IMPROVED ROBUSTNESS FORMULATIONS AND A SIMULATION-BASED ROBUST CONCEPT EXPLORATION METHOD

A Thesis
Presented to
The Academic Faculty

by

Markus Rippel

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Mechanical Engineering in the
George W. Woodruff School of Mechanical Engineering

Georgia Institute of Technology
December 2009

IMPROVED ROBUSTNESS FORMULATIONS AND A SIMULATION-BASED ROBUST CONCEPT EXPLORATION METHOD

Approved by:

Dr. Janet K. Allen, Advisor
G. W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Dr. Farrokh Mistree
G. W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Dr. Seung-Kyum Choi
G. W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: November 16, 2009

ACKNOWLEDGEMENTS

During my time as a Masters student at Georgia Tech, I was extremely lucky to learn with and from great educators, teachers, classmates and friends. Here, I want to thank especially those, who supported me during my research and the completion of this thesis.

First of all, I thank my advisor Dr. Janet K. Allen for her inspiration, support and guidance during my entire time at Georgia Tech. She always encouraged me to accept the challenge of research, which gave me a great insight to academia and on which I grew both intellectually and personally. I am also very thankful to my committee member Dr. Farrokh Mistree, the most dedicated educator I have met, for his support and contributions to my thesis. His insights and his consistent encouragement to think critically were major success factors over the course of my research. Furthermore, I owe great thanks to my committee member Dr. Seung-Kyum Choi, without whose guidance and support this thesis would have been nearly impossible. He provided great insights to statistics in engineering design and assisted during the generation of ideas for answering my research questions. I also thank Dr. Jitesh Panchal, Dr. Dirk Schaefer and Dr. Matthias Messer for their guidance, insights and friendship.

For his help with the application of JavaDSIDES, I owe great thanks to Dr. Matthew Marston. His patience in answering my questions and integrating my suggestions to JavaDSIDES were of enormous help for the progress of my research. For providing the simulation code for the LCA heat exchanger example I thank Dr. Carolyn C. Seepersad,

Dr. Hae-Jin Choi and Dr. Marco Gero Fernández. I am also thankful to Dr. Wei Chen, on whose PhD work my research is built, for her insights to RCEM.

Furthermore, I thank all faculty and students in the Systems Realization Laboratory both in Atlanta and in Savannah for being a great research family. Special thanks go to Timothy Dietz, Jiten Patel, Alex Ruderman, Chenjie Wang, Julie Bankston and Andrew Hyder for their friendship and encouragement during my time at Georgia Tech.

I owe the greatest thanks to my family, especially my mother, for their continuous support and encouragement during the course of my entire studies both in Germany and in the United States.

Finally, I thank the Federation of German-American Clubs (VDAC) and the German Academic Exchange Service for their support prior to and during my studies at Georgia Tech. Furthermore, I acknowledge the funding sources, which made this thesis possible:

- NSF grant DMI-060259 SACE: Statistics Aided Computer Experiments
- Georgia Tech Savannah
- The Center for Computational Materials Design; a joint project between Georgia Tech and the Pennsylvania State University

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

CHAPTER 2

$C(x)$	Capability of system
d_i^-, d_i^+	Deviation variable in a compromise DSP
$D(x)$	Demand of system
$g(x)$	Model/simulation outcome
g'_i, g_i	Coefficients in polynomial model approximation
$\hat{g}(x)$	Approximation of model response
$h(x)$	System constraint function
k	scaling factor
l	Number of noise factors
$L(y)$	Quadratic loss function
m	Number of system goals
n	Number of system variables
O_i	Orthonormal polynomials of order i
$p(x)$	Probability density function of x
p	Number of inequality constraints
q	Number of equality constraints
$Q(y)$	Expected quality loss function
T	Target value
w_i	Weight of i^{th} objective
x	Random design variable / system variable
$x_{i,L}$, $x_{i,U}$	Lower and upper bound of x
z	Noise factor
y	Performance function value
Z	Objective function / deviation function
μ, E[]	Mean Value
σ	Standard deviation
σ^2, Var[]	Variance

CHAPTER 3

d_i^-, d_i^+	Deviation variable in a compromise DSP
$f(x)$	Performance function
x	Random design variable
z	Noise factor
y	Performance function value
Z	Objective function
W_i	Weight of i^{th} objective
μ, E[]	Mean Value
σ	Standard deviation
σ^2, Var[]	Variance

CHAPTER 4

d_i^-, d_i^+	Deviation variable in a compromise DSP
L	Length of pressure vessel middle section
P	Internal pressure
R	Radius of pressure vessel
s_c	Maximum stress in cylindrical section
s_s	Maximum stress is spherical section
s_y	Material yield strength
SF	Safety factor
T	Thickness of the pressure vessel wall
V	Volume of pressure vessel
W	Weight of pressure vessel
W_i	Weight of i^{th} objective
$\mu, E[]$	Mean Value
ρ	Material density
σ	Standard deviation
$\sigma^2, Var[]$	Variance

CHAPTER 5

a_i	Coefficients of polynomial model approximation
d_i^-, d_i^+	Deviation variable in a compromise DSP
$f(x)$	Black-box performance function
H_i	Hermite polynomial of order i
W_i	Weight of i^{th} objective
x	Random design variable
y	Black-box model/simulation outcome
$\hat{y}$	Polynomial approximation of model response
z	Noise factor
Z	Objective function
μ, $E[]$	Mean Value
ξ	Standard distributed Gaussian random variable
σ	Standard deviation
σ^2, $Var[]$	Variance

CHAPTER 6

a_i	Coefficients of polynomial model approximation
d_i^-, d_i^+	Deviation variable in a compromise DSP
E_x/E_s	Overall elastic stiffness in x-direction, relative to the elastic modulus of the solid base material
E_y/E_s	Overall elastic stiffness in y-direction, relative to the elastic modulus of the solid base material
h_1	Height of top row
h_2	Height of bottom row
H	Overall height of the heat exchanger
H_i	Hermite polynomial of order i
k_s	Thermal conductivity of solid base material
L	Overall length of the heat exchanger
N_v	Number of rows
N_H	Number of columns
$\dot{Q}$	Total heat transfer rate
t_H	Wall thickness on horizontal direction
t_V	Wall thickness in vertical direction
T_{in}	Temperature of air at inlet
T_s	Temperature of heat source

$\dot{V}$	Volumetric flow rate of air/coolant
w_i	Width of cells in column i
W	Overall width of the heat exchanger
W_i	Weight of i^{th} objective
Z	Objective function
γ_1	Skewness; third standardized moment
γ_2	Kurtosis; fourth standardized moment
μ, $E[]$	Mean Value
ξ	Standard distributed Gaussian random variable
σ	Standard deviation
σ^2, $Var[]$	Variance

SUMMARY

The goal when applying robust engineering design methods is to improve a system's quality by reducing its sensitivity to uncertainty that has influence on the performance of the product. In the Robust Concept Exploration Method (RCEM) this approach is facilitated with additionally giving the designer the possibility to search for a compromise between the desired performance and a satisfying robustness. The current version of the RCEM, however, has some limitations that render it inapplicable for nonlinear design problems. These limitations, which are demonstrated in this thesis, are mainly connected to the application of global response surfaces and the Taylor series for variance estimations.

In order to analyze the limitation of the robustness estimation, several alternative methods are developed, assessed and introduced to a modified RCEM. The developed Multiple Point Method is based on the Sensitivity Index (SI) and improves the variance estimation in RCEM significantly, especially for nonlinear problems. This approach is applicable to design problems, for which the performance functions are known explicitly.

For problems that require simulations for the performance estimation, the simulation-based RCEM is developed by introducing the Probabilistic Collocation Method (PCM) to robust concept exploration. The PCM is a surrogate model approach, which generates local response models around the points of interests with a minimum number of simulation runs. Those models are utilized in the modified-RCEM for the uncertainty analysis of the system's performance.

The proposed methods are tested with two examples each. The modified RCEM is validated with an artificial design problem as well as the design of a robust pressure vessel. The simulation-based RCEM is validated using the same artificial design problem and the design of a robust multifunctional Linear Cellular Alloy (LCA) heat exchanger for lightweight applications such as mobile computing. The structure of the theoretical and empirical validation of the methods follows the validation square.

CHAPTER 1

FOUNDATIONS FOR ROBUST CONCEPT EXPLORATION

The principal goal for this thesis is to:

Further develop the Robust Concept Exploration Method (RCEM) to facilitate the rapid exploration of nonlinear design spaces and the generation of robust design solutions.

The motivation for conducting the research presented in this thesis is the need for a design method that can be applied in early stages of designing robust engineering systems. This method has to have the capability of helping designers when exploring a nonlinear, multi-dimensional and multi-functional design space. Thereby, the method should facilitate robust design, which means quality improvement through reducing the system's sensibility towards noise factors and uncertainty in design variables.

The research presented is based on the development of the original RCEM (Chen, 1995), which facilitates most of the above mentioned requirements but has two crucial limitations. These limitations can lead to fatal errors in nonlinear, multi-modular problems. It is demonstrated that those problems can be solved with the modified RCEM and the simulation-based RCEM, which are developed in this thesis.

One of the core steps in a robust design method is the evaluation of the robustness of a system at multiple points in the design space. Like in most methods, in RCEM the robustness is measured with the variance of the performance, however, a questionable estimation technique is used.

By investigating the robustness estimation method used in the original RCEM, the first-order Taylor Series approximation, the first limitation is addressed. Alternatives for the variance estimation are developed, assessed and introduced in a modified RCEM. This method is validated with an artificial design problem and with the design of a robust pressure vessel.

With the Probabilistic Collocation Method (PCM) (Tatang, 1995) in the simulation-based RCEM an alternative to the Response Surface Methodology (RSM) is introduced to robust concept exploration for problems that require model simulations for the performance assessment. The PCM is a local response model approach based on Gaussian Quadrature Integration and Polynomial Chaos Expansion (PCE) that requires only a minimum number of simulation runs for the generation of a local surrogate model. On this model, the uncertainty analysis is performed and the robustness estimated. The new simulation-based RCEM is validated with an artificial design problem as well as with the design of a robust multi-functional Linear Cellular Alloy (LCA) heat exchanger.

Obviously, the motivation for this research is congruent with the motivation for developing RCEM with the additional goal of improving the original RCEM. In Section 1.1, the motivation is explained in greater detail. In Section 1.2, the systematic engineering design process, robust design and concept exploration as well as the compromise Decision Support Problem technique are introduced as the frame of reference. The research questions and hypotheses that guide the presented work are established in Section 1.3. Finally, the validation strategy as well as the organization of this thesis is outlined in Section 1.4.

1.1 Motivation and Background for Developing Robust Concept Exploration Methods

In this section, the motivation and the background for the presented research are given. This includes the explanation of the need for analysis tools in the early stages of design as well as the need for tools that help designers to handle uncertainty in the concept phase of engineering design. Due to the similarity of the motivation, some parts in Sections 1.1, 1.1.1 and 1.1.2 are leveraged from (Chen, 1995).

As in most areas of industry, productivity is of major economic significance also in engineering design. In order to reduce cost and stay competitive in a fast and globalized economy, a short time-to-market is desired and heavily influenced by decisions in the early stages of design. This requires that, given the design requirements, alternative systems, options or solutions can be evaluated and compared quickly in the concept exploration stage. The introduction of rigorous analysis tools to this design phase promises fewer design changes in later design stages thus reduced cost and shorter time-to-market. This effect is schematically demonstrated in Figure 1.1. This assumption is made on the observation that in original design most analysis tool are used by specialists in individual disciplines when solving parts of the design problem in later stages of the design process. Although, some analysis tools are also applied in the early stages, they are still dominated by personal experience and historical data. By introducing rigorous analysis tool to this design phase it is assumed that the comprehensiveness and fidelity of design and concept analysis can be significantly improved and thereby the number of later design changes reduced and time-to-market shortened.

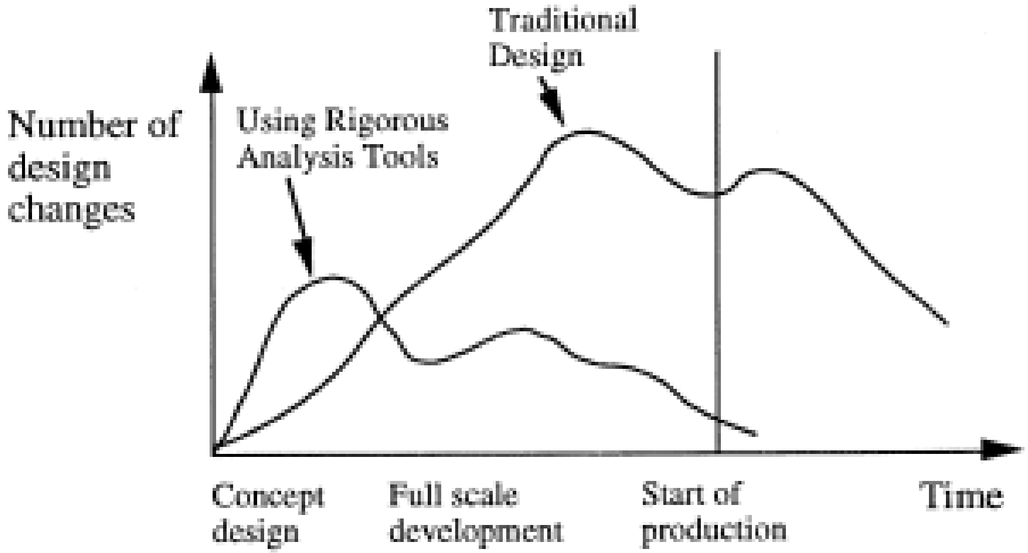


Figure 1.1: Advantages of Rigorous Analysis Tools in Conceptual Design (Chen, 1995)

Such an analysis tool is the Robust Concept Exploration Method (RCEM) (Chen, 1995; Chen, et al., 1996a; Chen, et al., 1997). The framework of this method has been applied to the design of product families (Simpson, 1998; Simpson, et al., 1996, 1999a; Simpson, et al., 1999b) and also in the field of materials design (Mistree, et al., 2002; Seepersad, 2004). In addition to the exploration of concepts, the RCEM enables the designer to improve the product's quality by reducing its sensitivity towards uncertainty in design variables or noise factors. From previous applications, it can be seen that the RCEM is an easy-to-apply tool for the designer that has the potential to improve the quality and shorten the time-to-market of engineering products. However, as presented in this thesis, there is the need for improvement of the applicability of RCEM to nonlinear problems.

In the following sections, the context of the RCEM is introduced. Engineering design process and the different design phases are introduced in Section 1.1.1, followed by a brief explanation of uncertainty in engineering design in Section 1.1.2. Furthermore, a brief introduction to the concept of robust design is given in this section.

1.1.1 Engineering Design Processes

As stated before, the methods proposed in this thesis are meant to be applied in the early stages of the engineering design process. In this section a short overview is given on how design processes are structured and where the RCEM is used.

In the engineering design community, various theories and methodologies have been developed for describing the engineering design process. In (Finger and Dixon, 1989a) and (Finger and Dixon, 1989b), taxonomy distinctions among design methods are provided. Based on observing how designers go about their work, they distinguish between:

- Descriptive models of design processes
- Prescriptive models of design processes
- Computer-based models of design processes

A comprehensive review of works in all three categories is given in (Mistree, et al., 1990). A brief overview over the most important achievements of the last forty years of design research is presented in (Cross, 2007). Although the approaches in the different fields of research show significant differences, there are some models that are widely accepted because they make intuitive sense to many designers. An example of such a process is the systematic approach to engineering design presented in (Pahl and Beitz, 1996). This prescriptive design process model was originally introduced by Pahl and Beitz in 1977 in Germany. It consists of the following four major design phases:

- ***Clarification of the Task*** – collecting information about the design problem and formulating the requirements and constraints to be embodied in the design solution

- ***Conceptual Design*** – establishing function structures and searching for suitable working principles and their combination into concept variants. Conceptual design also includes the development of those concepts and finally, the selection of the most promising one.

- ***Embodiment Design*** – starting from the concept, developing and determining the layout and form of the product under consideration of the requirements as well as technical and economic constraints.

- ***Detail Design*** – specifying and documenting the all the details of the final design as well as generating manufacturing drawings.

As mentioned before, the RCEM is developed for assisting designers in the early stages of the design process. In context of the four phases of the systematic approach from Pahl and Beitz, this would be the conceptual design phase. Based on the requirements and constraints established in the first phase, RCEM is meant to help the designer exploring a large design space of possible concepts in the search for a feasible and satisfying solution. Furthermore, RCEM can be used to find solutions that are robust to various sources of uncertainty.

The types of uncertainty that an engineering designer has to deal with are introduced in the following section. Additionally, a brief introduction to the concept of robust design is given.

1.1.2 Uncertainty in Engineering Design and the Concept of Robust Design

Designers are facing various sources of uncertainty when designing an engineering system. The most common uncertainties are variations in factors the designer can control, namely, design variables, and uncertainty from uncontrollable noise factors. According to this distinction, in (Chen, et al., 1996b) two broad categories of problems are defined as:

- *Type I Robust Design – minimizing variations in performance caused by variations in noise factors (uncontrollable parameters)*
- *Type II Robust Design – minimizing variations in performance caused by variations in control factors (design variables*

Type I variation are, for example, variations in environmental conditions of the system, like temperature changes or humidity that cannot be influenced. Type II variations can be, for example, manufacturing tolerances of design variables. The goal when applying robust design is to find a solution with satisfying performance while reducing the variations caused by uncertainty. An illustration of the difference between an "optimal" solution found with traditional optimization techniques and a robust solution is given in

Figure 1.2 and Figure 1.3. While under optimization an optimum for the specific situation is desired, in robust design each situation is left somewhat open to account for the possibility of deviations in design variables and noise factors.

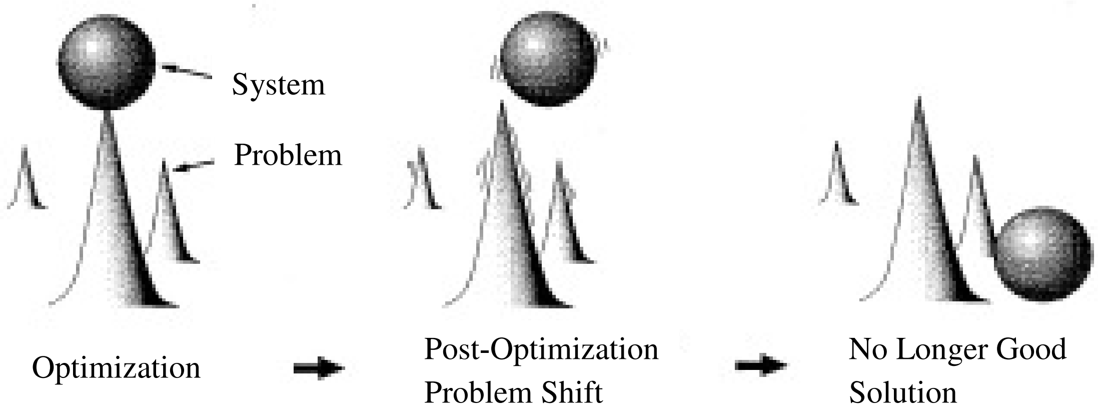


Figure 1.2: Uncertainty Influence on an Optimal Solution (Chen, 1995)

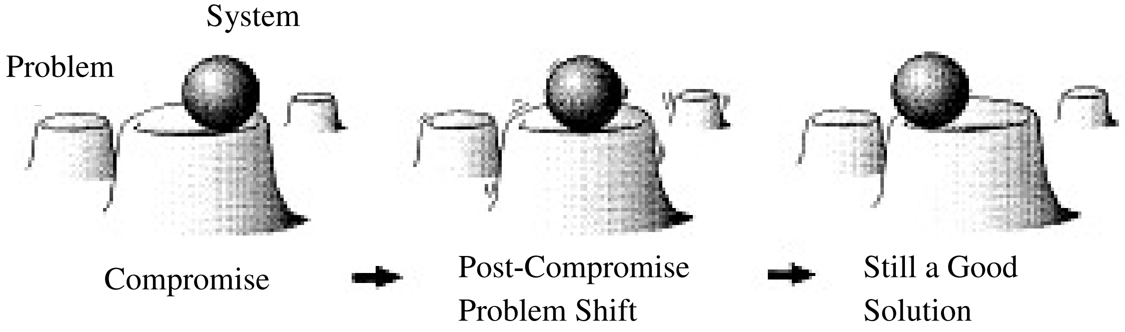


Figure 1.3: Uncertainty Influence on a Robust Solution (Chen, 1995)

In Figure 1.2, it can be seen that an optimal solution is, in general, a peak solution. If in such a case either the problem is shifted slightly or the solution shifts due to uncertainty, the quality of the solution might drop. In Figure 1.3, in contrast, the compromise solution is robust against those shifts and the solution remains good even under the influence of uncertainty.

In this thesis, a robust solution is defined as a compromise between the objective of a satisfying performance – *bringing the mean on target* – and the objective of robustness of the performance – *minimizing the variance*.

In this section, the motivation for developing robust concept exploration methods is presented. In the next section, the frame of reference for this thesis is established.

1.2 Frame of Reference

The frame of reference for this thesis is outlined in this section. The fundamental building blocks for the methods developed in this thesis are robust design methods, especially the Robust Concept Exploration Method (RCEM), and multi-objective decision support, particularly the compromise Decision Support Problem. Since the same components as addressed in (Seepersad, 2004), Section 1.2.1 and Section 1.2.2 of this frame of reference are leveraged with modifications. In Section 1.2.3 the frame of reference for uncertainty analysis is given.

1.2.1 Robust Design and Multi-Objective Concept Exploration

Robust design means to improve the quality of products and processes through the reduction of their sensitivity to variations, without removing the source of uncertainty

(Taguchi, 1986; Taguchi and Clausing, 1990). Typically, there are two objectives in robust design. The first goal is to minimize the product's performance sensitivity and the second goal is to bring the nominal or mean performance as close to the target value as possible; this can be minimizing, maximizing or matching a target. In Figure 2.3, the illustration presented in the previous section is brought in context to an example of a deviation function, which measures the deviation from the desired performance mean.

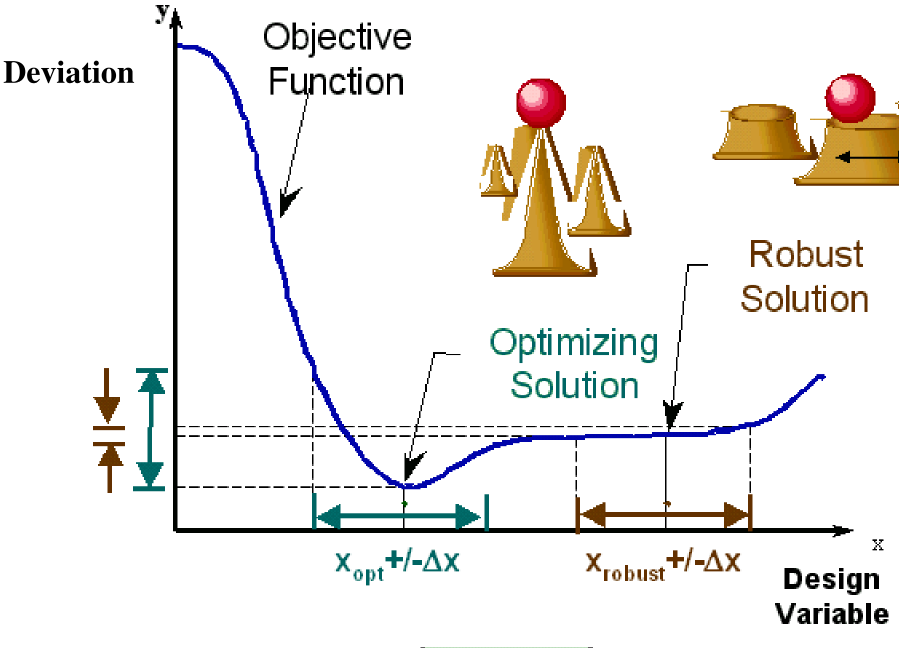


Figure 1.4: Optimal vs. Robust Solutions (Chen, et al., 1996b)

It can be seen that the optimizing solution is significantly more sensitive to deviations of the design variable than the robust solution is. However, the nominal performance of the robust solution is slightly less superior. This is a typical example for a Type II robust design (see Section 1.1.2).

Taguchi’s robust design principles are widely advocated both in industry and in academia. The statistical techniques used by him, however, have been criticized by many researchers. In fact, improving the statistical methodology used in robust design became an active research area (e.g. (Li and Wu, 1999; Myers and Montgomery, 1995; Nair, 1992; Tsui, 1992; Tsui, 1996; Vining and Myers, 1990)). Significant effort was also put into extending the robust design methods for a variety of fields and applications (e.g. (Cagan and Williams, 1993; Chen, et al., 1996a; Chen, et al., 1996b; Otto and Antonsson, 1993; Parkinson, et al., 1993)).

The majority of robust design literature addresses the embodiment and detail design phases of the engineering design process (see Section 1.1.1), where the focus is on adjusting dimensions to accommodate manufacturing variations. However, there has been some emphasis on applying robust design techniques in the earlier stages of design, where decisions are made that heavily influence the final design of a product and thus its quality. Chen and coauthors (Chen, et al., 1996a; Chen, et al., 1996b) propose the Robust Concept Exploration Method (RCEM), a domain-independent approach for generating robust multidisciplinary design solutions. The method is applicable in the early stages of design because it supports the exploration of a large design space, the search for multi-objective compromises as well as the integration of both Type I and Type II robust designs. The framework and the computational infrastructure of RCEM are given in Figure 2.3 and are discussed in greater detail in Chapter 2.

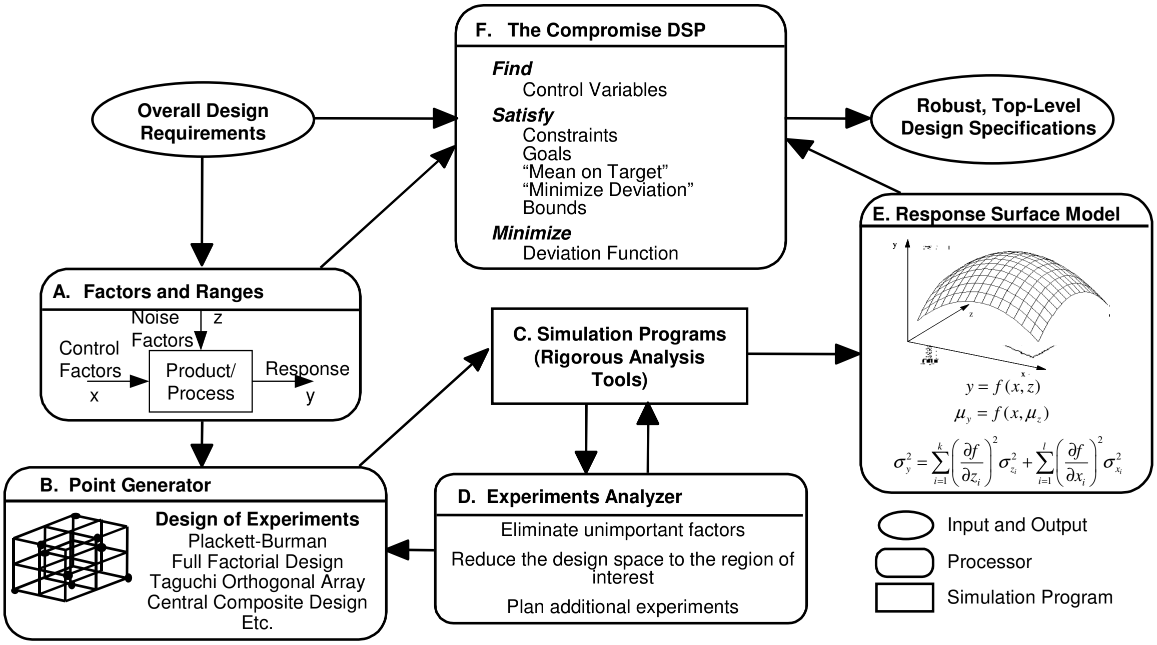


Figure 1.5: Framework of the Robust Concept Exploration Method (Chen, et al., 1996a)

1.2.2 Multi-Objective Decision-Making and the Compromise Decision Support Problem

The robust design methods proposed in this thesis are developed for multi-disciplinary applications and thus require means for multi-objective decision-making. This characteristic is also needed because the robust design approach itself has at least two objectives, first, meeting the desired nominal performance and second, reducing the performance variation.

In this thesis, the same technique is used as in RCEM, the compromise Decision Support Problem (cDSP). This framework provides a means for structuring, mathematically modeling and thus supporting design decisions that involve searching for compromises among conflicting objectives. The cDSP, which has been first introduced by Mistree and coauthors (Mistree, et al., 1993), is a domain-independent, multi-objective decision model that is based on mathematical programming and goal programming. It can be used to find a set of design variable values that satisfy the system's constraints and variable boundaries. Furthermore, a set of (conflicting) objectives is achieved as closely as possible. The designer can express preferences among the objectives by assigning specific weights. The cDSP is discussed in greater detail in Chapter 2.

1.2.3 Uncertainty Analysis and the Probabilistic Collocation Method

In this thesis, the limitations of RCEM (see Section 1.2.1) are demonstrated and means for improvement are introduced. The limitations of RCEM can be located in the application of the Response Surface Methodology for exploring the entire design space and the first-order Taylor Series approximation for the performance variance estimation from the generated surrogate model. In general, it is assumed that the function of the performance is not known explicitly and so the need for some metamodel is given for performing an uncertainty analysis. Since both uncertainty analysis techniques used in RCEM have been criticized in literature (e.g. (Beyer and Sendhoff, 2007; Lin, et al., 1999)), the focus in this thesis is on finding alternative techniques. The analysis of the limitations and development of alternatives is divided into two parts. First, the first-order Taylor Series approximation for the variance estimation is analyzed without considering the surrogate model. The new estimation methods introduced and assessed in this thesis are based on higher-order Taylor Series approximations (Phadke, 1989), the Sensitivity Index SI (Sundaresan, et al., 1995) and first-order approximations at multiple points (Lin, et al., 1999).

As an alternative for building global response surfaces, in this thesis, the Probabilistic Collocation Method (PCM) is considered (Tatang, 1995; Tatang, et al., 1997). The PCM is a local response model approach, which means that the response is only evaluated locally around a point of interest. The response model is built with a minimum number of simulation runs and thus is very efficient. The method has been previously applied to forecast and analyze future carbon emissions (Webster, 1997), numerical geophysical models (Tatang, et al., 1997), a simple ocean model (Webster, et al., 1996), the evaluation of uncertainty in dynamic simulations of power system models (Hockenberry and Lesieutre, 2004) and the analysis of flow in random porous media (Li and Zhang, 2007). In those applications the method has been found very efficient and effective for estimating the statistical moments and the probability density function of the systems' performance under uncertainty. In this thesis, the Probabilistic Collocation Method is introduced to Robust Engineering Design. The PCM is discussed in greater detail in Chapter 2.

In this section, the building blocks of this thesis are introduced. In the next section, the research questions and hypotheses are presented and discussed.

1.3 Research Focus and Contributions

The motivation and background for this thesis are presented in Section 1.1 and Section 1.2. The primary focus in this section is on discussing the research questions addressed in this thesis and outlining the contributions expected from this research. The two questions that guided the presented research are answered by validating two research hypotheses, which are introduced in this chapter as well.

The fundamental research questions and hypotheses are presented in Section 1.3.1. In Section 1.3.2, a set of contributions is presented that summarize the value of this research. In Section 1.4, the strategy for validating the research hypotheses and establishing the contributions is outlined.

1.3.1 Research Questions and Hypotheses

The principal goal for this thesis is to:

> Further develop the Robust Concept Exploration Method (RCEM) to facilitate the rapid exploration of nonlinear design spaces and the generation of robust design solutions.

To improve the original RCEM, its limitations are addressed in the research questions. As the first limitation, the first-order Taylor Series approximation for the performance variance estimation is identified. The first research question is motivated by this research gap:

> First Research Question:
>
> How can the formulation of robustness in compromise Decision Support Problems for multi-dimensional and multi-objective design problems be improved?

To answer this question, a set of alternative formulations for robustness estimation is tested and evaluated. Based on this analysis the first research hypothesis is validated:

First Research Hypothesis:

Using a global response surface, the nominal mean and a single point first order Taylor Series Approximation for the variance estimation is not accurate enough for finding robust solutions of nonlinear design problems. The Robust Concept Exploration Method can be improved by using appropriate formulations that account for the nonlinearity of the problem.

For the validation of this hypothesis the modified RCEM is introduced, in which a new variance estimation technique is implemented. Since the second limitation is not eliminated by this, an alternative for the Response Surface Methodology is required. This need is addressed in the second research question:

Second Research Question:

How can the RCEM be augmented to improve the accuracy of the robust concept exploration for nonlinear design problems that involve simulations?

For answering this question the Probabilistic Collocation Method is introduced to RCEM. This new method, the simulation-based RCEM, is used for the validation of the second research hypothesis:

Second Research Hypothesis:

It is hypothesized that estimating the performance and the robustness using the Probabilistic Collocation Method with Polynomial Chaos Expansion in a simulation-based approach to robust concept exploration can increase the method's accuracy and reliability over the original RCEM.

With answering the research questions and validating the hypotheses several contributions are made, which are summarized in the next section.

1.3.2 Research Contributions

In this thesis, the research contributions are made by justifying the research questions and by validating the hypotheses.

The first research contribution is the demonstration of the limitations of the original RCEM. In literature it has been claimed that RCEM can be applied to solve nonlinear design problems (e.g. (Chen, et al., 1996b)). In this thesis, it is demonstrated that RCEM can lead to fatal errors when applied to nonlinear problems. This is considered an important contribution, since several researchers have used and developed RCEM without noticing this crucial problem.

The second contribution is the introduction of the Probabilistic Collocation Method to robust concept exploration. The new simulation-based RCEM, which is used to validate the second research hypothesis, improves the accuracy, efficiency and reliability of the original RCEM. Additionally, it opens up new opportunities for a detailed uncertainty analysis of the performance and the consideration of alternative distributions of design variables and noise factors.

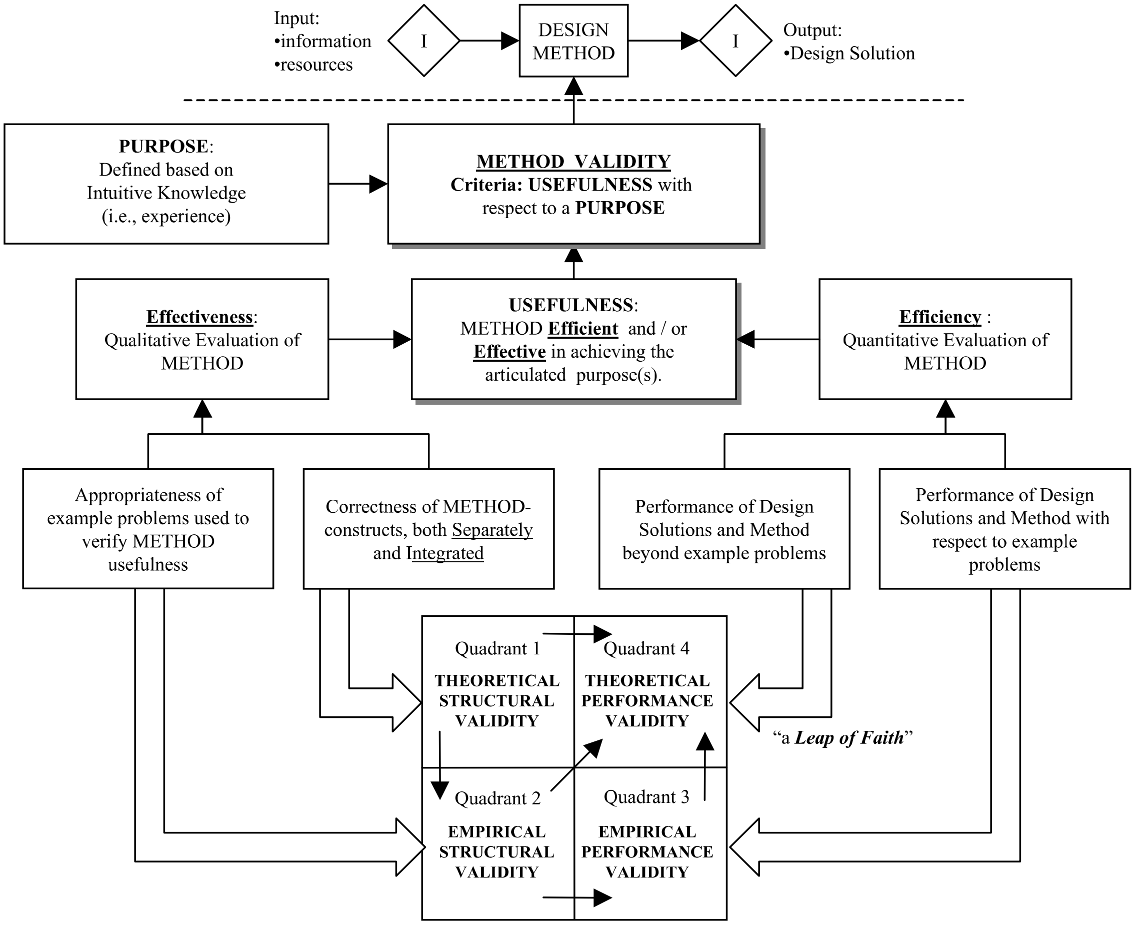
The third contribution is the establishment of a new computational framework for all versions of RCEM. The framework consists of a Java version of DSIDES, which can be used to solve cDSPs, and the integration of MATLAB functions for simulation-based problems or brute force uncertainty analyses like Monte Carlo.

In this section, the research focus and the contributions are summarized. The strategy for answering the research questions, validating the hypotheses and establishing the contributions is presented in the next section.

1.4 Validation and Verification of this Thesis

The validation and verification strategy for this thesis is based on the validation square, which was introduced for the validation of design methods (Pedersen, et al., 2000; Seepersad, et al., 2006b). The following explanation of the validation square and the strategy for the validation and verification of this thesis are inspired by Carolyn Seepersad's PhD dissertation (Seepersad, 2004). Typically, engineering research is based on formal, quantitative validation through logical induction and/or deduction. However, this approach is problematic for the validation of engineering design methods. As soon as a method is not solely based on mathematical modeling but also on subjective statements, an alternative to logical induction and/or deduction is needed. With the validation square, an approach to the validation of engineering design methods is proposed, which is based on a relativistic notion of epistemology in which "knowledge validation becomes a process of building confidence in its usefulness with respect to a purpose."

The validation square is a framework for validating design methods in which the 'usefulness' of a design method is associated with whether the method provides design solutions correctly (structural validity) and whether it provides correct design solutions (performance validity). This process is illustrated in Figure 1.6.



Input:
•information
•resources
I
DESIGN METHOD
I
Output:
•Design Solution
PURPOSE: Defined based on Intuitive Knowledge (i.e., experience)
METHOD VALIDITY
Criteria: USEFULNESS with respect to a PURPOSE
Effectiveness: Qualitative Evaluation of METHOD
USEFULNESS: METHOD Efficient and / or Effective in achieving the articulated purpose(s).
Efficiency : Quantitative Evaluation of METHOD
Appropriateness of example problems used to verify METHOD usefulness
Correctness of METHOD-constructs, both Separately and Integrated
Performance of Design Solutions and Method beyond example problems
Performance of Design Solutions and Method with respect to example problems
Quadrant 1
THEORETICAL STRUCTURAL VALIDITY
Quadrant 4
THEORETICAL PERFORMANCE VALIDITY
"a Leap of Faith"
Quadrant 2
EMPIRICAL STRUCTURAL VALIDITY
Quadrant 3
EMPIRICAL PERFORMANCE VALIDITY

Figure 1.6: The Validation Square (Pederson, et al., 2000)

Both structural and performance validity are further divided into theoretical and empirical validity which leads to the four quadrants:

1.4.1 Quadrant 1: Theoretical Structural Validity

Theoretical structural validity involves showing the internal consistency of the individual constructs constituting the method as well as showing the overall internal consistency of their assembly. This can be achieved by searching and referencing to literature related to the single constructs, which are already validated elsewhere. Furthermore, the correctness of the information flow throughout the entire design method has to be demonstrated. For this step a flow chart may be useful. To ease the comparison of the theoretical structure and the expected outcomes to the intended properties of the design method, a requirements list should be formulated.

1.4.2 Quadrant 2: Empirical Structural Validity

Empirical structural validity involves building confidence in the appropriateness of the chosen example problems for illustrating and verifying the performance of the developed design method. This means, it has to be shown that the examples are good representations of design problems, for which the method is designed and that the associated data can be used to support a conclusion.

1.4.3 Quadrant 3: Empirical Performance Validity

Empirical performance validity includes showing the usefulness of the method for solving the example problems. The results achieved using the design method have to be analyzed and assessed. Measurements of the usefulness should be related to the desired specifications, which are formulated in the requirements list. Furthermore it has to be shown that the achieved usefulness is, in fact, a result of applying the developed method, for example by comparing the generated outcomes to solutions acquired without the method. The analysis should also include assessment of data with regard to internal consistency, for example multiple starting points and convergence in optimization exercises.

1.4.4 Quadrant 4: Theoretical Performance Validity

Theoretical performance validity involves a “leap of faith” from the usefulness of the design method for the chosen example problems to the general validity of the method, which means building confidence in the generality of the method and accepting that the method is useful beyond the example problems. This can be supported by showing that the example problems are representative for a general class of engineering design problems as well as a final critical analysis of the entire validation process.

1.4.5 Implementation of the Validation Framework in this Thesis

The validation square is a framework that suggests a logical step by step approach for the validation of a design method. Successfully building confidence for the validities in quadrants one, two, and three followed by a critical analysis allows for the “leap of faith” in quadrant four. This “leap of faith” to the general validity of the developed method is characteristic for the validation of design methods that include subjective statements. Since this is the case for the methods presented in this thesis, the validation square is an appropriate framework for the validation and verification of the presented work.

In Figure 1.7, the outline of the validation strategy is presented with regard to the four quadrants of the validation square. The tasks for each step are given with reference to the chapter in which they are addressed.

In the flowchart provided in Figure 1.8, the organization of this thesis is illustrated. The intention behind this specific structure is the implementation of the validation strategy outlined in Figure 1.7.

Quadrant 1: Theoretical Structural Validation

• Critical review of literature relating to the components that are foundational for the methods proposed in this thesis. • What are the characteristics, limitations and domains of application of the building blocks? Are the underlying assumptions compatible? How do the individual components relate to the research questions and hypotheses?	Chapter 2
• Presentation and discussion of the improvements to RCEM and showing the relation to the first research question and the first hypothesis. • What are the characteristics, limitations and domains of application of the new method? Does it represent the first hypothesis proposed in Chapter 1?	Chapter 3
• Presentation and discussion of the new robust decision support method and showing its relation to the second research question and second hypothesis. • What are the characteristics, limitations and domains of application of the new method? Does it represent the second hypothesis proposed in Chapter 1?	Chapter 5

Quadrant 2: Empirical Structural Validation

• Discussion of the appropriateness of the example problems chosen in Chapter 4 (Pressure Vessel), which is part of validating the first hypothesis, and Chapter 6 (Linear Cellular Alloy), which is related to the second hypothesis: o Argue that the examples are representative for engineering design problems in the domain the methods are designed for. o Document that the method theoretically is applicable to the chosen examples. o Show that the expected data and results can support a conclusion whether the hypotheses are valid or not.	Chapter 4 & Chapter 6

Quadrant 3: Empirical Performance Validation

• Building confidence in the utility of the proposed methods using the example problems: o Demonstrate that the method is working as designed (converges, makes compromises, results in robust points) o Show that the observed utility is actually linked to the application of the new method by applying an alternative method and comparing the performance o Verify that the results are correct by comparing to Latin hypercube samplings	Chapter 4 & Chapter 6

Quadrant 4: Theoretical Performance Validation

• Building confidence in the general applicability of the methods beyond the examples used in this thesis.	Chapter 7

Figure 1.7: Validation Strategy for this Thesis with Associated Tasks and Questions

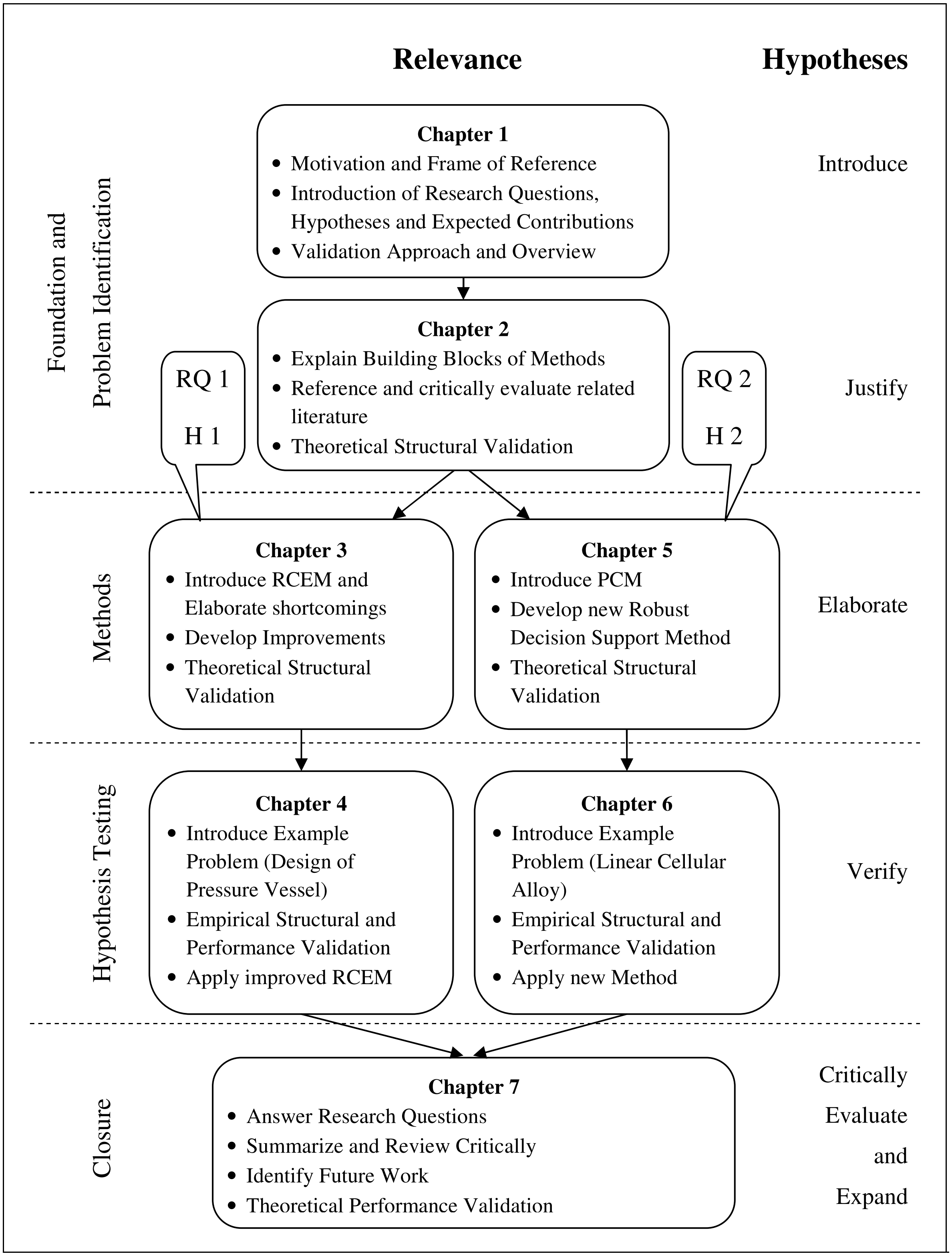

Relevance
Hypotheses
Foundation and Problem Identification
Chapter 1
Motivation and Frame of Reference
Introduction of Research Questions, Hypotheses and Expected Contributions
Validation Approach and Overview
Introduce
Chapter 2
Explain Building Blocks of Methods
Reference and critically evaluate related literature
Theoretical Structural Validation
RQ 1
H 1
RQ 2
H 2
Justify
Methods
Chapter 3
Introduce RCEM and Elaborate shortcomings
Develop Improvements
Theoretical Structural Validation
Chapter 5
Introduce PCM
Develop new Robust Decision Support Method
Theoretical Structural Validation
Elaborate
Hypothesis Testing
Chapter 4
Introduce Example Problem (Design of Pressure Vessel)
Empirical Structural and Performance Validation
Apply improved RCEM
Chapter 6
Introduce Example Problem (Linear Cellular Alloy)
Empirical Structural and Performance Validation
Apply new Method
Verify
Closure
Chapter 7
Answer Research Questions
Summarize and Review Critically
Identify Future Work
Theoretical Performance Validation
Critically Evaluate and Expand

Figure 1.8: Thesis Overview and Roadmap

In **Chapter 1,** the foundation of this thesis is built with the presentation of the motivation and the frame of reference. The research questions and hypotheses are introduced and explained and the expected contribution is summarized. Based on the validation strategy, the outline is developed and a brief overview for each chapter is given.

In **Chapter 2**, the theoretical foundations for the methods developed in this thesis are explained and discussed. These foundations include robust design, meta-modeling techniques, multi-objective decision support problems and stochastic analysis. Understanding the foundations and underlying theories is crucial to fully understand the research questions and hypotheses proposed in this thesis. In order to support the theoretical structural validity of the introduced methods, relevant literature for the building blocks is referenced, analyzed and critically discussed in this chapter. Showing the internal consistency of the components builds the basis for the theoretical structural validity of the new methods. Additionally, a computational framework is explained that can be used for the application of the RCEM as well as the new methods.

In **Chapter 3**, RCEM is analyzed and discussed. A simple example is demonstrated to explain the development of the first research question. Following the first hypothesis, several possible improvements are introduced, explained and tested on the simple example. For the theoretical structural validity, emphasis is placed on verifying the internal consistency of the new method as well as discussing advantages and limitations.

In **Chapter 4**, an engineering example problem, the design of a pressure vessel, is introduced, which will be used to show the performance of the modified RCEM discussed in Chapter 3. For the empirical structural validity, it is argued that the example represents a typical engineering problem with characteristic assumptions, goals, variables and constraints, which is appropriate to be solved with the new method. For the empirical performance validity, a comparison of the results from the improved method with the original RCEM and a Latin hypercube sampling (LHS) is made. Finally, an outlook is given to build the transition to the second research question.

In **Chapter 5**, the second research question is addressed and a new method is introduced. The components of this method are discussed in detail from the perspective of embodying the second hypothesis presented in Chapter 1. The internal consistency and the applicability of the new method, the simulation-based RCEM, are shown to support the theoretical structural validity. Furthermore, the advantages and limitations of the introduced method are critically discussed. Finally, the computational framework is briefly explained that can be used to apply the new method.

In **Chapter 6**, an engineering example problem, the design of a heat exchanger, is introduced, which is used to validate the method presented in Chapter 5. For empirical structural validation, it is discussed that the example problem is representative for problems, which the method is designed for. By solving the design problem the applicability of the method is demonstrated. For empirical structural validation, the results are compared to Latin hypercube samplings.

In **Chapter 7**, the thesis is summarized and the research questions are answered by discussing the validity of the hypotheses introduced in Chapter 1. Furthermore, the work is critically reviewed and the contributions are identified. Hereby, emphasis is placed on the underlying assumptions and conditions under which the proposed methods work. For the theoretical performance validation, it is argued that the methods presented in this thesis as well as the conclusions are relevant and valid beyond the two example problems. Finally, potential future research topics are identified that could enhance the proposed methods and make them more efficient or applicable to a broader range of engineering design problems.

In the **Appendix**, a manual for JavaDSIDES, including a description of the MATLAB integration is presented. Furthermore, the MATLAB code for the application of the modified RCEM to the pressure vessel example is given and code of the implementation of the simulation-based RCEM is presented.

1.5 Chapter Synopsis

In this chapter, the motivation for the research presented in this thesis is given and the frame of reference is established. After the explanation of the research questions and the hypotheses, the validation strategy is outlined. Besides the structure for the validation, also a thesis roadmap is presented. In the following chapter, a literature review is given and the building blocks of the research and the proposed methods are explained.

CHAPTER 2

ROBUST DESIGN: REVIEW OF LITERATURE AND IDENTIFICATION OF RESEARCH OPPORTUNITIES

In this chapter, the foundations for designing robust systems are investigated. As presented in Chapter 1, the primary goal in this thesis is to improve the original RCEM. Therefore, the fundamental components of RCEM are introduced in this chapter. An explanation of multi-objective decision support is given in Section 2.1. The foundation of RCEM, the robust design paradigm, is discussed in Section 2.2. This includes a description of Taguchi's robust design (Section 2.2.1) as well as a review of improvements and extensions made to this methodology. RCEM is described in detail in Section 2.2.3 with its limitations outlined in Section 2.2.4. In Section 2.3, a brief introduction to uncertainty analysis is given and the Probabilistic Collocation Method is introduced. Finally, the computational framework developed and used in this thesis is presented in Section 2.4.

2.1 Multi-Objective Decision Support and the Decision Support Problem Technique

The core requirement of a design method is that it should assist designers in making decisions. Since these decisions usually depend on more than one objective and often a compromise among conflicting objectives is required, an appropriate framework is needed. Parts of the following description of the compromise DSP are leveraged with modifications from (Seepersad, 2004).

In this thesis, the compromise Decision Support Problem (cDSP) technique (Mistree, et al., 1993; Muster and Mistree, 1988) is used for formulating the objectives of engineering design problems. The cDSP was developed especially for engineering design applications and supports multi-objective decision-making. It is a hybrid form based on mathematical programming and goal programming. In goal programming, target values for all objectives are established and the focus is on achieving each of these goals as closely as possible (Charnes and Cooper, 1977). This idea is borrowed in the cDSP; for each objective an achievement function, $A_i(\mathbf{x})$, is formulated that represents the value of the i^{th} objective as a function of a set of design variables, **x**. The deviation variables, d_i^- and d_i^+, represent the extent to which a target value, G_i, is over- or underachieved:

$$A_i(\mathbf{x}) + d_i^- - d_i^+ = G_i \qquad i = 1,\dots,m \qquad \text{Eq. 2.1}$$

Usually, when applying the cDSP, the goal formulations are normalized to guarantee comparability of the deviations from different objectives. The normalization conventions are explained in detail in (Mistree, et al., 1993). Since under- and overachievement can only be positive and both cannot be achieved simultaneously, which means that at least one of them has to be zero, the following restrictions apply:

$$d_i^-, d_i^+ \geq 0 \qquad i = 1,\dots,m \qquad \text{Eq. 2.2}$$

$$d_i^- \cdot d_i^+ = 0 \qquad i = 1,\dots,m \qquad \text{Eq. 2.3}$$

The overall objective function is formulated as a function of the deviation variables as follows:

$$Z = \underset{i=1,\ldots,m}{f} \left(d_i^-, d_i^+\right) \qquad \text{Eq. 2.4}$$

This function is often called deviation function and could take any form, the simplest of which is the weighted sum formulation:

$$Z = \sum_{i=1}^{m} \left(w_i^- d_i^- + w_i^+ d_i^+\right) \qquad \text{Eq. 2.5}$$

Besides an Archimedean formulation as in Eq. 2.5, also preemptive (lexicographic) formulations are possible (Mistree, et al., 1993).

The disadvantage of goal programming formulations is that they do not support constraints. However, equality and inequality constraints are an integral part of engineering design problems. This is why in the cDSP, this feature is borrowed from mathematical programming and constraints can be formulated as:

$$h_i(x) \geq 0 \qquad i = 1,\ldots, p \qquad \text{Eq. 2.6}$$

$$h_i(x) = 0 \qquad i = p+1,\ldots, p+q \qquad \text{Eq. 2.7}$$

where p and q are the number of inequality and equality constraints, respectively. The bounds on the set of design variables, **x**, are defined as:

$$x_{i,L} \leq x_i \leq x_{i,U} \qquad i = 1,\ldots, n \qquad \text{Eq. 2.8}$$

where n is the number of design variables, by which each solution is defined, and $x_{i,L}$ and $x_{i,U}$ are the lower and upper bound of the i^{th} variable.

