical components to reduce the design cost.

The geometric design of each component is changed, in principle, to modify its average stiffness (*avgstiff*) under the corresponding crash condition. This parameter can be regarded as a feature of each component, since it is closely related to energy absorption, acceleration and intrusion. The system-level design is to determine the *avgstiff* requirement of each main energy-absorbing component so that their combination ensures the vehicle's overall safety. In this step, the design synthesis is coarse, and the geometric parameters meant to change each part's attribute (i.e. *avgstiff*) level are limited to a small number ($\leq$3). The wall thickness, diameter and bounding box are some typical geometric parameters. Using a sampling algorithm such as the Latin hypercube method (LHM), a

large number of DOEs for critical components can be created and simulated, and the results form a large dataset. This dataset contains the information and simulation results for selected components (i.e. main geometric parameters, mass, energy absorption, deflection, etc.) and the response of the main EASs (i.e. total energy absorption, intrusion, peak acceleration, etc.).

Based on the requirements specified in the safety regulations or the designer's preference, the overall performance of each DOE (e.g. specific energy) is labeled as "good/g," "intermediate/m," or "poor/p." Then, the decision tree method described in Section 3.2 is used to classify the performance of each key component and link the components to the overall performance label. Likewise, the boundary condition for each component is determined by following the decision-making rules for a specific leaf node.

3.3.2 Step 2: Design at the component level

This step establishes large simulation datasets and conducts data mining at the component level to identify the relationships between the responses and the detailed geometric features. The interrelationships among the key geometric design variables and responses are also revealed. Based on the information explored, a decision-making rule at the component level can be derived and used in the detailed component design.

The critical components identified in Step 1 are further screened from a selected route. Before the detailed design takes place, these components are parameterized with a number of geometric features (for a simple geometry) or data points (for a complex geometry) to describe their profile. For complex geometries, a dimension-reduction algorithm can be used to decrease the dimensions of the design variables. DOEs of the

components selected are generated by modifying the design variables, that is, the geometric features for simple geometries and point locations for complex geometries. Component crush simulations are conducted on these DOEs to generate the simulation dataset at the component level. Based on the predefined design objectives, the performance of each component again labeled as “good/g,” “intermediate/m,” or “poor/p.” The component simulation dataset is mined to generate a decision tree for each component. The optimum route through each tree is identified. It represents the interrelationships among the design variables and the design rule, from component level to design variables. New designs with expected performance can also be generated by following the rules identified from this route.

3.3.3 Step 3: Design method verification

This task is to verify the performance of the design methodology using a whole-vehicle model. The critical energy absorbing components designed through Steps 1 and 2 are integrated into the vehicle FE model as replacement for the original designs. The whole-vehicle crash simulations are performed again, and the simulated responses, such as peak acceleration, energy absorption and maximum intrusion, are compared with the result of the original design. Better performance would indicate design improvement and verify the advancement of the new method.

3.4 Summary

This chapter proposed a systematic design methodology for vehicle crashworthiness based on data mining theory. This technique allows exploration of a large crash simulation dataset to discover underlying complicated relationships between response

and design variables at multiple levels (main energy-absorbing systems, components, and geometric features) and allows derivation of design rules based on the whole-vehicle safety requirements that can inform decisions about the component- and subcomponent-level design. Full vehicle and component simulation datasets are mined to build decision trees at two levels. Based on the decision trees, the interrelationships among the design variables can be revealed, and decision-making rules that lead to a set of good designs can be determined. To verify this method, an additional whole-vehicle crash simulation based on the updated components will be conducted, and the results are compared to those with the original design.

Chapter 4 Detailed component design: A dimension-based approach

4.1 Introduction

Building on the theoretical framework proposed in Chapter 3, in this chapter, the focus is on the methodology for detailed component design. The goal is to explore the interrelationships of design variables (or geometric features) and to establish the design rules for the critical energy-absorbing components identified in the first step. The profile of a component can be represented by a number of geometric parameters, such as the length, angle and thickness. By systematically adjusting these dimension values, the detailed design can be developed. This process is known as the *dimension-based approach*, which is widely used to design structures with a relatively regular shape. These geometric parameters or features can be used as the input features in the subsequent data mining. For those parts with highly irregular shapes, a node-based approach will be used, as detailed in Chapter 5.

This chapter is organized as follows: Section 4.2 introduces the workflow of the dimension-based design method. Then, in Sections 4.3 and 4.4, dataset generation and data mining processes are described.

4.2 Workflow

The four-step workflow for the dimension-based structural design process is shown in Figure 4.1. In Step 1, the structure to be designed is parameterized by its geometric features, together with the constraints determined prior to the component-level design. The geometric features are represented by their dimensions, and their ranges are also

determined. To consider different combinations of these parameters, a large number of DOEs are generated in Step 2. Each DOE represents a design alternative, and the population of DOEs forms a design space. Their geometric models are then built using Computer-Aided Design (CAD) software and converted to finite elements using a mesh tool. FE simulations are conducted on all DOEs; the results form a large dataset.

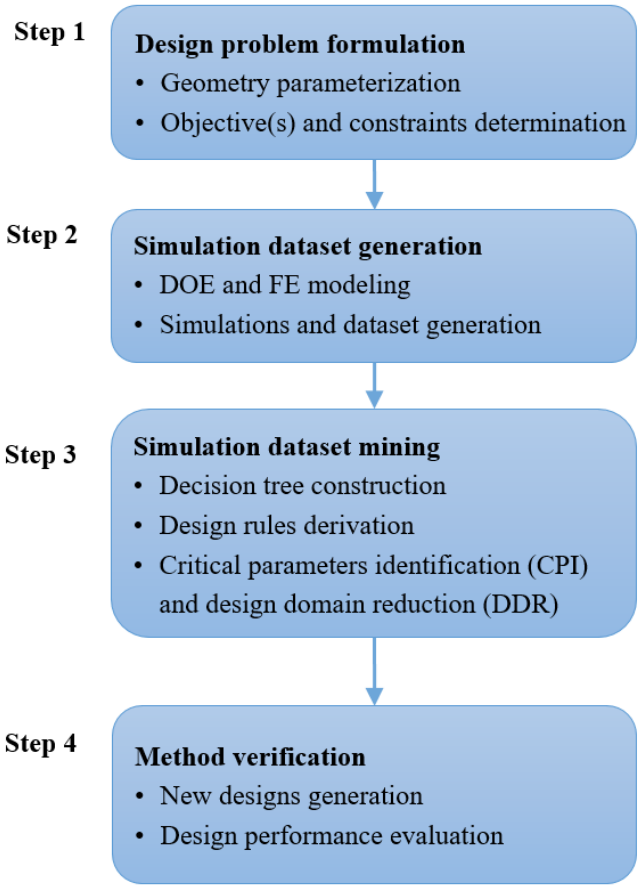


Figure 4.1. Workflow of the detailed component design using the dimension-based approach.

In Step 3, the simulation dataset is mined to discover the relationships among design variables and their influence on the results. A decision tree is built to determine decision-

making rules or the workflow to link the key geometric parameters and design results. In this way, the variables leading to satisfactory design sets can be determined in the right sequence. In Step 4, using the decision-making rules obtained, the design space is reduced, and the subsequent designs are developed within this reduced space. The strategies to identify high-impact design variables and to reduce their design space are termed critical parameter identification (CPI) and design domain reduction (DDR), respectively. These two strategies are detailed in Section 4.4.3. A group of designs with good performance are generated and evaluated using FE simulation. The results are then used to verify the new design method.

4.3 Dataset generation

System-level design was carried out for a typical passenger car, the Ford Taurus. The FE model for this car was obtained from the National Highway Traffic Safety Administration (NHTSA: http://www.nhtsa.gov/) database. The whole-vehicle frontal crash simulation results show that the thin-walled side rail (also known as S-shaped beam or S-beam) is the critical structure for energy absorption in the frontal impact scenario. Therefore, the S-shaped beam was selected for detailed component design.

4.3.1 Component parameterization and geometry modeling

To conduct a detailed geometry design on the S-beam, its profile must be parameterized. The fully parameterized S-beam model is shown in Figure 4.2. The total length of the beam is 1,000 mm. The geometric parameters, including upper segment length (*UL*), upper radius (*UR*), slope segment angle (*AN*), bottom segment radius (*DR*), height (*H*), tube cross-section length (*CL*) and thickness (*T*), are design variables. Points **r** and **f** are

two center points at the ends. Points **a**-**d** are the transition points between different segments of this beam.

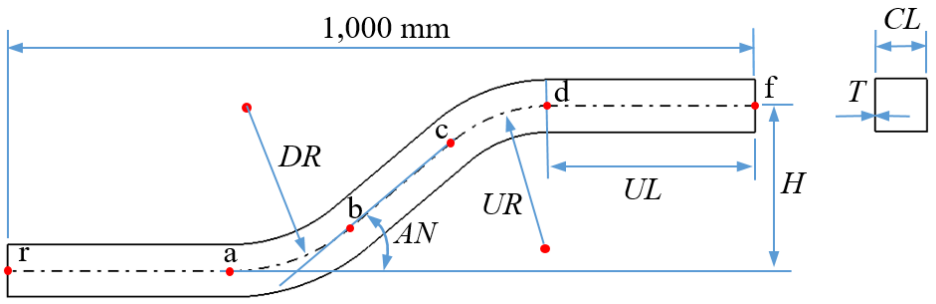


Figure 4.2. Parameterization of an S-shaped beam.

The following geometric constraints were used: $H - (UR + DR) \times (1 - \cos(AN)) > 0$ in the vertical direction, which makes the length of segment **bc** greater than zero. $1{,}000 - (UL + H/\tan(AN) + (UR + DR) \times \tan(AN/2)) > 0$ in the horizontal direction, which guarantees the length of segment **ra** is greater than zero. The ranges of the design variables are as follows: 100 mm $\leq UL \leq$ 400 mm; 150 mm $\leq UR \leq$ 600 mm; 20° $\leq UL \leq$ 40°; 150 mm $\leq DR \leq$ 600 mm; 150 mm $\leq H \leq$ 300 mm; 50 mm $\leq CL \leq$ 70 mm; and 1.5 mm $\leq T \leq$ 3.0 mm. The performance of a crashworthy structure can be quantified by specific energy absorption (*SEA*), which reflects energy-absorbing efficiency (Bai et al., 2015) in the form of

$$SEA = \frac{E_{\text{int}}}{M}, \qquad (4.1)$$

where E_{int} is the total internal energy, and M represents the mass of the S-shaped beam. Our goals in this study are (1) discovering the influence of the design variables on SEA and the interrelationships among design variables and SEA and (2) determining the right

design sequence. To achieve these goals, a large design space was built. Based on the geometric constraints and parametric ranges indicated above, 150 DOEs were generated with the LHM (Chen et al., 2013, Iman, 2008) using commercial multidisciplinary optimization software modeFRONTIER v4.5 (ESTECO, Trieste, Italy). A CAD tool, Catia V5 (Dassault, Courbevoie, France), was then coupled with modeFRONTIER to generate 150 geometry models automatically. The 3D geometric models were saved in ".stp" format, ready for a batch meshing process with the control of element quality.

4.3.2 Finite element modeling and simulation dataset generation

Meshing was implemented using the software Hypermesh V13 (Altair, Detroit, MI). The S-shaped beam geometric model was converted to an FE model with 8-mm quadrilateral shell elements, which are fine enough to model S-shaped beam crash problems (Nguyen et al., 2014). To account for the vehicular mass inertia, a 500 kg mass block was attached to the far end of the beam, as shown in Figure 4.3 (Khakhali et al., 2010b). A rigid block was attached to the frontal end of the beam to represent a highly simplified bumper. Then, an initial velocity of 10 m/s was assigned to the S-shaped beam and it was made to hit a rigid wall (Khakhali et al., 2010b), which is similar to the conditions used in the NCAP frontal impact test (NHTSA, 2008). Contact interfaces were used to model the interaction between the beam and the wall. The simulations were performed using the FE analysis software package LS-DYNA V971_R4.2.1 (LSTC, Livermore, CA).

The beam FE model was validated against published data to ensure its accuracy (Khakhali et al., 2010a). The material was described using an elasto-plastic material law with the following parameters: density ρ = 7,800 kg/m^3, Young's modulus E = 206 GPa,

Poisson's ratio $\upsilon = 0.3$, tangent modulus $E_T = 1.4$ GPa, yield stress $\sigma_0 = 162$ MPa. The geometric parameters of five design cases, as well as a comparison of model predictions and the data in the literature, are shown in Table 4.1. The results show that the maximum discrepancy of the model predicted energy absorption values (denoted as "Model") and the data reported in the literature (denoted as "Ref") is 6.3% (<10%), which indicates good agreement. Therefore, the beam FE model can be deemed as sufficiently validated.

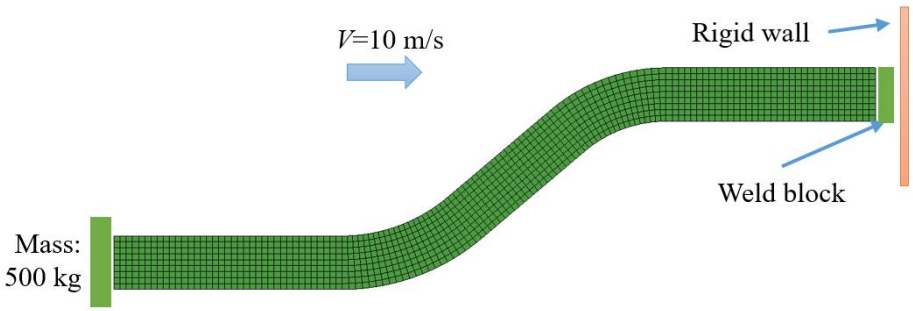


Figure 4.3. FE model of the S-shaped beam and boundary and loading conditions.

Table 4.1

Geometric parameters of five S-beam design cases and comparison between the model predicted energy absorption values (denoted as "Model") and the data reported in the literature (denoted as "Ref")

NO.	NO. in Ref.	*UL* (mm)	*UR*/mm	*AN* (˚)	*DR* (mm)	*H* (mm)	*CL* (mm)	*T* (mm)	Energy (J)		Error/%
									Ref	Model	
1	A1	255	435	33	435	150	40	3	4,826	4,841	0.32
2	A3	370	150	60	150	150	60	2	4,026	3,869	-3.92
3	A4	370	150	60	150	150	70	2	4,507	4,487	-0.44
4	D2	240	300	60	300	300	60	2.5	3,689	3,796	2.90
5	D3	240	300	60	300	300	70	2	2,787	2,963	6.30

It should be noted that in the model validation, no strain-rate effect was taken into account in the material law. This was done to maintain consistency with the conditions reported in the literature. After the model was validated, the material model was updated by including the strain-rate effect. In this work, the Cowper–Symonds model was used, which is in the form of (Huh and Kang, 2002).

$$\frac{\sigma_{Yd}}{\sigma_0} = 1 + (\dot{\varepsilon}/C)^{1/P} \quad , \qquad (4.2)$$

where σ_{Yd} is the dynamic yield stress, σ_0 is the quasi-static yield stress, $\dot{\varepsilon}$ is the effective strain rate, and *C* and *P* are the strain-rate-dependent constants. The actual material constants are listed as follows (Bai et al., 2015): density ρ = 7,800 kg/m^3, Young's modulus *E* = 210 GPa, Poisson's ratio υ = 0.3, yield stress σ_0 = 162 MPa, rate-dependent parameters C = 40.1/s, P = 5. All of the 150 design cases were simulated under the aforementioned loading conditions, and the simulation results formed a large dataset for further data mining and analysis.

4.4 Data mining

4.4.1 Dataset labeling

After the simulation dataset was created, based on the requirements specified in the safety regulations or the designer's preference, the predicted *SEA* of each DOE was classified and labeled as "good/g," "intermediate/m" or "poor/p." In this study, the vehicle performance was labeled as "g" if *SEA* ≥ 2,000 J/kg, as "m" when 1,500 J/kg ≤ *SEA* < 2,000 J/kg, and as "p" for *SEA* < 1,500 J/kg. A decision tree was then built based on the C4.5 (J48 in the Weka software) algorithm using the software Weka v3.8.0 (Waikato University, New

Zealand), as shown in Figure 4.4. This was used to classify the performance of each key design variable and to link each to the overall performance label.

4.4.2 Decision tree analysis

In the decision tree, the nodes of design variables and performance are represented as elliptical and rectangular boxes, respectively. In the box of a leaf node, the number behind the label p/m/g indicates the number of DOEs in that particular class. The node splitting criteria are shown on the paths. The whole decision tree includes ten leaf nodes that correspond to 8 branches (labeled as b1 to b8). Each branch represents a decision-making process for a particular design subset.

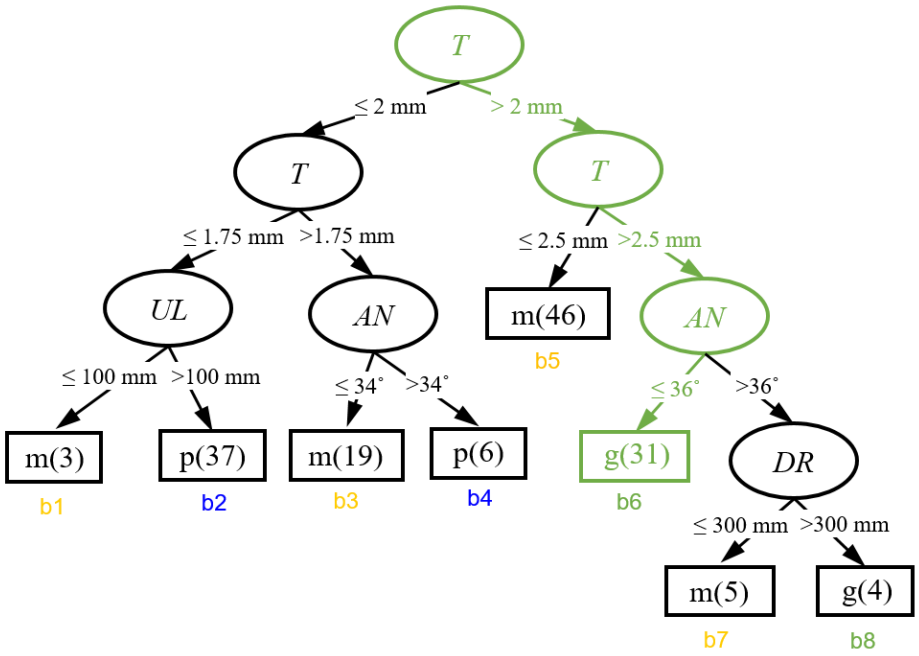


Figure 4.4. A decision tree generated by the S-beam simulation dataset (labels for structural energy absorbing performance: g = good, m = intermediate and p = poor).

With the decision tree in Figure 4.4, the design of an S-shaped beam starts from the top node, that is, wall thickness *T*. The top node has the greatest influence on the response classification. If the thickness is greater than 2 mm, we then move to the right path and check its child node, which is *T* again. If $T \leq 2.5$ mm, then this design produces an "intermediate" level of performance. If the thickness is greater than 2.5 mm, we then check the next child node, *AN*. If its value is equal to or less than 36˚, a "good" design is achieved. There may be more than one path yielding designs with a "g" label. If the path with the greatest number of cases is selected, this implies a larger subspace and robustness. The best path, b6, is marked in green in Figure 4.4.

4.4.3 Critical parameters identification (CPI) and design domain reduction (DDR)

As indicated in the workflow in Figure 4.1, after the construction of a decision tree, the CPI and DDR can be performed. CPI is to identify those design variables that have a substantial effect on the system performance. Based on these key variables, DDR is then performed to reduce the range of values of the variables, thus decreasing the design space.

In the decision tree shown in Figure 4.4, following branch b6 identifies two design variables (*AN* and *T*) as the key parameters, and the S-beam design process will focus on these two variables. The correlation analysis also confirms that *AN* and *T* have the most significant effects on the response. The ranges of their values have been determined and are clearly shown on branch b6. With the reduced range for *AN* and *T* values, DDR is achieved, as shown in Table 4.2. The other five design variables, namely, *DR*, *UL*, *UR*, *H* and *CL*, do not have a significant influence on the energy-absorption response. They are

set as constants and assigned to their corresponding values from the best design of the original dataset, as listed in Table 4.2.

Table 4.2

Comparison between initial and reduced design spaces after data mining

Design variables	The best design in the original dataset (*SEA* = 3,979.6 J/kg)	Initial design space		Reduced design space	
		Lower limit	Upper limit	Lower limit	Upper limit
AN/ ˚	22	20	40	20	36
T/ mm	2.75	1.5	3	2.75	3
DR/ mm	200	150	600	200	
UL/ mm	100	100	400	100	
UR/ mm	150	150	600	150	
H/ mm	290	150	600	290	
CL/ mm	52	50	70	52	

4.5 Discussion

4.5.1. Comparison between DMM and conventional approach

After the CPI and DDR are completed, the S-shaped beam design process can be highly simplified by using a smaller number of design variables in a narrower design domain. In this section, the same structure is designed using a conventional response surface method (RSM) (Craig et al., 2005, Kurtaran et al., 2002, Shimoyama et al., 2009) and the new method. The results are then compared to verify the performance of the new DMM.

Based on the results of CPI and DDR, 20 designs were generated for the S-beam design using the DMM. This number of design alternatives should be sufficient to ensure the

accuracy of results in the current reduced design space (Shi et al., 2012, Yang et al., 2005). Another design was then developed with the conventional approach using the original 150 designs without CPI and DDR. In both designs, the design objective (i.e. *SEA*) can be expressed as a nonlinear polynomial function (i.e. RSM) of design variables by fitting the response-design variable curves. Using the DMM with CPI and DDR, only two key design variables (i.e. *AN* and *T*) and their combinations need to be considered; whereas if CPI and DDR are not used, all seven geometric parameters and their combinations must be taken into account. The comparison of the two RSM functions in Table 4.3 shows that with the DMM, the structural response can be predicted using a much shorter equation, which results in a faster convergence time and less computational cost, as confirmed by the smaller convergence iterations and times in Table 4.3. Additionally, the new DMM exhibits a higher accuracy in terms of sum of squares for error (SSE), R and discrepancy between simulation results and RSM predictions, since the irrelevant information was eliminated during the generation of the decision-making rules.

Figure 4.5(a) and (b) illustrate the *SEA* values as a function of two key design variables, *AN* and *T*, predicted by the conventional method and the DMM, respectively, together with the simulation results. The color stripes quantify the simulated *SEA* of the designs, while the contour lines with numbers are RSM-predicted *SEA*. In Figure 4.5(a), the dependence of the *SEA* value on the variation of two key design variables, *AN* and *T*, is shown without including the other five parameters, although they also contribute to the results. The *SEA* increases with a larger *T* and a smaller *AN* in the original design space. However, after the DDR, the range of *T* values was reduced from 1.5 mm $\leq T \leq$ 3 mm to

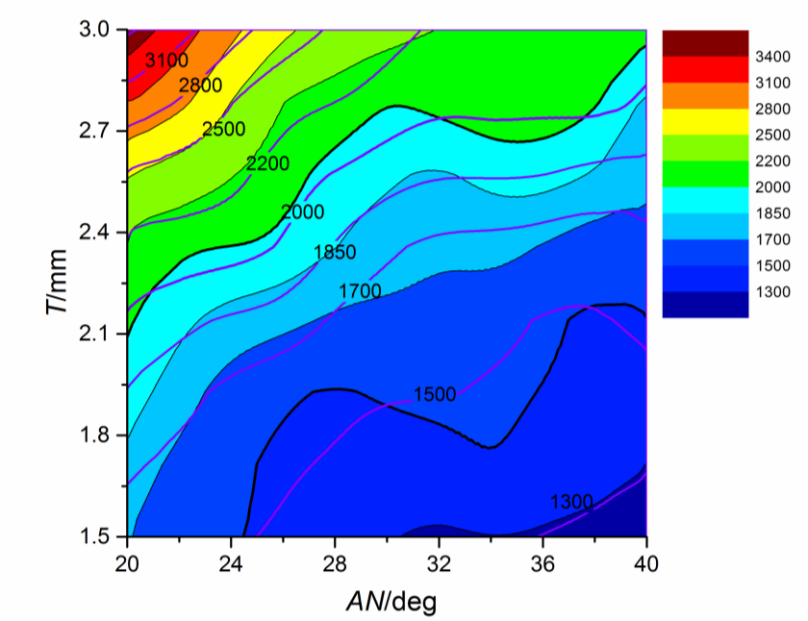
2.75 mm ≤ T ≤ 3 mm. As shown in Figure 4.5(b), *SEA* is not sensitive to T within that small range but decreases almost linearly with an increase of the value of *AN*. This finding indicates that for a given design objective and constraints, a highly nonlinear problem (i.e. Figure 4.5(a)) can be simplified to a nearly linear problem (i.e. Figure 4.5(b)) using CPI and DDR.

Table 4.3

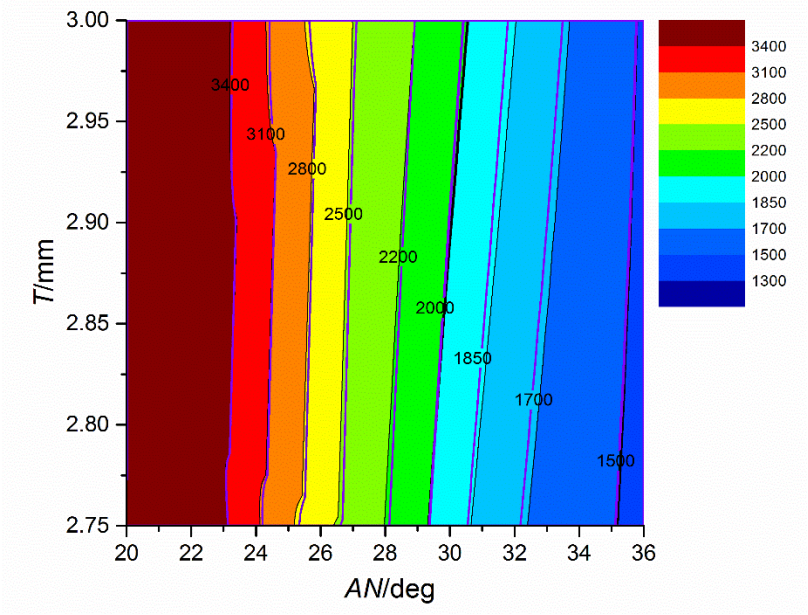
Comparison of the two design methods (with/without CPI and DDR) in the S-beam design

	Data mining method	Conventional method
Polynomial RSM equation	$SEA = 1.023 - 2.483 \times AN + 0.008 \times T + 1.464 \times AN^2 - 0.057 \times T^2 + 0.151 \times AN \times T$	$SEA = 0.238 - 0.145 \times DR + 0.458 \times H - 0.129 \times CL - 0.499 \times AN + 0.261 \times T + 0.06 \times UR + 0.188 \times UL + 0.002 \times DR^2 + 0.061 \times H^2 + 0.022 \times CL^2 + 0.285 \times AN^2 + 0.039 \times T^2 + 0.02 \times UR^2 - 0.319 \times UL^2 - 0.01 \times DR \times H + 0.103 \times DR \times CL + 0.21 \times DR \times AN - 0.038 \times DR \times T - 0.005 \times DR \times UR + 0.074 \times DR \times UL - 0.019 \times H \times CL - 0.416 \times H \times AN + 0.041 \times H \times T - 0.325 \times H \times UR - 0.325 \times H \times UL - 0.047 \times CL \times AN - 0.025 \times CL \times T - 0.069 \times CL \times UR + 0.197 \times CL \times UL - 0.081 \times AN \times T + 0.270 \times AN \times UR + 0.166 \times AN \times UL + 0.137 \times T \times UR - 0.021 \times T \times UL - 0.084 \times UR \times UL$
Fitting time/ s	2	1,761
Iterations	18	21
Loss[2]	0.039	0.516
R	0.99	0.92
Difference[3]	11.9	74.6

Note: 1) The data used to construct the RSMs were normalized so that the coefficients could be limited to the same order of magnitude. The equation, $\bar{X} = (X - X_{min})/(X_{max} - X_{min})$, was used, where for a specific design variable, $\bar{X}$ and X are variable values after and before normalization; X_{max} and X_{min} are upper and lower bounds of its values, respectively. 2) Loss: sum of squares for error (SSE); 3) Difference: max difference with simulation/ %



(a)



(b)

Figure 4.5. SEA values as a function of two key design variables, *AN* and *T*, predicted by (a) the conventional design method based on the initial DOE (150 designs) and (b) the data mining method based on the new DOE (20 designs). Color fringe: simulation results; contour lines with numbers: RSM predicted results

Regarding the RSM accuracy, Figure 4.5(a) and (b) also show the comparison of simulated *SEA* (color stripes) and RSM predicted results (contour lines with numbers) without and with CPI and DDR, respectively. Compared with the results of the conventional method (Figure 4.5(a)), the DMM yields a better match between design and simulation, since the highly nonlinear RSM equation in the traditional method is not able to accurately describe the complicated mutual effects of design variables. After data mining, the nearly linear response shown in Figure 4.5(b) can be easily captured by the simplified RSM model. Moreover, the better response, shown in Figure 4.5(b), was caused by the elimination of the redundant "poor" design subspace through data mining.

The accuracy of the RSM predictions with the two methods is further compared in Figure 4.6, where the simulated *SEA* is plotted against the RSM predictions using the response equations. The perfect match line with the slope of 1 indicates perfect agreement, and the distance from this line represents the degree of discrepancy. From the results, it is clear that the DMM is more accurate (i.e. its results are closer than the conventional method results to the perfect match line).

Another comparison was conducted on the speed of convergence. Optimal designs on the S-beam were carried out based on the two equations in Table 4.3 with a genetic algorithm (GA) (Goldberg and Samtani, 1986, Goel et al., 2010). As a general optimization method, a GA can solve a wide range of problems whether or not the analytical solutions are available. The optimization was ended after 50 generations with 20 populations in each generation. The iteration records of these two methods are illustrated in Figure 4.7 to show the convergence histories. The data mining method predicts a maximum *SEA* of 4,549.6 J/kg, which is higher than that predicted by the conventional method (4,279.2

J/kg). Moreover, the data mining method reaches convergence in only two iterations, much faster than the conventional method, which converges after 15 iterations. This is because the CPI and DDR reduced the design domain by eliminating "poor" designs, leaving the "good" designs remaining in the design space, as shown in Figure 4.5(b). This suggests that for a system with high-dimensional design variables, the data mining design method could overcome the limitations of the conventional approach and improve efficiency through CPI and DDR.

