

LINKÖPING STUDIES IN SCIENCE AND TECHNOLOGY  
THESIS NO 1556

# **Design and Optimization under Uncertainties**

A Simulation and Surrogate Model Based Approach

**Johan Persson**



**Linköping University**  
**INSTITUTE OF TECHNOLOGY**

DIVISION OF MACHINE DESIGN  
DEPARTMENT OF MANAGEMENT AND ENGINEERING  
LINKÖPINGS UNIVERSITET  
SE-581 83 LINKÖPING, SWEDEN

LINKÖPING 2012

ISBN 978-91-7519-753-1  
ISSN 0280-7971

Copyright © October 2012 by Johan Persson  
Department of Management and Engineering  
Linköpings universitet  
SE-581 83 Linköping, Sweden

Printed in Sweden by LiU-Tryck Linköping, 2012

## To Ingrid and Lars

*A heart is not judged by how much you love; but by how much you are loved by  
others.*

*The Wizard of Oz, 1939*

*In this world nothing can be said to be certain, except death and taxes*  
Benjamin Franklin

# ABSTRACT

---

This thesis deals with development of complex products via modeling and simulation, and especially the use of surrogate models to decrease the computational efforts when probabilistic optimizations are performed. Many methods that can be used to perform probabilistic optimizations exist and this thesis strives to present and demonstrate the capabilities of a few of them. Hopefully, this information can be helpful for someone who wants to choose a method.

Knowledge about several different topics is required to perform a probabilistic optimization. First, it is necessary to incorporate the probabilistic behavior into the analysis by estimating how the uncertainties and variations in the model and its parameters are affecting the performance of the system. The focus in this thesis is on sampling based methods to estimate these probabilities. Secondly, an optimization algorithm should be chosen so that the computer can search for and present an optimal solution automatically.

The probabilistic optimization process can be computationally demanding since numerous simulations of the model are performed each time the value of the objective function is estimated. It is therefore desirable to speed up the process by incorporating computationally effective surrogate models. This is especially important if the simulated model is computationally demanding on its own, e.g. a finite element model with many nodes.

Each of these topics is presented in its own chapter of this thesis. A few methods are presented and their performances demonstrated for each topic.

Surrogate models can also be used to improve the performances of optimization algorithms when the desire is to optimize computationally expensive objective functions. With this in mind, efforts have been made to improve the Complex-RF optimization algorithm. A modified algorithm is presented in this thesis and the main difference is that it creates and utilizes surrogate models iteratively during the optimization process. The modified algorithm is compared with Complex-RF and is demonstrated to be superior for computationally expensive models.

*We make a living by what we get, but we make a life by what we give.*  
Winston Churchill

# ACKNOWLEDGEMENTS

---

The research presented in this thesis has been performed at the Division of Machine Design at Linköping University. First and foremost, I would like to thank my supervisor Professor Johan Ölvander for all support and guidance I have received during this journey. You have always had patience with all my notions and discussions while guiding me forward.

The research has received funding from the European Community's Seventh Framework Programme under grant agreement no. 234344 ([www.crescendo-fp7.eu](http://www.crescendo-fp7.eu)), which I hereby acknowledge. I would like to thank all participants in the CRESCENDO project for the cooperation, and especially Sören Steinkellner at SAAB Aeronautics. Our common endeavor has finally reached its end.

Furthermore, I would like to thank all my colleagues at the university for the support I have received and the fruitful discussions we have had. In truth, so many of you have supported me that I will not mention any names due to fear of forgetting to mention anyone. You make the university a place I walk to with a smile on my face.

Finally, I would like to thank my friends and family for always being there for me. I could not have done this without you.

*A reader lives a thousand lives before he dies. The man who never reads live only one.*

George R.R. Martin, *A Dance with Dragons*, 2011.



# PAPERS

---

This thesis is based on the following four appended papers, which will be referred to by their Roman numerals. The papers are printed in their originally published state except for some changes in format and the correction of some minor errata.

In paper [I],[III] and [IV] the first author is the main author, responsible for the work presented, with additional support from other co-authors. In paper [II], the first author is responsible for most of the work presented, whereas the second author is responsible for the comparison of the surrogate models.

- [I] Persson, J.A., Ölvander, J., "*Comparison of Sampling Methods for a Dynamic Pressure Regulator*", Proceedings of the 49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Paper no AIAA-2011-1205, Orlando, Florida, Jan. 4-7, 2011.
- [II] Tarkian, M., Persson, J. A., Ölvander, J., and Feng, X., "*Multidisciplinary Design Optimization of Modular Industrial Robots*", Accepted for publication in Journal of Mechanical Design, Nov, 2012.
- [III] Persson, J.A., Ölvander, J., "*Comparison of Different Uses of Metamodels for Robust Design Optimization*", Accepted for publication at AIAA Annual Science Meeting, Grapevine, Texas, Jan 7-10, 2013.
- [IV] Persson, J.A., Ölvander, J., "*Optimization of the Complex-RFM Optimization Algorithm*", In manuscript.

The following publications are not appended to this thesis, but constitute an important part of the background.

- [V] Persson, J.A., Ölvander, J., "*A Modified Complex Algorithm Applied to Robust Design Optimization*", Proceedings of the 13th AIAA Non-Deterministic Approaches Conference, Paper no AIAA-2011-2095, Denver, Colorado, Apr. 4-7, 2011.
  
- [VI] Tarkian, M., Persson, J. A., Ölvander, J., and Feng, X., "*Multidisciplinary Design Optimization of Modular Industrial Robots*", Proceedings of the ASME 2011 International Design Engineering Technical Conferences & Computers and information in Engineering Conference, 2011, pp. 1-10.

# CONTENTS

---

<b>1</b>	<b>INTRODUCTION .....</b>	<b>15</b>
<b>2</b>	<b>HANDLING UNCERTAINTIES IN PRODUCT DESIGN.....</b>	<b>21</b>
2.1	Statistical Entities	21
2.1.1	Standard Distributions .....	22
2.2	Sampling Based Methods for Propagating Uncertainties	25
2.2.1	Monte Carlo Simulation .....	25
2.2.2	Latin Hypercube Sampling .....	25
<b>3</b>	<b>OPTIMIZATION ALGORITHMS .....</b>	<b>27</b>
3.1	Complex-RF	28
3.2	Fmincon	30
3.3	Genetic Algorithms	30
3.4	Optimization Algorithms which uses Surrogate Models	32
3.5	Reducing the dimensionality	32
3.5.1	Screening .....	32
3.5.2	Local Sensitivity Analysis .....	33
3.5.3	Surrogate Model Parameters .....	34
<b>4</b>	<b>SURROGATE MODELS.....</b>	<b>35</b>
4.1	Fitting of Surrogate Models	35
4.2	Polynomial Response Surfaces	36
4.3	Kriging	37
4.4	Radial Basis Functions	38
4.5	Determining the Accuracy of the Surrogate Models	39
4.5.1	Leave-one-out method.....	39
4.5.2	Precision measurements .....	39
<b>5</b>	<b>PROBABILISTIC OPTIMIZATION.....</b>	<b>41</b>
5.1	Robust Design Optimization	42
5.2	Reliability Based Design Optimization	42
<b>6</b>	<b>INDUSTRIAL APPLICATIONS.....</b>	<b>47</b>

6.1	Dynamic Pressure Regulator	47
6.2	Industrial Robot	49
6.3	Electric Motorcycle	50
<b>7</b>	<b>COMPARISON OF SAMPLING BASED METHODS.....</b>	<b>53</b>
<b>8</b>	<b>COMPARISON OF SURROGATE MODELS .....</b>	<b>57</b>
<b>9</b>	<b>PERFORMANCE INDEX FOR OPTIMIZATIONS .....</b>	<b>61</b>
<b>10</b>	<b>ROBUST DESIGN OPTIMIZATION.....</b>	<b>63</b>
<b>11</b>	<b>MODIFICATION OF THE COMPLEX ALGORITHM.....</b>	<b>69</b>
<b>12</b>	<b>DISCUSSION.....</b>	<b>75</b>

**PART I**

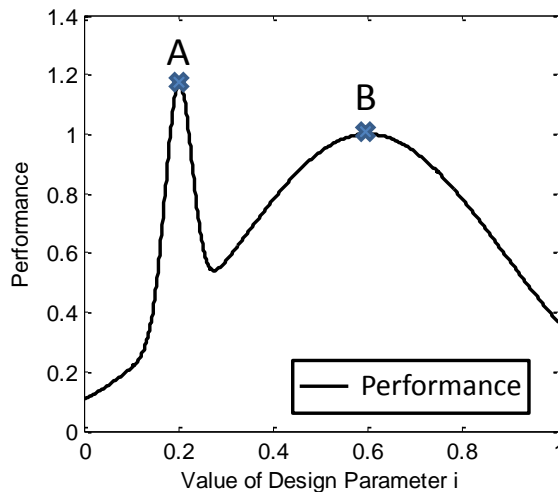
**INTRODUCTION**

*Why can't people just sit and read books and be nice to each other?*  
David Baldacci, *The Camel Club*, 2005.

# INTRODUCTION

---

In real life, variations and uncertainties are present almost everywhere and it may be important to estimate how these uncertainties and variations affect the performance of a product [1]. Imagine that the curve that is shown in Figure 1 represents the performance of a product as a function of a design parameter  $i$ . The point denoted A is the optimal design of the product if it is subject to almost no uncertainties. But point B would be a preferable design if the uncertainties and/or variations in  $i$  are large, since a small variation of  $i$  only affects the performance marginally.



**Figure 1.** A figure that displays a robust and a deterministic optimum.

During the development of new products it is therefore desirable to investigate the variations and uncertainties that affect the product and their impact on the

performance of the product. These phenomena may be estimated by performing experiments, but it is also possible to create a model of the product and perform simulations.

The product developers must be able to trust the results of a simulation of a model for simulations to be useful and therefore a model needs to be verified and validated [2],[3]. But modeling and simulations have several advantages compared to building and evaluating physical prototypes anyway [4]. Ideally, it is both cheaper and faster to perform simulations since a model may be simulated several times with different parameter settings once it has been developed and its results are deemed credible. Meanwhile, a new physical prototype may need to be manufactured for each unique design parameter setting which means that precious manufacturing machines are occupied. Additionally, some physical experiments may be hazardous or infeasible to perform.

When a model has been verified and validated, it is tempting to perform an optimization to let the computer suggest design parameters which lead to a product with optimal performance. Consequently the computer will perform numerous simulations of the model until an optimal solution or the maximum number of model simulations is reached. If the model is computationally expensive this may result in unrealistically long wall-clock times. One remedy to this problem is to replace the original models with computationally efficient surrogate models [5]-[7].

To investigate how robust the performance of a product is or the probability of success, the computational effort is increased compared to a deterministic analysis [8]. This further increases the desire to replace computationally demanding models with surrogate models.

To perform a probabilistic optimization efficiently, knowledge about estimating uncertainties, optimization algorithms and surrogate models are needed. Many different methods have been proposed and this thesis strives to give an idea of the capabilities and performances of several of them. This information can hopefully be of some guidance to someone who is interested in performing probabilistic optimization of complex products.

Complex-RF is an optimization algorithm which has been used for optimizing complex system models by several authors [9]-[11]. Another aim with this thesis is to investigate whether it is possible to improve the performance of Complex-RF by modifying it so it creates and uses surrogate models iteratively during the optimization process.

Additionally, the methods which are presented in this thesis are used to analyze two industrial applications. One is a system model of an electric motorcycle implemented in Simulink [12] and the other is a dynamic pressure regulator model implemented in Dymola [13]. The benefits are twofold; the industrial applications are analyzed and the capabilities of the different methods presented. A geometric model of another industrial application, an industrial robot, is used to demonstrate the performances of different surrogate models.



This thesis is divided into three parts, with the introduction constituting the first. The second part contains the background and contributions by others that this research is based on. The third part presents the contributions found in the appended papers and discusses the contributions and possible future work.

*Surprising what you can dig out of books if you read long enough, isn't it?*  
Robert Jordan, *The Shadow Rising*, 1991

**PART II**

**FRAME OF  
REFERENCE**

*There are three kinds of lies: lies, damned lies, and statistics*  
Mark Twain

# 2

## HANDLING UNCERTAINTIES IN PRODUCT DESIGN

---

It is desirable to estimate how variations and uncertainties affect the performance of a product to limit the risk of undesirable effects [14]. This may be estimated by performing an uncertainty analysis, but first the variations and uncertainties need to be estimated. This may be done by carrying out experiments, using engineering intuition or experience. It is convenient to describe the behavior of the uncertainties with statistical entities so they may be described by just a few parameters. This chapter is divided into two parts. The first part briefly describes basic statistics that is used in this thesis, whereas the second part presents a few methods that can be used to estimate the resulting uncertainties and variations in the product performance.

### 2.1 STATISTICAL ENTITIES

When a probabilistic analysis of a product is made, it is desirable to estimate the expected value and spread of its performance. These may be estimated as the sample mean and standard deviation respectively.