Given

An alternative to be improved through modification

Assumptions used to model the domain of interest

The system parameters:

n number of system variables

$p+q$ number of system constraints

p inequality constraints

q equality constraints

m number of system goals

$C_i(\mathbf{x})$ Capability of the system

$D_i(\mathbf{x})$ Demand of the system

$h_i(\mathbf{x})$ System constraint function

$h_i(\mathbf{x}) = C_i(\mathbf{x}) - D_i(\mathbf{x})$

Find

x_i System variables $i = 1, \ldots, n$

d_i^-, d_i^+ Deviation Variables $i = 1, \ldots, m$

Satisfy

System constraints (linear, nonlinear):

$h_i(\mathbf{x}) \geq 0$ $\quad$ $\mathrm{i} = 1, \ldots, p$

$h_i(\mathbf{x}) = 0$ $\quad$ $i = p+1, \ldots, p+q$

System goals:

$A_i(\mathbf{x}) + d_i^- - d_i^+ = G_i$ $\quad$ $i = 1, \ldots, m$

Bounds

$x_{i,L} \leq x_i \leq x_{i,U}$ $\quad$ $i = 1, \ldots, n$

$d_i^-, d_i^+ \geq 0$ $\quad$ $i = 1, \ldots, m$

$d_i^- \cdot d_i^+ = 0$ $\quad$ $i = 1, \ldots, m$

Minimize

Deviation function: Archimedean formulation

$$Z = \sum_{i=1}^{m} \left(w_i^- d_i^- + w_i^+ d_i^+ \right) \qquad i = 1, \ldots, m$$

Figure 2.1: The Framework of the Compromise DSP (modified from (Bras and Mistree, 1993))

The formulations from goal programming and mathematical programming combined with the additional constructs of the cDSP are presented in Figure 2.1. The cDSP is used to find a set of design variables that satisfy the bounds and constraints of the problem, while achieving a set of (conflicting) objectives as closely as possible. Through the combination of components from both goal and mathematical programming, the cDSP is a flexible framework; it accommodates multiple constraints and objectives, as well as both quantitative information and information that may be based on the designer's judgment and experience (Marston, et al., 2000). The segmented structure of the cDSP (Given, Find, Satisfy, Minimize) guides the designer through the collection of data. By following the structure, the designer is less likely to miss or forget important information when formulating the design problem.

The compromise DSP has been utilized for designing various types of engineering systems. Especially the overall objective function has been a research focus and was modified to facilitate physical programming (Hernandez, et al., 2001), Bayesian (Vadde, et al., 1994), fuzzy (Zhou, et al., 1992) and utility-theory formulations (Seepersad, 2001) for specific contexts. Since the compromise DSP has been validated extensively in literature, the contribution in this thesis is limited to incorporating the cDSP into the modified RCEM and the simulation-based RCEM as the foundational framework.

In this section, the details of the compromise DSP are explained. In the next section the foundations of robust design are reviewed and the Robust Concept Exploration Method (RCEM), which also utilizes the cDSP, is introduced.

2.2 Robust Design and Robust Concept Exploration

"Robustness is the state where technology, product, or process performance is minimally sensitive to factors causing variability (either in the manufacturing or user's environment) and aging at the lowest unit manufacturing cost" (Taguchi, et al., 2000). The foundations for robust design are based on the philosophy of Genichi Taguchi, a Japanese industrial consultant, who introduced his concept in the mid 80s (Taguchi, 1986; Taguchi and Clausing, 1990). In 1993, Box defines said that "Robustifying a product is the process of defining its specifications to minimize the product's sensitivity to variation" (Box and Fung, 1993). Suh defines robust design as "the design that satisfies the functional requirements even though the design parameters and the process variables have large tolerances for ease of manufacturing and assembly" (Suh, 2001). Although the formulations are slightly different, their core notion is the same: increasing robustness means reducing performance variability.

The robust design philosophy is the core component of the research presented in this thesis. Therefore it is important to understand the underlying theory and the different concepts introduced in academia. In this Section, the fundamentals of robust design are presented. In Section 2.2.1, Taguchi's robust design approach is explained briefly. This philosophy got significant industrial and academic attention resulting in many proposed improvements of the Taguchi approach. These critiques, improvements and extensions are reviewed in Section 2.2.2. The Robust Concept Exploration Method (RCEM), which the research presented is based on, is described in Section 2.2.3. Finally, in Section 2.2.4, the limitations of RCEM are outlined as the research opportunities addressed in this thesis.

2.2.1 Taguchi's Robust Design

"Robust design is a method for improving the quality of products and processes by reducing their sensitivity to variations, thereby, reducing the effects of variability without removing its sources" (Seepersad, 2004). This means that a robust product, if exposed to uncertainty, remains a constant level of quality and performance. This design philosophy is established by Genichi Taguchi, a Japanese industrial consultant. His robust design approach is a two-step process. It is subdivided in parameter design and tolerance design. In parameter design, the goal is to find a robust region in the design space without modifying or eliminating the sources of uncertainty or variation. In tolerance design, the variations are analyzed and tolerances are adjusted. For this thesis, only the parameter design is of importance, since the variations are assumed to be not controllable. Taguchi's parameter design is explained in the following.

Taguchi's approach is based on the assumption that deviation from the intended performance of a product or process can be expressed monetarily. Supposed *y* is the quality or performance characteristic of a product or process and *T* its target value defined by the designer, Taguchi defines a quadratic loss function as:

$$L(y) = k(y - T)^2 \qquad \text{Eq. 2.9}$$

This function is a measure for the quality loss due to the performance deviation, $|y\text{-}T|$. The main advantage of this function is that it allows the decomposition shown in Eq. 2.10 when calculating the average or expected quality loss *Q*.

$$Q(y) = \mathrm{E}\left[k(y - T)^2\right] = k\left[(\mu_y - T)^2 + \sigma_y^2\right] \qquad \text{Eq. 2.10}$$

In this equation μ_y is the mean and σ_y^2 is the variance of *y* and E[ ] is the expectation. The goal is to minimize Eq. 2.10, which is equivalent to maximizing the signal-to-noise factor, a ratio that includes the performance mean and its variance, introduced by Taguchi. For a brief explanation of the formulations of the signal-to-noise ratios it is referred to (Park, et al., 2006). While this step, the parameter design, does not increase the cost, tolerance design is expensive through the cost associated with tightening the tolerance. This is the reason why Taguchi developed a two step approach. According to his philosophy, tolerance design should only be utilized when robustness cannot be designed into a product or process with parameter design. In Figure 2.2, the probability density functions of three different designs are illustrated in comparison to the quality loss function. In this scenario, design A is preferred over design B due to the smaller deviation of its mean from the performance target and over C due to its smaller variance.

When analyzing the design space, Taguchi clearly separates control factors, which are design variables that can be controlled, from noise factors, which are difficult or impossible to control. This is the basis for the definition of Type I and Type II robust designs by (Chen, et al., 1996b) (see Section 1.1.2).

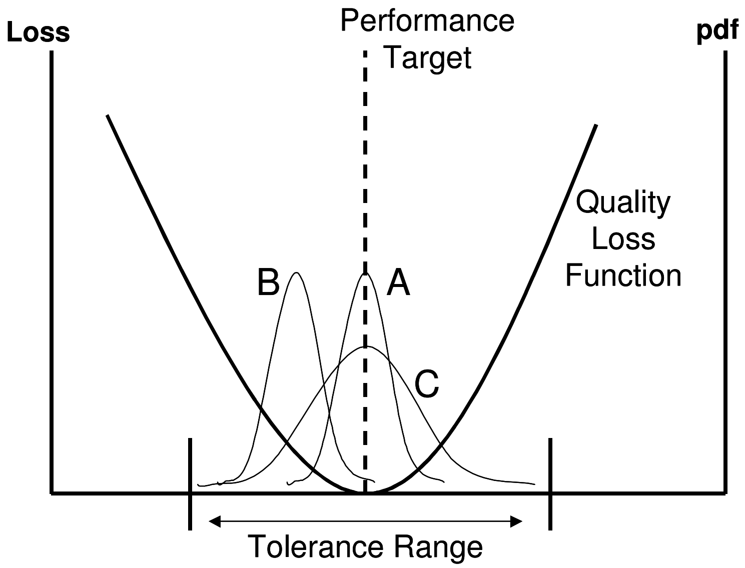


Figure 2.2: Taguchi's Quality Loss Function (Seepersad, 2004)

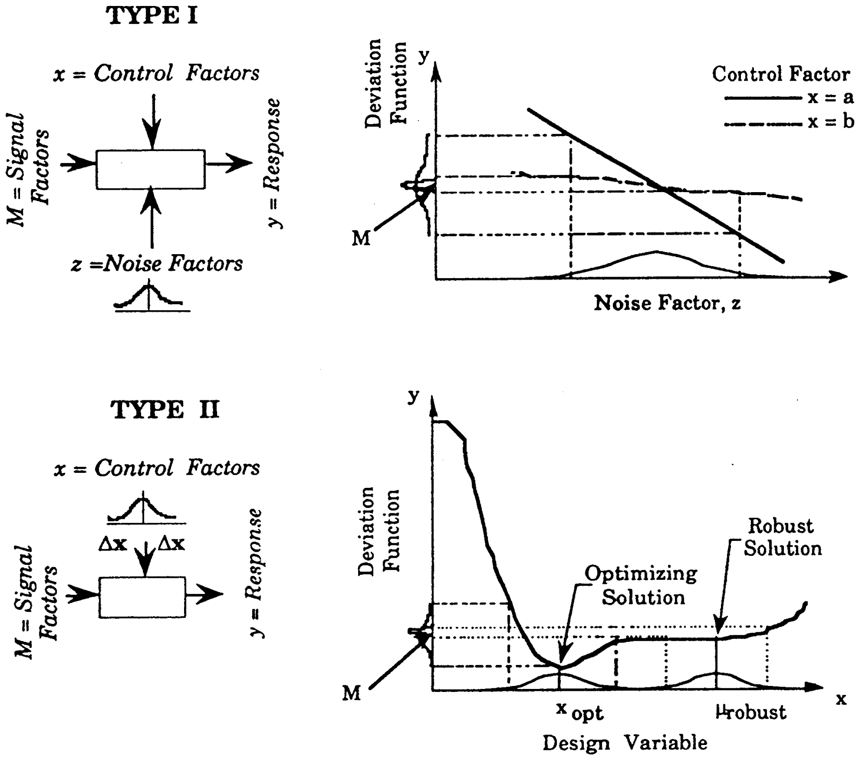


Figure 2.3: Robust Design for Variations in Noise and Control Factors (Chen, et al., 1996b)

Finding a robust solution means to take advantage of nonlinear relationships defining the performance. This is illustrated in Figure 2.3. It can be seen, that control factors are chosen that the performance variation caused by the variations of control *and* noise factors are minimal. Hereby, the robust solution is usually different from the optimal solution, meaning that the optimal solution is rarely robust.

In this section, Taguchi's approach to robust design is introduced. Certainly, Taguchi initiated a paradigm shift in engineering design. Although he received a lot of credit for this development, academia and industry expressed concern and criticism on his approach and have been constantly trying to improve robust design and make it applicable for a variety of fields and problems. A review of the critiques and further developments is given in the next section.

2.2.2 Limitations and Improvements of Taguchi's Robust Design Methodology

In this section, the criticism on Taguchi's robust design approach is reviewed and some improvements and further developments are presented. This section is leveraged with minor modifications from (Allen, et al., 2006).

Although Taguchi's robust design principles are advocated widely in industrial and academic settings, his statistical techniques, including orthogonal arrays and the signal-to-noise ratio, have been criticized extensively. Especially improving the statistical methodology for robust design has been an active area of research (Myers and Montgomery, 1995; Nair, 1992; Tsui, 1992; Tsui, 1996). In the panel discussion reported by Nair (Nair, 1992), practitioners and researchers discuss Taguchi's robust design methodology, the underlying engineering principles and philosophy, and alternative statistical techniques for implementing it. For example, many alternative experimental designs have been proposed (Giunta, et al., 1997; Lin, 2004; Santner, et al., 2003; Wu and Hamada, 2000), and a significant point of debate and scholarly research has been dedicated to the comparative advantages of Taguchi's cross arrays (in which control factors and noise factors are varied according to separate plans) versus combined arrays (in which control and noise factors are varied jointly according to a single plan). A key advantage of the combined array approach is that it provides flexibility for the designer to rule out certain effects a priori and thereby accomplish computational savings (see e.g. (Borror and Montgomery, 2000; Shoemaker, et al., 1991; Welch, et al., 1990)). Despite the convincing theoretical case for combined arrays, empirical studies suggest that cross arrays provide superior outcomes under a wide range of conditions (Frey and Li, 2004). In (Shoemaker, et al., 1991) it is noted that combined arrays are successful to the extent that the model fits well.

It has been demonstrated that using the signal-to-noise ratio as the objective in robust design can hide information about the effects of noise (Box, 1988; Shoemaker, et al., 1991). Nair (Nair, 1992) also reports that a panel of statisticians suggest independently modeling the mean response and variability directly or via statistical data transformations (Box, 1988; Tsui, 1992; Vining and Myers, 1990), rather than modeling the signal-to-noise ratio - a practice that discards useful information about the response particularly by confounding mean response with variance information.

In (Chen, et al., 1996b) and (Bras and Mistree, 1993), robust design problems are formulated as multi-objective decisions using the compromise Decision Support Problems (cDSPs) (see Section 2.1). In this approach, control and noise factors are considered as potential sources of variation, and constraints are modeled in a worst-case formulation to ensure feasibility robustness. Separate goals of bringing the mean on target and minimizing variation for each design objective are included in a goal programming formulation of the objective function. This provides flexibility for achieving compromises among multiple performance objectives as well as individual or collective compromises among mean values and variations for all objectives. In more recent work, Chen et al. have extended the approach to include alternative formulations of the objective function, such as compromise programming (Chen, et al., 1999) and physical programming (Chen, et al., 2000).

Researchers have also developed alternative analysis procedures including "response modeling" (Welch, et al., 1990), dual response approaches (Vining and Myers, 1990), and rejection of pre-decided criteria in favor of graphical analysis and discovery (Box, 1988).

Taguchi's robust design principles have also been extended by applying them to simulation-based design. With increasing pressure to cut development costs and reduce development times, robust design is increasingly conducted by evaluating the relationship between input factors and responses using computer simulations rather than prototypes and physical experiments. Given that computer experiments lack pure error, different experimental design and analysis strategies are recommended (Simpson, et al., 2001b). In response to this need, the field of design and analysis of computer experiments has grown rapidly in recent decades providing a variety of useful techniques (Santner, et al., 2003).

Once a robust design problem has been formulated, it must be solved. The computational burden can be significant, particularly for design problems, in which a broad design space must be explored. Solving a robust design problem is distinguished by the need to evaluate not only a nominal value for each response but also the variation of each response due to control or noise factor variation. If a response y is a function of control factors $\mathbf{x}$ and noise factors $\mathbf{z}$, then:

$$y = f(\mathbf{x}, \mathbf{z}) \qquad \text{Eq. 2.11}$$

where the function f can be a detailed simulation model, a surrogate model, or a physical system. The challenge is to estimate the expected value μ_y and variance σ_y^2 of the response. As presented in Section 1.3.1, this challenge is the research focus in this thesis. There are many techniques for transmitting or propagating variation from input factors to responses, and each technique has strengths and limitations. Monte Carlo analysis is a simulation-based approach that requires a very large number of experiments (Liu, 2001). It is typically very accurate for approximating the distribution of a response, provided that probability distributions are available for the input factors. On the other hand, it is very computationally expensive, especially if there are large numbers of variables or if expensive simulations are needed to evaluate each experimental data point. Du and Chen review several approaches for maintaining feasibility robustness and introduce a most probable point (MPP) based approach that offers accuracy similar to Monte Carlo based approaches with fewer computations (Du and Chen, 2000). If only a moderate number of experimental points are affordable, a variety of space-filling experimental designs are available, and experimental designs such as fractional factorials or orthogonal arrays can be used (Myers and Montgomery, 1995; Wu and Hamada, 2000). These experimental designs provide adequate estimates of the range of the response rather than its distribution, and they require fewer experimental points. All of these experimental techniques can be used in two ways: (1) to provide estimates of the variation or distribution in responses at a particular design point or (2) to construct surrogate models of the response that can then be used in place of a computationally expensive simulation model for evaluating mean responses and variations (e.g. (Chen, et al., 1996b)). All of these methods suffer from the problem of size identified in (Koch, et al., 1997), in which the number of experiments becomes prohibitively large (given the computational expense of most engineering simulations) as the number of input factors or design variables increases. As an alternative method for propagating variation, the Taylor Series expansions has been applied (Phadke, 1989). A first-order Taylor Series expansion, for example, can be used to relate variation in response, Δy, to variation in a noise factors, $\Delta \mathbf{z}$, or a control factor, $\Delta \mathbf{x}$, as follows:

$$\Delta y = \sum_{i=1}^{n} \left| \frac{\partial f}{\partial x_i} \Delta x_i \right| + \sum_{i=1}^{l} \left| \frac{\partial f}{\partial z_i} \Delta z_i \right| \qquad \text{Eq. 2.12}$$

where the variation may represent a tolerance range or may be a multiple of the standard deviation. In Section 2.2.3, the mathematical details for deriving Eq. 2.12 are given. Higher-order Taylor Series expansions can be formulated to provide a better approximation of the variation in response, but higher-order expansions also require higher-order partial derivatives of the response function with respect to control and noise factors. Taylor Series expansions are relatively accurate for small degrees of variation in control or noise factors but lose their accuracy for larger variations or highly nonlinear functions. A Taylor Series expansion requires evaluation of the partial derivative or sensitivity of the response function with respect to changes in control or noise factors. If analytical expressions are available for the sensitivities, this can be a computationally attractive and relatively accurate approach even for large numbers of control and noise factors.

A crucial point that has not been addressed in literature is that with the Taylor Series expansion a performance variation of zero is estimated in stationary points. This means that optima, which are in general not considered robust, have perfect robustness when evaluated with this technique. This renders this approach inappropriate for problems, where the design space might contain local performance optima. In this thesis, this limitation is addressed in the research questions and hypotheses (see Section 1.3.1). The Taylor Series is also utilized in the original RCEM (Chen, 1995), which is explained in the next section.

2.2.3 The Robust Concept Exploration Method

A foundational component for the research presented in this thesis is the Robust Concept Exploration Method (RCEM) introduced by Chen, et al. (Chen, 1995; Chen, et al., 1996a; Chen, et al., 1996b). This robust design method is explained in this section. The explanation is leveraged with modifications from (Seepersad, 2004).

RCEM is a domain-independent, systematic approach that is formulated by integrating statistical experimentation and approximate models, robust design techniques and multi-objective decisions. The framework and the information flow of RCEM are illustrated in Figure 2.4. In stage A, design variables, noise factors and their ranges as well as the responses are defined. Statistical experiments are designed and simulation data is generated in stages B-D. This data is utilized for building global surrogate models for each response in stage E. For that, the Response Surface Methodology (RSM) is used. The performance values and variances calculated from the response models with a Taylor Series expansion are processed in the compromise Decision Support Problem (cDSP), which builds the frame of this approach. The design problem, which is entirely formulated in the cDSP, can then be solved using a computer system like DSIDES (see Section 2.4).

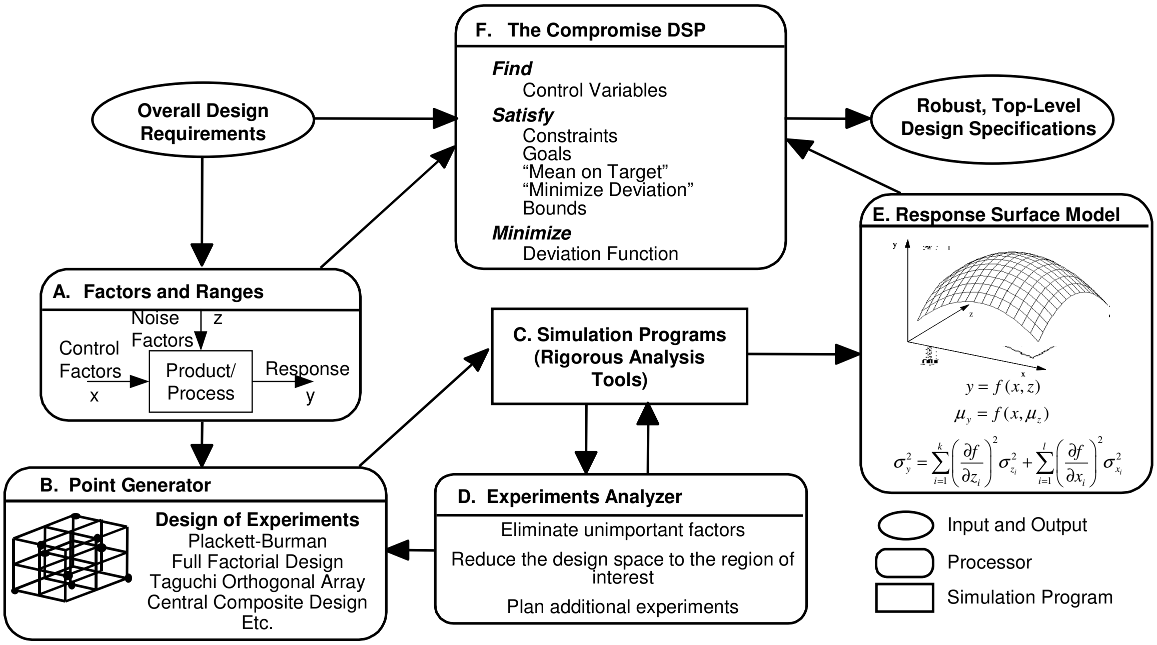


Figure 2.4: Framework of the Robust Concept Exploration Method (Chen, et al., 1996a)

A component that is criticized in this thesis is the variance estimation used in RCEM. In RCEM, the first-order Taylor Series is used to derive the function for the estimation of the performance variance (Phadke, 1989). Variance is defined as the expected squared difference between the mean and the random variable *X:*

$$Var(X) = \sigma_X^2 = E\left[(X-\mu)^2\right] \qquad \text{Eq. 2.13}$$

The variance has the following property:

$$Var(cX_1 + dX_2) = c^2\,Var(X_1) + d^2\,Var(X_2) + 2cd\,Cov(X_1, X_2) \qquad \text{Eq. 2.14}$$

where *c* and *d* are constant factors, X_1 and X_2 are random variables and *Cov(X₁, X₂)* is their covariance. If the two random variables are independent, the covariance is equal to zero and the third term of Eq. 2.14 can be ignored. The first-order Taylor Series expansion of the surrogate model or explicit performance function *f* about the point (*a*,*b*) is defined as:

$$f(x_1, x_2) \approx f(a,b) + \left(\frac{\partial f(a,b)}{\partial x_1}\right)(x_1 - a) + \left(\frac{\partial f(a,b)}{\partial x_2}\right)(x_2 - b) \qquad \text{Eq. 2.15}$$

The variance of the performance $f(x_1, x_2)$ can be calculated using the property given in Eq. 2.14. Assuming that x_1 and x_2 are Gaussian distributed with the variances $\sigma_{x_1}^2$ and $\sigma_{x_2}^2$, the performance variance can be derived as follows:

$$Var[f(x_1, x_2)] \approx Var[f(a,b)] + \left(\frac{\partial f(a,b)}{\partial x_1}\right)^2 Var(x_1 - a) + \left(\frac{\partial f(a,b)}{\partial x_2}\right)^2 Var(x_2 - b) \qquad \text{Eq. 2.16}$$

With $Var[f(a,b)] = 0$, $Var(x_1 - a) = Var(x_1)$ and $Var(x_2 - b) = Var(x_2)$ respectively, the estimation for the performance variance using first order Taylor Series Expansion is:

Eq. 2.17

$$\sigma_y^2 \approx \left(\frac{\partial f}{\partial x_1}\right)^2 \sigma_{x_1}^2 + \left(\frac{\partial f}{\partial x_2}\right)^2 \sigma_{x_2}^2$$

Eq. 2.17 can be generalized for *n* design variables *x* and *m* noise factors *z:*

Eq. 2.18

$$\sigma_y^2 \approx \sum_{i=1}^{n}\left(\frac{\partial f}{\partial x_i}\right)^2 \sigma_{x_i}^2 + \sum_{i=1}^{m}\left(\frac{\partial f}{\partial z_i}\right)^2 \sigma_{z_i}^2$$

Although RCEM has some limitations, which are outlined in the next section, it has been employed successfully for a simple structural problem and design of a solar powered irrigation system (Chen, 1995), a High Speed Civil Transport (HSCT) (Chen, et al., 1996a), and a General Aviation Aircraft (Simpson, et al., 1996). In addition, RCEM has been extended to facilitate the design of complex hierarchical systems (Koch, 1997) and product platforms (Simpson, et al., 2001a).

Within the RCEM framework, the compromise DSP is used as a multi-objective decision support tool for determining the values of design variables that satisfy a set of constraints and balance a set of (conflicting) goals. These goals contain both performance and robustness objectives. In this setting, the cDSP is particularly flexible. Families of robust designs can be generated by changing the priority levels or weights of goals or by changing the target values for design requirements - none of which require reformulation of the compromise DSP.

In this section the RCEM is presented, which the research in this thesis is based on. In the research conducted the limitations of RCEM are addressed, which are outlined in the following section.

2.2.4 Limitations of RCEM

The principal goal for this thesis is to:

Further develop the Robust Concept Exploration Method (RCEM) to facilitate the rapid exploration of nonlinear design spaces and the generation of robust design solutions.

For achieving this goal, the limitations of RCEM, which are outlined in this section, are addressed.

The first limitation found in RCEM is the variance estimation with the first-order Taylor Series expansion (Eq. 2.18). In stationary points of the performance function this technique estimates a performance variance of zero, independent of the deviation of the design variables or noise factors, because the derivative in those points is zero. Since optimal solutions are, in general, not considered to be robust (see e.g. Figure 2.3), the application of this technique in nonlinear problems can lead to fatal errors. This problem is demonstrated with an example in Chapter 3 and is addressed in the first research question:

How can the formulation of robustness in compromise Decision Support Problems for multi-dimensional and multi-objective design problems be improved?

To answer this question the modified RCEM is introduced in Chapter 3, which utilizes an alternative method for the performance variance estimation.

The second limitation of RCEM is the utilization of global response surfaces. Especially the RSM can lead to surrogate models that do not reflect the robust areas of the design space correctly. An example is given in Chapter 3 (see Figure 3.3). The error in the global response models increases as the nonlinearity of the performance function increases. This limitation of RCEM is addressed in the second research question:

> How can the RCEM be augmented to improve the accuracy of the robust concept exploration for nonlinear design problems that involve simulations?

This question is answered with the development of the simulation-based RCEM. In the new method, the Probabilistic Collocation Method (PCM) is incorporated. This uncertainty analysis tool is a local response model approach. The fundamental concept of this method is explained in the next section.

2.3 The Probabilistic Collocation Method

In the last section, the limitations of the original RCEM are outlined. One of the limitations is linked to the application of global response surfaces for design problems that require simulations for the performance evaluation of a design solution. For this reason an alternative surrogate modeling technique is needed. To account for nonlinear problems, in this thesis, a local approach is considered to be a promising alternative. With a local response model approach, surrogate models are only created for a small region around a point of interest as opposed to a global response surface for the entire design space.

In this thesis, the Probabilistic Collocation Method (PCM) is investigated, which was developed by Tatang (Tatang, 1995) for applications in Chemical and Environmental Engineering. It is a very efficient method that generates response models with a minimum number of simulation runs. The method has been previously applied to forecast and analyze future carbon emissions (Webster, 1997), numerical geophysical models (Tatang, et al., 1997), a simple ocean model (Webster, et al., 1996), the evaluation of uncertainty in dynamic simulations of power system models (Hockenberry and Lesieutre, 2004) and the analysis of flow in random porous media (Li and Zhang, 2007). In those applications, the method has been found very efficient and effective for estimating the statistical moments and the probability density function of the system performance under uncertainty. In this thesis, the Probabilistic Collocation Method is introduced to Robust Engineering Design.

The following description of the concept of the Probabilistic Collocation Method is leveraged with modifications from the explanations in (Hockenberry, 2000) and (Webster, et al., 1996).

The basic concept underlying the Probabilistic Collocation Method is to generate an approximation of the response of a simulation as a polynomial function of the uncertain input factors. This polynomial approximation can then be used for traditional uncertainty analysis tools, such as Monte Carlo or Latin hypercube sampling. However, in contrast to the original RCEM, with the Probabilistic Collocation Method the model response is estimated locally instead generating a global response surface. For this local approximation a very small number of simulation runs is needed.

Assuming that in a one dimensional problem, *x* is the uncertain parameter, *p(x)* its probability density function and *g(x)* the outcome of interest of order (2*n*-1), the approximation would have the following form:

$$\hat{g}(x) = g'_0 + g'_1 x + \ldots + g'_{2n-1} x^{2n-1}$$ Eq. 2.19

where g'_i are constants, that could be determined by performing many simulations and applying a least square approach. The minimum number of simulations required to determine all coefficients would be 2*n*. If the primary interest is to just match the expected outcome, the ideas underlying the Gaussian Quadrature Integration can be borrowed. For details on this integration approach, see for example (Davis and Rabinowitz, 1975). Following the Gaussian Quadrature Integration the following relationship can be derived:

$$E[g(x)] = \int p(x) g(x) dx = \sum_{i=1}^{n} p_i g(x_i) = E[\hat{g}(x)]$$ Eq. 2.20

The crucial part in this method is the selection of the sample points, x_i, which is again done according to the Gaussian Quadrature technique. From *p(x),* a set of orthonormal polynomials ***O*** can be computed and the roots of O_n can be calculated. These roots or collocation points are independent of any particular output and are then used to determine the coefficients for the approximation of all outputs, which is presented in Eq. 2.21.

$$\hat{g}(x) = g_0 O_0(x) + g_1 O_1(x) + \ldots + g_{n-1} O_{n-1}(x)$$ Eq. 2.21

With the simulation results at the *n* collocation points, a linear system of equations can be built and solved to determine the coefficients g_i. In case x is Gaussian distributed, the Hermite polynomials can be used as orthogonal polynomials so that Eq. 2.21 has the form of a Polynomial Chaos Expansion (PCE) (Choi, et al., 2007). An overview of the steps of the PCM is given in Figure 2.5. The PCM is explained in greater detail with an example in Section 5.1.1.

From the description it can be seen that the PCM is a method for calculating the coefficients of a polynomial approximation with the bare minimum number of simulation runs. It is designed to reproduce the moments of the outcome with high fidelity by approximating the model response particularly well in regions of high probability. The representation of the outcome in low probability regions might not be as accurate. This negative effect is assumed to have a minor effect on the accuracy when the PCM is used in RCEM. Compared to a global response surface, which is used in the original RCEM, a significant improvement of the accuracy and reliability in non-linear regions is expected. In Chapter 5, the PCM is introduced to robust concept exploration and the simulation-based RCEM is developed. This method used for the validation of the second research hypothesis of this thesis.

In this section, the Probabilistic Collocation Method has been introduced and the underlying theoretical and mathematical concept was presented. In Section 5.1.1, this explanation is extended on a simple example. The computational framework used for applying the new design methods proposed in this thesis is introduced in the next section.

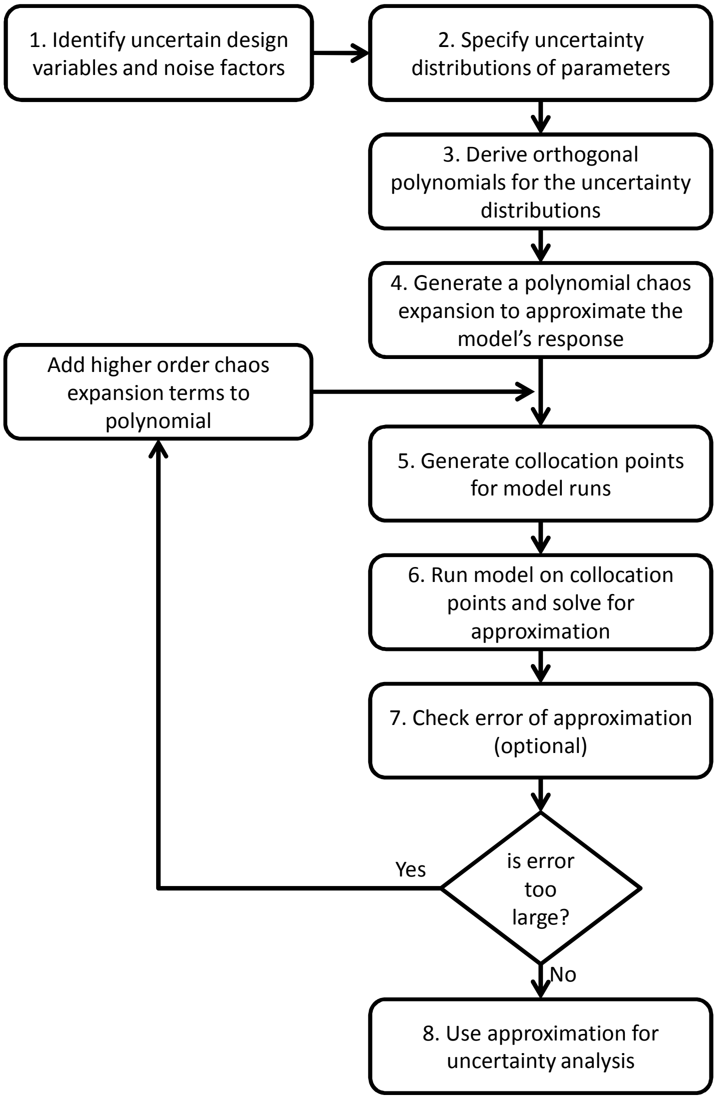

1. Identify uncertain design variables and noise factors
2. Specify uncertainty distributions of parameters
3. Derive orthogonal polynomials for the uncertainty distributions
4. Generate a polynomial chaos expansion to approximate the model's response
Add higher order chaos expansion terms to polynomial
5. Generate collocation points for model runs
6. Run model on collocation points and solve for approximation
7. Check error of approximation (optional)
is error too large?
Yes
No
8. Use approximation for uncertainty analysis

Figure 2.5: Information Flow Diagram of the Probabilistic Collocation Method (Webster, et al., 1996)

2.4 New Computational Framework for RCEM

For the application of the new design methods proposed in this thesis an appropriate computational framework is required. In this section, the framework that was developed for this thesis is outlined.

The foundational framework of all RCEM variants is the compromise DSP. The design problems are always formulated as a cDSP, which has to be solved for finding the desired solutions. Solving a cDSP means to minimize the deviation function while satisfying all constraints and bounds (Section 2.1). The Adaptive Linear Programming (ALP) is considered to be an appropriate algorithm for solving cDSPs (Mistree, et al., 1993). This algorithm and the computer program, which implements ALP, are introduced briefly in Section 2.4.1. The new extended computational framework developed for this thesis is presented in Section 2.4.2.

2.4.1 Adaptive Linear Programming and DSIDES

In this section, the Adaptive linear Programming (ALP) as well as the original DSIDES (Decision Support In the Design of Engineering Systems) are briefly introduced. Since the focus in this thesis is not on testing or modifying this component, it is referred to (Mistree, et al., 1993) for greater details.

The ALP has been developed for the design of large, highly constrained, complex systems (Mistree, et al., 1981). According to (Mistree, et al., 1993) the following three features contribute to the success of the ALP algorithm:

- the use of second-order terms in linearization
- the normalization of the constraints and goals and their transformation into generally well-behaved convex functions in the region of interest
- an "intelligent" constraint suppression and accumulation scheme.

These features are described in detail in (Mistree, et al., 1981). The implementation of the ALP algorithm for solving cDSPs is illustrated in Figure 2.3.

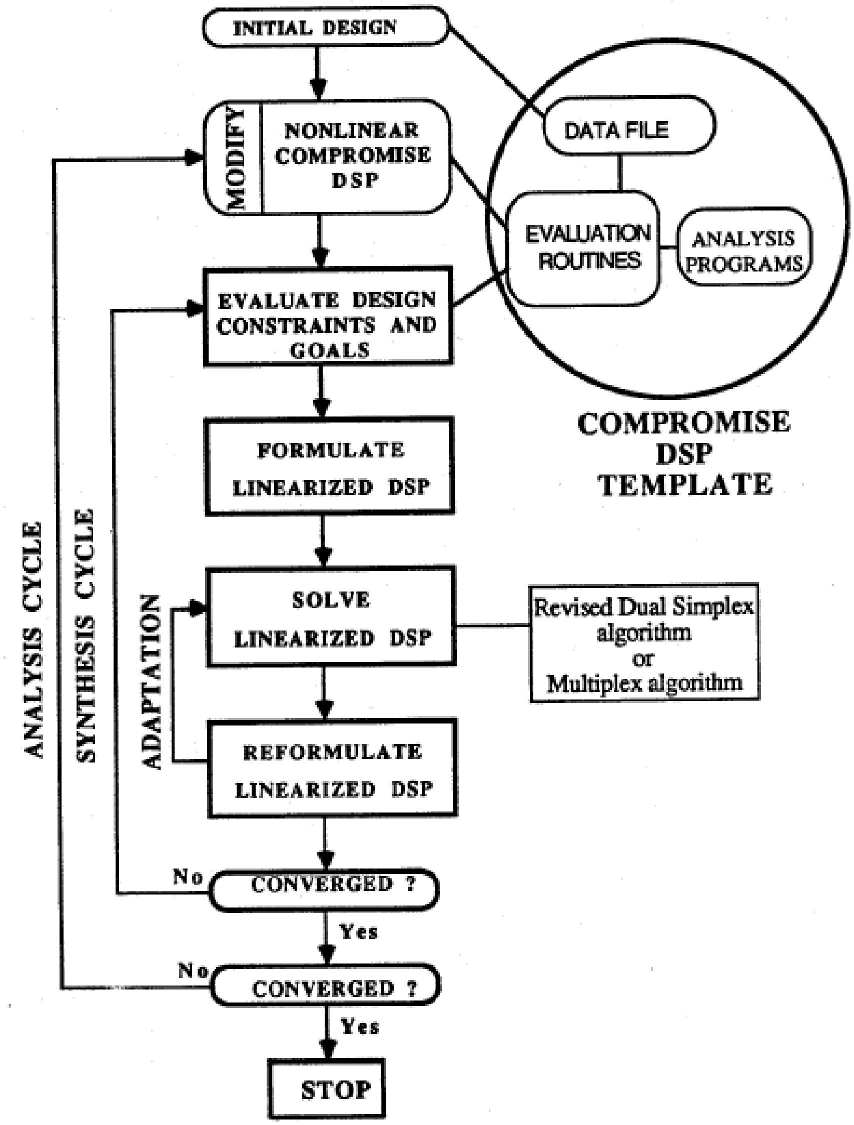


Figure 2.6: Implementation of the ALP for Solving cDSPs (Mistree, et al., 1993)

ALP has been implemented in DSIDES, a computer system written in FORTRAN that can be used for solving cDSPs. The original DSIDES is appropriate for solving both compromise and selection DSPs. However, since the FORTRAN is not very widely used, Matthew Marston has written the cDSP part of DSIDES in Java. This JavaDSIDES is significantly easier to apply and further develop than the original FORTRAN version. JavaDSIDES is used as the foundation of the computational framework used in this thesis. A manual for JavaDSIDES is presented in Appendix A. The additional components of the computer system used in this thesis are introduced in the following section.

2.4.2 New Computational Framework with JavaDSIDES and MATLAB Integration

In this section, the entire computational framework used in this thesis is introduced briefly. It is based on JavaDSIDES, which is described in the previous section. For the application of JavaDSIDES, the cDSP of the design problem is formulated in an XML document. Since many designers are not familiar with the XML syntax, this has been found a limitation of JavaDSIDES. In order to make the XML document creation more convenient, a MATLAB program has been developed as part of this thesis. This MATLAB cDSP XML Generator offers a clear and intuitive interface for the designer. The designer can type in the cDSP almost as he or she would formulate it as a word document. The code and the interface are presented in Figure A.5 in Appendix A. If a global response surface is used or the performance and constraint functions are known explicitly, JavaDSIDES is executed right from this interface and the results are displayed in the MATLAB command window and are saved as an output text file and a spreadsheet in the DESIDES root folder. Examples of these files are given in Figure A.9 as well in Table A.1 and Table A.2 in Appendix A.

To make JavaDSIDES applicable in this thesis, especially for the simulation-based RCEM, the integration with MATLAB has to be established. This is accomplished by wrapping the MATLAB code in Java and thus, making it executable as native Java functions. This was required, since the PCM implementation as well as the simulation code for the LCA heat exchanger is written in MATLAB. The MATLAB add-on, MATLAB Builder JA, can be used for deploying the MATLAB functions. A description of this process is given in Appendix A.8. The deployed MATLAB Java function packages can then be imported to the JavaDSIDES model file. The detailed steps for this process are given in Appendix A.9. A simplified scheme of the computational framework is shown in Figure 2.3.

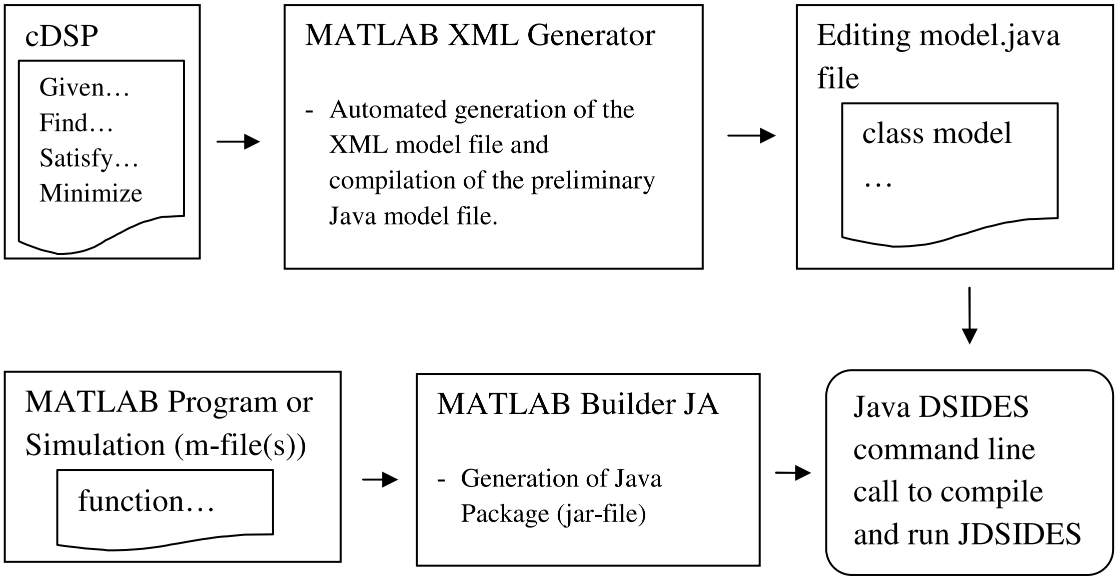

cDSP
Given…
Find…
Satisfy…
Minimize
MATLAB XML Generator
- Automated generation of the XML model file and compilation of the preliminary Java model file.
Editing model.java file
class model
…
MATLAB Program or Simulation (m-file(s))
function…
MATLAB Builder JA
- Generation of Java Package (jar-file)
Java DSIDES command line call to compile and run JDSIDES

Figure 2.7: Computational Framework used in this Thesis

This framework allows for implementing all functions and simulations written in MATLAB with JavaDSIDES. This also applies to brute force Monte Carlo Simulations or Latin hypercube samplings that are written in MATLAB, as used for example in Chapter 3 of this thesis.

2.5 Chapter Synopsis

As presented in Figure 2.8, in this chapter the fundamental concepts and methods, which the research presented in this thesis is based on, are introduced. The compromise DSP is explained in Section 2.1; the robust design approach are reviewed and discussed in Section 2.2 with the RCEM introduced in Section 2.2.3. The limitations of RCEM are outlined as the research opportunities addressed in this thesis in Section 2.2.4. Furthermore, the Probabilistic Collocation Method as an integral component of the new simulation based RCEM is explained in Section 2.3. Finally, the computational framework required for the application of the proposed methods is presented in Section 2.4.

After discussing all foundations, the research conducted in presented in the next chapters. Chapter 3 contains the demonstration of the limitations of RCEM and the development of alternative variance estimation techniques. The modified RCEM, which is introduced in Section 3.2.4, is validated in Chapters 3 and 4. In Chapter 5 the simulation-based RCEM is developed and validated. The validation of this method is continued in Chapter 6.

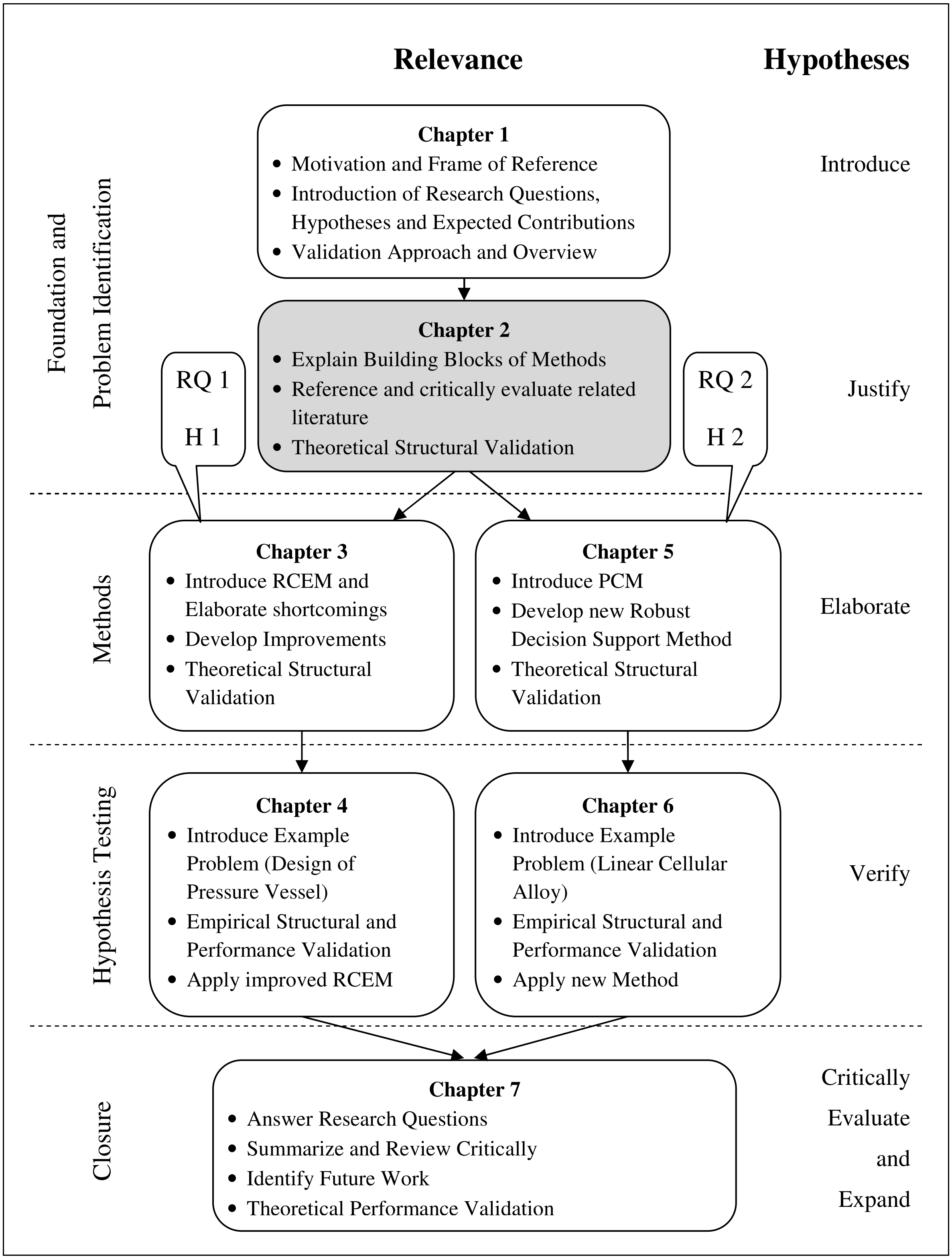

Relevance
Hypotheses
Foundation and Problem Identification
Chapter 1
Motivation and Frame of Reference
Introduction of Research Questions, Hypotheses and Expected Contributions
Validation Approach and Overview
Introduce
Chapter 2
Explain Building Blocks of Methods
Reference and critically evaluate related literature
Theoretical Structural Validation
RQ 1
H 1
RQ 2
H 2
Justify
Methods
Chapter 3
Introduce RCEM and Elaborate shortcomings
Develop Improvements
Theoretical Structural Validation
Chapter 5
Introduce PCM
Develop new Robust Decision Support Method
Theoretical Structural Validation
Elaborate
Hypothesis Testing
Chapter 4
Introduce Example Problem (Design of Pressure Vessel)
Empirical Structural and Performance Validation
Apply improved RCEM
Chapter 6
Introduce Example Problem (Linear Cellular Alloy)
Empirical Structural and Performance Validation
Apply new Method
Verify
Closure
Chapter 7
Answer Research Questions
Summarize and Review Critically
Identify Future Work
Theoretical Performance Validation
Critically Evaluate and Expand

Figure 2.8: Thesis Roadmap

CHAPTER 3

MODIFYING THE ROBUST CONCEPT EXPLORATION METHOD

In traditional engineering design, the designers make their decisions with the mindset of achieving the highest feasible performance of their product. In robust design, however, designers want to find a trade-off between performance and robustness, which measures the product's sensitivity to uncertain factors. It is crucial at this stage to find a reliable way for assessing the robustness of a system. The success of a robust design method heavily depends on how robustness is defined and formulated. This is addressed in the first research question introduced in Chapter 1:

First Research Question:

How can the formulation of robustness in compromise Decision Support Problems for multi-dimensional and multi-objective design problems be improved?

In this chapter the answer to this research question is developed. Therefore, in Section 3.1 the Robust Concept Exploration Method, which was introduced by Wei Chen (Chen, 1995), is reviewed, analyzed, and its limitations and resulting research gaps are presented. Thereupon, several possible ways to improve the method are discussed and tested in Section 3.2. These improvements are based on the first research hypothesis, which is also described in Chapter 1:

First Research Hypothesis:

Using a global response surface, the nominal mean and a single point first order Taylor Series Approximation for the variance estimation is not accurate enough for finding robust solutions of nonlinear design problems. The Robust Concept Exploration Method can be improved by using appropriate formulations that account for the nonlinearity of the problem.

In order to validate the new method, for which the validation strategy is explained in Section 3.3, a new computational framework is required. This framework was also developed as part of this thesis and is introduced in Section 2.4.

3.1 Characteristics and Limitations of RCEM

Based on the detailed description of the Robust Concept Exploration Method in Chapter 2, the focus in this section is on analyzing and discussing the method. The underlying assumptions are compared to the claims and the performance is assessed. This critical review leads to the limitations of the method, which build the foundation for the first research question.

What is the objective of applying RCEM?

The Robust Concept Exploration Method was developed to support the early stages of the design of complex systems. It is intended to provide a tool for designers to examine the entire design space effectively and efficiently and determine feasible robust solutions. To explain the desired outcome of RCEM, often Figure 3.1 is used in literature (e.g. (Chen, 1995; Chen, et al., 1996b)).

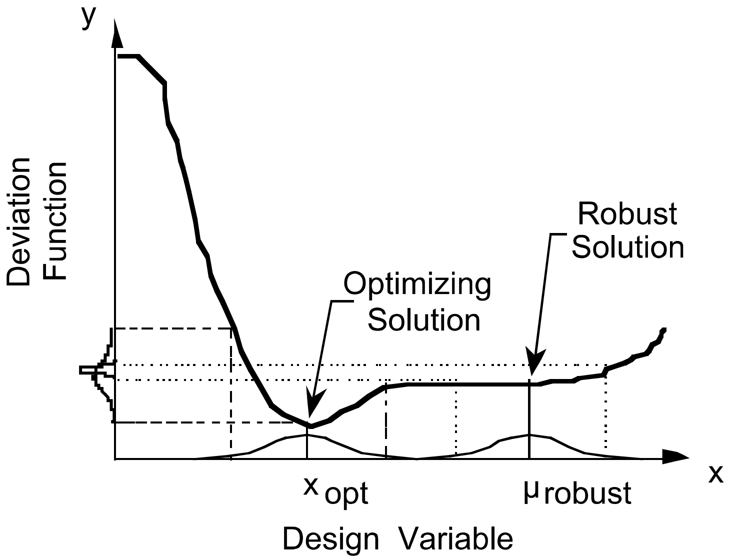


Figure 3.1: Optimizing Solution vs. Robust Solution (Chen, 1995; Chen, et al., 1996b)

In Figure 3.1, a univariate nonlinear design problem is illustrated. In this example, *x* is an uncertain design variable and *y* is the deviation function that does not consider robustness. In this case, the deviation function can be seen as a signal factor, which has to be minimized for better performance of the engineering system. Using traditional engineering optimization, the designer would choose x_{opt} as the preferred solution. However, by introducing robustness as an objective, the robust solution μ_{robust} becomes the better solution because it is less sensitive to variations of the design variable *x* and still has a satisfying performance. RCEM is developed to find these robust solutions. To accomplish this, a synthesis of ideas from the Taguchi method, statistical experimentation methods and nonlinear programming methods is used. Although the combination of those methods is very promising, the framework of RCEM has some crucial gaps.

What are the limitations of RCEM?

The limitations of RCEM are explored and explained along the three steps of the method which are according to (Chen, et al., 1996b):

> "Step 1: Build response surface models to relate each response to all important control- and noise- factors using the Response Surface Methodology
>
> Step 2: Derive functions of mean and variance of the responses based on the type of robust design applications.
>
> Step 3: Use the compromise DSP to find the robust design solution"

These steps are also illustrated in Figure 2.3.

The limitations of Step 1 are related to the accuracy of the surrogate model generated using the Response Surface Methodology. Considering that RCEM is meant to assist designers in the early stages of design, it can be assumed that a) the design space is still relatively large and b) the design might change, which requires a repetition of the method. Most engineering design problems have several design variables, which makes the generation of a response surface for a large design space computationally expensive. This response surface becomes invalid as soon as the underlying model is modified. Furthermore, the second or third order response surfaces might not be able to provide a good fit for a large design space, especially if the problem is nonlinear or even highly nonlinear. In a nonlinear design problem, it can lead to fatal problems, if the fit of the response surface is not accurate enough around a robust region. In such a case, the best goal formulations cannot detect it. This problem is addressed in (Lin, et al., 1999), where the authors use Eq. 3.1 as an artificial performance function.

Eq. 3.1

$$f(x)=\sum_{i=1}^{9} a_i (x-900)^{(i-1)} \qquad with\, x=[910...976]$$

$a_1 =$	−659.23	$a_4 =$	0.82691	$a_7 =$	−3.2446 x 10-6
$a_2 =$	190.22	$a_5 =$	−0.021885	$a_8 =$	1.6606 x 10-8
$a_3 =$	−17.802	$a_6 =$	0.0003463	$a_9 =$	−3.5757 x 10-11

This function is chosen so that it is representative for an engineering problem, which RCEM is designed for. The example is not complex; however, it has characteristics, with which many concept exploration methods have difficulties; it has (a) a global performance maximum, (b) a global minimum and (c) a robust area, which are illustrated in Figure 3.2.

In order to illustrate the limitation of the first RCEM step, a quadratic and a third order response surface for this problem are generated. The surrogate models are fitted to 23 sample points evenly distributed over the design space. The response surfaces, together with the performance function and sample points, are illustrated in Figure 3.3.