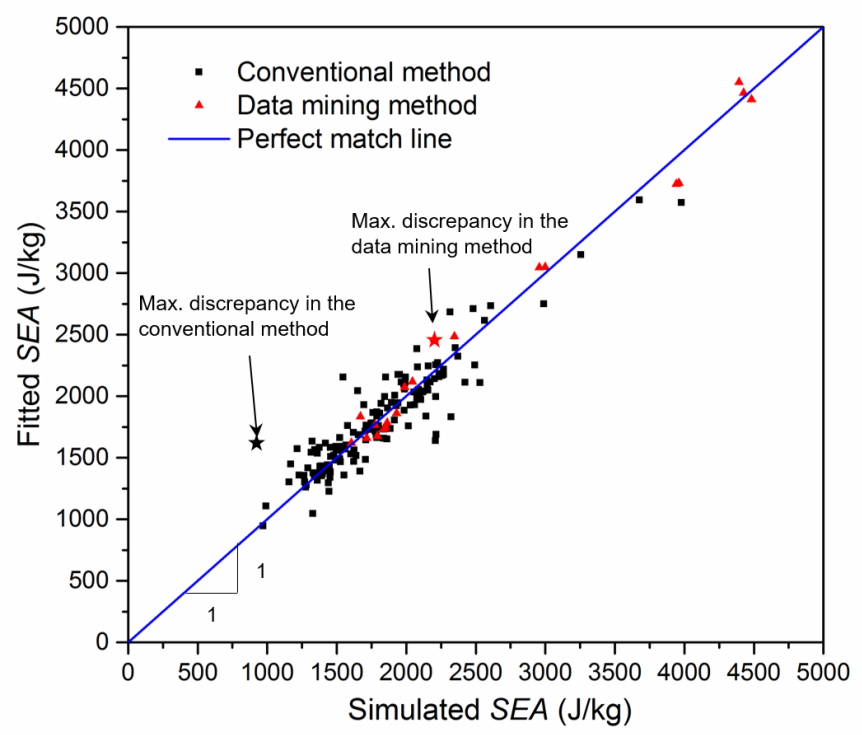


Figure 4.6. Comparison of the accuracy of RSM predictions based on two design methods against simulation results. The red and black stars represent the maximum discrepancies from the simulation data of the data mining method and the conventional method, respectively.

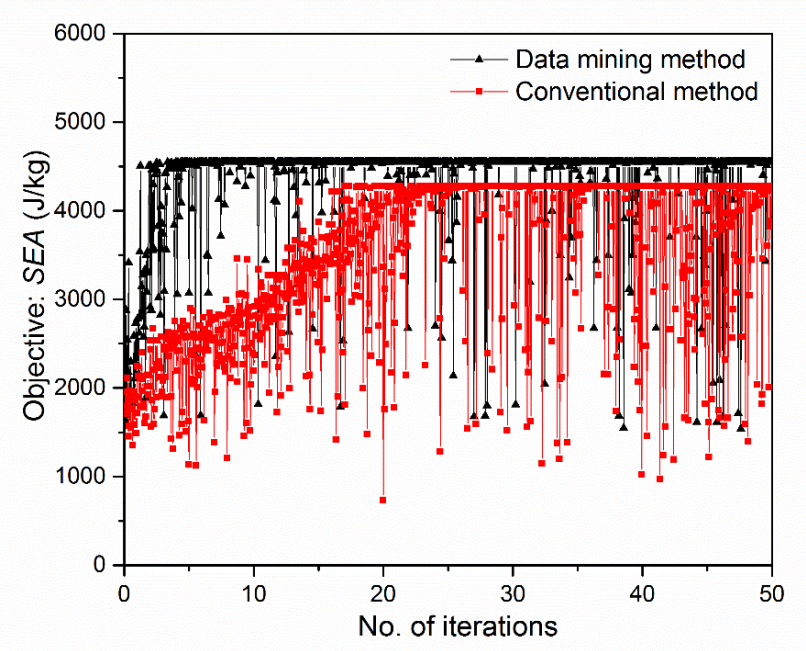
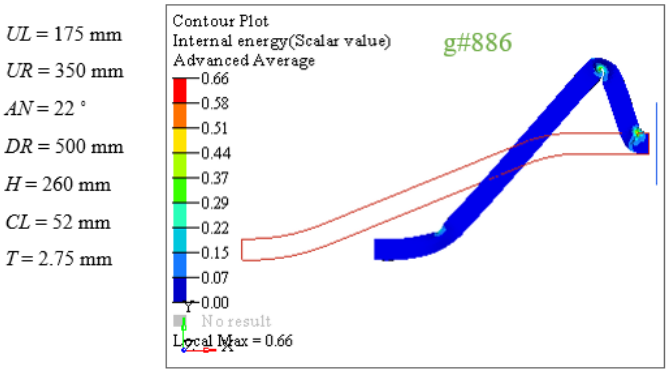


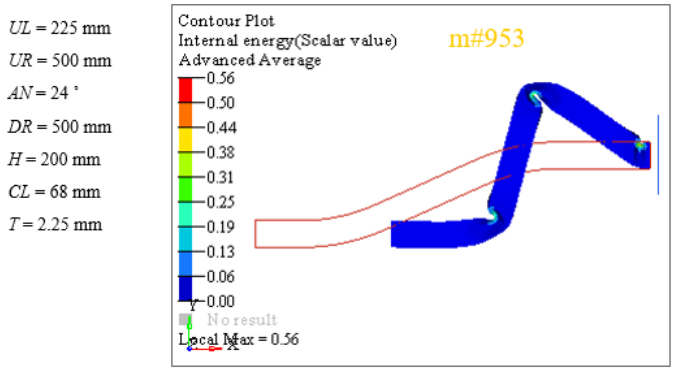
Figure 4.7. The iteration histories to maximize SEA using the data mining method and the conventional design method

4.5.2 Bending mode analysis

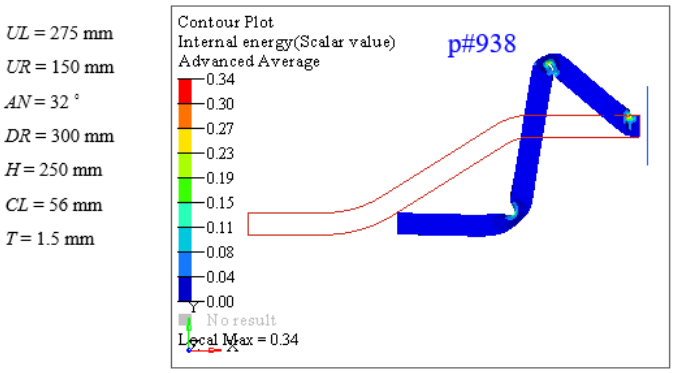
A further analysis was carried out on the failure modes of the beams in the g, m and p groups. The typical modes in all three groups predicted by the simulations are shown in Figure 4.8, where the design case number and values of design variables are also included. In Figure 4.8, the fringe levels indicate the value of internal energy density, which is defined as energy dissipation per unit volume, and it is proportional to the magnitude of SEA. One can see that design case 886 in the good design class exhibits the highest internal energy density (0.66 J/mm^3), followed by case 953 in the intermediate class (0.56 J/mm^3) and case 938 in the poor class (0.34 J/mm^3). The difference in energy absorption is believed to be caused by different strain values at the location of plastic bending. The bending modes shown in Figure 4.8(b) and (c) generally should be avoided.



(a)



(b)



(c)

Figure 4.8. Typical failure modes in the (a) good, (b) intermediate and (c) poor design groups predicted with the simulations (fringe: internal energy density in J/mm^3).

4.6 Summary

A dimension-based design methodology was established and applied to the detailed design of a thin-walled vehicular structure, that is, an S-shaped beam, against crash loading. The component was parameterized by a number of geometric features, which can be used as design variables. The method allows the mining of the large crash simulation dataset to build a decision tree, with which one can discover the underlying relationships between response and design variables and derive design rules based on the structural response (SEA) to make decisions about the geometric design. Using the decision tree, CPI and DDR were performed to ensure a more efficient design. The results suggest that the newly developed approach can be used to design a high-dimensional system efficiently and effectively and that it outperforms conventional methods. However, this approach still has a limitation. For a component with complex geometry that cannot be readily represented by a small number of geometric features, different parameterization methods must be used. For example, the profile of such a component can be expressed in the form of a nonuniform rational basis spline (NURBS) using a large number of control points, and these control points can then be used as design variables and mined. This method is presented in detail in Chapter 5.

Chapter 5 Detailed component design: A node-based approach

5.1 Introduction

As demonstrated in Chapter 4, the dimension-based method can work quite well for structures with a regular shape, where only a small number of design variables are needed. However, complicated structures with irregular profiles can hardly be represented concisely by these geometric features. Thus, they cannot be conveniently designed in this traditional way.

To resolve this issue, one can use node-based design methods, which apply a group of nodes, or a cloud of points, as design variables to represent a complex geometry. The node group can serve as the nodes in the FE model or the control points of the geometric profile. By moving the locations of the nodes, the local or global shape of any structure can be changed. If an FE model is used, mesh morphing techniques are applied frequently to avoid mesh distortion (Hojjat et al., 2014, Georgios and Dimitrios, 2009). Most morphing methods, however, are implemented by defining morphing boxes and changing the initial FE model by changing the parameters of boxes and predefined handles (Georgios and Dimitrios, 2009). Although the morphing is widely used, it is still difficult to be applied to the design of structures with large deformations (Duddeck and Zimmer, 2013), since they may lead to the deterioration of mesh quality. To better model large geometric changes and control the mesh quality, another option, for example, a geometry point cloud such as the control points of a NURBS (Clune et al., 2014, Vishwanathan, 2017), can be taken as design variables. However, given the vast number of points, it is expensive to develop a good design with satisfactory structural performance by changing

the location of each node or point. This is an intrinsic issue with most node-based approaches, such as implicit parameterization (Duddeck and Zimmer, 2013).

To overcome the problem of dataset size, in this study, a new principal component analysis (PCA)-based structural design method is developed. The new method follows the basic idea of a point- or node-based modeling strategy. The surface of a structure is described by the control points in the form of a NURBS (Lee, 1999, Clune et al., 2014, Mosavi et al., 2012, Vishwanathan, 2017). The large control point dataset is compressed using a technique known as PCA. PCA is a mathematical procedure to reduce high-dimensional data by expressing them with a set of linearly uncorrelated orthogonal basis vectors, or principal components (PCs). The original dataset can be recovered using a decompression procedure by multiplying the orthogonal basis with the principal component scores (PCSs) (Li et al., 2011, Reed and Parkinson, 2008). This mathematical theory has been applied to a number of engineering applications, for example, complex shapes analysis such as human-body anthropometry feature derivation (Li et al., 2011, Reed and Parkinson, 2008), experimental corridor development (Sun et al., 2016) or osteoarthritis analysis (Bredbenner et al., 2010).

In a particular dataset, the PC matrix is unique, and the PCSs can be regarded as the weight of each PC vector, which can be seen as specific geometry features hidden in the initial geometry set. By changing the value of the PCSs, the geometry of the part can be modified by the decompression process of PCA. Through this method, instead of directly handling a massive number of control points, one can perform the design by adjusting the values of a small number of PCSs, and thus the computational cost is significantly reduced. To demonstrate the performance of this strategy, the S-beam is selected again as

the component to be redesigned. After the design is completed, to fully take advantage of the simulation dataset generated a data mining process is performed to explore the implicit interrelationships between design variables (i.e. PCSs) and create design rules to guide the design procedure.

The remaining parts of this chapter are organized as follows: Section 5.2 describes the basic PCA design algorithm for structures and the workflow of the PCA design method with a data mining process. A case study on the S-shaped beam design is then presented step by step in Section 5.3, which includes the initial geometry dataset generation, the shape parameterization by PCA , the simulation dataset generation and the DMM implementation. Based on the results, design rules are derived in Section 5.4.

5.2 Theory and method

5.2.1 Node-based shape design by principal component analysis (PCA)

PCA is a mathematical process used to reduce the dimensions of a dataset by taking the first several PCs to represent the original high-dimensional dataset (Han et al., 2011). For a specific initial dataset, after PCA the average and PC matrices are unique, and each sample in the initial dataset can be represented by the linear combination of PCs with its PCSs as the coefficients.

The theory of PCA is detailed in the literature (Li et al., 2011, Reed and Parkinson, 2008) and briefly summarized here. The geometry dataset $\mathbf{D}_{n\times m}$ is composed of n samples, and each sample is a vector with m design variables. Each sample, meaning all nodes of a geometry model, contains information on the geometric features. PCA can compress the

dataset to reduce the number of dimensions. In the proposed PCA design algorithm, a dataset of geometry matrix $\mathbf{D}_{n\times m}$ is expressed in the form of PC matrix ($\mathbf{P}_{m\times m}$) and PCS matrix ($\mathbf{S}_{n\times m}$) as presented in Eq. 5.1:

$$\mathbf{D}_{n\times m} = \bar{\mathbf{D}}_{n\times m} + \mathbf{S}_{n\times m} \cdot \mathbf{P}_{m\times m} \ , \tag{5.1}$$

where each row of $\bar{\mathbf{D}}_{n\times m}$ (mean matrix) is the mean vector of matrix $\mathbf{D}_{n\times m}$. $\mathbf{S}_{n\times m}$ and $\mathbf{P}_{m\times m}$ are PCS and PC matrices, respectively, which can be calculated by eigenvalues and eigenvectors analysis of the covariance matrix of the centralized matrix $\mathbf{C}_{n\times m}$ in Eq. 5.2:

$$\mathbf{C}_{n\times m} = \mathbf{D}_{n\times m} - \bar{\mathbf{D}}_{n\times m} = \left[c_{\bullet 1}; c_{\bullet 2}; c_{\bullet 3}; \cdots; c_{\bullet m}\right] , \tag{5.2}$$

where $c_{\bullet i}$ (i = 1, 2, 3, $\cdots m$) is the row of the centralized matrix $\mathbf{C}_{n\times m}$, and then the covariance matrix for each dimension ($c_{\bullet i}$) in $\mathbf{C}_{n\times m}$ is expressed in Eq. 5.3:

$$\mathbf{COV(C}_{n\times m}\mathbf{)}_{m\times m} = \begin{bmatrix} \operatorname{cov}(c_{\bullet 1}, c_{\bullet 1}) & \operatorname{cov}(c_{\bullet 1}, c_{\bullet 2}) & \cdots & \operatorname{cov}(c_{\bullet 1}, c_{\bullet m}) \\ \operatorname{cov}(c_{\bullet 2}, c_{\bullet 1}) & \operatorname{cov}(c_{\bullet 2}, c_{\bullet 2}) & \cdots & \operatorname{cov}(c_{\bullet 2}, c_{\bullet m}) \\ \vdots & \vdots & \ddots & \vdots \\ \operatorname{cov}(c_{\bullet m}, c_{\bullet 1}) & \operatorname{cov}(c_{\bullet m}, c_{\bullet 2}) & \cdots & \operatorname{cov}(c_{\bullet m}, c_{\bullet m}) \end{bmatrix} . \tag{5.3}$$

In Eq. 5.3, the covariance element can be calculated by

$$\operatorname{cov}(c_{\bullet i}, c_{\bullet j}) = \frac{\sum_{l=1}^{n} (c_{li} - \bar{c}_{\bullet i})(c_{lj} - \bar{c}_{\bullet j})}{n-1} \ , \tag{5.4}$$

where $\bar{c}_{\bullet i}$ is the mean value of the dimension $c_{\bullet i}$ in Eq. 5.2.

The eigenvalues and eigenvectors of the covariance matrix $\mathbf{COV(C}_{n\times m}\mathbf{)}_{m\times m}$ can then be calculated. The unit eigenvectors are reorganized as the rows of the matrix $\mathbf{P}_{m\times m}$ with the descending order of corresponding eigenvalues, which can be written as

$$\mathbf{P}_{m\times m}=\left[p_1; p_2; p_3;\cdots; p_m\right] . \tag{5.5}$$

With Eqs. 5.2 and 5.5, the matrix $\mathbf{S}_{n\times m}$ in Eq. 5.1 is calculated by

$$\mathbf{S}_{n\times m} = \mathbf{C}_{n\times m} \times \mathbf{P}^{T}_{m\times m} . \tag{5.6}$$

All unit eigenvectors are orthogonal with each other, and the sequence of eigenvectors in $\mathbf{P}_{m\times m}$ represents the order of deviation in this direction, which means that the first unit eigenvector defines the direction with the maximum deviation with respect to the mean of the original dataset. Using a small number of eigenvectors, such as the first k ($n = k$), the original dataset can be described with lower-dimensional variables as

$$\mathbf{D}^{*}_{n\times m} = \begin{bmatrix} d_{1\bullet} \\ d_{2\bullet} \\ \vdots \\ d_{n\bullet} \end{bmatrix}^{*} = \bar{\mathbf{D}}_{n\times m} + \mathbf{S}_{n\times k}\mathbf{P}_{k\times m} = \bar{\mathbf{D}}_{n\times m} + \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1k} \\ s_{21} & s_{22} & \cdots & s_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nk} \end{bmatrix}\begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{k1} & p_{k2} & \cdots & p_{km} \end{bmatrix}, \tag{5.7}$$

where $\mathbf{D}^{*}_{n\times m}$ is the approximation of the initial geometry dataset by its first k ($k < n$) principal components. $\mathbf{S}_{n\times k}$ and $\mathbf{P}_{k\times m}$ are the first k columns and rows of PCS and PC matrices, respectively. In Eq. 5.7, for a specific data set of geometry and a determined k value, the PC matrix $\mathbf{P}_{k\times m}$ and mean matrix $\bar{\mathbf{D}}_{n\times m}$ are unique, and the elements in each row of matrix $\mathbf{S}_{n\times k}$ measure the weight of each PC for a specific sample corresponding to this row. The geometric variations among different samples in the initial geometry dataset can be presented by different values in $\mathbf{S}_{n\times k}$. Thus, changing the value of all PCSs would move the locations of control points and thus change the geometry. Therefore, the PCSs can be used as the design variables in the shape design to govern the locations of control points implicitly. Through the inverse process of PCA with calculated and selected k PCSs and PCs, the updated geometry model can be reconstructed.

5.2.2 The node-based design method combined with DMM

The aim of this new design method is to design structures with complex geometries to satisfy performance requirements using a reduced number of design variables. The whole procedure can be divided into four main steps, as shown in Figure 5.1.

In the first step, the geometry of the part to be designed (i.e. an S-beam) is parameterized. In other words, the continuous profile is represented by discrete NURBS control points. The initial geometry dataset is obtained by collecting the geometric models of several existing designs. They are all parameterized with control points, which determine the envelope (i.e. upper and lower bounds) of the structural geometry. In this procedure, the initial geometry (control points) dataset is generated.

Using the geometry dataset, PCA is performed in the second step. The PCs and PCSs are calculated for the geometry dataset, and the dataset is compressed using these. The number of PCSs can be determined by the designer considering a number of factors such as accuracy and computational cost. A larger number of PCSs increases modeling accuracy but requires more computational power. These PCSs are adopted as the design variables in the subsequent design process. The range for each PCS value is determined, and thus the design space is created.

In the third step, within the design space defined in Step 2, the geometry can be modified after decompression by adjusting the values of PCSs. A large number of design alternatives are generated in this way, and they are then decompressed to recover the geometry and are further converted into FE models to simulate their response(s) under specific loading conditions (such as frontal or side impact). Although the nodes in the FE

model can be directly used as the control points, the large local location change of these nodes will cause severe mesh distortion, lower the computational accuracy and potentially even stop the simulation process. Therefore, instead of using FE nodes, NURBS control points are employed in this study, as these are related only to the geometry and are independent of the mesh. After the structural profiles are generated from the NURBS, these surfaces are then remeshed. In the case study of this chapter, SEA and crush force efficiency (CFE) are the structural responses of interest. They are calculated, and the simulation results together with the design variables form the simulation dataset.

In Step 4, the simulation dataset obtained in the third step is mined. The decision tree method is used to identify the interrelationships of design variables and the effect of each design variable (i.e. each PCS) on the results. Design rules are also generated from the decision tree. Based on the design rules derived in this step, new designs can be produced and should be evaluated by FE simulation to demonstrate the effectiveness of the new method.

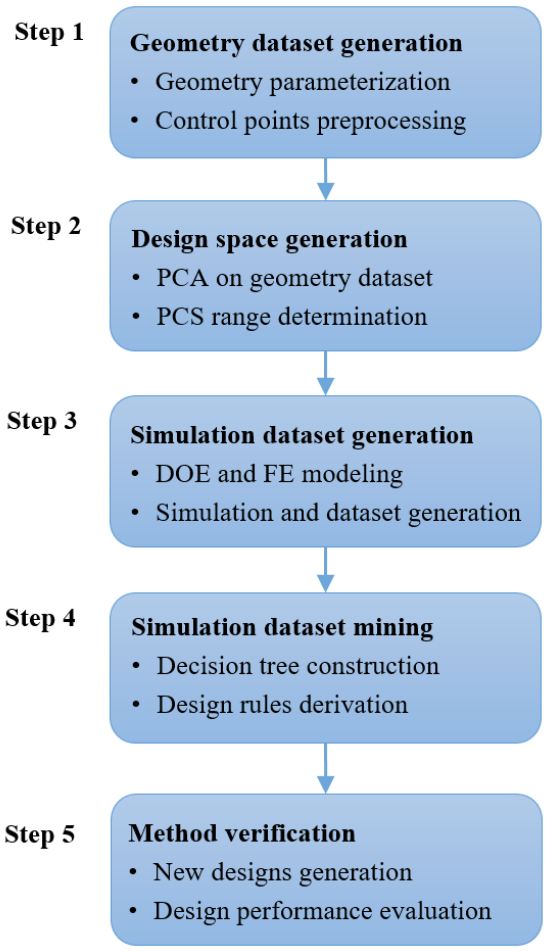


Figure 5.1. Workflow of the new node-based design method with DMM.

5.3 Application of DMM to the S-beam design

5.3.1 Generation of an initial geometry dataset for PCA

In this study, the S-shaped beam was assumed to have a complex shape. The envelope of the initial design (i.e. upper and lower corridors of the component geometry) was determined using six existing designs on commercially available passenger cars (https://www.nhtsa.gov/crash-simulation-vehicle-models#ls-dyna-fe), namely, Buick Regal, Dodge Neon, Honda Accord, Toyota Camry, Toyota Corolla and Toyota Yaris.

All passenger cars selected were within a similar weight class to ensure that the S-beams on the vehicles are of similar size but with different shapes.

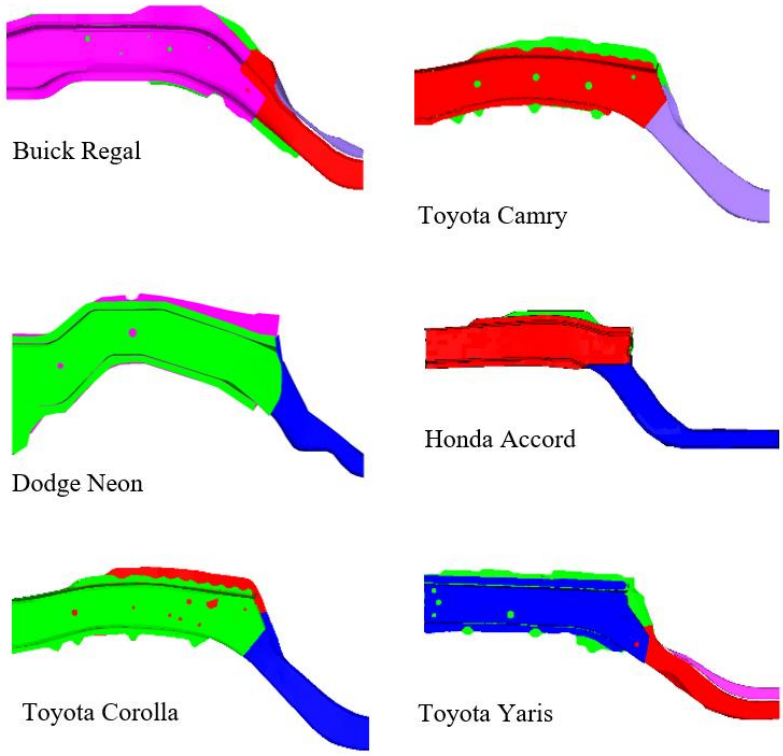
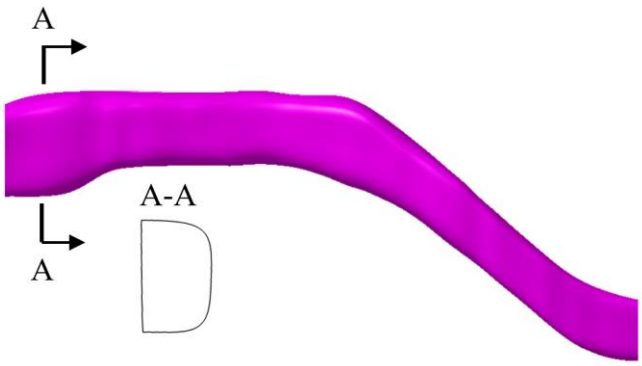


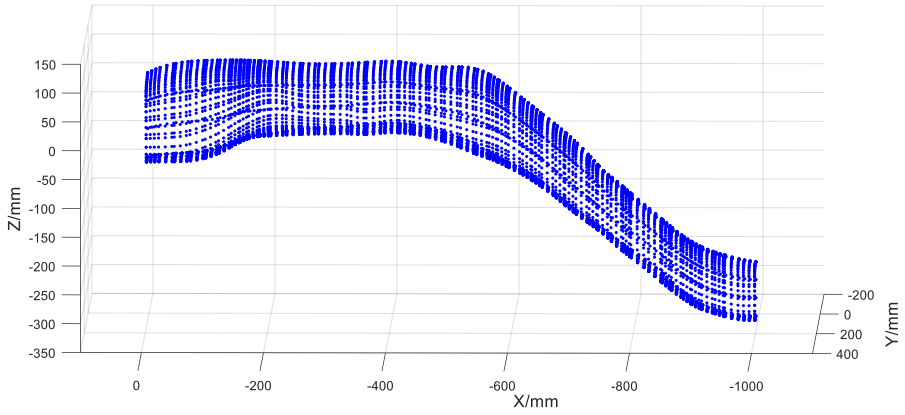
Figure 5.2. CAD models of the S-shaped beams from six passenger cars (different metal sheets are denoted with different colors).

The CAD models of the six beams were aligned at the frontal (left) end and were all cut off at the same 1.0-m length in the longitudinal direction, as shown in Figure 5.2, in which different colors represent different parts of the beam. The S-beam of the Buick Regal is used as an example to illustrate the control point generation procedure, which was presented in Figure 5.3. To simplify the geometry and subsequent modeling work, some unnecessary features such as small holes were removed. These small features, as well as thickness variations of different parts, were not modeled in the next steps. The S-

beam was redesigned as a single part, and only the basic size was kept, to fit the dimensions of the vehicle. It should be emphasized that the only purpose of using the six existing designs was to determine the dimensional range of initial designs.



(a)



(b)

Figure 5.3. (a) Geometry model for the S-beam on the Buick Regal and (b) its corresponding NURBS control points.

Adjusting the positions of these control points changed the profile of the structure, and new designs were achieved. Using this beam as the baseline model, its control points were mapped to the surfaces of the other five beam geometry models to obtain their

control points. After this geometry preprocessing step, all six beam models had the same number of control points, and these control points had the same relative positions.

The control points of all six models were then exported to form the initial geometry dataset. With this dataset, the envelope of initial geometry models (i.e. upper and lower corridors of the component geometry) was determined in the 3D space. In other words, the geometry variations of the new designs are limited to within this domain determined by the six original geometric models. In this study, 7,289 control points were generated for each beam. If the control points were used as design variables, considering the three degrees of freedom (x, *y* and *z*) for each control point, the total number of design variables would be 21,867. The optimal design or near-optimal design for this structure is costly.

5.3.2 Construction of design space

In this step, PCA was conducted on the geometry (control points) dataset built in the last step. The PCA converted the design variables from the coordinates of control points to PCSs. As a result, a 6 × 21,867 PCS matrix was generated, and a 21,867 × 21,867 PCs matrix was constructed. Generally, a larger value for *k* in Eq. 5.7 would increase the accuracy of modeling but also increase the computational cost. In the present study, six (n = 6) samples were used to demonstrate the proposed method, and only five nonzero columns of $\mathbf{S}_{n\times m}$, in Eq. 5.1, were calculated. Hence, in this study, k was set to equal 5, with all nonzero PCS elements included, which could ensure the original data would regenerate completely.

The calculated PCS values for the six beams and their ranges are listed in Table 5.1. The highest and lowest values for each PCS were set as the upper and lower bounds. The design space was determined by scaling up the initial range of PCSs with a scaling factor of 1.2. In this way, the design could be implemented in a larger space to reduce the chance that good design alternatives would be missed. Even larger scaling factors, such as 2, would increase the design space, but they could significantly change the size of the structures so that they may not fit the vehicles well. The larger scaling factor could also lead to geometry deterioration and then some unrealistic structural profiles. The geometry could be changed between the upper and lower bounds determined by the design space (the scaled PCS range in Table 5.1) and controlled by the combinations of PCS values.

Table 5.1

PCS values and their ranges for the six S-beams

PCS		1st PCS	2nd PCS	3rd PCS	4th PCS	5th PCS
PCS values	Buick Regal	1,164.4	788.0	-56.8	1,429.4	5.5
	Honda Accord	-5,156.0	2,083.6	164.6	-537.8	-0.4
	Toyota Camry	716.7	-858.4	1,126.4	-129.4	-126.4
	Toyota Corolla	728.1	-911.1	1,154.9	-205.3	122.5
	Dodge Neon	7,248.2	495.6	-1,001.4	-570.2	-1.1
	Toyota Yaris	-4,701.4	-1,597.7	-1,387.8	13.3	-0.1
PCS ranges	Max	7,248.2	2,083.6	1,154.9	1,429.4	122.5
	Min	-5,156.0	-1,597.7	-1,387.8	-570.2	-126.4
Scaled PCS ranges		8,697.9	2,500.3	1,385.9	1,715.3	147.0
		-6,187.2	-1,917.3	-1,665.4	-684.2	-151.7

5.3.3 Generation of a design dataset for DMM

In this step, within the design space defined in the last section, the geometry of the S-beam can be modified after decompression by adjusting the values of PCSs. A large number of design alternatives (DOEs) were created in this way and converted into FE models so their crash responses could be simulated. The simulation results together with the design variables formed a simulation dataset for further data mining.

In the present study, the objective is to apply the PCA algorithm to achieve a fast and robust structural design and to use a simple case study to verify its performance. In actual vehicle crashworthiness design practice, the proposed design method can be applied to all of the main energy-absorbing structures in the car simultaneously to implement the system-level design with more realistic boundary conditions.

The S-beam of the Buick Regal described above was used as an example and baseline model to show the FE modeling procedure. The basic element was defined as an 8 mm × 8 mm shell (Nguyen et al., 2014) with a constant thickness (3 mm). In the present study, the thickness was not taken as a design variable, since the design is implemented by changing the locations of the control points, which may produce an effect equivalent to thickness variation. In addition, a uniform thickness could reduce the number of design variables since each original part has its own thickness value and make it possible to model the entire structure using one single part. The value of the thickness (3 mm) was determined based on the results of our previous study (Du and Zhu, 2018) and the range (1 to 4 mm) reported in other literature (Nguyen et al., 2014, Khakhali et al., 2010b, Fang

et al., 2017, Tian et al., 2015, Khakhali et al., 2010a). The material, loading and boundary conditions were set to match those in Chapter 4, and the details are not repeated here. The simulated deformation mode of the baseline model and the internal energy distribution is illustrated in Figure 5.5(a). The results show that the impact energy was mainly dissipated at three locations (marked with red circles 1, 2, 3) due to large plastic bending. Figure 5.5(b) shows the impact force–displacement relationship. The response is consistent with previous studies in the literature in terms of the magnitude and shape of the curve (Fang et al., 2017, Khakhali et al., 2010a, Khakhali et al., 2010b, Nguyen et al., 2014, Tian et al., 2015, Zhou et al., 2011, Abbasi et al., 2015, Yin et al., 2013).