The mean of a population is calculated as the sum of the values in the whole population divided by the number of individuals. However, it is impractical to calculate the values of each individual in a large population. Instead, samples of the population are drawn and a mean value for the whole population is estimated from the samples. An unbiased estimator of the sample mean is

calculated according to Eq. (1), where  $N$  is the number of samples and  $y_i$  the value of each sample [15].

$$\mu = \frac{1}{N} \sum_{i=1}^N y_i \quad (1)$$

The standard deviation may be estimated in two different ways, depending on the context. Equation (2) may be used to estimate the standard deviation of  $N$  samples, whereas Eq. (3) may be used as an unbiased estimator of the standard deviation of the underlying population by using the  $N$  samples. The difference between the two estimations is negligible for large number of samples.

$$s_N = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2} \quad (2)$$

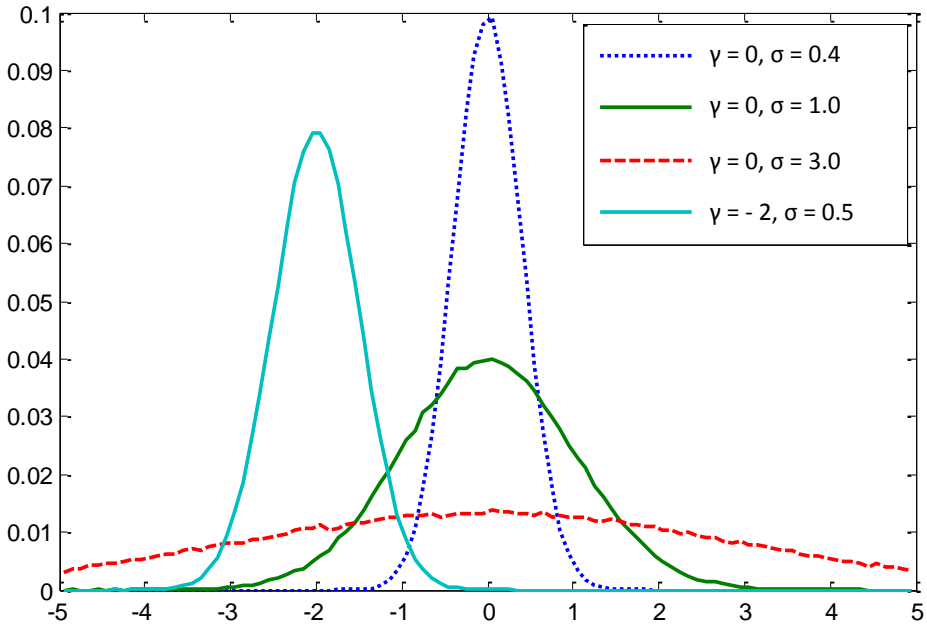
$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (y_i - \bar{y})^2} \quad (3)$$

A similar measure of the spread is the variance, which is the standard deviation squared. It is often more convenient to use the standard deviation instead of the variance for describing the spread of a population since it has the same unit as the mean value. A drawback with using the variance or standard deviation as a measure of the spread is that they are sensitive to outliers, since they use the difference squared in the calculations.

### 2.1.1 STANDARD DISTRIBUTIONS

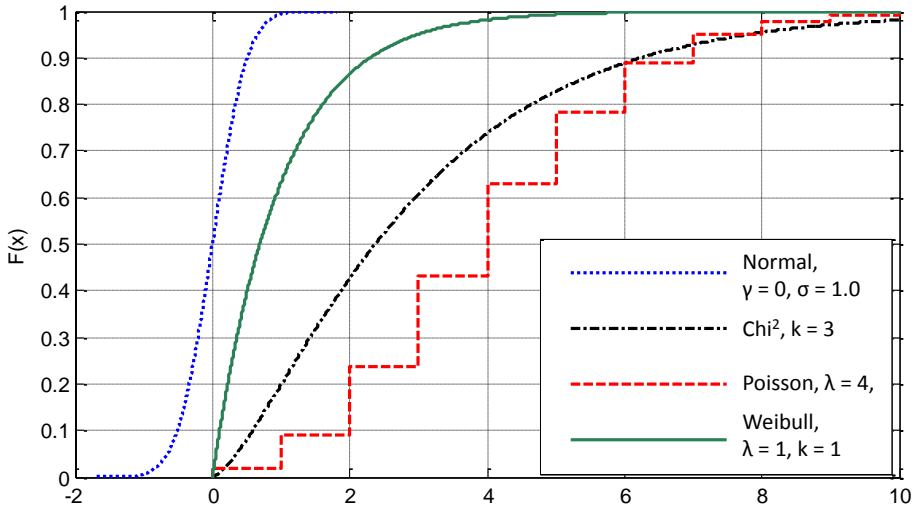
Standard distributions are functions which describes the values of a data set with just a few parameters. It is therefore desirable to fit a data set gained from experiments to a standard distribution.

A widely used distribution is the Normal or Gaussian distribution. It resembles a peak with the center located at the sample mean. The slopes of the peak are determined by the standard deviation and are steeper for lower values of the standard deviation. Figure 2 displays four different curves for the Normal distributions, with different values for the mean value and standard deviation.



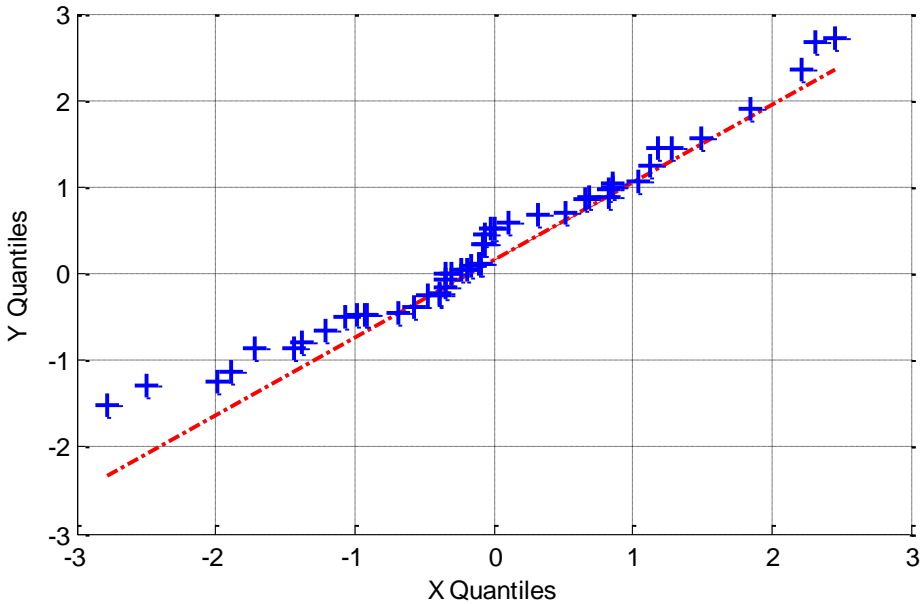
**Figure 2.** A plot of four Gaussian distributions with different mean values and standard deviations.

There exist numerous other standard distributions and the cumulative distributions of a few of them are shown in Figure 3, together with a cumulative distribution for a Normal distribution.



**Figure 3.** A plot of the cumulative distributions for a Normal, a Chi-squared, a Poisson and a Weibull distribution.

It might be difficult beforehand to make an assumption regarding which standard distribution that reanimates the data set best. One solution is to fit the data to several promising standard distributions and see which distribution that results in the best fit. A graphical comparison can be made by drawing a QQ-plot as shown in Figure 4. The data set is sorted in ascending order and an equal amount of samples are drawn from the standard distribution with a probability interval of  $1/N$  between each sample. These values are then paired together with the data set value on the x-axis and the corresponding standard distribution value on the y-axis. If the distributions agree well, the paired values should lie on a straight line.



**Figure 4.** A QQ-plot that compares the X and Y distributions.

It is probably more desirable to receive a numerical value for how well the two data sets agree. The Mean Squared Error (MSE), whose equation is shown in Eq. (4), estimates how well two data sets agree by summarizing the squared difference between each pair. A low MSE indicates that the two data sets agree well.

$$MSE = \frac{1}{N} \sum_{i=1}^N (Y_i - X_i)^2 \quad (4)$$

A drawback with the MSE is that the square makes it sensitive to outliers. It is also possible to use the measures which are mentioned in section 4.5.2.



## 2.2 SAMPLING BASED METHODS FOR PROPAGATING UNCERTAINTIES

The methods that are used in this thesis to estimate how uncertainties and variations in the model and its parameters affect the results are sampling based methods. Others methods exist, e.g. First Order Methods, Polynomial Chaos Expansions, but are not presented in this thesis [16],[17].

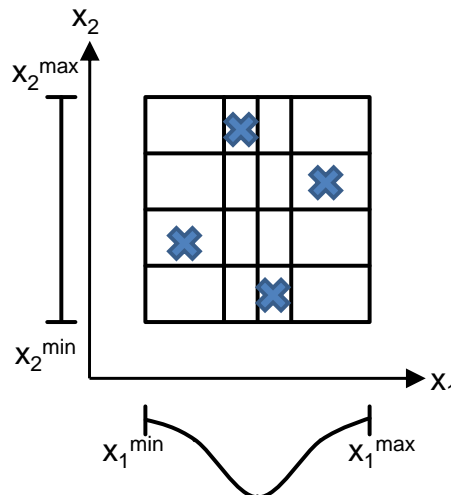
### 2.2.1 MONTE CARLO SIMULATION

The most intuitive sampling based method is probably the Monte Carlo Simulation (MCS). An MCS is carried out by performing experiments with the same settings and by the law of large numbers the results will converge as the number of experiments increase. One example is a dice where approximately one sixth of the throws will be ones if enough throws are made.

An MCS may unfortunately need thousands of samples to converge, making it an unrealistic option for time demanding experiments or computationally expensive models [18]. More efficient sampling methods have been developed and in this thesis Latin Hypercube Sampling (LHS) has been used.

### 2.2.2 LATIN HYPERCUBE SAMPLING

LHS is an effective sampling based method, where the user begins by specifying the desired number of samples,  $n$  [19]. Then, LHS divides the probability distributions into  $n$  intervals of equal probability as shown in Figure 5 for four samples and two parameters. The first parameter follows a normal distribution, whereas the second parameter is uniformly distributed.



**Figure 5.** A picture that shows an LHS consisting of four samples, for two variables. The first variable is normally distributed whereas the other is uniform.

One benefit with using LHS is that it operates in the same manner regardless of the number of uncertainty sources. A major drawback with LHS compared to MCS is that it is difficult to add new samples to the data set. For MCS it is possible to add samples until a desired accuracy is reached. For LHS it is only possible to add samples so that the total number of samples in the LHS is a multiple of the original sampling scheme.

*There are three roads to ruin; women, gambling and technicians. The most pleasant is with women, the quickest is with gambling, but the surest is with technicians.*

Georges Pompidou

# 3

## OPTIMIZATION ALGORITHMS

---

The purpose of an optimization algorithm is to let the computer automatically explore the design space of a mathematical problem and present an optimal solution to it. The optimization problem may be formulated as in Eq. (5), where a mathematical function,  $f$ , should be minimized while the  $p$  inequality and  $r$  equality constraints are fulfilled [9].

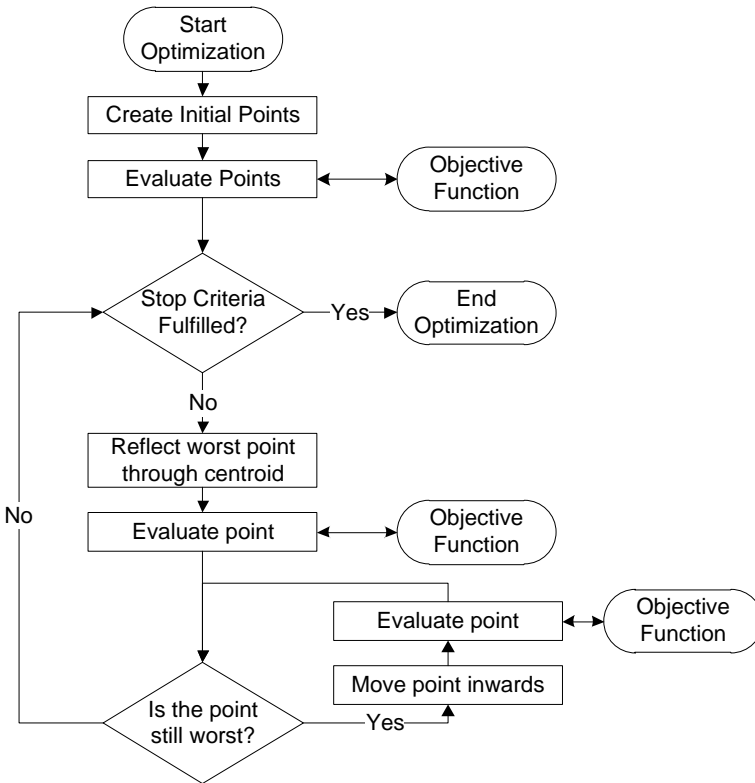
$$\begin{aligned} \min_x & f(x) \\ \text{s.t.} & \\ & g_j(x) \leq 0, \quad j = 1, \dots, p \\ & h_k(x) = 0, \quad k = 1, \dots, r \\ & x_i^{low} \leq x_i \leq x_i^{up} \end{aligned} \tag{5}$$

Numerous optimization algorithms have been proposed and some of them have the ability to handle constrained optimization problems. It is possible to use an optimization algorithm which is unable to handle constraints even though the optimization problem is constrained, for example by adding the constraints to the objective function as penalty functions. If a solution violates any of the constraints, a penalty is added to the objective function value, which worsens the suggested solution drastically. This approach is shown in Eq. (6), where the coefficients  $a$ ,  $b$  and  $c$  determine how large the penalty should be when the constraints are violated.