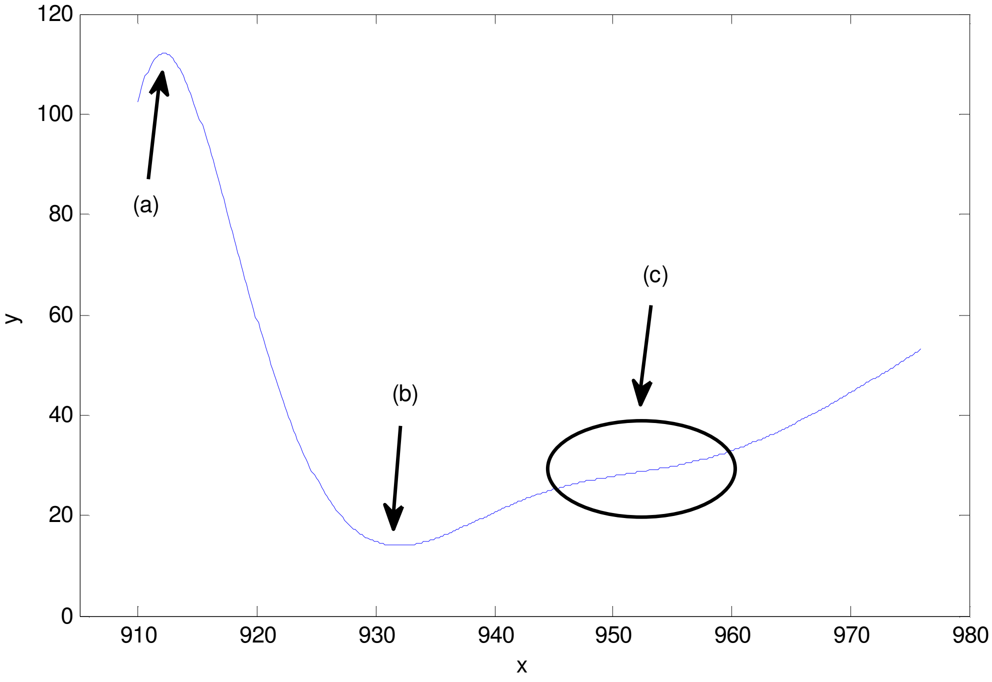


Figure 3.2: Example Function with (a) Maximum, (b) Minimum and (c) Robust Area

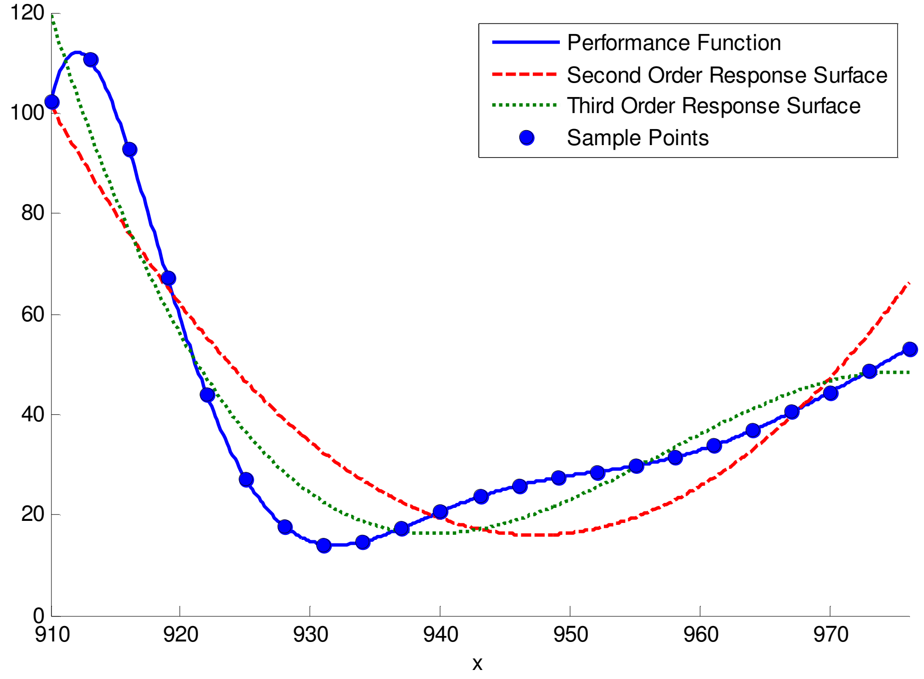


Figure 3.3: 2nd and 3rd Order Response Models of the Example Function.

It can be seen that both surrogate models do not represent the robust area accurately, which would make the exploration method giving wrong results. This example raises the question of whether response surfaces that are generated over the entire design space are at all appropriate for robust engineering design problems. Since robust design does not make sense for linear problems (equal robustness over the entire design space), it can be assumed that all robust exploration problems are at least of quadratic order; in the case of systems engineering with many interdependencies between components, the designer often has to deal with highly nonlinear performance functions. Considering that decisions in the early stages of design should be made extremely thoroughly, a better method is needed for problems where the performance function is not known explicitly.

This shortcoming of RCEM influenced the second research question and is addressed more detailed in Chapters 5 and 6.

For the remainder of Chapters 3 and 4, no response surfaces are considered; instead, examples are used, in which explicit performance functions are available and no surrogate model is required. This allows focusing on the performance and robustness formulations.

Another critical part of RCEM is Step 2, where the functions for the performance mean and the performance variance (robustness) are derived from the response surface or the explicit performance function. In RECM, the performance mean μ_y is simply assumed to be the function value at the mean of design variables μ_x and noise factors μ_z (Eq. 3.2). This is called the nominal mean.

$$\mu_y \approx f(\mu_x, \mu_z) \qquad \text{Eq. 3.2}$$

This estimation is only accurate for linear or nearly linear functions. If *f(x)* is nonlinear, a nominal value shift can be observed (Giovagnoli and Romano, 2008; Li and Wu, 1999). This means that the actual mean shifts away from the nominal mean. In Figure 3.4, this phenomenon is illustrated.

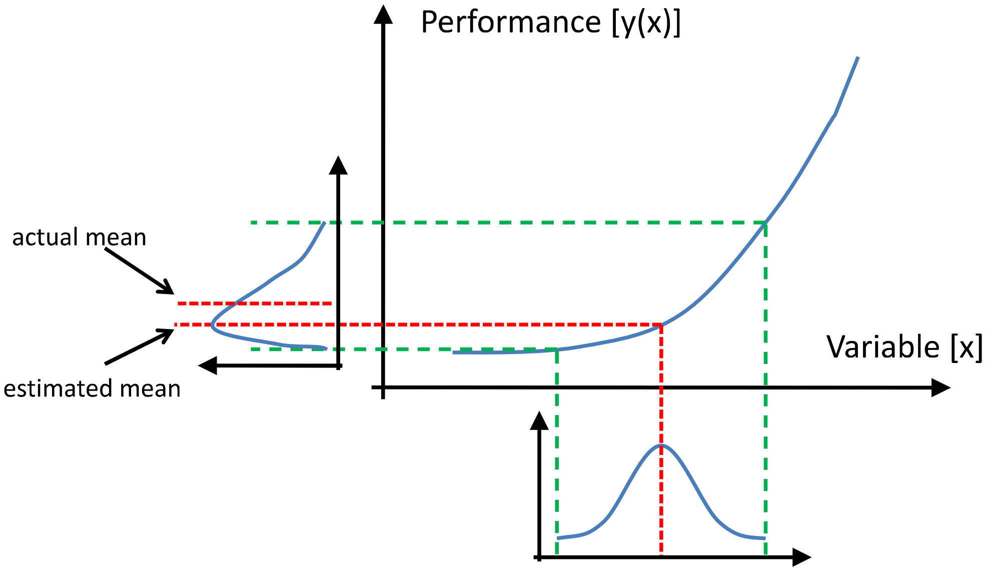


Figure 3.4: Nominal Value Shift

The nonlinearity of the performance function can lead to a nonsymmetrical output distribution, where the mean is no longer equal to the mode. For example, in Figure 3.4, the output has a positive skew in which case the actual mean is shifted to the right from the estimated mean.

In Section 3.2.1, a possibility for estimating the performance mean is introduced, which accounts for the nominal value shift.

A similar, however more severe problem can be found in the estimation of the performance variance. In RCEM, the first-order Taylor Series is used to estimate the output variance, which is used to formulate and assess the robustness (see Section 2.2.3).

$$\sigma_y^2 \approx \sum_{i=1}^{n} \left(\frac{\partial f(\mathbf{x},\mathbf{z})}{\partial x_i} \right)^2 \sigma_{x_i}^2 + \sum_{i=1}^{m} \left(\frac{\partial f(\mathbf{x},\mathbf{z})}{\partial z_i} \right)^2 \sigma_{z_i}^2 \qquad \text{Eq. 3.3}$$

In this equation, simply the square of the derivative at the design variable or noise factor mean is multiplied with the variance of the design variable or noise factor. This means that the steeper the function is at the point of interest, the higher is the expected variance. This approach works as long as the function is nearly linear. When dealing with nonlinear functions, this approach can lead to fatal errors. The biggest problem occurs in local optima; here the derivative is equal to zero, which leads to a performance variance estimation of zero, while the actual variance can be very high if it is a peak optimum. This is why an exploration method using the first order Taylor Series Expansion will converge in a local optimum (if there is one) instead of converging in a robust area. This concern is confirmed with the following example using Eq. 3.1 as the performance function, where a design is considered better if it has a lower function value and a lower performance variance (higher robustness). By assigning a high weight to the robustness goal, the designer would then expect the robust solution to be in area (c) of Figure 3.2. However, with the example it will be shown that the exploration algorithm will find the optimum (b) as the “robust solution”. This is because, for the used formulation, this point is the best solution, since it has the best performance and the Taylor Series leads to a variance estimation of zero (maximum robustness).

GIVEN

Performance function $f(x)$: Eq. 3.1

Derived estimations for mean $\mu_{f(x)}$ and variance $\sigma^2_{f(x)}$: Eq. 3.2and Eq. 3.3

Variance of design variable $\sigma_x^2 = 4$ (Gaussian distributed)

FIND

Design variable: x

Deviation variables: $d_i^-, d_i^+ \quad i = 1,2$

SATISFY

GOALS

Performance goal: $$\frac{1/f(x)}{1/10} + d_1^- - d_1^+ = 1$$

Robustness goal: $$\frac{1/\sigma^2_{f(x)}}{1/0.5} + d_2^- - d_2^+ = 1$$

BOUNDS

$910 \le x \le 976$

$d_i^-, d_i^+ \ge 0 \quad i = 1,2$

$d_i^- \cdot d_i^+ = 0 \quad i = 1,2$

MINIMIZE

$Z = W_1 \cdot d_1^- + W_2 \cdot d_2^- \qquad d_1^+, d_2^+ = 0$

Scenario 1:	Scenario 2:
$W_1 = 0.5$	$W_1 = 0.1$
$W_2 = 0.5$	$W_2 = 0.9$

Figure 3.5: cDSP for Illustrative Example

The cDSP for the example is given in Figure 3.5. This cDSP is solved using a Java implementation of DSIDES (Mistree, et al., 1993) that allows solving cDSPs with explicit goal functions. The Scenario 1 results of the robust design exploration for different starting points are given in Table 3.1 and Figure 3.6. For evaluation of the results, accurate reference values for the deviation function are generated using a Latin hypercube sampling. For experiment run #4 the convergence plot is shown in Figure 3.7.

Table 3.1: RCEM Results for Illustrative Example (Scenario 1)

Run #	Initial Point	Solution	Estimated Deviation		Actual Deviation (LHS)	
			d_1^-	d_2^-	d_1^-	d_2^-
1	x = 910	910	0.90232	0.99879	0.8883	0.9847
2	x = 912	911.9925	0.91077	0	0.9040	0.9619
3	x = 920	931.9491	0.28378	0	0.3211	0.5671
4	x = 940	931.9489	0.28378	0	0.3211	0.5674
5	x = 960	931.9494	0.28378	0	0.3211	0.5655
6	x = 976	931.9487	0.28378	0	0.3211	0.5664

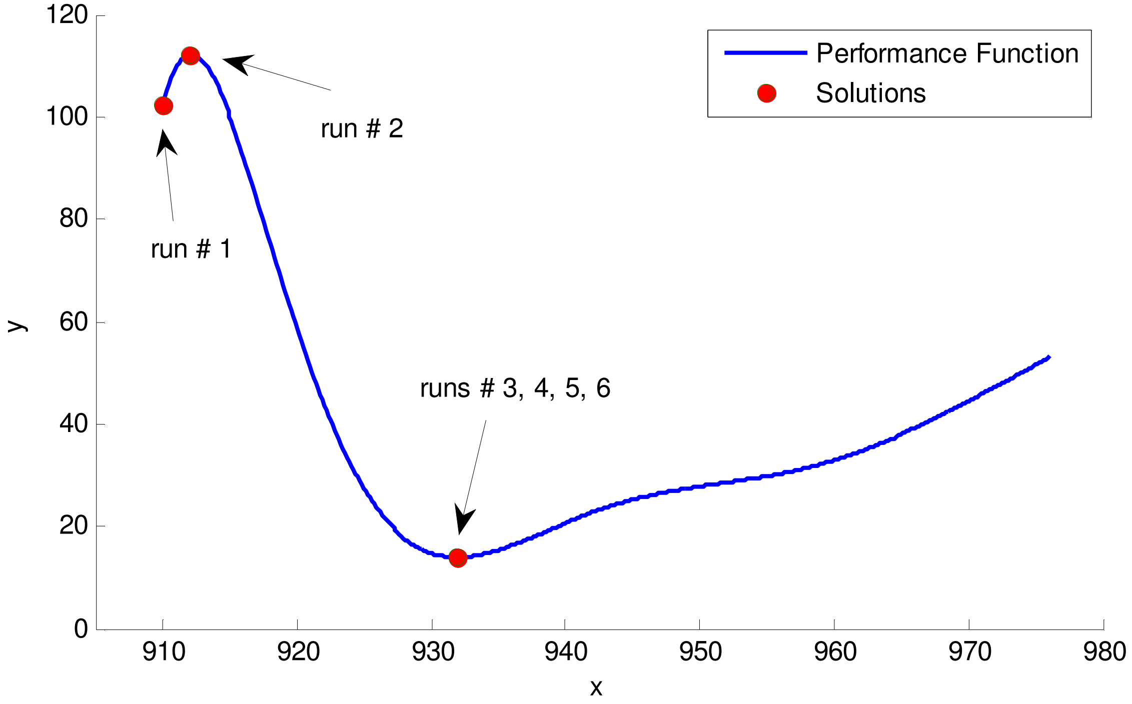


Figure 3.6: Plot of RCEM Results

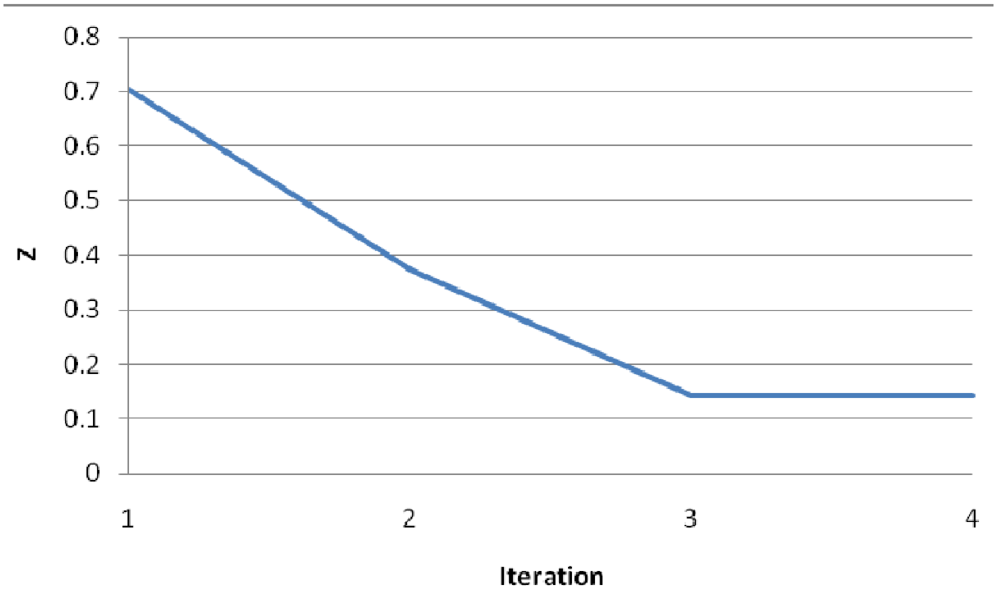


Figure 3.7: Convergence Plot of Run #4

From the results and Figure 3.6, it can be seen that the solutions do not lie in the robust area. In Section 3.2, it is shown that this result is actually correct and a good compromise for the stated objectives. However, the estimated deviation from the variance goal in the solution points is significantly different from the actual deviation, which is calculated using a Latin hypercube sampling with 10,000 points (Table 3.1). To confirm the concerns about RCEM, the example is repeated with a heavy weight on the robustness goal ($W_1 = 0.1, W_2 = 0.9$). In this scenario, a design in the robust area is a significantly better compromise than a solution around the minimum of the performance function (see Section 3.2). The results of the original RCEM for this scenario are presented in Table 3.2.

Table 3.2: RCEM Results for Illustrative Example (Scenario 2)

Run #	Initial Point	Solution	Estimated Deviation		Actual Deviation (LHS)	
			d_1^-	d_2^-	d_1^-	d_2^-
1	x = 910	931.9484	0.2837	0	0.3211	0.5659
2	x = 912	911.9925	0.9107	0	0.9040	0.9619
3	x = 920	931.9526	0.2837	0	0.3211	0.5699
4	x = 940	931.9528	0.2837	0	0.3211	0.5699
5	x = 960	931.9528	0.2837	0	0.3211	0.5699
6	x = 976	931.9488	0.2837	0	0.3211	0.5657

The results of the second scenario are not significantly different from the first, which means they are also very close to the global minimum at x = 931.9499. This confirms a major gap in RCEM, which leads to the first research question:

> How can the formulation of robustness in compromise Decision Support Problems for multi-dimensional and multi-objective design problems be improved?

A hypothesis for answering this question is introduced in the next section.

3.2 Advancements of RCEM

In the previous section, several gaps in RCEM are identified. In this section the hypothesis for answering the first research question is elaborated and the shortcomings in Step 2 of RCEM are addressed.

3.2.1 Nominal Value Shift

How can the estimation of the performance mean be improved?

The problem in RCEM is that the estimation of the performance mean is simply the function value of the nominal values (means) of the design and noise factors. This is only accurate for linear problems. If the input distribution is transformed by a nonlinear performance function, a nominal value shift can be observed (Section 3.1). To account for the nominal value shift, a Taylor Series Expansion can be used (Beyer and Sendhoff, 2007; Li and Wu, 1999; Vuchkov and Boyadjieva, 2001).

The expected value of the performance can be derived from the second order Taylor Series. The mathematical proof is shown in the following.

The second order Taylor Series Expansion (without cross product terms) around the point $\mathbf{x_0}$ for **independent** variables is shown in Eq. 3.4. When assuming independent variables, the cross product terms can be ignored in this calculation.

Eq. 3.4

$$y = f(\mathbf{x}) \approx f(x_{1_0},\ldots,x_{n_0}) + \sum_{i=1}^{n}\left(\frac{\partial f(x_{i_0})}{\partial x_i}\right)(x_i - x_{i_0}) + \frac{1}{2}\sum_{i=1}^{n}\left(\frac{\partial^2 f(x_{i_0})}{\partial x_i^2}\right)(x_i - x_{i_0})^2$$

Eq. 3.5

$$\mu_y = E[y] = E[cx_1 + dx_2] = cE[x_1] + dE[x_2]$$

Eq. 3.6

$$\sigma_X^2 = Var[X] = E[(X - \mu_X)^2]$$

The properties used to derive the mean of y from Eq. 3.4 are given in Eq. 3.5 and Eq. 3.6. In this step, the cross product terms would become zero because independent variables, which have a covariance of zero, are assumed. In Eq. 3.7, the result of the transformation is presented.

Eq. 3.7

$$\mu_y = E[y] \approx f(x_{1_0},\ldots,x_{n_0}) + \frac{1}{2}\sum_{i=1}^{n}\left(\frac{\partial^2 f(\mathbf{x_0})}{\partial x_i^2}\right)\sigma_{x_i}^2$$

For problems, in which design variables and noise factors are considered, the formulation is given in Eq. 3.8:

$$\mu_y = E[y] \approx f(\mathbf{x_0},\mathbf{z_0}) + \frac{1}{2}\sum_{i=1}^{n}\left(\frac{\partial^2 f(\mathbf{x_0},\mathbf{z_0})}{\partial x_i^2}\right)\sigma_{x_i}^2 + \frac{1}{2}\sum_{i=1}^{m}\left(\frac{\partial^2 f(\mathbf{x_0},\mathbf{z_0})}{\partial z_i^2}\right)\sigma_{z_i}^2 \qquad \text{Eq. 3.8}$$

Part of the first research hypothesis is that the introduction of this formulation to RCEM as a replacement for the nominal mean will improve the accuracy of the exploration method.

Validation

The new formulation for estimating the mean of the performance, which accounts for the nominal value shift, will be tested on several points of the design space and a comparison will be made to a Latin hypercube sampling as well as to the results generated with the previous formulation used in RCEM.

The easiest way to validate the effectiveness of the new formulation Eq. 3.8 is to test it on various point of the design space and compare the estimation to values calculated from a LHS sample. In order to demonstrate the improvement towards the formulation in the original RCEM, the results generated using Eq. 3.2 are also given for the test points. In Figure 3.8, the locations of the chosen test points are shown and in Table 3.3 and Table 3.4, the results of the new and old formulations are compared. In this test, design variables are assumed to be Gaussian distributed with variances of $\sigma_{x1}^2 = 4$ and $\sigma_{x2}^2 = 9$.

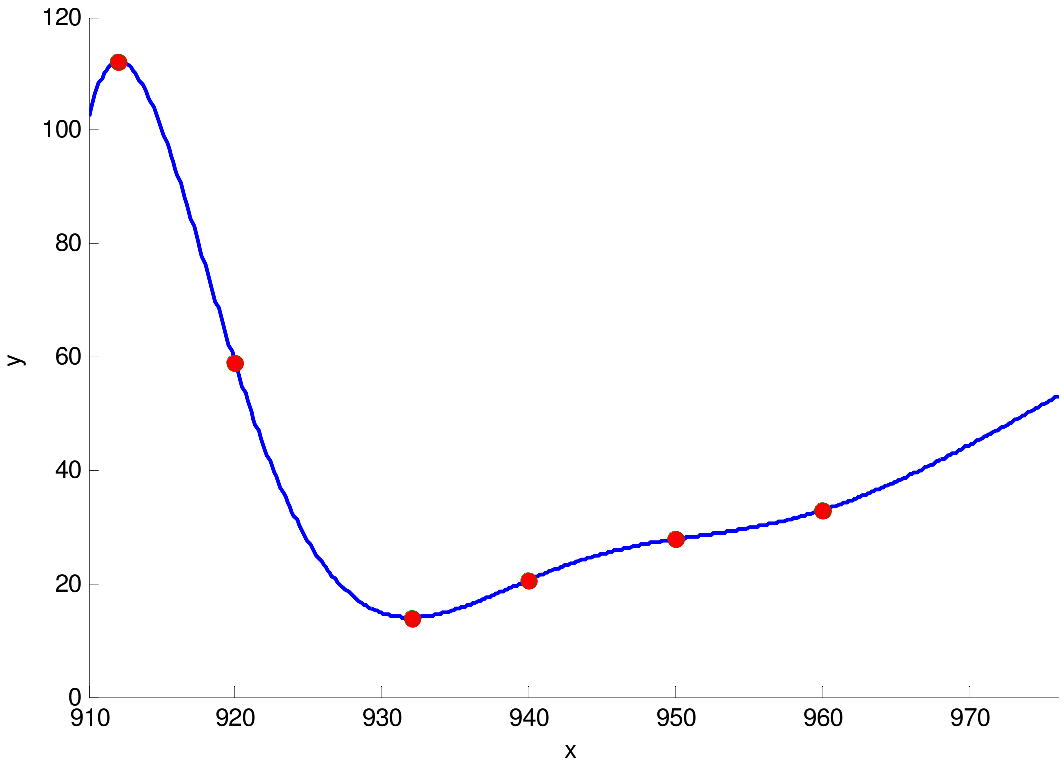


Figure 3.8: Location of Test Points

Table 3.3: Comparison of Mean Predictions with σ^2=4

Test Point	LHS (10000 samples)	New Formulation Eq. 3.8	Nominal mean
912	104.1838	104.544	112.0770
920	59.8057	59.9255	58.8959
932	14.7218	14.7173	13.9628
940	20.5691	20.5552	20.6384
950	27.7717	27.7704	27.8247
960	33.0813	33.086	32.9216

Table 3.4: Comparison of Mean Predictions with σ^2=9

Test Point	LHS (10000 samples)	New Formulation Eq. 3.8	Nominal mean
912	93.3716	95.1279	112.0770
920	60.5949	61.2126	58.8959
932	15.6821	15.6605	13.9628
940	20.5204	20.4512	20.6384
950	27.7094	27.7026	27.8247
960	33.2683	33.2915	32.9216

From the results, a clear advantage of the new formulation can be observed particularly at the test points around the optima. However, in all cases the new formulation led to a more accurate performance mean estimation. This is why in this thesis it is recommended to incorporate the new formulation into RCEM.

3.2.2 Alternative Methods for Variance Approximation

How can the estimation of the performance variance be improved, especially around local optima?

The answer to this question will be the major part of the first research hypothesis. Therefore, three possible formulations are introduced in this section and their effectiveness is tested on a simple illustrative example. Upon the evaluation of the results, the most promising formulation is chosen and will be used to modify the original RCEM. The three methods are:

1. Variance estimation using second order Taylor Series Expansion (Second Derivative Method)
2. Variance estimation using first order Taylor Series Expansion at the nominal value and points around the mean (Multiple Derivative Method)
3. Evaluation of the function values at and around the nominal value (Multiple Point Method)

Similar approaches can be found in literature, e.g. (Lin, et al., 1999) and (Sundaresan, et al., 1995), however, a comprehensive comparison of their effectiveness, especially for their application in a robust concept exploration method, is missing.

Second Derivative Method

The Second Derivative Method is derived from the second order Taylor Series Expansion. In Section 3.1, it was shown that the first order Taylor Series fails as a good approximation, since it does not account for the nonlinearity of the performance function. By considering the second derivative of the performance function, the second order Taylor Series might have the potential to eliminate this problem. The cross product terms are ignored in order to simplify the calculation in Eq. 3.9.

Eq. 3.9

$$y = f(\mathbf{x}) \approx f(x_{1_0}, \ldots, x_{n_0}) + \sum_{i=1}^{n} \left(\frac{\partial f(x_{i_0})}{\partial x_i} \right)(x_i - x_{i_0}) + \frac{1}{2} \sum_{i=1}^{n} \left(\frac{\partial^2 f(x_{i_0})}{\partial x_i^2} \right)(x_i - x_{i_0})^2$$

With the assumption of independent variables as well as the properties of the variance in Eq. 2.13 and Eq. 2.14, the performance variance can be derived as:

Eq. 3.10

$$\sigma_y^2 \approx \sum_{i=1}^{n}\left(\frac{\partial f}{\partial x_i}\right)^2 \sigma_{x_i}^2 + \sum_{i=1}^{m}\left(\frac{\partial f}{\partial z_i}\right)^2 \sigma_{z_i}^2 + \frac{1}{4}\sum_{i=1}^{n}\left(\frac{\partial^2 f}{\partial x_i^2}\right)^2 \sigma_{x_i}^4 + \frac{1}{4}\sum_{i=1}^{m}\left(\frac{\partial^2 f}{\partial z_i^2}\right)^2 \sigma_{z_i}^4$$

for the design variables $\mathbf{x}$ and noise factors $\mathbf{z}$.

Multiple Derivative Method

The idea for developing the Multiple Derivative Method is to consider not only the nominal input values but also the points around the mean. Therefore, the first order Taylor Series approximation is used for the variance prediction (Eq. 3.3) and is evaluated at three points for each design variable and noise factor. This formulation prevents the method from estimating a performance variance of zero in local optima.

Eq. 3.11

$$\begin{aligned}\sigma_y^2 \quad & \approx W_1\left[\sum_{i=1}^{n}\left(\frac{\partial f}{\partial x_i}\right)^2 \sigma_{x_i}^2 + \sum_{i=1}^{m}\left(\frac{\partial f}{\partial z_i}\right)^2 \sigma_{z_i}^2\right] \\ & + W_2\left[\sum_{i=1}^{n}\left(\frac{\partial f(\ldots, x_i + \Delta x_i, \ldots)}{\partial x_i}\right)^2 \sigma_{x_i}^2 + \sum_{i=1}^{m}\left(\frac{\partial f(\ldots, z_i + \Delta z_i, \ldots)}{\partial z_i}\right)^2 \sigma_{z_i}^2\right] \\ & + W_3\left[\sum_{i=1}^{n}\left(\frac{\partial f(\ldots, x_i - \Delta x_i, \ldots)}{\partial x_i}\right)^2 \sigma_{x_i}^2 + \sum_{i=1}^{m}\left(\frac{\partial f(\ldots, z_i - \Delta z_i, \ldots)}{\partial z_i}\right)^2 \sigma_{z_i}^2\right]\end{aligned}$$

In Eq. 3.11, the formulation of the Multiple Derivative Method is shown, which is basically a weighted average of the first-order Taylor Series approximation at the input means (nominal x and z values) and edge points ($x_i \pm \Delta x_i, z_i \pm \Delta z_i$). Since the formulation consists of three terms it is reasonable to divide the input distribution into three arrays to determine $W_i, \Delta x_i$ and Δz_i .

If Gaussian distributions are assumed for the uncertain input, according to Figure 3.9, the following parameter values for Eq. 3.11 can be suggested:

$x_i = \mu_{x_i}$ and $z_i = \mu_{z_i}$ with $W_1 = 0.7$

$\Delta x_i = 2\sigma_{x_i}$ with $W_2 = W_3 = 0.15$

The middle section in Figure 3.9 is represented by the mean and covers about 70% of the probability density function. The outer sections are represented by $x_i + \Delta x_i$ and $x_i - \Delta x_i$ respectively. These sections cover about 15% of the PDF each.

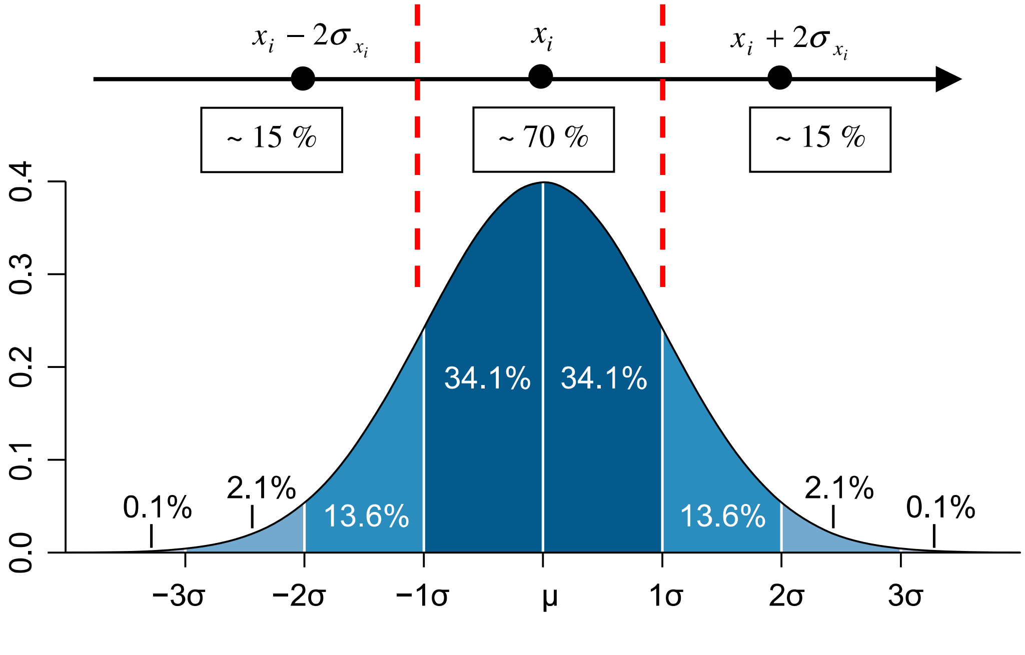


Figure 3.9: Confidence Intervals of a Normal Distribution (modified from (Kemp, 2005))

Multiple Point Method

The Multiple Point Method was introduced in (Lin, et al., 1999) and is based on the Sensitivity Index (SI) (Sundaresan, et al., 1995). In this method, the weighted average of the performance difference between the points x_i and $x_i \pm \Delta x_i$ is calculated as in Eq. 3.12 in order to estimate the level of robustness.

Eq. 3.12

$$\begin{aligned}
&\sum_{i=1}^{k}\left(\frac{(s_{i1})^2+(s_{i2})^2+w(s_{i3})^2}{3}\right)\\
&=\sum_{i=1}^{k}\left(\frac{(y_i-y_i^-)^2+(y_i-y_i^+)^2+w(y_i^- - y_i^+)^2}{3}\right)\\
&=\frac{1}{3}\sum_{i=1}^{k}\big(f(x_1,x_2,...,x_i,...,x_n)-f(x_1,x_2,...,(x_i-\Delta x_i),...,x_n)\big)^2\\
&+\frac{1}{3}\sum_{i=1}^{k}\big(f(x_1,x_2,...,x_i,...,x_n)-f(x_1,x_2,...,(x_i+\Delta x_i),...,x_n)\big)^2\\
&+\frac{1}{3}\sum_{i=1}^{k}w\big(f(x_1,x_2,...,(x_i-\Delta x_i),...,x_n)-f(x_1,x_2,...,(x_i+\Delta x_i),...,x_n)\big)^2
\end{aligned}$$

In Eq. 3.12, w is the weight for $(s_{i3})^2$. In (Lin, et al., 1999) it is set to 0.25 because it is expected that $s_{i3}=(y_i^- - y_i^+)=2(y_i-y_i^-)=2(y_i-y_i^+)=2s_{i1}=2s_{i2}$ and the three terms should contribute equally to the goal formulation. However, no suggestions are made on how to choose Δx_i. In this thesis, $2\sigma_{x_i}$ will be used as Δx_i. Assuming that the inputs are Gaussian distributed, this means that 95.6% of the input values lie between $x_i+\Delta x_i$ and $x_i-\Delta x_i$ (see Figure 3.9). Using $2\sigma_{x_i}$ for Δx_i means that using Eq. 3.8 will lead to an estimation of $4\sigma_y^2$ and therefore the equation is modified to:

$$
\begin{aligned}
\sigma_y^2 &\approx \sum_{i=1}^{k}\left(\frac{(s_{i1})^2+(s_{i2})^2+w(s_{i3})^2}{12}\right) \\
&= \sum_{i=1}^{k}\left(\frac{(y_i-y_i^-)^2+(y_i-y_i^+)^2+w(y_i^- - y_i^+)^2}{12}\right) \\
&= \frac{1}{12}\sum_{i=1}^{k}\left(f(x_1,x_2,...,x_i,...,x_n)-f(x_1,x_2,...,(x_i-\Delta x_i),...,x_n)\right)^2 \\
&+ \frac{1}{12}\sum_{i=1}^{k}\left(f(x_1,x_2,...,x_i,...,x_n)-f(x_1,x_2,...,(x_i+\Delta x_i),...,x_n)\right)^2 \\
&+ \frac{1}{12}\sum_{i=1}^{k} w\left(f(x_1,x_2,...,(x_i-\Delta x_i),...,x_n)-f(x_1,x_2,...,(x_i+\Delta x_i),...,x_n)\right)^2
\end{aligned}
\qquad \text{Eq. 3.13}
$$

with $\Delta x_i = 2\sigma_{x_i}$.

With Eq. 3.13, the value of the performance variance can be estimated and therefore this formulation can be used to substitute the old robustness formulation in RCEM.

In this section, three methods for estimating the robustness are introduced. Whether these methods have the potential to improve RCEM will be tested in the next section, where their performances are compared to Latin hypercube samplings (LHS).

3.2.3 Assessment of Alternative Methods

The three methods, which are introduced in the previous section, are assessed in this section using the example from Section 3.1 (Eq. 3.1). The ability of the three formulations to estimate the performance variance, and by this the robustness of a design solution, are compared in this section. Reference values are generated with the simulation of a Latin hypercube sample. The findings are confirmed by evaluating the effectiveness of the three robustness formulations in a modified RCEM framework. These results are again compared to those of Latin hypercube samples.

The goal is to identify the most promising formulation, which will be introduced to the modified RCEM. In order to have a good basis for comparison, the same tests as with the original RCEM in Section 3.1 are performed. Hereby, the original formulation for the mean will be used, without considering the nominal value shift, so that improvements can be clearly linked to the new robustness formulation.

In Section 3.1, it was discovered that the local minimum of the example function is the critical point of the performance function. The original RCEM fails in solving the design problem because its estimation of the performance variance at this point is wrong. This is why, in this section, the variance estimations of the three new methods in this point are compared to the estimation generated from a Latin hypercube sample with 100,000 sample points. For the comparison, Gaussian input distributions are assumed with standard deviations of 2 ($\sigma_x^2 = 4$) and 3 ($\sigma_x^2 = 9$) respectively. The test point is the local minimum at x = 931.95. In Table 3.5, the performance variance estimations σ_y^2 for the three new formulations and the normal distributed Latin hypercube sample are presented.

Table 3.5: Variance Estimation of New Formulations

Method	**Scenario 1:** $\sigma_x^2 = 4$	**Scenario 2:** $\sigma_x^2 = 9$
Latin Hypercube Sampling	$\sigma_y^2 = \mathbf{1.32}$	$\sigma_y^2 = \mathbf{7.52}$
Second Derivative Method	$\sigma_y^2 = 0.08$	$\sigma_y^2 = 0.41$
Multiple Derivative Method	$\sigma_y^2 = 3.19$	$\sigma_y^2 = 18.68$
Multiple Point Method	$\sigma_y^2 = \mathbf{1.67}$	$\sigma_y^2 = \mathbf{9.44}$

From the results, it can be seen that the Multiple Point Method returns the variance estimates that are closest to the Latin hypercube outcomes, although they are still not very accurate. Considering that the Multiple Point Method only needs three function calls, whereas the Latin hypercube sampling is done with 100,000 function calls, the Multiple Point Method seems to be a promising augmentation to the RCEM framework. The estimations of the Second Derivative Method are too low to be considered a good estimate. This is due to the formulation's similarity to the first order Taylor series approximation, which is equal to 0 at this point. The influence of the second derivative is too little to be able to correct this result. In the formulation of the Multiple Derivative Method, the influence of the derivatives at $x_i \pm 2\sigma$ lead to an estimate that is too high in both examples.

The results generated here are confirmed in the following. The three new formulations are used in a modified RCEM framework for the performance variance estimation. In the example an artificial robust design problem is solved with these methods. The cDSP for the example is given in Figure 3.10.

GIVEN

Performance function Eq. 3.1: $f(x)=\sum_{i=1}^{9} a_i (x-900)^{(i-1)}$

a$_1$ =	–	659.23	a$_4$ =		0.82691	a$_7$ =	–	3.2446 x 10-6
a$_2$ =		190.22	a$_5$ =	–	0.021885	a$_8$ =		1.6606 x 10-8
a$_3$ =	–	17.802	a$_6$ =		0.0003463	a$_9$ =	–	3.5757 x 10-11

Estimation for mean $\mu_{f(x)} = f(x)$ Eq. 3.2

Formulation for variance estimation $\sigma^2_{f(x)}$:

a) Second Derivative Method
b) Multiple Derivative Method
c) Multiple Point Method
d) Latin hypercube sampling

Variance of design variables $\sigma_x^2 = 4$

FIND

Design variable: x

Deviation variables: $d_i^-, d_i^+ \qquad i = 1,2$

SATISFY

GOALS

Performance goal: $\dfrac{1/f(x)}{1/10} + d_1^- - d_1^+ = 1$

Robustness goal: $\dfrac{1/\sigma^2_{f(x)}}{1/0.5} + d_2^- - d_2^+ = 1$

BOUNDS

$910 \le x \le 976$

$d_i^-, d_i^+ \ge 0 \qquad i = 1,2$

$d_i^- \cdot d_i^+ = 0 \qquad i = 1,2$

MINIMIZE

$Z = W_1 \cdot d_1^- + W_2 \cdot d_2^- \qquad d_1^+, d_2^+ = 0$

Scenario 1:	Scenario 2:
W$_1$ = 0.5	W$_1$ = 0.1
W$_2$ = 0.5	W$_2$ = 0.9

Figure 3.10: cDSP for Test of Robustness Formulations

The results generated using **d) Latin hypercube sampling** are presented first, in order to provide a basis for discussing the results of the three methods to be tested. The results are generated using 10,000 LHS points to determine the performance mean and variance. In Table 3.6 and Table 3.7, the results of the two scenarios are presented together with the estimated deviation variables and function values.

Table 3.6: Results for RCEM using the Latin Hypercube Sampling (Scenario 1)

Run #	**Initial Point**	**Solution**	**Deviation**		
			d_1^-	d_2^-	**Z**
1	x = 910	910	0.888254	0.984668	0.936461
2	x = 912	932.7186	0.320906	0.498915	0.409911
3	x = 920	932.7188	0.320907	0.498914	0.409911
4	x = 940	932.7178	0.320901	0.498921	0.409911
5	x = 960	932.7189	0.320908	0.498913	0.409911
6	x = 976	932.7183	0.320904	0.498917	0.409911

Table 3.7: Results for RCEM using the Latin Hypercube Sampling (Scenario 2)

Run #	**Initial Point**	**Solution**	**Deviation**		
			d_1^-	d_2^-	**Z**
1	x = 910	951.67	0.6486	0.2389	0.2804
2	x = 912	951.67	0.6486	0.2389	0.2804
3	x = 920	951.67	0.6486	0.2389	0.2804
4	x = 940	951.67	0.6486	0.2389	0.2804
5	x = 960	951.67	0.6486	0.2389	0.2804
6	x = 976	951.67	0.6486	0.2389	0.2804

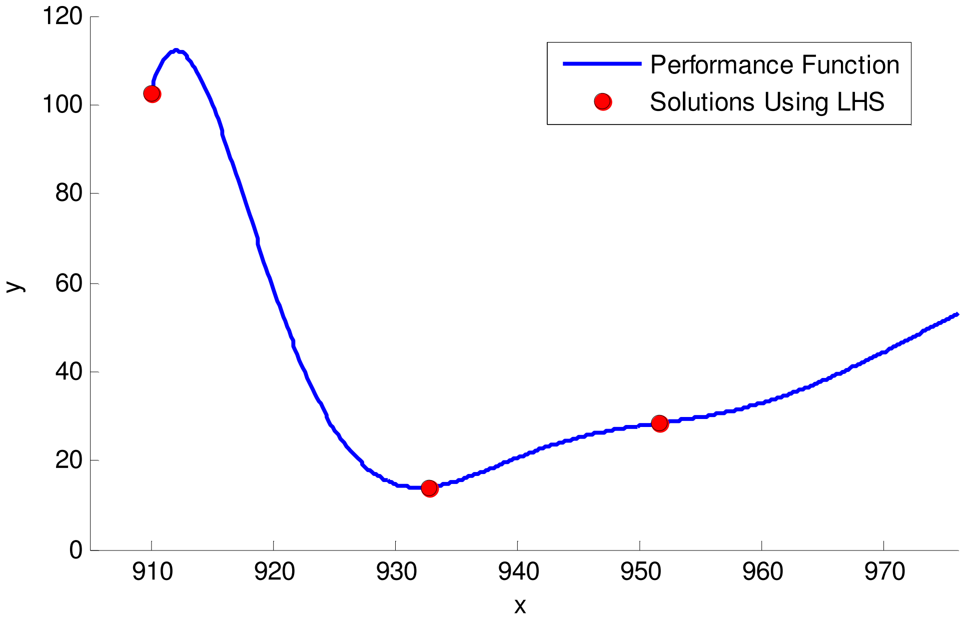


Figure 3.11: Location of the Solutions Using Formulation d) Latin Hypercube Sampling

The results are summarized in Figure 3.11, in which the locations of the solutions are shown. These results are taken as reference values for the remaining chapter. Using 10,000 samples per algorithm iteration on a single design variable, allows the assumption that these values are very similar to the actual values and results. From these results it can be seen that the location of the solution depends on how the designer formulates the preferences through the weighting parameters. However, it also depends on the setting of the goal values. If, for example, one goal value is set relatively far away from the actual feasible performance, its deviation will only change insignificantly compared to an objective, for which the goal is set close to the feasible area. This leads to a higher weight on the later objective, although the actual weighting parameters in the deviation function remain unchanged. It is very important that the designer is aware of this phenomenon when using the cDSP to formulate and solve engineering design problems.

In this example, Scenario 1 represents a situation where the designer assigns equal weights on performance and robustness, which results in a solution close to the minimum. This means that increasing the robustness would not outweigh the loss in performance when moving the solution towards the robust area. This is different in Scenario 2, where the focus is on finding a robust solution with only little weight on performance. In this case, the gain in robustness in the robust area is valued higher than the loss of performance. The outlier at x = 910 can be explained with an unfortunate selection of the starting point; the maximum at around x = 912 prevents the exploration algorithm from finding the points on its right side. This can be seen in for the following examples as well.

In the following tests, the three approximation methods are assessed using the same example and identical scenarios. This will show which methods are capable of suggesting the best compromises for each scenario. In Scenario 1, the methods should suggest a solution close to the minimum at around x = 932 and in Scenario 2, a solution around x = 952 would confirm the capability of the method.

The results for using RCEM with variance goal formulation **a) Second Derivative Method** are given in Table 3.8 and Table 3.9.

From the results it can be seen that the Second Derivative Method (second order Taylor Series) does not lead the designer to the robust area (see Figure 3.12) in the second Scenario. Using a modified RCEM with the Second Derivative Method as the robustness formulation leads to the same incorrect results as the original RCEM.

Table 3.8: Results for RCEM using the Second Derivative Method (Scenario 1)

Run #	Initial Point	Solution	Estimated Deviation		Actual Deviation	
			d_1^-	d_2^-	d_1^-	d_2^-
1	x = 910	910	0.9023	0.9991	0.8883	0.9995
2	x = 912	910	0.9023	0.9991	0.8883	0.9995
3	x = 920	932.14	0.2841	0.1049	0.3201	0.5762
4	x = 940	932.14	0.2841	0.1049	0.3201	0.5762
5	x = 960	932.14	0.2841	0.1049	0.3201	0.5762
6	x = 976	932.14	0.2841	0.1049	0.3201	0.5762

Table 3.9: Results for RCEM using the Second Derivative Method (Scenario 2)

Run #	Initial Point	Solution	Estimated Deviation		Actual Deviation	
			d_1^-	d_2^-	d_1^-	d_2^-
1	x = 910	932.14	0.2841	0.1049	0.3201	0.5762
2	x = 912	932.14	0.2841	0.1049	0.3201	0.5762
3	x = 920	932.14	0.2841	0.104	0.3201	0.5762
4	x = 940	932.14	0.2841	0.1049	0.3201	0.5762
5	x = 960	932.14	0.2841	0.1049	0.3201	0.5762
6	x = 976	932.14	0.2841	0.1049	0.3201	0.5762

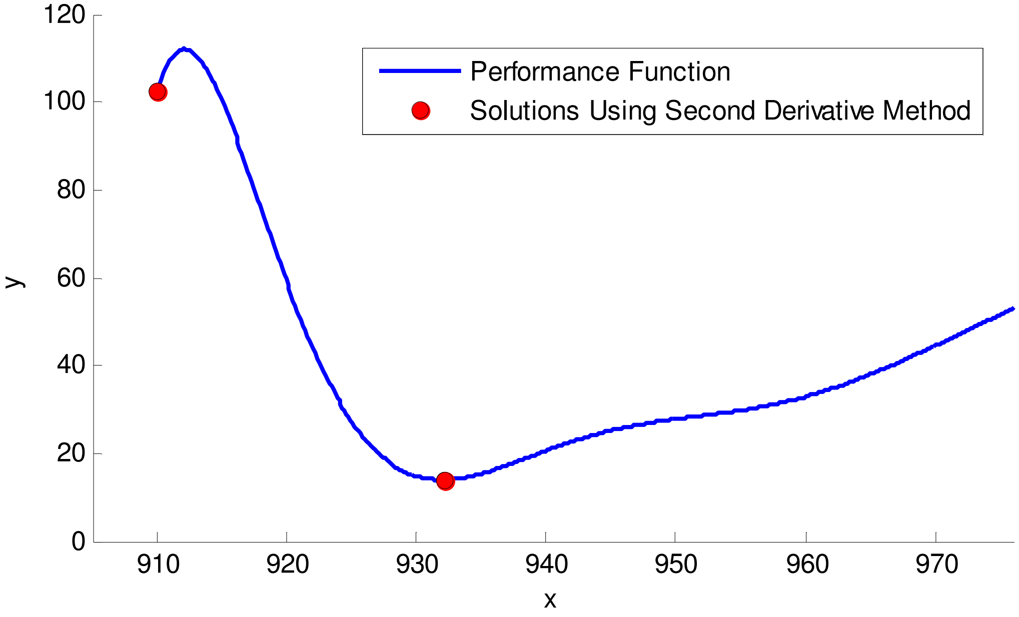


Figure 3.12: Location of the Solutions Using Formulation a) Second Derivative Method

From the very small values of d_2^- it can be seen that the Second Derivative Method gives a very low estimate of the performance variance. This is because the mathematical formulation that is derived from the second order Taylor Series approximation is very similar to the original formulation in RCEM (first order Taylor Series). Note that the second derivative in Eq. 3.10 only has a weight of 0.25 and the first two terms are identical to the variance estimation used in the original RCEM (Eq. 3.3). Thus, the additional terms only have very little influence on the performance variance estimation and the formulation fails for the same reason as the first order Taylor Series.

The results for using RCEM with variance goal formulation **b) Multiple Derivative Method** are given in Table 3.10 and Table 3.11.

Table 3.10: Results for RCEM using the Multiple Derivative Method (Scenario 1)

Run #	Initial Point	Solution	Estimated Deviation		Actual Deviation (LHS)	
			d_1^-	d_2^-	d_1^-	d_2^-
1	x = 910	910	0.9023	0.9997	0.8883	0.9995
2	x = 912	910	0.9023	0.9997	0.8883	0.9995
3	x = 920	951.62	0.6484	0.2724	0.6484	0.2392
4	x = 940	951.62	0.6484	0.2724	0.6484	0.2392
5	x = 960	951.62	0.6484	0.2724	0.6484	0.2392
6	x = 976	951.62	0.6484	0.2724	0.6484	0.2392

Table 3.11: Results for RCEM using the Multiple Derivative Method (Scenario 2)

Run #	Initial Point	Solution	Estimated Deviation		Actual Deviation (LHS)	
			d_1^-	d_2^-	d_1^-	d_2^-
1	x = 910	951.69	0.6487	0.2722	0.6487	0.2392
2	x = 912	951.69	0.6487	0.2722	0.6487	0.2392
3	x = 920	951.69	0.6487	0.2722	0.6487	0.2392
4	x = 940	951.69	0.6487	0.2722	0.6487	0.2392
5	x = 960	951.69	0.6487	0.2722	0.6487	0.2392
6	x = 976	951.69	0.6487	0.2722	0.6487	0.2392

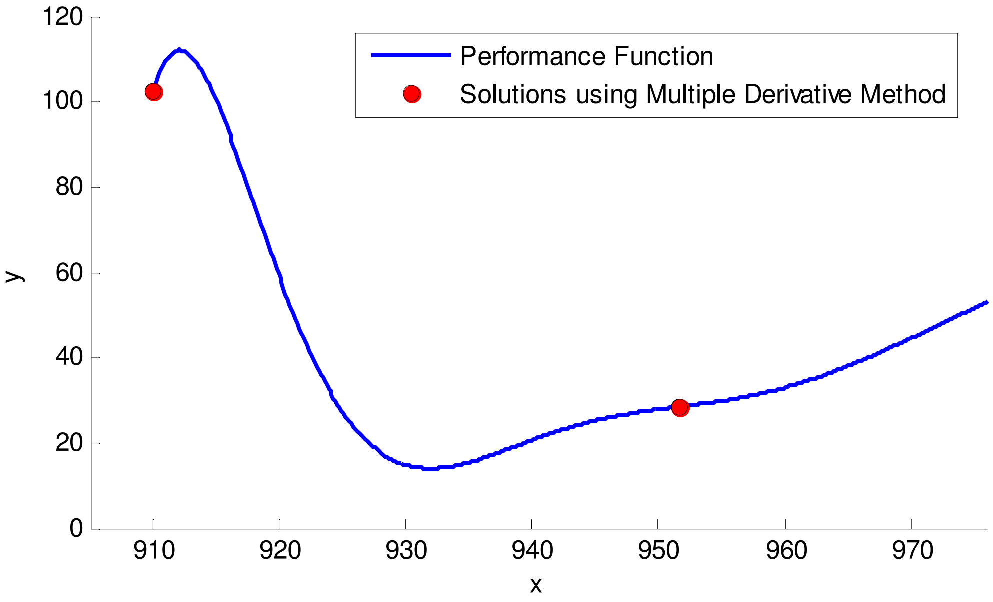


Figure 3.13: Location of the Solutions Using Formulation b) Multiple Derivative Method

In Figure 3.13, it is shown that, by using the Multiple Derivative Method, the designer would be led to solutions in the robust area in both scenarios. This means that the method leads to an estimation of the variance in the robust area that is lower than the actual variance. Furthermore, around the minimum, the variance is estimated to be higher than the actual performance variance, which makes this method useful if the designer puts a high weight on the robustness when looking for a compromise.

The results for using RCEM with variance goal formulation **c) Multiple Point Method** are given in Table 3.12 and Table 3.13.

Table 3.12: Results for RCEM using the Multiple Point Method (Scenario 1)

Run #	Initial Point	Solution	Estimated Deviation		Actual Deviation (LHS)	
			d_1^-	d_2^-	d_1^-	d_2^-
1	x = 910	910	0.9023	0.9996	0.8883	0.9995
2	x = 912	910	0.9023	0.9996	0.8883	0.9995
3	x = 920	932.96	0.2331	0.5633	0.3228	0.5074
4	x = 940	932.96	0.2331	0.5633	0.3228	0.5074
5	x = 960	932.96	0.2331	0.5633	0.3228	0.5074
6	x = 976	932.96	0.2331	0.5633	0.3228	0.5074

Table 3.13: Results for RCEM using the Multiple Point Method (Scenario 2)

Run #	Initial Point	Solution	Estimated Deviation		Actual Deviation (LHS)	
			d_1^-	d_2^-	d_1^-	d_2^-
1	x = 910	951.69	0.6487	0.2757	0.6487	0.2391
2	x = 912	951.69	0.6487	0.2757	0.6487	0.2391
3	x = 920	951.69	0.6487	0.2757	0.6487	0.2391
4	x = 940	951.69	0.6487	0.2757	0.6487	0.2391
5	x = 960	951.69	0.6487	0.2757	0.6487	0.2391
6	x = 976	951.69	0.6487	0.2757	0.6487	0.2391

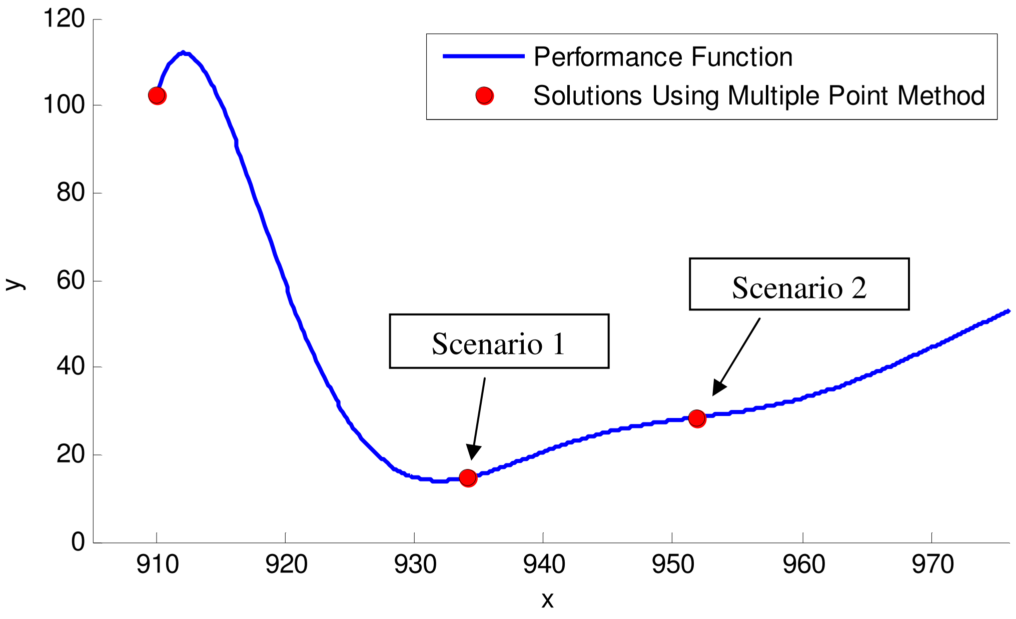


Figure 3.14: Location of the Solutions Using Formulation c) Multiple Point Method

The results from using RCEM with the robustness formulation c) look very promising. When comparing Figure 3.14 to Figure 3.11, it can be seen that the solution for both scenarios lie very close to the solutions generated with LHS. The method leads to a good estimation of the variance in the robust area as well as in the area around an optimum. This can be attributed to the mathematical formulation of this method, which accounts for nonlinearity by evaluating the performance differences between the mean and the tail ends of the input distribution as well as the difference between both tail ends. Particularly, this second aspect leads to good variance estimation around local optima.