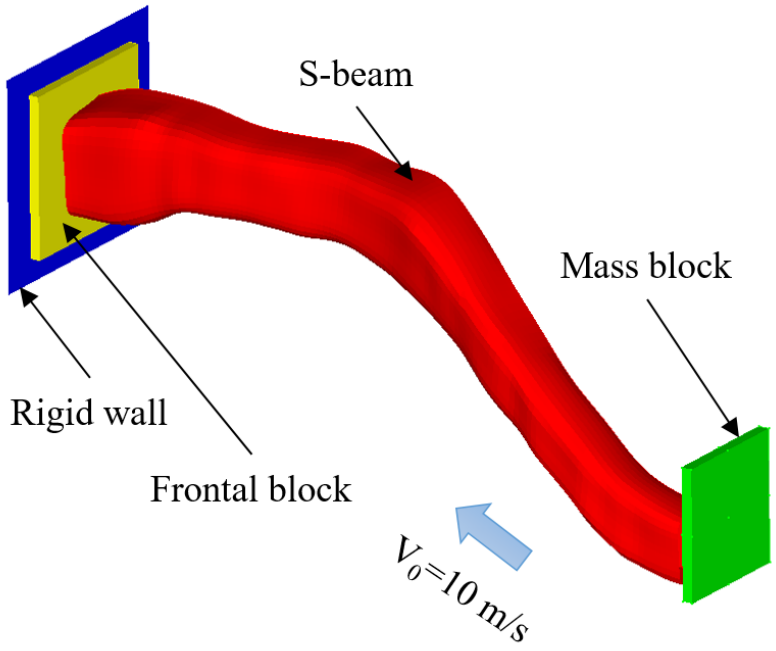


Figure 5.4. FE model setup for the S-beam under frontal impact.

Together with the SEA used in Chapter 4, another response, CFE, was employed as the second design objective to evaluate the performance of the S-beam. Both parameters taken together can generally reflect the overall crashworthiness of the structure. The two quantities are defined in Eq. 4.1 and 5.8, respectively (Abbasi et al., 2015, Yin et al., 2013).

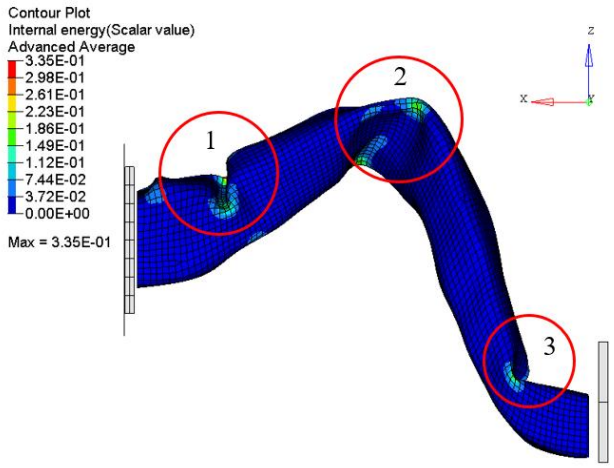
$$CFE = \frac{F_{mean}}{F_{\max}} , \qquad (5.8)$$

where E_{int} is the internal energy absorbed by the S-beam, and m is the mass; F_{max} is the peak force, and F_{mean} is the mean crush force, which is determined by

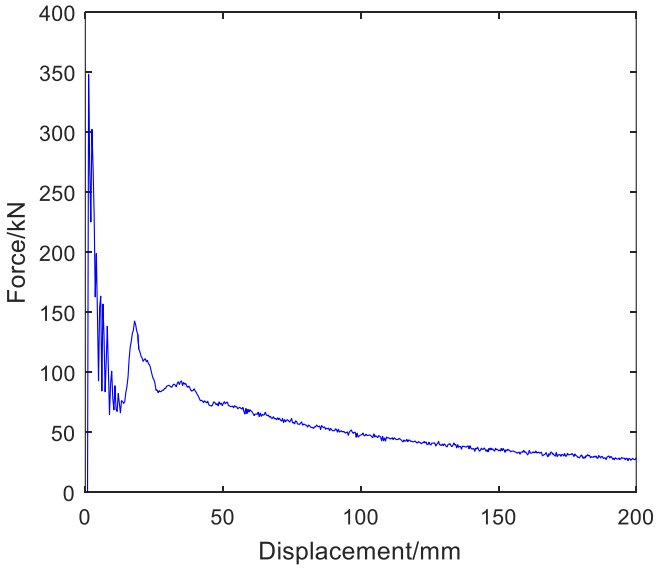
$$F_{mean} = \frac{E_{\text{int}}}{S_E} , \qquad (5.9)$$

where E_{int} is the absorbed internal energy, and S_E is the maximum crush distance.

Using the five PCSs as design variables, within the design space determined in Table 5.1, a total of 8,400 CAD models were built by changing the values of the five PCSs. To date, no research has been devoted to discussing the effect of DOE value, and it has been generally accepted that the number of designs depends on specific problems. Based on our previous studies on similar design problems, 8,400 design cases could be deemed as sufficient in the present study. Simulations were then conducted with the aforementioned loading and boundary conditions.



(a)



(b)

Figure 5.5. Simulation results of the S-beam baseline model: (a) deformation mode and distribution of internal energy per unit volume (unit: J/mm^3); (b) impact force-displacement relationship.

5.3.4 Data mining on the design dataset

Before the data could be mined, the design alternatives were labeled in terms of simulated SEA and CFE performances. The labeling criteria can be set in an arbitrary way based on the costumer's requirements or the designer's experience. In the present study, the following criteria were used:

g: SEA > 1,800 J/kg and CFE > 0.1

m: 1,400 < SEA ≤ 1,800 J/kg and 0.08 < CFE ≤ 0.1

p: SEA ≤ 1,400 J/kg and CFE ≤ 0.08

The distribution of the simulation results (SEA and CFE) and design alternatives with corresponding labels is illustrated in Figure 5.6. Data mining was performed on the labeled simulation dataset by generating a decision tree, which is illustrated in Figure 5.7.

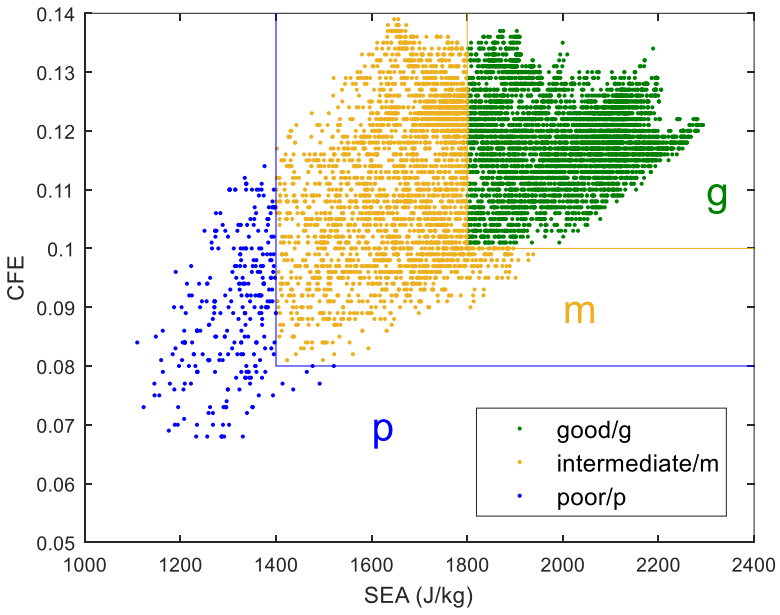


Figure 5.6. Distribution of the SEA and CFE responses of all design alternatives with corresponding labels.

5.4 Results and discussion

5.4.1 Decision tree analysis

In Figure 5.7, the whole decision tree includes five leaf nodes that correspond to five branches (labeled as b1 through b5). Each branch represents a decision-making process or design rule for a particular design subset. The design started from the root node (i.e. the first PCS), which was identified as the most critical design variable. If it was greater than -4, then we moved to the right path and checked its child node, which was the first PCS again. If its value was less than or equal to 3,883.9, then this design path produced an "intermediate/m" design. In this way, all of the design alternatives were classified with the decision tree, and the interrelationships between design variables as well as the ranges of their values were clearly revealed. Each branch of the decision tree represents a decision-making rule.

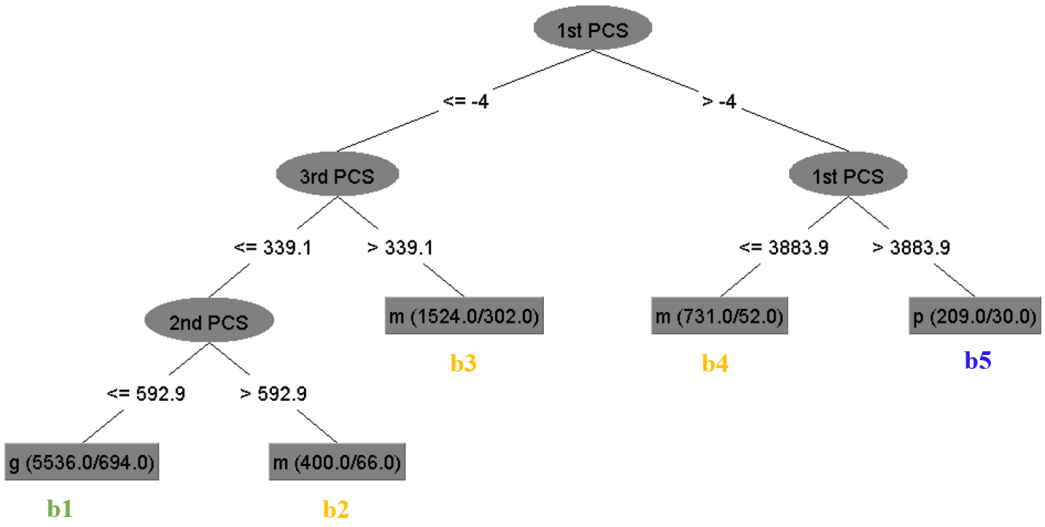


Figure 5.7. Decision tree generated by mining the S-beam simulation dataset.

Figure 5.7 indicates that out of the five branches of the decision tree, three branches (b2, b3 and b4) yielded “intermediate/m” designs and only one branch produced “good/g” designs and one “poor/p” designs. The numbers on each leaf node indicate the accuracy of the partition. For example, on leaf node b1, g (5,536/694) means that 5,536 samples were classified as “good/g” designs with 694 non-“good/g” design alternatives included. The classification accuracy of each branch can be estimated by the percentage of correctly classified instances. Figure 5.7 indicates that the classification accuracy levels for “g,” “m” and “p” are 87.5%, 84.2% and 85.6%, respectively. The average classification accuracy for all of the three groups is 86.1%, which can be deemed as good enough for data mining (Zhao et al., 2010).

5.4.2 Decision tree validation and bending modes study

Additional validation on the decision tree performance was conducted by generating 300 new design alternatives using the decision rules shown in Figure 5.7. In each performance class, 100 new designs were created using LHM (Chen et al., 2013, Du and Zhu, 2018), which could ensure relatively uniform distribution in the corresponding design subspace. All of the 300 new designs were converted to FE models so their structural responses could be simulated under the aforementioned loading conditions shown in Figure 5.4. The simulation results (i.e. the SEA and CFE values) are plotted in Figure 5.8, which shows reasonable data mining accuracy. The calculated correct classification rates are 90%, 88% and 88% for the “g,” “m” and “p” design groups, respectively. With these new design cases, the accuracy of the decision tree predictions can be verified.

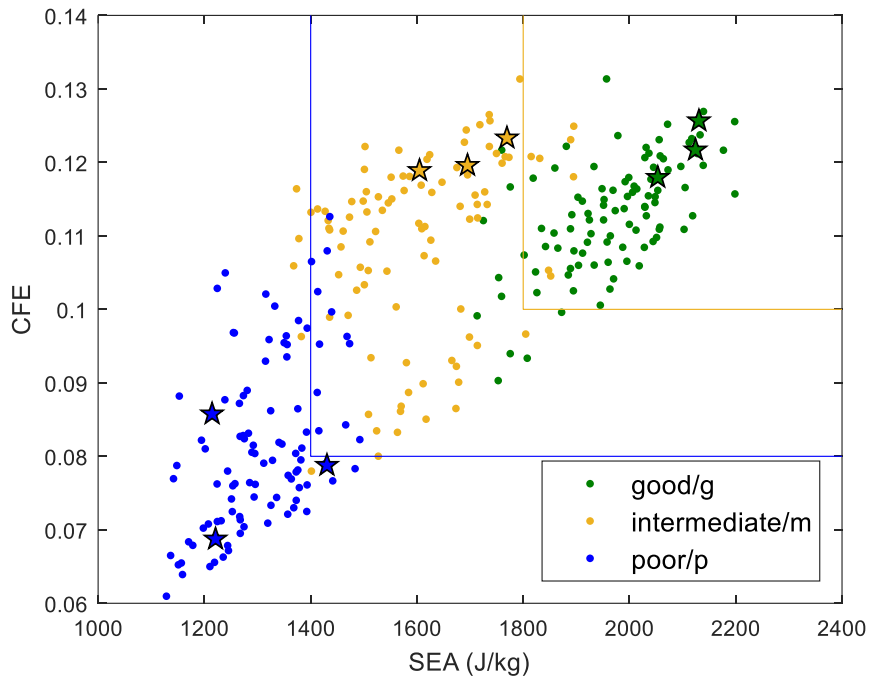
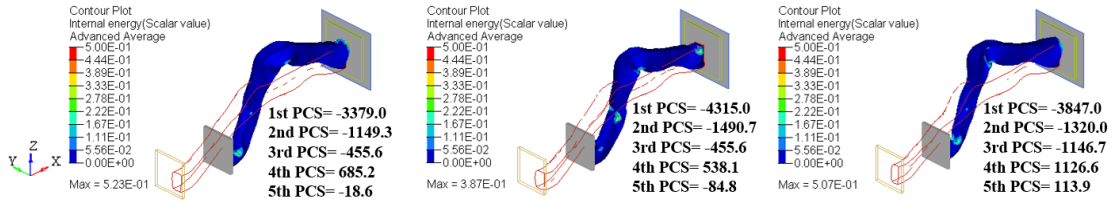


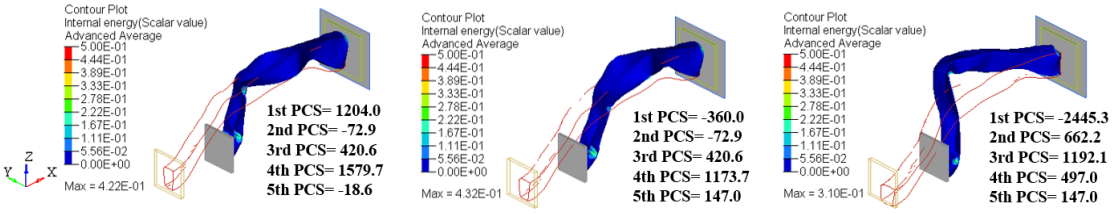
Figure 5.8. A plot of 300 new design alternatives generated using the decision rules shown in Figure 5.7 (100 in each performance class) and their simulation results. Three design cases (denoted with stars) in each category were selected randomly and their bending modes were studied.

Three designs were randomly selected from each of the three design groups, as denoted with stars in Figure 5.8, and used as samples to study the effect of deformation mode on structural responses. The simulated deformation modes at the maximum crush for these nine designs are compared in Figure 5.9, together with the internal energy distribution. The results show that the deformation of all of the beams is dominated by large plastic bending. Three good designs in the b1 group tend to bend in the XZ plane of the vehicle system, while the poor designs in the b5 group are bent in the XY plane. Those "intermediate" designs in the b2, b3 and b4 groups exhibit a mixed mode combining the

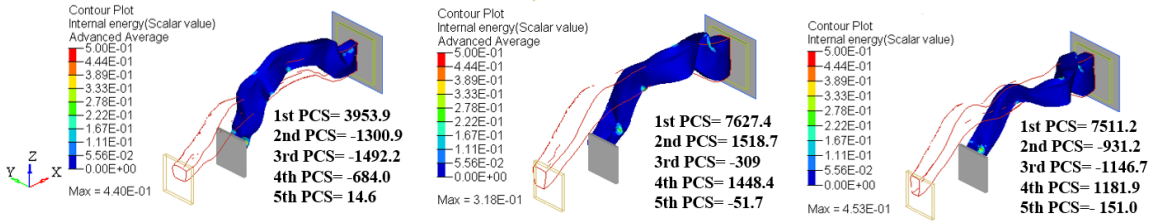
deformation patterns of good and poor designs. The findings suggest that the structural responses of an S-beam rely on the deformation modes, which are further dependent on geometric designs. In the PCA method, the geometric information in the six initial designs is well described by the five design variables (i.e. the PCSs) since the geometry could be fully recovered from the compressed dataset using these design variables. The information hidden in the design dataset can be effectively revealed through the DMM.



(a)



(b)



(c)

Figure 5.9. Deformation modes and internal energy density (unit: J/mm^3) of representative S-beam designs in the three groups of designs: (a) b1/g; (b) b2, b3 and b4/m; and (c) b5/p.

5.4.3 Design rules

In this section, the geometric designs in the three performance classes (g, m and p) are compared to reveal the relationship between the beam features and structural responses. The design results in the three "intermediate/m" branches (b2, b3, and b4) exhibit similar levels of performance, and the b4 group has the highest classification accuracy. Therefore b4 was selected as the representative of intermediate/m design. Figure 5.10 illustrates the graphic comparison of the ranges of the design variables in the three design groups, together with the whole design space from Table 5.1. These ranges (i.e. upper and lower bounds) of the PCSs were determined through the decision tree shown in Figure 5.7, and they quantitatively reflect the geometric features, namely, the upper and lower bounds of the beam profiles.

The comparison results show that the ranges of the first three PCSs (the first, second and third PCSs) are subsets of the design space. The ranges of first PCS values for "good/g" and "poor/p" designs tend to be close to the lower and upper bounds of design space, respectively, while the "intermediate/m" designs are in the middle. In terms of the second PCS and third PCS, the good design groups exhibit larger ranges, which are still close to the lower bound of the design space, while the other two groups have ranges that span the entire design space. For the fourth PCS and fifth PCS, all groups have the same range, which spans the original design space. This finding indicates that the first PCS is the key design variable for distinguishing the three design groups, and it has the greatest impact on the design results. The second PCS and third PCS can distinguish the good designs from the other two groups, but they have less impact on the results than does the first PCS. The fourth PCS and fifth PCS should have little influence on the classification,

since all of the groups have a range equal to the full design space. This result is consistent with the predictions of the decision tree in Figure 5.7, where fourth PCS and fifth PCS are not included.

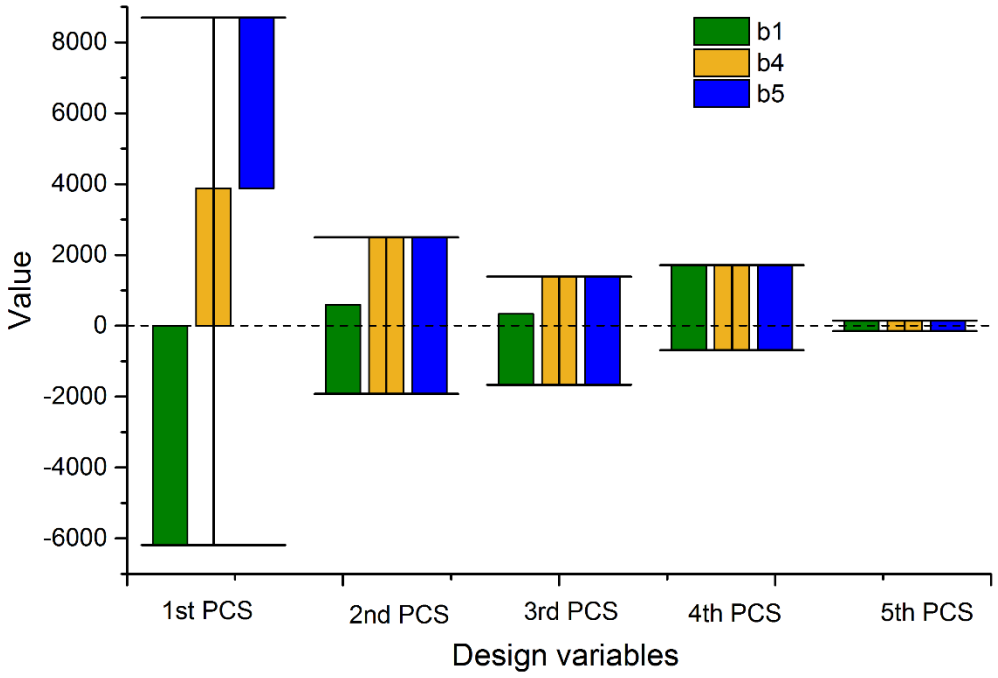
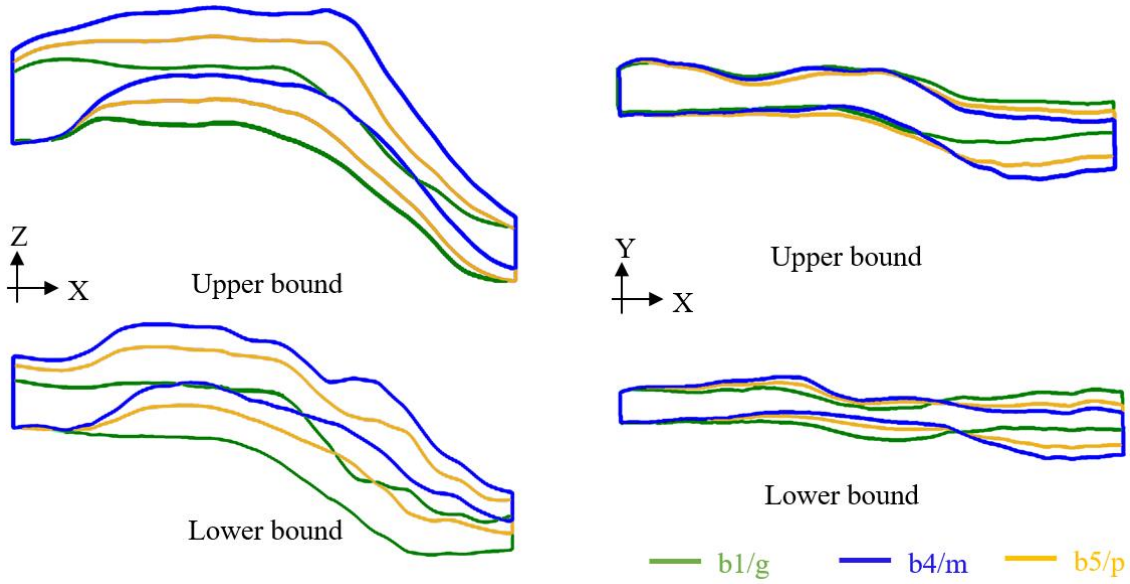


Figure 5.10. Ranges (i.e. upper and lower bounds) determined through the decision tree of the PCSs in the three design classes (b1, b4 and b5), together with the range of the whole design space as shown in Table 5.1.

An additional comparison of the geometric profiles of the beams in the three design groups is made in Figure 5.11 that considers only extreme conditions, meaning the profiles of the beams corresponding to the PCS values at the lower and upper bounds. As seen in the side view in Figure 5.11(a), the "good/g" designs tend to have a straighter profile in the longitudinal direction. In other words, larger angles (or smaller curvature) in the transition area of two segments in the XZ plane would lead to higher

crashworthiness performance. A similar trend can also be observed in the top view (Figure 5.11(b)), where the good designs have a smaller curvature in the XY plane. Actually, the straight shape of the S-beam would increase its stiffness and its bending moment, which influences energy absorption. These findings suggest that a good design is characterized by a smaller curvature in the XZ and XY planes.



(a) (b)

Figure 5.11. Upper and lower bounds of the geometry profiles of the three design groups (b1/g; b4/m; b5/p): (a) side view; (b) top view.

5.5 Summary

In this chapter, the DMM was implemented in node-based shape design based on PCA theory to handle high-dimensional design variables. This approach could be used to design vehicular structures with complex geometries to satisfy the requirements of crash energy-absorbing performance using a reduced number of design variables and to

discover the clear implicit interrelationships of these variables. An S-beam with an irregular shape was used as a case study to demonstrate the method's performance.

In this design process, the geometry of the beam was initially parameterized by a large number of NURBS control points. The initial geometry dataset was obtained by collecting the geometric models of several existing designs. PCA was then performed on this dataset, and the PCs and PCSs were calculated, with the PCs regarded as the extracted geometric features. The PCSs were adopted as the design variables in the subsequent design process to modify the beam's geometry. A large number of design alternatives were generated, and the geometry was then recovered and further converted into FE models to simulate the designs' responses to a frontal impact. Two structural responses, SEA and CFE values, were calculated and stored in the simulation dataset. After the simulations were completed to generate a dataset, the decision tree method was used to identify the interrelationships of design variables and the effect of each design variable (i.e. each PCS) on the results. Design rules, that is, the workflows to determine the values of the design variables, were also derived. The results demonstrate that the new PCA-based approach can be used to efficiently design a complicated structure with an irregular shape. The relationships among geometric features, failure behavior and crash performance were also revealed. The findings suggest that the data mining process enhances the design procedure by uncovering the important interrelations hidden in the design datasets, which helps designers better understand the design problem and improve design performance.

Chapter 6 Study of surrogate modeling as applied in structural analysis

6.1 Introduction

In the previous chapters, a large number of FE simulations were run to generate datasets for data mining. The high computational cost of numerical modeling and these many simulations could be a concern. To bring down the computational cost, surrogate models (Boursier Niutta et al., 2018, Du and Zhu, 2018) have been widely adopted as substitutes for FE simulations and used to predict structural responses. A surrogate model can approximate the system response by fitting a mathematical model using a design dataset. This process is known as model training. Once the model is trained, more designs can be generated by this surrogate model at a relatively low cost. To high accuracy surrogate model, machine learning algorithms (MLAs) have been applied to develop surrogate models because of the powerful learning abilities and high prediction accuracy of these algorithms (Ben Salem and Tomaso, 2018, Mehmani et al., 2017, Shi et al., 2012, Díaz-Manríquez et al., 2011, Couckuyt et al., 2011, Jin et al., 2001, Zhao and Xue, 2010). In the open literature, however, MLAs are often used as "black-boxes" in surrogate modeling, without comprehensive analyses of the effects of their parameters (Song et al., 2012). Lacking this knowledge makes it very difficult to fully understand and improve the performance of these algorithms in engineering design practice.

A typical MLA includes two types of parameters: model parameters and hyperparameters (Hutter et al., 2011, Snoek et al., 2012, Bardenet et al., 2013). Model parameters are constants, which can be optimized and trained with datasets, while hyperparameters are the control parameters governing the training process and model structure. They should

be specified prior to model training and have a great influence on model accuracy and robustness (Snoek et al., 2012, Bardenet et al., 2013, Yogatama and Mann, 2014, Klein et al., 2016). To date, very few studies have been reported that have studied the effects of hyperparameters associated with MLAs in engineering structural design, and their values are often determined using very rough estimation methods, such as several trial-and-error experiments. A more detailed study of these important parameters is imperative for surrogate modeling.

In this chapter, the performances of four machine learning methods, namely Gaussian process regression (GPR), support vector machine (SVM), random forest regression (RFR), and ANNs are compared. These MLAs were selected because they are the most frequently used methods in structural engineering, according to the literature review as summarized in Table 6.1. In this study, the levels of accuracy of these four methods are studied by modeling four typical structures with various geometries, materials, and loading and boundary conditions. Their associated hyperparameters are also optimized to compare optimal performances. One of the MLAs studied here (i.e. GPR) is also used in Chapter 7 to build a surrogate model for uncertainty analysis.

The remaining parts of this chapter are organized as follows: Section 6.2 presents the multi-objective hyperparameters optimization strategy. Section 6.3 describes the four structural analysis examples and corresponding development of the FE model used to generate the datasets for training the MLAs. Section 6.4 details the surrogate modeling and hyperparameter optimization with a comprehensive analysis of the effects of hyperparameters in GPR. The chapter is summarized in Section 6.5.

Table 6.1

Literature on machine learning algorithms applied as surrogate modeling techniques in structural engineering

Literature	Algorithms	Applications
(Mukherjee and Deshpande, 1995)	ANN	RC beam design
(Kapania and Liu, 1998)	ANN	An aerospace continuum beam design with lattice structure
(Nagendra et al., 2004)	ANN	A turbine disk optimal design prediction using various input parameters
(Lee et al., 2007)	ANN	A suspension design
(Tang and Chen, 2009)	SVM	Robust design of sheet metal forming process and demonstrated by a cup drawing example
(Guo and Bai, 2009)	SVM	Reliability analysis of deployable mechanism for huge space station
(Pan et al., 2010)	SVM	B-pillar weight minimization under roof crush and side impact loading
(Wang et al., 2010)	LS-SVM	The response prediction of a cylinder and whole-vehicle crash
(Huang et al., 2011)	GPR	Optimal design of aeroengine turbine disc
(Zhu et al., 2012)	SVM	Vehicle crashworthiness and lightweight design under roof crush and frontal crash conditions, respectively
(Zhang et al., 2012)	GPR	To optimize the crashworthiness of a foam-filled bitubal square column
(Haleem and Gan, 2013)	RFR	To predict the severity of traffic accident with respect to some numerical and categorical factors
(Song et al., 2013)	RSM, GPR, SVM and RBF	A foam-filled tapered thin-walled structure response prediction
(Yin et al., 2014)	RSM, RBF, GPR and SVM	Foam-filled, multi-cell, thin-walled structures

(Lukaszewicz et al., 2014) and (Lukaszewicz and AG, 2015)	RFR	To predict the influence of manufacturing variations on structure impact performance
(Rodriguez-Galiano et al., 2014) and (Rodriguez-Galiano et al., 2015)	ANN, DT, RFR and SVM & RFR	Used to map the statistical distribution of mineral prospectivity based on images
(Fang et al., 2014)	GPR	To explore the multi-objective design of foam-filled bitubal structures under uncertainty
(Ferreira and Serpa, 2015)	RSM, GPR, RBNN and SVM	Analytical and real-world vehicle crashworthiness analysis
(Fang et al., 2015)	GPR	For optimization of multi-cell tubes
(Tang et al., 2016)	RFR	To predict the response of a train crash with respect to different parameters
(Liu et al., 2016)	GPR	Demonstrated by a thin-walled box beam and a long cylinder pressure vessel example
(Yu et al., 2018)	ANN	Deep learning for topology design without iterations
(Raihan et al., 2018)	RFR	Used to find important variables for accident data explaining
(Duan et al., 2018)	SVM	Multi-objective optimization of a new vehicle longitudinal beam
(Palar and Shimoyama, 2018)	GPR	Airfoil models design

Note:

1) GPR is similar with the Kriging method.

2) RBF: radial basis function; RBNN: radial basis neural network; DT: decision tree; LS-SVM: least square-SVM

6.2 Hyperparameter optimization

As important parameters in MLAs, hyperparameters have a great effect on model accuracy, and their values should be assigned before training. Optimization of hyperparameters could improve the learning ability of the MLA. To achieve this optimization, a strategy is proposed to tune their hyperparameters automatically. Four MLAs selected for the surrogate modeling study are introduced under the context of surrogate modeling in Appendix A.