$$\begin{aligned} \min_x & f(x) + a(p(x)^b + h(x)^c) \\ p(x) &= \max(0, g(x)) \\ x_i^{low} &\leq x_i \leq x_i^{up} \end{aligned} \tag{6}$$

### 3.1 COMPLEX-RF

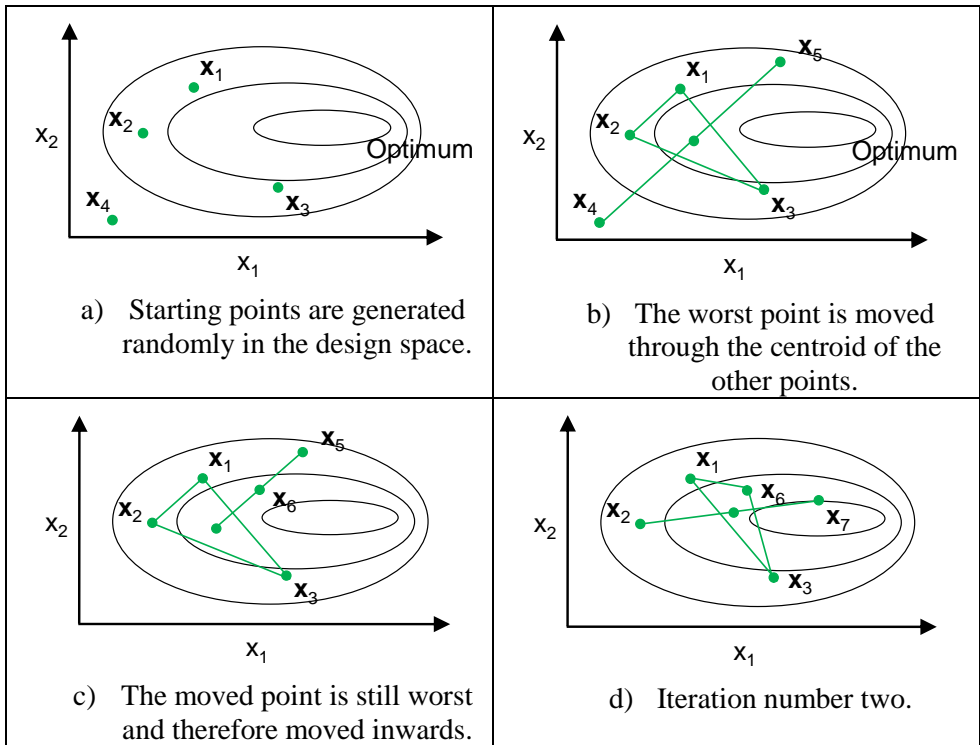
Proposed by Box [20], the Complex algorithm is an optimization algorithm developed from the Neelder Mead Simplex algorithm [21]. The algorithm converges by iteratively moving  $k$  points in the design space.



**Figure 6.** A chart over the workflow of the Complex-RF optimization algorithm.

A schematic workflow is shown in Figure 6 and the algorithm begins by randomly spreading the  $k$  points in the design space [22]. These points are spanning the Complex and their corresponding objective function values are found by calling the objective function. The point with the worst objective function value is moved along an imaginary line between itself and the centroid of the other points, to the other side of the Complex. If the point still is the

worst, it is moved towards the centroid of the other points until it no longer is the point with the worst objective function value. Then, the point which now is the worst is moved in a similar fashion along an imaginary line to the other side of the centroid of the other points. This process is shown in Figure 7 and progresses until the difference in the function values or the distance in the design space between the points is small enough. Additionally, the algorithm stops if the maximum number of objective function calls is reached, to avoid unnecessary long optimization times.



**Figure 7.** Four schematic pictures which displays the working operations of Complex-RF.

More advanced versions of the Complex algorithm have been suggested by adding randomization and forgetting factors [23]. The randomization factor enables a moving point to be placed randomly beside the imaginary line between itself and the centroid of the other points. This increases the robustness of the algorithm by making it improbable that it gets stuck and just repeats its moves along a few lines until the maximum number of function evaluations is reached.

The forgetting factor penalizes old points in the Complex by a small factor each time a point is moved. Consequently, an optimization with a high forgetting factor leads to a Complex which predominantly is spanned by recent

points. This is useful for handling problems with plateaus or when the objective function varies with time.

Since the starting points are spread randomly in the design space, different optimal solutions may be suggested by the algorithm when several optimizations are performed.

Several authors have chosen Complex-RF for optimizing engineering problems and thereby demonstrated its performance [9]-[11].

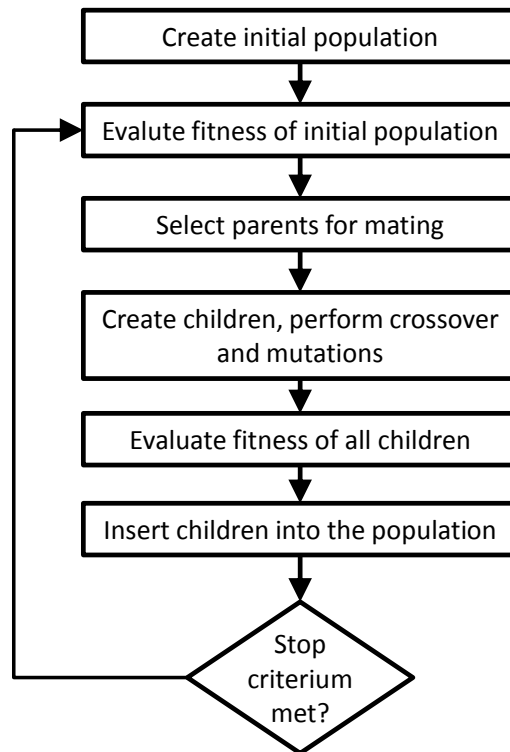
## 3.2 FMINCON

Fmincon is a collection of gradient-based optimization methods pre-implemented in the mathematical software MATLAB [24]. Unlike the Complex algorithm, a starting point must be specified by the user. Fmincon then uses the gradients at the starting point to choose a specific optimization method and move to a better point. If the gradients at the starting point are known, it is possible to give them as inputs to the algorithm. Otherwise, fmincon will try to approximate the gradients by finite differences.

Fmincon performs the same operations every time the same problem is optimized using the same starting point, meaning that the algorithm always suggests the same optimum. The gradient-based optimization method is excellent for solving unimodal problems where no local minimums exist, but makes fmincon highly starting point dependent for multi-modal problems.

## 3.3 GENETIC ALGORITHMS

A genetic algorithm reanimates the evolution seen in nature by implementation of survival of the fittest [25],[26]. A schematic workflow of a GA is shown in Figure 8 and it basically generates a predefined number of generations, consisting of a constant number of individuals. It starts by spreading an initial population randomly in the design space and each individual is given a fitness value by calling the objective function. The individuals with the best fitness values are selected for mating and a new generation with the same number of individuals is created. Since the best individuals are selected as parents, the next generation should be slightly better than the current. The GA progresses until a predefined number of generations is reached. It is possible to tune the optimization by for example changing the number of individuals in the population and the number of generations.



**Figure 8.** A schematic workflow for a simple genetic algorithm.

There exist more advanced versions of genetic algorithms which enable more options. One example is letting the best individuals survive to the next generation, meaning that good individuals may survive for several generations. Another example is mutation, where the individuals may change randomly. This is a way of avoiding local minimums, but a too high mutation factor prevents the algorithm from converging.

A drawback with genetic algorithms is that they usually converge quite slowly and require many objective function evaluations since the fitness values of all individuals in each generation needs to be calculated. Consequently, the number of objective function evaluations that is required to find the optimal solution is the number of individuals in each generation multiplied by the number of generations, and this number often tend to be quite high [15].

An advantage with genetic algorithms from a computational view is that it is possible to calculate the objective function values of all individuals in the present generation in parallel. If enough computers are available this means that the wall-clock time of an optimization may be reduced to the time it takes to evaluate one individual multiplied by the number of generations created.

### 3.4 OPTIMIZATION ALGORITHMS WHICH USES SURROGATE MODELS

The performances of several optimization algorithms for computationally expensive models have been improved by incorporating surrogate models in the optimization process. One example is the genetic algorithm presented by DuVigneau et al. [15], which uses a kriging surrogate model. The algorithm operates similar to a normal genetic algorithm with two exceptions.

1. Whenever the original objective function is called to calculate the objective function value, the values of both the variables and the objective function are stored in a database.
2. For generation two and onwards, a kriging model is created from the database and used to estimate the objective function values of all new individuals. Afterwards, the objective function values of the 30% most promising individuals are evaluated by calling the original objective function.

This means that many of the necessary objective function evaluations can be made by calling a surrogate model instead of the computationally expensive original objective function.

Another way to use surrogate models together with optimizations is described by Forrester et al. [27]. First, a surrogate model of the original model is created. Then an optimization algorithm is used to find the optimum. After a model call at the optimum, the surrogate model is updated to incorporate this point as well. Then a new optimization is performed on the surrogate model and the surrogate model updated again. This process continues until a stop criterion is met, e.g. that approximately the same point was suggested to be optimal consecutive times.

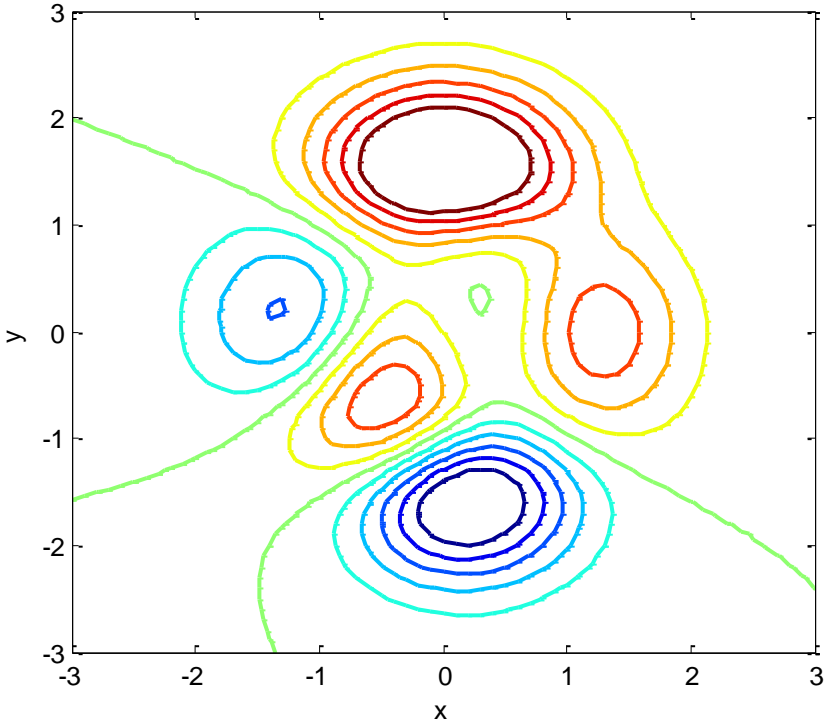
### 3.5 REDUCING THE DIMENSIONALITY

A problem with optimization of complex products is that the performance of the product usually depends on numerous variables. Fortunately, the different variables affect the performance to different degrees. It is therefore desirable to focus the effort on the variables which affect the product performance most and neglect the others. This means that it is useful to use a method to identify the most important variables, and fortunately a few have been suggested.

#### 3.5.1 SCREENING

A straight forward method to graphically analyze how the objective function varies is screening, where two variables at a time are varied between their lower and upper bounds. The values of the objective function are then plotted with the values of the first variable on the x-axis and the values of the second variable on the y-axis. One example of a contour plot of how the objective function varies is shown in Figure 9.





**Figure 9.** A contour plot of the Peaks function.

Two drawbacks with this method are that no numerical value for the importance of the variables is gained and that numerous plots need to be created if the interactions between each possible pair of variables are desired, since only two variables at a time can be displayed.

### 3.5.2 LOCAL SENSITIVITY ANALYSIS

A local sensitivity analysis is performed around a reference design point by estimating partial derivatives of how the objective function varies when each variable is altered [28]. This is shown in Eq. (7), where  $y_i$  denotes objective  $i$  and  $x_j$  stands for design variable  $j$ .

$$s = \frac{\partial y_i}{\partial x_j} = \frac{\Delta y_i}{\Delta x_j} \quad (7)$$

The variables which result in partial derivatives close to zero are the ones which influence the objective function least and may be disregarded for small design changes [29]. The different variables may have different units and therefore be

of different orders of magnitude. This affects the partial derivatives and makes a direct comparison unfair. To enable a comparison, the partial derivatives need to be normalized as shown in Eq. (8).

$$ns = \frac{\partial y_i}{\partial x_j} \frac{x_j}{y_i} \quad (8)$$

A drawback with this method is that the measures of the importance of the variables are local and depend on the chosen step lengths when the partial derivatives are estimated.

### 3.5.3 SURROGATE MODEL PARAMETERS

Another method to assess the importance of the variables is to create a surrogate model of the mathematical problem. Then the coefficients for the different variables can be compared with each other [27]. For a polynomial response surface as described in section 4.2, the first order coefficients describe the importance of each variable if the design variables are normalized to [-1,1] [30].

*But there's a world beyond what we can see and touch, and that world lives by its own laws. What may be impossible in this very ordinary world is very possible there, and sometimes the boundaries between the two worlds disappear, and then who can say what is possible and impossible?*

David Eddings, *Pawn of Prophecy*, 1982.