3.2.4 Critical Evaluation of Results and Introduction of the Modified RCEM

In this paragraph, the findings of Sections 3.1 and 3.2 are summarized and the improved RCEM is introduced. Three major limitations of the original RCEM were identified and improvements were suggested (see Table 3.14).

Table 3.14: Improvements to RCEM

Limitation of the Original RCEM	Suggestion for Improvements
Inaccuracy of global response surface	RCEM should only be used if performance function is known explicitly
Inaccurate estimation of the performance mean	New formulation for the estimation of the mean is introduced
Inaccurate estimation of the performance variance	New formulation for the estimation of the variance is introduced

One suggestion is that the original or modified RCEM should only be applied if the performance functions are available explicitly, i.e. the mathematical function is known. Generating a global response surface can lead to fatal errors and ultimately to an engineering solution that is far away from being satisfactory. In cases where the explicit performance functions are not known, the designer should refer to simulation-based design methods that generate local surrogate models in each iteration of the exploration algorithm. An example of such a method is developed later in this thesis beginning in Chapter 5.

For the estimation of the mean in RCEM, the nominal value shift should be considered if the performance functions are known and differentiable, since more accurate estimations can be achieved with this formulation.

For the formulation of the robustness, it was shown that the simplest of the three formulations generates the most accurate results. For the modified RCEM it is suggested to use the Multiple Point Method for estimating the performance robustness. This method has the advantage that no derivative is needed. The framework of the modified RCEM is shown in Figure 3.15.

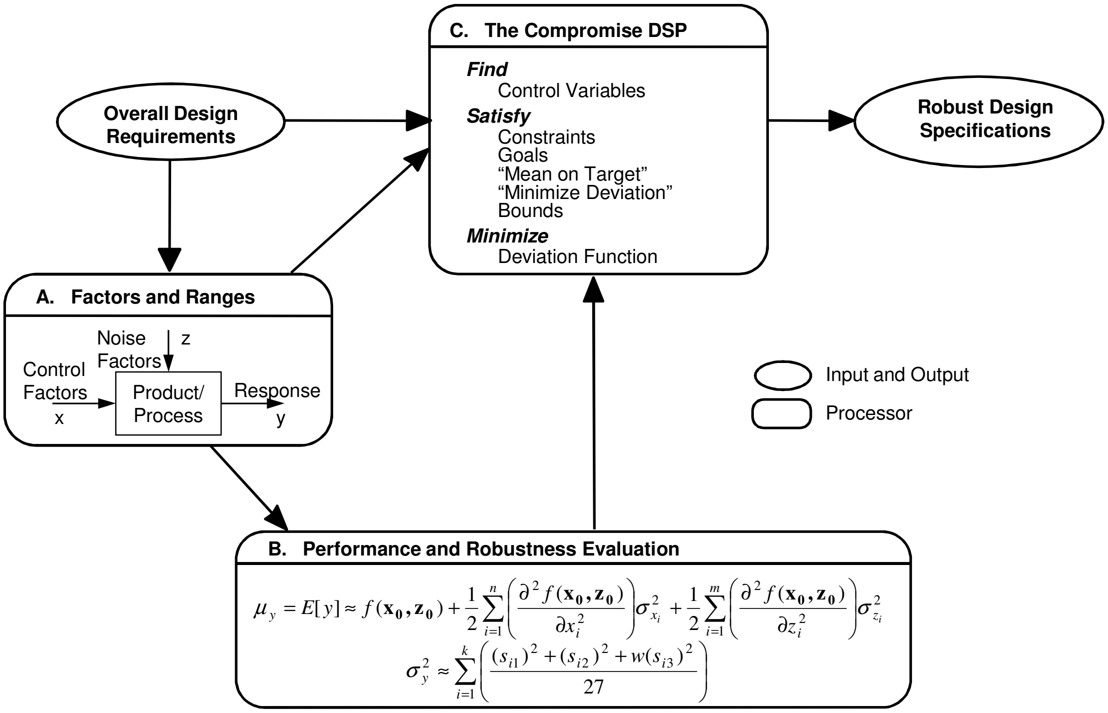

C. The Compromise DSP
Find
Control Variables
Satisfy
Constraints
Goals
"Mean on Target"
"Minimize Deviation"
Bounds
Minimize
Deviation Function
Overall Design Requirements
Robust Design Specifications
A. Factors and Ranges
Noise Factors
z
Control Factors
x
Product/ Process
Response
y
Input and Output
Processor
B. Performance and Robustness Evaluation
$\mu_y = E[y] \approx f(\mathbf{x_0},\mathbf{z_0}) + \frac{1}{2}\sum_{i=1}^{n}\left(\frac{\partial^2 f(\mathbf{x_0},\mathbf{z_0})}{\partial x_i^2}\right)\sigma_{x_i}^2 + \frac{1}{2}\sum_{i=1}^{m}\left(\frac{\partial^2 f(\mathbf{x_0},\mathbf{z_0})}{\partial z_i^2}\right)\sigma_{z_i}^2$
$\sigma_y^2 \approx \sum_{i=1}^{k}\left(\frac{(s_{i1})^2 + (s_{i2})^2 + w(s_{i3})^2}{27}\right)$

Figure 3.15: Framework of the Modified RCEM

3.3 Guide for Validation of the First Research Questions

The first research hypothesis can be verified by validating its two parts:

Part 1: Using a global response surface, the nominal mean and a single point Taylor Series Approximation for the variance estimation is not accurate enough for finding robust solutions of nonlinear design problems.

Part 2: The Robust Concept Exploration Method can be improved by using appropriate formulations that account for the nonlinearity of the problem.

The research focus is clearly on the second part of the hypothesis. The first part of Hypothesis I is validated in Section 3.1, where the limitations of RCEM are demonstrated. To validate the second part, a modified RCEM is introduced in Section 3.2. For validating this method, the Validation Square is employed (see Section 1.4).

3.3.1 Theoretical Structural Validity

The internal consistency of the individual constructs of the improved RCEM is shown in the previous sections and is summarized here.

The main components of the new method are:

- cDSP
- Mean estimation accounting for the nominal value shift
- Variance estimation using Taylor Series expansion on multiple points

The cDSP has been validated in literature before (Bras and Mistree, 1993), so the internal consistency does not have to be shown here. The internal consistency of the new performance mean and variance estimation has been shown in Section 3.2. This was done by explaining the mathematical and statistical backgrounds. Furthermore, the formulations have been tested in a simple example. The results of these tests clearly supported the internal consistency of the components.

3.3.2 Empirical Structural Validity

In order to demonstrate empirical structural validity, the appropriateness of the examples used for illustrating and verifying the method has to be shown. The improved method, introduced previously in this chapter, is validated using two examples. First, in Section 3.2 a simple example is used to assess different options and also to validate the new method. This example, although it is artificial, is representative of an engineering design problem with a nonlinear performance function. The designer's goal is to find a satisfying compromise between robustness and high performance. The second example, which is the design of a robust pressure vessel, is introduced in Chapter 4. This example is also representative for a typical robust engineering design problem. The modified RCEM can be applied to this problem because the performance functions are known explicitly. It is a multivariate problem with several performance and robustness objectives that require the designer to find a compromise. Further details about the example are given in the next chapter. So both examples, which are used to validate the new method, are appropriate and representative for engineering problems the method is developed for.

3.3.3 Empirical Performance Validity

The empirical performance validity of the design method is shown through a critical analysis of the results from the two examples. In Section 3.2.3 it is demonstrated that the method using the new formulations is clearly superior because it is able to solve a design problem the old method is not able to solve correctly. In Chapter 4, the results of the second design example are critically evaluated. The accuracy of the method is shown by a comparison with results generated using Latin hypercube sampling.

3.3.4 Theoretical Performance Validity

The theoretical performance validity of the proposed design method will be discussed in Chapter 7 based on the performance, characteristics and limitations of the method and its components.

3.4 Chapter Synopsis

As shown in Figure 3.16, the focus in this chapter is on elaborating Hypothesis 1. This includes a critical analysis of the characteristics and limitations of the original RCEM as well as the development of formulations that improve RCEM. These advancements are tested and critically evaluated. Furthermore the validation strategy for the new method is elaborated. In the next chapter, the method introduced in this chapter will be applied to another engineering design problem. The design of a robust pressure vessel will be used to support the validity of the method and eventually Hypothesis 1 can be verified.

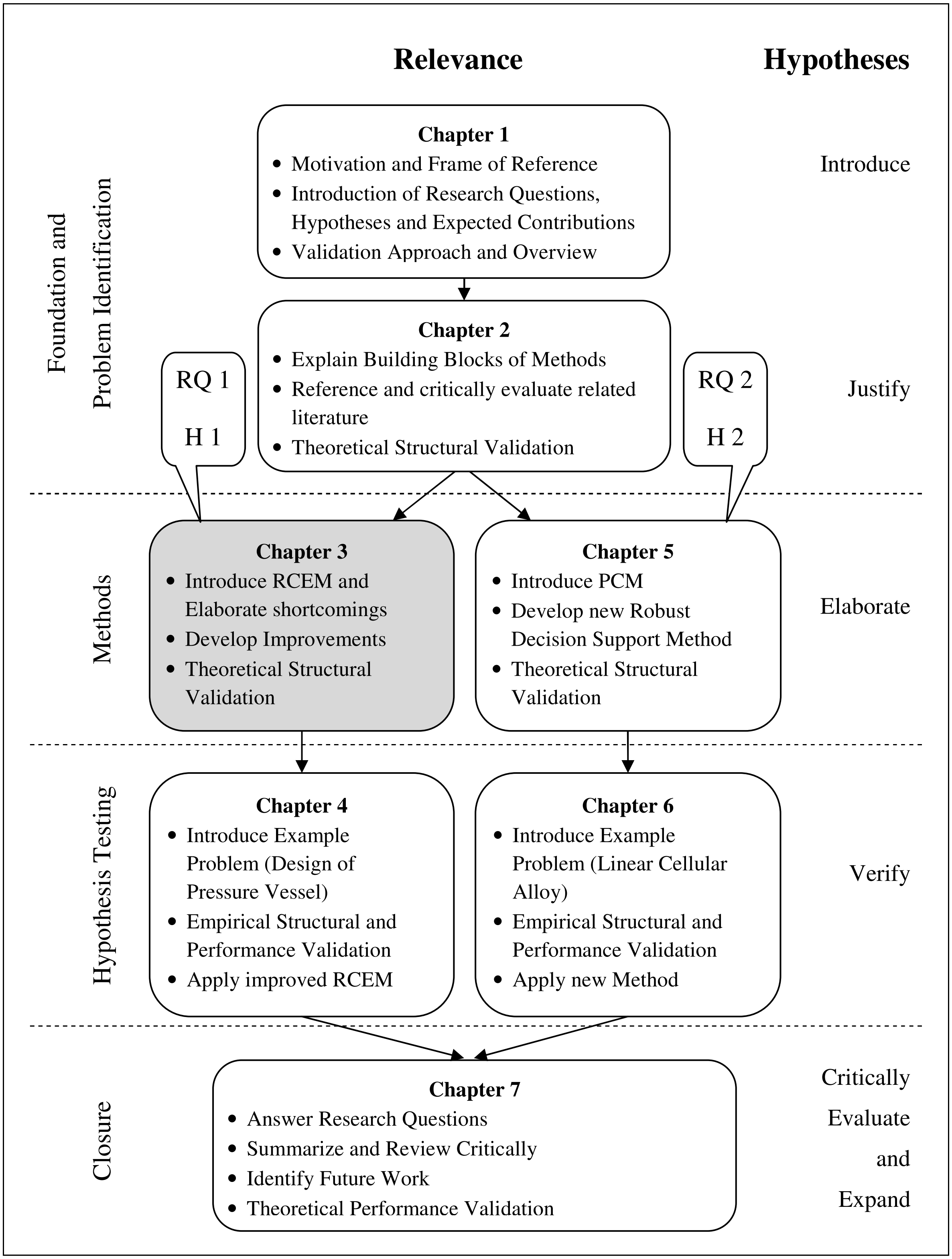

Relevance
Hypotheses
Foundation and Problem Identification
Chapter 1
Motivation and Frame of Reference
Introduction of Research Questions, Hypotheses and Expected Contributions
Validation Approach and Overview
Introduce
Chapter 2
Explain Building Blocks of Methods
Reference and critically evaluate related literature
Theoretical Structural Validation
RQ 1
H 1
RQ 2
H 2
Justify
Methods
Chapter 3
Introduce RCEM and Elaborate shortcomings
Develop Improvements
Theoretical Structural Validation
Chapter 5
Introduce PCM
Develop new Robust Decision Support Method
Theoretical Structural Validation
Elaborate
Hypothesis Testing
Chapter 4
Introduce Example Problem (Design of Pressure Vessel)
Empirical Structural and Performance Validation
Apply improved RCEM
Chapter 6
Introduce Example Problem (Linear Cellular Alloy)
Empirical Structural and Performance Validation
Apply new Method
Verify
Closure
Chapter 7
Answer Research Questions
Summarize and Review Critically
Identify Future Work
Theoretical Performance Validation
Critically Evaluate and Expand

Figure 3.16: Thesis Roadmap

CHAPTER 4

MULTI-OBJECTIVE ROBUST DESIGN OF A PRESSURE VESSEL

Using the modified RCEM, it is possible to consider the robustness of a system as an objective in the decision making process during the conceptual design phase. In this chapter, it will be shown that the modified RCEM has the capability to solve multidimensional robust design problems with multiple objectives. When applying robust design methods, the designer wants to find a compromise between high performance and low sensitivity of the performance towards uncertainty in design variables and noise factors.

The effectiveness of the modified RCEM is demonstrated in this chapter with the design of a pressure vessel as an example. The results of this example are used to support Research Hypothesis 1.

After an introduction to the details of the pressure vessel, the validation and verification strategy for this example is explained. The results of the modified RCEM applied to the example will then be used to support the presentation of the empirical structural validation as well as the empirical performance validation of the method. A critical review of the results and an outlook to the next chapters of the thesis conclude Chapter 4.

4.1 The Robust Design of a Pressure Vessel

The purpose of this example is the validation of the modified RCEM and thereby, the validation of Hypothesis 1. With the example, it is intended to demonstrate the effectiveness of the method for multi-objective robust engineering design problems. The design of a pressure vessel has been investigated previously by authors in several areas of engineering design (Hernández, 2001; Lewis and Mistree, 1998; Sandgren, 1990; Simpson, 1998). It is very suitable for showing the effectiveness of the modified RCEM because it fulfills the following requirements:

- all performance functions are known explicitly; no simulations are required
- the design has multiple design variables
- the design has multiple objectives

The design variables in this example are the radius of the vessel (R), the length of the middle section (L) and the thickness of the vessel wall (T). A schematic of the pressure vessel is given in Figure 4.1.

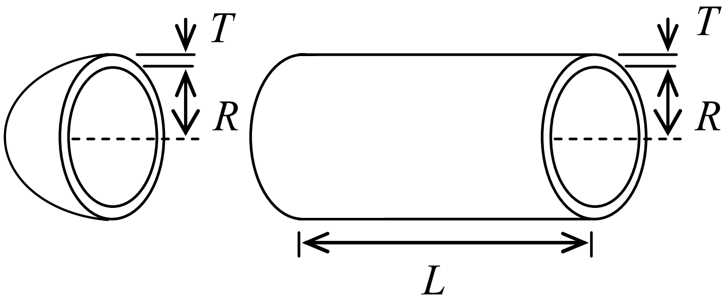


Figure 4.1: Schematic of Cylindrical Pressure Vessel (Hernández, 2001)

The symbols used in this chapter are introduced in Table 4.1.

Table 4.1: Nomenclature for Chapter 4

V	Volume of the pressure vessel [*in*^{3}]
W	Weight [*lb*]
R	Radius [*in*]
L	Length of middle section [*in*]
T	Thickness of wall [*in*]
$\sigma_R^2 = 4/9$ $\sigma_L^2 = 4/9$ $\sigma_T^2 = 1/9$	Variances of R, L and T [*in*^{2}]
σ_V^2, σ_W^2	Performance variances of V and W [*in*^{2}]
μ_{V}	Performance mean of V
$P = 1000$	Internal pressure [*psi*]
$SF = 2$	Safety factor
$\rho = 0.284$	Material density [*lb/in*^{3}]
s_y= 26,000	Material yield strength [*psi*]
s_s	Maximum stress in spherical section [*psi*]
s_c	Maximum stress in cylindrical section [*psi*]

The application requirements demand a vessel that is able to contain a gas under 1,000 *psi* of pressure with a minimum volume of 20,000 *in*^{3} and a maximum vessel weight of 2,500 *lb*. The material is UNS G10100 HR, with a yield strength of 26,000 *psi* and a density of 0.284 *lb/in*^{3}.

The bounds on the radius are 6 inch and 36 inch respectively (Eq. 4.1). The middle section of the vessel has to be between 6 inch and 48 inch (Eq. 4.2) and the thickness between 1.5 inch and 6 inch (Eq. 4.3).

$$6.0 \le R \le 36.0 \; in \qquad \text{Eq. 4.1}$$

$$6.0 \le L \le 48.0 \; in \qquad \text{Eq. 4.2}$$

$$1.5 \le T \le 6.0 \; in \qquad \text{Eq. 4.3}$$

Due to space limitations, the overall radius is required to be less than 40 inch (Eq. 4.4) and the overall length less than 60 inch (Eq. 4.5).

$$R + T \le 40 \; in \qquad \text{Eq. 4.4}$$

$$L + 2R + 2T \le 60 \; in \qquad \text{Eq. 4.5}$$

Furthermore, the radius has to be at least 5 times greater than the thickness of the vessel.

$$R \ge 5T \qquad \text{Eq. 4.6}$$

Due to inaccuracy and uncertainty in the manufacturing process, deviations in the input values of the design variables can be observed. For this example, these deviations are assumed to be Gaussian. The variance of the radius and length of the middle section are 0.444 in^2; the variance of the thickness is 0.111 in^2.

Industry regulations specify that the design has to comply with a minimum safety factor, *SF*, of 2 for the maximum stress appearing in both the cylindrical and spherical sections of the vessel (Eq. 4.7 and Eq. 4.8) (see (Roark and Young, 1975)).

$$s_c = P\left[\frac{(R+T)^2 + R^2}{(R+T)^2 - R^2}\right] = P\left[\frac{2R^2 + 2RT + T^2}{2RT + T^2}\right] \le \frac{s_y}{SF} \qquad \text{Eq. 4.7}$$

$$s_s = P\left(\frac{R + 0.5T}{2T}\right) \le \frac{s_y}{SF} \qquad \text{Eq. 4.8}$$

The basis for the performance objectives, in this example, are the volume (Eq. 4.9) and the weight (Eq. 4.10) of the pressure vessel.

$$V(R,L) = \pi\left(\frac{4}{3}R^3 + R^2 L\right) \ \ [in^3] \qquad \text{Eq. 4.9}$$

$$W(R,L,T) = \pi\rho\left[\frac{4}{3}(R+T)^3 + (R+T)^2 L - \left(\frac{4}{3}R^3 + R^2 L\right)\right] \ \ [lb] \qquad \text{Eq. 4.10}$$

The design objectives (goals) in this example are:

- High vessel volume mean μ_{V} (as close to 50,000 in^3 as possible)
- Low vessel volume variance σ_V^2 (as close to 5,000,000 in^6 as possible)
- Low vessel weight variance σ_W^2 (as close to 200,000 lb^2 as possible)

Only the aforementioned conditions, bounds and constraints are considered in this design example and a constraint violation of 1% is allowed, which is the default setting in JavaDSIDES. All other sources of variability like temperature and material imperfections are ignored.

Prior to the development of the cDSP formulation for this design problem, which is described in Section 4.3, the validation strategy for this example is outlined in the next section.

4.2 Validation and Verification with the Example

The pressure vessel example is intended to provide evidence for the validity of Hypothesis 1 (see Section 1.3) by demonstrating that the modified RCEM is an effective and efficient method to solve robust engineering design problems. The research presented in this thesis is validated along the four quadrants of the validation square (Figure 1.6). In this section, the empirical structural validation is discussed and a plan for the empirical performance validation is outlined.

4.2.1 Empirical Structural Validation

Empirical structural validation of Hypothesis 1 involves documenting that the example is actually similar to design problems for which the modified RCEM is intended and may be applied. Furthermore, it has to be shown that the data associated with the example can be used to support the hypothesis.

Is this example similar to design problems for which the modified RCEM is intended and may be applied?

The following characteristics of the pressure vessel example qualify it as a suitable problem for the modified RCEM:

- The design of the pressure vessel is determined by three design variables. This means that the example is a multidimensional design problem that requires a method that can handle multiple input variables. In this example the three dimensions are the radius, the length and the wall thickness.

- The performance of the design is dependent on several uncertain factors. In this case, the three design variables contain uncertainty in form of manufacturing inaccuracy. Showing that the modified RCEM is capable of handling multiple uncertain design variables implies that the method can also solve problems with multiple noise factors. For the uncertainty analysis, noise factors can be regarded as uncertain design variables with a constant mean value.

- The objective is to design a pressure vessel that is robust towards uncertain design variables or noise factors. This means that the goal is to find a design variable configuration, which results in a low variation of the volume and the weight of the vessel without changing or modifying the source of uncertainty. A method that can help solving this kind of problems requires a means to estimate accurately the robustness of the design at various points in the design space.

- The example design problem has multiple design objectives that have to be considered by the designer. The goal is to find a compromise among the several objectives. This example has one performance objective (high vessel volume) and two robustness objectives (low variances of volume and weight). Since these objectives oppose each other, a trade-off has to be found that satisfies the designer's preferences.

Can the data from this example be used to support conclusions with respect to the hypothesis?

The data for this example is generated using the Java implementation of DSIDES, which is introduced in 2.4. The results of the modified RCEM are compared to results of the original RCEM and to results of a brute force Latin hypercube simulation. This comparison will be used to demonstrate the effectiveness and efficiency of the modified RCEM in solving multidimensional and multi-objective robust design problems. Furthermore, the generated data includes estimations of the values for all three objectives. These will be used to support the hypothesis that the modified RCEM predicts the performance and variances more accurately than the original method. Finally, the results will also be compared to the outcome of a traditional single-objective optimization of the problem. This comparison will support the hypothesis that the modified RCEM is capable of finding a compromise between high volume and low variances of volume and weight.

4.2.2 Empirical Performance Validation

The pressure vessel example is also used to demonstrate the empirical performance validation of the modified RCEM and thus Hypothesis 1. Therefore, it has to be shown (1) that the results meet the expectations, (2) that the demonstrated effectiveness and efficiency is linked to the application of the new method and (3) that the data generated and used for validation is accurate and internally consistent.

How can the results be evaluated with respect to the expected outcome of the method?

The results of the pressure vessel example will be generated for two different design scenarios (Table 4.2). Scenario 1 is equivalent to a traditional design approach where the goal is to maximize the performance without considering uncertainty. The second scenario demands for a robust design method that is able to compromise performance and robustness objectives. The expected outcome is that using the modified RCEM on Scenario 2, a design can be found that is less sensitive towards uncertainty than a solution found with a traditional maximizing design method (Scenario 1). Since in Scenario 2 a compromise is intended, the volume will be smaller than in Scenario 1, however, the variances of the volume and the weight are expected to be smaller than in Scenario1, too. Meeting these expectations, the results support Hypothesis 1.

Table 4.2: Design Scenarios for the Empirical Performance Validation

Scenario 1	Scenario 2
The design objective is to **maximize the volume** of the pressure vessel under the given constraints and bounds.	The goal is to design a **robust** pressure vessel with **high volume and low variances of the volume and weight**. A compromise among the three design objectives has to be found, which satisfies the designer's preferences.

How can it be demonstrated that the observed effectiveness and efficiency is linked to the application of the modified RCEM?

In order to show the link between the effectiveness and efficiency and the application of the modified RCEM, the design problem scenario 2 is solved using the original RCEM, too, and both results are compared to the outcomes of a brute force Latin hypercube approach, which is used as a reference. If the results of the modified RCEM are of higher quality, which means they are closer to the outcome of the reference method, Hypothesis 1 is supported. The efficiency of the new method is demonstrated by comparing the computational costs of the modified RCEM to the original RCEM as well as the LHS approach.

How are the accuracy and the internal consistency of the results verified?

First, the accuracy of the results is verified by comparing the outcome of the modified RCEM to those of a Latin hypercube approach, which is taken as reference. Using a computationally expensive Latin hypercube sampling within the DSIDES algorithm to determine performance and robustness is assumed to give results very close to reality. Showing that the results of the modified RCEM are close to those values will support the argument for the accuracy of the method. The internal consistency of the results is confirmed by testing multiple starting points for the algorithm as well as by an analysis of the convergence of the algorithm.

4.3 Application of the Modified RCEM to the Design of a Pressure Vessel

The modified RCEM is developed and introduced in detail in CHAPTER 3. In this section, the new method is applied to the pressure vessel example. In Section 4.3.1, the design task is clarified and the cDSP is formulated; in Section 4.3.2, the modified RCEM is applied to the problem and the results are presented. For comparison, the original RCEM is applied to the example in Section 4.3.3. Furthermore, in this section the accuracy of the results is verified. The data associated with this example is used to support Hypothesis 1 in Section 4.4.

4.3.1 Clarification and Formulation of the Design Problem

The details of the design problem are described in Section 4.1. In this section, the compromise Decision Support Problems (cDSPs) for both design scenarios (see Table 4.2) are formulated and presented. Formulating a cDSP is a neat way for the designer to summarize the design problem. By filling in the information into the four sections of the cDSP, “Given”, “Find”, “Satisfy”, and “Minimize”, the designer is guided through the definition of the design problem. This way, it is likely that the designer will realize if the information or data is incomplete. The cDSP formulations are the basis for applying RCEM. In Figure 4.2, the cDSP for the pressure vessel example is presented. In the first section of the cDSP, all the information about the pressure vessel is summarized. For this example all details are **given** in Section 4.1. This includes the model of the vessel, all constants and variable definitions as well as the constraint and goal functions. Furthermore, the assumptions made for this example (see Section 4.1) are stated here. In the next sections of the cDSP, the variables are identified for which the designer has to **find** the values. These are the design variables *R*, *L* and *T* as well as the deviation variables *d*, which are used in the goal formulations. In the third section of the cDSP, all constraints, bounds and goal formulations are listed that have to be **satisfied**. Finally, the deviation function *Z* is presented, which has to be **minimized**. By assigning weights to the deviations from the individual targets, this function is used to formulate the compromise. In this example, Scenario 1 represents the optimization for maximum volume, which is equivalent to assigning a weight of 1.0 to the volume goal and weights of 0.0 to the robustness goals. This way, only the deviation to the volume goal is minimized. In Scenario 2, the designer is asked to find a robust pressure vessel. This means that the designer is looking for a compromise among the three goals. This is realized by assigning the following weights to the three deviation variables: $W_1 = 0.45,\ \ W_2 = 0.25,\ \ W_3 = 0.30$. The mindset behind choosing these values is that the robustness should have a larger weight (0.25+0.30=0.55) than the performance (0.45).

In the following section, it is described and explained how the modified RCEM is implemented to solve the presented cDSP.

Given

Assumptions for the design of the pressure vessel as presented in Section 4.1.
Nomenclature, constants and design schematic as given in Table 4.1 and Figure 4.1.
System constraint functions as given in Eq. 4.4 to Eq. 4.8 and below.
System goal functions:

Mean of *V*: μ_V (see Section 4.3.2)

Variances of *V* and *W*: σ_V^2, σ_W^2 (see Section 4.3.2)

Goal target values:

High vessel volume mean (as close to 50,000 in^3 as possible)
Low vessel volume variance (as close to 5,000,000 in^6 as possible)
Low vessel weight variance (as close to 200,000 lb^2 as possible)

Find

R	Radius [*in*]
L	Length of middle section [*in*]
T	Thickness of wall [*in*]
$d_i^-, d_i^+ \quad i = 1,2,3$	Deviation variables

Satisfy

Constraints:

$R + T \le 40\ in$	see Eq. 4.4
$L + 2R + 2T \le 60\ in$	see Eq. 4.5
$R \ge 5T$	see Eq. 4.6
$s_c \le \frac{s_y}{SF}, \qquad s_s \le \frac{s_y}{SF}$	see Eq. 4.7 and Eq. 4.8

Goals (normalized):

$$\frac{\mu_V(R,L)}{50{,}000 in^3} + d_1^- - d_1^+ = 1$$

$$\frac{5{,}000{,}000 in^6}{\sigma_V^2} + d_2^- - d_2^+ = 1 \qquad \frac{200{,}000 lb^2}{\sigma_W^2} + d_3^- - d_3^+ = 1$$

Bounds:

$6.0 \le R \le 36.0\ in$

$6.0 \le L \le 48.0\ in$

$1.5 \le T \le 6.0\ in$

$d_i^- \cdot d_i^+ = 0\ ;\ d_i^-, d_i^+ \ge 0 \qquad i = 1,2,3$

Minimize

$$Z = W_1 d_1^- + W_2 d_2^- + W_3 d_3^-$$

Scenario 1:
$W_1 = 1,\ W_2 = 0,\ W_3 = 0$
Scenario 2:
$W_1 = 0.45,\ W_2 = 0.25,\ W_3 = 0.30$
(see Table 4.2)

Figure 4.2: cDSP for the Robust Design of the Pressure Vessel

4.3.2 Implementation of the Modified RCEM

In this section, the modified RCEM is applied to the pressure vessel example which is formulated in the previous section. As presented in Chapter 3 and in Figure 3.15, the modified RCEM has two major components, the cDSP as well as the performance and robustness evaluation. The cDSP is the framework, in which the design problem is formulated and the deviation function is defined. Since the modified RCEM is developed for robust design applications, there is at least one uncertain design variable or noise factor in the design problem. In order to determine the deviation function value of a design, the performance mean(s) and performance variance(s) have to be estimated under this uncertainty. As noted before, this part is the focus of this thesis. After having formulated the cDSP for the design example, in this section, the estimation of the performance and the robustness of the pressure vessel is explained. Furthermore, the cDSP is solved in this section using JavaDSIDES, which is introduced in Section 2.4. The results are then analyzed and their internal consistency is shown.

In the pressure vessel example, one of the goals is to have a high vessel volume with a target value of 50,000 in^3. In Section 3.1, it is shown that the mean of the performance can be different from its nominal value (see Figure 3.4). This is why, within the modified RCEM, the performance mean is calculated using Eq. 3.8, which accounts for the nominal value shift when having nonlinear performance functions under uncertainty. With the function for the volume *V* given in Eq. 4.9, the estimation of the volume mean μ_V can be derived as:

Eq. 4.11

$$\mu_V \approx V(R,L) + \frac{1}{2}\frac{\partial^2 V(R,L)}{\partial R^2}\sigma_R^2 + \frac{1}{2}\frac{\partial^2 V(R,L)}{\partial L^2}\sigma_L^2$$

Since the function for the vessel volume is known explicitly and is fairly simple, the second derivative, which is needed in Eq. 4.11, can be easily calculated.

In the modified RCEM, the robustness of a design is measured with the variance of the performance. In the case of the pressure vessel example, the variance of the volume σ_V^2 and the variance of the weight σ_W^2 are the indicators for robustness. They are estimated using the Multiple Point Method, which is introduced in Section 3.2.2. The functional definition of the Multiple Point Method is presented in Eq. 3.13. With the explicit function for the vessel volume *V*, which is given in Eq. 4.9, the estimation of the volume variance can be derived as:

Eq. 4.12

$$\begin{aligned}\sigma_V^2 \approx{} & \frac{1}{12}\left[V(R,L) - V((R-2\sigma_R),L)\right]^2 + \frac{1}{12}\left[V(R,L) - V(R,(L-2\sigma_L))\right]^2 \\ & + \frac{1}{12}\left[V(R,L) - V((R+2\sigma_R),L)\right]^2 + \frac{1}{12}\left[V(R,L) - V(R,(L+2\sigma_L))\right]^2 \\ & + \frac{1}{48}\left[V((R-2\sigma_R),L) - V((R+2\sigma_R),L)\right]^2 \\ & + \frac{1}{48}\left[V(R,(L-2\sigma_L)) - V(R,(L+2\sigma_L))\right]^2\end{aligned}$$

The variance of the weight σ_W^2 can be calculated similarly with the function for the vessel weight given in Eq. 4.10:

$$\begin{aligned}\sigma_V^2 \approx & \frac{1}{12}\left[W(R,L,T)-W((R-2\sigma_R),L,T)\right]^2+\frac{1}{12}\left[W(R,L,T)-W(R,(L-2\sigma_L),T)\right]^2 \\ &+\frac{1}{12}\left[W(R,L,T)-W(R,L,(T-2\sigma_T))\right]^2 \\ &+\frac{1}{12}\left[W(R,L,T)-W((R+2\sigma_R),L,T)\right]^2+\frac{1}{12}\left[W(R,L,T)-W(R,(L+2\sigma_L),T)\right]^2 \\ &+\frac{1}{12}\left[W(R,L,T)-W(R,L,(T+2\sigma_T))\right]^2 \\ &+\frac{1}{48}\left[W((R-2\sigma_R),L,T)-W((R+2\sigma_R),L,T)\right]^2 \\ &+\frac{1}{48}\left[W(R,(L-2\sigma_L),T)-W(R,(L+2\sigma_L),T)\right]^2 \\ &+\frac{1}{48}\left[W(R,L,(T-2\sigma_T))-W(R,L,(T+2\sigma_T))\right]^2\end{aligned} \qquad \text{Eq. 4.13}$$

With Eq. 4.12 and Eq. 4.13, the cDSP for the design of the pressure vessel (Figure 4.2) is now completely defined with explicit functions and can be solved using JavaDSIDES. A special MATLAB program (JavaDSIDES XML Generator) was developed for this thesis, which offers a clear and simple interface for the designer to formulate the cDSP. This XML Generator takes the designer's input and generates the problem specific XML input file and executes JavaDSIDES. The code for Scenario 2 of the pressure vessel example is given in the Appendix B. The JavaDSIDES output files, for which examples are given in the Appendix B as well, provide the final values of the design variables as well as the intermediate results of each iteration cycle. From these values the designer can confirm the internal consistency of the results by analyzing the convergence of the DSIDES algorithm.

The JavaDSIDES results of the modified RCEM for design scenario 1 are presented in Table 4.3. From the results it can be seen that the DSIDES algorithm is sensitive towards the selection of the initial point. This is a known limitation of many gradient based search algorithms and is not further investigated in this thesis. For this design problem the best results could be achieved in run # 5. The quality of a particular solution can be read from the deviation value *Z*. The smaller the deviation from the goal(s) the more satisfactory is the solution. The fact that the algorithm converged in similar points supports the internal consistency of the results.

Table 4.3: Scenario 1 Results (modified RCEM)

Run #	Initial point {*R*,*L*,*T*} in [*in*]	JavaDSIDES Solution {*R*,*L*,*T*} in [*in*]	Estimated Volume $[in^3]$	Estimated Variance of Volume $[in^6]$	Estimated Variance of Weight $[lb^2]$	Estimated Deviation *Z*
1	{ 6, 6, 1.5}	{18.72, 8.79, 1.5}	37,281	13,755,928	376,214	0.2543
2	{ 6, 25, 6}	{18.72, 8.79, 1.5}	37,281	13,755,928	376,214	0.2543
3	{20, 25, 1.5}	{18.72, 9.27, 1.5}	37,809	14,030,351	383,336	0.2438
4	{20, 25, 4}	{18.91, 8.42, 1.5}	37,913	14,054,754	383,268	0.2417
5	{20, 48, 4}	{18.91, 8.41, 1.5}	37,914	14,054,950	383,264	0.2417
6	{36, 6, 4}	{18.73, 9.25, 1.5}	37,811	14,030,969	383,334	0.2438
7	{36, 48, 6}	{18.72, 9.27, 1.5}	37,809	14,030,324	383,335	0.2438

For run # 5 the convergence plot is given in Figure 4.3. Although the algorithm starts with an infeasible solution, it converges smoothly to a feasible solution. In Figure 4.4, the convergence plot for run # 1 is presented. Also here, a smooth convergence of the algorithm can be observed.

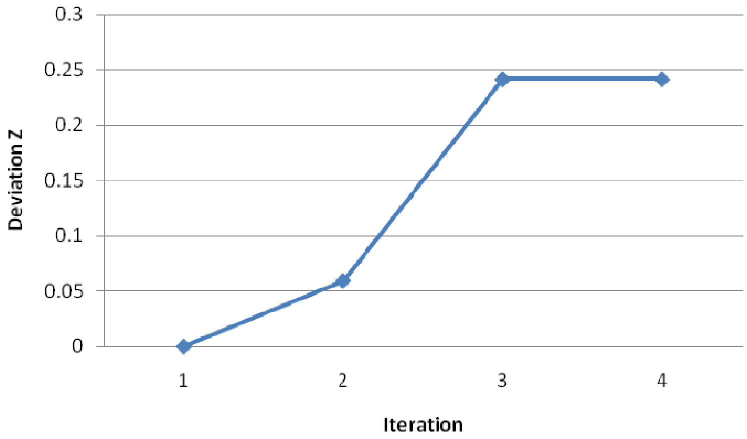


Figure 4.3: Convergence Plot for Run # 5 (Scenario 1)

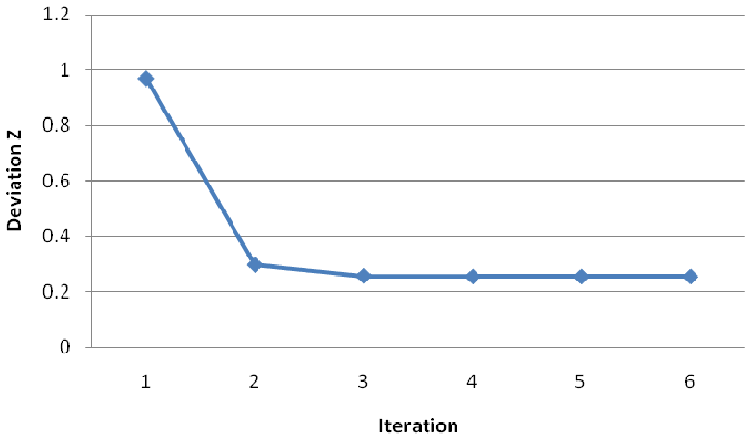


Figure 4.4: Convergence Plot for Run # 1 (Scenario 1)

Both plots support the internal consistency of the results for Scenario 1.

In Table 4.4, the JavaDSIDES results of the modified RCEM for Scenario 2 are given. Again, it can be seen that the convergence of the algorithm depends on the initial point. The internal consistency of the results can be concluded from the observation that the results of the different runs are almost the same and all show a smooth convergence. The design solutions with the lowest deviation from the goals are the results from runs # 1 and 2. However, here the volume constraint is violated to the maximum extent of 1%. The solutions of the remaining runs do not violate the volume constraint. Convergence plots for runs # 1 and 5 and are presented in Figure 4.5.

Table 4.4: Scenario 2 Results (modified RCEM)

Run #	Initial point {*R*,*L*,*T*} in [*in*]	JavaDSIDES Solution {*R*,*L*,*T*} in [*in*]	Estimated Volume [in^3]	Estimated Variance of Volume [in^6]	Estimated Variance of Weight [lb^2]	Estimated Deviation *Z*
1	{ 6, 6, 1.5}	{15.38, 6.0, 1.5}	19,800	5,904,472	176,411	0.31009
2	{ 6, 25, 6}	{15.38, 6.0, 1.5}	19,800	5,904,472	176,411	0.31009
3	{20, 25, 1.5}	{15.44, 6.0, 1.5}	20,000	5,983,542	178,477	0.31109
4	{20, 25, 4}	{15.44, 6.0, 1.5}	20,000	5,983,542	178,477	0.31109
5	{20, 48, 4}	{15.44, 6.0, 1.5}	20,000	5,983,542	178,477	0.31109
6	{36, 6, 4}	{15.44, 6.0, 1.5}	20,000	5,983,542	178,477	0.31109
7	{36, 48, 6}	{15.44, 6.0, 1.5}	20,000	5,983,542	178,477	0.31109

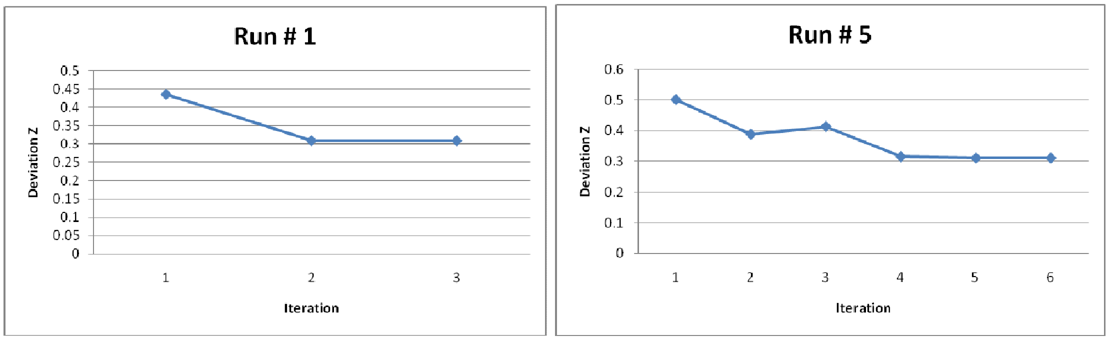


Figure 4.5: Convergence plots of Runs # 1 and 5 (Scenario 2)

A comparison of the solutions of the runs # 5 for both scenarios is given in Table 4.5. The results are as expected. The design solutions for Scenario 2 have a lower variance of the volume and the weight than the designs in Scenario 1. Since the Scenario 2 designs are a compromise between performance and robustness, the volume in those designs is lower, yet still within the given constraints.

Table 4.5: Comparison of Scenario 1 and Scenario 2 Results (modified RCEM)

Scenario	JavaDSIDES Solution $\{R,L,T\}$ in $[in]$	Estimated Volume $[in^3]$	Estimated Variance of Volume $[in^6]$	Estimated Variance of Weight $[lb^2]$
Scenario 1	{18.91, 8.41, 1.5}	37,914	14,054,950	383,264
Scenario 2	{15.44, 6.0, 1.5}	20,000	5,983,542	178,477

The computational costs of these modified RCEM runs are almost negligible. The runtime to test one initial point is less than 0.1 seconds.

From the presented results it can be seen that the method is very efficient and that the Scenario 2 solutions are indeed more robust than the Scenario 1 designs. For the validation of the effectiveness and the efficiency, however, some additional data is required. As outlined in Section 4.2, the accuracy of the results has to be verified and the efficiency has to be compared to alternative methods. In the following section, the results of the modified RCEM are verified by comparing them to Latin hypercube samplings. Furthermore, the results are compared to results of the original RCEM.

4.3.3 Verification of the Results and Comparison to the Original RCEM

In this section, a computationally expensive but very accurate Latin hypercube sampling is used to verify the accuracy of the results from the previous section. Furthermore, the original RCEM is applied to the pressure vessel design problem. A comparison of the results will be used to validate the performance of the modified RCEM.

For the verification of the accuracy of the results generated using the modified RCEM, a method is applied to the example, which implements a Latin hypercube sampling for estimating the performance mean and variances. This method is similar to the original and also the modified RCEM. A flow diagram of the method is given in Figure 4.6. If compared to the modified RCEM in Figure 3.15, it can be seen that the only difference is the estimation of the performance and the robustness. This way, the results are comparable because any differences in the results can be directly linked to the estimation of the performance mean and variances.

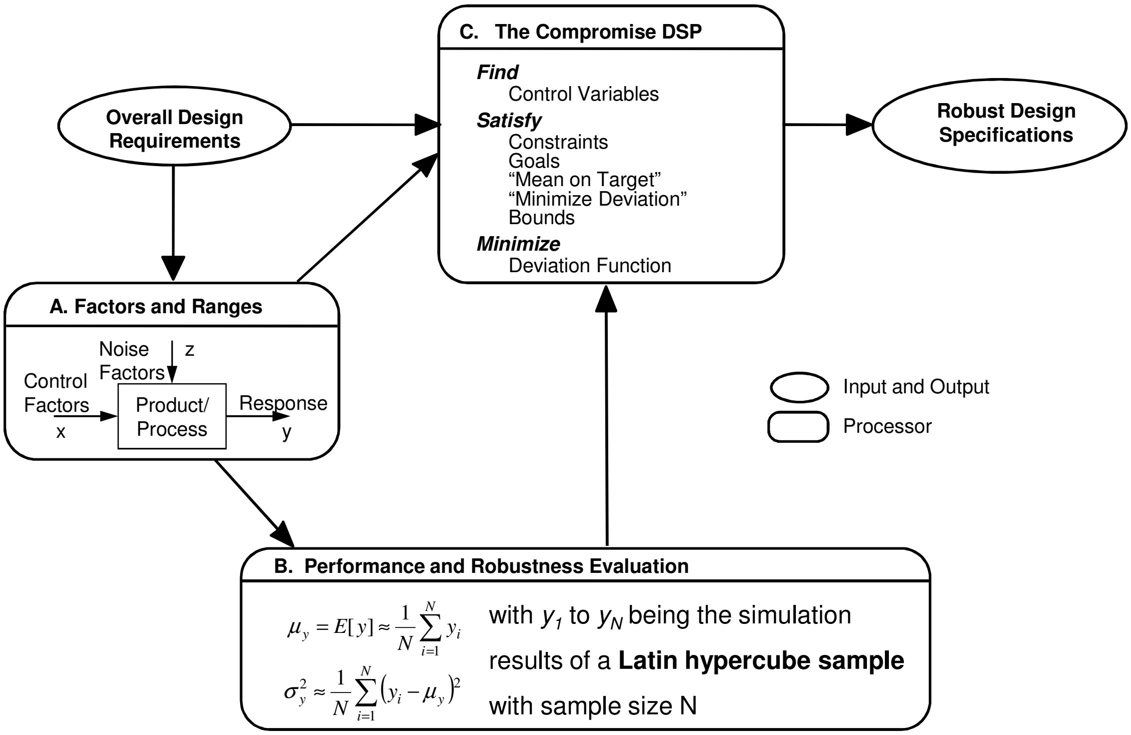

C. The Compromise DSP
Find
Control Variables
Satisfy
Constraints
Goals
"Mean on Target"
"Minimize Deviation"
Bounds
Minimize
Deviation Function
Overall Design Requirements
Robust Design Specifications
A. Factors and Ranges
Noise Factors
z
Control Factors
x
Product/ Process
Response
y
Input and Output
Processor
B. Performance and Robustness Evaluation
$\mu_y = E[y] \approx \frac{1}{N}\sum_{i=1}^{N} y_i$
$\sigma_y^2 \approx \frac{1}{N}\sum_{i=1}^{N}(y_i - \mu_y)^2$
with y_1 to y_N being the simulation results of a Latin hypercube sample with sample size N

Figure 4.6: Framework of the used Latin Hypercube Method

In this example, a sample size of 100,000 is used. For three uncertain variables (*R, L, T*), this sample size is considered large enough to get results that are very close to the *real* values. The JavaDSIDES solutions for the Latin hypercube method are presented in Table 4.6 for Scenario 1 and in Table 4.7 for Scenario 2.

Table 4.6: Scenario 1 Results (Latin Hypercube)

Run #	Initial point {*R*,*L*,*T*} in [*in*]	JavaDSIDES Solution {*R*,*L*,*T*} in [*in*]	Estimated Volume [in^3]
1	{ 6, 6, 1.5}	{18.72, 8.79, 1.5}	37,281
2	{ 6, 25, 6}	{18.72, 8.79, 1.5}	37,281
3	{20, 25, 1.5}	{18.72, 9.27, 1.5}	37,809
4	{20, 25, 4}	{18.91, 8.42, 1.5}	37,913
5	{20, 48, 4}	{18.91, 8.41, 1.5}	37,914
6	{36, 6, 4}	{18.73, 9.25, 1.5}	37,811
7	{36, 48, 6}	{18.72, 9.27, 1.5}	37,809

Comparing Table 4.6 and Table 4.3 (modified RCEM) it can be observed that the solutions are very similar. Only the estimations of the vessel volume are slightly different. However, with a difference of 0.01% in the estimation of the volume, the results of the modified RCEM can be considered very accurate.

Table 4.7: Scenario 2 Results (Latin Hypercube)

Run #	Initial point {*R*,*L*,*T*} in [*in*]	JavaDSIDES Solution {*R*,*L*,*T*} in [*in*]	Estimated Volume [in^3]	Estimated Variance of Volume [in^6]	Estimated Variance of Weight [lb^2]
1	{ 6, 6, 1.5}	{15.38, 6.0, 1.5}	19801	5887928	177090
2	{ 6, 25, 6}	{15.38, 6.0, 1.5}	19801	5887928	177090
3	{20, 25, 1.5}	{15.44, 6.0, 1.5}	20,000	5966442	179148
4	{20, 25, 4}	{15.44, 6.0, 1.5}	20,000	5966442	179148
5	{20, 48, 4}	{15.44, 6.0, 1.5}	20,000	5966442	179148
6	{36, 6, 4}	{15.44, 6.0, 1.5}	20,000	5966442	179148
7	{36, 48, 6}	{15.44, 6.0, 1.5}	20,000	5966442	179148

The same is true for Scenario 2. The JavaDSIDES results in Table 4.7 and Table 4.4 (modified RCEM) are again very similar. Furthermore, the estimations for the vessel volume are the same and the estimations of the volume and weight variances differ less than 1%. These observations allow the conclusion that the accuracy of the modified RCEM results is verified.

A comparison of the design solutions found with the modified RCEM and those found with the original RCEM is required to clearly link the efficiency and accuracy of the results to the modifications in the new method. The design solutions for Scenario 2 generated using the original RCEM are given in Table 4.8.

Table 4.8: Scenario 2 Results (Original RCEM)

Run #	Initial point {R,L,T} in [in]	JavaDSIDES Solution {R,L,T} in [in]
1	{ 6, 6, 1.5}	{15.41, 6.0, 1.5}
2	{ 6, 25, 6}	{15.41, 6.0, 1.5}
3	{20, 25, 1.5}	{15.46, 6.0, 1.5}
4	{20, 25, 4}	{15.46, 6.0, 1.5}
5	{20, 48, 4}	{15.46, 6.0, 1.5}
6	{36, 6, 4}	{15.46, 6.0, 1.5}
7	{36, 48, 6}	{15.46, 6.0, 1.5}

When comparing these results to those from the modified RCEM (Table 4.4) and the Latin hypercube sampling (Table 4.7), it can be seen that the original RCEM leads to design solutions that are slightly different. Since the Latin hypercube sampling method is assumed to be closest to the *real* values, the modified RCEM has a higher accuracy and thus is more effective than the original RCEM.

In this section, the modified RCEM was applied to the pressure vessel example and the accuracy was verified with a comparison to results generated using Latin hypercube samples. Furthermore, the results were compared to those from the original RCEM. This comparison allowed for linking the effectiveness of the new methods to the modifications. In the next section, the results and comparisons are critically discussed with regard to the validation of Hypothesis 1. The focus is on empirical performance validation.

4.4 Critical Discussion of the Example Results and Validation of Hypothesis 1

The data collected in Sections 4.3.2 and 4.3.3 is critically discussed in this section with respect to the validation of Hypothesis 1. The structure of this section will follow the outline of the validation strategy for the pressure vessel example presented in Section 4.2. While the empirical structural validation was discussed in Section 4.2.1, the empirical performance validation is discussed in this section.

First Research Hypothesis:

Using a global response surface, the nominal mean and a single point Taylor Series Approximation for the variance estimation is not accurate enough for finding robust solutions of nonlinear design problems. The Robust Concept Exploration Method can be improved by using appropriate formulations that account for the nonlinearity of the problem.

In Chapter 3, it was presented that using a global response surface, the nominal mean and a single point Taylor Series approximation for the variance estimation is not accurate enough for finding robust solutions of nonlinear design problems. In Section 3.2, modifications to the original RCEM are introduced that are hypothesized to be able to solve the discovered gaps. In Chapter 3, it is shown that the modified RCEM is capable of solving nonlinear problems. In this chapter, the validation is concluded with an engineering design problem. As outlined in Section 4.2.2, for the empirical performance validation it has to be shown (1) that the results meet the expectations, (2) that the demonstrated effectiveness and efficiency is linked to the application of the new method and (3) that the data generated and used for validation is accurate and internally consistent.

It has to be shown…

… (1) that the results meet the expectations:

In Section 4.3.2, the modified RCEM is applied to both pressure vessel example design scenarios. In Scenario 2, the design goal is to find a robust solution, while Scenario 1 demands for an optimization for maximum volume. These goals are achieved by using the modified RCEM. In Table 4.5, it is shown that the design goals are achieved and the modified RCEM is effective for solving this design problem. So the outcome was exactly as expected.

… (2) that the demonstrated effectiveness and efficiency is linked to the application of the new method:

A comparison of the solutions from the modified RCEM to those of the original RCEM is presented in Section 4.3.3. It is demonstrated that the solutions of the modified RCEM are slightly closer to the reference solutions from the Latin hypercube sampling. This shows that the effectiveness of the method is linked to the application of the new method. The modified RCEM is about as computationally efficient as the original RCEM, but significantly more efficient than the Latin hypercube sampling. While a JavaDSIDES run with the modified RCEM takes less than 0.1 seconds, a JavaDSIDES run with Latin hypercube sampling takes more than 4 minutes. For the experiments in this chapter a Windows XP PC with an Intel Core2 Quad CPU at 2.4 GHz and 3.25 GB of Ram was used.

… (3) that the data generated and used for validation is accurate and internally consistent:

The accuracy of the method's results is verified in Section 4.3.3 by a comparison to results of a method that implements a Latin hypercube sampling to estimate the performance mean and variances. In this method, 100,000 samples are generated for each evaluation of the goals and constraints in the search algorithm. The outcome of the modified RCEM was very similar to the sampling, only the estimations for the variances were slightly different but still very accurate. Previously in Chapter 3, it was presented that the modified RCEM is more effective in estimating performance variances than the original RCEM.

The internal consistency of the results from modified RCEM is demonstrated in Section 4.3.2. For both design scenarios multiple starting points are tested with the result that the algorithm always converges in one of two possible design solutions, which are almost identical. Furthermore, the internal consistency of the results was verified by demonstrating that the search algorithm within JavaDSIDES converges smoothly to the design solutions (Figure 4.3, Figure 4.4, and Figure 4.5).

In this section, the empirical performance of the modified RCEM is validated. This supports and confirms Hypothesis 1, which is stated at the beginning of this section. In the next section, this chapter is concluded and an outlook over the next chapters is given.

4.5 Chapter Synopsis

As shown in Figure 4.7, the focus in this chapter is the empirical structural and the empirical performance validation of the modified RCEM and Hypothesis 1. The modified RCEM is effective and efficient in solving design problems with explicit performance and constraint formulations. However, the modified RCEM is not capable of solving design problems that require simulations for the evaluation of goals or constraints. In Chapter 5, a method is developed that can solve nonlinear simulation-based design problems.