6.2.1 Measures

Prior to the optimization process, the measures to be used as surrogate model accuracy indices should be determined. In this study, the root mean square error (RMSE) is used as a measure, as expressed in Eq. 6.1.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (\tilde{f}(x^i) - f(x^i))}{N}} \quad , \qquad (6.1)$$

where $\tilde{f}(x^i) - f(x^i)$ is the error of response prediction relative to the real value of the *i*th design (x^i), and N is the number of training or test data points.

Maximum absolute error (MXAE) is also used to assess the maximum prediction error using Eq. 6.2, which represents the local error control ability of the MLA.

$$MXAE = \text{Max}(\left| \tilde{f}(x^i) - f(x^i) \right|) \qquad (6.2)$$

Finally, the training computational time (T) of MLAs is also evaluated. This helps inform a decision on the selection of a suitable MLA and dataset size in terms of computational cost.

6.2.2 Multi-objective hyperparameter optimization

Hyperparameter tuning for MLAs is often a manual process involving a few training trials. However, optimal values cannot be obtained in this way. In addition, the effects of these important parameters are not well understood, and therefore, the manually tuning results are not very informative (Bardenet et al., 2013, Yogatama and Mann, 2014). To better quantify hyperparameters, they can be optimized using a number of algorithms. In this work, sequential model-based optimization (SMBO) is selected, as it has been frequently applied in tuning hyperparameters. Its basic scheme is illustrated in Figure 6.1 (Yogatama and Mann, 2014, Klein et al., 2016).

Figure 6.1 shows that the first step in SMBO is to define the hyperparameters' space, including the objective measures of optimization, hyperparameters selection and their tuning domain. After that, the initial designs, that is, combinations of hyperparameters, are generated and evaluated by the MLA using the values of objective measures. The training dataset is then constructed with the determined hyperparameter values and corresponding measures, which can be used to train a surrogate model. The relationship between evaluation measures and hyperparameters is approximated by a surrogate model. Based on the surrogate model, optimal design alternatives can be generated by acquisition function and then evaluated by MLA. When the termination criterion, for example, the number of evaluations, is reached, this tuning process is complete. Otherwise, these new designs are added back to the training dataset and used to update the surrogate model.

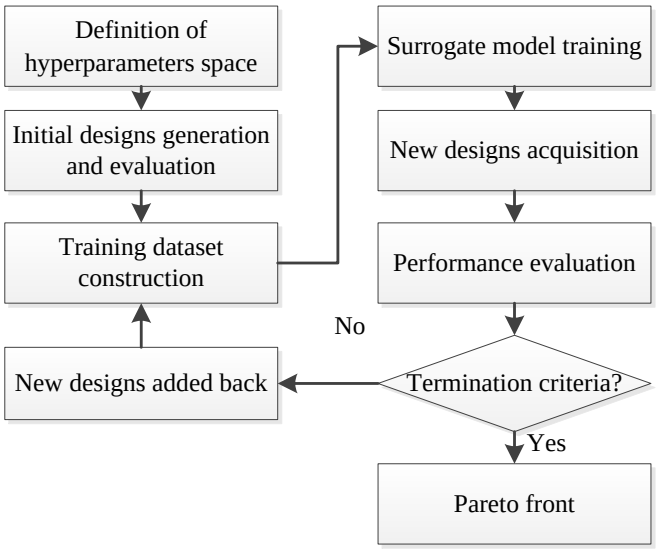


Figure 6.1. The algorithm for multi-objective hyperparameter optimization.

Some effective evaluation criteria can be used to achieve the proper trade-off between the exploration and exploitation of the generated designs, for example, the expected improvement (EI), the GP upper confidence bound, the maximum possibility of improvement, the minimum conditional entropy or the lower confidence bound (LCB) (Yogatama and Mann, 2014, Klein et al., 2016, Bischl et al., 2017). In this study, the lower confidence bound is used:

$$LCB(\boldsymbol{x},\varphi) = \hat{\mu}(\boldsymbol{x}) - \varphi \cdot \hat{s}(\boldsymbol{x}) \ , \qquad (6.3)$$

where $\hat{\mu}(\boldsymbol{x})$ and $\hat{s}(\boldsymbol{x})$ are the posterior mean and standard deviation, respectively, and φ is a constant. The workflow in Figure 6.1 is established and used to optimize the hyperparameters. At a specific training cost (100 evaluations in this study), 30 initial designs are sampled for a preliminary surrogate model.

6.3. Structural analysis cases for evaluation of MLAs

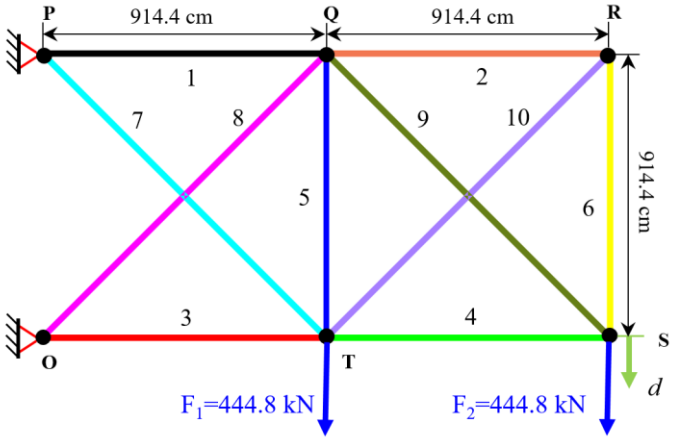
In this study, four representative engineering structures are taken as examples to compare the performances of the aforementioned MLAs and optimize their hyperparameters. In these four problems, the variations of geometry (bar, sheet and block), loading (static vs. dynamic) and boundary conditions (fully constrained and contact), as well as deformation modes (small vs. large deformation) are all considered. Hence, the methods and results associated with these case studies can be easily extended to a wide range of structures.

- **Structures subject to static loading**

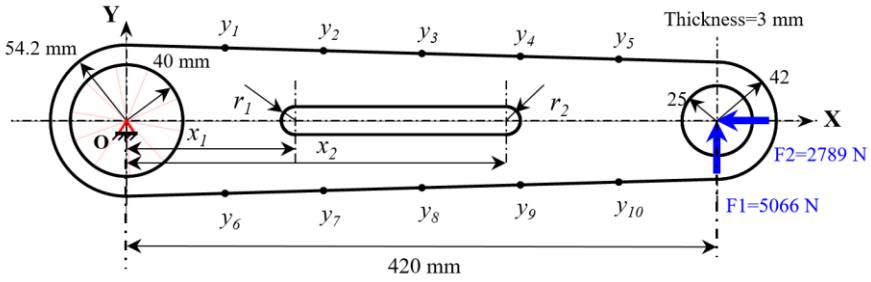
Under static loading conditions, two typical models are adopted, namely, a 10-bar planer truss (TbPT) (Lee et al., 2017) and a torque arm (TqA) (Kim and Chang, 2005, Van Miegroet, 2012, Cai et al., 2014, Bennett and Botkin, 1985), as shown in Figure 6.2, since they are frequently used as examples to verify structural design algorithms. In the TbPT model, the circular cross-section areas are taken as design variables with the range from 0.6 to 225.8 cm^2, and the other dimensions are listed in Figure 6.2(a). Two loads with the same magnitude (444.8 kN) are applied on joints T and S**,** and the vertical displacement of joint S is taken as the response of this system.

To design the TqA, geometric constraints must be considered. The constant thickness of this structure is 3 mm, and the distance between the two circle centers is 420 mm. The design variables and their ranges are listed in Table 6.2. The left circle is fully constrained, and loads are added to the right circle, as presented in Figure 6.2(b). To

avoid structural failure, the stress is limited to under 800 MPa, and the total mass is used as the objective.



(a)



(b)

Figure 6.2. Two structures under static loading: (a) TbPT and (b) TqA.

- **Structures subject to dynamic loading**

Under dynamic loading conditions, two crashworthy components are studied: the thin-walled S-shaped beam (ShB) discussed in our previous studies (Du et al., 2017, Du and Zhu, 2018) and a thin-walled octagonal multi-cell tube (OMcT) reported in literature (Bai et al., 2018). The geometry parameterization and FE models of these two components are illustrated in Figure 6.3. As critical energy-absorbing parts on a passenger car, these

structures can sustain large plastic deformation and dissipate a large amount of kinetic energy from an impact.

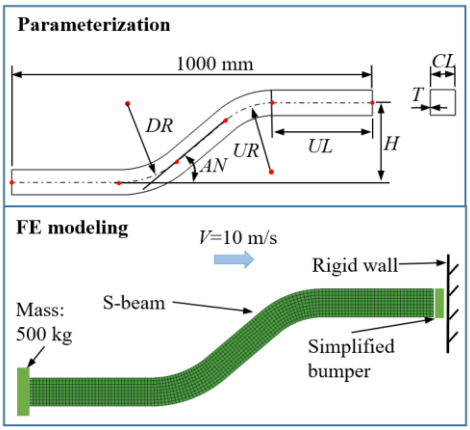
In Figure 6.3(a), the shape of the ShB is fully described by seven design variables, and its total length is 1,000 mm. Figure 6.3(a) also shows the FE model of the ShB, which is subjected to a frontal impact at 10 m/s. The SEA is set as the design objective in Eq. 4.1. In the OMcT model, the cross-section is composed of the inner and outer octagons with the ribs connected. The edge sizes of inner and outer octagons are 30 and 60 mm, respectively, and the total length is 310 mm. Different colors in Figure 6.3(b) represent ribs with different thicknesses. Each thickness is considered one design variable; therefore, there are nine design variables in total. The OMcT FE model impacts a fully constrained rigid wall (velocity: 30 km/h) with the rear end attached to a mass block (600 kg) to represent the vehicle inertia effect. The CFE is calculated as the design objective to assess the energy-absorption efficiency as defined by Eq. 5.8.

Table 6.2

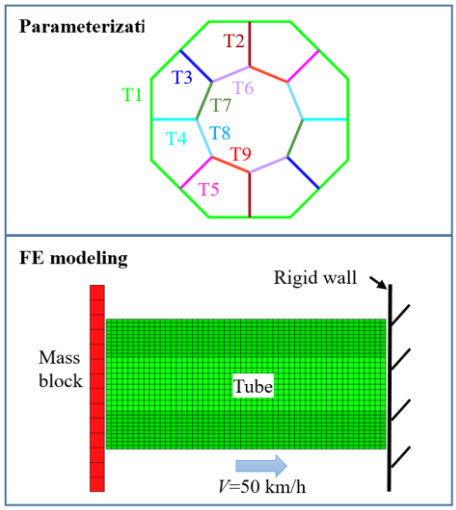
Design variables and their ranges for the TqA structure

Design variable	Initial value/mm	Range/mm	Design variable	Initial value/mm	Range/mm
y_1	52	[30, 62]	y_8	48	[22, 58]
y_2	50	[26, 60]	y_9	46	[18, 56]
y_3	48	[22, 58]	y_{10}	44	[14, 54]
y_4	46	[18, 56]	x_1	120	[60, 200]
y_5	44	[14, 54]	x_2	270	[110, 395]
y_6	52	[30, 62]	r_1	10	[10, 40]
y_7	50	[26, 60]	r_2	10	[5, 40]

The detailed material parameters and loading and boundary conditions can be seen in Chapters 4 and 5. These two models were validated in those studies, and the validation is not repeated here. In the current study, the impact speed is increased to 50 km/h to represent the loading condition defined in the NCAP standard (National Highway Traffic Safety Administration https://www.nhtsa.gov/laws-regulations).



(a)



(b)

Figure 6.3. Two structures under dynamic loading: (a) the vehicle frontal S-shaped side beam (ShB) and (b) the octagonal multi-cell tube (OMcT), together with design variables and FE models.

6.4. The performance of MLAs

Once the basic structural FE models are developed, they are used to generate datasets for MLA training by FE simulations. Using the simulation dataset, the best performance of the MLAs is reached using the hyperparameter optimization process. Subsequently, the optimized ML models are compared in terms of their structural response prediction capability.

6.4.1 Data collection and preprocessing

After the four structural FE models are developed, they are used to generate datasets by DOEs and FE simulations. In this study, the uniform LHM is used to sample the design points in the design space. Then, 1,000 designs are sampled for each structure, the corresponding FE models are developed and their responses are calculated using FE simulations. In these 1,000 designs for each structure, 800 designs are used for training and 200 for validation.

It is also noted that in these four structure models, design variables, such as *H* and *T* in ShB, may be of different orders of magnitudes. This difference would mean features with larger values would outweigh features with smaller values (Mohamad and Usman, 2013). Therefore, normalization is used to scale the values of different design variables into the same range [0, 1] by

$$\bar{X} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}, \qquad (6.6)$$

where, for a specific variable, $\bar{X}$ is the normalized value, X is the variable value, and $X_{\max}$ and $X_{\min}$ are the maximum and minimum values of this variable, respectively.

6.4.2 Hyperparameter optimization

Using the algorithm shown in Figure 6.1, the hyperparameters for GPR, SVMs, RFR, and ANNs are optimized. Although the optimal hyperparameters may vary for different datasets, the values of good hyperparameters would be close for different design datasets with similar sizes and design variable dimensions (Brochu et al., 2010, Yogatama and Mann, 2014). In other words, the optimal hyperparameter values obtained from this study can be applied to other similar design problems without significant change or can be used as the basis for optimization (Bardenet et al., 2013).

In each algorithm, the hyperparameters should be specified in advance. For GPR, the only hyperparameters are the kernel parameters, that is, the bold parameters in Eq. 6.3.

$$\textbf{\textit{Kernels}}: \begin{cases} \text{RBF: } k(x, x') = \exp(-\boldsymbol{\sigma}\|x - x'\|^2) \\ \text{Polynomial: } k(x, x') = (\boldsymbol{scale} <x, x'> + \boldsymbol{offset})^{\boldsymbol{deg}} \\ \text{Hyperbolic tangent (tanh): } k(x, x') = \tanh(\boldsymbol{scale} <x, x'> + \boldsymbol{offset}) \\ \text{Laplacian: } k(x, x') = \exp(-\boldsymbol{\sigma}\|x - x'\|) \end{cases}, \tag{6.3}$$

where $<x, x'>$ and $\|x - x'\|$ are the dot product and Euclidean distance of vectors x and x', ***scale*** and ***offset*** are used to scale the result of $<x, x'>$ and add an offset, ***deg*** defines the degree of the polynomial kernel, and σ is a parameter to account for the weight of nodes with different distances to the current node.

For an SVM, as a kernel-based algorithm, in addition to these kernel parameters, the ε bounds and the error penalization weighting factor C are also hyperparameters. For RFR, the number of trees in the random forest is a critical factor for regression accuracy, which also influences the computational cost: that is, more decision trees equals a higher

computational cost. Meanwhile, the decision-tree-related hyperparameters are also tuned: the number of randomly selected features for each split (NF), the minimum terminal node size (Min TS) and the maximum numbers of terminal nodes (Max TN).

For ANNs, considering the current structural complexity and dataset size, one hidden layer will be sufficient. Some critical hyperparameters related to the ANN structure are selected for optimization: the number of hidden-layer neurons, activation functions, optimizers, mini-batch sizes and learning speed parameters, that is, learning rate and momentum, where the momentum is only applicable to Stochastic gradient descent (**sgd**) optimizer.

RMSE and MXAE are taken as the objectives of the multi-objective hyperparameter optimization process, which has been defined in Eqs. 6.1 and 6.2. For each of the 16 tuning models (4 MLAs × 4 examples), 30 initial designs are generated to construct the initial surrogate model. Another 70 evaluations are used to optimize the hyperparameters iteratively and assess the results, update the surrogate model and reach the Pareto front.

The 5-fold cross-validation is used to train the model, which divides the training dataset into five subsets to train the model in iterations. After the hyperparameter optimization, one hyperparameter group is selected randomly from the Pareto front for each model and summarized in Table 6.3 with corresponding mean loss values of five cross-validations.

Table 6.3

Selected optimal hyperparameters for ML models with respect to the four structural datasets

Models		Hyperparameters				Loss	
		Kernels	Degree	Scale	Offset	RMSE	MXAE
GPR	TbPT	polynomial	3	7.22	-2.24	0.060	0.379
	TqA	polynomial	2	1.73	1.07	0.052	0.459
	ShB	polynomial	3	2.30	1.08	0.041	0.234
	OMcT	polynomial	3	7.67	9.30	0.055	0.219

		C	ε	Kernels	σ	Degree	Scale	Offset	RMSE	MXAE
SVM	TbPT	0.81	0.07	polynomial	NA	7	2.73	5.87	0.063	0.378
	TqA	2.81	0.41	polynomial	NA	1	4.03	-2.09	0.055	0.495
	ShB	9.77	0.13	Laplacian	0.37	NA	NA	NA	0.047	0.324
	OMcT	9.20	0.05	polynomial	NA	2	9.25	2.96	0.073	0.258

		Trees	NF	Min TS	Max TN	RMSE	MXAE
RFR	TbPT	773	10	1	304	0.046	0.417
	TqA	569	9	1	682	0.095	0.441
	ShB	718	7	1	1,000	0.045	0.298
	OMcT	879	9	1	493	0.079	0.377

		Hidden neurons	Activ-ation	Optimizer	Batch size	Learning rate	Momentum	RMSE	MXAE
ANN	TbPT	26	relu	sgd	199	0.77	0.83	0.047	0.295
	TqA	19	relu	adagrad	108	0.57	NA	0.051	0.450
	ShB	8	tanh	adagrad	85	0.30	NA	0.041	0.257
	OMcT	36	tanh	sgd	98	0.97	0.92	0.042	0.159

Note: relu: rectified linear unit; tanh: hyperbolic tangent; adagrad: Adaptive subgradient;

Using the optimal hyperparameters in Table 6.3, 16 models are trained. The RMSE and MXAE values of 5-fold cross-validation are presented in Figures 6.4 and 6.5, respectively, as boxplots. The corresponding training time for a single cross-validation process is also presented in Figure 6.6 with the standard error among 5-fold cross-validation.

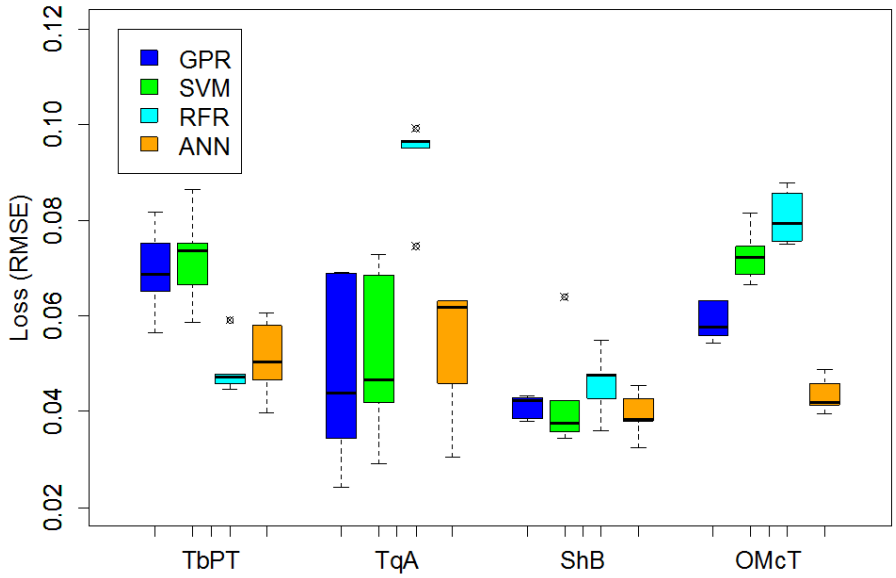


Figure 6.4. Comparison of RMSE accuracy of four regression methods trained by the four structural datasets using the hyperparameters in Table 6.3, presented in the box plots using the RMSE data from the test loss of 5-fold cross-validation.

From Table 6.3, for GPR, the third-degree polynomial kernel is recommended. It also performs well for the SVM. Although more regression trees in RFR could increase its accuracy, it is preferred to set the number of trees equal to one-half to three-quarters of dataset size. The minimum terminal node size should be as small as possible for a small dataset, due to the low computational cost as shown in Figure 6.6. Also, all design variables should be ready for each node split if the input feature dimension is not high.

After 2000 epochs of training, the ANN shows higher performance if the **adagrad** or **sgd** is used as the optimizer with the *tanh* or *relu* activation function. It is better to keep the total number of weights less than one-quarter of the training dataset size. Additionally, dividing the whole dataset into 4 to 10 mini-batches, that is, a mini-batch size of 80 to 200 in this study, is a good practice to obtain a satisfactory trade-off between accuracy and training time. Meanwhile, a learning rate between 0.1 and 0.9 is recommended, which is important to increase training efficiency.

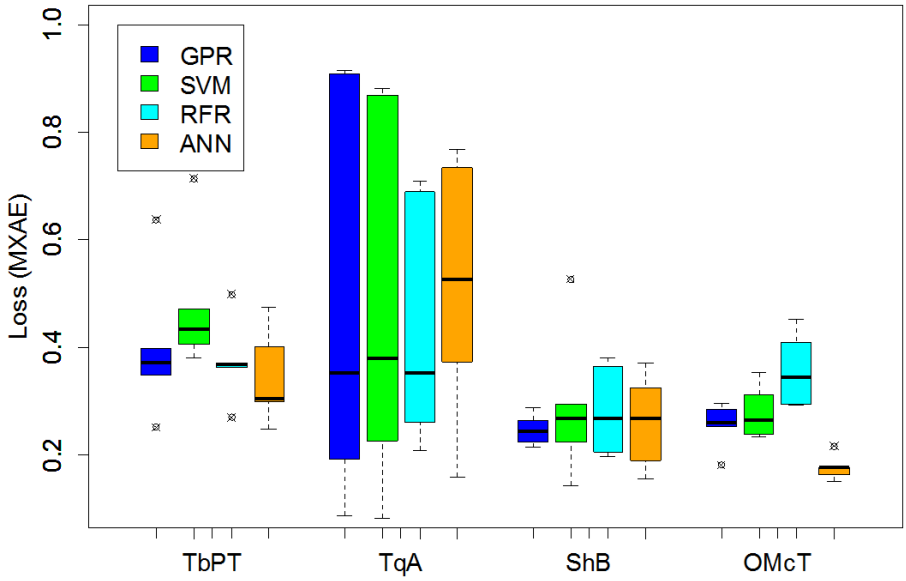


Figure 6.5. Comparison of MXAE accuracy of four regression methods trained by the four structural datasets using the hyperparameters in Table 6.3.

As shown in Figs. 6.4 and 6.5, the ANN exhibits the best performance, indicated by its relatively low RMSE and MXAE, and is followed by GPR. However, considering the high computational cost of ANN training (approximately 70 times that of GPR, on

average) shown in Figure 6.6, GPR would be more suitable under limited computational resources and a large dataset size.

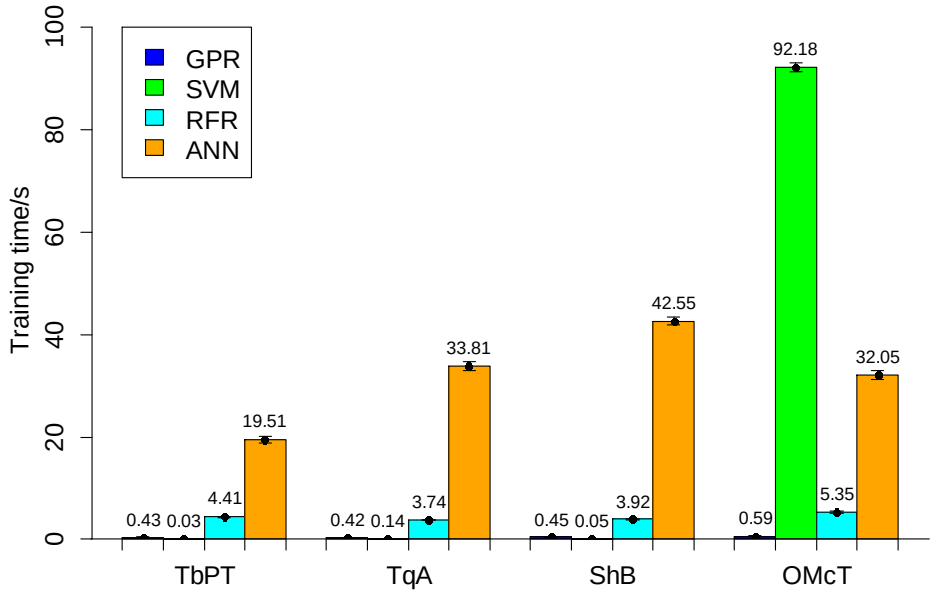


Figure 6.6. Comparison of the four regression methods in terms of their training time based on the hyperparameters in Table 6.3 (computational power: Dell Precision Tower 5810 with Intel Xeon CPU E5-2690 v3: 2.6 GHz turbo up to 3.5 GHz and 32 GB RAM).

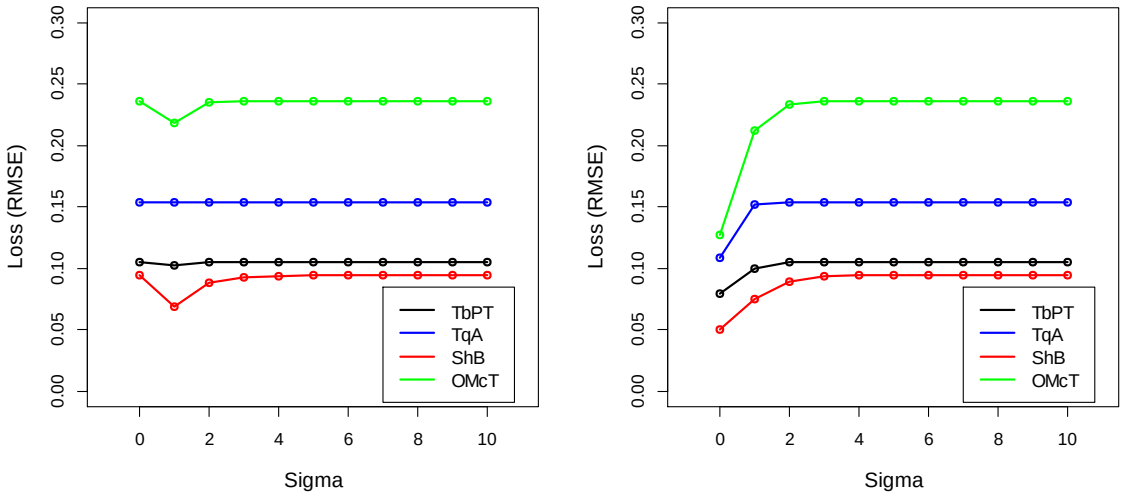
In Figure 6.6, the SVM training time varied greatly with the change of hyperparameters in Table 6.3. For the SVM trained by the OMcT dataset, the large error penalization $\boldsymbol{C}$ (i.e. 9.2) means that the error term in Eq. A-8 takes a high weight. Meanwhile, the low ε (i.e. 0.05) in Table 6.3 suggests more samples may have fallen out of ε-bounds and more error items were added into the error term in Eq. A-8, which increased the objective function complexity, the number of $\boldsymbol{C}$. These facts cause the high computational cost of training with the OMcT dataset. Hence, a smaller $\boldsymbol{C}$ and larger ε could reduce the computational cost.

In this section, the two MLAs demonstrating better performance, GPR and ANN, were further analyzed. The effects of their hyperparameters were also discussed. In this part, RMSE was the only measure used to quantify modeling accuracy.

6.4.3 Hyperparameter effects on GPR

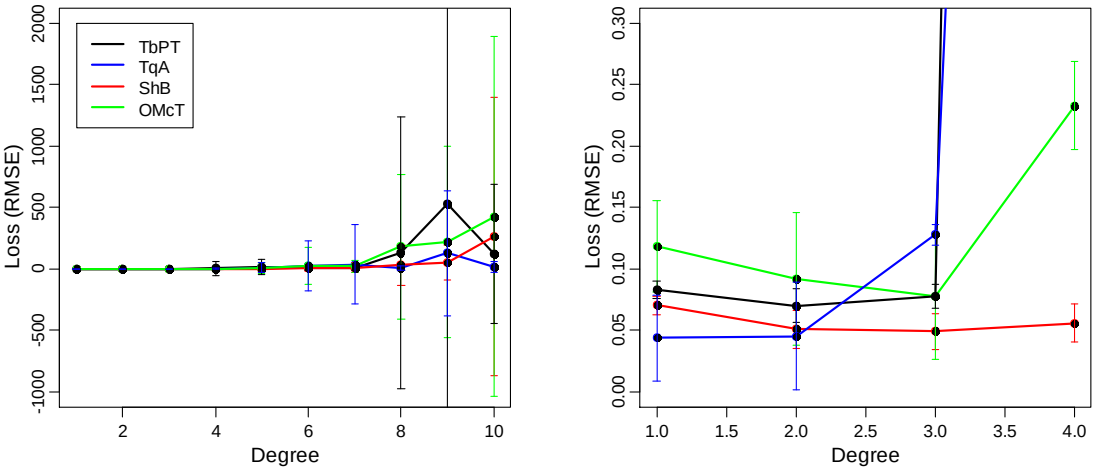
The only hyperparameters in GPR are kernel-related parameters. Four kernel functions, the RBF, polynomial, tanh and Laplacian, as listed in Eq. 6.3, are used. The GPR performances with different *Sigma* (σ) values are plotted in Figure 6.7(a) and (b) for the Laplacian and RBF kernels, respectively. The low-level σ (< 2) has a critical influence. The best GPR performance σ values are 1 and 0 for the RBF and Laplacian kernels, respectively.

Regarding the polynomial kernel, different *degrees* significantly influence the model accuracy. Figure 6.8(a) illustrates the standard error bars of loss to account for the effect of *scale* and *offset* variation. Table 6.3 shows that the optimal degrees would be 2 or 3, so low-level degrees in Figure 6.8(a) are amplified in Figure 6.8(b) to reveal the details of their effect. The results also match well with the data in Table 6.3, and the second- or third-order polynomial would be good choices. The influence of *scale* and *offset* increase with higher degrees, which is indicated by the larger standard errors.



(a) (b)

Figure 6.7. Effects of kernel hyperparameter sigma on the GPR performance for (a) RBF dot and (b) Laplacian kernels.



(a) (b)

Figure 6.8. Effect of the polynomial kernel degree on the GPR performance with the error to account for the influence of the scale and offset: (a) the full range of degrees and (b) the enlarged view when the degree is in the range (1~4).

Furthermore, for *scale* and *offset*, their absolute values should be greater than 1, since this will significantly reduce the RMSE error, as shown in Figure 6.9. Absolute values greater than 1 for *scale* and *offset* should generally work well for a high-accuracy GPR, although their optimal RMSE values vary across the four cases.