# 4

## SURROGATE MODELS

---

Surrogate models are also known as metamodels (MMs) and are numerical approximations of how an entity varies when the entities that affect it are varied. One application of surrogate models is to create a simplified model of a system with unknown behavior and to use the surrogate model to predict the system characteristics for a given set of system parameters values instead of performing experiments. Another application is to use surrogate models instead of computationally demanding models to speed up analyzing processes.

The main drawback when surrogate models are used is that estimation errors are introduced to the problem since a surrogate model is an approximation of the original model. But it is often necessary to use surrogate models anyway if a model simulation is computationally expensive.

The next sections describe how to fit a surrogate model to a system and a few different types of surrogate models – Polynomial Response Surfaces, Kriging and Radial Basis Functions. Even though other surrogate models have been tested in this research, they are not presented in the appended papers, and are therefore not included here.

### 4.1 FITTING OF SURROGATE MODELS

A surrogate model is created by performing a few experiments or simulations of the system which the surrogate model is intended to reanimate. Then the surrogate model is fitted to these samples, meaning that the accuracy of the estimations of the surrogate models generally is higher closer to the samples used to fit the surrogate model. Consequently, the placements of the samples used for fitting the surrogate models are of great importance and it has also been given a name – design of experiments (DoE) [6].

Numerous DoEs have been suggested and the interested reader is encouraged to read works by Myers et al. [31] and Simpson et al. [32] where

several methods are presented. In this research Latin Hypercube Sampling (LHS) has been used as DoE to fit the surrogate models. It has also been used by several other authors for similar purposes [4]. A benefit with using LHS as DoE is that it operates in the same manner regardless of the number of dimensions the surrogate model is intended to handle.

Another method is described by for example Forrester et al. [27]. It is suggested to fit a surrogate model according to a DoE, but to save some expensive model simulations for later. An infilling criterion is then used to decide which simulations that should be performed to improve the surrogate model. The criterion can for example be the optimum of the surrogate model or the point where the expected improvement is large.

## 4.2 POLYNOMIAL RESPONSE SURFACES

One of the most intuitive types of surrogate models is probably the polynomial response surface which tries to approximate a surrogate model as a polynomial of arbitrary degree. Eq. (9) shows an example of a polynomial response surface of degree two [6].

$$\hat{y} = \beta_0 + \sum_{i=1}^N \beta_i x_i + \sum_{i=1}^N \sum_{j=1}^N \beta_{ij} x_i x_j \quad (9)$$

This can also be written in matrix form as seen in Eq. (10).

$$\hat{y} = \mathbf{X}(x)\boldsymbol{\beta} \quad (10)$$

To be able to use this response surface, the coefficients  $\beta_i$  needs to be determined. Fortunately, it is possible to do this with a least square estimation.

To perform a least square estimation a matrix system is created as seen in Eq. (11). Each row in the matrix system corresponds to Eq. (10) for a certain sample. For example, the top row is Eq. (10) for the first sample that was drawn to fit the surrogate model.

$$\mathbf{X}\boldsymbol{\beta} = \mathbf{y} \quad (11)$$

This equation system is solved in a least square sense by performing the matrix operations seen in Eq. (12). A least square estimation finds the coefficients which minimize the sum of the errors squared for Eq. (11).

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'\mathbf{y} \quad (12)$$

At least as many samples as the number of coefficients need to be drawn to fit a polynomial, but it is recommended to oversample with 50% [33] or 100% [34].

The two major drawbacks with this surrogate model are that the characteristics that is reanimated needs to be able to be approximated as a polynomial; and the requirement of numerous samples if the interaction terms between the different system parameters are included in the response surface [6],[32].

### 4.3 KRIGING

Kriging is a surrogate modeling technique which originates from geostatistics, where it is desirable to estimate the appearances of the mineral fields from a few drill samples. This technique has been proven to be effective as surrogate model for simulation models of complex systems [16],[35].

Issaks and Srivastava have written a good introduction to Kriging and its capabilities where the mathematics is described in detail [36].

Kriging is an interpolating stochastic model which estimates global trends. When the model is used as a deterministic surrogate model, the expected value at the desired point is used as the approximate value at the point [35]. Kriging is ideally linear, unbiased and minimizes the model error variance.

A kriging model is a combination of a polynomial model and local deviations of the form seen in Eq. (13) [32],[34].

$$\hat{y} = \sum_{i=1}^k \beta_i f_i(x) + Z(x) \quad (13)$$

$\beta_i$  is the  $i^{\text{th}}$  regression coefficient, whereas  $f_i(x)$  is a function of the variables  $x$  and  $Z(x)$  a stochastic process with zero mean and a spatial correlation function given by Eq. (14).

$$\text{Cov}[Z(x_i), Z(x_j)] = \sigma^2 R(x_i, x_j) \quad (14)$$

$\sigma^2$  is the variance of the process and  $R(x_i, x_j)$  is a correlation function between  $x_i$  and  $x_j$ . The first term in Eq. (13) is an approximation of the trend of the model output, resembling a polynomial response surface, and is modified according to the problem [37]. Kriging is a stationary process, which means that it is suitable for problems where the function value is somewhat constant throughout the design space.

The correlation function includes unknown parameters and unfortunately it is time-consuming to determine the values of the parameters. This is an optimization problem in its own right and may be solved by using maximum likelihood estimators; see for example works by Martin and Simpson [35].

Unfortunately, the correlation matrix can become singular if the samples are placed too close to each other [38]. This needs to be handled if kriging models are fitted iteratively during an optimization.

## 4.4 RADIAL BASIS FUNCTIONS

RBFs are surrogate models where the value of a new point is interpolated from the values of the known points [15]. If the function values of  $n$  samples are known, the expression for interpolating the value of a new point is of the form seen in Eq. (15).

$$\hat{f}(x) = \sum_{i=1}^n \omega_i \phi_i(\|x - x_i\|) + \sum_{i=1}^k \beta_i f_i(x) \quad (15)$$

$\phi$  is a function of the Euclidian distance between the new point and the known point with index  $i$ . Several choices of distance functions have been suggested, and here the Gaussian function of the form seen in Eq. (16) is chosen.

$$\phi(r) = e^{-\frac{r^2}{s^2}} \quad (16)$$

The reason for choosing the Gaussian function is that Nakayama [39] suggests that the parameter  $s$  may be approximated according to Eq. (17).

$$s = d_{\max} (n \cdot m)^{-\frac{1}{m}} \quad (17)$$

$d_{\max}$  is the distance between the two points that are furthest away from each other,  $m$  is the number of variables and  $n$  is the number of samples.

The second term of Eq. (15) is a polynomial response surface of arbitrary degree and might be added if the overall behavior of the function is believed to follow a polynomial of a certain degree.

The weights,  $\omega$  and  $\beta$ , are determined by solving the equation system seen in Eq. (18).

$$\begin{bmatrix} \Phi & \mathbf{P} \\ \mathbf{P}^t & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\omega} \\ \boldsymbol{\beta} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f} \\ \mathbf{0} \end{Bmatrix} \quad (18)$$

$\Phi$  is the  $n \times n$  matrix containing the distance function values between all samples. The other matrixes are seen in Eq. 19 [6].

$$\mathbf{P} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \quad \boldsymbol{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_n \end{bmatrix} \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{bmatrix} \quad \mathbf{f} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{bmatrix} \quad (19)$$

Since  $s$  can be approximated with Eq. (17), the  $\Phi$ -matrix can be approximated as well. Fortunately, only the weights,  $\omega$  and  $\beta$ , remain to be determined. This can be done by performing a simple least square estimation. Eq. (12) shows how a least square estimation is performed on the equation system seen in Eq. (11).

A slightly lower accuracy of the surrogate model is probably achieved compared to when the parameters is obtained by optimization. The gain in speed should however make up for it in many cases.

## 4.5 DETERMINING THE ACCURACY OF THE SURROGATE MODELS

It is desirable to estimate the accuracy of the predictions from the surrogate models to realize how credible they are. This is ideally done by comparing the values predicted by the surrogate model with the ones from the original model or function. For non-interpolating surrogate models it is possible to use the samples that were used to fit the surrogate for the comparison. But these samples cannot be used to assess the accuracy of interpolating surrogate models. Additional samples, which have not been used to fit the interpolating surrogate model, need to be used for the comparison instead.

### 4.5.1 LEAVE-ONE-OUT METHOD

A common situation is that a maximum number of calls of the original function can be made to prevent the computational time from being too long. Instead of using all samples to fit a surrogate model, it might be more suitable to leave a few samples out of the fitting process. These samples can then be used to assess the precision of the surrogate model. This process can be made iteratively, leaving different samples out each time, and then the surrogate model which displays the best precision is used for the analysis.

### 4.5.2 PRECISION MEASUREMENTS

There exist several measurements of how accurate a surrogate model is. A common measure is the Normalized Root Mean Squared Error (NRSME), seen in Eq. (20). It is a measure of the overall accuracy and returned in the

percentage form since the errors are normalized. Here,  $y_i$  stands for the true value at point  $i$ , whereas  $\hat{y}_i$  is the corresponding predicted value.

$$NRSME = \sqrt{\frac{\sum_{i=1}^n \left( \frac{y_i - \hat{y}_i}{y_i} \right)^2}{n}} \quad (20)$$

Another measure of the global accuracy is R Square, seen in Eq. (21). A high value of R Square indicates that the overall accuracy is good.  $\bar{y}$  stands for the mean of the true values [6],[34].

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (21)$$

The Relative Average Absolute Error (RAAE) is measure which summarizes the errors between all estimations and their corresponding true values as seen in Eq. (22).  $\sigma$  stands for the standard deviation of the true values [38].

$$RAAE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n \cdot \sigma} \quad (22)$$

The Relative Maximum Absolute Error (RMAE) is a measure of the local accuracy and returns the maximum error normalized by the standard deviation of the true values [34]. This should be as low as possible and the mathematical expression for the estimation is shown in Eq. (23) [40].

$$RMAE = \frac{\max(|y_1 - \hat{y}_1|, |y_2 - \hat{y}_2|, \dots, |y_n - \hat{y}_n|)}{\sigma} \quad (23)$$

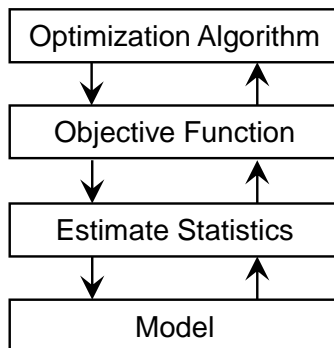


# 5

# PROBABILISTIC OPTIMIZATION

---

Probabilistic optimization can be divided into two categories – Robust Design Optimization (RDO) and Reliability Based Design Optimization (RBDO). For both of these optimizations, the effects of the uncertainties and variations on the performance of the product need to be estimated when two product designs are compared. This can be done by using a sampling method as shown in Figure 10. This might be extremely time-consuming for computationally expensive models, making it desirable to replace the models with faster evaluated surrogate models.



**Figure 10.** A schematic picture of the actions that are performed during a probabilistic optimization.

## 5.1 ROBUST DESIGN OPTIMIZATION

The purpose of RDO is to find a robust optimal design, i.e. an optimal design which is insensitive to uncertainties and variations [17],[18]. It is desirable to let the computer perform an optimization to find the design automatically and therefore the objective function needs to take both into account. This can be written as a multi-objective optimization problem as in Eq. (24), where the objective function is a linear combination of the mean value and standard deviation of the product performance [18],[34].

$$\begin{aligned} \min_x \quad & \alpha \cdot \frac{\mu[f(x)]}{\mu_0} + \beta \frac{\sigma[f(x)]}{\sigma_0} \\ \text{s.t.} \quad & g_j(x) \leq 0, j = 1, \dots, p \\ & h_k(x) = 0, k = 1, \dots, r \\ & x_i^{low} \leq x_i \leq x_i^{up} \end{aligned} \tag{24}$$

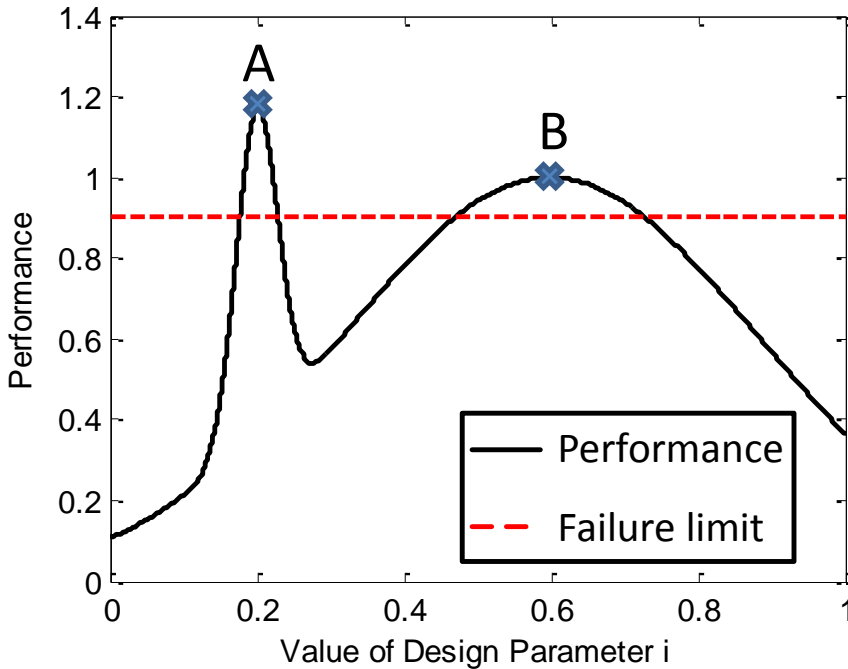
The two weights,  $\alpha$  and  $\beta$ , determine how important the expected performance and variation are with respect to each other. A high value of  $\alpha$  means that the expected performance is of importance, whereas a value of zero turns the optimization problem into a problem which focuses on minimizing the standard deviation exclusively.