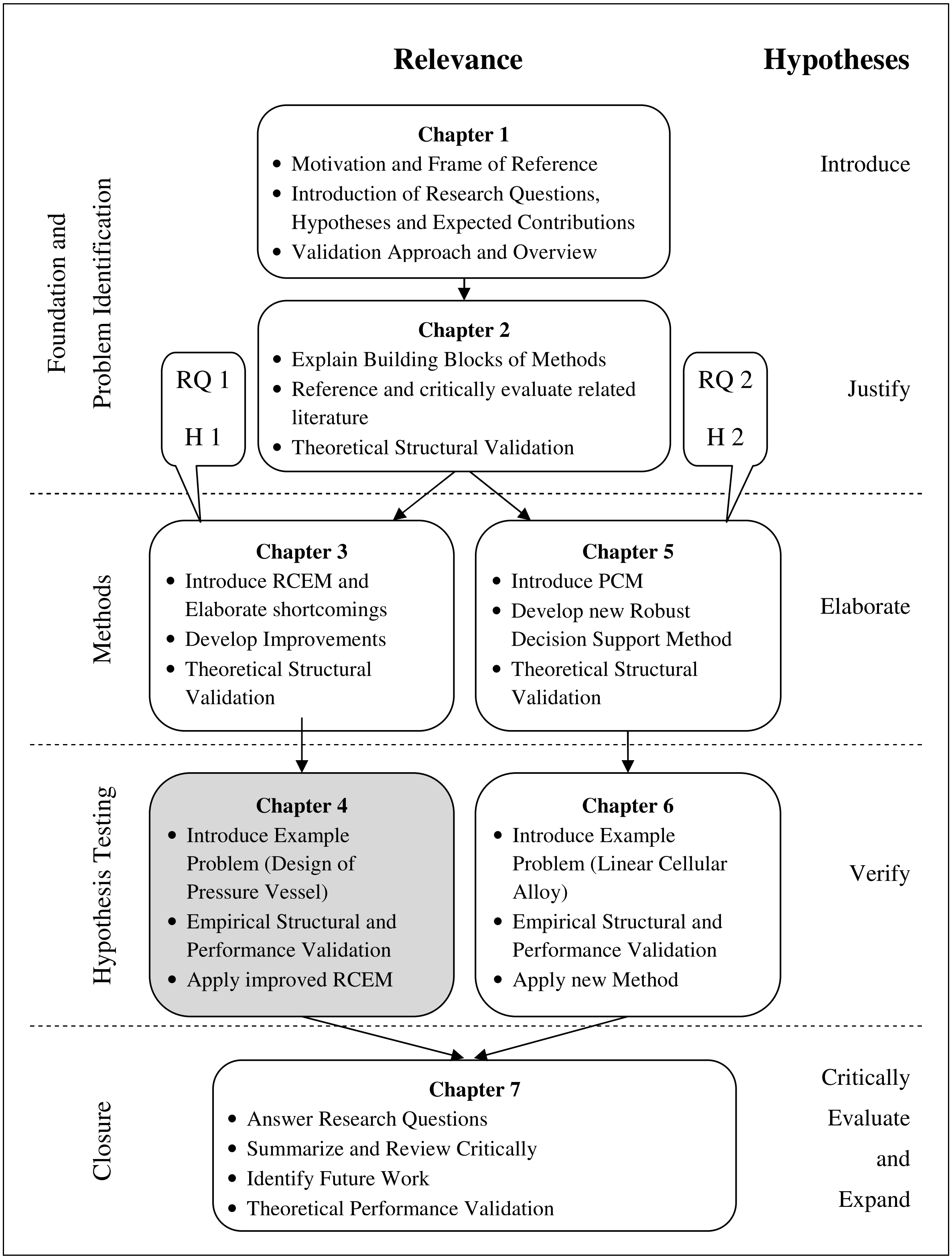

Relevance
Hypotheses
Foundation and Problem Identification
Chapter 1
• Motivation and Frame of Reference
• Introduction of Research Questions, Hypotheses and Expected Contributions
• Validation Approach and Overview
Introduce
Chapter 2
• Explain Building Blocks of Methods
• Reference and critically evaluate related literature
• Theoretical Structural Validation
RQ 1
H 1
RQ 2
H 2
Justify
Methods
Chapter 3
• Introduce RCEM and Elaborate shortcomings
• Develop Improvements
• Theoretical Structural Validation
Chapter 5
• Introduce PCM
• Develop new Robust Decision Support Method
• Theoretical Structural Validation
Elaborate
Hypothesis Testing
Chapter 4
• Introduce Example Problem (Design of Pressure Vessel)
• Empirical Structural and Performance Validation
• Apply improved RCEM
Chapter 6
• Introduce Example Problem (Linear Cellular Alloy)
• Empirical Structural and Performance Validation
• Apply new Method
Verify
Closure
Chapter 7
• Answer Research Questions
• Summarize and Review Critically
• Identify Future Work
• Theoretical Performance Validation
Critically Evaluate and Expand

Figure 4.7: Thesis Roadmap

CHAPTER 5

DEVELOPMENT OF A SIMULATION-BASED ROBUST CONCEPT EXPLORATION METHOD

In the previous two chapters, Research Question 1 is answered and Hypothesis 1 validated. With the modified RCEM, a method is introduced that is able to assist designers when designing a robust engineering system. However, the prerequisite to the design problem is that the performance and constraint functions are known explicitly since the derivatives are required to estimate the performance mean (see Section 3.2). In case the designer faces a design problem that requires simulations for the determination of the performance or constraint violation, e.g. FEM, Fluent or MATLAB, the modified RCEM is not applicable anymore. This problem is addressed in the Second Research Question, which is introduced in Chapter 1:

Second Research Question:

How can the RCEM be augmented to improve the accuracy of the robust concept exploration for nonlinear design problems that involve simulations?

In this chapter, the answer to this question is developed by investigating possibilities for a method that allows for simulation-based robust design. The main motivation for developing such an approach is that the use of global surrogate models like response surfaces can lead to wrong results. This limitation of the original RCEM is presented in

Section 3.1. In this chapter, particularly the Probabilistic Collocation Method (PCM), which was first introduced by Menner A. Tatang (Tatang, 1995), is assessed as a potential augmentation to RCEM. The goal is to develop a simulation-based Robust Concept Exploration Method that can be used for the validation of the Second Research Hypothesis, which is introduced in Chapter 1:

Second Research Hypothesis:

It is hypothesized that estimating the performance and the robustness using the Probabilistic Collocation Method with Polynomial Chaos Expansion in a simulation-based approach to robust concept exploration can increase the method's accuracy and reliability over the original RCEM.

In the following sections, the Probabilistic Collocation Method is described, discussed, and assessed. Afterward, the PCM is introduced to the RCEM framework to create a design method that can be used to validate the second Research Hypothesis. Following a detailed description of the Probabilistic Collocation Method and its implementation in RCEM in the next section, the strategy for validating the new method and Research Hypothesis 2 is presented. The remaining chapter, as well as Chapter 6, follow the structure determined by this outline.

5.1 Towards a Simulation-Based RCEM

Instead of running real experiments, in science and industry often resources are invested in generating computable models. These models are supposed to represent the real life relations between input and output of a specific system. The mindset behind building models can be very diverse; some real experiments are too expensive to be repeated very often (e.g., car crash tests), others are too dangerous (e.g., statics tests of buildings or bridges) or the goal of the experiment is a prediction (e.g., weather or climate predictions). Having a computer model opens up new possibilities, especially, the ability to assess the model under uncertainty of some or all of the input factors. This means that uncertainties of the input factors are assumed as random variables with specific distributions. If the model is computationally inexpensive, Monte Carlo Simulations (MCS) or Latin hypercube samplings (LHS) are the easiest way to investigate the model's behavior under uncertainty and to determine the mean, variance, higher moments or even the probability density functions (PDF) of the outputs. Since these methods involve 10,000 or more simulation runs for just one input setting, they are inapplicable for larger, computationally more expensive models. Especially, if the analysis is part of an iterative algorithm, which is the case for robust concept exploration methods, the need for alternative, more efficient methods is raised. For these reasons, methods have been developed, which can help to reduce the number of required simulations runs. In this thesis such a method, the Probabilistic Collocation Method, is introduced to RCEM to allow for simulation-based robust concept exploration. The underlying concept of the Probabilistic Collocation Method (PCM) is explained in greater detail in Section 2.3. In the following sub-sections first, the Probabilistic Collocation Method is demonstrated along an example and then introduced to RCEM. Finally, the computational framework for this design method is described briefly.

5.1.1 The Probabilistic Collocation Method

The following description of the Probabilistic Collocation Method is leveraged with modifications from (Webster, et al., 1996).

The PCM is a local surrogate modeling technique based on the Gaussian Quadrature Integration and the Polynomial Chaos Expansion (PCE). The details of the underlying concept of the PCM are explained in greater detail in Section 2.3. The steps of the method are illustrated in the flow diagram in Figure 5.1. In this section, the PCM is explained along the artificial design problem used in Chapter 3. For this description, Eq. 3.1 is taken as a black box model and the function calls represent the simulations.

Step 1 and 2: Identify and specify uncertain design variables and noise factors

First, the uncertain parameters (design variables or noise factors) have to be identified and their uncertainty distribution has to be determined. The determination of the distribution can either be based on the designer's experience or on available empirical statistical data like known manufacturing tolerances, temperature or climate observations. In the example (Eq. 3.1), there is only one design variable, which is Gaussian distributed. For this example a standard deviation of 3 is assumed and the analysis is done for the local minimum at $x = 931.95$.

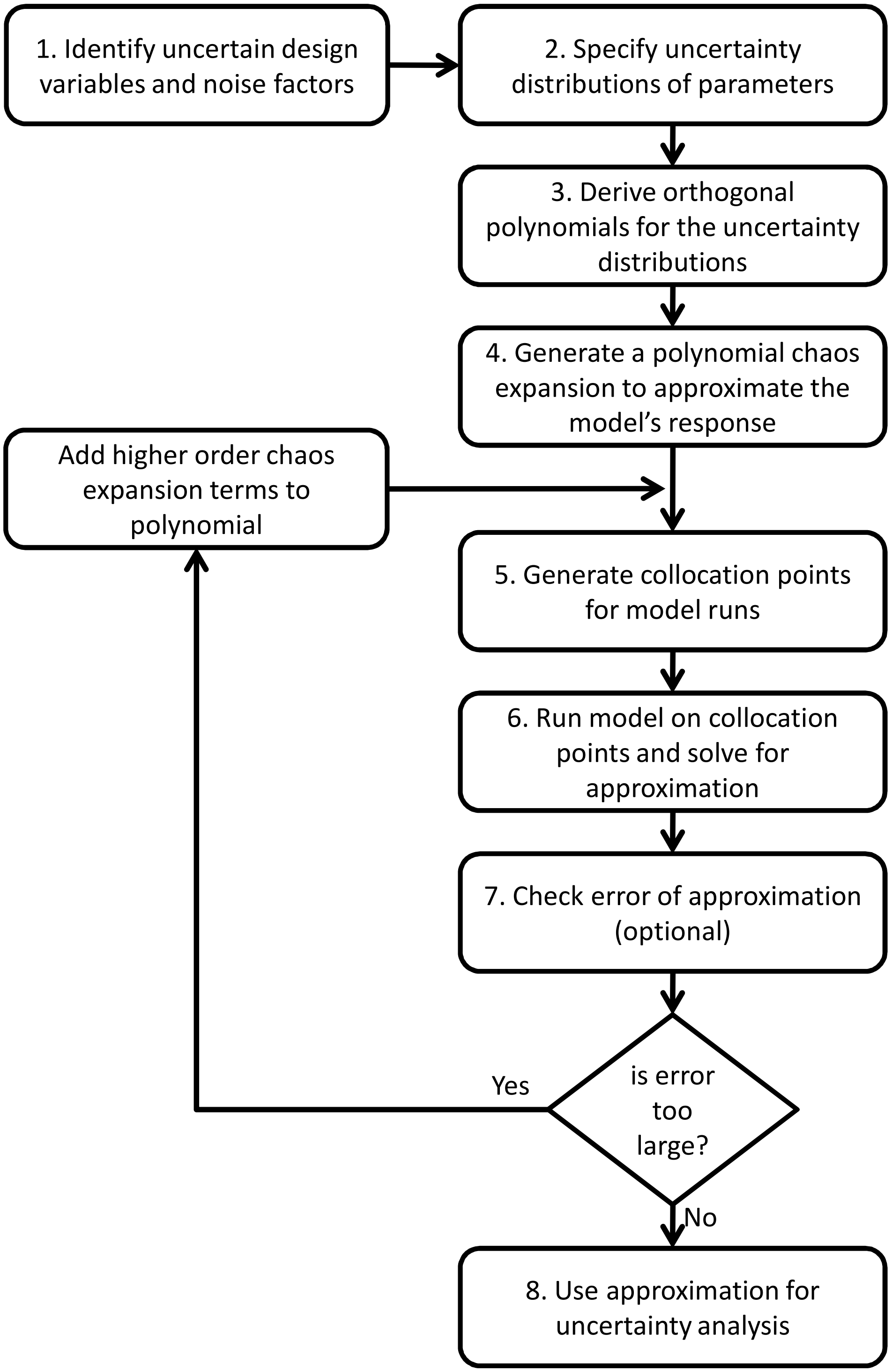

1. Identify uncertain design variables and noise factors
2. Specify uncertainty distributions of parameters
3. Derive orthogonal polynomials for the uncertainty distributions
4. Generate a polynomial chaos expansion to approximate the model's response
Add higher order chaos expansion terms to polynomial
5. Generate collocation points for model runs
6. Run model on collocation points and solve for approximation
7. Check error of approximation (optional)
Yes
is error too large?
No
8. Use approximation for uncertainty analysis

Figure 5.1: Information Flow Diagram of the Probabilistic Collocation Method (Webster, et al., 1996)

Step 3: Derive orthogonal polynomials for the uncertainty distributions

In the third step, the orthogonal polynomials for the distributions determined in the previous step have to be derived. If the approximation of the model response should be of order *n* the orthogonal polynomials up to order *n+2* have to be determined. A recursive algorithm for the construction of orthogonal polynomials can be found in (Davis and Rabinowitz, 1975); alternatively an algorithm called ORTHPOL (Gautschi, 1994) can be used to derive the required polynomials. Since in this example a Gaussian distribution is assumed, the following simplification can be used. All Gaussian distributed random variables *X* can be represented by:

$$X = \mu + \sigma(H_1(\xi)) \qquad \text{Eq. 5.1}$$

where μ is the mean and σ is the standard deviation of *X*. $H_1(\xi)$ is the first order Hermite polynomial. Hermite polynomials are a set of polynomials that are orthogonal to the standard normal distribution ξ, which has a mean of 0 and a variance of 1. This allows for using the same set of orthogonal polynomials for all Gaussian distributions instead of deriving orthogonal polynomials for each specific distribution. Since in this example an approximation of third order is demonstrated, the first five Hermite polynomials are required:

$$H_1(\xi) = \xi \qquad \text{Eq. 5.2}$$

$$H_2(\xi) = \xi^2 - 1 \qquad \text{Eq. 5.3}$$

$$H_3(\xi) = \xi^3 - 3\xi$$ Eq. 5.4

$$H_4(\xi) = \xi^4 - 6\xi^2 + 3$$ Eq. 5.5

$$H_5(\xi) = \xi^5 - 10\xi^3 + 15\xi$$ Eq. 5.6

For the example in this chapter, $\sigma_x = 3$ is assumed and the analysis for $\mu_x = 931.95$ is demonstrated. According to Eq. 5.1, design variable *x* can be represented as:

$$x = 931.95 + 3(H_1(\xi))$$ Eq. 5.7

<u>*Step 4: Generate a polynomial chaos expansion to approximate the model's response*</u>

In this step, a polynomial expression is generated to represent the performance or response variable as a function of the orthogonal polynomials of ξ. The third order approximation used in this example is given in Eq. 5.8.

$$y \approx \hat{y} = a_0 + a_1 H_1(\xi) + a_2 H_2(\xi) + a_3 H_3(\xi)$$ Eq. 5.8

Since $H_1(\xi)$, $H_2(\xi)$ and $H_3(\xi)$ are known, Eq. 5.8 has four unknowns; a_0, a_1, a_2 and a_3. To solve for these variables, four simulation runs are required. The points for those runs are determined in the next step. If the model has more than one uncertain parameter, cross product terms can be added to the polynomial chaos expansion.

Step 5: Generate collocation points for model runs

The goal when applying the Probabilistic Collocation Method is to find a good approximation with the fewest number of simulations. This is why it is important to carefully select the collocation points in which the model is evaluated. The method for selecting the collocation points is derived from the same idea as the Gaussian Quadrature method to numerically solve integrals (for example, see (Davis and Rabinowitz, 1975)). In the PCM, the collocation points are selected from the roots of the orthogonal polynomials of the next higher order ($n+1$) for each uncertain parameter. In case of only one uncertain parameter, the number of roots equals the number of required points. If the model has more than one uncertain dimension, it is suggested in the literature to select the collocation points with the highest joint probability. For this example the roots of $H_4(\xi)$ are (2.3344, 0.7420, -0.7420, -2.3344). Substituting into Eq. 5.1, the collocation points are:

$$\mathbf{x} = \begin{bmatrix} 938.9532 \\ 934.1759 \\ 929.7241 \\ 924.9468 \end{bmatrix} \qquad \text{Eq. 5.9}$$

Step 6: Run model on collocation points and solve for approximation

Running the example model with the collocation points (Eq. 5.9) as inputs gives the following results:

$$\mathbf{y} = \begin{bmatrix} 19.4843 \\ 14.7726 \\ 15.0445 \\ 27.4895 \end{bmatrix}$$ Eq. 5.10

With the four pairs (ξ_i, y_i) the approximation in Eq. 5.8 can be solved for the unknowns using, for example, the following equation:

$$\mathbf{a} = \left(\mathbf{B}^T\mathbf{B}\right)^{-1}\mathbf{B}^T\mathbf{y}$$ Eq. 5.11

with

$$\mathbf{B} = \begin{bmatrix} 1 & H_1(\xi_1) & H_2(\xi_1) & H_3(\xi_1) \\ 1 & H_1(\xi_2) & H_2(\xi_2) & H_3(\xi_2) \\ 1 & H_1(\xi_3) & H_2(\xi_3) & H_3(\xi_3) \\ 1 & H_1(\xi_4) & H_2(\xi_4) & H_3(\xi_4) \end{bmatrix}$$ Eq. 5.12

In this example the coefficients are:

a_0 = 15.6957

a_1 = - 0.9489

a_2 = 1.7510

a_3 = - 0.3126

Step 7: Check error of approximation (optional)

Before the approximation is used for the uncertainty analysis, the goodness of the fit can be tested. For this error check a few more simulation runs are required. The collocation points for these runs are derived from the next higher order orthogonal polynomials. Those points are always between the collocation points the approximation is fitted to, which allows for a reliable error check. In this thesis, R^2 is used as a measure of the fit. In literature also the sum-square-root and the relative sum-square-root are recommended (see (Webster, et al., 1996)). For the example, in this section R^2 equals 0.9996, which allows the conclusion that the fit is sufficiently good.

<u>*Step 8: Use approximation for uncertainty analysis*</u>

Since orthogonal polynomials are used in the polynomial chaos expansion, some of the output or performance moments can be derived directly from the coefficients a_i. Details on this can be found in (Hockenberry, 2000). For example, the mean is simply a_0, which is 15.6957 in this example. The second moment, the variance can be determined by solving the following equation:

Eq. 5.13

$$\sigma_{y(x)}^2 = \sum_{i=1}^{n-1} a_i^2$$

In the example, $\sigma_{y(x)}^2$ equals 4.0643. Of course, the approximation can now be used for any uncertainty analysis including Monte Carlo Simulations. Applied to the approximation, the MCS requires a faction of the computational time that it would take if applied to the simulation.

The MATLAB code required to reproduce these results can be found in Appendix B.1.

5.1.2 The New Simulation-Based Robust Concept Exploration Method

In the previous section, the Probabilistic Collocation is described. With this method, the uncertainty propagation in simulations can be estimated. By introducing the PCM to the

RCEM, a simulation-based robust concept exploration method is created in this section. The mindset behind developing this new design method is to eliminate the limitation of the original RCEM, which are identified in Section 3.1. Particularly the limitations of global response surfaces when having a design problem that involves simulations are addressed. The PCM substitutes the response surfaces as well as the Taylor series approximations in the original RCEM. The new framework is presented in Figure 5.2.

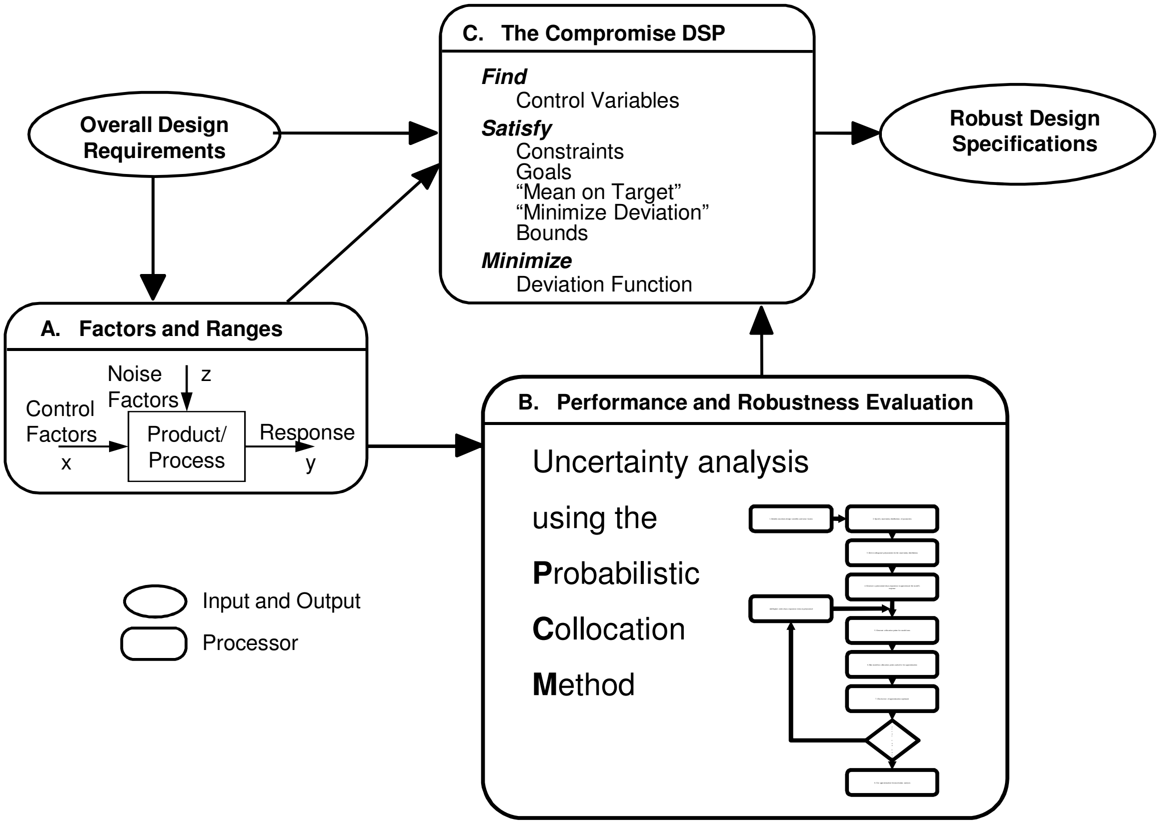


Figure 5.2: Framework of the Simulation-Based RCEM

With the new method the designer does not have to build a global response surface before analyzing the concepts. With the PCM integrated into the cDSP, small local response surfaces at the points of interest are generated within the DSIDES algorithm. This process is fully automated and by this simplifies the application of RCEM. The implications on the performance of the design method are evaluated in Section 5.4 as well as in Chapter 6.

5.1.3 Computational Framework for the Simulation-Based RCEM

The simulation-based RCEM is computationally realized within JavaDSIDES, which is described in Section 2.4 and Appendix A. Since PCM is written in MATLAB, the code has to be made compatible with JavaDSIDES. The module MATLAB Builder JA is capable of wrapping MATLAB code in Java and thus allows deploying the PCM code as a Java package file. This file can be imported to the JavaDSIDES model so that the algorithm can execute it for the uncertainty analysis. For details on this process see Appendix A.

5.2 Validation and Verification Strategy for the Simulation-Based RCEM

In this and the next chapter, the simulation-based RCEM and Hypothesis 2 are validated and verified. As outlined in Section 1.4, the validation square provides the structure for the validation of this thesis. The validation square consists of four quadrants: Theoretical Structural Validation, Empirical Structural Validation, Empirical Performance Validation, and Theoretical Performance Validation (see Figure 1.6). The theoretical structural validation is discussed in this section. The empirical structural validation and the empirical performance validation are outlined in this section and demonstrated in

Section 5.4 and Section 6.4 in the next chapter. The theoretical performance validation for both Hypothesis 1 and 2 is discussed in Chapter 7.

5.2.1 Theoretical Structural Validation

For the theoretical structural validation, the internal consistency of the method has to be shown. The simulation-based RCEM (see Figure 5.2) consists of the following two main components:

- Compromise Decision Support Problem (cDSP)
- Probabilistic Collocation Method (PCM)

The cDSP has been validated in literature before (Bras and Mistree, 1993), so the internal consistency of this component does not have to be shown here. The internal consistency of the new component in the RCEM, the PCM, is presented in Section 2.3 and Section 5.1.1 with the detailed explanation of the method. The description contains all steps of the PCM as well as the underlying mathematical formulations. Furthermore, the PCM has been used successfully in literature before (see Section 5.1). In Figure 5.2, the flow diagram of the new design method is given and the relation between the components is shown. The design problem formulation in the cDSP requires the estimation of the performance mean and variance, which is provided by the PCM. With this, the theoretical structural validity is demonstrated.

5.2.2 Empirical Structural Validation

Demonstrating the empirical structural validity involves discussing the appropriateness of the chosen examples for illustrating and verifying the design method. The simulation-based Robust Concept Exploration Method, which is introduced in Section 5.1.2, is validated with two example problems. The first example is the artificial design problem used previously in Chapter 3. For the validation in this chapter, this function (Eq. 3.1) is regarded as a black box and thus simulating a model with unknown response. This way, the example is representative for a simple one-dimensional design problem with one uncertain design variable. In Chapter 6, the new method is applied to the design of a robust Linear Cellular Alloy cooling structure. The empirical structural validation of this example is addressed in Chapter 6.

Is the example similar to design problems for which the simulation-based RCEM is intended and may be applied?

The following characteristics of the artificial design problem qualify it as a suitable problem for the simulation-based RCEM:

- The problem is non-linear and thus requires a method that can handle non-linearity. Particularly this example problem could not be solved with the original RCEM due to its non-linear characteristics.
- The example includes a black box model for which the functional relation between input and output is regarded as unknown.

- The design has two objectives, of which one is a robustness goal. This means that a method is required that can handle multiple objectives and is able to find compromises. Furthermore, the method has to be capable of evaluating the robustness of a potential design solution.
- The design example has an uncertain design variable and thus demands for a method that can handle uncertain inputs.

Can the data from these examples be used to support conclusions with respect to the hypothesis?

In order to validate Research Hypothesis 2, it has to be shown that the simulation-based RCEM increases accuracy and reliability over the original RCEM. Accuracy in this context means that the new method's estimations for the performance mean and variance have to be closer to the real values. Reliability means that the new method has to be able to bring better results especially for highly non-linear problems, for which the original RCEM fails. Both improvements in accuracy and reliability can be shown with data from the first example, the artificial problem. With both methods, the performance mean and variance is estimated. A comparison of those values to reference values generated using a Latin hypercube sampling or Monte Carlo simulation will support conclusions regarding the accuracy. Furthermore, the simulation-based RCEM is applied to the example problem. From Section 3.1, it is known that the original RCEM fails to solve this problem. The data from applying the new method will support conclusions regarding the reliability of the method. Additionally, it can be seen if the new method is able to find a robust solution. This will also be supported by the data from the second example, the design of a robust LCA cooling structure. Furthermore, this data is used to evaluate the ability of the method to solve multidimensional, multi-objective problems with multiple uncertain parameters. Finally, the data from the LCA example will also support the conclusion that the new method can handle simulation-based problems effectively and efficiently without requiring a global response surface or a similar global surrogate model.

5.2.3 Empirical Performance Validation

The two examples are used to demonstrate the empirical performance validation of Hypothesis 2 and the simulation-based RCEM as well. Therefore, it has to be shown (1) that the results meet the expectations, (2) that the demonstrated effectiveness and efficiency is linked to the application of the new method and (3) that the data generated and used for validation is accurate and internally consistent.

How can the results be evaluated with respect to the expected outcome of the method?

The results for both examples are generated for two different scenarios, of which the first one will ask the designer to focus on maximum performance and the second on a compromise between high performance and high robustness (low performance variability). If the results for the second scenario are design solutions that are more robust than the results for the first scenario, the outcome is as expected and Hypothesis 2 is supported.

How can it be demonstrated that the observed effectiveness and efficiency is linked to the application of the simulation-based RCEM?

The first example is solved with both methods, the original RCEM and the simulation-based RCEM. A direct comparison of the results will allow conclusions regarding the link between the observed effectiveness and efficiency and the application of the simulation-based RCEM.

The results of the LCA example will allow conclusions especially about the efficiency of the new method. The efficiency is regarded as inversely proportional to the computational costs. A comparison of the required computational time of applying the simulation-based RCEM to the time required for using the original RCEM will be used to support the Research Hypothesis 2.

How are the accuracy and the internal consistency of the results verified?

First, the accuracy of the results is verified by comparing the outcome of the simulation-based RCEM to Latin hypercube samplings. Using a computationally expensive Latin hypercube sampling within the DSIDES algorithm (see Figure 4.6) to determine performance and robustness is assumed to give results very close to reality. Showing that the results of the simulation-based RCEM are close to those values will support the argument for the accuracy of the method. The internal consistency of the results is confirmed by testing multiple starting points for the algorithm as well as by an analysis of the convergence of the algorithm.

The following sections and Chapter 6 are structured according to the validation strategy outlined in this section. In the next section, the simulation-based RCEM is applied to the artificial design problem introduced in Chapter 3 followed by a critical discussion of the results.

5.3 Application of the Simulation-Based RCEM to the Artificial Design Problem

In this section, the simulation-based RCEM is applied to the artificial design problem (Eq. 3.1), which is introduced in Chapter 3. First, the performance of the mean and variance estimation used in the simulation-based RCEM is compared to the Taylor series approximation used in the original RCEM. Afterwards, the new method is implemented within JavaDSIDES for solving the design problem. All results will be verified and checked for internal consistency.

A crucial part of a robust concept exploration method is the accurate estimation of the performance mean and variance values. In Section 3.1, it is shown that the original RCEM has limitations here when it comes to non-linear problems. Similar to Section 3.2, the accuracy of the Probabilistic Collocation Method, which is responsible for the mean and variance estimation in the simulation-based RCEM, is tested using the artificial performance function given in Eq. 3.1. In Section 3.1, it was discovered that the local minimum of the example function is the crucial point. The original RCEM fails in solving the design problem because its estimation of the performance variance in this point is wrong. This is why, in this section, the mean and variance estimations of the new method in this point are compared to the estimation with a Latin hypercube sample with 100,000 sample points. For the comparison, Gaussian input distributions are assumed with standard deviations of 2 ($\sigma_x^2 = 4$) and 3 ($\sigma_x^2 = 9$) respectively. The test point is the local minimum of the performance function at *x* = 931.95 (see Figure 3.2) with a nominal function value (performance) of *y* = 13.96. In Table 5.1, the performance mean estimations μ_y and the variance estimations σ_y^2 are presented.

Table 5.1: Mean and Variance Estimations at x = 931.95

Method	**Scenario 1:** $\sigma_x^2 = 4$		**Scenario 2:** $\sigma_x^2 = 9$	
	μ_y	σ_y^2	μ_y	σ_y^2
Latin Hypercube Sampling	**14.72**	**1.32**	**15.69**	**7.52**
simulation-based RCEM	14.72	0.69	15.69	4.07
original RCEM	13.96	0.00	13.96	0.00

From the results it can be seen that the accuracy of the estimations from the Probabilistic Collocation Method in the simulation-based RCEM are superior to the estimation methods used in the original RCEM. Both the estimations for the mean and the variance of the new method are closer to the reference values from the Latin hypercube sampling than those of the old method. Although the new method shows a significant improvement, there is still some potential for further development. These results are confirmed by a second test at the point *x* = 950.00, which lies in the robust area of the performance function. The outcome of this example, which is given in Table 5.2, supports the previous finding that the new method has an improved accuracy in estimating the performance mean and variance.

Table 5.2: Mean and Variance Estimations at x = 950.00

Method	**Scenario 1:** $\sigma_x^2 = 4$		**Scenario 2:** $\sigma_x^2 = 9$	
	μ_y	σ_y^2	μ_y	σ_y^2
Latin Hypercube Sampling	**27.77**	**0.74**	**27.71**	**1.97**
simulation-based RCEM	27.77	0.73	27.71	1.94
original RCEM	27.82	0.62	27.82	1.40

In the next step, the complete simulation-based RCEM is tested on the artificial design problem. The cDSP for this example is given in Figure 5.3. The design problem is similar to the one in Chapter 3. However, the comparability of the results is limited because the performance function is regarded as a black box model in this section. The goal for the performance mean is μ_y = 10 and for the variance σ_y^2 = 0.5. The design variable x is assumed to be uncertain with a Gaussian distribution and a variance $\sigma_x^2 = 9$. As in Chapter 3, the problem is solved for two scenarios. In the first scenario, the designer is putting a high weight on achieving maximum performance without really considering the robustness of the solution. The designer in the second scenario is looking for a compromise between high performance and high robustness. The example problem is solved with two methods: the simulation-based RCEM (Section 5.1.2), which is computationally realized as described in Section 5.1.3 and a Latin hypercube sampling approach, which is illustrated in Figure 4.6.

GIVEN

Black-box performance function Eq. 3.1: $f(x) = \sum_{i=1}^{9} a_i (x - 900)^{(i-1)}$

$a_1 =$	– 659.23	$a_4 =$	0.82691	$a_7 =$	– 3.2446 x 10-6
$a_2 =$	190.22	$a_5 =$	– 0.021885	$a_8 =$	1.6606 x 10-8
$a_3 =$	– 17.802	$a_6 =$	0.0003463	$a_9 =$	– 3.5757 x 10-11

Formulation for variance estimation $\sigma^2_{f(x)}$:

a) Probabilistic Collocation Method

b) Latin hypercube sampling

Variance of design variable $\sigma_x^2 = 9$

FIND

Design variable: x

Deviation variables: $d_i^-, d_i^+ \quad i = 1,2$

SATISFY

GOALS

Performance goal: $\dfrac{1/f(x)}{1/10} + d_1^- - d_1^+ = 1$

Robustness goal: $\dfrac{1/\sigma^2_{f(x)}}{1/0.5} + d_2^- - d_2^+ = 1$

BOUNDS

$910 \le x \le 976$

$d_i^-, d_i^+ \ge 0 \quad i = 1,2$

$d_i^- \cdot d_i^+ = 0 \quad i = 1,2$

MINIMIZE

$Z = W_1 \cdot d_1^- + W_2 \cdot d_2^- \qquad d_1^+, d_2^+ = 0$

Scenario 1:	Scenario 2:
$W_1 = 0.5$	$W_1 = 0.1$
$W_2 = 0.5$	$W_2 = 0.9$

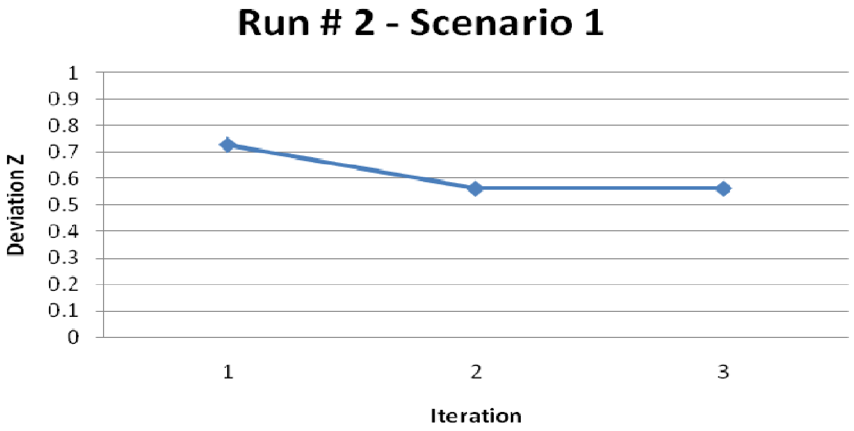
Figure 5.3: cDSP for First Example Design Problem

Table 5.3: Design Solutions for Scenario 1 and Scenario 2

Run #	Initial Point	Scenario 1		Scenario 2	
		Solution of the LHS Method	Solution of the Simulation-Based RCEM	Solution of the LHS Method	Solution of the Simulation-Based RCEM
1	x = 920	933.57	933.26	951.70	951.70
2	x = 940	933.57	933.26	951.70	951.70
3	x = 960	933.57	933.26	951.70	951.70
4	x = 976	933.57	933.26	951.70	951.70

The results, which are presented in Table 5.3, are exactly as expected. The design solutions of the simulation-based RCEM are very similar to the results of the LHS reference method. The two design scenarios are solved correctly. The solutions for Scenario 2 lie in the robust region (compare to Figure 3.2), whereas the solutions for Scenario 1 are very close to the local minimum at x = 931.95. The reason why the algorithms did not converge in the minimum is the small weight that is put on robustness in Scenario 1. Since the performance is more robust to the right side of the minimum, the solutions are around x = 933.57. For the validation of the internal consistency of the results, the convergence plots of the algorithm for run #2 in both scenarios are given in Figure 5.4.

For completeness, it has to be mentioned that the simulation-based RCEM does not find the robust solution in Scenario 2, if the design variable variance is 4 or less. In Table 5.1, it can be seen that the variance estimation of the PCM at the minimum is close to 0.5 if $\sigma_x^2 = 4$, which is the target in the cDSP. One limitation of the cDSP and DSIDES is that values that are close to the target are weighted higher by the algorithm than those further away from the target. For this reason the method converges in a design solution that is not satisfying for the designer.



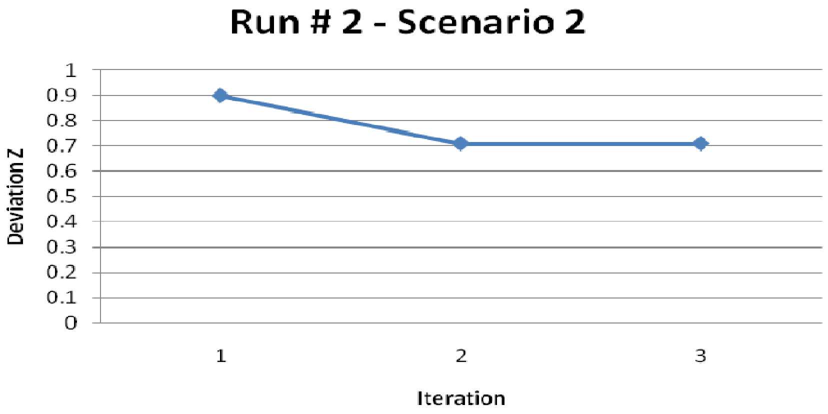


Figure 5.4: Convergence plots of Run # 2 for Scenario 1 and 2

In this section, the simulation-based RCEM was successfully applied to the artificial design problem and the results were verified with Latin hypercube samples and convergence analyses. In the next section, the results are discussed with regard to the validation of the new design method as well as Research Hypothesis 2.

5.4 Critical Evaluation of the Results

In this section, the validation of the simulation-based RCEM and Research Hypothesis 2 is discussed. As outlined in Section 1.4 for the entire thesis and in Section 5.2 for the simulation-based RCEM, the validation is presented along the four quadrants of the validation square (see Figure 1.6). The first two quadrants, the theoretical structural validation and the empirical structural validation, are discussed and verified in Sections 5.2.1 and 5.2.2. The empirical performance validation is addressed in the following.

In order to demonstrate the empirical performance validation, it has to be shown (1) that the results meet the expectations, (2) that the demonstrated effectiveness and efficiency is linked to the application of the new method and (3) that the data generated and used for validation is accurate and internally consistent.

(1) Do the results meet the expectation?

As mentioned in Section 5.3, the results of the new method meet the expectations. The estimations for the mean of the performance are very accurate. While the estimations of the performance variance from the new method are significantly better than from the original RCEM, they still leave room for improvement. The results from applying the new method to the design problem are very accurate as seen in the last section. However, design scenarios and parameter constellations have been found, where the solutions of the new method are just equal to the original RCEM. With a further improvement of the estimation accuracy of the performance variance, the new method could find satisfying solutions for these scenarios as well. This is discussed more detailed in Chapter 7.

(2) Is the demonstrated effectiveness and efficiency linked to the application of the new method?

The demonstrated effectiveness is clearly linked to the application of the new method. In Section 5.3, all results of the new method are compared to results of the original RCEM. It is demonstrated that all results of the new method are superior to the old method. This is true for the estimations of the performance mean and the variance as well as for the application of the entire method to a design problem.

In this example, the simulation-based RCEM does not show a significantly higher efficiency than the original RCEM. This is because the example was not a real simulation-based problem but an explicit performance function that was handled as a black-box for the new method. This means that the original RCEM was applied to the problem without generating a response surface. The efficiency of the new design method will be addressed in Chapter 6 in greater detail.

(3) Is the data generated and used for validation verified and internally consistent?

The internal consistency of the generated data is demonstrated with the convergence plots in Section 5.3 (Figure 5.4). It can be seen that the algorithm shows a smooth convergence. Furthermore, several starting points are tested in Section 5.3, which all converge to the same solution. This allows the conclusion that the data is internally consistent.

The accuracy of the data from the new method is verified in Section 5.3 with comparisons to the results of Latin hypercube samples with a sample size of 100,000. The sample size is large enough for this example so that the LHS results can be considered very accurate and close to the real values. With these comparisons it is demonstrated that the accuracy of the new method is as expected.

The empirical performance validation was discussed in this section. This discussion is continued in Section 6.4 with the results of the heat exchanger example, which is introduced in Chapter 6.

5.5 Chapter Synopsis

As presented in Figure 5.5, the focus in this chapter is on the introduction of the Probabilistic Collocation Method in Section 5.1.1 and the simulation-based Robust Concept Exploration Method in Section 5.1.2. Furthermore, the new method is validated with an artificial design problem in Section 5.3. The Validation of the new method and Research Hypothesis 2 is outlined and discussed in Sections 5.2 and 5.4. In Chapter 6, the new method is validated with a second example, the design of a robust Linear Cellular Alloy cooling structure.

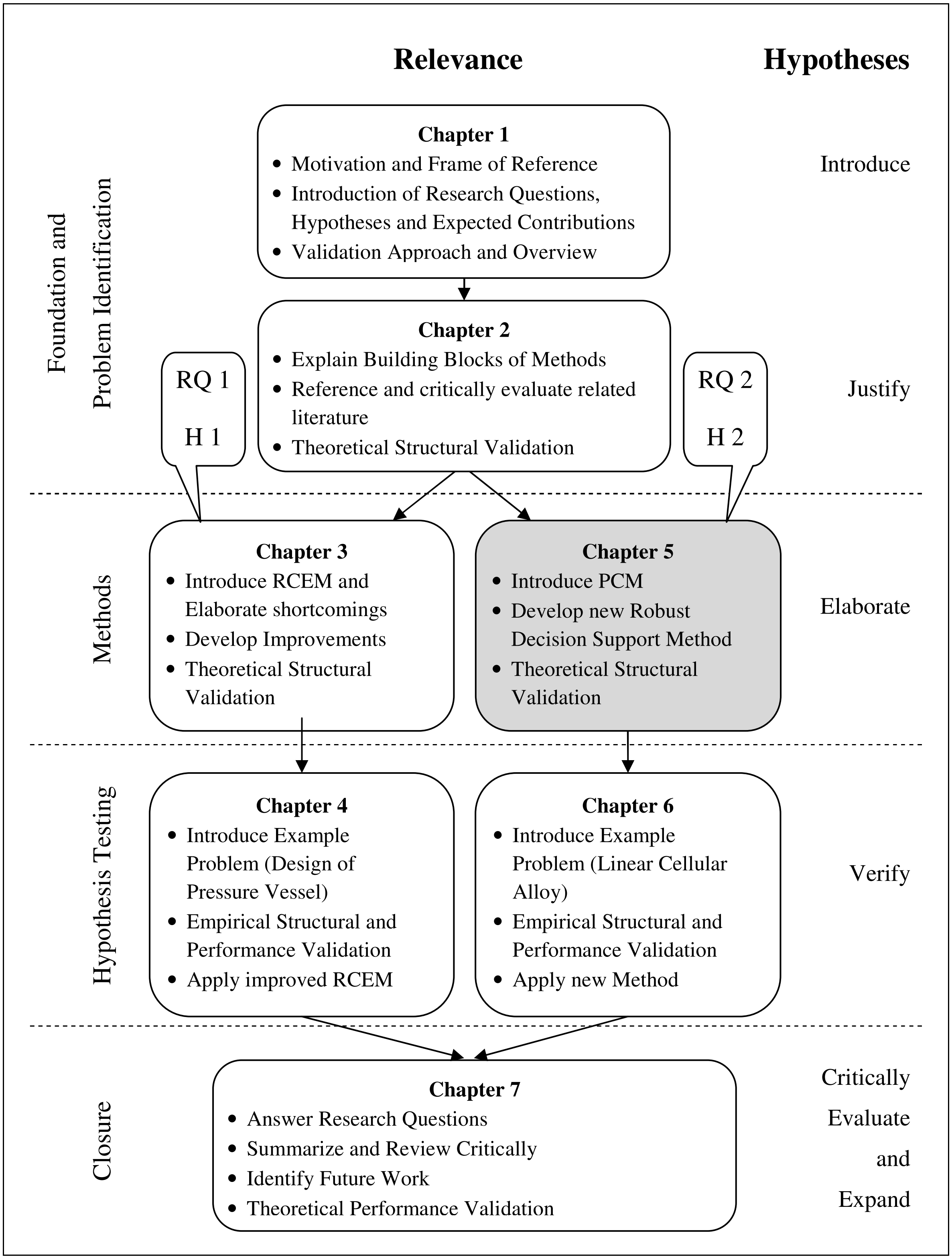

Relevance
Hypotheses
Foundation and
Problem Identification
Chapter 1
Motivation and Frame of Reference
Introduction of Research Questions, Hypotheses and Expected Contributions
Validation Approach and Overview
Introduce
Chapter 2
Explain Building Blocks of Methods
Reference and critically evaluate related literature
Theoretical Structural Validation
RQ 1
H 1
RQ 2
H 2
Justify
Methods
Chapter 3
Introduce RCEM and Elaborate shortcomings
Develop Improvements
Theoretical Structural Validation
Chapter 5
Introduce PCM
Develop new Robust Decision Support Method
Theoretical Structural Validation
Elaborate
Hypothesis Testing
Chapter 4
Introduce Example Problem (Design of Pressure Vessel)
Empirical Structural and Performance Validation
Apply improved RCEM
Chapter 6
Introduce Example Problem (Linear Cellular Alloy)
Empirical Structural and Performance Validation
Apply new Method
Verify
Closure
Chapter 7
Answer Research Questions
Summarize and Review Critically
Identify Future Work
Theoretical Performance Validation
Critically
Evaluate
and
Expand

Figure 5.5: Thesis Roadmap

CHAPTER 6

SIMULATION-BASED ROBUST DESIGN OF A LINEAR CELLULAR ALLOY COOLING SRTUCTURE

In the previous chapter, Research Question 2 is addressed. For the validation of the Research Hypothesis 2, the simulation-based Robust Concept Exploration Method is developed and its validation is discussed. Since the example in Chapter 5 is only one-dimensional, a multi-dimensional design example is presented in this chapter. In the heat exchanger example described in this chapter, the simulation-based RCEM is applied for designing a robust Linear Cellular Alloy (LCA) cooling structure. With multiple uncertain design variables, multiple objectives and the required MATLAB simulations the example is suitable for the validation of the new design method. The two main components of the simulation-based RCEM, the compromise Decision Support Problem (cDSP) and the Probabilistic Collocation Method (PCM), provide a set of tools to handle exactly these kinds of design problems. The cDSP assists in compromising between multiple objectives and the PCM handles uncertain simulation input.

While the simulation-based RCEM's ability of coping with non-linear models is demonstrated in Chapter 5, the focus of this chapter is in the efficiency of the new method. First, the LCA technology and the design problem are introduced briefly and the validation strategy is outlined. Then the simulation-based RCEM is applied to the example and the results are discussed.

6.1 Linear Cellular Alloy Cooling Structure

In this section, the Linear Cellular Alloy technology is introduced briefly and the cooling structure design example is presented.

The LCA design example is taken from the research area of multifunctional material design. In this case, the cooling structure is designed to provide a certain structural stiffness as well as high heat transfer rate for cooling applications. Similar examples have been used previously (e.g. (Seepersad, 2004; Seepersad, et al., 2006a; Seepersad, et al., 2004)). Some parts in the description of the LCA technology in the next section are taken from these publications.

6.1.1 Linear Cellular Alloy

Linear Cellular Alloys (LCAs) are extruded cellular honeycomb materials. Some examples are presented in Figure 6.1.

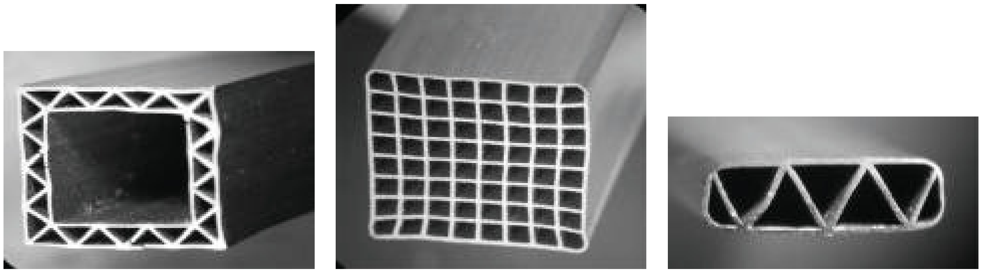


Figure 6.1: Examples of Linear Cellular Alloys (Seepersad, et al., 2006a)

Certain structural and thermal properties of these materials are superior to those of metallic foams with equivalent densities (see (Evans, et al., 2001) and (Hayes, et al., 2004)). This is why these linear or two-dimensional cellular materials are particularly suitable for multifunctional applications that require both structural performance and lightweight thermal or energy absorption capabilities. Due to larger surface area density and lower pressure drop, LCAs are suitable for applications as heat exchangers. Those two factors can compensate for lower heat transfer coefficients for laminar forced convection than for turbulent convection in stochastic metal foams with comparable relative densities (see (Lu, 1999)).

The manufacturing process for LCAs was developed by the Lightweight Structures Group at the Georgia Institute of Technology (Cochran, et al., 2000). To build LCAs, powder slurries are extruded through a die and then exposed to thermal and chemical treatments. Those structures can be produced in nearly any cellular topology, solely limited by paste flow and die manufacturability. In (Church, et al., 2001) it is reported that wall thicknesses as small as fifty microns and cell diameters of several hundred microns have been realized.

6.1.2 Design of the LCA Cooling Device

In this section, the boundary conditions and objectives for a structural heat exchanger for electronic cooling applications are outlined and the cDSP for the design example is formulated.

The cooling device that is designed in this chapter would be well-suited for application in electronic devices. The goal is to design a multifunctional heat exchanger that can provide a certain structural support in addition to the cooling capabilities. Areas of application could be for example laptop computers. For this example, only rectangular cell topologies are considered with an overall structure of the heat exchanger as presented in Figure 6.2. A topology with two rows and twelve columns with an overall height and width of 25 mm and an overall length of 75 mm is selected.

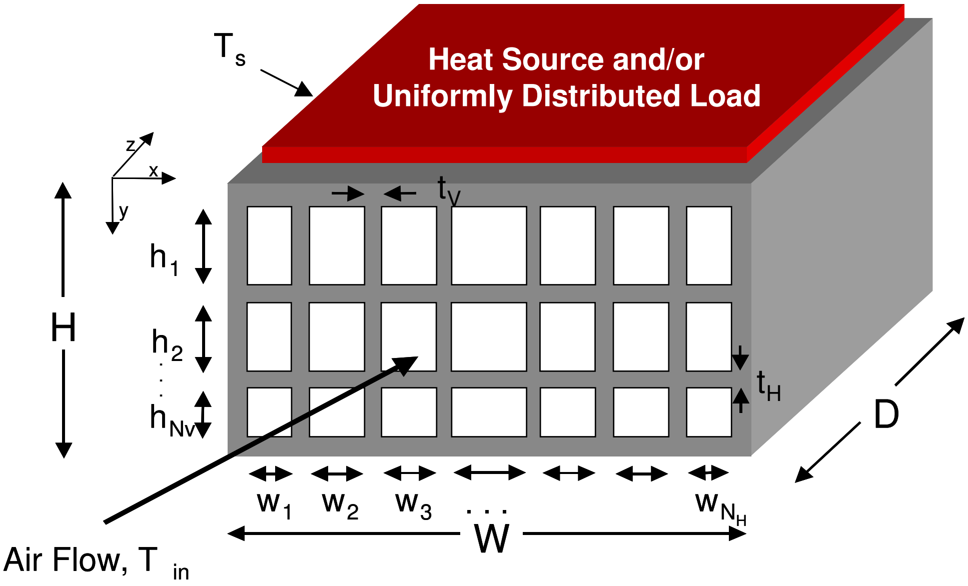


Figure 6.2: Compact, Forced Convection Heat Exchanger with Graded Rectangular LCAs (Seepersad, 2004)

The design variables, parameters, objectives, bounds and constraints are summarized in Table 6.1 and explained in the following.

Table 6.1: Nomenclature and Boundary Conditions for Chapter 6

Design Variables and Bounds	
μ_{t_H} (0.15 – 1.20 mm)	Mean of wall thickness in horizontal direction
μ_{t_V} (0.15 – 1.20 mm)	Mean of wall thickness in vertical direction
μ_{h_2} (3.00 – 22.00 mm)	Mean of height of bottom row
μ_{h_1} $(h_1 = H - h_2 - 3t_V)$	Mean of height of top row (determined by h_2)
Constants	
$H = 25$ mm	Overall height of the heat exchanger
$W = 25$ mm	Overall width of the heat exchanger
$L = 75$ mm	Overall length of the heat exchanger
$N_v = 2$	Number of rows
$N_H = 12$	Number of columns
w_i	Width of cells in column i (in this example the columns are evenly distributed)
$T_s = 373$ K	Temperature of heat source
$T_{in} = 293$ K	Temperature of air at inlet
$k_s = 363$ W/(m*K)	Thermal conductivity of solid base material
$\dot{V} = 0.00182$ m^3/s	Volumetric flow rate of air/coolant
$\sigma_{h_2}^2 = 1.0$ mm^2 $\sigma_{t_H}^2 = \sigma_{t_V}^2 = 0.0009$ mm^2	Variances of the bottom row height and the wall thickness in both directions
Objectives	
$\dot{Q}$	Total heat transfer rate (target: $\dot{Q}$ = 150 W)
$\sigma_{\dot{Q}}^2$	Variance of total heat transfer rate (target: $\sigma_{\dot{Q}}^2$ = 0.8 W^2)
Constraints	
$E_x/E_s \geq 0.01$	Overall elastic stiffness in x-direction, relative to the elastic modulus of the solid base material
$E_y/E_s \geq 0.01$	Overall elastic stiffness in y-direction, relative to the elastic modulus of the solid base material
Further constraints on the manufacturability that do not require the introduction of new variables are introduced in the following	

The overall design specifications boundary conditions are taken from (Seepersad, 2004), where a very similar example problem is used for the validation of a two-step topology design method. For this thesis, the problem is slightly modified to suit the requirements of the simulation-based RCEM.

The intended application for the heat exchanger demands for satisfactory performance in two physical domains: overall structural elastic stiffness and overall rate of steady state heat transfer.

The rectangular cell topology of the cellular structure of the heat exchanger can be tailored by adjusting the height of the two rows (μ_{h_1}, μ_{h_2}) as well as the thickness of the walls in x- and y-direction (μ_{t_H}, μ_{t_V}). For this example, the design variables are assumed to be uncertain due to manufacturing tolerances. The wall thicknesses are Gaussian distributed with a variance of 0.0009 mm^2 and the location of the horizontal wall between the two rows is Gaussian distributed with a variance of 1.0 mm^2. The device is intended to dissipate heat from the heat source with an elevated Temperature, T_s = 373 K, to cooling air that flows through the cellular alloy with a fixed volumetric flow rate, $\dot{V}$ = 0.00182 m^3/s, and a fixed entry temperature, T_{in} = 293 K. According to (Church, et al., 2001) the thermal conductivity, k_s, of copper samples fabricated with the thermal-chemical LCA extrusion process has been measured to be 363 W/(m*K). The goal is to design an exchanger that can provide a robust heat transfer rate with a target rate of $\dot{Q}$ = 150 Watts and a target variance of $\sigma_{\dot{Q}}^2$ = 0.8 W^2. These goals are formulated as design objectives in this example. For the calculation of the steady state heat transfer rate of the cooling device, a MATLAB Finite Element Analysis is implemented. The simulation has been developed by Carolyn Conner Seepersad, Hae-Jin Choi and Marco Gero Fernández at Georgia Tech.