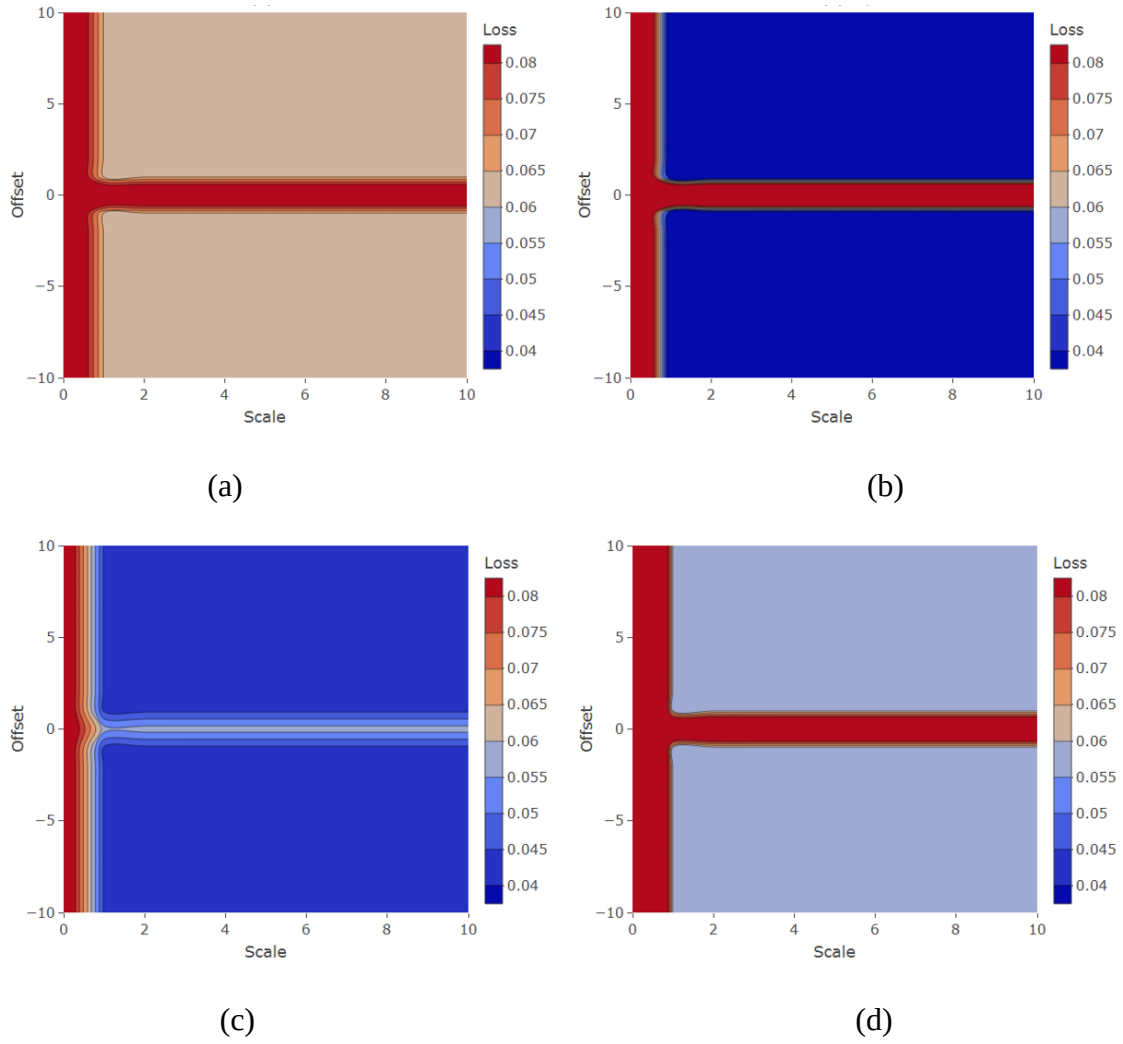


Figure 6.9. GPR performance with different scale and offset for polynomial kernel functions under degrees shown in Figure 6.8(b): (a) ShB with degree 3, (b) OMcT with degree 3, (c) TbPT with degree 2 and (d) TqA with degree 2.

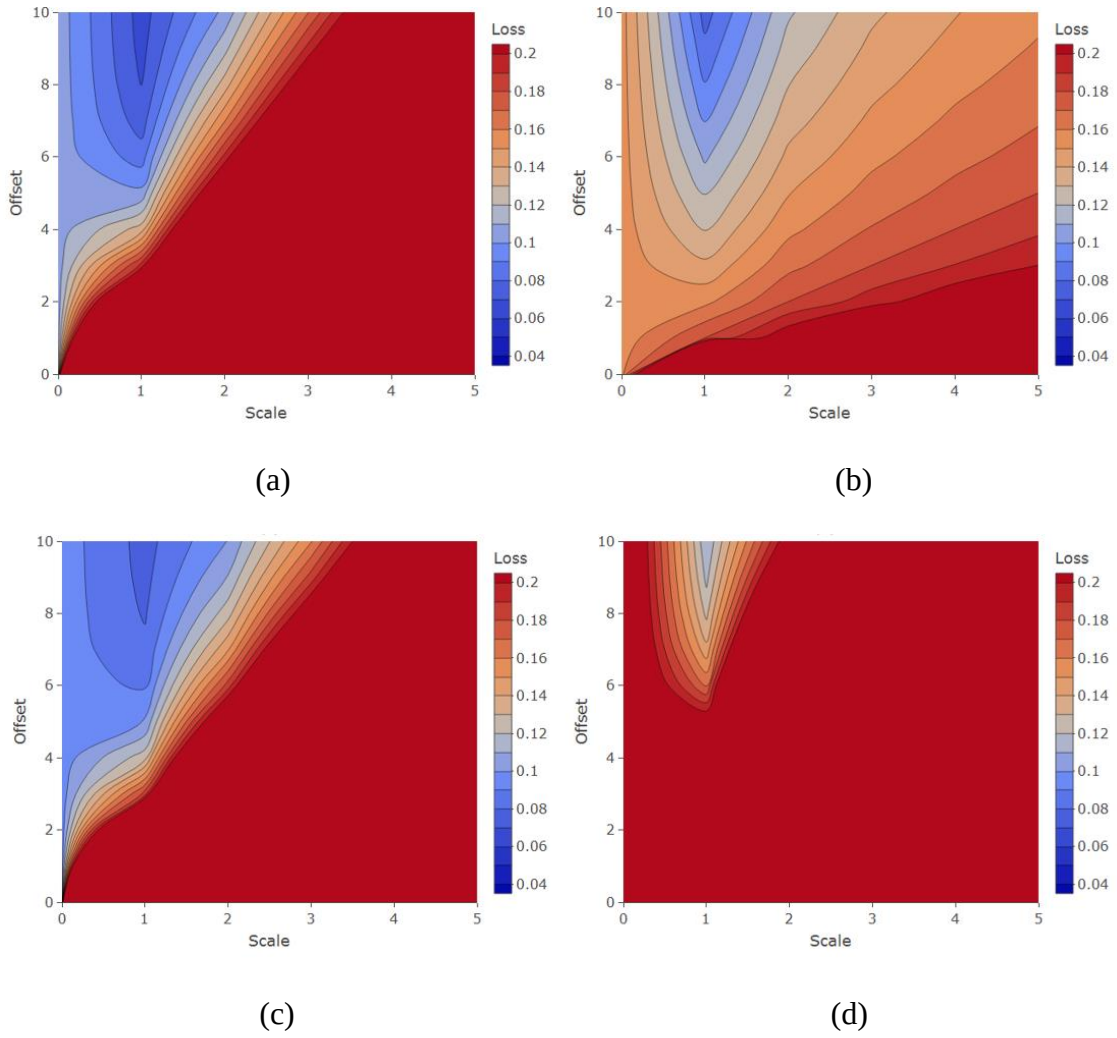
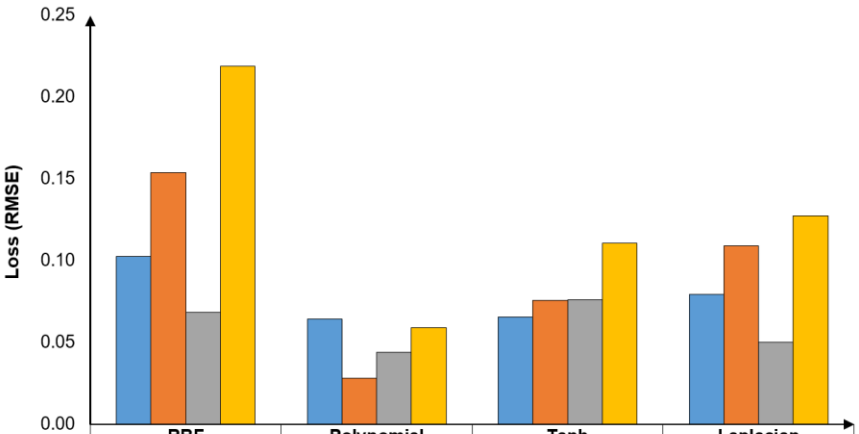


Figure 6.10. GPR performance with the change of the tanh kernel function's scale and offset for (a) ShB, (b) OMcT, (c) TbPT and (d) TqA.

The *tanh* kernel function also has only two hyperparameters: *scale* and *offset*. Figure 6.10 shows the performance changes of GPR with different values for them, indicating a similar trend in the four cases. For a high-accuracy GPR model, recommended *scale* and *offset* values are 1 and 10, respectively. Likewise, large *scale* and low *offset* values should be avoided to prevent the deterioration of model accuracy.

In summary, the RMSE accuracy of 16 models (4 kernels × 4 structural cases) trained with the best kernel hyperparameter values is presented in Figure 6.11. Clearly, the polynomial kernel shows the best performance in terms of GPR accuracy, while the RBF kernel demonstrates a large scatter and unstable performance.



	RBF		Polynomial				Tanh			Laplacian	
	Sigma	RMSE	Degree	Scale	Offset	RMSE	Scale	Offset	RMSE	Sigma	RMSE
TbPT	1	0.103	2	1	1	0.064	1	10	0.066	0	0.079
TqA	1	0.154	2	1	-1	0.028	1	10	0.076	0	0.109
ShB	1	0.069	3	1	1	0.044	1	10	0.076	0	0.050
OMcT	1	0.219	3	1	1	0.059	1	10	0.111	0	0.128

Figure 6.11. Comparison of the modeling performances of the four kernel functions in the GPR algorithm.

6.5 Summary

In this section, four MLAs were selected to study their ability to predict structural responses through modeling of four typical engineering structures. These MLAs can be used to build surrogate models to supplement or partially replace expensive numerical modeling and simulations. Using a multi-objective hyperparameter optimization framework, the hyperparameters of these four MLAs were optimized and their optimal performances compared. The results show that GPR and ANN outperformed the other

two MLAs. In addition, the effects of hyperparameters on the GPR results were discussed in detail. A summary follows.

1) The multi-objective hyperparameter optimization algorithm could tune the hyperparameters effectively and efficiently and produce a highly accurate surrogate model within a limited number of evaluations (100 in this study).
2) The result shows that in the current structural design problems, GPR would be the best choice to train a high-accuracy model with low computational cost, while the ANN has more potential to be improved by additional training data and iterations.
3) The polynomial kernel is preferred for GPR with degree 2 or 3; its scale and absolute offset value should be equal to or greater than 1.

The GPR algorithm with corresponding tuned hyperparameters is used in Chapter 7 to build the surrogate model for uncertainty analysis.

Chapter 7 Data mining method (DMM) with uncertainty

7.1 Introduction

In the previous chapters, the decision tree algorithms were created at various levels by learning from deterministic datasets. However, they have no ability to learn the decision-making rules from an uncertain dataset. In vehicle crashworthiness design, uncertainty is unavoidable due to the design and manufacturing process as well as the working conditions (Fang et al., 2014, Eichmueller and Meywerk, 2019), such as thickness variations in the sheet metal due to manufacturing (Zimmermann et al., 2017, Hesse et al., 2017), mechanical property changes resulting from heat treatment (Li et al., 2017) and random numerical errors in the simulations. These factors may cause uncertainty in both design variables and responses. A new algorithm for decision tree development from an uncertain dataset is needed.

Several MLAs have been proposed to handle uncertain datasets (Sun et al., 2014, Angiulli and Fassetti, 2013, Tavakkol et al., 2017). Efforts have also been made to extend regular decision trees to learn from uncertain data (Meenakshi and Venkatachalam, 2015, Tsang et al., 2011, Qin et al., 2010). Although the performance of these algorithms has been verified by some public datasets, they cannot generate consistent results. In addition, no such design trees have been reported in engineering design applications. To resolve these issues, a new decision tree for uncertain datasets (DTUD) is developed and presented in this chapter. Section 7.2 introduces its basic theory and implementation. The new algorithm is then applied in Section 7.3 to design an S-beam. Additional analysis of its performance is presented in Section 7.4.

7.2 Methods

To represent uncertainty in this study, the input features or design variables were assumed to have an interval with a certain distribution.

7.2.1 A new decision tree for uncertain datasets (DTUD)

In traditional decision tree construction, a key step is the selection of a split point. Typically, all attribute values are taken as candidates for split points. In the current dataset of a non-leaf node, the selected split point should result in a maximum or minimum value for splitting criteria, for example, information gain in ID3, gain ratio in C4.5 or Gini index in CART (Han et al., 2011). However, the interval with uncertain data is not an exact value but a range with no available exact values as candidates for split points. To generate some exact values, N points are placed evenly in each interval. If a specific value (S_j) of the jth feature is selected as the split point, it may have three possible conditions with respect to its ith interval ([A_{ij}^{CL} A_{ij}^{CU}]).

Prior to an in-depth discussion of the three conditions, two basic definitions should be introduced. **Tuple probability (TP)** is the probability of a tuple belonging to a specific partition or, finally, a leaf node. In this way, the summation of the tuple probabilities of an observation is equal to one. The **label probability (LP)** is the probability of a specific label in a dataset. Demonstrably, the summation of LPs for all labels in a dataset is also equal to one.

1) If the split point falls to the left side of interval ($S_j \leq A_{ij}^{CL}$), this tuple is assigned to the right branch with TP p_R^t calculated by Eq.7.1 and with the left branch TP $p_L^t = 0$.

$$p_R^t = \int_{A_{i1}^{CL}}^{A_{i1}^{CU}} \cdots \int_{\boldsymbol{A_{ij}^{CL}}}^{\boldsymbol{A_{ij}^{CU}}} \cdots \int_{A_{in}^{CL}}^{A_{in}^{CU}} f(\vec{x}) d\vec{x} \ , \qquad (7.1)$$

where $\vec{x} = (x_1, x_2, \cdots, x_j, \cdots, x_n)$ represents the vector of the input feature, and $f(\vec{x})$ is the joint probability of input features. $\boldsymbol{A_{ij}^{CL}}$ and $\boldsymbol{A_{ij}^{CU}}$ are the lower and upper limits of the ***i***th tuple's ***j***th feature interval in the current dataset, respectively.

2) If the split point falls within the interval ($A_{ij}^{CL} < \boldsymbol{S_j} \leq \boldsymbol{A_{ij}^{CU}}$), this tuple is partitioned into two parts. The left half ($\leq \boldsymbol{S_j}$) is assigned to the left branch with TP calculated in Eq. 7.2.

$$p_L^t = \int_{A_{i1}^{CL}}^{A_{i1}^{CU}} \cdots \int_{\boldsymbol{A_{ij}^{CL}}}^{\boldsymbol{S_j}} \cdots \int_{A_{in}^{CL}}^{A_{in}^{CU}} f(\vec{x}) d\vec{x} \qquad (7.2)$$

The right half ($> \boldsymbol{S_j}$) is included in the right branch with TP determined in Eq. 7.3.

$$p_R^t = \int_{A_{i1}^{CL}}^{A_{i1}^{CU}} \cdots \int_{\boldsymbol{S_j}}^{\boldsymbol{A_{ij}^{CU}}} \cdots \int_{A_{in}^{CL}}^{A_{in}^{CU}} f(\vec{x}) d\vec{x} \qquad (7.3)$$

3) If the split point falls to the right side of the interval ($\boldsymbol{A_{ij}^{CU}} \leq \boldsymbol{S_j}$), this tuple is assigned to the left branch with TP p_L^t expressed in Eq. 7.4 and with the right branch TP $p_R^t = 0$.

$$p_L^t = \int_{A_{i1}^{CL}}^{A_{i1}^{CU}} \cdots \int_{\boldsymbol{A_{ij}^{CL}}}^{\boldsymbol{A_{ij}^{CU}}} \cdots \int_{A_{in}^{CL}}^{A_{in}^{CU}} f(\vec{x}) d\vec{x} \qquad (7.4)$$

If all input features are independent of each other, the joint distribution $f(\vec{x})$ can be represented by the multiplication of all individual distributions. Thus, an observation can be partitioned into several leaf nodes. It is counted by its TP in each leaf. Accordingly, two basic parameters used in the traditional decision tree for calculating attribute

selection measures can be adjusted in this new algorithm. The LP (P^{Lt}) that a tuple belongs to label L_t in the current dataset can be calculated by Eq. 7.5.

$$P^{Lt} = \frac{\sum_{i=1}^{n_{Lt}} p_i^{t_{Lt}}}{\sum_{i=1}^{n_c} p_i^t}, \tag{7.5}$$

where $p_i^{t_{Lt}}$ denotes the TP of the ***i***th tuple (with L_t label), and n_{Lt} is the number of tuples with label L_t in the current dataset; n_c and p_i^t are the total number of tuples in the current dataset and their TP, respectively. The P^{Lt} is related only to the label content of the current dataset. Another parameter is the weight of each partition with respect to a specific split point, which, traditionally, is estimated by the portion of tuples in each partition. In a DTUD, it can be expressed by Eq. 7.6 for each of the two resulting partitions.

$$\frac{|D_{sp}|}{|D|} = \frac{\sum_{i=1}^{n_{sp}} p_i^{t_{sp}}}{\sum_{i=1}^{n_c} p_i^t} \tag{7.6}$$

Unlike in Eq. (7.5), the numerator here is changed to the ratio of a partition's total TP to the TP summation in the current dataset; n_{sp} and $p_i^{t_{sp}}$ are the numbers of tuples in the ***sp***th partition and the ***i***th TP, respectively. Based on these definitions, the attribute selection measures can be calculated to grow a decision tree in a recursive way.

To evaluate the accuracy of a decision tree in training, the classification accuracy (*ACC*) is calculated by dividing the TP summation of correctly classified tuples by the total dataset size, as in Eq. (7.7). Following the partition rules defined, the content in each leaf node is the TP summation of all tuples for each label but not the exact number of observations. Furthermore, LP in a leaf node can be calculated by Eq. 7.5. If an exact

label should be assigned for a leaf node to define the correct classification, the label with the maximum probability can be designated.

$$ACC = \frac{\sum_{i=1}^{n_l} P_i^c}{n_m}, \quad (7.7)$$

where n_m is the size of training dataset, and n_l and P_i^c are the number of leaf nodes and the TP summation of correctly classified tuples in the *i*th leaf node, respectively.

A method similar to the training method is followed to evaluate the validation ACC. An uncertain observation may be assigned to multiple leaf nodes with corresponding TP P_s^t (s = 1, 2, ⋯, n_l) calculated by Eqs. 7.1–7.4. The label predicted by a tree can be determined as the label with the maximum value of the summation of P_s-weighted leaf LPs, as expressed in Eq. 7.8.

$$Max \quad (\overrightarrow{p_L} = P_s^t \cdot \sum_{i=1}^{n_l} \overrightarrow{p_i^L} \quad), \quad (7.8)$$

where $\overrightarrow{p_i^L}$ and $\overrightarrow{p_L}$ are the vectors of LP of the ***i***th leaf node and the calculated P_s^t-weighted LP summation, respectively.

7.2.2 Implementation of the new algorithm

Before the DTUD training, the joint probability distribution of design variables should be determined. Due to the influence of random events (e.g. vibration, measurement and deformation) in engineering design and manufacturing, we use the Gaussian distribution as the default one if there is no prior knowledge of the interval distribution (e.g. $[A_{ij}^{CL}, A_{ij}^{CU}]$), where $A_{ij}^m = \frac{A_{ij}^{CL} + A_{ij}^{CU}}{2}$ is defined as the mean. The 3-sigma rule, which could

cover most probability distributions, is adopted over the whole interval range with the probability density function defined in Eq. 7.9.

$$f(A_{ij}) = \mathrm{C} \cdot N(A_{ij}^{m}, (\frac{(A_{ij}^{CU} - A_{ij}^{CL})/2}{3})^2) \,, \qquad (7.9)$$

where N is the normal distribution, and C = 1/0.997 is a factor to adjust the probability within an interval to 1. To fully present the validation process, we used the dataset for the S-beam from Chapter 4 as an example. A part of the dataset is shown in Figure 7.1. To generate an uncertain dataset, each original value was used as the mean ($A_{ij} = A_{ij}^{m}$), and a ratio ($\boldsymbol{R}$ = 0.1 in this example) of the mean was taken as the deviation ($\boldsymbol{R} \cdot A_{ij} = (A_{ij}^{CU} - A_{ij}^{CL})/2$). Due to their independent nature, the joint distribution of seven design variables can be seen as the multiplication of individual Gaussian distributions.

Deterministic dataset

DR	H	CL	AN	T	UR	UL	L
200	180	68	22	2.3	200	250	g
450	180	60	34	3.0	250	225	g
550	190	58	30	1.5	200	300	p
350	220	56	36	2.5	200	250	m
550	270	70	34	1.8	150	225	g
200	280	60	40	2.0	250	150	p
200	210	64	38	2.3	250	400	p
200	240	52	26	3.0	200	200	g
350	230	62	28	2.0	300	350	p
250	200	50	36	1.8	300	275	p

Interval uncertain dataset

DR	H	CL	AN	T	UR	UL	L
[180 220]	[162 198]	[61 75]	[20 24]	[2.0 2.5]	[180 220]	[225 275]	g
[405 495]	[162 198]	[54 66]	[31 38]	[2.7 3.3]	[225 275]	[203 248]	g
[495 605]	[171 209]	[52 64]	[27 33]	[1.4 1.7]	[180 220]	[270 330]	p
[315 385]	[198 242]	[50 62]	[33 40]	[2.3 2.8]	[180 220]	[225 275]	m
[495 605]	[243 297]	[63 77]	[31 38]	[1.6 1.9]	[135 165]	[203 248]	g
[180 220]	[252 308]	[54 66]	[36 44]	[1.8 2.2]	[225 275]	[135 165]	p
[180 220]	[189 231]	[58 70]	[34 42]	[2.0 2.5]	[225 275]	[360 440]	p
[180 220]	[216 264]	[47 57]	[23 29]	[2.7 3.3]	[180 220]	[180 220]	g
[315 385]	[207 253]	[56 68]	[25 31]	[1.8 2.2]	[270 330]	[315 385]	p
[225 275]	[180 220]	[45 55]	[33 40]	[1.6 1.9]	[270 330]	[248 303]	p

Figure 7.1. Transforming the deterministic dataset to an uncertain dataset for the S-beam design described in Chapter 4.

Using the generated uncertain dataset, a decision tree with uncertainty was built using the algorithm introduced in Section 7.2, as shown in Figure 7.2. The first design in Figure 7.1 was selected as an example to show the validation or prediction process. Normal distribution was defined for each interval feature of this tuple. By following the DTUD, this tuple can be assigned to branches b1, b2, b3 and b6. Each non-leaf node partitions a current interval feature into two parts. The final TPs of this tuple in each of four leaves were calculated by Eq. (7.10) as

$$\begin{cases} P_{b1} = \mathrm{C} \cdot \int_{2.4}^{2.5} \mathcal{N}\left(2.25, \left(\frac{0.25}{3}\right)^2\right) \mathrm{d}T = 0.035 \\ P_{b2} = \mathrm{C} \cdot \int_{2.36}^{2.4} \mathcal{N}\left(2.25, \left(\frac{0.25}{3}\right)^2\right) \mathrm{d}T = 0.057 \\ P_{b3} = \mathrm{C}^2 \cdot \int_{2.0}^{2.36} \mathcal{N}\left(2.25, \left(\frac{0.25}{3}\right)^2\right) \mathrm{d}T \cdot \int_{180}^{215.5} \mathcal{N}\left(200, \left(\frac{20}{3}\right)^2\right) \mathrm{d}DR = 0.9 \\ P_{b6} = \mathrm{C}^2 \cdot \int_{2.0}^{2.36} \mathcal{N}\left(2.25, \left(\frac{0.25}{3}\right)^2\right) \mathrm{d}T \cdot \int_{215.5}^{220} \mathcal{N}\left(200, \left(\frac{20}{3}\right)^2\right) \mathrm{d}DR = 0.008 \end{cases} . \quad (7.10)$$

The predicted LP of this tuple is then

$$\begin{Bmatrix} P_g \\ P_m \\ P_p \end{Bmatrix} = 0.035 * p_{b1}^L + 0.057 * p_{b2}^L + 0.9 * p_{b3}^L + 0008 * p_{b6}^L = \begin{cases} 0.263 \\ 0.724 \\ 0.013 \end{cases}, \quad (7.11)$$

where p_{b1}^L, p_{b2}^L, p_{b3}^L and p_{b6}^L are the LPs of branches b1, b2, b3 and b6, respectively.

Although this aforementioned design alternative is a “g” design in the original dataset, it is labeled as an “m” design by the uncertain design rules, which is the same prediction result achieved with the deterministic decision tree in Chapter 4.

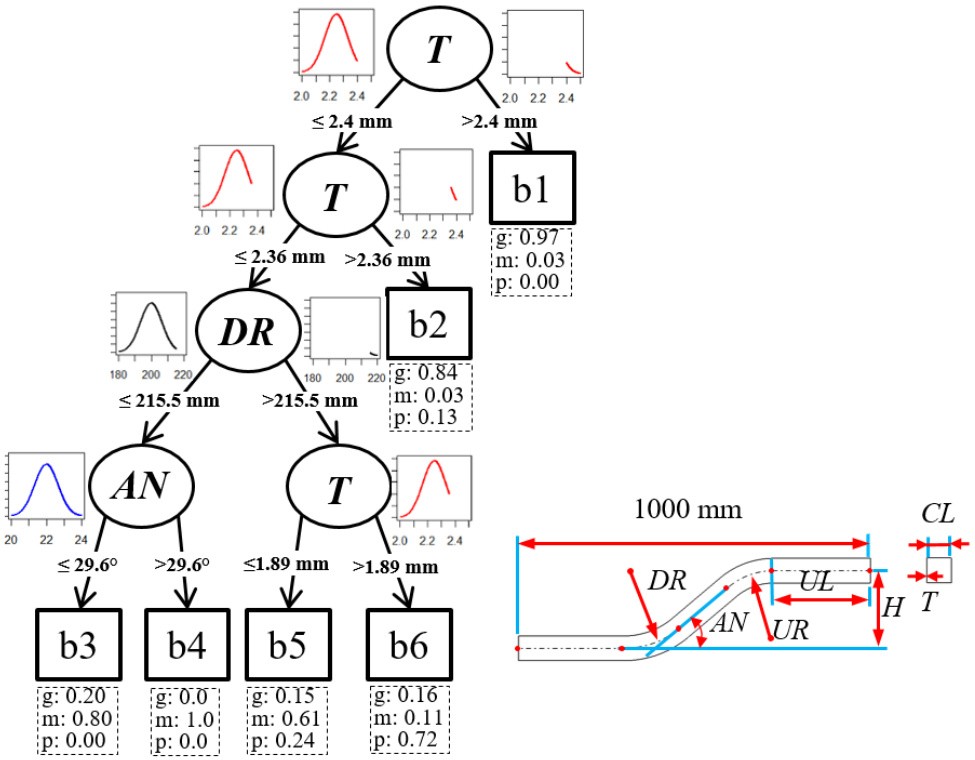


Figure 7.2. A decision tree with uncertainty for the S-beam described in Chapter 4.

7.3 Design of the S-beam by DTUD

Based on these analyses, the DTUD was verified by some datasets from the University of California, Irvine (UCI) public repository (https://archive.ics.uci.edu/ml/datasets.php) for MLAs. The details are included in Appendix B. In this section, the process of designing an S-beam using the DTUD is described in detail.

7.3.1 S-beam parameterization

The S-beam was designed based on the dimension-based approach but parameterized in a way different from that used in Chapter 4 to facilitate the uncertain design. Figure 7.3 shows the strategy. **W** and **H** represent the width and height of the bounding box,

respectively. The total length of the S-beam is fixed as 1,000 mm with three segments (**S1**, **S2** and **S3)**, and their corresponding thicknesses are $\mathbf{T_1}$, $\mathbf{T_2}$ and $\mathbf{T_3}$, respectively. $\mathbf{T_2}$ is assumed to be the mean of $\mathbf{T_1}$ and $\mathbf{T_3}$ to represent a gradual change. $\mathbf{L_1}$ and $\mathbf{L_2}$ are the lengths of **S1**and **S2,** respectively. In Figure 7.3, the heights of four cross-sections ($\mathbf{H_1}$, $\mathbf{H_2}$, $\mathbf{H_3}$ and $\mathbf{H_4}$) are also defined as design variables, since the ratio of width and height can influence the structural response significantly. In S1 and S3, the height of any cross-section is assumed as a linear interpolation between the heights of their cross-sections at the two ends.

Using these rules, an initial design was created with the design variable values as follows: $\mathbf{L_1}$ = 300 mm, $\mathbf{L_2}$ = 400 mm, $\mathbf{H_1}$ = 75 mm, $\mathbf{H_2}$ = 75 mm, $\mathbf{H_3}$ = 75 mm, $\mathbf{H_4}$ = 75 mm, $\mathbf{T_1}$ = 2.0 mm, $\mathbf{T_2}$ = 2.0 mm, $\mathbf{T_3}$ = 2.0 mm, **H** = 75 mm and **W** = 60 mm. The dimension constraints for these 11 design variables were assumed as 300 mm ≤ **L1** ≤ 500 mm; 200 mm ≤ $\mathbf{L_2}$ ≤ 400 mm; 50mm ≤ $\mathbf{H_1}$ ≤ 100 mm; 50 mm ≤ $\mathbf{H_2}$ ≤ 100 mm; 50 mm ≤ $\mathbf{H_3}$ ≤ 100 mm; 50 mm ≤ $\mathbf{H_4}$ ≤ 100 mm; 1 mm ≤ $\mathbf{T_1}$ ≤ 3 mm; 1 mm ≤ $\mathbf{T_3}$ ≤ 3 mm; 200 mm ≤ **H** ≤ 500 mm; 150 mm ≤ **W** ≤ 300 mm and $\mathbf{T_2}$ = (T1 + T3)/2. The geometric model of the S-beam was further converted to an FE model to simulate the frontal impact response. The loading and boundary conditions have been described in Chapters 4 and 5 and are not repeated here.

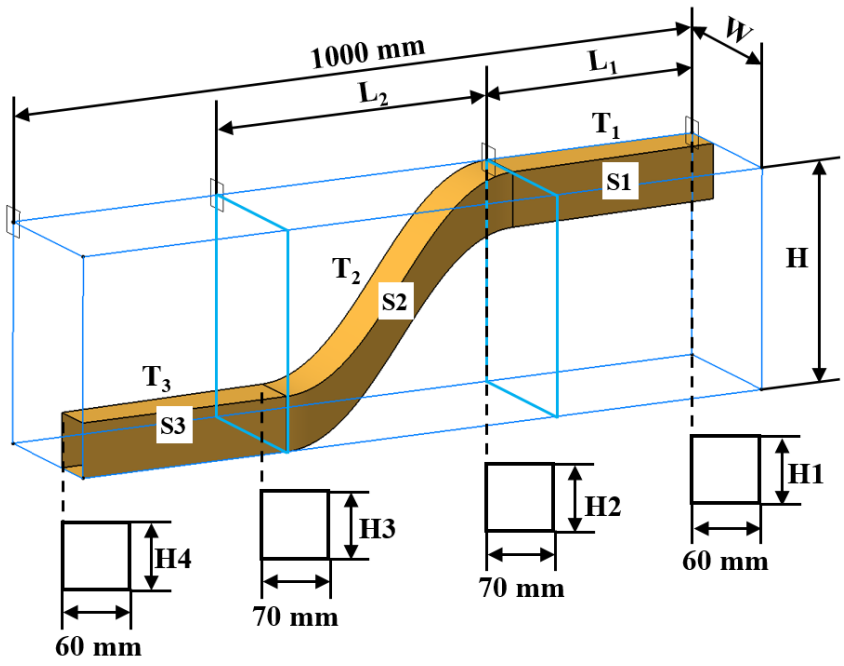


Figure 7.3. Parameterization of the S-beam for uncertainty analysis.