Both the expected performance and its standard deviation may be normalized with the reference values  $\mu_0$  and  $\sigma_0$  to ensure that the expressions standing after  $\alpha$  and  $\beta$  are of the same order of magnitude [18]. These reference values are often estimated from a reference design point. This design point is usually either the initial design or somewhere close to the middle of the design space.

As the expected value of the performance and its variation is used in the objective function, they need to be estimated for each suggested design. This may be done using any method for estimating uncertainties, for example one of the methods presented in 2.2.

## 5.2 RELIABILITY BASED DESIGN OPTIMIZATION

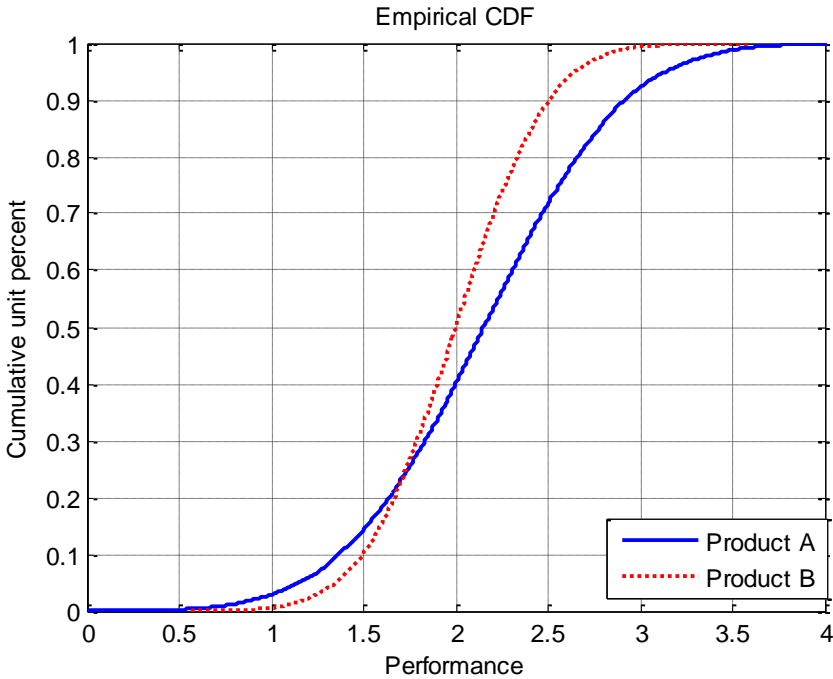
RBDO strives to find a design which minimizes the probability of failure. Consequently, the probability of failure needs to be estimated for each suggested design. The most intuitive method for estimating the probability of failure is probably to estimate the cumulative distribution of the performance by using a sampling based method. It makes it possible to calculate how many percent of the samples that result in inadequate performances.



**Figure 11.** A plot over the performance of a product as a function of the value of design parameter  $i$ . The dashed line displays the minimum required performance.

RBDO is exemplified in Figure 11 and Figure 12. As shown in Figure 11, Product A has a better performance than product B for the best units that are manufactured. Variations in the value of design parameter  $i$  may however lead to units which have inadequate performance. This probability of failure is higher for product A than product B since smaller variations from the expected design leads to an unsatisfactory unit.

It is also possible to use the cumulative distribution of the performance to optimize the performance of the  $j^{\text{th}}$  percentile of the product. One example is to perform an optimization which returns a design with as good performance as possible for the 80<sup>th</sup> percentile of the product. Ideally, this means that if one hundred units are manufactured and sorted in performance order, the 80<sup>th</sup> unit will have a better performance than the 80<sup>th</sup> unit that is manufactured with any other design parameters. This is exemplified in Figure 12, where product A has a better performance for most units, whereas product B has a better performance for the 21<sup>th</sup> worst unit. The 21<sup>th</sup> worst unit corresponds to the 80<sup>th</sup> best unit if 100<sup>th</sup> units are manufactured which means that product A has a better performance for its 80<sup>th</sup> percentile.



**Figure 12.** A plot which displays the cumulative distributions of the performance of the manufactured units for Product A and B.

*You can never know everything, and part of what you know is always wrong. Perhaps even the most important part. A portion of wisdom lies in knowing that. A portion of courage lies in going anyway.*  
Robert Jordan, *Winter's Heart*, 2000.

# **PART III**

# **RESULTS**



# 6

## INDUSTRIAL APPLICATIONS

Three different industrial applications are analyzed in the frames of this thesis and presented in this section. These are models of an aircraft system, an industrial robot and an electric motorcycle.

### 6.1 DYNAMIC PRESSURE REGULATOR

The studied aircraft system is a dynamic pressure regulator (DPR) which is shown in Figure 13 below.

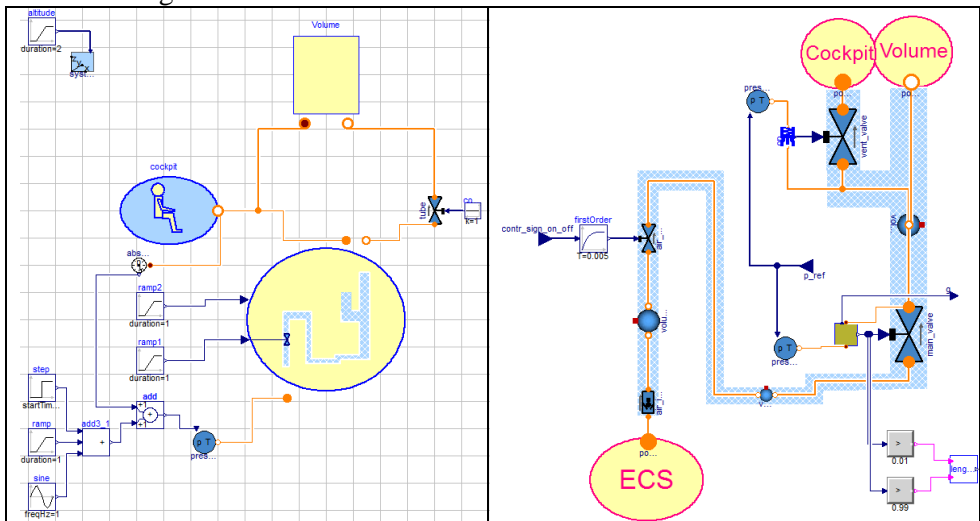


Figure 13. Two screenshots of the Dynamic Pressure Regulator modeled in Dymola.

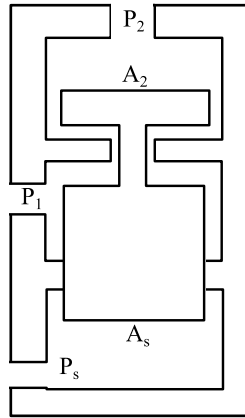
The system is modeled in the Modelica [41] based software Dymola [13] and its purpose is to regulate the air pressure that is delivered to an environmental control system (ECS). The most important functions of the DPR are to ensure that the ECS is filled fast enough and to the correct pressure level. Therefore, the most important system characteristics are the times it takes to fill the system and the end pressure inside it. First, a sensitivity analysis is performed to identify the system parameters that affect these system characteristics most, see Table 1.

**Table 1.** Normalized sensitivities for parameters of the dynamic pressure regulator.

	Tank pressure, $p_t$	Max opening area of on/off -valve, $A_o$	Max opening area of vent valve, $A_v$	Max opening area of main valve, $A_m$	Friction coefficient in main valve	Mass of piston, $m_p$	Preloading of piston	Piston area for output flow, $A_2$	Piston area for support pressure, $A_s$
End pressure	1.1	0.13	-1.0	0.35	0.055	0.053	0.033	0.15	0.013
Filling time	1E-4	-6E-5	-8E-10	3E-4	0.008	0.50	5E-4	-0.25	-0.26

The normalized uncertainties for several parameters are estimated according to Eq. (8) and shown in Table 1. According to these numbers, the most important parameters are the geometries of the main valve as well as the tank pressure. Extra care of these parameters should be taken during the design process of the DPR since these parameters affect the performance most. Figure 14 shows a schematic picture of the main valve where several of the parameters in Table 1 are displayed.

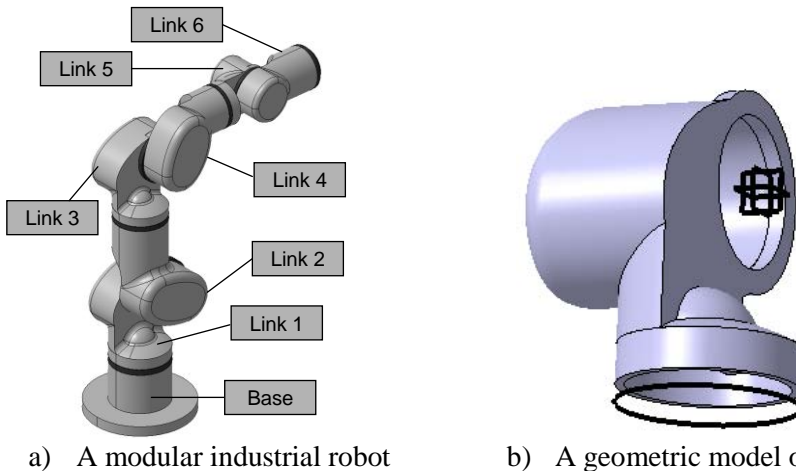




**Figure 14.** A schematic picture of the main valve inside the DPR.

## 6.2 INDUSTRIAL ROBOT

Paper [II] describes an optimization framework that is used to support the development of industrial robots. The details are found in the paper and the framework consists of a geometry model, a dynamic model, a finite element model and a simple cost model. In this thesis, the focus is on the geometrical model, and how it could be represented by different types of surrogate models. The robot that is studied is a serial mechanism where a set of links are interconnected to form a complete robot structure. Each link is equipped with an actuator that performs the motion of the robot. A picture of the robot is shown in Figure 15 together with a geometric model of a link.



a) A modular industrial robot

b) A geometric model of a link

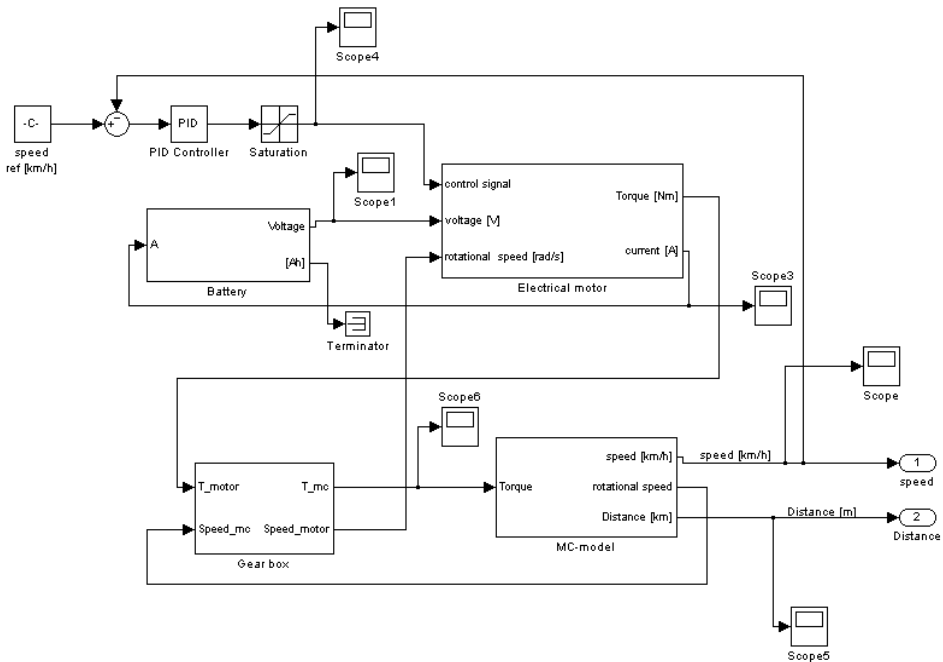
**Figure 15.** Two pictures of a) the studied industrial robot and b) a geometric model of a link.

The geometric model is built up of high-level CAD-templates, which ensures that the shape of the model can change both morphologically (e.g. can be stretched out) and topologically (e.g. can change actuator). The lengths of the links and the discrete choices of which actuators that should be used are the inputs which control the shape of the geometric model. The outputs are the masses of the links, the moments of inertia and the center of gravities. This information is then passed to the dynamic simulation model and the FE-model, which are used for performance and lifetime calculations respectively.

### 6.3 ELECTRIC MOTORCYCLE

The final application is an electric motorcycle, modeled in MATLAB Simulink [12] as shown in Figure 16.

The model is built from physical equations and consists of four parts; the battery, the electrical motor, the gear box and the chassis. The outputs from the model are the speed and distance traveled as functions of the elapsed time. It is possible to change the value of any parameter in the model, but for these analyses the focus is on the design of the gearbox. The parameters of the gearbox are the gear ratios for the first and second gear as well as the speed at which the electric motorcycle travels when the gear is shifted from the first to the second gear.