Additionally, the cellular structure of the heat exchanger should provide a minimum elastic stiffness in the x- and y-directions in Figure 6.2. The application demands for a elastic stiffness relative to the modulus of the base material of at least 0.01 ($E_x/E_s \geq 0.01$, $E_y/E_s \geq 0.01$). For the calculation of these values a simplified approach is taken, which is presented in (Seepersad, 2004). In a prismatic cellular material with non-periodic rectangular cells that is loaded along the x- or y-axis in Figure 6.2, elastic deformation occurs due to axial extension or compression of the walls. If it is then assumed that there are no imperfections in structure and material, there is no bending contribution to the deformation in this particular configuration. Thus, the overall elastic stiffness in x- and y-directions can be approximated as the fraction of the total width *W* or height *H*, respectively, occupied by the cell walls:

$$E_x / E_s \cong \frac{\mu_{t_H} (N_V + 1)}{H} \geq 0.01 \qquad \text{Eq. 6.1}$$

$$E_y / E_s \cong \frac{\mu_{t_V} (N_H + 1)}{W} \geq 0.01 \qquad \text{Eq. 6.2}$$

where E_s is the elastic modulus of the solid cell wall material. In this design example, the minimum requirement for this fraction in both directions is 0.01.

While meeting the design requirements, the heat exchanger has to be manufacturable. In (Seepersad, et al., 2004) it is reported that, as a rule of thumb, the aspect ratio of cell wall thickness to cell dimension has to be 0.08 or above to guarantee manufacturability (Eq. 6.3, Eq. 6.4, Eq. 6.5).

Eq. 6.3

$$\frac{12 \cdot \mu_{t_H}}{W - (13 \cdot \mu_{t_H})} \geq 0.08$$

Eq. 6.4

$$\frac{\mu_{t_V}}{h_1} \geq 0.08$$

Eq. 6.5

$$\frac{\mu_{t_V}}{h_2} \geq 0.08$$

Furthermore, the MATLAB simulation used to calculate the heat transfer rate is intended for predominantly open structures, which demands for the constraints given in Eq. 6.6, Eq. 6.7, and Eq. 6.8.

Eq. 6.6

$$\frac{12 \cdot \mu_{t_H}}{W - (13 \cdot \mu_{t_H})} \leq 1.00$$

Eq. 6.7

$$\frac{\mu_{t_V}}{h_1} \leq 1.00$$

Eq. 6.8

$$\frac{\mu_{t_V}}{h_2} \leq 1.00$$

In addition, the row height should not be smaller than 3 mm. For the bottom row, *h_2*, this is ensured through the bound on the variable. For the top row, *h_1*, the constraint given in Eq. 6.9 has to be formulated.

$$H - 3 \cdot \mu_{t_V} - \mu_{h_2} \geq 3mm \qquad \text{Eq. 6.9}$$

The design problem described in this section is used for the validation of the simulation-based RCEM in Section 6.3. In that section, the cDSP for the design problem is formulated, too. In the next section, the strategy for the validation with this example is outlined.

6.2 Validation and Verification with the Example

In this section, the strategy for the validation of the simulation-based RCEM with the LCA example is outlined. As presented in Section 1.4, the validation square provides the structure for the validation of this thesis. The validation square consists of four quadrants: Theoretical Structural Validation, Empirical Structural Validation, Empirical Performance Validation, and Theoretical Performance Validation (see Figure 1.6). The theoretical structural validation and the empirical structural validation are demonstrated in this section. The empirical performance validation is outlined in this section and discussed in Section 6.4. The theoretical performance validation of both Hypothesis 1 and 2 is discussed in Chapter 7.

6.2.1 Theoretical Structural Validation

The theoretical structural validation of the simulation-based RCEM is discussed and demonstrated in Section 5.2.1 and therefore not elaborated here.

6.2.2 Empirical Structural Validation

Demonstrating the empirical structural validity involves discussing the appropriateness of the chosen example for illustrating and verifying the design method. The simulation-based Robust Concept Exploration Method, which is introduced in Section 5.1.2, is validated with the LCA design example in this chapter.

Is the example similar to design problems for which the simulation-based RCEM is intended and may be applied?

The following characteristics of the Linear Cellular Alloy problem qualify it as a suitable problem for the simulation-based RCEM:

- The design of the LCA cooling structure is modeled in multiple MATLAB functions that require a simulation each time the performance of a certain design variable configuration is evaluated. This requires the design method to be able of handling problems where the functional relationship between input and output is not known explicitly.
- The design and the performance of the LCA cooling structure is determined by three design variables. This means that the example is a multidimensional design problem that requires a method that can handle multiple input variables.

- The performance of the design is dependent of several uncertain factors. In this case, the three design variables contain uncertainty in form of manufacturing inaccuracy. Showing that the simulation-based RCEM is capable of handling multiple uncertain design variables implies that the method can also solve problems with multiple noise factors. In these design problems, noise factors can be regarded as uncertain "design variables" with a constant mean value.
- The objective is to design a cooling structure that is robust towards uncertain design variables or noise factors. This means that the goal is to find a design variable configuration, which results in a low variation of the heat transfer rate without changing or modifying the source of uncertainty. A method that can help solving this kind of problems requires a means to estimate accurately the robustness of the design at various points in the design space.
- The example design problem has multiple design objectives that have to be considered by the designer. The goal is to find a compromise among the several objectives. This example has one performance objective (high heat transfer rate) and one robustness objective (low variance of the heat transfer rate). Since these objectives oppose each other, a trade-off has to be found that satisfies the designer's preferences.

Can the data from these examples be used to support conclusions with respect to the hypothesis?

In order to validate Research Hypothesis 2, it has to be shown that the simulation-based RCEM increases accuracy and reliability over the original RCEM. Accuracy in this context means that the new method’s estimations for the performance mean and variance has to be closer to the real values and that the application of the method actually lead to robust design solutions. Reliability means that the new method has to be able to bring better results especially for highly non-linear problems, for which the original RCEM fails. Both improvements in accuracy and reliability are shown with data from the first example in Sections 5.3 and 5.4. This can be supported by the data from the second example, the design of a LCA cooling structure. This data is used especially to evaluate the ability of the method to solve multidimensional, multi-objective problems with multiple uncertain parameters. Furthermore, it can support the conclusion that the new method can handle simulation-based problems effectively and efficiently without requiring a global response surface or a similar global surrogate model.

6.2.3 Empirical Performance Validation

The LCA example is used to demonstrate the empirical performance validation of Hypothesis 2 and the simulation-based RCEM. Therefore, it has to be shown (1) that the results meet the expectations, (2) that the demonstrated effectiveness and efficiency is linked to the application of the new method and (3) that the data generated and used for validation is accurate and internally consistent.

How can the results be evaluated with respect to the expected outcome of the method?

The results for the example problem are generated for two different scenarios, of which the first one asks the designer to focus on maximum performance and the second on a compromise between high performance and high robustness (low performance variability). If the results for the second scenario are design solutions that are more robust than the results for the first scenario, the outcome is as expected and Hypothesis 2 is supported.

How can it be demonstrated that the observed effectiveness and efficiency is linked to the application of the simulation-based RCEM?

The results of the LCA example will allow conclusions especially about the efficiency of the new method. The efficiency is regarded as inversely proportional to the computational costs. A comparison of the required computational time of applying the simulation-based RCEM to the time required for using the original RCEM will be used to support the Research Hypothesis 2.

How are the accuracy and the internal consistency of the results verified?

First the accuracy of the results is verified by comparing the outcome of the simulation-based RCEM to Latin hypercube samplings. Showing that the results of the simulation-based RCEM are close to those values will support the argument for the accuracy of the method. The internal consistency of the results is confirmed by testing multiple starting points for the algorithm as well as by an analysis of the convergence of the algorithm.

In this section, the validation strategy for the LCA example is outlined. In the next section, the data for this validation is generated by applying the simulation-based RCEM to the LCA design problem.

6.3 Application of the Simulation-Based RCEM to the LCA Design Example

In this section, the simulation-based RCEM is applied to the LCA design problem, which is introduced in Section 6.1.2. First, the problem is solved for two scenarios using the new method. The first scenario demands for a cooling device with maximum heat transfer with less focus on the robustness of the performance. The second design scenario, on the other hand, requires the designer to find a compromise between a high heat transfer rate and a low performance variation. The results of the method are verified using a Latin hypercube sampling. Furthermore, the computational costs for applying the simulation-based RCEM are compared to those of the original RCEM.

For the application of the simulated RCEM to the LCA design example, first, the problem has to be formulated as a cDSP. The cDSP generated from the information given in Section 6.1.2 is presented in Figure 6.3.

GIVEN

FEA analysis simulation of LCA structure in MATLAB
Constants and boundary conditions (Table 6.1)
Variance of design variables $\sigma_{h_2}^2 = 1.0 \text{ mm}^2$, $\sigma_{t_H}^2 = \sigma_{t_V}^2 = 0.0009 \text{ mm}^2$

FIND

Design variable: μ_{t_H}, t_V, h_2
Deviation variables: d_i^-, d_i^+ $i = 1,2$

SATISFY

CONSTRAINTS

on minimum structural stiffness (see Eq. 6.1 and Eq. 6.2):

$$E_x / E_s \cong \frac{\mu_{t_H}(N_V+1)}{H} \geq 0.01; \; E_y / E_s \cong \frac{\mu_{t_V}(N_H+1)}{W} \geq 0.01$$

on manufacturability (see Eq. 6.3, Eq. 6.4, and Eq. 6.5):

$$\frac{12 \cdot \mu_{t_H}}{W - (13 \cdot \mu_{t_H})} \geq 0.08 \;; \qquad \frac{\mu_{t_V}}{h_1} \geq 0.08 \;; \qquad \frac{\mu_{t_V}}{h_2} \geq 0.08$$

on applicability of simulation (see Eq. 6.6, Eq. 6.7, and Eq. 6.8):

$$\frac{12 \cdot \mu_{t_H}}{W - (13 \cdot \mu_{t_H})} \leq 1.00 \;; \qquad \frac{\mu_{t_V}}{h_1} \leq 1.00 \;; \qquad \frac{\mu_{t_V}}{h_2} \leq 1.00$$

on minimum cell height (see Eq. 6.9):

$$H - 3 \cdot \mu_{t_V} - \mu_{h_2} \geq 3mm$$

GOALS

Performance goal: $\frac{\dot{Q}}{150W} + d_1^- - d_1^+ = 1$

Robustness goal: $\frac{1 / \sigma_{\dot{Q}}^2}{1 / 0.8W^2} + d_2^- - d_2^+ = 1$

BOUNDS

$$0.15mm \leq \mu_{t_H} \leq 1.20mm$$

$$0.15mm \leq \mu_{t_V} \leq 1.20mm$$

$$3.00mm \leq \mu_{h_2} \leq 22.00mm$$

$$d_i^-, d_i^+ \geq 0 \qquad i = 1,2$$

$$d_i^- \cdot d_i^+ = 0 \qquad i = 1,2$$

MINIMIZE

$$Z = W_1 \cdot d_1^- + W_2 \cdot d_2^- \qquad d_1^+, d_2^+ = 0$$

Scenario 1:	Scenario 2:
$W_1 = 1.0$	$W_1 = 0.2$
$W_2 = 0.0$	$W_2 = 0.8$

Figure 6.3: cDSP for LCA Example Design Problem

For this example, it has been found that a second order polynomial chaos expansion without cross product terms for the probabilistic collocation method is sufficiently accurate with an R^2 around 99%. The polynomial chaos expansion used here is given in Eq. 6.10.

$$\begin{aligned}\dot{Q} \approx a_0 &+ a_1 H_1\left(\xi_{t_H}\right) + a_2 H_1\left(\xi_{t_V}\right) + a_3 H_1\left(\xi_{h_2}\right) \\ &+ a_4 H_2\left(\xi_{t_H}\right) + a_5 H_2\left(\xi_{t_V}\right) + a_6 H_2\left(\xi_{h_2}\right)\end{aligned} \qquad \text{Eq. 6.10}$$

with H_1 and H_2 being the first and second order Hermite polynomials. According to Eq. 5.1 in Section 5.1.1 the design variable have the following characteristics:

$$t_H = \mu_{t_H} + \sigma_{t_H}\left(H_1\left(\xi_{t_H}\right)\right) \qquad \text{Eq. 6.11}$$

$$t_V = \mu_{t_V} + \sigma_{t_V}\left(H_1\left(\xi_{t_V}\right)\right) \qquad \text{Eq. 6.12}$$

$$h_2 = \mu_{h_2} + \sigma_{h_2}\left(H_1\left(\xi_{h_2}\right)\right) \qquad \text{Eq. 6.13}$$

Having seven unknowns, a_i, means that each time the performance and robustness is evaluated using the PCM in this example, seven MATLAB FEA simulations of the LCA are required.

In the following, the results of the second order simulation-based RCEM for the design example are presented.

First, the simulation-based RCEM results for design scenario 1 are generated. This scenario is equivalent to an optimization problem without the consideration of the robustness of the performance. However, in order to evaluate the results of the second scenario the variances of the total heat transfer rate in Scenario 1 are calculated and presented together with the results in Table 6.2. The results are verified using a Latin hypercube sample with 1000 sample points; the LHS outcome is given in parenthesis.

Table 6.2: LCA Design Results for Scenario 1

Run #	Initial Point $\{t_H, t_V, h_2\}$ in [mm]	Solution of the S-B RCEM $\{t_H, t_V, h_2\}$ in [mm]	Mean of $\dot{Q}$ estimated with the S-B RCEM (LHS)	Variance of $\dot{Q}$ estimated with the S-B RCEM (LHS)
1	{0.5, 0.5, 3}	{1.00, 1.14, 8.43}	135.70 (135.70)	1.50 (1.50)
2	{0.15, 1.2, 10}	{1.00, 1.20, 8.94}	135.57 (135.57)	1.53 (1.52)
3	{1.2, 0.15, 10}	{1.00, 1.20, 8.53}	135.65 (135.65)	1.54 (1.54)
4	{0.5, 0.5, 20}	{1.00, 1.20, 8.98}	135.57 (135.57)	1.52 (1.53)

From the results it can be seen that the target value for the total heat transfer rate could not be achieved and that 135.70 W is the highest heat transfer rate that can be achieved within the given bounds and constraints. Furthermore, it can be seen that the solutions from the simulation-based RCEM are very similar of those from the Latin hypercube sample, which leads to the conclusion that the simulation-based RCEM is very accurate in this example. The similar solutions that are generated from the different starting points indicate the internal consistency of the results. This is confirmed by the smooth convergence of the algorithm. The convergence plot of run #1 is given in Figure 6.4.

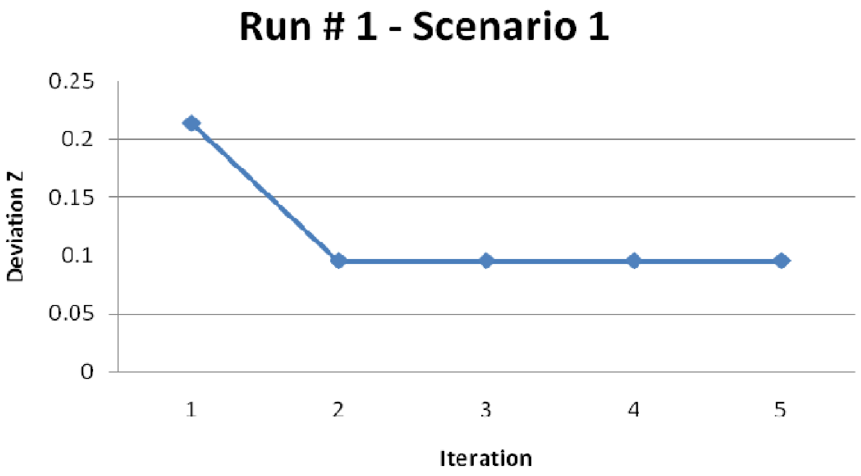


Figure 6.4: Convergence Plot for Run #1 in Scenario 1

With a heat transfer rate variance of about 1.50, the robustness of the performance is relatively low in scenario 1. The simulation-based RCEM results for scenario 2, which are presented in Table 6.3, have a significantly higher robustness than the solutions from the optimization in Scenario 1.

Table 6.3: LCA Design Results for Scenario 2

Run #	Initial Point $\{t_H, t_V, h_2\}$ in [mm]	Solution of the S-B RCEM $\{t_H, t_V, h_2\}$ in [mm]	Mean of $\dot{Q}$ estimated with the S-B RCEM in [W] (LHS)	Variance of $\dot{Q}$ estimated with the S-B RCEM in [W^2] (LHS)
1	{0.5, 0.5, 3.0}	{0.15, 0.89, 11.17}	103.36 (103.36)	1.02 (1.02)
2	{0.15, 1.2, 10.0}	{0.15, 0.92, 10.97}	103.25 (103.26)	1.03 (1.02)
3	{1.2, 0.15, 10.0}	{0.15, 1.00, 12.51}	102.83 (102.83)	1.03 (1.05)
4	{0.5, 0.5, 20.0}	{1.00, 0.89, 11.16}	135.33 (135.33)	1.35 (1.35)

In run #1 the best results can be observed. Both the mean and the variance of the heat transfer rate decrease from 135.70 Watt to 103.36 Watt and 1.50 W^2 to 1.02 W^2, respectively compared to Scenario 1. From this, the effectiveness of the simulation-based RCEM can be concluded. The solutions presented in Table 6.3 are compromises between high performance and high robustness. Furthermore, the estimations for the performance mean and the variance of the new method are again very similar to the LHS outcomes. This observed accuracy can be linked to the Probabilistic Collocation Method used in the simulation-based RCEM. The internal consistency of the results is demonstrated using different initial points. Most of them converge close to the robust solution. Only run #4 converged to a solution that is obviously not satisfying. This can be seen when looking at the deviation function values, which are 0.237 for run #1 and 0.346 for run #4. The convergence plot of run# 1 (see Figure 6.5), in which a smooth convergence can be seen, supports the internal consistency of the results for Scenario 2.

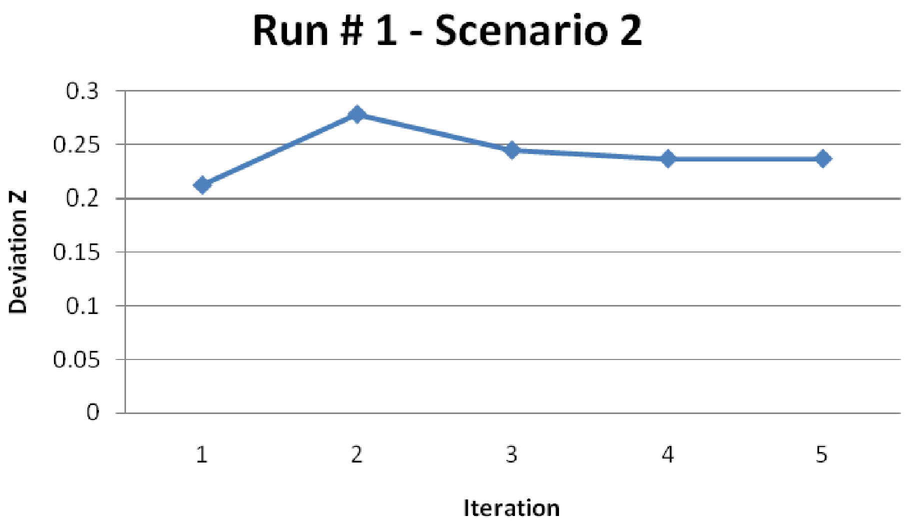


Figure 6.5: Convergence Plot for Run #1 in Scenario 2

With the presented data, the effectiveness of the simulation-based RCEM can be validated. However, in this chapter there is a special focus also on the efficiency of the new method in comparison to the original RCEM and the brute force LHS approach. For solving the LCA design problem in run #1 of scenario 2 the JavaDSIDES algorithm has evaluated the performance and robustness 106 times. This means that the PCM has been invoked 106 times with seven simulations per run. This is a total of 742 LCA FEA MATLAB simulations, which takes a Windows XP PC with an Intel Core2 Quad CPU at 2.4 GHz and 3.25 GB of RAM eleven minutes and one seconds to solve. This is about 0.9 seconds per simulation. Using a Latin hypercube approach as presented in Figure 4.6 with 1000 sample points for each performance evaluation would require 106,000 simulations, which would take about 2 days 2 hours and 30 minutes. This is almost 150 times as much time as the simulation-based RCEM takes. For the application of the original RCEM first, the response surface has to be built. Assuming a full factorial design and 20 sample points for each dimension (t_H, t_V, h_2), 8000 simulations are required just for building the surrogate model. Ignoring the computational time for applying the original RCEM itself, including calculating the required derivative, this would take about 2 hours; this is more than 10 times as much as the simulation-based RCEM. A summary of the computational cost of the three methods is given in Table 6.4.

Table 6.4: Comparison of the Computational Costs

Method	Number of Required LCA FEA Simulations	Required Computational Time
Simulation-Based RCEM	742	11 Minutes 1 Second
Original RCEM	about 8,000	about 2 hours
Brute Force LHS approach	about 106,000	about 2 Days 2 Hours 30 Minutes

Another advantage of using the Probabilistic Collocation Method is that the local response can be used to estimate higher statistical moments of the performance or even the probability density function. For the design solution of run # 1 in Scenario 2 of this example the first four (standardized) statistical moments (mean, variance, skewness and kurtosis) as well as a histogram of the performance are generated in the following. The outcome is compared to a Latin hypercube sampling. The skewness of a sample or a distribution is defined as the third standardized moment(see (Wasserman, 2004)):

$$\gamma_1 = \frac{E[(x-\mu)^3]}{\sigma^3} \qquad \text{Eq. 6.14}$$

The skewness is a measure of the asymmetry of the probability density function of a random variable. In the LCA example, this random variable is the total heat transfer rate $\dot{Q}$. The kurtosis is a measure of how "peaked" a PDF is; this means a high kurtosis indicates that more of the variance is due to infrequent extreme deviations, as opposed to frequent more modest sized deviations. Equivalent to the skewness, the kurtosis is defined as the fourth standardized moment (see (Joanes and Gill, 1998)):

$$\gamma_2 = \frac{E[(x-\mu)^4]}{\sigma^4} \qquad \text{Eq. 6.15}$$

The skewness, the kurtosis, a histogram or the probability density function of the performance of a design can provide important information for the designer, depending on the design preferences. In this section, only the possibility of estimating these values is presented, however, their role in engineering design is not discussed in this thesis.

Using the Probabilistic Collocation Method on the considered design solution, a local surrogate model is built, on which a Latin hypercube sampling can be performed. In this example this requires seven LCA FEA simulations (see above) and, when using 10,000 sample points, 10,000 model function calls. From these samples the statistical moments can be calculated and a histogram can be generated. The entire process takes less than 7 seconds of computational time. The outcome of the PCM is compared to a brute force 10,000 points LHS simulation. Executing the 10,000 simulations takes about 2 hours and 10 minutes; 1000 times longer than the PCM. In Figure 6.6, the histograms of the two approaches are presented. It can be seen that they are very similar. This is confirmed by the moment estimations from both methods, which are shown in Table 6.5. Furthermore, the samples from both histograms are tested with the two-sample Kolmogorov-Smirnov test (Young, 1977). The test is used for testing if two samples are from the same distribution. The null hypothesis is that the samples have the same underlying distribution. In this example, the null hypothesis is not rejected on a 99.9% confidence level and thus, it can be assumed that the samples have the same underlying distribution. This is another indication for the very accurate estimation of the probability density function with the PCM.

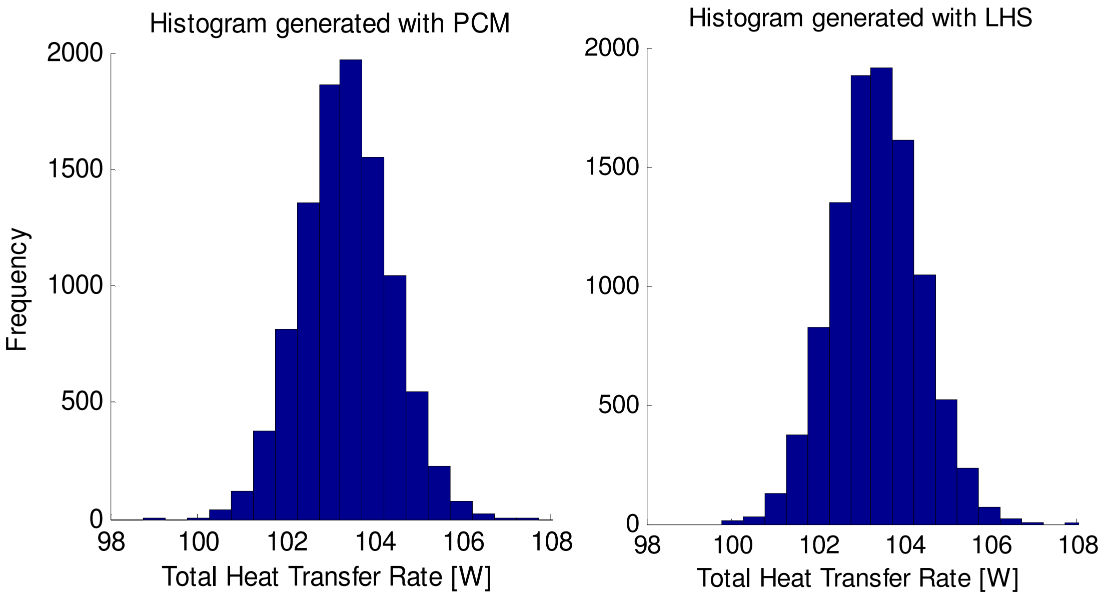


Figure 6.6: Histograms of the Performance of the Design Solution from Run #1 in Scenario 2

Table 6.5: Comparison of Moment Estimation

Heat Transfer Rate	**Probabilistic Collocation Method**	**Brute Force Latin Hypercube Sampling**
Mean	103.36 W	103.36 W
Variance	1.02 W^2	1.02 W^2
Skewness	0.04	0.05
Kurtosis	3.03	3.02

The MATLAB code required to perform the Probabilistic Collocation Method on the LCA heat exchanger example is given in Appendix C.2. By integrating this code to JavaDSIDES following the instructions in the JavaDSIDES Manual (see Appendix A) the simulation-based RCEM can be performed.

In this section, the simulation-based RCEM is applied to the LCA heat exchanger design example for the validation of Research Hypothesis 2. The results are presented and compared to alternative methods for verification. In the next section, the results are critically discussed and related to the validation strategy outlined in Section 6.2.

6.4 Critical Discussion of Example Results

In this section, the results presented in the previous section are discussed and related to the validation of Research Hypothesis 2. As outlined for the entire thesis in Section 1.4 and specifically for the LCA heat exchanger in Section 6.2, the Validation Square is used for the validation of the research presented here. This means that the theoretical structural validity, the empirical structural validity, the empirical performance validity and the theoretical performance validity of the simulation-based RCEM have to be demonstrated for the validation of Research Hypothesis 2. In this section, this is done for the results from the heat exchanger example.

The **theoretical structural validity** of the simulation-based RCEM is discussed and demonstrated in Section 5.2.1. It is shown that the components of the simulation based RCEMare valid and that the information flow is internally consistent.

The **empirical structural validity** for the LCA example is discussed in Section 6.2.2 by demonstrating that the heat exchanger example is appropriate for the validation of the simulation-based RCEM.

For confirming the **empirical performance validity** it has to be shown that (1) the results meet the expectations, that (2) the demonstrated effectiveness and efficiency is linked to the application of the new method and that (3) the data generated and used for the validation is accurate and consistent.

(1) Do the results of the LCA heat exchanger design problem meet the expected outcome?

It is expected that by using the simulation-based RCEM for design scenario 2, where a robust solution is desired, a design solution can be found, which has a higher robustness than the result for Scenario 1, where a optimizing solution is preferred. When comparing the results in Table 6.2 and Table 6.3, it can be seen that this expectation is met. Since the results for both scenarios are generated using the simulation-based RCEM it is also demonstrated that modifying the weight for the objectives changes the compromise that is made.

(2) Is the demonstrated effectiveness and efficiency linked to the application of the new method?

The focus of this chapter is mainly the on validating the improvement in efficiency of the simulation-based RCEM over the original RCEM. The improvement in effectiveness is shown in Chapter 5. However, the effectiveness of the new method is also evident for the LCA example. It is demonstrated by meeting the expectations and by verifying the results in Table 6.2 and Table 6.3 with a Latin hypercube sampling. The presented efficiency can be clearly linked to the application of the simulation-based RCEM as well. It is shown that the computational cost for implementing the new method is significantly lower than for the application of the original RCEM (see Table 6.4). This is especial due to the global response surface that has to be generated as part of applying the original

RCEM. Additionally, further advantages of new method are demonstrated, which include for example the efficient estimation of higher statistical moments of the performance.

(3) Is the data generated and used for this validation accurate and internally consistent?

The internal consistency of the data is demonstrated by applying the new method to the different starting points of the LCA heat exchanger design (see Table 6.2 and Table 6.3). Except for one starting point all runs converge very close to the same satisfying solution. Furthermore, the internal consistency of the results is confirmed by the smooth convergence of the algorithm, which is presented with the convergence plots in Figure 6.4 as well as Figure 6.5.

The accuracy of the results is verified by comparing the outcome of the simulation-based RCEM to computationally expensive Latin hypercube samplings with sampling sizes of 1,000 and 10,000 respectively. With these comparisons, it is shown that the accuracy of the new method is as expected.

For the validation of the simulation-based RCEM and Research Hypothesis 2 only the demonstration of the theoretical performance validation is missing. This quadrant of the validation square is part of the discussion in Chapter 7.

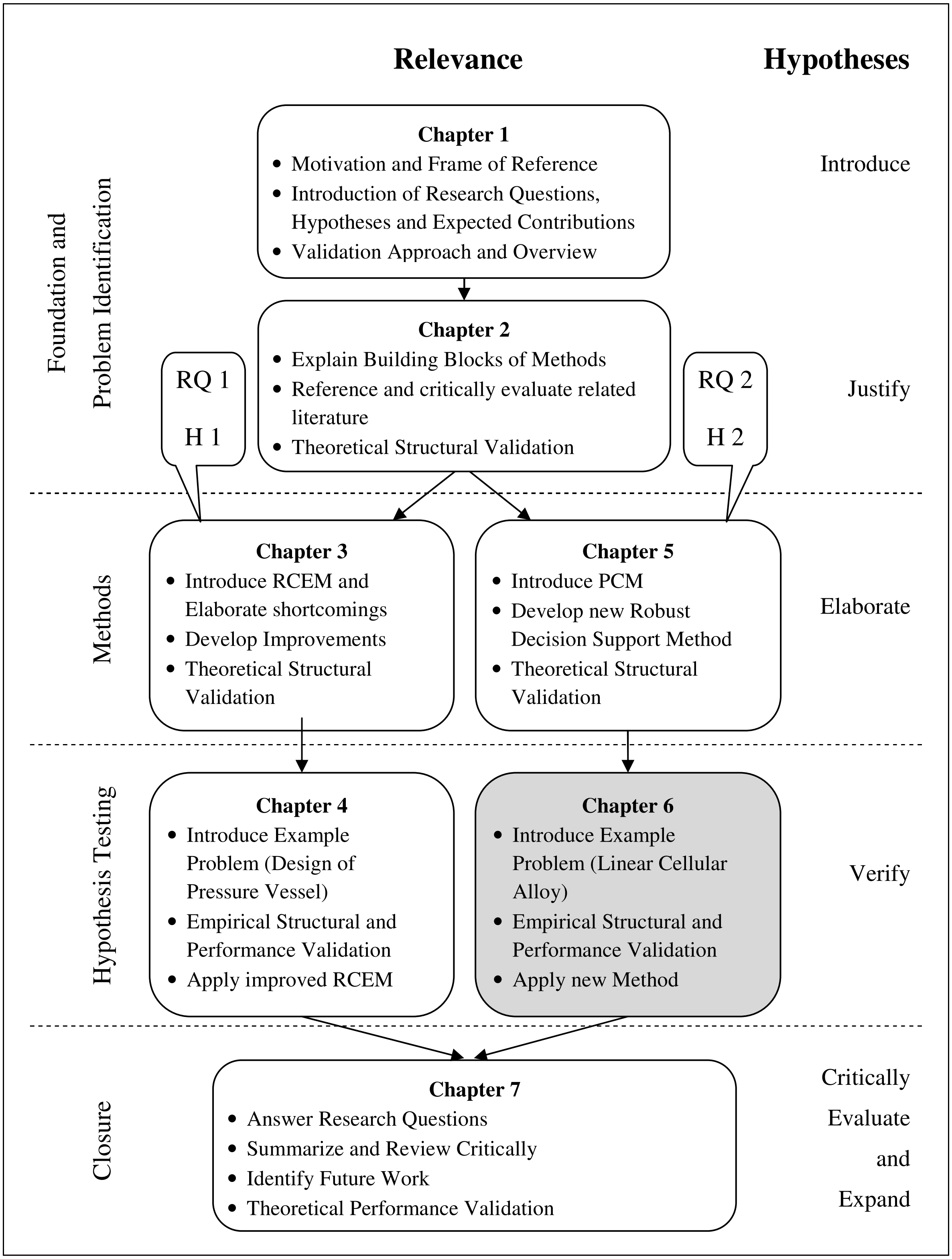

Relevance
Hypotheses
Foundation and Problem Identification
Chapter 1
Motivation and Frame of Reference
Introduction of Research Questions, Hypotheses and Expected Contributions
Validation Approach and Overview
Introduce
Chapter 2
Explain Building Blocks of Methods
Reference and critically evaluate related literature
Theoretical Structural Validation
RQ 1
H 1
RQ 2
H 2
Justify
Methods
Chapter 3
Introduce RCEM and Elaborate shortcomings
Develop Improvements
Theoretical Structural Validation
Chapter 5
Introduce PCM
Develop new Robust Decision Support Method
Theoretical Structural Validation
Elaborate
Hypothesis Testing
Chapter 4
Introduce Example Problem (Design of Pressure Vessel)
Empirical Structural and Performance Validation
Apply improved RCEM
Chapter 6
Introduce Example Problem (Linear Cellular Alloy)
Empirical Structural and Performance Validation
Apply new Method
Verify
Closure
Chapter 7
Answer Research Questions
Summarize and Review Critically
Identify Future Work
Theoretical Performance Validation
Critically Evaluate and Expand

Figure 6.7: Thesis Roadmap

6.5 Chapter Synopsis

As shown in Figure 6.7, the focus in this chapter is on the empirical structural validation and the empirical performance validation of the simulation-based RCEM. The effectiveness and efficiency of the new method is shown with its application to the LCA heat exchanger design problem. It is demonstrated that robust solutions in problems can be found, which require simulations for the performance evaluation. This chapter concludes the generation of data for the validation of the Research Hypotheses 1 and 2. For the final validation of both hypotheses everything but the theoretical performance validation of the modified RCEM and the simulation-based RCEM is presented in this and in the previous chapters. The last quadrant is discussed in the next chapter. Chapter 7 also contains the closure of this thesis including an outlook and the identification of possible further research directions.

CHAPTER 7

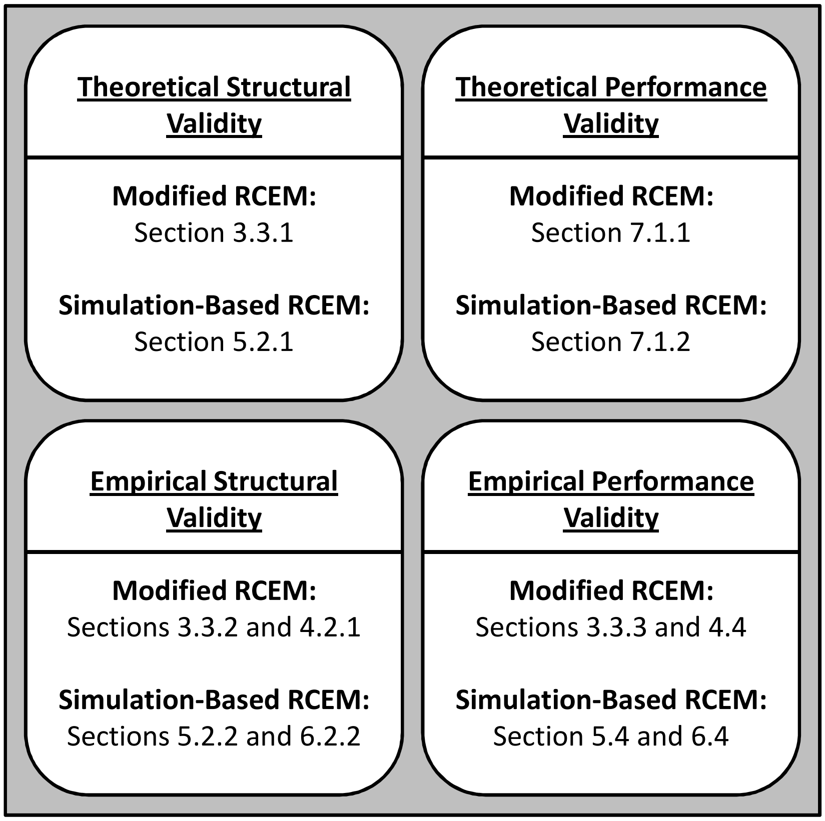
CLOSURE

The principal goal in this thesis is on further developing the Robust Concept Exploration Method based on the Research Questions and Hypotheses introduced in Section 1.3.

This chapter concludes this thesis with a critical analysis of the work presented. In Section 7.1, the final validation of the modified RCEM and the simulation-based RCEM is discussed. The achievements and research contributions made in this thesis are summarized in Section 7.2. Finally, an analysis of the limitations of the presented research as well as an outlook into research opportunities based on this work is given in Section 7.3.

7.1 Answering the Research Questions and Validation of the Research Hypotheses

In this section, the Research Questions proposed in Chapter 1 are answered by validating the Research Hypotheses. Therefore, the validation of the two developed methods is summarized here in context of the validation square (see Section 1.4 and Figure 7.1). In Section 7.1.1, Research Hypothesis 1 and the modified RCEM (see Chapters 3 and 4) is addressed. The Validation of Hypothesis 2 and the simulation-based RCEM (see Chapters 5 and 6) is discussed in Section 7.1.2.



Theoretical Structural Validity	Theoretical Performance Validity
Modified RCEM: Section 3.3.1 **Simulation-Based RCEM:** Section 5.2.1	**Modified RCEM:** Section 7.1.1 **Simulation-Based RCEM:** Section 7.1.2
Empirical Structural Validity	**Empirical Performance Validity**
Modified RCEM: Sections 3.3.2 and 4.2.1 **Simulation-Based RCEM:** Sections 5.2.2 and 6.2.2	**Modified RCEM:** Sections 3.3.3 and 4.4 **Simulation-Based RCEM:** Section 5.4 and 6.4

Figure 7.1: Validation of this Thesis in Context of the Validation Square

7.1.1 Validation of Research Hypothesis 1

In this thesis, several limitations of the original RCEM are presented (see Section 3.1). In RCEM, the robustness is measured with the variance of the performance. One of the limitations of the original RCEM is that this variance is estimated using a first-order Taylor Series approximation of the engineering system's performance. Because of this, the original RCEM may converge in local optima of the performance without considering the robustness of the system in non-linear and multi-modular design problems. This gap is addressed in the first research question:

First Research Question:

How can the formulation of robustness in compromise Decision Support Problems for multi-dimensional and multi-objective design problems be improved?

To answer this question the following Research Hypothesis is validated in this thesis:

First Research Hypothesis:

Using a global response surface, the nominal mean and a single point first order Taylor Series Approximation for the variance estimation is not accurate enough for finding robust solutions of nonlinear design problems. The Robust Concept Exploration Method can be improved by using appropriate formulations that account for the nonlinearity of the problem.

Hypothesis 1 is validated in this thesis with the development of the modified RCEM in Chapter 3. As outlined in Section 1.4, the validation in this thesis follows the four quadrants of the validation square. The theoretical structural validity, the empirical structural validity and the empirical performance validity of the modified RCEM are demonstrated with two examples, an artificial design problem (Chapter 3) as well as the design of a robust pressure vessel (Chapter 4) and are summarized here. Afterward, the theoretical performance validity is discussed and Hypothesis 1 is validated.

The theoretical structural validity of the modified RCEM is demonstrated in Section 3.3.1 by showing the internal consistency of the individual components of the method as well as the consistency of their assembly.

The empirical structural validity of the new method is demonstrated by arguing that the examples used for the validation are representative for engineering design problems the method is designed for and might be applied to. In Section 3.3.2, the specific characteristics of the artificial design problem that qualify it as an appropriate example, are pointed out. The same is done in Section 4.2.1 for the pressure vessel example.

The empirical performance validity of the modified RCEM is demonstrated with the data from the two example problems. In Section 3.3.3, it is shown that the generated results from the first example are as expected and that the improvements over the original RCEM are directly linked to the modifications. This is confirmed with the data from the pressure vessel example in Section 4.4. The solutions from this example are also as expected and support the validity of the modified RCEM.

As discussed in Section 1.4, theoretical performance validation involves establishing confidence that the method is applicable and useful for problems beyond the examples. This step is the logical consequence from validating the first three quadrants of the validation square. For the modified RCEM, it is demonstrated that the method's mathematics and its internal information flow is consistent, that the example problems used are appropriate, that the results are valid and as expected and that the observed effectiveness and efficiency is linked to the application of the new method. Based on this, its usefulness for general classes of problems is inferred.

In Hypothesis 1, it is hypothesized that by using modified formulations and estimations for the mean and the variance of the performance, the reliability of RCEM in non-linear design problems can be improved. The modified RCEM was successfully applied to a non-linear, multi-modular design problem, which the original RCEM failed to solve correctly. The modifications in the new method do not limit the functionality or the applicability of RCEM and remain the same efficiency. From these findings, the validity of Hypothesis 1 is inferred.

7.1.2 Validation of Research Hypothesis 2

The second limitation of the original RCEM, which is demonstrated in Section 3.1, is the generation of a global response surface for simulation-based design problems. The inaccuracy in these models can lead to fatal errors in highly non-linear design problems. This gap is addressed in Research Question 2:

Second Research Question:

How can the RCEM be augmented to improve the accuracy of the robust concept exploration for nonlinear design problems that involve simulations?

This question is answered in this thesis with the validation of the second Research Hypothesis:

Second Research Hypothesis:

It is hypothesized that estimating the performance and the robustness using the Probabilistic Collocation Method with Polynomial Chaos Expansion in a simulation-based approach to robust concept exploration can increase the method's accuracy and reliability over the original RCEM.

Hypothesis 2 is validated with the development of the simulation-based RCEM and the introduction of the Probabilistic Collocation Method to robust concept exploration. The new method's theoretical structural validity, empirical structural validity and empirical performance validity are demonstrated with an artificial design problem in Chapter 5 and the design of a LCA heat exchanger in Chapter 6. The validation of the first three quadrants of the validation square is summarized here. Afterward, the theoretical performance validity is discussed and Hypothesis 2 is validated.

The theoretical structural validity of the simulation-based RCEM is demonstrated in Section 5.2.1 by showing the internal consistency of the individual components of the method as well as the consistency of their assembly.

The empirical structural validity of the new method is demonstrated by arguing that the examples used for validation are representative for engineering design problems the method is designed for and might be applied to. In Section 5.2.2, the specific characteristics of the artificial design problem that qualify it as an appropriate example, are pointed out. The same is done in Section 6.2.2 for the LCA heat exchanger example.

The empirical performance validity of the simulation-based RCEM is demonstrated with the data from the two example problems. In Section 5.4, it is shown that the generated results from the first example are as expected and that the improvements are directly linked to the application of the new method. This is confirmed with the data from the heat exchanger example in Section 6.4. The solutions from this example are also as expected and support the validity of the simulation-based RCEM.

As discussed in Section 1.4, theoretical performance validation involves establishing confidence that the method is applicable and useful for problems beyond the examples.

This step is the logical consequence from validating the first three quadrants of the validation square. For the simulation-based RCEM, it is demonstrated that the method's mathematics and its internal information flow is consistent, that the used example problems are appropriate, that the results are consistent, valid and as expected and that the observed effectiveness and efficiency is linked to the application of the new method. Based on this, its usefulness for general classes of problems is inferred.

In Hypothesis 2, it is hypothesized that by introducing the Probabilistic Collocation Method to robust concept exploration the accuracy and reliability can be improved when compared to the original RCEM, which uses a global response surface. In Section 3.1, it is shown that a global response surface, as used in the original RCEM, would lead to wrong results for the artificial design example. In Section 5.3, it is demonstrated that with the simulation-based RCEM the designer is able to solve this exact problem and finds a satisfying robust solution. With this, it is shown that the new method is more reliable for non-linear problems. Furthermore, it is shown in Section 5.3 and in Section 6.3 that the robustness estimations of the new method are very accurate and especially for the multi-modular problem in Chapter 5 significantly more accurate than the original RCEM. From this, the validity of Research Hypothesis 2 can be inferred. In addition to the improvements mentioned in Hypothesis 2, the simulation-based RCEM also increases the efficiency and flexibility of the RCEM. Since for the application of the simulation-based RCEM no computationally expensive global response surface has to be built, the new method is very flexible and efficient when changes in the model or simulation are made. In the original RCEM, such a change would require a new response surface, while the simulation-based RCEM can be simply applied to the new model without further changes. The integration of the Probabilistic Collocation Method, furthermore, opens the possibility to account for alternative input distributions. While the original RCEM is basically limited to the assumption of Gaussian or Uniform distributions, the PCM can account for any probability density function of the design variables and noise factors.

In the next section, the achievements and contribution to the field of engineering design, which are established by answering the research questions and validating research hypotheses, are summarized.

7.2 Achievements and Contributions

The major contributions of the presented research are the critical analysis of the original RCEM, the development and the validation of the simulation-based RCEM as well as the development of a computational framework for a user-friendly application of all RCEM versions.

- A critical analysis of the original RCEM is presented in this thesis. Two major limitations, which can lead to fatal errors, are explained and demonstrated on an example. It is shown that the first-order Taylor Series approximation for the estimation of a system's robustness is critical when used in an iterative exploration algorithm like in RCEM. Although, this has been realized by researchers for other applications before, it was not demonstrated for RCEM yet. In fact, several students already worked on developing RCEM without noticing this gap. Furthermore, it is shown that a global response surface, as used in the original RCEM, can lead to wrong results. This problem has been also addressed by other researchers before but has not been demonstrated for the application of RCEM.

- With the development of the modified RCEM, several alternatives to the criticized Taylor Series are demonstrated and assessed. The Multiple Point Method, which is introduced to the modified RCEM, is a modification of the Sensitivity Index. It is augmented for the specific estimation of the performance variance given a Gaussian distributed design variable or noise factor. The research on this topic was the basis for developing the simulation-based RCEM.
- It is presented that the methods for the variance estimation both in the original RCEM as well as in the modified RCEM have little computational advantage over a brute force method like Monte Carlo or Latin hypercube when an explicit function or an exact surrogate model is available. With today's available computational power, those brute force methods are very fast when applied to explicit functions, which questions the need for estimations in such cases.
- With the simulation-based RCEM, a design method is developed that fills both gaps found in the original RCEM. The introduction of the Probabilistic Collocation Method to robust concept exploration opens up entirely new possibilities. Besides increasing the accuracy and reliability of RCEM, it allows for considering any observed or assumed probability density functions of design variables and noise factors.
- With the development of the JavaDSIDES XML Generator and the integration of MATLAB simulations in JavaDSIDES, a computational framework for the application of all RCEM versions is provided. These advancements make

JavaDSIDES, which is written by Matthew Marston, more user-friendly and easier to apply to simulation-based engineering design problems.

The list of contributions in this section emphasizes solely the achievements established in the presented research. The aspects that could not be addressed or solved and the new questions and problems that were discovered are discussed in the following section.

7.3 Limitations and Opportunities for Future Research

In this section, the work presented in this thesis is critically evaluated and recommendations and ideas on further improvement are provided.

The first concern is the practical applicability of the presented computational framework for the simulation-based RCEM. JavaDSIDES is a powerful tool that has many levers to adjust and customize the algorithm. Those levers include, for example, the constraint violation, the constraint adaption or the convergence tolerance. For someone, who is not an expert in operations research, it is difficult to find the appropriate setting for a specific problem.

- Since an engineering designer is not necessarily also an expert in statistics and operations research, a more detailed documentation of the JavaDSIDES application and the DSIDES algorithm would be very helpful for industrial application of the simulation-based RCEM as well as for further research in robust concept exploration.

Another concern is the practical implementation of the modified RCEM. There are two open questions. First, one might argue that no practical validation of the method is performed in this thesis. Although the method is extensively verified with two examples, they might not represent the level of complexity an engineering designer might face in industry.

- Although the verification of the modified RCEM is throughout positive in this thesis, which gives reason to argue for the theoretical performance validity of the method, it should be tested on a larger case study. This data could then be used to support the finding in this thesis.

The second open question about the modified RCEM is its utility when compared to brute force methods like Monte Carlo. Like in the original RCEM, also in the new method the robustness of an engineering system is estimated from an explicit function, e.g., a performance function or a surrogate model. Today's available computational power allows for thousands of calculations is just split-seconds. This decreases the advantage of the developed estimation methods in computational time and might not justify the decrease in accuracy over the brute force methods.

- It might be worth investigating in which cases estimation is advantageous and for which scenarios brute force methods are the better choice. Here, the computational efficiency as well as the methods' accuracy, especially in non-linear, multi-modular design problems are the crucial factors that have to be evaluated.

Further concerns exist regarding practical applicability of the simulation-based RCEM. This method is extensively validated with two examples in this thesis. Although the results clearly support the validity of the method, one might argue that the examples' levels of complexity are not comparable to those of systems design problems in industry.

- To support the validity of the simulation-based RCEM, the method should be applied to a larger industry case study. This data could then be used to support the finding in this thesis.

The implementation of the Probabilistic Collocation Method within the simulation-based RCEM is very promising. The data generated with the examples strongly support the validity of the method. The method also opens up the possibility to take alternative design variable distributions into account. Any distribution that can be formulated with a probability density function can be considered in the PCM. The local response model generated with the PCM also allows for detailed investigations of the performance sensitivity and probability distribution. However, the method has several limitations. Due to the underlying mathematics, the method is very accurate for estimations in the region around the performance mean (high probability region). This is because the collocation points, which are used to generate the local response model, are located mainly around the mean. For this reason, the method can become inaccurate if the variance of a performance is largely influenced by the low probability regions in the tail ends of a distribution.

- In future research on the simulation-based RCEM, the PCM should be investigated in further detail. One could explore if there are alternative methods for selecting the collocation points, which lead to higher accuracy of the method in predicting the tail ends of the performance distribution. This investigation is especially necessary for multi-dimensional problems because here more collocation points are generated in the PCM then for the calculation of the local response model needed. Currently, the points are selected according to their joint probability. This supports the observation that the method is especially accurate for the high probability region. It could be studied if, for example, orthogonal arrays or d-optimal designs could be used for the selection of the collocation points and if this would increase the accuracy of the method. Since especially researchers in the field of reliability-based design are dealing with the problem of the tail end estimation, there might be some synergies to find in this area. It might even be possible to develop a combined reliability-based RCEM.

In the last concern that is to be mentioned in this thesis, the compromise DSP is addressed, which is the basis for all Robust Concept Exploration Methods. The cDSP is a very clear and almost self-explanatory framework, which designers can use for the formulation of their design problems. Its structure guides the designers through the collection of all important data for solving the problem. However, there are some aspects the designer has to be aware of when using the cDSP, especially when formulating the goals and assigning their target values and their weights. Usually, the designer would think that the weight of an objective is only influenced by the weighting factor in the objective or deviation function. This is not true for the cDSP. Here, the weight is also influenced by the selection of the target values for the design goals. In the deviation function, the goals are normalized with their target value. This means that the further away a target is from the current point, the smaller is the influence of a change of that goal on the deviation function, thus resulting in a smaller weight.

- If designers are aware of this phenomenon when using the cDSP they can minimize its interfering effect by thoroughly assigning the targets as close to the feasible area as possible and only using the weighting factors to express their preferences. However, there might be the opportunity for further research, too. The ideas of value-based and utility-based cDSPs have been investigated in literature before. Assessing these methods with respect to the above motioned phenomenon might help to further develop the cDSP framework and thus further improve the Robust Concept Exploration Methods presented in this thesis.

APPENDIX A

JAVADSIDES MANUAL

A.1 Acknowledgements

The optimization algorithm used in DSIDES and JavaDSIDES is called “Adaptive Linear Programming” and is developed by Farrokh Mistree, Owen F. Huges and Bert Brass (Mistree, et al., 1993). DSIDES (Decision Support In the Design of Engineering Systems) is designed to solve Decision Support Problems (DSPs) and was originally programmed in FORTRAN. Matthew Marston developed the Java version of DSIDES (JavaDSIDES), which is used in this manual. In contrast to the original DSIDES, JavaDSIDES can only solve compromise Decision Support Problems (cDSP) whereas the original DSIDES also has the functionality to solve selection DSPs. Matthew Marston’s help in developing this manual is greatly acknowledged.

A.2 System Information and Requirements

System and programs installed:

- Operating System: Windows XP
- JavaDSIDES
- MATLAB 2008b including the following components:
    - MATLAB Compiler
    - MATLAB Builder JA

- NetBeans IDE 6.5.1 including the Maven Plug-in (Integrated (Java) Development Environment, http://www.netbeans.org/)
- Java JDK 1.6.0_13 (Java Development Kit, http://java.sun.com/javase/downloads/index.jsp)
- Java JRE 1.6.0_07 (Java Runtime Environment, http://java.sun.com/javase/downloads/index.jsp)
- Apache Maven 2.1.0 and 2.0.10 (http://maven.apache.org/)

A.3 Overview

So far the manual is written for computers that have MATLAB installed and for users who have a basic knowledge of Java.

This manual describes how to solve design problems, formulated as compromise Decision Support Problems, using the Java version of DSIDES.

The Java version of DSIDES was originally designed to handle explicit functions as goal and constraint expressions. In this manual, it is described how to extend the functionality for using MATLAB functions as goal or constraint formulations.

There are two ways of running JavaDSIDES:

- Using a XML file containing the cDSP information as input
- Using a model file written in Java as input (the model Java file (‘model’.java) contains the cDSP information)

In case an XML file is used as input, JavaDSIDES generates the required model Java file automatically. In Figure A.1 the steps of this current approach of JavaDSIDES are illustrated. After the designer has formulated the design problem in the form of a compromise Decision Support Problem it has to be translated into a XML document using the instructions given in Section A.5. JavaDSIDES is then executed using the XML file as input. The DSIDES results are displayed on the screen and saved in output files.

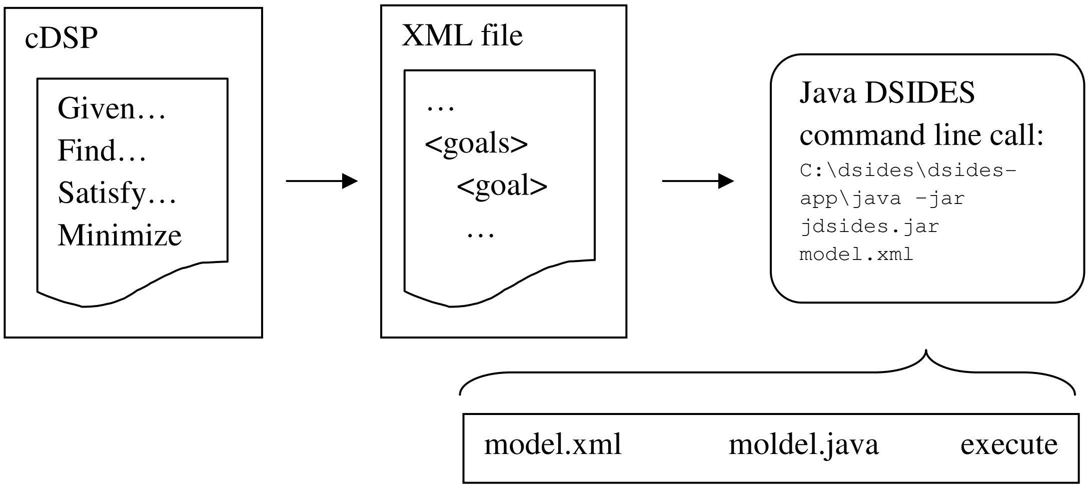


Figure A.1: Current JavaDSIDES Approach

Since many designers are not familiar with the XML syntax, and the XML formulation can become confusing for large design problems, a MATLAB function was developed that can generate the required XML model files automatically. It provides a simple interface between the designer and JavaDSIDES. In case of explicit function formulations, the MATLAB cDSP XML Generator function can call JavaDSIDES directly and display the results (see Section A.7).