7.3.2 Generation of an uncertain design dataset

Similar to the design problem formulated in Chapter 4, specific energy (SEA) was used as the main response or objective in this chapter for uncertainty design. 3,000 design alternatives were generated by FE simulations. To reduce the computational cost, a GPR surrogate model was trained using 80% of the simulation data. The SEA values computed by FE models are compared with those from the GPR model predictions for validation in Figure 7.4. Figure 7.4 shows a good match, which verifies the GPR accuracy.

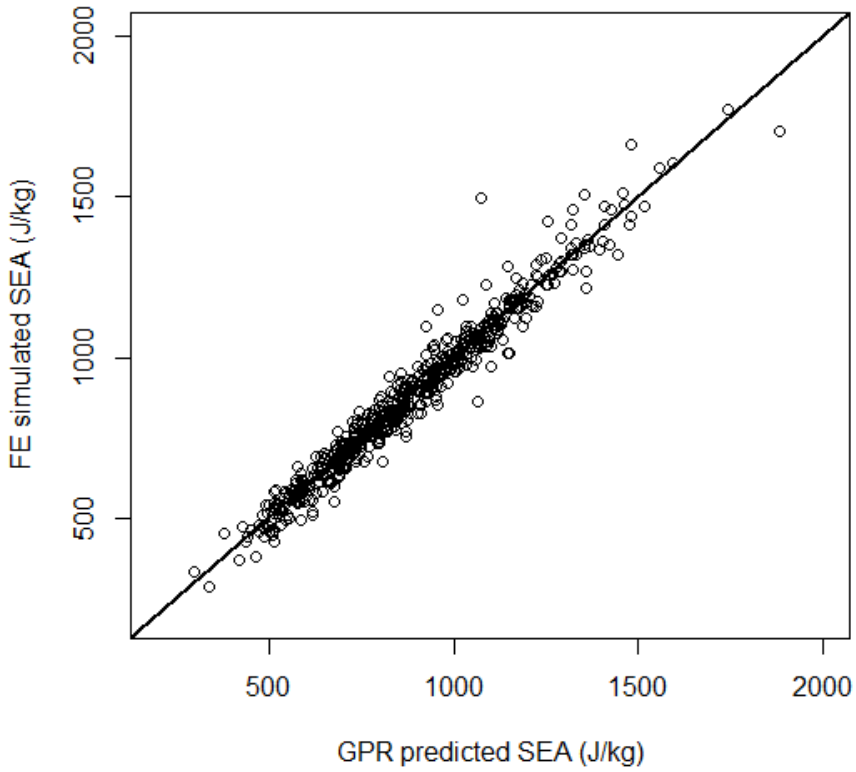


Figure 7.4. SEA values computed by FE models vs. those by GPR model predictions.

Using the trained GPR model, 1,000 additional design alternatives were generated. To include the uncertainty, Gaussian noise with a zero mean and 10% standard deviation of parameter value was added to the values of design variables. The Gaussian distribution was also defined over the whole interval of each design variable by applying the 3-sigma rule.

7.3.3 Uncertain dataset mining

The frequencies for SEA values of the 1,000 new designs created with GPR are plotted in Figure 7.5. The labeling criteria are defined as good (“g”): SEA > 1,200 J/kg; intermediate (“m”): 800 J/kg < SEA ≤ 1,200 J/kg and poor (“p”): SEA ≤ 800 J/kg.

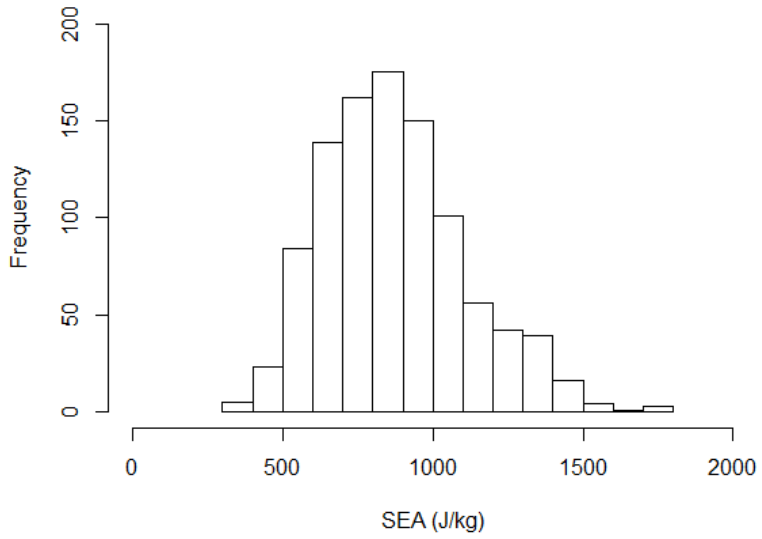


Figure 7.5. The distribution of SEA value frequencies of the 1,000 new designs created with GPR.

To train a DTUD, 80% of samples were used in the training process. To control the complexity of the generated tree, the number of layers was limited to six. Using the proposed algorithm, a decision tree was established and then validated with the remaining 20% of designs in the dataset.

7.3.4 Decision tree for the S-beam with uncertainty

Using the uncertain dataset for the S-beam, a decision tree was trained and validated, as shown in Figure 7.6. Training and validation classification accuracy were 0.77 and 0.73, respectively. The decision tree contains six layers and 23 leaf nodes. The content in each leaf node is the LP. If a design falls into this node, its performance should follow this probability density function (PDF). If a new design falls into multi-leaf nodes, its performance PDF can be calculated by following the strategies illustrated in Figure 7.2. In this way, it can be determined that the decision tree in Figure 7.6 contains five

branches for “g,” 10 branches for “m” and eight branches for “p.” The design rules for “g” can also be generated by following the paths from the root to the “g” leaf nodes; examples include **b12** and **b14**, due to their high LPs for “g.”

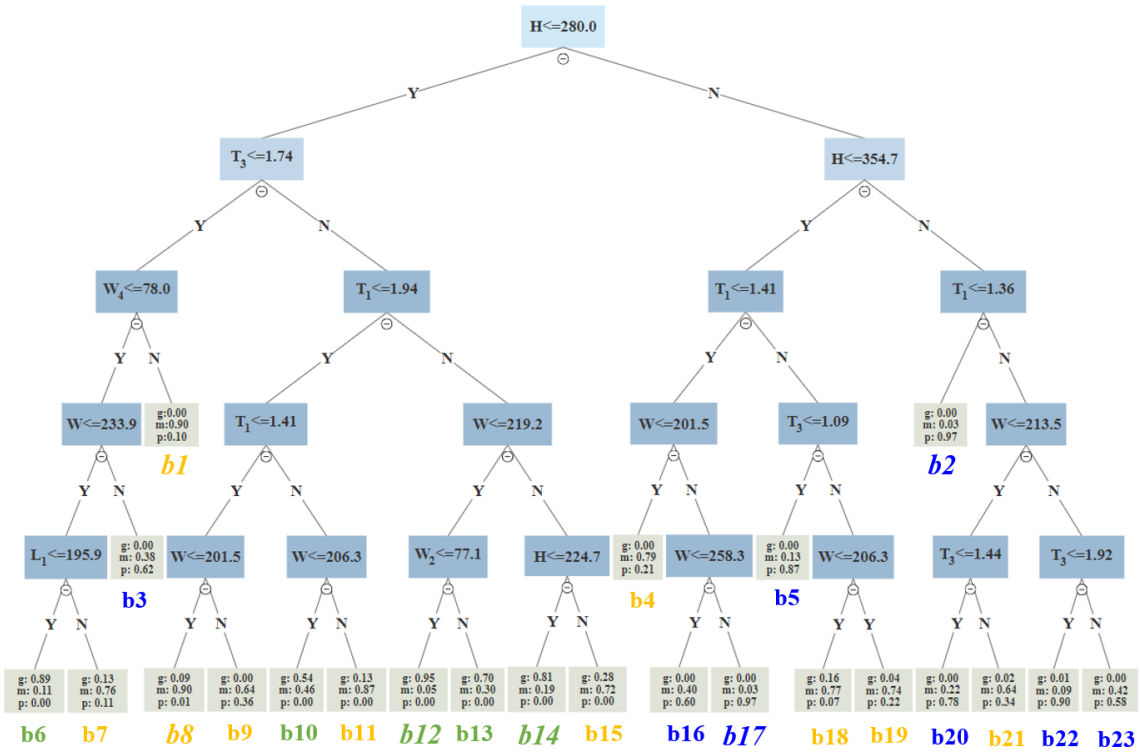


Figure 7.6. An uncertain decision tree generated using the S-beam uncertain dataset where six branches, that is, “g”: b12 and b14; “m”: b1 and b8; “p”: b2 and b17, are selected as the representatives of their labels (Y: Yes; N: No).

7.4 Discussion

In this section, additional analyses are performed to discuss (1) the ability of the DTUD to generate designs with expected performance and (2) the influence of uncertainty on the DTUD performance.

7.4.1 Ability of DTUD to generate designs with expected performance

To quantify the probability of generating designs with expected performance, two branches were selected as representatives for each class, as shown in Figure 7.7. A total of 50,000 designs were randomly created using the Monte Carlo method (MCM) in the subspace for each branch. The probability distribution of the GPR predicted response (measured by the SEA) exhibits a normal distribution, as shown in Figure 7.7 with the labeling bounds. The distribution parameters, for example, the mean and standard deviation, and the probability for each class are summarized in Table 7.1.

In Figure 7.7, the mean value of SEA distribution for each branch falls into its corresponding labeling value range, which indicates a high labeling probability or accuracy of the DTUD prediction. In Table 7.1, the b12 branch with the higher "g" LP (0.95) has a higher probability (0.90) of generating a good design than b14, which has a "g" LP of 0.81 and a "g" TP of 0.71. This suggests that the branch with higher LP could have a higher probability of generating expected performance designs.

Table 7.1

The normal distribution and label probability for each branch calculated using the Monte Carlo method (MCM)

Branches	Mean	*SD*	Probability		
			g	m	p
b12/g	1,404.6	161.7	**0.90**	0.10	0.00
b14/g	1,270.3	128.8	**0.71**	0.29	0.00
b1/m	980.1	163	0.09	**0.78**	0.13
b8/m	994.2	119.5	0.04	**0.91**	0.05
b2/p	582.2	101.4	0.00	0.02	**0.98**
b17/p	659.1	71.8	0.00	0.03	**0.98**

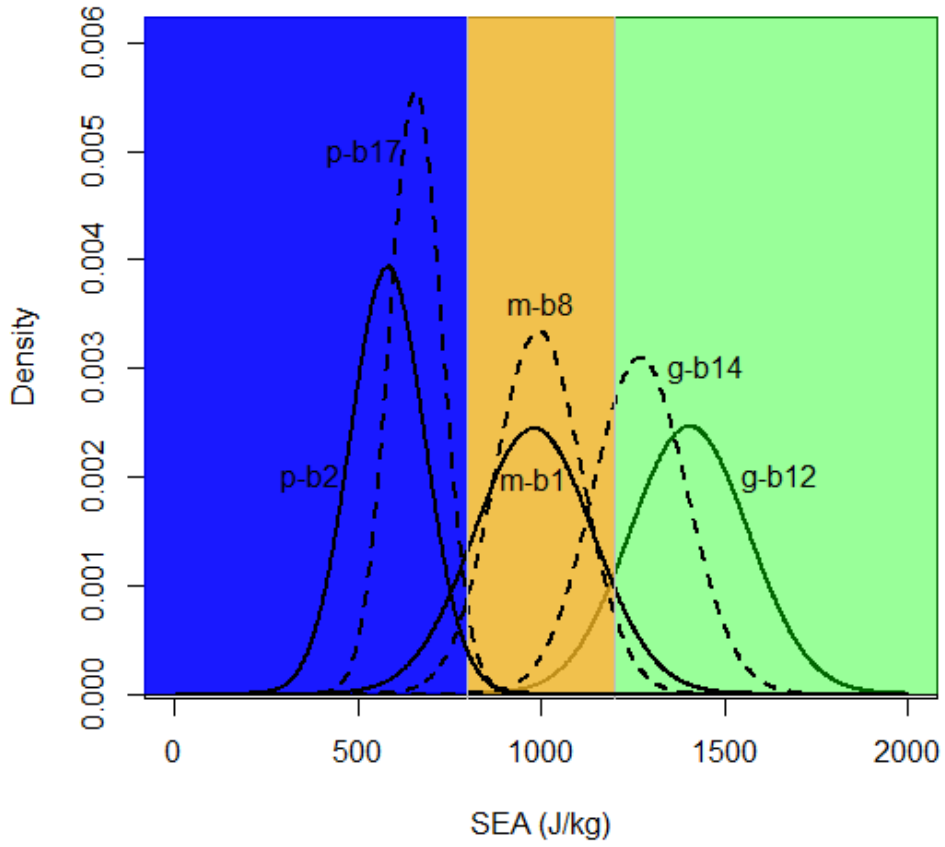


Figure 7.7. The performance (SEA) probability density function (PDF) for six branches produced by the DTUD.

7.4.2 Influence of uncertainty on DTUD performance

Uncertainty may influence design performance and the DTUD predictions. To evaluate such influences, 10 design alternatives were generated for each of the two "g" representatives (i.e. b12 and b14). Using R = 10%, the original exact values for design variables in these branches were replaced by distributions in intervals to reflect the uncertainty. The Gaussian distributions with a 3-sigma rule were defined for each design variable. These uncertain designs were tested by the decision tree to assess their LPs,

with the results listed in Table 7.2. Meanwhile, MCM was also adopted to quantify the response uncertainty of these 20 designs. Their "g" probabilities are listed in Table 7.2.

Table 7.2

The DTUD and Monte Carlo Method (MCM)-computed "g" probabilities of the 10 uncertain designs for b12 and b14

Design No.	b12 (**"g" LP: 0.95**)		b14 (**"g" LP**: 0.81)	
	DTUD	MCM	DTUD	MCM
1	0.95	1.00	0.77	1.00
2	0.95	0.99	0.81	0.93
3	0.91	1.00	0.81	0.68
4	0.95	0.99	0.78	0.63
5	**0.68**	**0.29**	0.79	1.00
6	**0.82**	**0.01**	**0.65**	**0.05**
7	**0.86**	**0.05**	**0.69**	**0.29**
8	0.95	1.00	0.76	0.94
9	0.95	1.00	**0.45**	**0.43**
10	0.94	1.00	0.80	0.99

In Table 7.2, there are three designs (bolded) in each group that present much lower probabilities than the nominal LP in their original leaf nodes. After adding the uncertainty, a design may be partitioned into multi-leaves, which may cause a decrease of "g" probability, especially for designs near the splitting bounds. When evaluated on performance distribution using the MCM, these three designs demonstrate a rather low "g" probability. This suggests that the performance of designs may be degraded by including uncertainty. This trend can also be indicated by the DTUD in predicted LPs that are much lower than the nominal LPs in leaf nodes.

7.5 Summary

To account for the uncertainty in engineering design, a DTUD algorithm was developed for the data with a given distribution. The accuracy of the algorithm was established by using nine datasets from a public UCI repository. To demonstrate the algorithm's capability in engineering design, the S-shaped beam model described in the previous chapters was employed again as an example. The design dataset for this component was generated using a GPR surrogate model.

A DTUD with six layers was built using this dataset. Its structure and accuracy were analyzed by examining the LP prediction capability. The results show the following:

1) A branch with a high LP can produce a design with a high probability of delivering the expected performance.

2) When considering uncertainty, the performance of a design rule may be degraded. This trend can be captured by the DTUD.

In summary, the DTUD offers an effective tool to deal with uncertain datasets with good classification accuracy. However, the DTUD based on the current algorithm can only be used for datasets with independent design variables and a label without uncertainty. In addition, the capability of this technique to handle categorical features and uncertainty in the response should be further developed.

Chapter 8 A comprehensive case study of whole-vehicle crashworthiness design using the new methodology

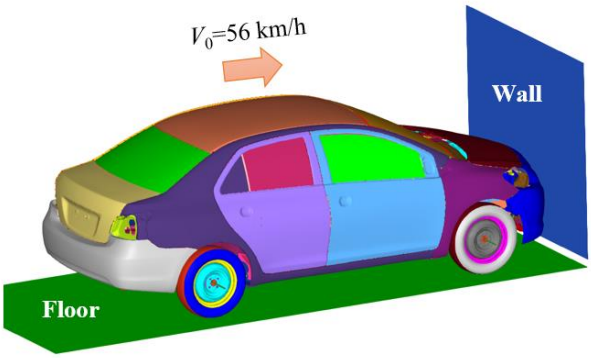
8.1 Introduction

To demonstrate the new design methodology developed in this work, a comprehensive case study was conducted. Here, a typical passenger car, a 2010 Toyota Yaris, was taken as the vehicle model for subsequent crashworthiness design. As described in the previous chapters, the design process includes the following steps: The FE model for the vehicle is first cleaned, simplified and validated using existing test data. Then, the system-level design is developed for the vehicle using the DMM (i.e. a decision tree) to identify the key energy-absorbing components, their boundary conditions and their design sequence. The following detailed design is focused on these components. Their simulation datasets are mined with the second decision tree to identify the key design variables and establish the design rules for the components. These redesigned components are then integrated back into the vehicle model to replace the original parts. A new crash simulation is conducted on the updated vehicle model to verify the performance of the new design methodology.

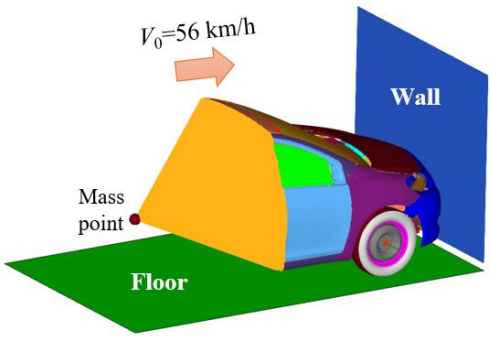
8.2 The basic vehicle model simplification and validation

A 2010 Toyota Yaris model was used in this case study. It is available in the NHTSA database (https://www.nhtsa.gov/es/crash-simulation-vehicle-models). It was developed by the National Crash Analysis Center (NCAC) of George Washington University through reverse engineering. The whole-vehicle model includes 974,383 elements, as illustrated in Figure 8.1(a). The response of this model has been validated by NHTSA

according to the NCAP full frontal impact conditions, as shown in Figure 8.2. More information related to this vehicle model is available on the NHTSA's website and is not repeated here.



(a)



(b)

Figure 8.1. The full and simplified FE models of the 2010 Toyota Yaris subject to an NCAP full frontal impact.

Under the frontal impact, the rear end of the vehicle is less important, since most of the impact energy is absorbed by the frontal structures and the deformation of the rear end is minor. Therefore, the whole vehicle can be simplified by removing the rear elements. A mass point can be added to ensure the simplified vehicle remains at the same weight, as

shown in Figure 8.1(b). The number of elements was reduced from 974,383 to 427,068. In this way, the computational cost of the simulations was significantly reduced. The simplified model was then validated under the same condition using two test cases (Nos. 05667 and 06221) available in NHTSA vehicle crash test database. Figure 8.2 shows the comparison of simulated accelerations by the simplified model and test data, together with the prediction of the original whole-vehicle model. Due to the elimination of rear seats, the mass point acceleration was taken as the substitute for the rear seat. The results show that the responses of the original and simplified are similar and can match the test data. Therefore, the response accuracy of the simplified model has been validated.

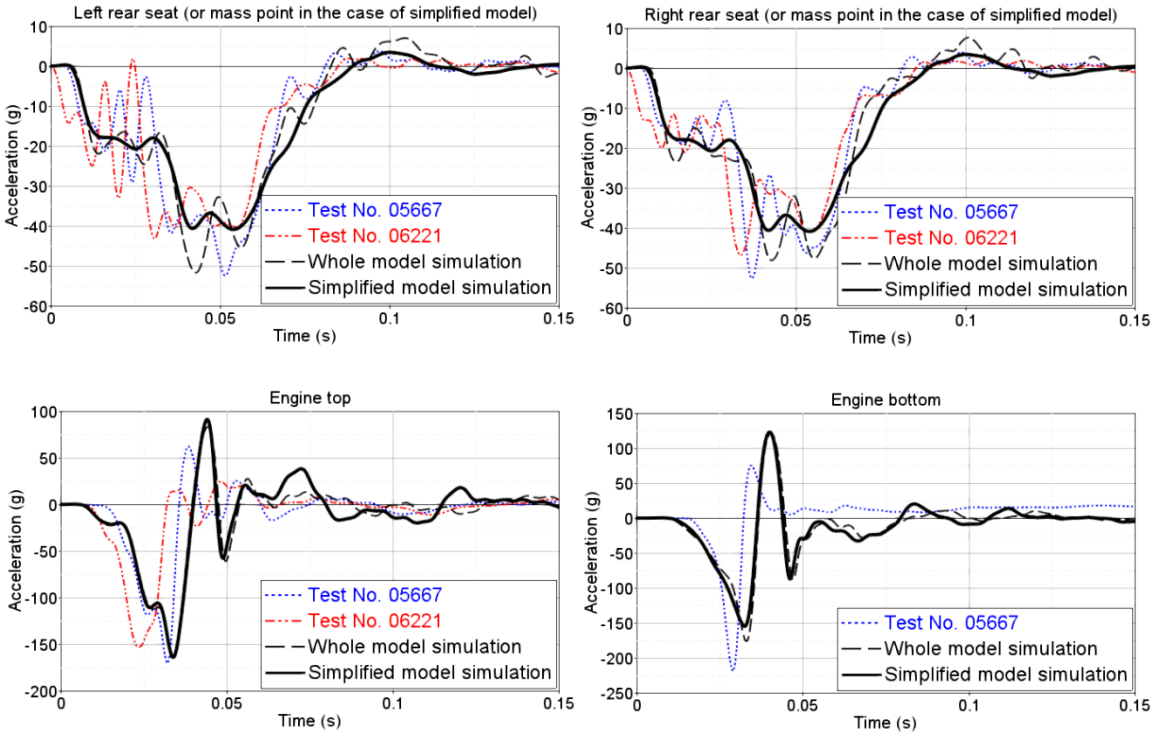


Figure 8.2. Validation of the original and simplified vehicle models.

8.3 Design at the system and component levels

Using the simplified vehicle model validated in Section 8.2, system-level and component-level crashworthiness designs were developed using the methods described in Chapters 3 through 7. The main steps are graphically shown in Figure 8.3.

8.3.1 System level

Based on the strategy in Chapter 3, prior to the design, one simulation was carried out to screen the key energy-absorbing components. The results show that four components absorb 85% of the energy. They are identified as P1: bumper (①), P2: front frame (② + ③), P3: rail (④ + ⑤) and P4: floor support (⑥ + ⑦). The subsequent design was then focused on these four components. Following the method in Chapter 3, the average stiffness (*avgstiff*) of each component was used as the attribute at the system-level design. The adjustment of this parameter could be implemented by changing the metal sheet thickness. Their ranges were determined as $2.0 \le T_1, T_6, T_7 \le 3.0$, $1.5 \le T_2, T_4, T_5, T_8 \le 2.5$ and $1.0 \le T_3 \le 2.0$, where T_i is the thickness of P_i (i = 1, 2, ⋯8). In this way, 150 design alternatives were generated by LHM and evaluated by the simplified model. The initial ranges for *avgstiff* values were as follows (unit: kN/m): $1{,}848.1 \le P1 \le 10{,}243.7$; $382.0 \le P2 \le 949.2$; $178.6 \le P3 \le 988.2$; and $120.5 \le P4 \le 296.1$. The intrusion into the passenger compartment, the peak impact force (*PkF*) and the mass of four selected parts were selected as the design objectives. The labeling strategy for these parameters was determined as "good" or "g": *PkF* < 800 kN, *Intrusion* < 220 mm and *Mass* < 27 kg; "intermediate" or "m": *PkF* < 800 kN, *Intrusion* ≤ 260 mm and *Mass* ≤ 28 kg; and "poor" or "p": other values.

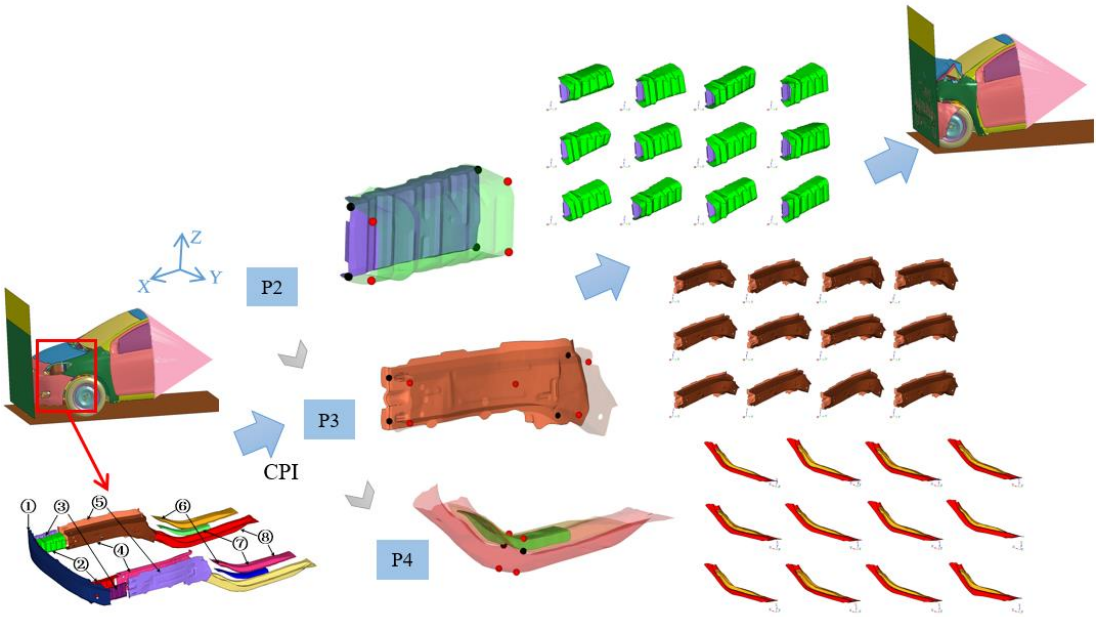


Figure 8.3. Systematic framework to develop the vehicle crashworthiness design in the case study.

Using the labeled dataset, a decision tree was trained as shown in Figure 8.4 containing 13 non-leaf and 14 leaf nodes. Branch b4 was selected as the "g" representative because it had the highest accuracy on the classification of good ("g") design cases. Using the rules for b4, three key components (P2, P3 and P4) were identified. Since P1 was not included in the decision tree, it is not shown in Figure 8.3. The ranges of *avgstiff* values for P2, P3 and P4 as well as the boundary conditions were determined. All of this information could be used in the next step for detailed component-level design.

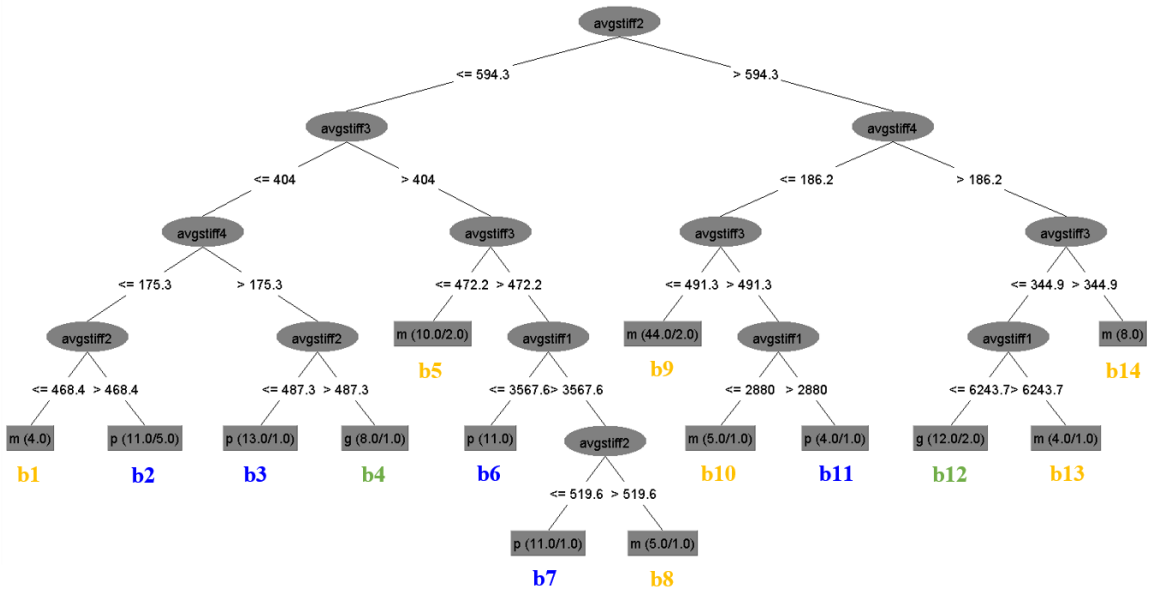


Figure 8.4. System-level decision tree in the current case study, where avgstiff1 represents the *avgstiff* of P1.

8.3.2 Component level

Based on the results of the system-level design, detailed designs for the aforementioned three components were performed. The parameterization of these components was achieved using node locations defined in several selected cross-sections, as shown in Figure 8.3. By changing the locations of the nodes, the geometry could be morphed to a new model correspondingly. For each component, 300 design alternatives were generated by LHM and simulated. Using the SEA and component mass as responses, the dataset was labeled and used to train a DTUD as introduced in Chapter 7 for each component. Design rules were generated for each component to help designers understand the component's design.

8.4 New design generation and whole-vehicle evaluation

Using the decision-making rules derived for components, CPI and DDR were performed to identify the key design variables and their value ranges. Then, new designs were generated.

8.4.1 New design generation and evaluation

For each of the three identified critical components, 10 design alternatives were generated randomly within the reduced design subdomain. In this way, each of the three components could be regarded as a discrete design variable with 10 values. Therefore, there were 1,000 (= 10 × 10 × 10) possible combinations in total. Twenty combinations were randomly generated from these 1,000 cases for further evaluation. These 20 combinations can be regarded as 20 new designs.

The 20 newly created designs were integrated back in to replace the corresponding parts in the whole-vehicle model to conduct the system-level simulations. The masses of four parts and passenger compartment intrusion were taken as the responses of interest. The mass and intrusion distributions of the 20 designs are shown in Figure 8.5, together with the results of the original design (mass = 25.3 kg and intrusion = 238.1 mm). The result shows that most of the designs produced by the new design method performed better in terms of both responses. The maximum mass saving is 13.4% (21.9 kg), and the maximum intrusion reduction is 25.9% (176.5 mm). This indicates the performance improvement from using the new design methodology.