**Figure 16.** A screenshot from Simulink that displays an electric motorcycle model.

The target of the design process is to design a gearbox which enables the motorcycle to accelerate as fast as possible. The acceleration is estimated as the speed of the motorcycle after five seconds. This optimization problem can be expressed with Eq. (25).

$$\begin{aligned} \min_x \quad & f(x) = -acc(x) \\ \text{s.t.} \quad & \\ & x = [u_1 \ u_2 \ v_{shift}] \\ & 1 \leq u_1 \leq 50 \\ & 1 \leq u_2 \leq 50 \\ & 1 \leq v_{shift} \leq 75 \end{aligned} \tag{25}$$

Several optimizations reveal that the maximum speed after five seconds is slightly above 75 km/h and can be reached if the gear ratio of the first gear is around 6, the gear ratio of the second gear around 11 and the gears are shifted at 38 km/h.

*If a man's wit be wandering, let him study the mathematics.*  
Francis Bacon

# 7

## COMPARISON OF SAMPLING BASED METHODS

---

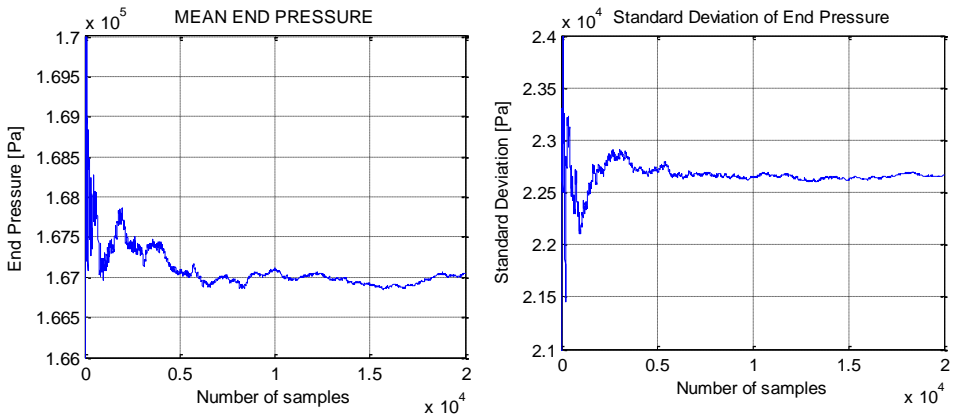
It is desirable to estimate how uncertainties and variations affect the performance of a product. This estimation can be done by using a sampling based method. Four sampling based methods are presented and compared in this section. More information can be found in paper [I].

The dynamic pressure regulator model which is described in section 6.1 is used to investigate the performance of the sampling based methods when it comes to estimating probability distributions of a model output.

The four compared methods are;

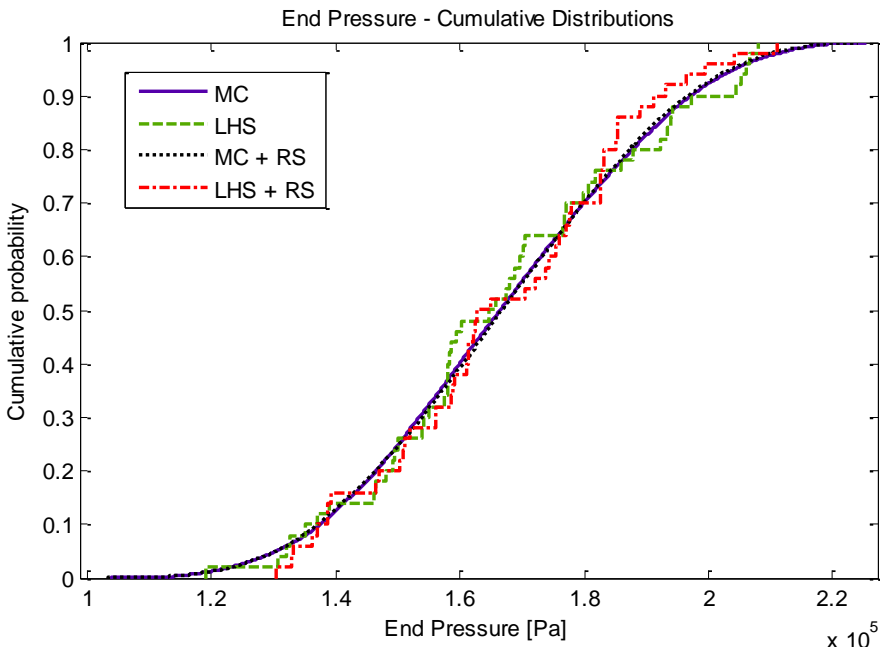
- A Monte Carlo Simulation with 20 000 samples.
- A Latin Hypercube Sampling with 50 samples.
- A Monte Carlo Simulation with 20 000 samples with the original model replaced with a second order response surface.
- A Latin Hypercube Sampling with 50 samples, with the original model replaced with a second order response surface.

The probability distribution of the Monte Carlo Simulation is seen as the reference since the original model is called for all samples and numerous samples are drawn. This assumption is also used by several other authors [35]. The other three methods will therefore be compared with it.



**Figure 17.** Two graphs over how the statistical entities a) mean value and b) standard deviation of the end pressure varies for each added Monte Carlo sample.

The variation of the values of the standard deviation and mean value of the end pressure inside the volume for each sample as the Monte Carlo Simulation progresses is shown in Figure 17. It seems like the curves have converged which makes it probable that further samples would not affect the values of the standard deviation and mean value much.



**Figure 18.** A graph that displays the cumulative distributions for the end pressure inside the volume estimated by four different methods.

The appearances of the probability distributions of the end pressure for all four methods are shown in Figure 18. All four curves agree quite well but the curves which represent LHS are a bit jagged. This stems from the fact that only 50 samples are used to estimate the probability distributions for LHS.

The resulting values of the mean values and standard deviations for all four methods are shown in Table 2. All four methods return approximately the same values for both the mean values and standard deviations. The root mean squared error (RMSE) is a measure of how much the other three methods deviate from the Monte Carlo simulation.

**Table 2.** The mean values and standard deviations which have been estimated using four different methods.

Entity	MC	LHS	MC + RS	LHS + RS
Simulation Time [s]	25 640	65	351	199
Number of samples	20 000	50	20 000	50
Number of Dymola calls	20 000	50	80	80
Mean end pressure [Pa]	166 944	166 776	166 881	166 763
Variance of end pressure	5.11E8	5.10E8	5.03E8	4.13E8
RMSE to MC [Pa]	-	2563	429	3 419
Mean filling time [s]	0.028228	0.028233	0.028218	0.28219
Variance of filling time	3.03E-6	3.67E-6	3.01E-6	3.14E-6
RMSE to MC [s]	-	2.99E-4	5.35E-5	2.19E-4

According to the graphs and tables in this chapter, it seems appropriate to use LHS whenever the mean value or standard deviation needs to be estimated. If the probability distribution is of interest, it may be better to create a surrogate model of the original model and perform a Monte Carlo Simulation on it instead.

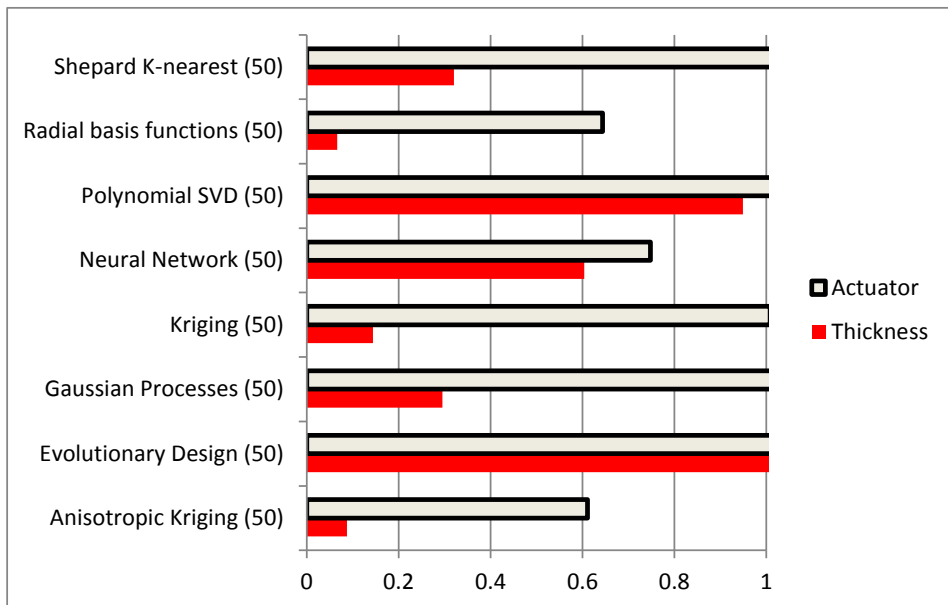
*Your theory is crazy, but it's not crazy enough to be true.*  
Niels Bohr



# COMPARISON OF SURROGATE MODELS

---

This section compares the performance of a set of surrogate models and how applicable they are on some specific engineering models. A detailed description of these findings can be found in paper [II].



**Figure 19.** A figure that displays the NRSMEs for each of the compared surrogate models.

The application that is used for the comparison is the geometric CAD-model of a link of the industrial robot presented in section 6.2. The inputs are the continuous variable link thickness as well as the discrete choice of actuator. The outputs from the model are the mass, inertia and center of gravity of the link.

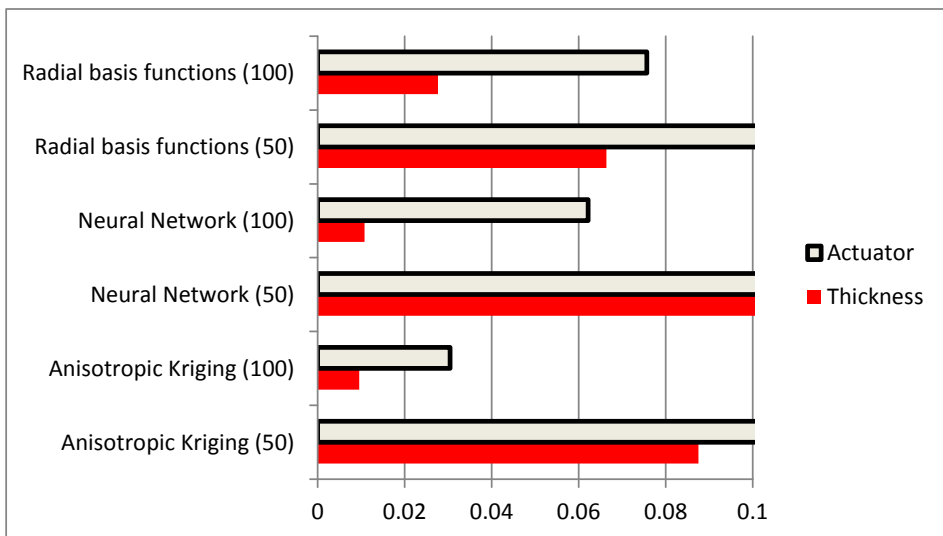
All surrogate models which are included in the design environment ModeFrontier [42] are used for the comparison. The surrogate models are fitted using the same 50 samples and their accuracies are determined with respect to 20 additional samples, which have not been used to fit the surrogate models. The used accuracy measures are the NRSME, RAAE and RMAE as defined in section 4.5.2.

**Table 3.** A table that contains accuracy measures for the compared surrogate models when they are compared with the original model.

<b>Actuator Variation</b>			
<b>Model (no of samples)</b>	<b>NRSME</b>	<b>RAAE</b>	<b>RMAE</b>
Anisotropic Kriging (50)	0.611	1.223	1.986
Evolutionary Design (50)	1.449	2.590	3.966
Gaussian processes (50)	1.699	5.660	10.977
Kriging (50)	1.009	2.665	4.724
Neural Networks (50)	0.748	1.968	2.489
Polynomial Svd (50)	3.324	11.121	12.805
Radial Basis Functions (50)	0.644	2.230	4.626
Shepard K-nearest (50)	1.539	6.324	7.157
Anisotropic Kriging (100)	0.030	0.107	0.218
Neural Networks (100)	0.062	0.238	0.465
Radial Basis Functions (100)	0.076	0.196	0.410
<b>Thickness Variation</b>			
<b>Model (no of samples)</b>	<b>NRSME</b>	<b>RAAE</b>	<b>RMAE</b>
Anisotropic Kriging (50)	0.346	0.095	0.346
Evolutionary Design (50)	0.989	0.249	0.989
Gaussian processes (50)	1.226	0.404	1.226
Kriging (50)	1.161	0.307	1.161
Neural Networks (50)	0.558	0.144	0.558
Polynomial Svd (50)	2.828	1.550	2.827
Radial Basis Functions (50)	0.501	0.176	0.501
Shepard K-nearest (50)	1.847	0.666	1.847
Anisotropic Kriging (100)	0.010	0.023	0.067
Neural Networks (100)	0.011	0.034	0.086
Radial Basis Functions (100)	0.028	0.025	0.065

The resulting measures from the comparison are seen in Table 3, whereas a graph over the NRSMEs is shown in Figure 19. The numbers inside the

parentheses are the number of samples that were used to fit the surrogate models. It is shown that the surrogate models generally have more problems to handle the discrete variable actuator type than the continuous variable thickness. The three most promising surrogate model types are Anisotropic Kriging, Neural Networks and Radial Basis Functions. Those three model types are fitted using 100 samples as well to estimate the impact of the number of samples used to fit the surrogate models. As shown both in Table 3 and Figure 20, the measures improve dramatically. An improvement is expected since more samples generally result in more accurate surrogate models, but it is a bit unexpected that this doubling of the number of samples results in such large improvements in accuracies.