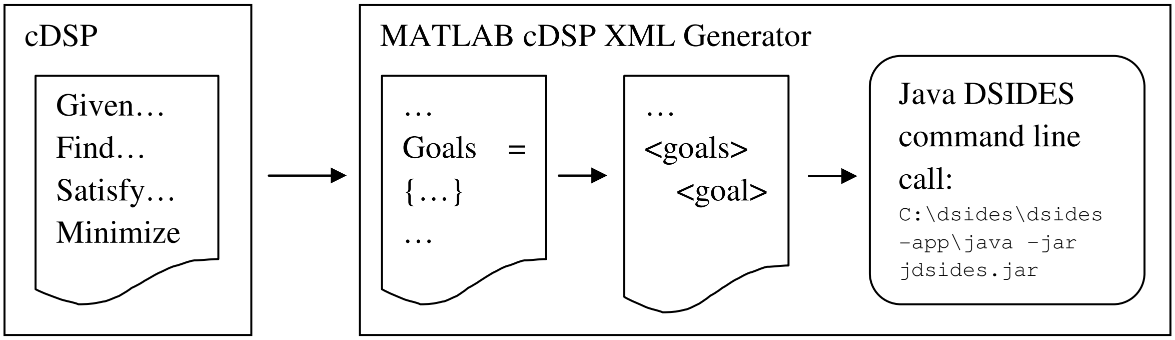


Figure A.2: MATLAB cDSP XML Generator

If MATLAB functions are used for calculating goals or constraints, some additional steps are required. Since JavaDSIDES cannot call external MATLAB functions directly, these have to be wrapped in Java. MATLAB offers the packages “MATLAB Compiler” and “MATLAB Builder JA” that have exactly this functionality. In Section A.8, their application to wrap MATLAB functions is described. The Java packages created with those tools can then be imported into the Java model file and the functions can be called (see Figure A.18). The easiest way to get the model Java file is to use the automatic generation of the model file integrated in JavaDSIDES using the MATLAB cDSP XML Generator (see Figure A.3). The Java model file has to be edited and compiled so that he wrapped MATLAB functions can be called (see Section A.9). Finally, JavaDSIDES can then be executed with the new model (See Section A.10).

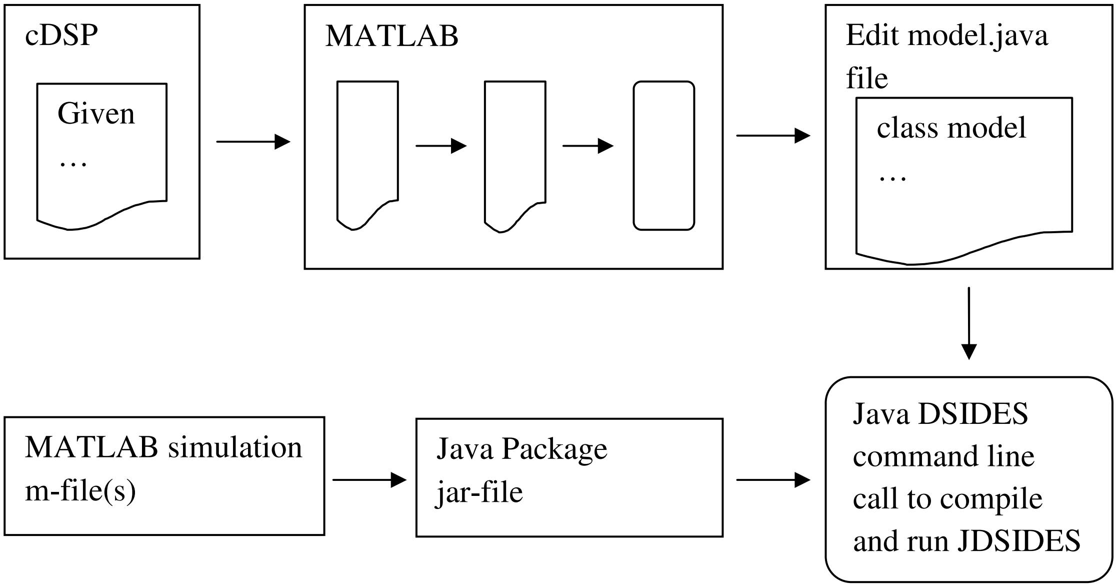

cDSP
Given
…
MATLAB
Edit model.java file
class model
…
MATLAB simulation m-file(s)
Java Package jar-file
Java DSIDES command line call to compile and run JDSIDES

Figure A.3: Integrated MATLAB JavaDSIDES Approach

In Section A.11 some possibilities for advanced editing of the Java model file are demonstrated, e.g., the implementation of a function counter.

The entire process for the case of explicit functions (Section A.12) and for the case of a "black box" MATLAB model (Section A.13) is demonstrated with a simple cantilever beam design problem.

A.4 Installing JDSIDES

The required files to run JavaDSIDES are contained in the ZIP archive **`dsides.zip`**. The content of this archive should be extracted to **`C:\dsides\`**. The file jdsides.jar should now have the following path: **`C:\dsides\dsides-app\jdsides.jar`**.

A.5 Setting up the cDSP for Implementation in JavaDSIDES

Due to the way JavaDSIDES handles and normalizes goals and constraints the following conventions are necessary for the formulations of goals and constraints.

Goal formulations:

JavaDSIDES is written to handle all goals as maximization goals, which requires the following changes:

1. If a goal is to be maximized with a certain target, no changes have to be made.

   A_i being the goal formulation and G_i the target value, JavaDSIDES uses the following normalization:

   $A_i(X)/G_i + d_i^- - d_i^+ = 1$ with d_i^- to be minimized and $d_i^+ = 0$ (*)

2. If a goal is to be minimized with a certain target, the goal formulation as well as the target have to be inverted, which makes it a maximization problem.

   Goal formulation $A_i^*(X) = 1/A_i(X)$ with goal $G_i^* = 1/G_i$ leads to:

   $$G_i / A_i(X) + d_i^- - d_i^+ = 1$$
   $$\Leftrightarrow [1/A_i(X)]/[1/G_i] + d_i^- - d_i^+ = 1$$
   $$\Leftrightarrow A*_i\,(X)/G*_i + d_i^- - d_i^+ = 1$$

   Which is now equivalent to (*) with d_i^- to be minimized and $d_i^+ = 0$

3. If the goal target of a minimization is zero, the transformation of the goal formulation $A_i(X)$ is a bit tricky. In the literature (Mistree, et al., 1993) the formulation for this case is:

   $A_i(X)/A_{i\max} + d_i^- - d_i^+ = 0$ (**)

with $A_{i\max}$ being the estimated maximum of $A_i(X)$. JavaDSIDES instead uses the formulation $A_i^*(X)/G_i^* + d_i^- - d_i^+ = 1$. (***)

With $A_i^*(X) = A_i(X) + A_{i\max}$ and $G_i^* = A_{i\max}$ (***) becomes equivalent to (**) where d_i^+ is to be minimized and $d_i^- = 0$.

Constraints:

JavaDSIDES normalizes the constraints by dividing the expression by the limit. This is why the limit is not permitted to be zero. If the technical limit is actually zero, the problem can be solved by adding a number of the same order of magnitude as the expression to both sides of the equation or inequality.

If for example the original constraint is $4 \cdot x - y \geq 0$ with $x = [10...50]$ and $y = [50...100]$, the constraint could be transformed to $4 \cdot x - y + 50 \geq 50$.

A.6 The Structure of the cDSP XML

As an interface between the designer and the JavaDSIDES an XML file is used. Creating the XML structure with the specific cDSP formulation of the design problem is easier than coding the required Java file of the model.

In Figure A.4, the structure of an XML input file for JavaDSIDES is shown with arbitrary values and formulations.

```xml
<?xml version="1.0" encoding="utf-8"?>
<Model xmlns="http://dsides.configuration/model"
autoNormalizeConstraintsAndGoals="true"
decisionVariableConvergenceTolerance="0.00000001"
deviationFunctionConvergenceTolerance="0.00000001" writeCSVFile="true">
   <Name>Name of Design Problem</Name>

   <DecisionVariables>
      <DecisionVariable>
         <Name>x</Name>
         <Minimum>10.0</Minimum>
         <Maximum>76.0</Maximum>
         <InitialValue>20.0</InitialValue>
      </DecisionVariable>
      (…further decision variables if needed…)
   </DecisionVariables>
   <Constants>
      <Constant>
         <Name>Constant1</Name>
         <Value>10</Value>
      </Constant>
      (…further constants if needed…)
   </Constants>
   <Goals>
      <IntermediateExpression>  (…if needed)
         <Name>int1</Name>
         <Expression>1.0+2.0+3.0+Constant1+x</Expression>
      </IntermediateExpression>
      (…further intermediate expressions if needed…)
      <Goal level="1" negativeDeviationCoefficient="0.1"
positiveDeviationCoefficient="0.0">
         <Name>goal1</Name>
         <Expression>1.0/(x-7.0)^2</Expression>
         <Target>1</Target>
      </Goal>
      (…further goals if needed…)
   </Goals>
   <Constraints>
      (…intermediate expressions if needed…)
      <Constraint type="LessThanOrEqual">
         <Name>Constraint1</Name>
         <Expression>Constant1 + x</Expression>
         <Limit>100</Limit>
      </Constraint>
   </Constraints>
</Model>
```

Figure A.4: XML Structure for JavaDSIDES Inputs

As it can be seen, the deviation variables d_i^- and d_i^+ do not have to be introduced neither in the goal formulations nor as design variables with boundaries; this is done automatically by JavaDSIDES. The corresponding weights for the deviation variables can be set as **`negativeDeviationCoefficient`** and **`positiveDeviationCoefficient`** for d_i^- and d_i^+ respectively for each goal. Furthermore the level can be set if a hierarchical optimization is desired. For the constraints the following types are allowed:

- "LessThenOrEqual"
- "GreaterThenOrEqual"
- "Equal"

It is easy to see that if the design problem is complex, the XML structure can become long and confusing. This is the reason why a MATLAB interface has been developed that provides a clear input interface for the designer.

A.7 The MATLAB cDSP XML Generator

The Matlab cDSP XML Generator is a MATLAB code that can serve as an interface between the designer and the JavaDSIDES. Since the direct generation of a cDSP XML file can be confusing for large design problems, this MATLAB code is intended to ease this step. It consists of two sections; the first requires the designer's inputs and the second will generate the XML file, save it and execute the JavaDSIDES with the generated file.

The first section of the MATLAB cDSP XML Generator code is shown in Figure A.5.

```
clear all;
clc;
% INPUT
    projectname = 'Template';
% Design Variable Information
    dvariables = {'Name',   'minimum',  'maximum',  'initial value';
                  'x',      '10.0',     '76.0',     '20.0';
                  'y',      '10.0',     '76.0',     '20.0';
                 };
% Constants
    constants = {'Name',    'value';
                 'constant1',   '10';
                };
% Intermediate Expressions
    % used in goal expressions:
    goalintermediate = {'Name',     'Expression';
                        'int1',    '1.0+2.0+3.0+constant1+x';
                       };
    % used in constraints expressions:
    constraintintermediate = {'Name',     'Expression';
                             };
% Goals (need to have at least one)
    % Equations
    goals = {'Name',  'NegD', 'PosD', 'level', 'Target', 'Expression';
             'goal1', '0.1',  '0.0',  '1',     '1',      '1.0/(x-
7.0)^2';
            };
% Constraints (need to have at least one)
    constraints = {'Name',        'type',
'Limit','Expression';
                   'Constraint1', 'LessThanOrEqual', '100,   'constant1
+ x';
                  };
% Set XML Model variables (Convergence Tolerance)
docNode = com.mathworks.xml.XMLUtils.createDocument('Model');
docRootNode = docNode.getDocumentElement;

docRootNode.setAttribute('xmlns', 'http://dsides.configuration/model');
docRootNode.setAttribute('decisionVariableConvergenceTolerance',
'0.00000001');
docRootNode.setAttribute('deviationFunctionConvergenceTolerance',
'0.00000001');
docRootNode.setAttribute('writeCSVFile', 'true');
docRootNode.setAttribute('autoNormalizeConstraintsAndGoals', 'true');

% ------------------------------------
% NO EDITING AFTER THIS POINT REQUIRED
% ------------------------------------
```

Figure A.5: MATLAB cDSP XML Generator Code

Also in the MATLAB cDSP XML Generator, the deviation variables d_i^- and d_i^+ do not have to be introduced neither in the goal formulations nor as design variables with boundaries; this is done automatically by JavaDSIDES. The corresponding weights for the deviation variables can be set as **`NegD`** and **`PosD`** for d_i^- and d_i^+ respectively for each goal. Furthermore, the level can be set if a hierarchical optimization is desired. For the constraints the following types are allowed:

- "LessThenOrEqual"
- "GreaterThenOrEqual"
- "Equal"

The MATLAB cDSP XML generator is pretty much self-explanatory. The input is handled with arrays that take string inputs and can be extended to the number of rows required to represent the cDSP of the specific design problem.

If the design problem contains only explicit functions, the DSIDES results are shown in the MATLAB command window and are saved under **`C:\dsides\dsides-app\`**. If the problem requires MATLAB functions to be called, they have to be exported as Java packages and the Java model file has to be edited accordingly.

In this case the following steps should be performed:

1. cDSP information is entered in the MATLAB cDSP XML generator
2. For the goals and constraints that require the call of a MATLAB function a random number is entered as an expression. This number will later be changed to the specific function call in the Java model file.

3. MATLAB cDSP XML Generator is executed. JavaDSIDES should run and display results. These results are ignored because they do not yet contain the MATLAB function calls.
4. The required MATLAB functions are wrapped in Java using MATLAB Compiler and MATLAB Builder JA (see Section A.8).
5. Java model file is edited.

A.8 Wrapping and Packaging MATLAB into Java

In this section, it is explained how MATLAB functions can be wrapped and packaged as Java functions. The result is a .jar file that contains the Java classes that are required to call the function(s) in a Java program. This allows the designer to use these functions in JavaDSIDES as goal or constraint formulations.

MATLAB has an integrated Compiler (MATLAB Compiler) and Java Builder tool (MATLAB Builder JA). Those are optional packages, which have to be selected when installing MATLAB.

A summary of steps to deploy a MATLAB function (detailed description see MATLAB Builder JA manual) is given in the following:

1. Open MATLAB and browse to the folder containing the MATLAB function to be wrapped.
2. Type **`deploytool`** in the MATLAB command window
3. Click “New Project” in the Deploytool Graphic User Interface (GUI)
   - Select MATLAB Builder JA – Java Package
   - Type the name of the project and browse to the location for the project folder (e.g. project name: “exproject”)

- Project name will become name of the *.jar package (e.g. “exproject.jar”)

4. Rename class folder

   - Folder name will become class name *.class (e.g. folder name “exfolder” -> “exfolder.class”)

5. Drag and drop all needed MATLAB *.m files in the class folder in the deploytool GUI

   - *.m file names will become the Java function names (e.g. exfunction.m will be function “exfunction” of the class “exfolder”)

6. Click “Build the project”.

Now the project folder will contain the required .jar file as well as additional documentation, which is not necessarily needed for JavaDSIDES if the designer remembers the names of the used functions.

For running the generated Java code on computers that have MATLAB installed, no further steps are required to implement the new class into a DSIDES model.

Please refer to the MATLAB Builder JA Manual if packaging is required.

A.9 Calling the MATLAB Function in a JavaDSIDES Model File

In this section, it is described how the JavaDSIDES model file has to be augmented in order to call a wrapped MATLAB function. It is assumed that JavaDSIDES was executed before with the corresponding XML file or the MATLAB cDSP XML Generator was invoked.

The model file (*.java) is located in the folder C:\dsides\dsides-app\model\model\ and can be edited in any Java or text editor.

Importing packages

First the following Java packages need to be imported in the .java file (class) that calls the wrapped MATLAB function:

```
import com.mathworks.toolbox.javabuilder.*;  /* MATLAB Builder Package
*/
import exproject.*;                          /* (projectname.jar) */
```

Adding functions that return the MATLAB function values

For every wrapped MATLAB function that has to be called, a java function, which provides the desired values, has to be added to the model class. The easiest way is to add these functions as part of the model class right after the declaration of the class that usually looks like:

```
public class ModelExample
  implements dsides.compiler.ModelImplementation
```

How such a function call can look like is shown in the following example:

```
public double customFunction (String[] args)
   {
      MWNumericArray n = null;   /* Stores input value */
      Object[] result = null;    /* Stores the result */
      exfolder exObject = null;     /* Stores example class instance */

      try
      {
         /* Convert and print input value*/
         n = new
MWNumericArray(Double.valueOf(args[0]),MWClassID.DOUBLE);
         /* Create new example object */
         exObject = new exfolder();
         /* Compute result */
         result = exObject.exfunction(1, n);
         MWNumericArray resultA = (MWNumericArray) result[0];
         /* Return result */
         return resultA.getDouble(1);
      }
      catch (Exception e)
      {
         System.out.println("Exception: " + e.toString());
      }

      finally
      {
         /* Free native resources */
         MWArray.disposeArray(n);
         MWArray.disposeArray(result);
         if (exObject != null)
            exObject.dispose();
      }
      return null;
   }
```

In the following the components of the code are described in detail.

In the declaration of the function, it can be seen that this example accepts an array of strings as input and gives a double as output. Of course, this can be changed so that the function uses for example a double value or array as input. In this case, the string to double conversion is obviously not needed.

Because the MATLAB function throws exceptions, which have to be caught, the structure of the function always needs to have the following components:

```
      try
      {
      …
            /* Function call */
      …
      }
      catch (Exception e)
      {
         System.out.println("Exception: " + e.toString());
      }

      finally
      {
            /* Free native resources */
      }
```

The MATLAB Java function accepts MWArrays as inputs. If the input data is not converted to such an array before calling the function, the function will convert the data automatically.

This means that both of the following expressions will have the same result:

1.

```
n = new MWNumericArray(Double.valueOf(args[0]),MWClassID.DOUBLE);
result = exObject.exfunction(1, n);
```

2.

```
double n = Double.valueOf(args[0]);
result = exObject.exfunction(1, n);
```

In both cases the first variable determines the number of outputs. If arrays are used when MATLAB functions have multiple outputs, it is always “1”.

Multiple inputs

Multiple inputs can either be passed as used in MATLAB:

```
result = exObject.exfunction(1, n, m);
```

Or they can also be passed as an object array (Object[]):

```
Object[] input = {n, m};
result = exObject.exfunction(1, input);
```

Handling output

First the result needs to be casted into an MWNumericArray:

```
MWNumericArray resultA = (MWNumericArray) result[0];
```

Then there are two ways to access the values in the new array:

- Accessing the values individually:

```
double a = resultA.getDouble(int);
```

Here `int` is the integer describing the position of the variable in the array ("1" for the first value).

- Converting the results into a double array:

```
double[][] a = (double[][])resultA.toDoubleArray();
return = a[0][int];
```

Here `int` is the integer describing the position of the variable in the array ("0" for the first value).

Augmenting the goal or constraint expressions

Now the added functions can be called in the "calculate" functions of the GoalProvider or ConstraintProvider class (further down in the Java model file). For example:

```
double d = customFunction(x[0]);
f[0] = (1.0 / d);
```

Used in the GoalProvider, this code fragment would pass the first design variable (x[0]) to the function that provides the MATLAB function results and would use its return in the formulation of the first design objective (f[0]).

A.10 Compiling and Executing JavaDSIDES Models

General

In this subsection, it is described how, in general, classes are compiled and executed that contain calls of wrapped MATLAB function. For the specific case of a JavaDSIDES model class please see the next subsection.

To execute the wrapped MATLAB function the following components are required:

1. exproject.jar (generated .jar file containing the MATLAB code)
2. javabuilder.jar (package containing classes needed to execute the wrapped MATLAB code. This file can be found and copied in the following MATLAB folder: **MATLAB_ROOT\toolbox\javabuilder\jar\**
3. the .java file that calls the wrapped MATLAB function (see 1.7)

The description assumes that the three files are in the same folder (preferably **C:\dsides\dsides-app\**) and the following system variables contain the presented information:

JAVA_HOME: the JDK directory (e.g. **C:\Java\jdk1.6.0_13**)

PATH: the JDK \bin directory (e.g. **C:\Java\jdk1.6.0_13\bin**)

1. Browse to the folder containing the three files in the Windows command prompt.

   Compiling the .java file to a Java class file:

```
javac –classpath javabuilder.jar;exproject.jar filename.java
```

2. Executing Java class main function:

```
java –classpath .\;javabuilder.jar;exproject.jar filename input
```

For details on these steps please refer to the MATLAB Builder JA manual.

JavaDSIDES model file

As mentioned in the previous section, the project.jar file and the javabuilder.jar file are required to compile and run the augmented model file. In the description it is assumed that both of them are copied into the dsides-app folder (**`C:\dsides\dsides-app\`**).

1. Browse to **`C:\dsides\dsides-app\`** in the Windows command prompt.
2. Compile the model ('model name'.java) file to a Java class file with the following command:

```
javac -g -classpath jdsides.jar;javabuilder.jar;exproject.jar -d model
model\model\modelname.java
```

   where **exproject.jar** contains the wrapped MATLAB functions and **modelname.java** is the model file edited before.

3. With the same name conventions, JavaDSIDES can now be executed using the following command:

```
java -classpath jdsides.jar;javabuilder.jar;exproject.jar;model
model.modelname
```

Every time the model Java file is edited, it has to be compiled again before the changes take effect. Furthermore, it is important to know that if the MATLAB cDSP XML generator is executed with the same project name, the model Java file is overwritten. This is why it might be useful to rename the edited class and the corresponding filename.

A.11 Advanced Editing

Adding a function counter

For some applications, the designer might be interested in the number of calls of a certain goal or constraint function. In these cases, a function counter can be added to the model Java class. To implement a function call counter first, a variable that holds the count, has to be initialized in the class containing the model ('modelname'.java).

e.g.

```
private static int functionCount;
```

This variable is now increased by one, each time the certain function is called. Therefore

```
functionCount++;
```

is added into the function right before the `return` command.

Now the function call counter is set up. However, in order to display the result at the end of the DSIDES run, two further modifications have to be introduced to the model class. First, a PrintStream has to be instantiated within the model class:

```
private PrintStream out;
```

For this command to work, the Java.IO package has to be imported to the model file. Therefore, the following command has to be added to the "import section" in the header of the file:

```
import java.io.*
```

Second, the **main** function of the model class has to be edited in order to give the function count to the PrintStream after the DSIDES run. This will allow printing the final count to the result output (screen and output file).

The original **main** function should look like the following:

```
public static void main (String[] args)
    throws java.io.IOException
 {
      ModelName impl = new ModelName();
      JDSIDES dsides = new JDSIDES(impl.getAlpData ());
      dsides.run ();
  }
```

where ModelName represents the specific name of the model.

This function has to be changed to the following:

```
public static void main (String[] args)
    throws java.io.IOException
  {
      ModelName impl = new ModelName ();
      AlpData data = impl.getAlpData ();
      impl.out = data.getOut();
      JDSIDES dsides = new JDSIDES(data);
      dsides.run ();
      impl.out.println("..." + functionCount + "...");
  }
```

The additional commands access the output stream of the model implementation and add the function count with a customizable line.

If it is necessary to count the function calls for a constraint and a goal function, the same concept can be used to implement additional counters.

Improving performance

Previously, it is described how to add a custom function that calls a wrapped MATLAB function and returns its results. To keep it simple, the instance of the wrapped MATLAB class was instantiated within the function:

```
exfolder exObject = null;     /* Stores example class instance */
```

This costs time and resources each time the function is called. This time can be saved if the object is instantiated outside the function, directly in the model class with:

```
private static exfolder exObject;
```

Repackaging jdsides.jar after changes in the main program

Using for example NetBeans IDE and the Maven plug-in, the dsides-app Java project can be imported and edited. When imported for the first time, NetBeans will discover several bugs and missing libraries. Building the project will solve this problem. The required libraries are downloaded from the online server and the missing files are created. After all desired changes are made, the project has to be built again, which will generate a .jar file in the **target** folder under **C:\dsides\dsides-app**. This **jdsides-1.0.0.Alpha1.jar** file does not yet contain the entire program.

To package the whole JavaDSIDES in one single .jar file, browse to the **C:\dsides\dsides-app** subdirectory in the Windows command prompt and invoke the command "**mvn assembly:assembly**". This will build a new distribution jar file in the **C:\dsides\dsides-app\target** subdirectory. The file will be called **jdsides-1.0.0.Alpha1-jar-with-dependencies.jar**. In order to replace the old version of JavaDSIDES, this file is copied to the **C:\dsides\dsides-app** folder and renamed to **`jdsides.jar`**. Apache Maven has to be installed on the computer for the packaging process. How to download and install the latest version of Maven can be found online.

A.12 Cantilever Beam Example with Explicit Functions

In this section, a simple cantilever beam design problem is used to demonstrate the functionality of JavaDSIDES, the MATLAB cDSP XML Generator and the MATLAB Builder JA.

The goal in the example is to minimize the weight as well as the maximum deflection of a cantilever beam with given load (P) and length (L). Also given are the Young's Modulus (E) and the density of the beam material aluminum alloy (ρ). The design variables are the breadth (b) and the height (h) of the beam. To keep the example simple, no uncertainty of the parameters or design variables is considered.

The example will be shown for two cases:

Case 1: The formulas (explicit functions) for the goals and constraints are known.

Case 2: The functions are unknown and are treated as a black box MATLAB model.

Using explicit functions

In this case the explicit functions for the goals and constraints are known, which means that no MATLAB-Java conversion is needed. The cDSP of the design problem, which is given in Figure A.6, can be directly translated into the MATLAB cDSP XML Generator, which will also run JavaDSIDES and display the results:

Step 1: Formulation of the general cDSP for the Design Problem.

Given:

- Constants:
  - Applied load: P = 4kN
  - Young's Modulus: E = 7.17x10^7 kN
  - Beam length: L = 5m
  - Aluminum alloy density: ρ = 2768 kg/m^3
- Equations:
  - Volume of beam: $V = L \cdot b \cdot h$
  - Mass: $m = \rho \cdot V$
  - Area moment of inertia: $I = \frac{bh^3}{12}$
  - Maximum tip displacement: $\delta = \frac{PL^3}{3EI}$

Find

- Beam breadth: b = [0.1 m, 1.5m]
- Beam Height: h = [0.1 m, 1.5m]
- d_i^-, d_i^+ = [0,1]

Satisfy

- **Bounds:**
  - See bounds on design variables
- **Goals:**
  - Minimize mass with goal 500 kg: $\frac{500\text{kg}}{\text{m(b, h)}} + d_1^- - d_1^+ = 1$
  - Minimize displacement with goal 0.02 m: $\frac{0.2\text{ m}}{\delta\text{(b, h)}} + d_2^- - d_2^+ = 1$
- **Constraints:**
  - $m \geq 500kg$
  - $m \leq 1000kg$
  - $\delta \geq 0.02m$
  - $\delta \leq 0.1667m$

Minimize

$Z = 0.4 \cdot d_1^- + 0.6 \cdot d_2^-$

Figure A.6: cDSP for Cantilever Beam Example with Explicit Functions

Step 2: Transformation of the goal and constraint formulation according to the conventions.

The goal functions have to be transformed to maximization problems:

Mass:

- minimize m(b,h)   maximize 1/m(b,h)
- min goal 500kg   max goal 1/500kg

Displacement:

- minimize δ(b,h)   maximize 1/ δ(b,h)
- min goal 0.02m   max goal 1/0.02m

Step 3: Translation of the cDSP into the MATLAB cDSP XML Generator

Using the MATLAB cDSP XML Generator, it is easy to solve this design problem. As the initial solution for the optimization algorithm {b, h} = {0.1m, 0.1m} is used.

```
% INPUT

projectname = 'CantBeamExplicit';

% Design Variable Information
    dvariables = {'Name',    'minimum',   'maximum',   'initial value';
                  'b',       '0.1',       '1.5',       '0.1';
                  'h',       '0.1',       '1.5',       '0.1';
                 };
% Constants
    constants = {'Name',     'value';
                 'P',        '4.0';
                 'E',        '7.17E7';
                 'L',        '5.0';
                 'rho',      '2768.0'
                };
% Intermediate Expressions
    % used in goal expressions:
    goalintermediate = {'Name',      'Expression';
                        'V',         'L*b*h';
                        'm',         'rho*V';
                        'I',        '(b*h^3)/12';
                        'dis',      '(P*L^3)/(3*E*I)';
                       };
    % used in constraints expressions:
    constraintintermediate = {  'Name',      'Expression';
                                'V',         'L*b*h';
                                'm',         'rho*V';
                                'I',         '(b*h^3)/12';
                                'dis',       '(P*L^3)/(3*E*I)';
                             };
% Goals (need to have at least one)
    % Equations

    goals = {'Name',         'NegD', 'PosD', 'level', 'Target',
'Expression';
            '1/Mass        ','0.4',  '0.0', '1',     '0.002',
'1.0/m';
            '1/Displacement','0.6',  '0.0', '1',     '50',
'1.0/dis';
          };
% Constraints (need to have at least one)
    constraints = {'Name',     'type',               'Limit',
'Expression';
                   'MassMin',  'GreaterThanOrEqual','500'       'm';
                   'MassMax',  'LessThanOrEqual',   '1000'      'm';
                   'DisplMin', 'GreaterThanOrEqual','0.02'      'dis';
                   'DisplMax', 'LessThanOrEqual',   '0.1667'    'dis';
                  };

% Set XML Model variables (Convergence Tolerance)

docNode = com.mathworks.xml.XMLUtils.createDocument('Model');
docRootNode = docNode.getDocumentElement;

docRootNode.setAttribute('xmlns', 'http://dsides.configuration/model');
docRootNode.setAttribute('decisionVariableConvergenceTolerance',
'0.00000001');
docRootNode.setAttribute('deviationFunctionConvergenceTolerance',
'0.00000001');
docRootNode.setAttribute('writeCSVFile', 'true');
docRootNode.setAttribute('autoNormalizeConstraintsAndGoals', 'true');

% ------------------------------------
% NO EDITING AFTER THIS POINT REQUIRED
% ------------------------------------
```

Figure A.7: XML Generator Code for Cantilever Beam Example

Step 4: Running the MATLAB cDSP XML Generator.

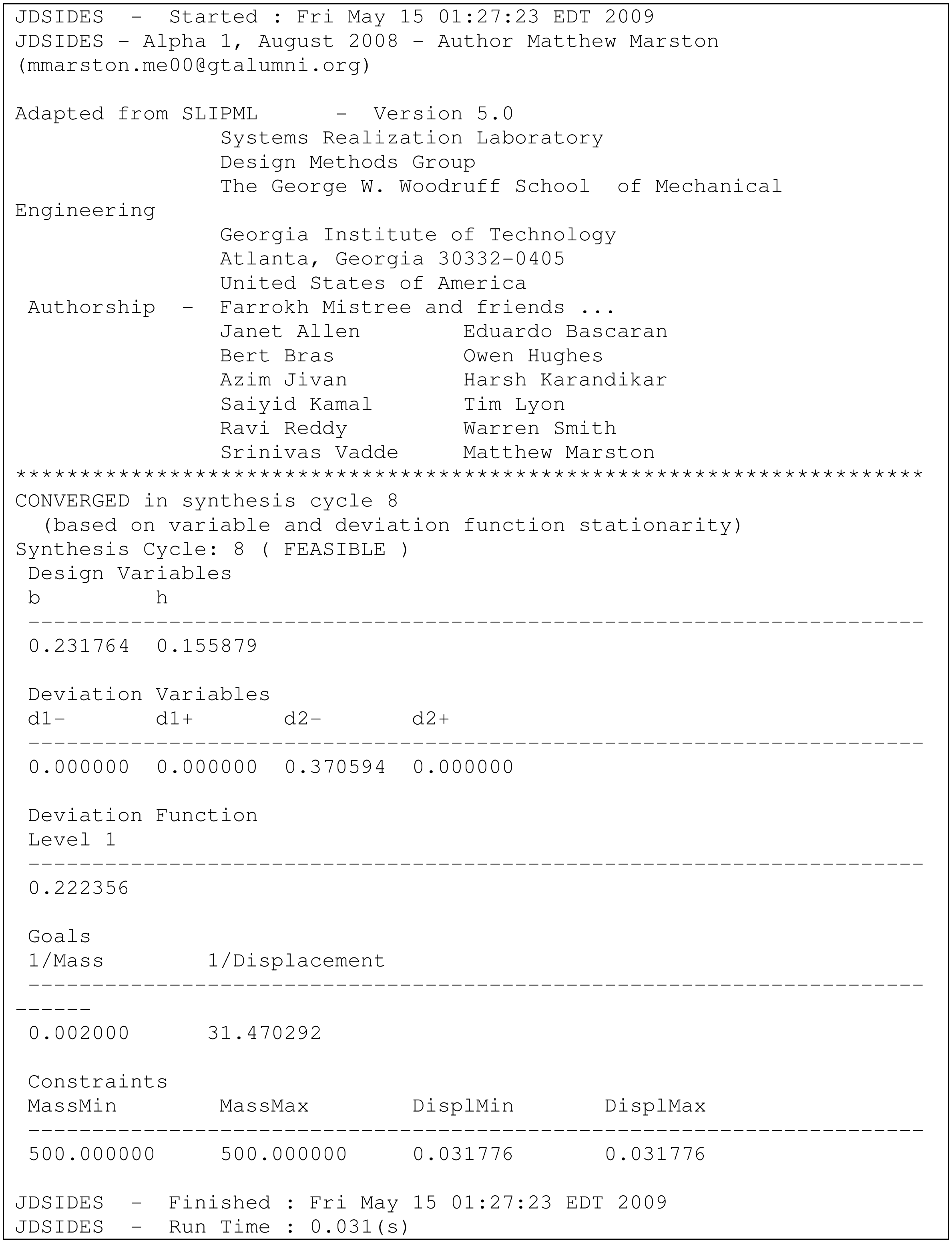
Running the MATLAB cDSP XML Generator will generate the XML file for this cDSP and will also execute JavaDSIDES for this design problem. The XML file to the inputs above looks like the following:

```
<?xml version="1.0" encoding="utf-8"?>
<Model xmlns="http://dsides.configuration/model"
autoNormalizeConstraintsAndGoals="true"
decisionVariableConvergenceTolerance="0.00000001"
deviationFunctionConvergenceTolerance="0.00000001" writeCSVFile="true">
   <Name>CantBeamExplicit</Name>
   <DecisionVariables>
      <DecisionVariable>
         <Name>b</Name>
         <Minimum>0.1</Minimum>
         <Maximum>1.5</Maximum>
         <InitialValue>0.1</InitialValue>
      </DecisionVariable>
      <DecisionVariable>
         <Name>h</Name>
         <Minimum>0.1</Minimum>
         <Maximum>1.5</Maximum>
         <InitialValue>0.1</InitialValue>
      </DecisionVariable>
   </DecisionVariables>
   <Constants>
      <Constant>
         <Name>P</Name>
         <Value>4.0</Value>
      </Constant>
      <Constant>
         <Name>E</Name>
         <Value>7.17E7</Value>
      </Constant>
      <Constant>
         <Name>L</Name>
         <Value>5.0</Value>
      </Constant>
      <Constant>
         <Name>rho</Name>
         <Value>2768.0</Value>
      </Constant>
   </Constants>
   <Goals>
      <IntermediateExpression>
         <Name>V</Name>
         <Expression>L*b*h</Expression>
      </IntermediateExpression>
      <IntermediateExpression>
         <Name>m</Name>
         <Expression>rho*V</Expression>
      </IntermediateExpression>
      <IntermediateExpression>
         <Name>I</Name>
         <Expression>(b*h^3)/12</Expression>
      </IntermediateExpression>
      <IntermediateExpression>
         <Name>dis</Name>
         <Expression>(P*L^3)/(3*E*I)</Expression>
      </IntermediateExpression>
      <Goal level="1" negativeDeviationCoefficient="0.4"
positiveDeviationCoefficient="0.0">
         <Name>1/Mass        </Name>
         <Expression>1.0/m</Expression>
         <Target>0.002</Target>
      </Goal>
      <Goal level="1" negativeDeviationCoefficient="0.6"
positiveDeviationCoefficient="0.0">
         <Name>1/Displacement</Name>
         <Expression>1.0/dis</Expression>
         <Target>50</Target>
      </Goal>
   </Goals>
   <Constraints>
      <IntermediateExpression>
         <Name>V</Name>
         <Expression>L*b*h</Expression>
      </IntermediateExpression>
      <IntermediateExpression>
         <Name>m</Name>
         <Expression>rho*V</Expression>
      </IntermediateExpression>
      <IntermediateExpression>
         <Name>I</Name>
         <Expression>(b*h^3)/12</Expression>
      </IntermediateExpression>
      <IntermediateExpression>
         <Name>dis</Name>
         <Expression>(P*L^3)/(3*E*I)</Expression>
      </IntermediateExpression>
      <Constraint type="GreaterThanOrEqual">
         <Name>MassMin</Name>
         <Expression>m</Expression>
         <Limit>500</Limit>
      </Constraint>
      <Constraint type="LessThanOrEqual">
         <Name>MassMax</Name>
         <Expression>m</Expression>
         <Limit>1000</Limit>
      </Constraint>
      <Constraint type="GreaterThanOrEqual">
         <Name>DisplMin</Name>
         <Expression>dis</Expression>
         <Limit>0.02</Limit>
      </Constraint>
      <Constraint type="LessThanOrEqual">
         <Name>DisplMax</Name>
         <Expression>dis</Expression>
         <Limit>0.1667</Limit>
      </Constraint>
   </Constraints>
</Model>
```

Figure A.8: XML Model for Cantilever Beam Example

The reason for developing the MATLAB cDSP XML Generator should be very obvious now. Although the design problem is not very complex, the XML is already several pages long.

The results are displayed in the MATLAB command window and they are also saved in two files in the `C:\dsides\dsides-app` folder. The output seen in the MATLAB command windows is saved in a .txt file and the intermediate results of the synthesis cycles are saved in a comma-separated-value file (.csv). The text file's content is presented in Figure A.9.



```
JDSIDES  -  Started : Fri May 15 01:27:23 EDT 2009
JDSIDES - Alpha 1, August 2008 - Author Matthew Marston
(mmarston.me00@gtalumni.org)

Adapted from SLIPML       -  Version 5.0
                Systems Realization Laboratory
                Design Methods Group
                The George W. Woodruff School  of Mechanical
Engineering
                Georgia Institute of Technology
                Atlanta, Georgia 30332-0405
                United States of America
 Authorship  -  Farrokh Mistree and friends ...
                Janet Allen        Eduardo Bascaran
                Bert Bras          Owen Hughes
                Azim Jivan         Harsh Karandikar
                Saiyid Kamal       Tim Lyon
                Ravi Reddy         Warren Smith
                Srinivas Vadde     Matthew Marston
***********************************************************************
CONVERGED in synthesis cycle 8
  (based on variable and deviation function stationarity)
Synthesis Cycle: 8 ( FEASIBLE )
 Design Variables
 b         h
 ----------------------------------------------------------------------
 0.231764  0.155879

 Deviation Variables
 d1-       d1+       d2-       d2+
 ----------------------------------------------------------------------
 0.000000  0.000000  0.370594  0.000000

 Deviation Function
 Level 1
 ----------------------------------------------------------------------
 0.222356

 Goals
 1/Mass        1/Displacement
 ----------------------------------------------------------------------
------
 0.002000      31.470292

 Constraints
 MassMin        MassMax        DisplMin       DisplMax
 ----------------------------------------------------------------------
 500.000000     500.000000     0.031776       0.031776

JDSIDES  -  Finished : Fri May 15 01:27:23 EDT 2009
JDSIDES  -  Run Time : 0.031(s)
```

Figure A.9: JavaDSIDES Onscreen Output

Step 5: Using and interpreting the results.

The csv file can be opened in a spreadsheet application like excel and contains the following data:

Table A.1: JavaDSIDES CSV File – Part 1

Synthesis Cycle	Constraint Violation	b	h	d1-	d1+	d2-	d2+	Level 1
1	1.396506	0.1	0.1	0	0	0.9283	0	0.55698
2	0.522998	0.172327	0.1	0	0	0.876441	0	0.525865
3	0.244324	0.231996	0.117677	0	0	0.728938	0	0.437363
4	0	0.259953	0.141543	0.018134	0	0.471464	0	0.290132
5	0	0.232047	0.155689	1.19E-06	0	0.372129	0	0.223278
6	0	0.231796	0.155857	1.43E-08	0	0.370772	0	0.222463
7	0	0.231769	0.155876	7.79E-10	0	0.370626	0	0.222375
8	0	0.231764	0.155879	1.46E-10	0	0.370594	0	0.222356
8	0	0.231764	0.155879	1.46E-10	0	0.370594	0	0.222356

Table A.2: JavaDSIDES CSV File – Part 2

Synthesis Cycle	1/Mass	1/Displacement	MassMin	MassMax	DisplMin	DisplMax
1	0.007225	3.585	138.4	138.4	0.27894	0.27894
2	0.004193	6.177931	238.5009	238.5009	0.161866	0.161866
3	0.002647	13.55311	377.8381	377.8381	0.073784	0.073784
4	0.001964	26.42682	509.2345	509.2345	0.03784	0.03784
5	0.002	31.39353	500.0006	500.0006	0.031854	0.031854
6	0.002	31.4614	500	500	0.031785	0.031785
7	0.002	31.46871	500	500	0.031778	0.031778
8	0.002	31.47029	500	500	0.031776	0.031776
8	0.002	31.47029	500	500	0.031776	0.031776

This information allows, for example, generating the convergence plots of the optimization algorithm:

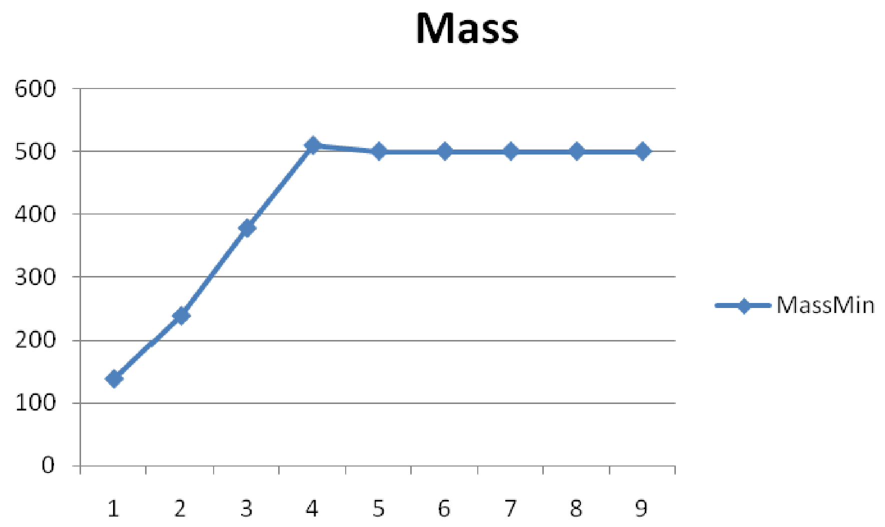


Figure A.10: Convergence Plot for Cantilever Beam Mass

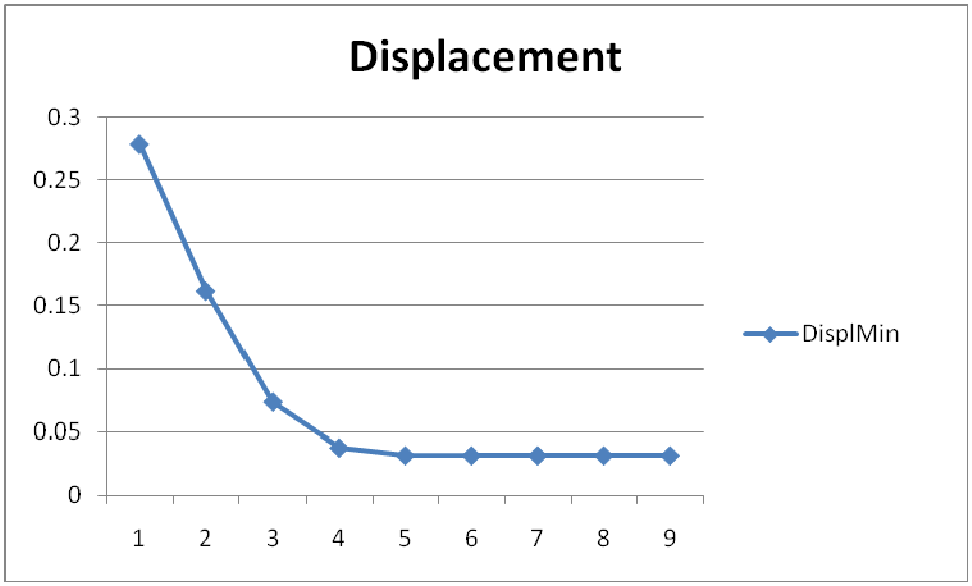


Figure A.11: Convergence Plot for Cantilever Beam Displacement

A.13 Cantilever Beam Example with Implicit MATLAB Functions

For this example, the explicit functions governing this example have been implemented in several MATLAB functions. The main function “CantBeam.m” returns an array containing the value for mass and displacement given a certain breadth *b* and height *h* of the beam. This means, the model is regarded as a black box model. However, for the completeness of this manual and to understand the MATLAB to Java conversion, the MATLAB functions are shown here:

```
function result = CantBeam(b,h)
%Constants
P = 4;
E = 7.17e7;
L = 5;
rho = 2768;
%Functions
V = CantBeamVolume(L,b,h);
I = CantBeamInertia(b,h);
mass = rho * V;
dis = (P*L^3)/(3*E*I);
result = [mass dis];
end
```

It can be seen that the CantBeam.m function calls CantBeamVolume.m and CantBeamInertia.m. Although it would be much faster if the functions would be integrated, the model consists of several functions in order to demonstrate the MATLAB to Java conversion more completely.

The MATLAB functions CantBeamVolume.m and CantBeamInertia.m are shown here:

```
function volume = CantBeamVolume(L,b,h)
volume = L*b*h;
end
```

```
function inertia = CantBeamInertia(b,h)
inertia = (b*h^3)/12;
end
```

Because JavaDSIDES cannot call MATLAB functions directly, they have to be wrapped in Java to allow to be called in the JavaDSIDES model class.

Step 1: Formulation of the general cDSP for the Design Problem.

Given:

- Constants:
  - All defined in the black box model
- Equations:
  - All defined in the black box model
- Models
  - Black box model for m and δ with inputs b and h

Find

- Beam breadth: b = [0.1 m, 1.5m]
- Beam Height: h = [0.1 m, 1.5m]
- d_i^- , d_i^+ = [0,1]

Satisfy

- **Bounds:**
  - See bounds on design variables
- **Goals:**
  - Minimize mass with goal 500 kg: $\frac{500\text{kg}}{\text{m(b, h)}} + d_1^- - d_1^+ = 1$
  - Minimize displacement with goal 0.02 m: $\frac{0.2\text{ m}}{\delta\text{(b, h)}} + d_2^- - d_2^+ = 1$
- **Constraints:**
  - $m \geq 500kg$
  - $m \leq 1000kg$
  - $\delta \geq 0.02m$
  - $\delta \leq 0.1667m$

Minimize

$$Z = 0.4 \cdot d_1^- + 0.6 \cdot d_2^-$$

Figure A.12: cDSP for Cantilever Beam Example with Implicit Functions

Step 2: Transformation of the goal and constraint formulation according to the conventions.

According to Section A.5, the goal functions have to be transformed to maximization problems:

Mass:

- minimize m(b,h) maximize 1/m(b,h)
- min goal 500kg max goal 1/500kg

Displacement:

- minimize δ(b,h) maximize 1/ δ(b,h)
- min goal 0.02m max goal 1/0.02m

Step 3: Translation of the cDSP into the MATLAB cDSP XML Generator

The MATLAB cDSP XML Generator is used to generate the preliminary java model class. Therefore, the cDSP is transferred to the XML Generator (Figure A.13). Note, that the cDSP formulation became simpler compared to the one in Section A.12 due to missing constants and intermediate expressions. For the goal and constraint formulations, where the MATLAB CantBeam function will be called later, a number is entered. This number is remembered and will be exchanged with the function call in the Java model class.

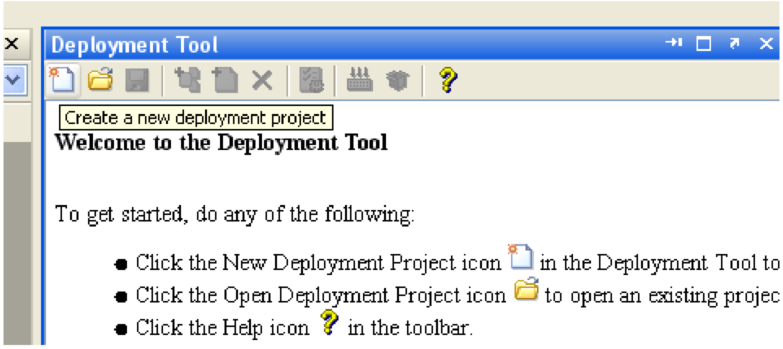
The formulation in the MATLAB cDSP XML Generator is presented in Figure A.13. In the code, “600” is used to replace the function call for the cantilever beam mass and “0.05” for its displacement. The results that are displayed when executing the XML Generator are ignored.

```
% INPUT
projectname = 'CantBeamImplicit';
% Design Variable Information
    dvariables = {'Name',    'minimum',  'maximum',  'initial value';
                  'b',       '0.1',      '1.5',      '0.1';
                  'h',       '0.1',      '1.5',      '0.1';
                 };
% Constants
    constants = {'Name',    'value';
                };
% Intermediate Expressions
    % used in goal expressions:
    goalintermediate = {'Name',     'Expression';
                       };
    % used in constraints expressions:
    constraintintermediate = {  'Name',     'Expression';
                             };
% Goals (need to have at least one)
    % Equations

    goals = {'Name',         'NegD', 'PosD', 'level', 'Target',
'Expression';
            '1/Mass        ','0.4',  '0.0', '1',     '0.002',
'1.0/600';
            '1/Displacement','0.6',  '0.0', '1',     '50',
'1.0/0.05';
          };
% Constraints (need to have at least one)
    constraints = {'Name',     'type',               'Limit',
'Expression';
                   'MassMin',  'GreaterThanOrEqual','500'        '600';
                   'MassMax',  'LessThanOrEqual',   '1000'       '600';
                   'DisplMin', 'GreaterThanOrEqual','0.02'       '0.05';
                   'DisplMax', 'LessThanOrEqual',   '0.1667'     '0.05';
                  };

% Set XML Model variables (Convergence Tolerance)
docNode = com.mathworks.xml.XMLUtils.createDocument('Model');
docRootNode = docNode.getDocumentElement;

docRootNode.setAttribute('xmlns', 'http://dsides.configuration/model');
docRootNode.setAttribute('decisionVariableConvergenceTolerance',
'0.00000001');
docRootNode.setAttribute('deviationFunctionConvergenceTolerance',
'0.00000001');
docRootNode.setAttribute('writeCSVFile', 'true');
docRootNode.setAttribute('autoNormalizeConstraintsAndGoals', 'true');

% ------------------------------------
% NO EDITING AFTER THIS POINT REQUIRED
% ------------------------------------
```

Figure A.13: XML Generator Formulation for Cantilever Beam Example

Step 4: Wrapping the MATLAB CantBeam functions in Java using MATLAB Compiler

In this step, the CantBeam MATLAB functions are wrapped in Java and exported. Therefore, the working directory in MATLAB is changed to the folder containing the CantBeam MATLAB files. With the MATLAB command "deploytool" then the MATLAB Compiler graphic user interface is started and a new deployment project is opened (see Figure A.14).



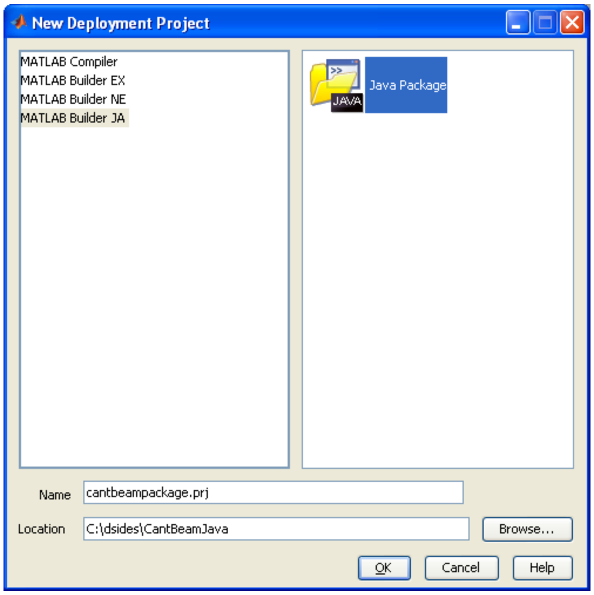
Welcome to the Deployment Tool

To get started, do any of the following:

- Click the New Deployment Project icon in the Deployment Tool to
- Click the Open Deployment Project icon to open an existing projec
- Click the Help icon in the toolbar.

Figure A.14: Starting a New Deployment Project in MATLAB Builder JA

The dialog box allows choosing the type of project, which is a MATLAB Builder JA Java Package project in this case. It also allows choosing in a project name and selecting an appropriate directory in which the project and the resulting java files are saved. The project name should be lower case since this will be the name of the .jar file later; here "cantbeampackage.prj' is used (see Figure A.15).



Name cantbeampackage.prj

Location C:\dsides\CantBeamJava

Figure A.15: New Project Dialog Box in MATLAB Builder JA

The project is created and two folders appear in the deployment tool GUI. The class folder should be renamed, so that it has a different name then the project or the function itself and starts with a capital letter. In this example, it is simply renamed to “CantBeamClass” (see Figure A.16).

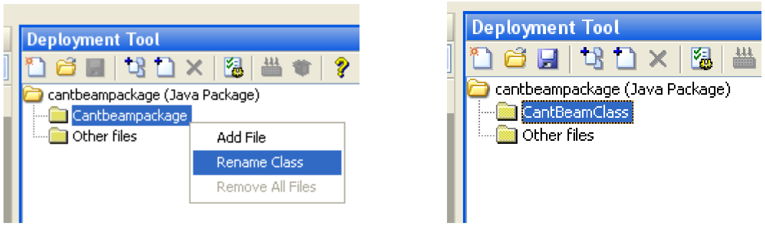


Figure A.16: Renaming Class in MATLAB Builder JA

After the class folder has been renamed, the required MATLAB function files are added to this class folder per drag & drop.

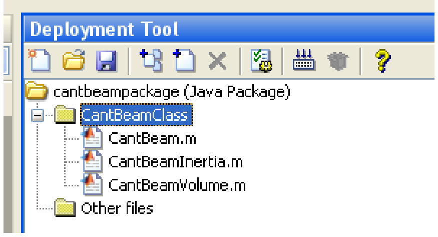


Figure A.17: Adding Functions to the Builder Project

After all functions are added (see Figure A.17), the project can be built by clicking "Build the project" (see Figure A.18). The default settings for building and packaging projects in MATLAB are fine for this application and don't have to be changed.