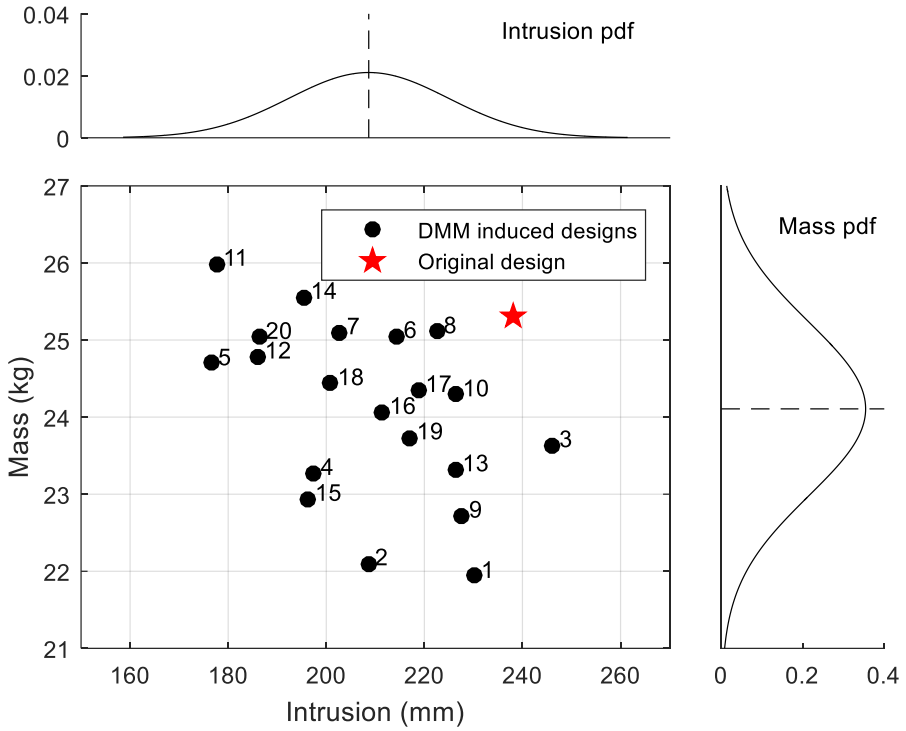


Figure 8.5. Distribution of the mass and intrusion for all 20 new designs, together with the results for the original design.

Likewise, the intrusion and mass of the 20 designs both exhibit a normal distribution. It can be determined that the probabilities that the new designs' intrusion and mass have lower values than those of the original design are 94.3% and 85.7%, respectively. This again indicates the high performance of the new method.

8.4.2 Dynamic response analysis

An additional comparison was made on the acceleration–time history at the location of the mass point, as shown in Figure 8.6. A lower acceleration indicates better crashworthiness. The results show that the acceleration response of the original design and four randomly selected new designs exhibit a similar pattern. The peak acceleration

occurs between 0.04 s and 0.05 s. The magnitudes of peak acceleration in the new designs are reduced by 4.2% ~ 12.5% compared to the original design.

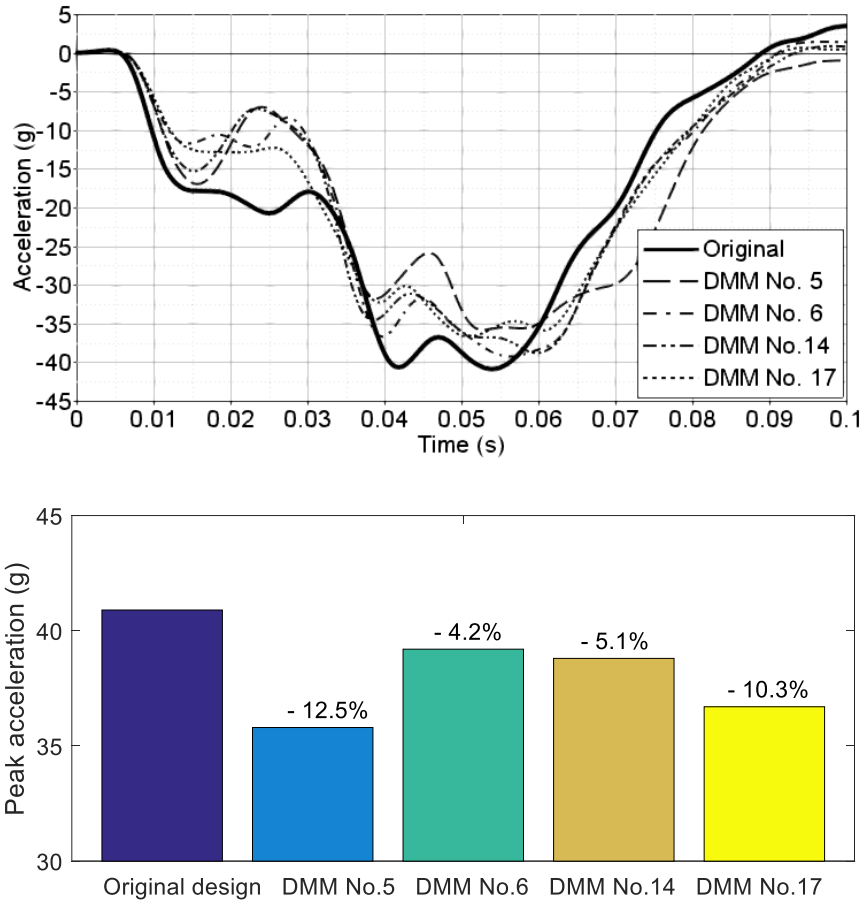


Figure 8.6. Comparison of acceleration history and peak value in the original design and new designs (Nos. 5, 6, 14 and 17).

In an additional comparison, the firewall deformation was examined. The firewall is a nearly 2D structure separating the engine and passenger compartment, and its intrusion or degree of deformation is another indicator of vehicle crashworthiness. The deformation contours of the four selected designs when t = 90 ms are shown in Figure 8.7, together

with the result for the original design. The comparison shows that the new designs can decrease the maximum deformation of the firewall by 20.3~35.6%.

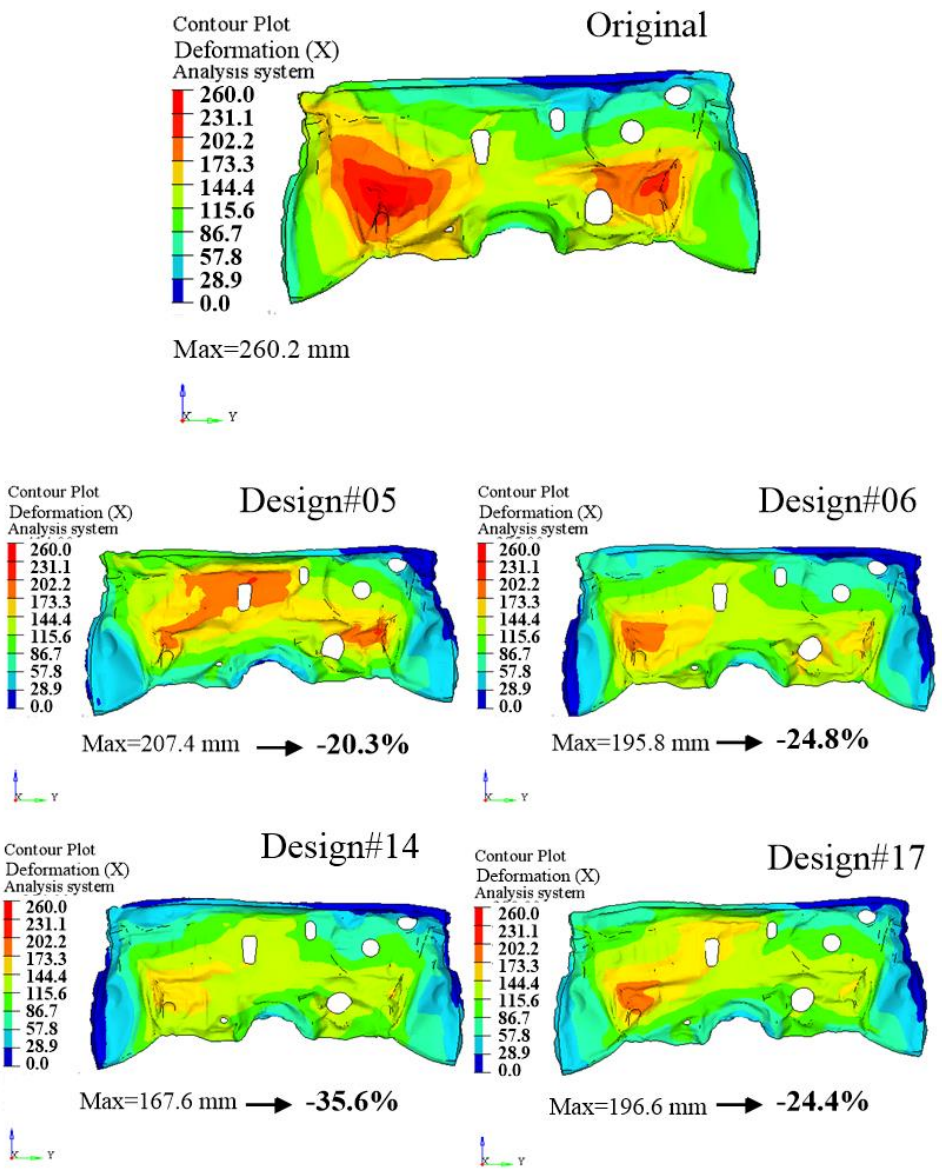


Figure 8.7. Comparison of the firewall deformation contours of the original design and four new designs at t = 90 ms. The maximum deformation magnitude is also shown together with the percentage decrease of maximum deformation compared to the original design.

8.5 Summary

A comprehensive case study was conducted to demonstrate the performance of the new design methodology developed in this research. A 2010 Toyota Yaris was taken as an example for subsequent crashworthiness design. The validated whole-vehicle FE model taken from the NHTSA database was simplified to reduce the computational cost. Using the simplified car model, DOEs were generated and simulated to form a simulation dataset at system level. A decision tree was created to identify the key energy-absorbing components, determine their boundary conditions and derive the design rules.

Using this information discovered at the system level, detailed designs were developed for the identified key components. Component-level decision trees were built for each component to identify the critical design variables, their ranges and the design rules.

Based on the above information at the component level, new component designs were generated and these new components were integrated back into the simplified vehicle model to simulate a frontal impact test. The results showed the designs based on the new method could outperform the original design in terms of the mass, intrusion and peak acceleration. The maximum decreases of these responses were 13.4%, 12.5% and 35.6%, respectively. Therefore, the performance of the new design methodology has been confirmed.

Chapter 9 Conclusions and future work

9.1 Conclusions

This research develops a new data-driven, knowledge-based methodology to design a complex mechanical system, such as a crashworthy vehicle, effectively and efficiently. A comprehensive literature review indicates that the two commonly applied traditional design methods cannot be used to achieve this goal. The iteration-based method, which features a large number of trial-and-error iterations, is unable to deal with complex design problems due to its low efficiency and effectiveness. The population-based methods generate a group of designs for performance evaluation by means of either numerical simulations or surrogate models. Such methods automate the design process and achieve an optimal design by applying a wide range of optimization algorithms. However, because they are unable to mine large datasets, they are unable to discover the implicit complex interrelationships of different components and design variables or to derive design rules for explaining a complex multilevel system. Knowledge-based design (KBD) approaches can be used to overcome the limitations associated with the aforementioned methods and to discover the desired relationships by exploring a large amount of design data. Such approaches can also be used to generate design rules to make appropriate decisions in a top-down design sequence without intensive human intervention.

The new design methodology for vehicle crashworthiness was developed based on data mining theory. The method allows the exploration of large crash simulation datasets to discover the underlying complicated relationships between response and design variables

at multiple levels (main energy-absorbing system, components and geometric features) and derive design rules based on the whole-vehicle body safety requirements, which can then be used to make decisions about the component and subcomponent-level design. The decision tree is a decision support tool that uses a tree-like graph to model decisions and their possible consequences. It is well suited to representing categorical knowledge such as design performance classes. Based on the decision trees, the interrelationships among the design variables can be revealed, and the design rules leading to good design sets can be derived by following the corresponding branches on the tree.

Based on the data mining method (DMM), two approaches were developed for the detailed component design, namely, a dimension-based method and a node-based method. A thin-walled vehicular structure, that is, an S-shaped beam, was designed using these two approaches to against crash loading. In the dimension-based approach, the component was represented by a number of geometric features, such as edge length, circle radius and sheet thickness. By adjusting the dimensions of these features, the design can be modified. Therefore, these geometric features with corresponding dimensions can be used as design variables. A large number of design alternatives were created, and simulations were conducted to evaluate their response as measured by specific energy absorption (SEA). The design variables and computed response formed a large design dataset. This dataset was then mined to build a decision tree. Based on the decision tree, the interrelationships among the design variables were revealed, and design rules were generated to produce good design sets. The accuracy of this design method was verified by using additional simulation data. After the data mining, the critical design parameters were identified and the design space could be reduced. Through these two

steps, future designs for similar problems can be significantly simplified. However, the dimension-based approach can only be applied to components with a regular shape. For those structures with a highly irregular shape, the node-based method should be used, because it is able to model complex shapes.

In the node-based approach, the geometry of the structure was initially parameterized by a large number of nonuniform rational basis spline (NURBS) control points. Principal component analysis (PCA) was then performed on this dataset, and the principal components (PCs) and principal component scores (PCSs) were calculated. The PCSs were adopted as the design variables in the subsequent design process to modify the geometry of the beam. Instead of directly handling a large number of NURBS points, one can modify the design by adjusting a small number of PCSs. A large number of design alternatives were generated in this way, and their geometries were recovered from the PCSs to the NURBS points and further converted into FE models to simulate their responses to a frontal impact. Two structural responses, SEA and CFE, were calculated and stored in the simulation dataset. Based on this dataset, the decision tree method was used to identify the interrelationships of design variables (i.e. PCSs) and their effects on the response. Design rules, that is, the workflows to determine the values of the design variables, were also derived. The results demonstrate that this new node-based approach powered by a PCA technique can be used to efficiently design a complex structure with an irregular shape.

To reduce the high cost of a large number of numerical modelings and simulations, efforts were made to select a proper surrogate model to supplement or partially replace the simulations. In this research, four learning algorithms (MLAs) for regression that

could be used to develop surrogate models were selected and compared: Gaussian process regression (GPR), support vector machines (SVMs), random forest regression (RFR) and artificial neural networks (ANNs). Four typical structural analysis problems were used to evaluate their performance. Using a multi-objective hyperparameter optimization strategy, the hyperparameters of these four MLAs were also optimized. The results show that GPR and ANN outperform the other two in terms of prediction accuracy and computational cost. GPR was used in the subsequent uncertainty analysis. The findings also show that the multi-objective hyperparameter optimization algorithm could tune the hyperparameters effectively and efficiently and reach a highly accurate surrogate model within a limited number of evaluations.

To account for the unavoidable uncertainty in engineering design, a decision tree for uncertain data (DTUD) algorithm was developed for the data with a given distribution. Its accuracy was proved using nine datasets from a public UCI repository. To demonstrate the DTUD's capability in engineering design, the S-shaped beam model described in the previous chapters was again employed as an example. The design dataset for this component was generated using a GPR surrogate model, and a DTUD with six layers was built using this dataset. Its structure and accuracy were analyzed by examining its label probability (LP) prediction ability. The results show that a branch with a high LP has a high probability of producing designs that achieve the expected performance. When considering uncertainty, the performance of a design rule may be degraded. This trend can be captured by the DTUD.

A comprehensive case study was conducted to design a vehicle system against a frontal crash to evaluate all of the new approaches developed in this work. A 2010 Toyota Yaris

was used as an example. The validated FE model was simplified to reduce computational cost. Using this simplified model, design alternatives were generated and simulated to form a simulation dataset at the system level. A decision tree was created to identify the key energy-absorbing components, determine their boundary conditions and derive design rules. Using this information discovered at the system level, detailed designs were developed for the identified key components. Component-level decision trees were built for each of these components to identify the critical design variables, their ranges and design rules. Based on the knowledge gained at the component level, new component designs were generated, and these new components were integrated back into the simplified vehicle model to simulate a frontal impact test. The results show that the designs based on the new method outperform the original design in terms of the mass, intrusion and peak acceleration. The maximum decreases of these responses were 13.4%, 12.5% and 35.6%, respectively. Therefore, the performance of the new design methodology has been confirmed.

9.2 Future work

Based on the results of the present research, the following future studies are suggested:

1) In the generation of decision trees, only one basic algorithm, C4.5, was used. The design results in the present study have proven its high performance. However, no comparison was made between this algorithm and the others available in the literature. Future work will include a comprehensive comparison of accuracy, robustness and computational cost.

2) Quality (e.g. degree of scattering) of the design space for DOEs and the size of datasets could influence the generated decision tree, and this issue will be considered in the future.
3) A high level of accuracy in a decision tree increases model complexity, which induces difficulties in generating clear decision-making rules. The trade-off between model accuracy and rule-generation will be studied.
4) Four surrogate models were discussed in this study, as they are commonly used in structural engineering. However, there are several others that can also potentially be used to design a complex system. These algorithms will be examined in detail as applied to mechanical system design.
5) In the present study, a DTUD offers an effective tool to deal with an uncertain dataset with good classification accuracy. A DTUD based on the current algorithm, however, is only suited to datasets with independent design variables. In addition, the capability of this technique to handle categorical features and uncertainty in the response is limited. These limitations will be considered in future work.
6) The data-driven method developed in this study has been applied to improve current vehicle safety design. An effort will be made to extend it as a general approach and apply it to the design of other complex mechanical systems as well, such as multi-physics vehicle battery systems and multi-scale material design.

References

ABBASI, M., REDDY, S., GHAFARI-NAZARI, A. & FARD, M. 2015. Multiobjective crashworthiness optimization of multi-cornered thin-walled sheet metal members. *Thin-Walled Structures,* 89**,** 31-41.

AMBROSIO, J. & DIAS, J. 2007. A road vehicle multibody model for crash simulation based on the plastic hinges approach to structural deformations. *International Journal of Crashworthiness,* 12**,** 77-92.

ANGIULLI, F. & FASSETTI, F. 2013. Nearest Neighbor-Based Classification of Uncertain Data. *ACM Transactions on Knowledge Discovery from Data,* 7**,** 1-35.

BAGHERI, S., KONEN, W. & BÄCK, T. Online selection of surrogate models for constrained black-box optimization. Computational Intelligence (SSCI), 2016 IEEE Symposium Series on, 2016. IEEE, 1-8.

BAI, Z., LIU, J., ZHU, F., WANG, F. & JIANG, B. 2015. Optimal design of a crashworthy octagonal thin-walled sandwich tube under oblique loading. *International Journal of Crashworthiness,* 20**,** 401-411.

BAI, Z., SUN, K., ZHU, F., CAO, L., HU, J., CHOU, C. C. & JIANG, B. 2018. Crashworthiness optimal design of a new extruded octagonal multi-cell tube under dynamic axial impact. *International Journal of Vehicle Safety,* 10**,** 40-57.

BARDENET, R., BRENDEL, M., KÉGL, B. & SEBAG, M. Collaborative hyperparameter tuning. International Conference on Machine Learning, 2013. 199-207.

BEN SALEM, M. & TOMASO, L. 2018. Automatic selection for general surrogate models. *Structural and Multidisciplinary Optimization,* 58**,** 719-734.

BENNETT, J. A. & BOTKIN, M. E. 1985. Structural shape optimization with geometric description and adaptive mesh refinement. *AIAA Journal,* 23**,** 458-464.

BERTHONNAUD, E., DIMNET, J., ROUSSOULY, P. & LABELLE, H. 2005. Analysis of the sagittal balance of the spine and pelvis using shape and orientation parameters. *Clinical Spine Surgery,* 18**,** 40-47.

BISCHL, B., RICHTER, J., BOSSEK, J., HORN, D., THOMAS, J. & LANG, M. 2017. mlrMBO: A modular framework for model-based optimization of expensive black-box functions. *arXiv preprint arXiv:1703.03373*.

BOURSIER NIUTTA, C., WEHRLE, E. J., DUDDECK, F. & BELINGARDI, G. 2018. Surrogate modeling in design optimization of structures with discontinuous responses. *Structural and Multidisciplinary Optimization,* 57**,** 1857-1869.

BREDBENNER, T. L., ELIASON, T. D., POTTER, R. S., MASON, R. L., HAVILL, L. M. & NICOLELLA, D. P. 2010. Statistical shape modeling describes variation in tibia and femur surface geometry between Control and Incidence groups from the osteoarthritis initiative database. *J Biomech,* 43**,** 1780-6.

BROCHU, E., BROCHU, T. & DE FREITAS, N. A Bayesian interactive optimization approach to procedural animation design. Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2010. Eurographics Association, 103-112.

BURROWS, S., STEIN, B., FROCHTE, J., WIESNER, D. & MÜLLER, K. Simulation data mining for supporting bridge design. Proceedings of the Ninth Australasian Data Mining Conference-Volume 121, 2011. Australian Computer Society, Inc., 163-170.

CAI, S., ZHANG, W., ZHU, J. & GAO, T. 2014. Stress constrained shape and topology optimization with fixed mesh: A B-spline finite cell method combined with level set function. *Computer Methods in Applied Mechanics and Engineering,* 278**,** 361-387.

CARVALHO, M. & AMBRÓSIO, J. 2010. Identification of multibody vehicle models for crash analysis using an optimization methodology. *Multibody System Dynamics,* 24**,** 325-345.

CHARYTANOWICZ, M., NIEWCZAS, J., KULCZYCKI, P., KOWALSKI, P. A., ŁUKASIK, S. & ŻAK, S. 2010. Complete gradient clustering algorithm for features analysis of x-ray images. *Information technologies in biomedicine.* Springer.

CHASE, N., SIDHU, R. & AVERILL, R. A new method for efficient global optimization of large systems using sub-models: HEEDS COMPOSE demonstrated on a crash optimization problem. LS-DYNA user forum, 2012.

CHEN, Y., WEN, J. & CHENG, S. 2013. Probabilistic load flow method based on Nataf transformation and Latin hypercube sampling. *IEEE Transactions on Sustainable Energy,* 4**,** 294-301.

CHOI, Y., FREEMAN, S. & LETAILLEUR, F. 2019. Constructing a Concept Vehicle Structure Optimized for Crashworthiness. SAE Technical Paper.

CLUNE, R., KELLIHER, D., ROBINSON, J. C. & CAMPBELL, J. S. 2014. NURBS modeling and structural shape optimization of cardiovascular stents. *Structural and Multidisciplinary Optimization,* 50**,** 159-168.

COUCKUYT, I., DE TURCK, F., DHAENE, T. & GORISSEN, D. Automatic surrogate model type selection during the optimization of expensive black-box problems. Simulation Conference (WSC), Proceedings of the 2011 Winter, 2011. IEEE, 4269-4279.

CRAIG, K. J., STANDER, N., DOOGE, D. A. & VARADAPPA, S. 2005. Automotive crashworthiness design using response surface-based variable screening and optimization. *Engineering Computations,* 22**,** 38-61.

DÍAZ-MANRÍQUEZ, A., TOSCANO-PULIDO, G. & GÓMEZ-FLORES, W. On the selection of surrogate models in evolutionary optimization algorithms. Evolutionary Computation (CEC), 2011 IEEE Congress on, 2011. IEEE, 2155-2162.

DU, X. & ZHU, F. 2018. A new data-driven design methodology for mechanical systems with high dimensional design variables. *Advances in Engineering Software,* 117**,** 18-28.

DU, X., ZHU, F. & CHOU, C. C. 2017. A New Data-Driven Design Method for Thin-Walled Vehicular Structures Under Crash Loading. *SAE Int. J. Trans. Safety,* 5**,** 188-193.

DUAN, L., JIANG, H., GENG, G., ZHANG, X. & LI, Z. 2018. Parametric modeling and multiobjective crashworthiness design optimization of a new front longitudinal beam. *Structural and Multidisciplinary Optimization*.

DUDDECK, F. & VOLZ, K. A new topology optimization approach for crashworthiness of passenger vehicles based on physically defined equivalent static loads. Proceedings ICRASH conference, Milano Google Scholar, 2012.

DUDDECK, F. & ZIMMER, H. 2013. Modular Car Body Design and Optimization by an Implicit Parameterization Technique via SFE CONCEPT. 195**,** 413-424.

EICHMUELLER, G. & MEYWERK, M. 2019. On computer simulation of components for automotive crashworthiness: validation and uncertainty quantification for simple load cases. *International Journal of Crashworthiness***,** 1-13.

EVETT, I. W. & SPIEHLER, E. J. 1987. Rule induction in forensic science. *KBS in Goverment***,** 107-118.

FAN, X., WANG, P. & HAO, F. 2019. Reliability-based design optimization of crane bridges using Kriging-based surrogate models. *Structural and Multidisciplinary Optimization,* 59**,** 993-1005.

FANG, H., SOLANKI, K. & HORSTEMEYER, M. 2005. Numerical simulations of multiple vehicle crashes and multidisciplinary crashworthiness optimization. *International Journal of Crashworthiness,* 10**,** 161-172.

FANG, J., GAO, Y., SUN, G., QIU, N. & LI, Q. 2015. On design of multi-cell tubes under axial and oblique impact loads. *Thin-Walled Structures,* 95**,** 115-126.

FANG, J., GAO, Y., SUN, G., ZHANG, Y. & LI, Q. 2014. Crashworthiness design of foam-filled bitubal structures with uncertainty. *International Journal of Non-Linear Mechanics,* 67**,** 120-132.

FANG, J., QIU, N., AN, X., XIONG, F., SUN, G. & LI, Q. 2017. Crashworthiness design of a steel–aluminum hybrid rail using multi-response objective-oriented sequential optimization. *Advances in Engineering Software,* 112**,** 192-199.

FERREIRA, W. G. & SERPA, A. L. 2015. Ensemble of metamodels: the augmented least squares approach. *Structural and Multidisciplinary Optimization,* 53**,** 1019-1046.

FERREIRA, W. G. & SERPA, A. L. 2017. Ensemble of metamodels: extensions of the least squares approach to efficient global optimization. *Structural and Multidisciplinary Optimization,* 57**,** 131-159.

FISHER, R. A. 1936. The use of multiple measurements in taxonomic problems. *Annals of eugenics,* 7**,** 179-188.

FU, Y. & SAHIN, K. H. 2004. Better optimization of nonlinear uncertain systems (BONUS) for vehicle structural design. *Annals of Operations Research,* 132**,** 69-84.

GANDIKOTA, I., RAIS-ROHANI, M., DORMOHAMMADI, S. & KIANI, M. 2014. Multilevel vehicle-dummy design optimization for mass and injury criteria minimization. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering,* 229**,** 283-295.

GEORGIOS, K. & DIMITRIOS, S. Multi-Disciplinary Design Optimization exploiting the efficiency of ANSA-LSOPT-META coupling. Proceedings 7th European LS-DYNA conference, Salzburg, 2009.

GOEL, T., HAFTKA, R. T., SHYY, W. & QUEIPO, N. V. 2007. Ensemble of surrogates. *Structural and Multidisciplinary Optimization,* 33**,** 199-216.

GOEL, T., STANDER, N. & LIN, Y.-Y. 2010. Efficient resource allocation for genetic algorithm based multi-objective optimization with 1,000 simulations. *Structural and Multidisciplinary Optimization,* 41**,** 421-432.

GOLDBERG, D. E. & SAMTANI, M. P. Engineering optimization via genetic algorithm. Electronic computation, 1986. ASCE, 471-482.

GUI, C., BAI, J. & ZUO, W. 2018. Simplified crashworthiness method of automotive frame for conceptual design. *Thin-Walled Structures,* 131**,** 324-335.

GUO, Z. & BAI, G. 2009. Application of least squares support vector machine for regression to reliability analysis. *Chinese Journal of Aeronautics,* 22**,** 160-166.

HALEEM, K. & GAN, A. 2013. Effect of driver's age and side of impact on crash severity along urban freeways: A mixed logit approach. *Journal of Safety Research,* 46**,** 67-76.

HAN, J., PEI, J. & KAMBER, M. 2011. *Data mining: concepts and techniques,* Elsevier.

HESSE, D., HOPPE, F. & GROCHE, P. 2017. Controlling product stiffness by an incremental sheet metal forming process. *Procedia Manufacturing,* 10**,** 276-285.

HOJJAT, M., STAVROPOULOU, E. & BLETZINGER, K.-U. 2014. The Vertex Morphing method for node-based shape optimization. *Computer Methods in Applied Mechanics and Engineering,* 268**,** 494-513.

HORTON, P. & NAKAI, K. A probabilistic classification system for predicting the cellular localization sites of proteins. Ismb, 1996. 109-115.

HOU, S., DONG, D., REN, L. & HAN, X. 2012. Multivariable crashworthiness optimization of vehicle body by unreplicated saturated factorial design. *Structural and Multidisciplinary Optimization,* 46**,** 891-905.

HUANG, M. 2002. *Vehicle crash mechanics,* CRC press.

HUANG, T. 2006. Methodological and Practical Aspects of Data Mining in the Product Development Process. *VALUE WORLD,* 29**,** 8.

HUANG, Z., WANG, C., CHEN, J. & TIAN, H. 2011. Optimal design of aeroengine turbine disc based on kriging surrogate models. *Computers & Structures,* 89**,** 27-37.

HUH, H. & KANG, W. 2002. Crash-worthiness assessment of thin-walled structures with the high-strength steel sheet. *International Journal of Vehicle Design,* 30**,** 1-21.

HUTTER, F., HOOS, H. H. & LEYTON-BROWN, K. Sequential model-based optimization for general algorithm configuration. International Conference on Learning and Intelligent Optimization, 2011. Springer, 507-523.

IIHS 2002. Federal Motor Vehicle Safety Standards (FMVSS). *Occupant Crash Protection and Side Impact Protection.* Washington, DC: US Department of Transportation.

IMAN, R. L. 2008. Latin hypercube sampling. *Encyclopedia of quantitative risk analysis and assessment*.

JIN, R., CHEN, W. & SIMPSON, T. W. 2001. Comparative studies of metamodelling techniques under multiple modelling criteria. *Structural and multidisciplinary optimization,* 23**,** 1-13.

KAPANIA, R. K. & LIU, Y. 1998. Applications of artificial neural networks in structural engineering with emphasis on continuum models.

KHAKHALI, A., DARVIZEH, A., MASOUMI, A., NARIMAN-ZADEH, N. & SHIRI, A. 2010a. Robust Design of S-Shaped Box Beams Subjected to Compressive Load. *Mathematical Problems in Engineering,* 2010**,** 1-18.

KHAKHALI, A., NARIMAN-ZADEH, N., DARVIZEH, A., MASOUMI, A. & NOTGHI, B. 2010b. Reliability-based robust multi-objective crashworthiness

optimisation of S-shaped box beams with parametric uncertainties. *International Journal of Crashworthiness,* 15**,** 443-456.

KIM, H. M., RIDEOUT, D. G., PAPALAMBROS, P. Y. & STEIN, J. L. 2003. Analytical target cascading in automotive vehicle design. *J. Mech. Des.,* 125**,** 481-489.

KIM, H. S. & WIERZBICKI, T. 2001. Effect of the cross-sectional shape of hat-type cross-sections on crash resistance of an "S"-frame. *Thin-Walled Structures,* 39**,** 535-554.