**Figure 20.** NRSMEs for Radial Basis Functions, Neural Networks and Anisotropic Kriging when 50 and 100 samples are used to fit the surrogate models.

The result of the comparison is used to choose the surrogate model type to be used in an optimization framework for industrial robots. Since anisotropic kriging performed best, it is the chosen surrogate model type. Anisotropic kriging is a version of kriging where the distance correlation function varies with the direction in the design space between two points.

*The scientific theory I like best is that the rings of Saturn are composed entirely of lost airline luggage.*

Mark Russell

# PERFORMANCE INDEX FOR OPTIMIZATIONS

---

In order to compare different optimization algorithms, a performance index which specifies a numerical value of the performance of an optimization algorithm is required. A numerical value enables comparison of the performances of two optimization algorithms for a specific problem. The performance index can also be used as objective function in an optimization with the aim of optimizing the performance of the optimization algorithm itself.

A few performance indices have been suggested [43],[44], but here the performance index which is used in paper [III] and [IV] is presented.

It is desirable to use an optimization algorithm which finds the optimum with as few calls to the objective function as possible. But it is also desirable to use an algorithm which has a high probability of finding the global optimum. The suggested performance index is therefore a combination of those two entities.

If the probability of finding the global optimum with one optimization is  $p_{opt}$ , then the probability of not finding the optimum is  $1-p_{opt}$ . The probability of not finding the global optimum with  $n$  optimizations can then be described by Eq. (26).

$$p_{not\_opt} = (1 - p_{opt})^n \quad (26)$$

Consequently, the probability of finding the global optimum can be calculated with Eq. (27).

$$p_{opt\_found} = 1 - (1 - p_{opt})^n \quad (27)$$

Since different optimization algorithms require different numbers of objective function calls, the performance index needs to take this into account to enable fair comparison. Therefore, the variable  $n$  in Eq. (27) needs to be replaced with the number of optimizations that are possible to perform with our available computational time. This can be expressed as the allowed number of function evaluations divided by the mean number of function evaluations required to perform one optimization, or  $n_{available}/n_{opt}$ . The available computer time varies with each problem, but here it is suggested to use a reference value of 100. The resulting performance index is shown in Eq. (28) as the probability of finding the global optimum if 100 objective function calls are made.

$$pi = 1 - (1 - p_{opt})^{100/n_{opt}} \quad (28)$$

A drawback with the performance index is that all optimization algorithms with a 100% hit-rate get performance indexes equal to one. But since they have the same accuracy, the one which requires the fewest objective function evaluations to converge is the most suitable one.

*Science, my lad, is made up of mistakes, but they are mistakes which it is useful to make, because they lead little by little to the truth.*

Jules Verne

# 10

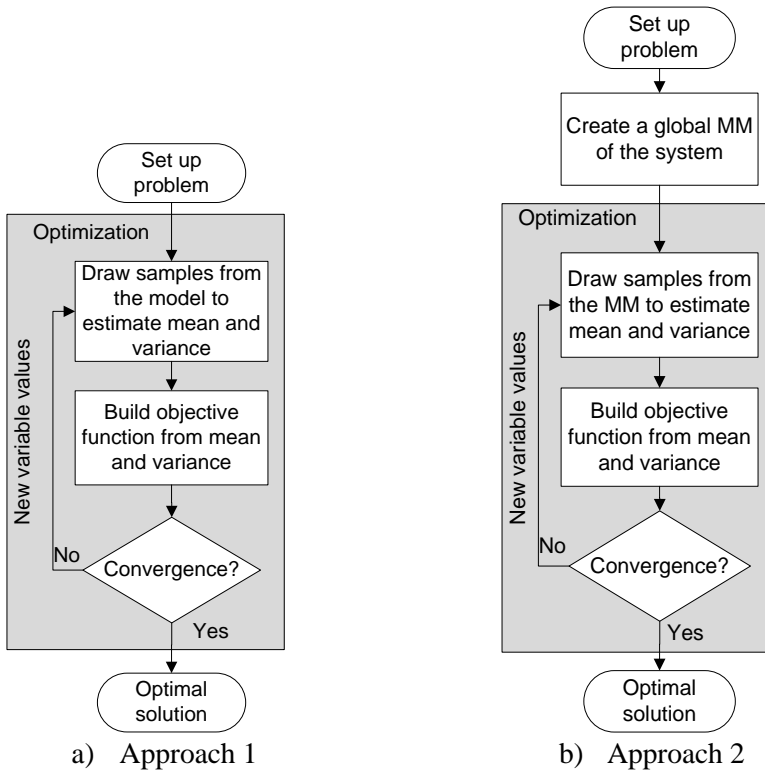
## ROBUST DESIGN OPTIMIZATION

---

Surrogate models may be applied at different stages of a probabilistic optimization. Either before the optimization is started or during it. The main benefit with creating a surrogate model before the optimization is started is that the surrogate model can be used for other analyzes afterwards. The motivations for creating a surrogate model during an optimization are that no unnecessary samples are drawn and that the surrogate model will be more accurate in the areas where the optimization is operating, i.e. the interesting areas.

This section strives to compare the performances of a few different approaches that can be used to perform Robust Design Optimization. More details can be found in paper III. Latin Hypercube Sampling is used to estimate the main value and standard deviation for each design, whereas Complex-RF is used as optimization algorithm.

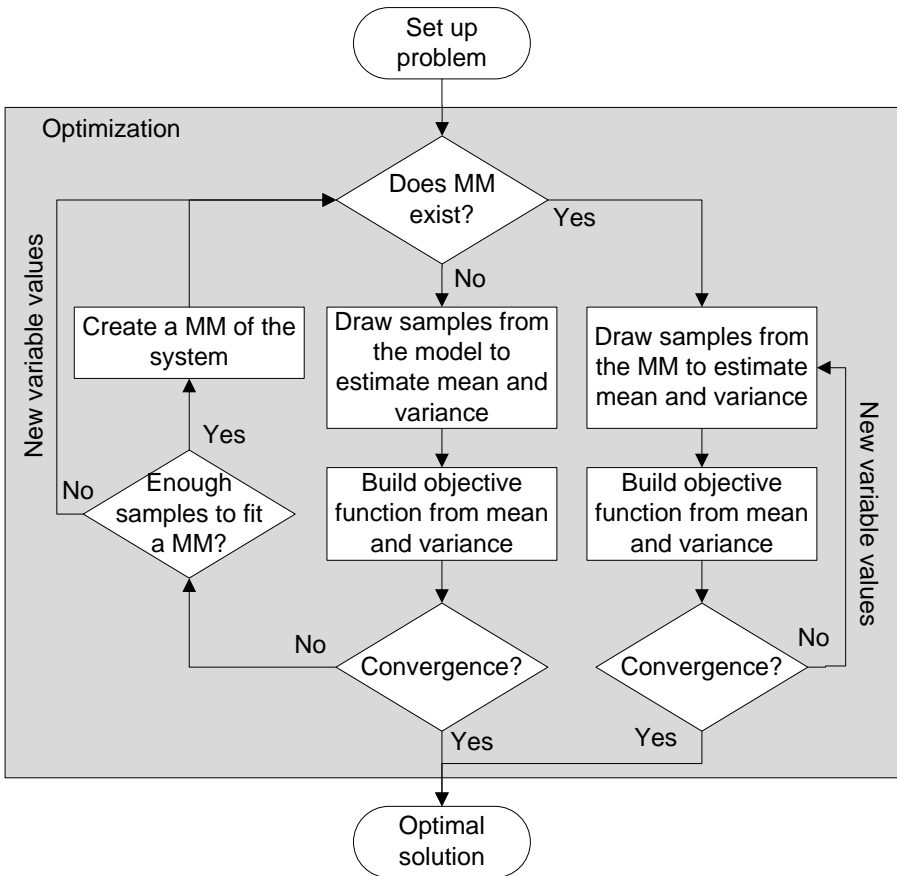
The first approach is shown in Figure 21a). The objective function is a combination of the mean value and standard deviation of an output from a function or a model. Consequently, an LHS is performed each time the optimization algorithm needs to determine the value of the objective function. The approach does not involve surrogate models. This means that the required number of function calls or model simulations to reach the optimum is the number of samples performed by the LHS multiplied by the required number of objective function evaluations by the optimization algorithm.



**Figure 21.** Schematic workflows for approach 1 and 2.

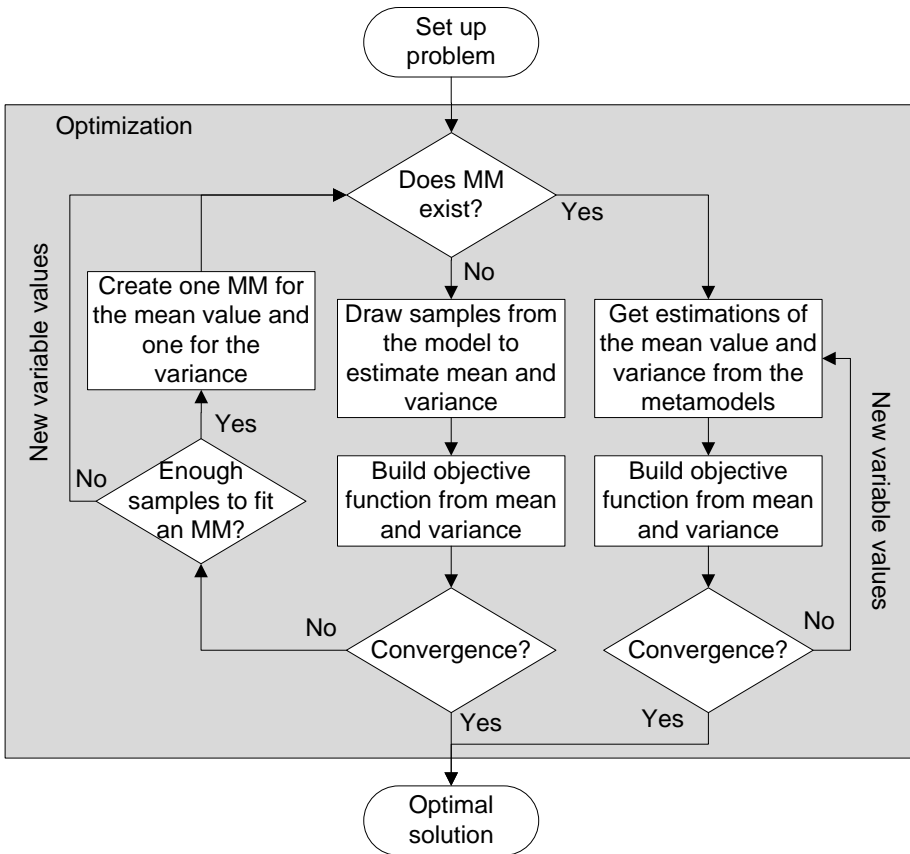
The second approach is shown in Figure 21b) and begins by drawing samples from the function or model. These samples are used to create a surrogate model of the desired entity. When the optimization starts, the surrogate model is called instead of the original objective function each time LHS estimates the mean value and standard deviation. The only true function calls or model simulations made are those that are needed to build the global surrogate model in the first place.





**Figure 22.** A Schematic workflow for approach 3.

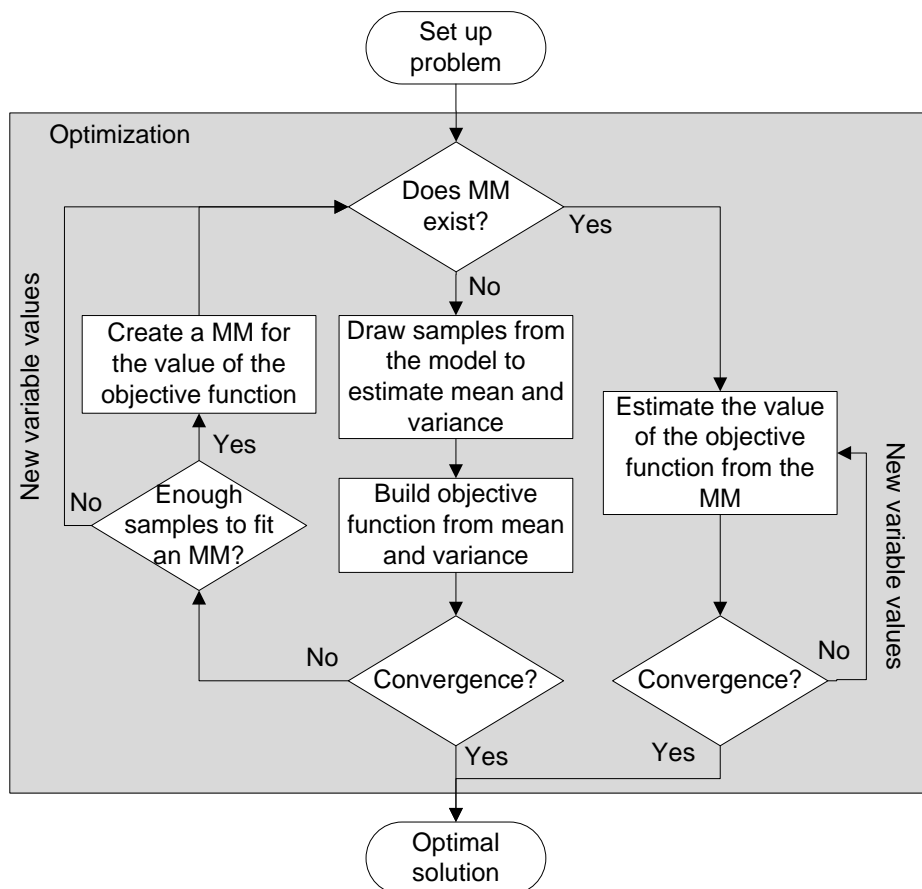
The third approach is shown in Figure 22 and begins similarly to the first approach. This means that it performs an LHS each time the value of the objective function needs to be estimated. When a predefined number of samples have been drawn, a surrogate model of the function value or model output is created. Similar to approach 2, the surrogate model is called by LHS during the remainder of the optimization. A somewhat similar approach is named Local quasi-random sampling by Aspenberg [18].