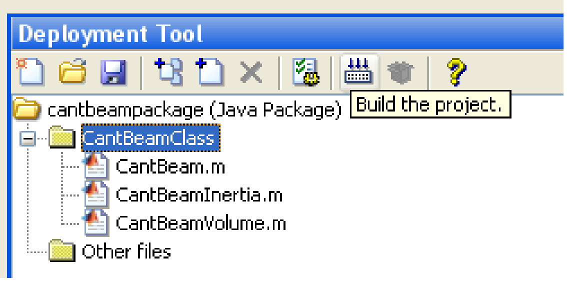


Figure A.18: Building the MATLAB Builder JA Project

The compiling and packaging process can take a couple of minutes. After the compilation is finished, the required .jar file "cantbeampackage.jar" can be found in the "…\CantBeamJava\cantbeampackage\distrib" subdirectory of the project folder. This file is copied into the “dsides-app” subdirectory (C:\dsides\dsides-app). The file “javabuilder.jar”, which can be found in the “MATLAB_ROOT\toolbox\javabuilder\jar” subfolder (e.g. “C:\ProgramFiles\MATLAB\R2008b\toolbox\javabuilder\jar”, is also copied (don’t move the file) to the “dsides-app” folder. Now all the Java packages are created and in the right place.

Step 5: Modifying the JavaDSIDES model file

The next step is to modify the Java model file, which can be found in the folder “C:\dsides\dsides-app\model\model” and in this case is named “CantBeamImplicit.java”. After opening the java file in an appropriate Java editor (here NetBeans) first, the javabuilder and cantbeamproject packages have to be imported:

```
package model;

import dsides.*;

import dsides.compiler.*;
import static java.lang.Math.*;

import java.util.*;

/* import packages here: */
import com.mathworks.toolbox.javabuilder.*;  /* MATLAB Builder Package
*/
import cantbeampackage.*;                    /* Canilever Beam
Functions */

public class CantBeamImplicit
  implements dsides.compiler.ModelImplementation
{
…
```

Next, a function providing the cantilever beam mass and displacement has to be added. This function will create an instance of the CantBeamClass class and call the CantBeam function with the given inputs.

```
public class CantBeamImplicit
  implements dsides.compiler.ModelImplementation
{
  public double[] cantBeamValues (double[] x)
   {
      CantBeamClass cantileverbeam = null;
      Object[] result = null;

      try
      {
         cantileverbeam = new CantBeamClass();
         /* Compute result */
         result = cantileverbeam.CantBeam(1, x[0], x[1]);
         MWNumericArray resultA = (MWNumericArray) result[0];
         /* Return result */
         double[][] a = (double[][])resultA.toDoubleArray();
         return a[0];
      }
      catch (Exception e)
      {
         System.out.println("Exception: " + e.toString());
      }
      finally
      {
         /* Free native resources */
         MWArray.disposeArray(result);
         if (cantileverbeam != null)
            cantileverbeam.dispose();
      }
      return null;
   }
…
```

This function can now be called in the "**`calculate`**" functions of the **`GoalProvider`** and the **`ConstraintProvider`** objects.

```
public double[] calculate (double[] x) /* in GoalProvider */
      {
        double[] f = new double[2];
        double[] r = cantBeamValues(x);

        f[0] = (1.0 / r[0]);
        f[1] = (1.0 / r[1]);

        return f;
      }
```

```
public double[] calculate (double[] x) /* in ConstraintProvider */
      {
        double[] f = new double[4];
        double[] r = cantBeamValues(x);

        f[0] = r[0];
        f[1] = r[0];
        f[2] = r[1];
        f[3] = r[1];

        return f;
      }
```

Now the model file can be saved and is ready to be compiled and executed.

<u>*Step 6: Compiling and executing the edited JavaDSIDES model file.*</u>

Therefore, the Windows command prompt is needed (WinKey+R, ”cmd”, Run). The command

```
cd c:\dsides\dsides-app
```

will browse to the dsides-app subdirectory, where the Java commands are invoked. With the following command the Java model file can now be compiled:

```
javac –classpath jdsides.jar;javabuilder.jar;cantbeampackage.jar –d
model model\model\CantBeamImplicit.java
```

If the compiling was successful, no output is displayed in the command window. After a successfully compiling, JavaDSIDES can be executed with the command:

```
java –classpath jdsides.jar;javabuilder.jar;cantbeampackage.jar;model
model.CantBeamImplicit
```

In this example, the same results as for the example with explicit functions should be displayed in the command window and saved in the output files. Note that due to the number of sub functions used in this section, JavaDSIDES will take much longer to finish than with explicit functions. The results can be used and interpreted exactly like in Step 5 of Section A.12.

APPENDIX B

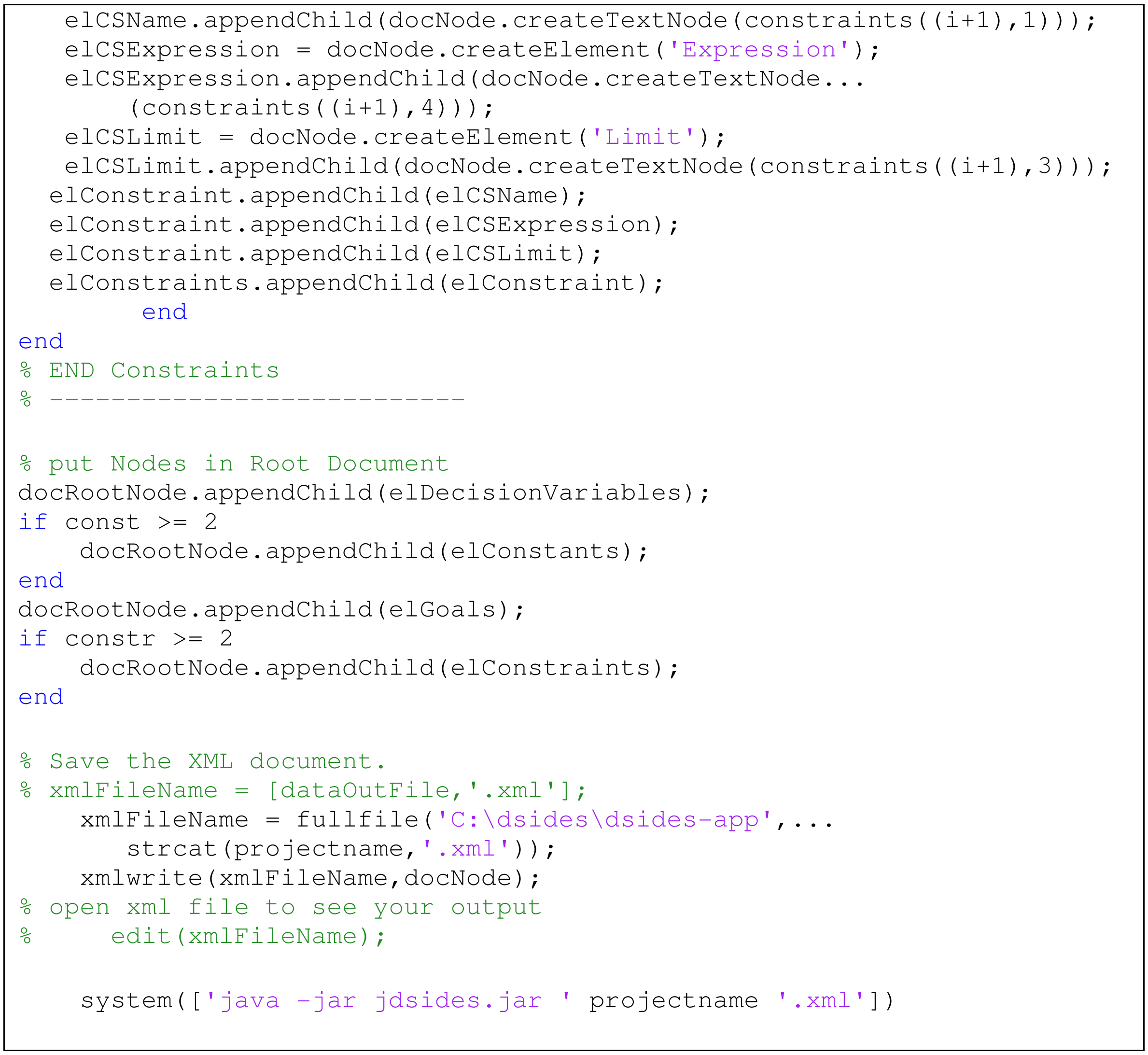
JavaDSIDES and the Modified RCEM: Implementation of the Pressure Vessel Example

In Section 4.3.2, the modified RCEM and JavaDSIDES are applied to the Pressure Vessel Example. In this section, the required MATLAB code is given and the implementation is explained. Furthermore, the JavaDSIDES output files are presented.

The MATLAB cDSP XML Generator code with the implemented cDSP (Figure 4.2) is shown below in Figure B.1. The first part of the code contains the translated cDSP with all variables, constants, intermediate expressions as well as goal and constraint formulations. The second part consists of the commands that translate and export the cDSP to an XML file. The last command executes JavaDSIDES with the generated input file. The results can be seen directly in the MATLAB command window.

```
clear all;
clc;

% MATLAB code that generates XML representation of a design problem and
% executes JavaDSIDES using the generated XML file
%
% by Markus Rippel, 2009 - markus.rippel@gatech.edu

% INPUT

projectname = 'Pressure_Vessel_Example_new_mRCEM_20_25_4';

% Design Variable Information
    dvariables = {'Name',    'minimum',   'maximum',   'initial value';
                  'R',       '6',         '36.0',      '20.0';
                  'L',       '6',         '48.0',      '25.0';
                  'T',       '1.5',       '6.0',       '4.0';
                  };

% Constants
    constants = {'Name',        'value';
                 'varianceR',   '0.44444';
                 'varianceL',   '0.44444';
                 'varianceT',   '0.11111';
                 'deltaR',      '1.33333';
                 'deltaL',      '1.33333';
                 'deltaT',      '0.66666';
                 'weight',      '0.25';
                 'pi',          '3.14159265';
                 'rho',         '0.284';
                 'ystrength',   '26000';
                 'SF',          '2';
                 'P',           '1000';
                };

 % Intermediate Expressions
    % used in goal expressions:
    goalintermediate = {
      'Name',       'Expression';
      'V',          'pi*(4/3*R^3 + R^2*L)';
      'W',          'pi*rho*(4/3*(R+T)^3+(R+T)^2*L-(4/3*R^3 + R^2*L))';
      'MeanV',      'V + 0.5*(pi*(4/3*6*R+2*L))*varianceR +
0.5*(0)*varianceL';
      'MeanW',      'W + 0.5*(pi*rho*(4/3*6*(R+T)+2*L-(4/3*6*R +
2*L)))*varianceR + 0.5*(0)*varianceL +
0.5*(pi*rho*(4/3*6*(R+T)+2*L))*varianceT';
      'V_plusdeltaR', 'pi*(4/3*(R+deltaR)^3 + (R+deltaR)^2*L)';
      'V_mindeltaR',  'pi*(4/3*(R-deltaR)^3 + (R-deltaR)^2*L)';
      'V_plusdeltaL', 'pi*(4/3*R^3 + R^2*(L+deltaL))';
      'V_mindeltaL',  'pi*(4/3*R^3 + R^2*(L-deltaL))';
      'W_plusdeltaR', 'pi*rho*(4/3*((R+deltaR)+T)^3+((R+deltaR)+T)^2*L-
(4/3*(R+deltaR)^3 + (R+deltaR)^2*L))';
      'W_mindeltaR',  'pi*rho*(4/3*((R-deltaR)+T)^3+((R-deltaR)+T)^2*L-
(4/3*(R-deltaR)^3 + (R-deltaR)^2*L))';
      'W_plusdeltaL', 'pi*rho*(4/3*(R+T)^3+(R+T)^2*(L+deltaL)-(4/3*R^3
+ R^2*(L+deltaL)))';
      'W_mindeltaL',  'pi*rho*(4/3*(R+T)^3+(R+T)^2*(L-deltaL)-(4/3*R^3
+ R^2*(L-deltaL)))';
      'W_plusdeltaT', 'pi*rho*(4/3*(R+(T+deltaT))^3+(R+(T+deltaT))^2*L-
(4/3*R^3 + R^2*L))';
      'W_mindeltaT',  'pi*rho*(4/3*(R+(T-deltaT))^3+(R+(T-deltaT))^2*L-
(4/3*R^3 + R^2*L))';
      'VarianceV',    '1/12*((V-V_mindeltaR)^2+(V-
V_mindeltaL)^2)+1/12*((V-V_plusdeltaR)^2+(V-
V_plusdeltaL)^2)+1/12*weight*((V_mindeltaR-
V_plusdeltaR)^2+(V_mindeltaL-V_plusdeltaL)^2)';
      'VarianceW',    '1/12*((W-W_mindeltaR)^2+(W-W_mindeltaL)^2+(W-
W_mindeltaT)^2)+1/12*((W-W_plusdeltaR)^2+(W-W_plusdeltaL)^2+(W-
W_plusdeltaT)^2)+1/12*weight*((W_mindeltaR-
W_plusdeltaR)^2+(W_mindeltaL-W_plusdeltaL)^2+(W_mindeltaT-
W_plusdeltaT)^2)';
                        };
    % used in constraints expressions:
    constraintintermediate = {
      'Name',                   'Expression';
      'sph_stresscon',          'P*(R+0.5*T)/(2*T)';
      'cyl_stresscon',          'P*(2*R^2+2*R*T+T^2)/(2*R*T+T^2)';
      'thinwallcon',            '5*T - R + 5';
      'overallradiuscon',       'R + T';
      'overalllengthcon',       'L + 2*R + 2*T';
      'V',                      'pi*(4/3*R^3 + R^2*L)';
      'W',            'pi*rho*(4/3*(R+T)^3+(R+T)^2*L-(4/3*R^3 + R^2*L))';
      'MeanV',        'V + 0.5*(pi*(4/3*6*R+2*L))*varianceR +
0.5*(0)*varianceL';
      'MeanW',        'W + 0.5*(pi*rho*(4/3*6*(R+T)+2*L-(4/3*6*R +
2*L)))*varianceR + 0.5*(0)*varianceL +
0.5*(pi*rho*(4/3*6*(R+T)+2*L))*varianceT';
                              };

% Goals (need to have at least one)
    % Equations

    goals = {
  'Name',           'NegD', 'PosD', 'level', 'Target',   'Expression';
  'V____________', '0.45', '0.0',  '1',    '50000',     'MeanV';
  '1/VarianceV__', '0.25', '0.0',  '1',    '0.0000002', '1/VarianceV';
  '1/VarianceW__', '0.30', '0.0',  '1',    '0.000005',  '1/VarianceW';
          };

% Constraints (need to have at least one)
    constraints = {
  'Name',            'type',                 'Limit',   'Expression';
  'Constraint1  ',   'LessThanOrEqual',      '5',       'thinwallcon';
  'Constraint2  ',   'LessThanOrEqual',      '40',      'overallradiuscon';
  'Constraint3  ',   'LessThanOrEqual',      '60',      'overalllengthcon';
  'Constraint4  ',   'LessThanOrEqual',      '13000',   'sph_stresscon';
  'Constraint5  ',   'LessThanOrEqual',      '13000',   'cyl_stresscon';
  'Constraint6  ',   'GreaterThanOrEqual',   '20000',   'MeanV';
  'Constraint7  ',   'LessThanOrEqual',      '2500',    'MeanW';
             };

% Set XML Model variables (Convergence Tolerance)

docNode = com.mathworks.xml.XMLUtils.createDocument('Model');
docRootNode = docNode.getDocumentElement;

docRootNode.setAttribute('xmlns', 'http://dsides.configuration/model');
docRootNode.setAttribute('decisionVariableConvergenceTolerance',...
'0.0000000001');
docRootNode.setAttribute('deviationFunctionConvergenceTolerance',...
'0.0000000001');
docRootNode.setAttribute('writeCSVFile', 'true');
docRootNode.setAttribute('autoNormalizeConstraintsAndGoals', 'true');
docRootNode.setAttribute('adaptConstraints', 'false');
% ------------------------------------
% NO EDITING AFTER THIS POINT REQUIRED
% ------------------------------------
```

```
elName = docNode.createElement('Name');
elName.appendChild(docNode.createTextNode(projectname));
docRootNode.appendChild(elName);

% ---------------------------
% START DecisionVariables
    elDecisionVariables = docNode.createElement('DecisionVariables');
    d=size(dvariables);
     for i = 1:d(1)-1
  elVariable = docNode.createElement('DecisionVariable');
     elDName = docNode.createElement('Name');
     elDName.appendChild(docNode.createTextNode(dvariables(i+1,1)));
     elDmin = docNode.createElement('Minimum');
     elDmin.appendChild(docNode.createTextNode(dvariables(i+1,2)));
     elDmax = docNode.createElement('Maximum');
     elDmax.appendChild(docNode.createTextNode(dvariables(i+1,3)));
     elDInitial = docNode.createElement('InitialValue');
     elDInitial.appendChild(docNode.createTextNode(dvariables(i+1,4)));
  elVariable.appendChild(elDName);
  elVariable.appendChild(elDmin);
  elVariable.appendChild(elDmax);
  elVariable.appendChild(elDInitial);
  elDecisionVariables.appendChild(elVariable);

     end
% END DecisionVariables
% ---------------------------
% START Constants
const=size(constants);
if const >= 2
elConstants = docNode.createElement('Constants');

    for i = 1:(const(1)-1)
     elConstant = docNode.createElement('Constant');
        elCName = docNode.createElement('Name');
        elCName.appendChild(docNode.createTextNode(constants(i+1,1)));
        elCValue = docNode.createElement('Value');
        elCValue.appendChild(docNode.createTextNode(constants(i+1,2)));
     elConstant.appendChild(elCName);
     elConstant.appendChild(elCValue);
     elConstants.appendChild(elConstant);
    end
end
% END Constants
% ---------------------------
% START Goals
elGoals = docNode.createElement('Goals');
    % put in intermediate expressions
        ginter=size(goalintermediate);
        if ginter >= 2
         for i=2:(ginter(1))
           % for each Intermediate Expressions
  elIntermediate1 = docNode.createElement('IntermediateExpression');
    elIName = docNode.createElement('Name');
    elIName.appendChild(docNode.createTextNode(goalintermediate(i,1)));
    elIExpression = docNode.createElement('Expression');
    elIExpression.appendChild(docNode.createTextNode...
       (goalintermediate(i,2)));
  elIntermediate1.appendChild(elIName);
  elIntermediate1.appendChild(elIExpression);
  elGoals.appendChild(elIntermediate1);
        end
       end
    % put in goals
        g=size(goals);
        for i=1:(g(1)-1)
            %for each Goal
   elGoal = docNode.createElement('Goal');
   elGoal.setAttribute('negativeDeviationCoefficient', goals((i+1),2));
   elGoal.setAttribute('positiveDeviationCoefficient', goals((i+1),3));
   elGoal.setAttribute('level', goals((i+1),4));
     elGName = docNode.createElement('Name');
     elGName.appendChild(docNode.createTextNode(goals((i+1),1)));
     elGExpression = docNode.createElement('Expression');
     elGExpression.appendChild(docNode.createTextNode(goals((i+1),6)));
     elGTarget = docNode.createElement('Target');
     elGTarget.appendChild(docNode.createTextNode(goals((i+1),5)));
   elGoal.appendChild(elGName);
   elGoal.appendChild(elGExpression);
   elGoal.appendChild(elGTarget);
   elGoals.appendChild(elGoal);
        end

% END Goals
% ---------------------------
% START Constraints
constr=size(constraints);
if constr >= 2
elConstraints = docNode.createElement('Constraints');
    % put in intermediate expressions
      cinter=size(constraintintermediate);
      if cinter >= 2
        for i=2:(cinter(1))
           % for each Intermediate Expressions
    elIntermediate1 = docNode.createElement('IntermediateExpression');
           elIName = docNode.createElement('Name');
           elIName.appendChild(docNode.createTextNode...
              (constraintintermediate(i,1)));
           elIExpression = docNode.createElement('Expression');
           elIExpression.appendChild(docNode.createTextNode...
              (constraintintermediate(i,2)));
    elIntermediate1.appendChild(elIName);
    elIntermediate1.appendChild(elIExpression);
    elConstraints.appendChild(elIntermediate1);
        end
      end
    % put in constraints
        for i=1:(constr(1)-1)
  elConstraint = docNode.createElement('Constraint');
  elConstraint.setAttribute('type', constraints((i+1),2));
   elCSName = docNode.createElement('Name');
```



```
   elCSName.appendChild(docNode.createTextNode(constraints((i+1),1)));
   elCSExpression = docNode.createElement('Expression');
   elCSExpression.appendChild(docNode.createTextNode...
       (constraints((i+1),4)));
   elCSLimit = docNode.createElement('Limit');
   elCSLimit.appendChild(docNode.createTextNode(constraints((i+1),3)));
  elConstraint.appendChild(elCSName);
  elConstraint.appendChild(elCSExpression);
  elConstraint.appendChild(elCSLimit);
  elConstraints.appendChild(elConstraint);
        end
end
% END Constraints
% ---------------------------

% put Nodes in Root Document
docRootNode.appendChild(elDecisionVariables);
if const >= 2
    docRootNode.appendChild(elConstants);
end
docRootNode.appendChild(elGoals);
if constr >= 2
    docRootNode.appendChild(elConstraints);
end

% Save the XML document.
% xmlFileName = [dataOutFile,'.xml'];
    xmlFileName = fullfile('C:\dsides\dsides-app',...
       strcat(projectname,'.xml'));
    xmlwrite(xmlFileName,docNode);
% open xml file to see your output
%     edit(xmlFileName);

    system(['java -jar jdsides.jar ' projectname '.xml'])
```

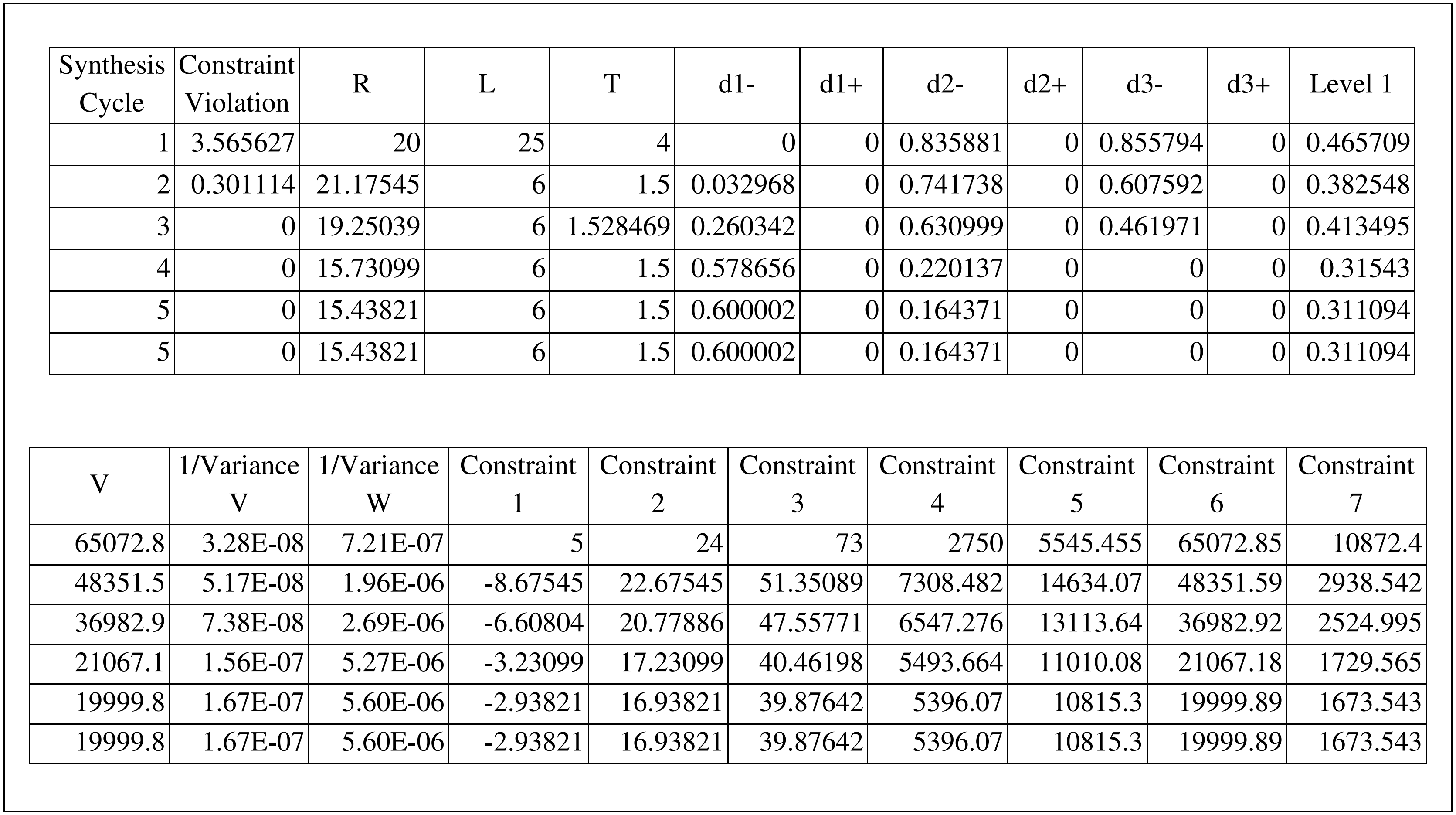
Figure B.1: MATLAB XML Generator Code for Java DSIDES

JavaDSIDES also provides two output files. One is a text file (filename.out), which contains a copy of the output screen of JavaDSIDES. The content of the output file from the example problem is given in Figure B.2.

```
JDSIDES  -  Started : Thu Sep 24 10:56:35 EDT 2009
JDSIDES - Alpha 1, August 2008 - Author Matthew Marston
(mmarston.me00@gtalumni.org)

Adapted from SLIPML      -  Version 5.0
               Systems Realization Laboratory
               Design Methods Group
               The George W. Woodruff School  of Mechanical Engineering
               Georgia Institute of Technology
               Atlanta, Georgia 30332-0405
               United States of America
 Authorship  - Farrokh Mistree and friends ...
               Janet Allen        Eduardo Bascaran
               Bert Bras          Owen Hughes
               Azim Jivan         Harsh Karandikar
               Saiyid Kamal       Tim Lyon
               Ravi Reddy         Warren Smith
               Srinivas Vadde     Matthew Marston
***********************************************************************
CONVERGED in synthesis cycle 5
  (based on variable and deviation function stationarity)
Synthesis Cycle: 5 ( FEASIBLE )
 Design Variables
 R         L         T
 ----------------------------------------------------------------------
 15.438209 6.000000  1.500000

 Deviation Variables
 d1-       d1+       d2-       d2+       d3-       d3+
 ----------------------------------------------------------------------
 0.600002  0.000000  0.164371  0.000000  0.000000  0.000000

 Deviation Function
 Level 1
 ----------------------------------------------------------------------
 0.311094

 Goals
 V_____________1/VarianceV__1/VarianceW__
 ----------------------------------------------------------------------
 19999.894863 0.000000     0.000006

 Constraints
 Constraint1  Constraint2  Constraint3  Constraint4  Constraint5
Constraint6  Constraint7
 ----------------------------------------------------------------------
 -2.938209    16.938209    39.876419    5396.069809  10815.304626
19999.894863 1673.543008

JDSIDES  -  Finished : Thu Sep 24 10:56:35 EDT 2009
JDSIDES  -  Run Time : 0.063(s)
```

Figure B.2: JavaDSIDES Onscreen Output and Content of Output 'filename.out'

The second output file is a comma separated value file (filename.csv), which can be opened in spreadsheet software like Microsoft Excel. This file contains all the intermediate results of the DSIDES iterations. The csv file from the example problem is shown in Figure B.3. The data found in this file can be used to verify the internal consistency of the result, since it allows, for example, plotting the algorithm's convergence.



Synthesis Cycle	Constraint Violation	R	L	T	d1-	d1+	d2-	d2+	d3-	d3+	Level 1
1	3.565627	20	25	4	0	0	0.835881	0	0.855794	0	0.465709
2	0.301114	21.17545	6	1.5	0.032968	0	0.741738	0	0.607592	0	0.382548
3	0	19.25039	6	1.528469	0.260342	0	0.630999	0	0.461971	0	0.413495
4	0	15.73099	6	1.5	0.578656	0	0.220137	0	0	0	0.31543
5	0	15.43821	6	1.5	0.600002	0	0.164371	0	0	0	0.311094
5	0	15.43821	6	1.5	0.600002	0	0.164371	0	0	0	0.311094

V	1/Variance V	1/Variance W	Constraint 1	Constraint 2	Constraint 3	Constraint 4	Constraint 5	Constraint 6	Constraint 7
65072.8	3.28E-08	7.21E-07	5	24	73	2750	5545.455	65072.85	10872.4
48351.5	5.17E-08	1.96E-06	-8.67545	22.67545	51.35089	7308.482	14634.07	48351.59	2938.542
36982.9	7.38E-08	2.69E-06	-6.60804	20.77886	47.55771	6547.276	13113.64	36982.92	2524.995
21067.1	1.56E-07	5.27E-06	-3.23099	17.23099	40.46198	5493.664	11010.08	21067.18	1729.565
19999.8	1.67E-07	5.60E-06	-2.93821	16.93821	39.87642	5396.07	10815.3	19999.89	1673.543
19999.8	1.67E-07	5.60E-06	-2.93821	16.93821	39.87642	5396.07	10815.3	19999.89	1673.543

Figure B.3: JavaDSIDES CSV Output File

APPENDIX C

MATLAB CODE FOR PCM IMPLEMENTATION

C.1 PCM MATLAB Code for Example in Chapter 5

In this section, the MATLAB code required to reproduce the results in Chapter 5 is presented. The main function code in given in Figure C.1, the sub-function myPCM, which is given in Figure C.2, was provided by Professor Seung-Kyum Choi. The function 'ThesisFunction()' is the MATLAB representation of Eq. 3.1. All other functions that are used in this code are native MATLAB functions.

```
function result = PCM_ExampleFunction(mu,variance)

% Function returns mean and variance of the example function
% (ThesisFunction) for the input mean and variance assuming a Gaussian
% distribution using the Probabilistic Collocation Method of third
% order
% by Markus Rippel, 2009 - markus.rippel@gatech.edu

sigma = sqrt(variance);

% Generating of Hermite Polynomials
h0 = myPCE(0);
h1 = myPCE(1);
h2 = myPCE(2);
h3 = myPCE(3);
h4 = myPCE(4);
h5 = myPCE(5);
h.a = h0;
h.b = h1;
h.c = h2;
h.d = h3;

% Deriving Collocation Points
r = roots(h4)
r5 = roots(h5);
```

```
B = mu +r.*sigma
Y = ThesisFunction(B)

% Solving for Coefficients of Polynomial Chaos Expansion
X = [polyval(h.a, r(1)), polyval(h.b, r(1)), polyval(h.c, r(1)),
polyval(h.d, r(1));
     polyval(h.a, r(2)), polyval(h.b, r(2)), polyval(h.c, r(2)),
polyval(h.d, r(2));
     polyval(h.a, r(3)), polyval(h.b, r(3)), polyval(h.c, r(3)),
polyval(h.d, r(3));
     polyval(h.a, r(4)), polyval(h.b, r(4)), polyval(h.c, r(4)),
polyval(h.d, r(4))]
y = pinv(X'*X)*X'*Y

% Deriving Moments
mean1 = y(1);
variance1 = y(2)^2 + y(3)^2 + y(4)^2;

% Checking Fitting Error
T = mu +r5.*sigma;
Z = ThesisFunction(T);
for i=1:5
    Ze(i,1) = y(1)*polyval(h0, r5(i))+ y(2)*polyval(h1, r5(i))+
y(3)*polyval(h2, r5(i))+ y(4)*polyval(h3, r5(i));
end
TSS = sum((Z-mean(Z)).^2);
RSS = sum((Z-Ze).^2);
Rsquared = 1 - RSS/TSS
% Displying Warning if Error is too Large
if Rsquared<=0.95
    [mu, Rsquared]
end

result = [mean1 variance1];
end
```

Figure C.1: MATLAB Code for PCM Example in Chapter 5 – “PCM_ExampleFunction.m"

```
function [hk]=myPCE(n)

% Return the coefficient of n_th order of PCE: x^(n+1-m)
% polyval(myPCE(n),x) evaluates H_n(x).
%
% Usage Example:
% hh = myPCE(n)
% x = polyval(hh, xi)

if n==0
    hk = 1;

elseif n==1
    hk = [1 0];

else
    hkm2 = zeros(1,n+1);
    hkm2(n+1) = 1;
    hkm1 = zeros(1,n+1);
    hkm1(n) = 1;
    for k=2:n
        hk = zeros(1,n+1);
        for e=n-k+1:1:n
            hk(e) = 1*(hkm1(e+1) - (k-1)*hkm2(e));
        end
        hk(n+1) = -1*(k-1)*hkm2(n+1);
        if k<n
            hkm2 = hkm1;
            hkm1 = hk;
        end
    end
end
```

Figure C.2: MATLAB Code for Generation of Hermite Polynomoals – “myPCE.m"

C.2 PCM MATLAB Code for Example in Chapter 6

In this section, the MATLAB code for the multi-dimension PCM in Chapter 6 is presented. The code consists of the function “PCM_LCA_Thesis()” as well as two important subfunctions “myPCE()” and “myPCM_solve()”.

The function PCM_LCA_Thesis.m is the main function that takes the user input, the mean values of the design variables and their variance, and returns the mean and variance of the LCA design solution. In the simulation-based RCEM, this function is called within the DSIDES algorithm. The code is given in Figure C.3.

Within the main function "PCM_LCA_Thesis()", the subfunction "myPCE()" generates Hermite polynomials of the desired order. The code was provided by Professor Seung-Kyum Choi and is given in Figure C.4.

The subfunction "myPCM_solve()" returns the coefficients of the Polynomial Chaos Expansion within the Probabilistic Collocation Method. The code of this function is presented in Figure C.5.

The code for the LCA FEA simulation, which is called with the function "MR_LCA_Thesis()" is not presented in this thesis since the focus is on the design method.

```
function result = PCM_LCA_Thesis_Q_2order(row, tx, ty,rowvar, txvar,
tyvar)
sigma(1) = rowvar^(0.5);
sigma(2) = txvar^(0.5);
sigma(3) = tyvar^(0.5);

% Define Terms of the Polynomial Chaos Expansion
 m =    [0 0 0;
         1 0 0;
         0 1 0;
         0 0 1;
         2 0 0;
         0 2 0;
         0 0 2];

%% Generate Hermite Polynomials
h0 = myPCE(0);
h1 = myPCE(1);
h2 = myPCE(2);
h3 = myPCE(3);
```

```
h4 = myPCE(4);
h(1,:) = [0,0,0,h0];
h(2,:) = [0,0,h1];
h(3,:) = [0,h2];
h(4,:) = [h3];

%% Generate and Select Required Number of Collocation Points
r = roots (h3);
cm = fullfact([3 3 3]);
CP = [r(cm(:,1)) r(cm(:,2)) r(cm(:,3))];
CP(:,4) = normpdf((CP(:,1).*1),0,1).*normpdf((CP(:,2).*1),0,1).
*normpdf((CP(:,3).*1),0,1);
CP = sortrows(CP,-4);
cp = CP(1:7,1:3);

%% Calculate Y (Simulated Values for Collocation Points)
C1 = row + cp(:,1).*sigma(1);
C2 = tx + cp(:,2).*sigma(2);
C3 = ty + cp(:,3).*sigma(3);

for i = 1:length(C1)
    Y(i,1) = MR_LCA_Thesis(C1(i),C2(i),C3(i));
end

%% Solve for the Coefficients q
res = myPCM_solve(Y,cp,h,m);
q = res{1};
m = res{2};
F = res{3};

%% Calculate mean and variance
qmean = q(1);
qvariance = 0;
for i = 2:length(q)
    qvariance = qvariance + q(i)^2;
end

% %% Checking Fitting Error
% r4 = roots(h4);
% tm = fullfact([4 4 4]);
% tp = [r4(tm(:,1)) r4(tm(:,2)) r4(tm(:,3))];
% for i = 1:length(m)
%     for j = 1:length(tp)
%         F(j,i) = polyval(h(m(i,1)+1,:),
tp(j,1))*polyval(h(m(i,2)+1,:), tp(j,2))*polyval(h(m(i,3)+1,:),
tp(j,3));
%     end
% end
% Ze = F*q;
%
% T1 = row + tp(:,1).*sigma(1);
% T2 = tx + tp(:,2).*sigma(2);
% T3 = ty + tp(:,3).*sigma(3);
% for i = 1:length(T1)
%     Z(i,1) = MR_LCA_Thesis(T1(i),T2(i),T3(i));
% end
% Z;
% TSS = sum((Z-mean(Z)).^2);
% RSS = sum((Z-Ze).^2);
% Rsquared = 1 - RSS/TSS
% % Displying Warning if Error is too Large
% if Rsquared<=0.95
%     [Rsquared]
% end

%%
result = [qmean qvariance];
end
```

Figure C.3: MATLAB Code for the PCM with the LCA example – PCM_LCA_Thesis.m

```
function [hk]=myPCE(n)

% Return the coefficient of n_th order of PCE: x^(n+1-m)
% polyval(myPCE(n),x) evaluates H_n(x).
% Usage Example:
% hh = myPCE(n)
% x = polyval(hh, xi)

if n==0
    hk = 1;
elseif n==1
    hk = [1 0];
else
    hkm2 = zeros(1,n+1);
    hkm2(n+1) = 1;
    hkm1 = zeros(1,n+1);
    hkm1(n) = 1;
    for k=2:n
        hk = zeros(1,n+1);
        for e=n-k+1:1:n
            hk(e) = 1*(hkm1(e+1) - (k-1)*hkm2(e));
        end
        hk(n+1) = -1*(k-1)*hkm2(n+1);
        if k<n
            hkm2 = hkm1;
            hkm1 = hk;
        end
    end
end
```

Figure C.4: MATLAB Code for the Generation of Hermite Polynomials – myPCE.m

```
function result = myPCM_solve(Y,cp,h,m)

for i = 1:length(m)
    for j = 1:length(m)
        F(j,i) = polyval(h(m(i,1)+1,:), cp(j,1))*polyval(h(m(i,2)+1,:),
cp(j,2))*polyval(h(m(i,3)+1,:), cp(j,3));
    end
end


q = pinv(F'*F)*F'*Y;
result = {q, m, F};
end
```

Figure C.5: MATLAB Code for the Calculation of the PCE Coefficients – myPCM_solve.m

REFERENCES

Allen, J. K., Seepersad, C., Choi, H. and Mistree, F. (2006), "Robust Design for Multiscale and Multidisciplinary Applications," *Journal of Mechanical Design*, Vol. 128(4), pp. 832-43.

Beyer, H.-G. and Sendhoff, B. (2007), "Robust Optimization–a Comprehensive Survey," *Computer Methods in Applied Mechanics and Engineering*, Vol. 196(33-34), pp. 3190-218.

Borror, C. M. and Montgomery, D. C. (2000), "Mixed Resolution Designs as Alternatives to Taguchi Inner/Outer Array Designs for Robust Design Problems," *Quality and Reliability International*, Vol. 16, pp. 117-27.

Box, G. (1988), "Signal-to-Noise Ratios, Performance Criteria, and Transformations," *Technometrics*, Vol. 30, pp. 1-18.

Box, G. E. P. and Fung, C. (1993), "Is Your Robust Design Procedure Robust?," *Quality Engineering*, Vol. 6(3), pp. 503-14.

Bras, B. A. and Mistree, F. (1993), "Robust Design Using Compromise Decision Support Problems," *Engineering Optimization*, Vol. 2(3), pp. 213-39.

Cagan, J. and Williams, B. C. (1993), "First-Order Necessary Conditions for Robust Optimality," *ASME Advances in Design Automation*, Albuquerque, NM, ASME DE-Vol. 65-1, 539-49.

Charnes, A. and Cooper, W. W. (1977), "Goal Programming and Multiple Objective Optimization - Part I," *European Journal of Operational Research*, pp. 39-54.

Chen, W. (1995). "A Robust Concept Exploration Method for Configuring Complex Systems," *PhD Dissertation*, G. W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA, USA.

Chen, W., Allen, J. K., Mavris, D. and Mistree, F. (1996a), "A Concept Exploration Method for Determining Robust Top-Level Specifications," *Engineering Optimization*, Vol. 26(2), pp. 137-58.

Chen, W., Allen, J. K., Tsui, K.-L. and Mistree, F. (1996b), "A Procedure for Robust Design: Minimizing Variations Caused by Noise Factors and Control Factors," *ASME Journal of Mechanical Design*, Vol. 118(4), pp. 478-85.

Chen, W., Allen, J. K. and Mistree, F. (1997), "A Robust Concept Exploration Method for Enhancing Productivity in Concurrent Systems Design," *Concurrent Engineering: Research and Applications*, Vol. 5(3), pp. 203-17.

Chen, W., Wiecek, M. and Zhang, J. (1999), "Quality Utility: A Compromise Programming Approach to Robust Design," *Journal of Mechanical Design*, Vol. 121, pp. 179-87.

Chen, W., Atul, S., Messac, A. and Sundararaj, G. (2000), "Exploration of the Effectiveness of Physical Programming in Robust Design," *Journal of Mechanical Design*, Vol. 122, pp. 155-63.

Choi, S.-K., Grandhi, R. V. and Canfield, R. A. (2007), *Reliability-Based Structural Design*, Springer, London.

Church, B. C., Dempsey, B. M., Clark, J. L., Sanders, T. H. and Cochran, J. K. (2001), "Copper Alloys from Oxide Reduction for High Conductivity Applications," *Proceedings of IMECE 2001, International Mechanical Engineering Congress and Exposition*, New York, November 11-16, 2001.

Cochran, J. K., Lee, K. J., McDowell, D. L. and Sanders, T. H. (2000), "Low Density Monolithic Honeycombs by Thermal Chemical Processing," *Proceedings of the 4th Conference on Aerospace Materials, Processes, and Environmental Technology*, Huntsville, AL.

Cross, N. (2007), "Forty Years of Design Research," *Design Studies*, Vol. 28(1), pp. 1-4.

Davis, P. J. and Rabinowitz, P. (1975), *Methods of Numerical Integration*, Academic Press, New York.

Du, X. and Chen, W. (2000), "Towards a Better Understanding of Modeling Feasibility Robustness in Engineering Design," *ASME Journal of Mechanical Design*, Vol. 122(4), pp. 385-94.

Evans, A. G., Hutchinson, J. W., Fleck, N. A., Ashby, M. F. and Wadley, H. N. G. (2001), "The Topological Design of Multifunctional Cellular Materials," *Progress in Materials Science*, Vol. 46(3-4), pp. 309-27.

Finger, S. and Dixon, J. R. (1989a), "A Review of Research in Mechanical Engineering Design. Part 1: Descriptive, Prescriptive, and Computer-Based Models of Design Processes," *Research in Engineering Design*, Vol. 1, pp. 51-67.

Finger, S. and Dixon, J. R. (1989b), "A Review of Research in Mechanical Engineering Design. Part 2: Representations, Analysis, and Design for the Life Cycle," *Research in Engineering Design*, Vol. 1, pp. 121-37.

Frey, D. D. and Li, X. (2004), "Validating Robust Parameter Design Methods," *ASME Design Technical Conferences*, Salt Lake City, UT, Paper Number DETC2004-57518.

Gautschi, W. (1994), "Algorithm 726: Orthpol–a Package of Routines for Generating Orthogonal Polynomials and Gauss-Type Quadrature Rules," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 20(1), pp. 21-62.

Giovagnoli, A. and Romano, D. (2008), "Robust Design Via Simulation Experiments: A Modified Dual Response Surface Approach," *Quality & Reliability Engineering International*, Vol. 24, pp. 401-16.

Giunta, A. A., Balabanov, V., Haim, D., Grossman, B., Mason, W. H., Watson, L. T. and Haftka, R. T. (1997), "Aircraft Multidisciplinary Design Optimization Using Design of Experiments Theory and Response Surface Modelling," *Aeronautical Journal*, Vol. 101(1008), pp. 347-56.

Hayes, A. M., Wang, A., Dempsey, B. M. and McDowell, D. L. (2004), "Mechanics of Linear Cellular Alloys," *Mechanics of Materials*, Vol. 36(8), pp. 692-713.

Hernandez, G., Allen, J. K. and Mistree, F. (2001), "The Compromise Decision Support Problem: Modeling the Deviation Function as in Physical Programming," *Engineering Optimization*, Vol. 33(4), pp. 445–72-45–72.

Hernández, G. (2001). "Platform Design for Customizable Products as a Problem of Access in Geometric Space," *PhD dissertation*, G.W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA.

Hockenberry, J. R. (2000). "Evaluation of Uncertainty in Dynamic, Reduced-Order Power System Models," Massachusetts Institute of Technology

Hockenberry, J. R. and Lesieutre, B. C. (2004), "Evaluation of Uncertainty in Dynamic Simulations of Power System Models: The Probabilistic Collocation Method," *IEEE Transactions on Power Systems*, Vol. 19(3), pp. 1483–91-83–91.

Joanes, D. N. and Gill, C. A. (1998), "Comparing Measures of Sample Skewness and Kurtosis," *Journal of the Royal Statistical Society. Series D (The Statistician)*, Vol. 47(1), pp. 183-89.

Kemp, J. (2005), "Standard Deviation," *http://en.wikipedia.org/w/index.php?title=Standard_deviation&oldid=324120010 (accessed 11/2009)*, Wikipedia, The Free Encyclopedia.

Koch, P. N. (1997). "Hierarchical Modeling and Robust Synthesis for the Preliminary Design of Large Scale Complex Systems," *PhD Dissertation*, G. W. Woodruff

School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA, USA.

Koch, P. N., Allen, J. K., Mistree, F. and Mavris, D. (1997), "The Problem of Size in Robust Design," *ASME Advances in Design Automation*, Sacramento, CA, ASME, Paper Number: DETC97/DAC-3983.

Lewis, K. and Mistree, F. (1998), "Collaborative, Sequential, and Isolated Decisions in Design," *Journal of Mechanical Design*, Vol. 120(4), pp. 643-52.

Li, H. and Zhang, D. (2007), "Probabilistic Collocation Method for Flow in Porous Media: Comparisons with Other Stochastic Methods," *Water Resources Research*, Vol. 43, pp. W09409-W09.

Li, W. and Wu, C. F. J. (1999), "An Integrated Method of Parameter Design and Tolerance Design," *Quality Engineering*, Vol. 11(3), pp. 417 - 25.

Lin, Y., Krishnapur, K., Allen, J. K. and Mistree, F. (1999), "Robust Design: Goal Formulations and a Comparison of Metamodeling Methods," *1999 ASME Design Engineering Technical Conferences*, Las Vegas, Nevada, ASME, DETC99/DAC-8608.

Lin, Y. (2004). "An Efficient Robust Concept Exploration Method and Sequential Exploratory Experimental Design," *PhD Dissertation*, The George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA.

Liu, J. S. (2001), *Monte Carlo Strategies in Scientific Computing*, Springer, New York.

Lu, T. J. (1999), "Heat Transfer Efficiency of Metal Honeycombs," *International Journal of Heat and Mass Transfer*, Vol. 42(11), pp. 2031-40.

Marston, M., Allen, J. K. and Mistree, F. (2000), "The Decision Support Problem Technique: Integrating Descriptive and Normative Approaches in Decision Based Design," *Journal of Engineering Valuation and Cost Analysis*, Vol. 3(2), pp. 107-29.

Mistree, F., Hughes, O. F. and Phuoc, H. B. (1981), "An Optimization Method for the Design of Large, Highly Constrained Complex Systems," *Engineering Optimization*, Vol. 5(3), pp. 179–97.

Mistree, F., Smith, W. F., Bras, B. A., Allen, J. K. and Muster, D. (1990), "Decision-Based Design: A Contemporary Paradigm for Ship Design," *Transactions, Society of Naval Architects and Marine Engineers*, Vol. 98, pp. 565-97.

Mistree, F., Hughes, O. F. and Bras, B. A. (1993), "The Compromise Decision Support Problem and the Adaptive Linear Programming Algorithm," In M. P. Kamat (Ed.

Structural Optimization: Status and Promise (pp. 247-86), Washington, D.C., AIAA.

Mistree, F., Seepersad, C. C., Dempsey, B. M. and McDowell, D. L. (2002), "Robust Concept Exploration Methods in Materials Design," *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, GA.

Muster, D. and Mistree, F. (1988), "The Decision Support Problem Technique in Engineering Design," *International Journal of Applied Engineering Education*, Vol. 4(1), pp. 23-33.

Myers, R. H. and Montgomery, D. C. (1995), *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, Wiley, New York.

Nair, V. N. (1992), "Taguchi's Parameter Design: A Panel Discussion," *Technometrics*, Vol. 34, pp. 127-61.

Otto, K. N. and Antonsson, E. K. (1993), "Extensions to the Taguchi Method of Product Design," *ASME Journal of Mechanical Design*, Vol. 115(1), pp. 5-13.

Pahl, G. and Beitz, W. (1996), *Engineering Design - a Systematic Approach*, Springer, Berlin, Heidelberg, New York.

Park, G. J., Lee, T. H., Lee, K. H. and Hwang, K. H. (2006), "Robust Design: An Overview," *AIAA Journal*, Vol. 44(1), pp. 181–91.

Parkinson, A., Sorenson, C. and Pourhassen, N. (1993), "A General Approach for Robust Optimal Design," *ASME Journal of Mechanical Design*, Vol. 115(1), pp. 74-80.

Pedersen, K., Emblemsvag, J., Bailey, R., Allen, J. and Mistree, F. (2000), "The 'Validation Square' - Validating Design Methods," *ASME Design Theory and Methodology Conference*, New York.

Pederson, K., Emblemsvag, J., Allen, J. K. and Mistree, F. (2000), "Validating Design Methods and Research - the Validation Square," *ASME DETC, Design Theory and Methodology Conference*, Baltimore, MA, USA.

Phadke, M. S. (1989), *Quality Engineering Using Robust Design*, Prentice Hall, Englewood Cliffs, NJ.

Roark, R. J. and Young, W. C. (1975), *Formulas for Stress and Strain*, McGraw-Hill.

Sandgren, E. (1990), "Nonlinear Integer and Discrete Programming in Mechanical Design Optimization," *Journal of Mechanical Design*, Vol. 112(2), pp. 223-29.

Santner, T. J., Williams, B. J. and Notz, W. I. (2003), *The Design and Analysis of Computer Experiments*, Springer, New York.

Seepersad, C. C. (2001). "The Utility-Based Compromise Decision Support Problem with Applications in Product Platform Design," *MS Thesis*, The G.W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA.

Seepersad, C. C. (2004). "A Robust Topological Preliminary Design Exploration Method with Materials Design Applications," *PhD Dissertation*, G. W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA, USA.

Seepersad, C. C., Dempsey, B. M., Allen, J. K., Mistree, F. and McDowell, D. L. (2004), "Design of Multifunctional Honeycomb Materials," *AIAA Journal*, Vol. 42(5), pp. 1025–33-25–33.

Seepersad, C. C., Allen, J. K., McDowell, D. L. and Mistree, F. (2006a), "Robust Design of Cellular Materials with Topological and Dimensional Imperfections," *Journal of Mechanical Engineering*, Vol. 128(November), pp. 1285-97.

Seepersad, C. C., Pedersen, K., Emblemsvåg, J., Bailey, R., Allen, J. K. and Mistree, F. (2006b), "The Validation Square: How Does One Verify and Validate a Design Method?," In K. E. Lewis, W. Chen and L. C. Schmidt (Eds.), *Decision Making in Engineering Design* (pp. 303-14), New York, NY, ASME Press.

Shoemaker, A. C., Tsui, K.-L. and Wu, C. F. J. (1991), "Economical Experimentation Methods for Robust Design," *Technometrics*, Vol. 33, pp. 415-27.

Simpson, T. W., Chen, W., Allen, J. K. and Mistree, F. (1996), "Conceptual Design of a Family of Products through the Use of the Robust Concept Exploration Method," *AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Bellevue, WA.

Simpson, T. W. (1998). "A Concept Exploration Method for Product Family Design," *PhD Dissertation*, The G.W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA.

Simpson, T. W., Chen, W., Allen, J. K. and Mistree, F. (1999a), "Use of the Robust Concept Exploration Method to Facilitate the Design of a Family of Products," In U. Roy, J. M. Usher and H. R. Parsaei (Eds.), *Simultaneous Engineering: Methodologies and Applications* (pp. 247-78), Amsterdam, The Netherlands, Gordon and Breach Science Publishers.

Simpson, T. W., Maier, J. R. A. and Mistree, F. (1999b), "A Product Platform Concept Exploration Method for Product Family Design," *ASME Design Theory and Methodology Conference*, Las Vegas, NV, USA.

Simpson, T. W., Maier, J. R. A. and Mistree, F. (2001a), "Product Platform Design: Method and Application," *Research in Engineering Design*, Vol. 13(1), pp. 2-22.

Simpson, T. W., Peplinski, J. D., Koch, P. N. and Allen, J. K. (2001b), "Metamodels for Computer-Based Engineering Design: Survey and Recommendations," *Engineering with Computers*, Vol. 17, pp. 129-50.

Suh, N. P. (2001), *Axiomatic Design: Advances and Applications*, Oxford University Press.

Sundaresan, S., Ishii, K. and Houser, D. R. (1995), "A Robust Optimization Procedure with Variations in Design Variables and Constraints," *Engineering Optimization*, Vol. 24, pp. 101-17.

Taguchi, G. (1986), *Introduction to Quality Engineering : Designing Quality into Products and Processes*, Asian Productivity Organization, Tokyo.

Taguchi, G. and Clausing, D. (1990), "Robust Quality," *Harvard Business Review*, Vol. Jan/Feb, pp. p.65-75.

Taguchi, G., Chowdhury, S. and Taguchi, S. (2000), *Robust Engineering*, McGraw-Hill Professional, New York.

Tatang, M. A. (1995). "Direct Incorporation of Uncertainty in Chemical and Environmental Engineering Systems," Massachusetts Institute of Technology, Dept. of Chemical Engineering

Tatang, M. A., Pan, W., Prinn, R. G. and McRae, G. J. (1997), "An Efficient Method for Parametric Uncertainty Analysis of Numerical Geophysical Models," *Journal of Geophysical Research-Atmospheres*, Vol. 102(D18).

Tsui, K.-L. (1992), "An Overview of Taguchi Method and Newly Developed Statistical Methods for Robust Design," *IIE Transactions*, Vol. 24(5), pp. 44-57.

Tsui, K.-L. (1996), "A Critical Look at Taguchi's Modeling Approach for Robust Design," *Journal of Applied Statistics*, Vol. 23(1), pp. 81-95.

Vadde, S., Allen, J. K. and Mistree, F. (1994), "The Bayesion Compromise Decision Support Problem for Multilevel Design Involving Uncertainty," *Transactions of the ASME*, Vol. 116, pp. 388-95.

Vining, G. G. and Myers, R. H. (1990), "Combining Taguchi and Response Surface Philosophies: A Dual Response Approach," *Journal of Quality Technology*, Vol. 22, pp. 38-45.

Vuchkov, I. N. and Boyadjieva, L. N. (2001), *Quality Improvement with Design of Experiments : A Response Surface Approach* (Vol. 7), Kluwer Academic Publishers, Dordrecht ; Boston.

Wasserman, L. (2004), *All of Statistics: A Concise Course in Statistical Inference*, Springer Verlag, New York, NY.

Webster, M., Tatang, M. A. and McRae, G. J. (1996), "Application of the Probabilistic Collocation Method for an Uncertainty Analysis of a Simple Ocean Model," *Joint Program on the Science and Policy of Global Change, MIT, Cambridge, MA, Technical Report*, Vol. 4.

Webster, M. D. (1997), "Uncertainty in Future Carbon Emissions: A Preliminary Exploration," *Joint Program on the Science and Policy of Global Change, MIT, Cambridge, MA, Technical Report*, Vol. 30.

Welch, W. J., Yu, T.-K., Kang, S. M. and Sacks, J. (1990), "Computer Experiments for Quality Control by Parameter Design," *Journal of Quality Technology*, Vol. 22(1), pp. 15-22.

Wu, C. F. J. and Hamada, M. (2000), *Experiments: Planning, Analysis and Parameter Design Optimization*, John Wiley and Sons, New York.

Young, I. T. (1977), "Proof without Prejudice: Use of the Kolmogorov-Smirnov Test for the Analysis of Histograms from Flow Systems and Other Sources," *Journal of Histochemistry and Cytochemistry*, Vol. 25(7), pp. 935-35.

Zhou, Q.-J., Allen, J. K. and Mistree, F. (1992), "Decisions under Uncertainty: The Fuzzy Compromise Decision Support Problem," *Engineering Optimization*, Vol. 20, pp. 21-43.