KIM, N. H. & CHANG, Y. 2005. Eulerian shape design sensitivity analysis and optimization with a fixed grid. *Computer Methods in Applied Mechanics and Engineering,* 194**,** 3291-3314.

KIM, P. & DING, Y. 2005. Optimal engineering system design guided by data-mining methods. *Technometrics,* 47**,** 336-348.

KITAYAMA, S., SRIRAT, J., ARAKAWA, M. & YAMAZAKI, K. 2013. Sequential approximate multi-objective optimization using radial basis function network. *Structural and Multidisciplinary Optimization,* 48**,** 501-515.

KLEIN, A., FALKNER, S., BARTELS, S., HENNIG, P. & HUTTER, F. 2016. Fast Bayesian Optimization of Machine Learning Hyperparameters on Large Datasets. *arXiv preprint* arXiv:1605.07079.

KURTARAN, H., ESKANDARIAN, A., MARZOUGUI, D. & BEDEWI, N. E. 2002. Crashworthiness design optimization using successive response surface approximations. *Computational Mechanics,* 29**,** 409-421.

LEE, J., JEONG, H. & KANG, S. 2007. Derivative and GA-based methods in metamodeling of back-propagation neural networks for constrained approximate optimization. *Structural and Multidisciplinary Optimization,* 35**,** 29-40.

LEE, K. 1999. Principles of CAD/CAM/CAE Systems. *Assison Wesley Longman Inc. United States of America,* 640.

LEE, S., HA, J., ZOKHIROVA, M., MOON, H. & LEE, J. 2017. Background Information of Deep Learning for Structural Engineering. *Archives of Computational Methods in Engineering,* 25**,** 121-129.

LI, C., WHITE, R., FANG, X., WEAVER, M. & GUO, Y. 2017. Microstructure evolution characteristics of Inconel 625 alloy from selective laser melting to heat treatment. *Materials Science and Engineering: A,* 705**,** 20-31.

LI, Z., HU, J., REED, M. P., RUPP, J. D., HOFF, C. N., ZHANG, J. & CHENG, B. 2011. Development, validation, and application of a parametric pediatric head finite element model for impact simulations. *Ann Biomed Eng,* 39**,** 2984-97.

LIAO, X., LI, Q., YANG, X., LI, W. & ZHANG, W. 2008. A two-stage multi-objective optimisation of vehicle crashworthiness under frontal impact. *International Journal of Crashworthiness,* 13**,** 279-288.

LITTLE, M. A., MCSHARRY, P. E., ROBERTS, S. J., COSTELLO, D. A. & MOROZ, I. M. 2007. Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *Biomedical engineering online,* 6**,** 23.

LIU, X., WU, Y., WANG, B., DING, J. & JIE, H. 2016. An adaptive local range sampling method for reliability-based design optimization using support vector

machine and Kriging model. *Structural and Multidisciplinary Optimization,* 55**,** 2285-2304.

LOHWEG, V., HOFFMANN, J. L., DÖRKSEN, H., HILDEBRAND, R., GILLICH, E., HOFMANN, J. & SCHAEDE, J. Banknote authentication with mobile devices. Media Watermarking, Security, and Forensics 2013, 2013. International Society for Optics and Photonics, 866507.

LUKASZEWICZ, D. & AG, B. A Design Method for Robust Automotive and Aerospace Composite Structures Including Manufacturing Variations. 24th International Technical Conference on the Enhanced Safety of Vehicles (ESV) National Highway Traffic Safety Administration, 2015.

LUKASZEWICZ, D., HESSE, S., GRAFF, L., KERSCHER, S., DEOBALD, L., PARK, C. Y. & DESAI, N. A Design and Analysis Method for Automotive and Aerospace Composite Structures including Manufacturing Variations. Proc. of the American Society for Composites 29th Conference, 2014.

MEENAKSHI, S. & VENKATACHALAM, V. 2015. FUDT: A Fuzzy Uncertain Decision Tree Algorithm for Classification of Uncertain Data. *Arabian Journal for Science and Engineering,* 40**,** 3187-3196.

MEHMANI, A., CHOWDHURY, S., MEINRENKEN, C. & MESSAC, A. 2017. Concurrent surrogate model selection (COSMOS): optimizing model type, kernel function, and hyper-parameters. *Structural and Multidisciplinary Optimization,* 57**,** 1093-1114.

MOHAMAD, I. B. & USMAN, D. 2013. Standardization and its effects on K-means clustering algorithm. *Research Journal of Applied Sciences, Engineering and Technology,* 6**,** 3299-3303.

MOSAVI, A. 2014a. Application of data mining in multiobjective optimization problems. *International Journal for Simulation and Multidisciplinary Design Optimization,* 5**,** A15.

MOSAVI, A. 2014b. Data mining for decision making in engineering optimal design. *Journal of AI and Data Mining,* 2**,** 7-14.

MOSAVI, A., HOFFMANN, M. & MILANI, A. Optimal design of the NURBS curves and surfaces utilizing multiobjective optimization and decision making algorithms of RSO. Conference of PhD Students in Mathematics, 2012.

MUKHERJEE, A. & DESHPANDE, J. M. 1995. Application of Artificial Neural Networks in Structural Design Expert-Systems. *Computers & Structures,* 54**,** 367-375.

NAGENDRA, S., STAUBACH, J. B., SUYDAM, A. J., GHUNAKIKAR, S. J. & AKULA, V. R. 2004. Optimal rapid multidisciplinary response networks: RAPIDDISK. *Structural and Multidisciplinary Optimization,* 29**,** 213-231.

NGUYEN, V. S., WEN, G., YIN, H. & VAN, T. P. 2014. Optimisation design of reinforced S-shaped frame structure under axial dynamic loading. *International Journal of Crashworthiness,* 19**,** 385-393.

NHTSA 1997. Federal Motor Vehicle Safety Standards (FMVSS). *Occupant Crash Protection.* Washington, DC: US Department of Transportation.

NHTSA 2008. New Car Assessment Program (NCAP). Washington, DC: US Department of Transportation.

OZCANAN, S. & ATAHAN, A. O. 2019. RBF surrogate model and EN1317 collision safety-based optimization of two guardrails. *Structural and Multidisciplinary Optimization***,** 1-20.

PALAR, P. S. & SHIMOYAMA, K. 2018. Efficient global optimization with ensemble and selection of kernel functions for engineering design. *Structural and Multidisciplinary Optimization,* 59**,** 93-116.

PAN, F., ZHU, P. & ZHANG, Y. 2010. Metamodel-based lightweight design of B-pillar with TWB structure via support vector regression. *Computers & Structures,* 88**,** 36-44.

PRADO, D. R., LÓPEZ-FERNÁNDEZ, J. A., ARREBOLA, M. & GOUSSETIS, G. 2018. Support Vector Regression to Accelerate Design and Crosspolar Optimization of Shaped-Beam Reflectarray Antennas for Space Applications. *IEEE Transactions on Antennas and Propagation***,** 1-1.

QIAN, J., YI, J., CHENG, Y., LIU, J. & ZHOU, Q. 2019. A sequential constraints updating approach for Kriging surrogate model-assisted engineering optimization design problem. *Engineering with Computers***,** 1-17.

QIN, B., XIA, Y. & PRABHAKAR, S. 2010. Rule induction for uncertain data. *Knowledge and Information Systems,* 29**,** 103-130.

RAIHAN, M., HOSSAIN, M. & HASAN, T. 2018. Data mining in road crash analysis: the context of developing countries AU - Raihan, Md Asif. *International Journal of Injury Control and Safety Promotion,* 25**,** 41-52.

REED, M. P. & PARKINSON, M. B. 2008. Modeling Variability in Torso Shape for Chair and Seat Design. 561-569.

ROCHA NETO, A. R. & ALENCAR BARRETO, G. 2009. On the application of ensembles of classifiers to the diagnosis of pathologies of the vertebral column: A comparative analysis. *IEEE Latin America Transactions,* 7**,** 487-496.

RODRIGUEZ-GALIANO, V., CHICA-OLMO, M. & CHICA-RIVAS, M. 2014. Predictive modelling of gold potential with the integration of multisource information based on random forest: a case study on the Rodalquilar area, Southern Spain AU - Rodriguez-Galiano, V.F. *International Journal of Geographical Information Science,* 28**,** 1336-1354.

RODRIGUEZ-GALIANO, V., SANCHEZ-CASTILLO, M., CHICA-OLMO, M. & CHICA-RIVAS, M. 2015. Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines. *Ore Geology Reviews,* 71**,** 804-818.

ROSHANIAN, J., BATALEBLU, A. A. & EBRAHIMI, M. 2018. A novel evolution control strategy for surrogate-assisted design optimization. *Structural and Multidisciplinary Optimization*.

SHI, L., YANG, R. J. & ZHU, P. 2012. A method for selecting surrogate models in crashworthiness optimization. *Structural and Multidisciplinary Optimization,* 46**,** 159-170.

SHIMOYAMA, K., LIM, J. N., JEONG, S., OBAYASHI, S. & KOISHI, M. 2009. Multi-Objective Robust Optimization Assisted by Response Surface Approximation and Visual Data-Mining. *In:* GOH, C.-K., ONG, Y.-S. & TAN, K.

C. (eds.) *Multi-Objective Memetic Algorithms.* Berlin, Heidelberg: Springer Berlin Heidelberg.

SNOEK, J., LAROCHELLE, H. & ADAMS, R. P. Practical bayesian optimization of machine learning algorithms. Advances in neural information processing systems, 2012. 2951-2959.

SOBIESZCZANSKI-SOBIESKI, J., KODIYALAM, S. & YANG, R. 2001. Optimization of car body under constraints of noise, vibration, and harshness (NVH), and crash. *Structural and multidisciplinary optimization,* 22**,** 295-306.

SONG, X., SUN, G., LI, G., GAO, W. & LI, Q. 2012. Crashworthiness optimization of foam-filled tapered thin-walled structure using multiple surrogate models. *Structural and Multidisciplinary Optimization,* 47**,** 221-231.

SONG, X., SUN, G., LI, G., GAO, W. & LI, Q. 2013. Crashworthiness optimization of foam-filled tapered thin-walled structure using multiple surrogate models. *Structural and Multidisciplinary Optimization,* 47**,** 221-231.

SUN, W., JIN, J. H., REED, M. P., GAYZIK, F. S., DANELSON, K. A., BASS, C. R., ZHANG, J. Y. & RUPP, J. D. 2016. A method for developing biomechanical response corridors based on principal component analysis. *J Biomech,* 49**,** 3208-3215.

SUN, Y., YUAN, Y. & WANG, G. 2014. Extreme learning machine for classification over uncertain data. *Neurocomputing,* 128**,** 500-506.

TANG, Y. C. & CHEN, J. 2009. Robust design of sheet metal forming process based on adaptive importance sampling. *Structural and Multidisciplinary Optimization,* 39**,** 531-544.

TANG, Z., ZHU, Y., NIE, Y., GUO, S., LIU, F., CHANG, J. & ZHANG, J. 2016. Data-driven train set crash dynamics simulation. *Vehicle System Dynamics,* 55**,** 149-167.

TAO, J., WU, Q.-M. & SUN, C. 2009. Optimal structural design of complicated mechanical products based on data mining. *Journal of Shanghai Jiaotong University (Science),* 14**,** 720-724.

TAVAKKOL, B., JEONG, M. K. & ALBIN, S. L. 2017. Object-to-group probabilistic distance measure for uncertain data classification _ Elsevier Enhanced Reader. *Neurocomputing,* 230**,** 143-151.

TIAN, L., GAO, Y., FANG, J. & AN, X. 2015. Multi-objective optimisation of hybrid S-shaped rails under oblique impact loading. *International Journal of Heavy Vehicle Systems,* 22**,** 137.

TSANG, S., KAO, B., YIP, K. Y., HO, W. S. & LEE, S. D. 2011. Decision Trees for Uncertain Data. *Ieee Transactions on Knowledge and Data Engineering,* 23**,** 64-78.

VAN MIEGROET, L. 2012. *Generalized shape optimization using XFEM and level set description.* Université de Liège, Liège, Belgique.

VAPNIK, V. 2013. *The nature of statistical learning theory,* Springer science & business media.

VISHWANATHAN, A. 2017. Shape Optimization of a NURBS Modelled Coronary Stent Using Kriging and Genetic Algorithm. *Cardiology and Cardiovascular Research,* 1**,** 9.

WANG, H., LI, E. & LI, G. Y. 2010. Probability-based least square support vector regression metamodeling technique for crashworthiness optimization problems. *Computational Mechanics,* 47**,** 251-263.

WENG, Y., JIN, X., ZHAO, Z. & ZHANG, X. 2010. Car-to-pedestrian collision reconstruction with injury as an evaluation index. *Accident Analysis & Prevention,* 42**,** 1320-1325.

WITTEN, I. H., FRANK, E. & HALL, M. A. 2011. *Data Mining: Practical Machine Learning Tools and Techniques,* Morgan Kaufmann Publishers Inc.

YANG, H. Z., CHEN, J. F., MA, N. & WANG, D. Y. 2012. Implementation of knowledge-based engineering methodology in ship structural design. *Computer-Aided Design,* 44**,** 196-202.

YANG, R., WANG, N., THO, C., BOBINEAU, J. & WANG, B. 2005. Metamodeling development for vehicle frontal impact simulation. *Journal of Mechanical Design,* 127**,** 1014-1020.

YEH, I.-C., YANG, K.-J. & TING, T.-M. 2009. Knowledge discovery on RFM model using Bernoulli sequence. *Expert Systems with Applications,* 36**,** 5866-5871.

YILDIZ, A. R. & SOLANKI, K. N. 2012. Multi-objective optimization of vehicle crashworthiness using a new particle swarm based approach. *The International Journal of Advanced Manufacturing Technology,* 59**,** 367-376.

YIN, H., WEN, G., HOU, S. & QING, Q. 2013. Multiobjective crashworthiness optimization of functionally lateral graded foam-filled tubes. *Materials & Design,* 44**,** 414-428.

YIN, H., WEN, G., LIU, Z. & QING, Q. 2014. Crashworthiness optimization design for foam-filled multi-cell thin-walled structures. *Thin-Walled Structures,* 75**,** 8-17.

YOGATAMA, D. & MANN, G. 2014. Efficient Transfer Learning Method for Automatic Hyperparameter Tuning. *Artificial Intelligence and Statistics***,** 1077-1085.

YU, Y., HUR, T., JUNG, J. & JANG, I. G. 2018. Deep learning for determining a near-optimal topological design without any iteration. *Structural and Multidisciplinary Optimization*.

ZHANG, Y., SUN, G., LI, G., LUO, Z. & LI, Q. 2012. Optimization of foam-filled bitubal structures for crashworthiness criteria. *Materials & Design,* 38**,** 99-109.

ZHAO, D. & XUE, D. 2010. A comparative study of metamodeling methods considering sample quality merits. *Structural and Multidisciplinary Optimization,* 42**,** 923-938.

ZHAO, Z., JIN, X., CAO, Y. & WANG, J. 2010. Data mining application on crash simulation data of occupant restraint system. *Expert Systems with Applications,* 37**,** 5788-5794.

ZHOU, Y., LAN, F. & CHEN, J. 2011. Crashworthiness research on S-shaped front rails made of steel–aluminum hybrid materials. *Thin-Walled Structures,* 49**,** 291-297.

ZHU, P., PAN, F., CHEN, W. & ZHANG, S. 2012. Use of support vector regression in structural optimization: Application to vehicle crashworthiness design. *Mathematics and Computers in Simulation,* 86**,** 21-31.

ZIMMERMANN, M., KÖNIGS, S., NIEMEYER, C., FENDER, J., ZEHERBAUER, C., VITALE, R. & WAHLE, M. 2017. On the design of large systems subject to uncertainty. *Journal of Engineering Design,* 28**,** 233-254.

ZUO, W. & BAI, J. 2016. Cross-sectional shape design and optimization of automotive body with stamping constraints. *International Journal of Automotive Technology,* 17**,** 1003-1011.

Appendix A

Introduction of four regression machine learning algorithms

In this appendix, four MLAs, i.e. the GPR, SVM, RFR, and ANN, are briefly introduced in the context of engineering design. The design variables are presented as $\boldsymbol{x} \in \mathbf{R}^{n}$ (real value space with $\boldsymbol{n}$ dimensions), and the responses are noted as $\boldsymbol{y} \in \mathbf{R}^{m}$. The surrogate modeling aims to establish a prediction function between design variables and responses.

A.1 Gaussian Process Regression (GPR)

GPR is a non-parametric regression through Gaussian processes (GP), which is similar with the Kriging model with relative low training cost. It takes the function as a sample in GP and can be presented in the following form

$$\boldsymbol{y} = f(\boldsymbol{x}) + \delta \tag{A-1}$$

where, δ is a random noisy variable with standard normal distribution. The prior distribution of $f(\boldsymbol{x})$ is assumed to be a zero-mean Gaussian distribution (in Eq. A-2) to simplify the modeling since the mean value can be fitted easily. Hence, GPR only models the residual errors by

$$f(\boldsymbol{x}) \sim \mathcal{GP}(0, k(\boldsymbol{x}, \boldsymbol{x}^{'})) \tag{A-2}$$

where $k(\boldsymbol{x}, \boldsymbol{x}^{'})$ denotes covariance function. The training set is T = (X, Y), so any set of new or test dataset input $\boldsymbol{X}^{*}$ have joint multivariate Gaussian distribution with the training dataset in the form

$$\left[\boldsymbol{f}(\boldsymbol{X}), \boldsymbol{f}(\boldsymbol{X}^{*})\right] \left[\boldsymbol{X}, \boldsymbol{X}^{*}\right] \sim \boldsymbol{N}(\boldsymbol{0}, \begin{bmatrix} \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) & \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}^{*}) \\ \boldsymbol{K}(\boldsymbol{X}^{*}, \boldsymbol{X}) & \boldsymbol{K}(\boldsymbol{X}^{*}, \boldsymbol{X}^{*}) \end{bmatrix}) \tag{A-3}$$

By computing the conditional distribution, the prediction result $\boldsymbol{f}(\boldsymbol{X}^*)$ is also a Gaussian distribution:

$$\boldsymbol{f}(\boldsymbol{X}^*)\left\|\left[\boldsymbol{f}(\boldsymbol{X}), \boldsymbol{X}, \boldsymbol{X}^*\right] \sim \boldsymbol{N}(\mu^*, \Sigma^*)\right. \quad \text{(A-4)}$$

where,

$$\mu^* = \boldsymbol{K}(\boldsymbol{X}^*, \boldsymbol{X})\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})^{-1}\mathbf{Y} \quad \text{(A-5)}$$

$$\Sigma^* = \boldsymbol{K}(\boldsymbol{X}^*, \boldsymbol{X}^*) - \boldsymbol{K}(\boldsymbol{X}^*, \boldsymbol{X})\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})^{-1}\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}^*) \quad \text{(A-6)}$$

Covariance functions **K** are also called kernel functions, e.g. the linear, polynomial, exponential, Gaussian, Laplacian kernels, etc. Even the kernel parameters can be estimated by training, they can also be taken as hyperparameters for tuning a better model performance. Therefore, its tunable hyperparameters include the selection of kernel functions and corresponding different kernel parameters.

A.2 Support Vector Machine (SVM)

Different from the GP-based GPR, the SVM seeks to fit the structural response linearly with respect to the design variables although it also a kernel-based method. However, a complex structure often exhibits highly nonlinear responses and therefore, it can not be fitted linearly. In SVM, by mapping input features (independent variables) into a new higher *h* dimensional space, it can realize a linear regression in this new space with the scheme shown in Figure A-1. All data points can be fitted linearly by $f(\boldsymbol{x})$ by

$$f(\boldsymbol{x}) = \boldsymbol{\omega}^T \mu(\boldsymbol{x}) + b \quad \text{(A-7)}$$

where $\mu(\boldsymbol{x})$ is the mapping function: $\boldsymbol{x}_i \to \mu(\boldsymbol{x}_i) \in \mathrm{R}^h$, $\boldsymbol{x}_i$ is the input features of an observation.

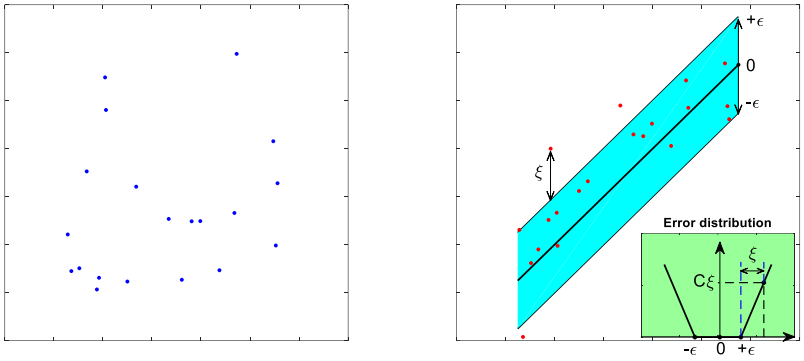


Figure A-1 Mechanism of design space mapping in the SVM: the original data on the left was mapped to a space with higher dimensions on the right

The training objective of SVM can be summarized as Eq. A-8. The model complexity is controlled by the first term, i.e. $\frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{\omega}$. As presented in Figure A-1, the Vapnik's ε-intensive cost function penalizes the data points outside ε-bands, which defines the penalization proportional to the distance from a specific ε-bounds. This is represented by the second term in Eq. A-8 under the constraints defined by Eq. A-9.

$$\boldsymbol{min} \quad \frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{\omega} + \boldsymbol{C}\sum_{i=1}^{N}(\boldsymbol{\xi}+\boldsymbol{\xi}^*) \tag{A-8}$$

subject to:

$$\begin{cases} y_i - (\boldsymbol{\omega}^T k(\boldsymbol{x}_i) + b) \le \varepsilon + \xi_i \\ (\boldsymbol{\omega}^T k(\boldsymbol{x}_i) + b) - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \quad \xi_i^* \quad \ge 0 \end{cases} \tag{A-9}$$

where $\boldsymbol{C}$ represents the weight of error penalization, which defines the trade-off between the model complexity and error term. ξ_i and ξ_i^* are two slack variables to deal with the points out of the ε bounds introduced by Vapnik et al. (Vapnik, 2013), which takes these points out of the ε bounds back to constraints. The Lagrange method is used by

introducing the constraints into the objective through Lagrange multiplier α and α^* and solving a dual problem through the quadratic programming procedure with the solution in Eq. A-10.

$$f(x) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) \langle \mu(\boldsymbol{x}_i), \mu(\boldsymbol{x}) \rangle + b \qquad \text{(A-10)}$$

where the dot product of $\mu(\boldsymbol{x}_i)$ and $\mu(\boldsymbol{x})$, $K(x_i, x) = \langle \mu(\boldsymbol{x}_i), \mu(\boldsymbol{x}) \rangle$ is the kernel function in GPR, so the kernel function selection and the kernel-related parameters are also tuned as hyperparameters. Meanwhile, the $\boldsymbol{C}$ and $\boldsymbol{\varepsilon}$ are two important hyperparameters also for SVM, which is also a specific feature different from other kernel-based methods.

A.3 Random Forest Regression (RFR)

RFR is a non-parametric method to train an ensemble of regression decision trees and uses the mean of all trees output as the output in Eq. A-11. Different from other method, it can take the categorical variable directly. Each decision tree is trained by using the error reduction method as the splitting criterion.

$$f(\boldsymbol{x}) = \frac{1}{\mathrm{N_R}} \sum_{d=1}^{\mathrm{N_R}} D_d(\boldsymbol{x}) \qquad \text{(A-11)}$$

where $\mathrm{N_R}$ is the pre-defined number of decision trees; $D_d(\boldsymbol{x})$ is the output of the *d*th decision tree.

The bagging technique is used to train an RFR model, which samples training data into some subsets randomly with replacement before training. Each subset is used to train a single decision tree. This could increase the diversity of decision trees and then, improve its robustness and reduces the generalization error. Meanwhile, each decision trees is

grown without pruning, which increases its diversity and soften the boundaries of decision tree node splitting. This could also reduce the prediction variance and then further improve the robustness. For RFR, the structure related parameters, e.g. the number of decision trees, maximum number of terminal nodes et al were tuned.

A.4 Artificial Neural Network (ANN)

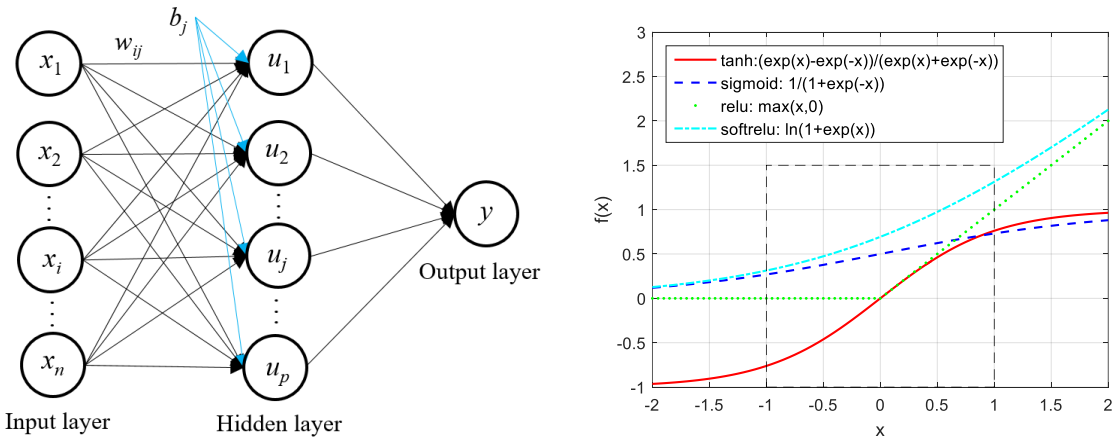
ANN is inspired by the bio-nervous system. Generally, one input layer, multi-hidden layers, and one output layer are fully connected by their neurons. For a single hidden layer ANN as shown in Figure A-2(a), the sum of weighted (w_{ij}) design variables (x_i) is taken as the input of jth hidden neuron and the activation function's result is its output in Eq. A-12, where b_j is a bias. After training, the weights and biases can be learned from the dataset and the trained model can predict new designs' responses.

$$u_j = f\left(\sum_{i=1}^{n} w_{ij} x_i + b_j\right) \tag{A-12}$$

where, x_i (i = 1, 2, ···, n) is a design variable; n is the number of design variables; $f()$ is the activation function. Some activation functions can be used, for example, **tanh**, **sigmoid**, **relu**, **softrelu,** etc. (Figure A-2b). The model output is expressed in Eq. A-13 without activation function.

$$f_o = \sum_{j=1}^{n_h} w_{jo} u_j \tag{A-13}$$

where, n_h is the number of hidden neurons. An activation function can also be used in the output neuron similar to the definition in Eq. A-12.



(a) (b)

Figure A-2 (a) General structure of ANN and (b) four activation functions for ANN hidden neurons

The training is to seek the optimum by using the gradient descent method to find a local minimum of a pre-defined loss function, e.g. the root mean square error. Some optimizers, e.g. **sgd**, **adagrad**, **adadellta**, **adam,** etc., can be used to control the optimum searching process. The optimizers and related parameters, activation functions, hidden layers, number of neurons in each hidden layers, etc. are tunable hyperparameters to improve the model accuracy and robustness.

Appendix B

DTUD verification by the datasets from UCI repository

From the UCI machine learning repository, nine datasets with real number input features are selected with the basic information summarized in Table B.1. For applying these datasets in the DTUD, their corresponding uncertain datasets were formed by following the same procedure in Chapter 7 with R = 10%. In this way, nine corresponding uncertain datasets are generated with interval input features. The Gaussian distribution with 3-sigma rule is also used to represent the probability on intervals. For each interval, 10 splitting points were distributed evenly. For each dataset, 80% observations were used in training and the remained data will be used in the validation process. To demonstrate the DTUD performance, the traditional decision tree (DTDD) is also adopted and trained by the corresponding deterministic dataset of these uncertain dataset for comparison.

The validation results are presented in Table B.2. Meanwhile, the validation ACCs of DTUD and DTDD are compared. In Table B.2, DTUD presents higher ACC than DTDD in seven of nine datasets, which suggest the superior performance of the DTUD by considering the uncertainty in the original dataset.

Table B.1

Basic information of nine datasets from the UCI repository

Names	No. of tuples	No. of input features	No. of labels	Years	References
BA	1372	4	2	2013	(Lohweg et al., 2013)
Ecoli	336	7	6	1996	(Horton and Nakai, 1996)
Glass	214	9	6	1987	(Evett and Spiehler, 1987)
Iris	150	4	3	1988	(Fisher, 1936)
Parkinsons	195	22	2	2008	(Little et al., 2007)
BT	784	4	2	2008	(Yeh et al., 2009)
Seeds	210	7	3	2012	(Charytanowicz et al., 2010)
VC 2D	310	6	2	2011	(Rocha Neto and Alencar Barreto, 2009)
VC 3D	310	6	3	2011	(Berthonnaud et al., 2005)

Note: VC: Vertebral Column; BA: Banknote Authentication; BT: Blood Transfusion

Table B.2

Comparison of ACC between DTUD and DTDD on nine datasets

Datasets	DTUD Validation	DTDD Validation
BA	**0.98**	0.96
Ecoli	**0.84**	0.80
Glass	**0.84**	0.65
Iris	**1.00**	0.94
Parkinsons	**0.87**	0.82
BT	0.73	0.76
Seeds	**0.93**	0.89
VC 2D	0.74	0.82
VC 3D	**0.85**	0.82

List of Publications

Journal papers:

DU, Xianping & Zhu, F. 2019. A novel principal components analysis (PCA) method for energy absorbing structural design enhanced by data mining. *Advances in Engineering Software,* 127**,** 17-27.

DU, **Xianping** & Zhu, F. 2018. A new data-driven design methodology for mechanical systems with high dimensional design variables. *Advances in Engineering Software,* 117**,** 18-28.

DU, **Xianping**, Deng, Y., Zhang, G., Cao, L. & Zhu, F. 2019. Conceptual design and evaluation of a novel bilateral pretension seatbelt: a computational study. *International Journal of Crashworthiness***,** 1-10.

DU, **Xianping**, Dori, A., Divo, E., Huayamave, V. & Zhu, F. 2018. Modeling the motion of small unmanned aerial system (sUAS) due to ground collision. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering,* 232**,** 1961-1970.

DU, **Xianping**, Zhu, F. & Chou, C. C. 2017. A New Data-Driven Design Method for Thin-Walled Vehicular Structures under Crash Loading. *SAE International Journal of Transportation Safety,* 5.

Zhu, F., Lei, J., **DU**, **Xianping**, Currier, P., Gbaguidi, A. & Sypeck, D. Crushing Behavior of Vehicle Battery Pouch Cell and Module: A Combined Experimental and Theoretical Study. SAE International Journal of Materials and Manufacturing 11.2018-01-1446 (2018): 341-348.

Conference presentations:

Keynote Speech, "A machine learning method for vehicle crashworthiness design." USNCCM15, Jul. 28- Aug. 01, 2019, Austin, USA.

"A Data Mining Method for Vehicle Crashworthiness Design." The 3rd Xiangjiang River International Forum for Outstanding Overseas Young Scholars（Sub-Forum of Traffic&Transportation Engineering）, May. 18-19, 2019, Changsha, China.

"A new crashworthiness design method for complex vehicular structures using principal components analysis (PCA) and data mining." ASME 2018 IMECE, Nov. 9-15, 2018, Pittsburgh, PA, USA.