**Figure 23.** A schematic workflow for approach 4.

The workflow of approach 4 is shown in Figure 23. It operates similarly to approach 3 in the aspect that it creates surrogate models during the optimization process. The difference is that two surrogate models are created in this approach – one of the mean value and one of the standard deviation.

The fifth approach also creates a surrogate model during the optimization. But in this approach the surrogate model which is created reanimates the objective function value, i.e. an aggregate of both the mean and the standard deviation. A schematic picture of its workflow is shown in Figure 24.



**Figure 24.** A schematic workflow of approach 5.

These different approaches are evaluated on two mathematical functions and the DPR problem which is presented in section 6.1. Table 4 displays the results from the comparison. The numbers represent the value of the performance index which was presented in section 9 for the five approaches for these problems. Approaches 2 through 5 are allowed to make 50 model simulations or calls of the mathematical functions before the surrogate models are created. It is shown that approach 1 has a poor performance for all problems even though the hit-rates are high. This stems from the numerous model simulations and function calls that are needed to make the optimization converge. Approach 2, which creates a surrogate model before the optimization is started, outperforms the other approaches for most of these problems.

**Table 4.** Performance indices for the five approaches for the different problems.

Problem	App 1	App 2	App 3	App 4	App 5
Peaks	0.014	0.428	0.309	0.003	0.001
Aspenberg	0.410	0.227	0.212	0.053	0.062
DPR	0.070	0.762	0.719	0.452	0.404

Another benefit with using approach 2 is that a surrogate model of the model is received. This model can be used for additional analyzes later. The surrogate models which are created in the other approaches can also be used later, but they either reanimate other things than the model output and/or are accurate in a smaller domain of the design space

*Any sufficiently advanced technology is indistinguishable from magic.*  
Arthur C Clarke

# MODIFICATION OF THE COMPLEX ALGORITHM

---

With inspiration from other optimization algorithms, a modified version of the Complex-RF optimization algorithm is proposed, Complex-RFM. The aim is to increase the effectiveness of the Complex-RF algorithm for computationally demanding objective functions by reducing the number of required calls to the objective function. The modified algorithm creates and calls a surrogate model, a second order response surface, of the objective function during the optimization progress.

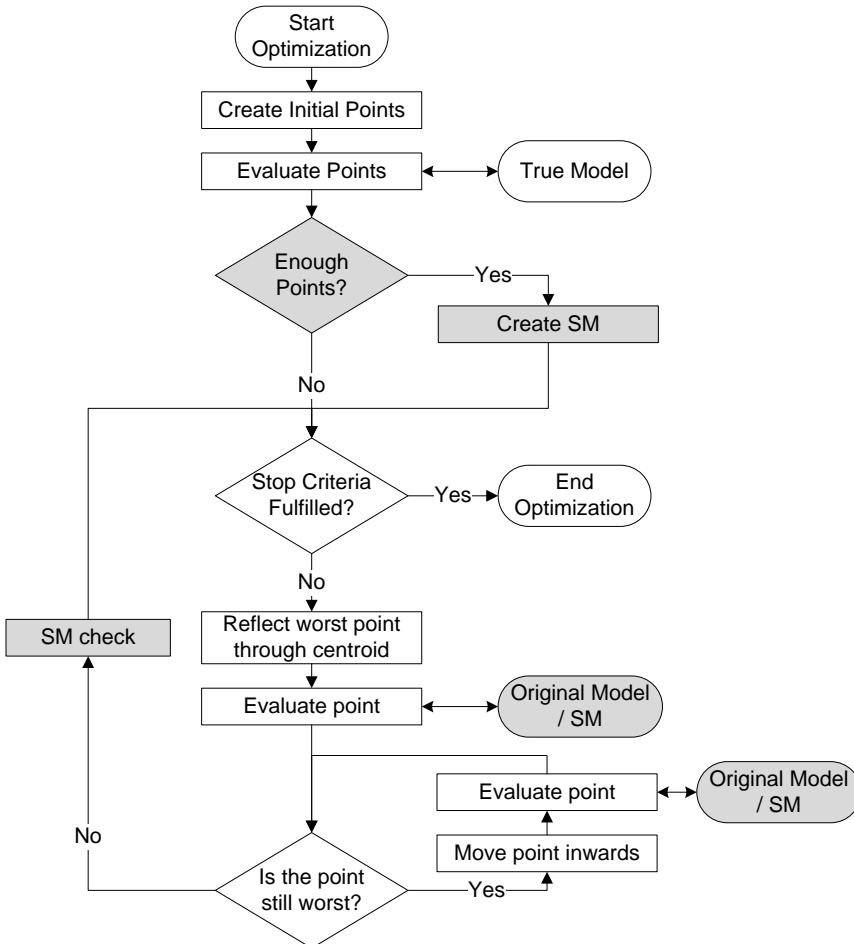
The performance of the modified algorithm is compared with a few existing algorithms in Table 5. The numbers correspond to the values of the performance index that is presented in section 9 for each problem and optimization algorithm. More detailed results and definitions of the problems can be found in paper [IV]. The compared algorithms are Complex-RF, ComplexRFM, fmincon and a genetic algorithm. 1000 optimizations are performed with each algorithm to estimate the hit-rates and required number of function calls that are needed for the algorithm to converge. For fmincon, the starting point is randomly selected within the design space for each optimization. The genetic algorithm focuses on precision rather than speed in this comparison but is interrupted when the optimum has been found.

The parameters of Complex-RF and Complex-RFM are optimized for each problem, which makes a direct comparison with the other algorithms unfair. But the comparison nevertheless gives an idea of the performance of Complex-RFM and it can be noted that Complex-RFM performs better than Complex-RF for all problems. Even though fmincon excels for two functions it struggles with the engineering problem.

**Table 5.** Comparison of the performances of different optimization algorithms for a few problems. The numbers represent the performance indices for each algorithm and problem.

Problem	fmincon	GA	Complex-RF	Complex-RFM
Peaks	0.726	0.333	0.647	<b>0.910</b>
Rosenbrock	<b>0.999</b>	0.086	0.783	0.893
Hart6	<b>0.624</b>	0.040	0.248	0.412
EL-MC	0.080	0.380	0.322	<b>0.721</b>

The operations that are performed by the proposed algorithm are shown in Figure 25 and a more thorough description is found in paper IV.

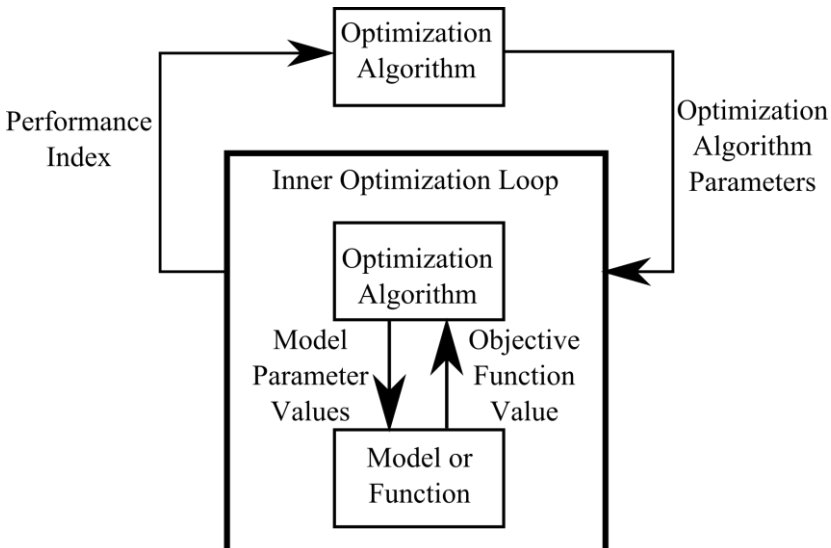


**Figure 25.** A schematic workflow for Complex-RFM.

It basically operates similar to Complex-RF, however the grey activities are different. The original function is called until enough samples have been drawn to create a second order response surface with 50% oversampling. From that point, the response surface is called each time the value of the objective function needs to be determined by the optimization algorithm. The activity “SM check” is added to ensure that the response surface is accurate. If the differences between the coefficients of the two latest surrogate models are small enough, the response surface is deemed accurate and the “SM check” activity is omitted for a few iterations.

The criterion for how large the percentage difference between the coefficients of the latest response surfaces are allowed to be is one of the parameters of the optimization algorithm and denoted  $\varepsilon$ . Another parameter is the number of iterations for which the activity “SM check” activity is omitted when two response surfaces are deemed equal, and it is denoted  $\zeta$ .

The values of these two parameters, as well as the randomization and forgetting factors,  $rfak$  and  $\gamma$ , can be used as variables in an optimization with the purpose of optimizing the performance of the optimization algorithm. A schematic figure of this process is shown in Figure 26. The performance index is used as objective function for the optimization and each time the performance index is estimated, 1000 optimizations with the same parameter values are performed.



**Figure 26.** A schematic workflow for optimization of the parameters of an optimization algorithm.

This optimization returns different optimal parameter values for each function and engineering problem respectively. These values are shown in Table 6.

**Table 6.** Optimal parameter values for Complex-RFM for a few problems.

Problem	$\varepsilon$	$\zeta$	rfak	$\gamma$
Peaks	0.01	25	1	0.15
Rosenbrock	0.12	1	0.15	0.96
Hart6	0.002	2	0.21	0.05
EL-MC	0.002	48	0.32	0
All problems (Eq. (29))	0.002	2	0.21	0.05

It is desirable to recommend standard settings for the algorithm when it is used to optimize an arbitrary function. An optimization which takes the performance indices of all functions into account is therefore performed. The used objective function is shown in Eq. (29). The performance index of each problem is divided by the corresponding utopian performance index, which is shown in the “Complex-RFM” column of Table 5.

$$\begin{aligned}
 \min_x f(x) = & -\left(\frac{pi_{peaks}(x)}{0.91}\right)^2 - \left(\frac{pi_{rosenbrock}(x)}{0.893}\right)^2 - \left(\frac{pi_{hart6}(x)}{0.412}\right)^2 \\
 & - \left(\frac{pi_{elmc}(x)}{0.721}\right)^2 \\
 \text{s.t.} & \\
 x = & [\varepsilon \quad \xi \quad rfak \quad \gamma] \\
 0.001 \leq & \varepsilon \leq 0.2 \\
 1 \leq & \xi \leq 50 \\
 0 \leq & rfak \leq 1 \\
 0 \leq & \gamma \leq 1
 \end{aligned} \tag{29}$$

The resulting optimal parameter values are seen in the last row of Table 6. These are also the parameters that are recommended to use as standard parameters for an arbitrary problem. The difference in performance for Complex-RFM when its parameters are optimized and when the suggested standard values of the parameters are used is shown in Table 7. The performance indices are a bit worse for the optimizations that are performed with the standard parameter values than when optimal parameters values are used, which was expected. But the overall performance of the algorithm is still good.



**Table 7.** Performance indices for Complex-RFM when its parameters are optimized for each specific problem and when the standard values are used.

Problem	Optimal Parameters	Standard Parameters
Peaks	0.910	0.807
Rosenbrock	0.893	0.826
Hart6	0.412	0.412
EL-MC	0.721	0.711

The purpose of Complex-RFM is to speed up the optimization for moderately heavy simulation models. It might therefore be better to weight the engineering problem a bit more in Eq. (29). But it is difficult to say how an arbitrary problem behaves beforehand.

*No amount of experimentation can ever prove me right; a single experiment can prove me wrong.*

Albert Einstein

# DISCUSSION

---

The methods that are presented in this thesis may all be used to perform a probabilistic optimization. Their performances are demonstrated for several mathematical functions and engineering system models. Since 1000 optimizations are performed to estimate the accuracies of the methods, the accuracies are sensitive to the criteria which decide if an optimization is successful or not. Even though this affects the results for the comparisons, the comparisons should be possible to use as guidelines to choose a method.

The performance index is a tool which can be used to compare the performances of different methods. Since it is a numerical value it enables optimization of the parameters of the optimization algorithms, a tool which is useful to determine standard values for the parameters of new algorithms are suggested. This is demonstrated for the Complex-RFM algorithm.

In the case of similar performance indices for two different methods, it is doubtful which method to choose. The author would probably choose the method which requires the least number of model simulations to converge. Then several optimizations are performed and the best solution used.

The choice of performance index may also affect the results from the comparisons. This is especially important for the optimization of the Complex-RFM algorithm where the performance index is used as objective function with the aim of optimizing the parameters of Complex-RFM.

The RDO approaches which create surrogate models during the probability optimization process, approach three to five, perform worse than approach two. This should not be interpreted as such that all methods which create surrogate models during the optimizations are worthless. There probably exist more sophisticated methods which perform better, but they have not been implemented in the scope of this thesis.

Since the methods are intended to be used to design products, it might seem redundant to compare the performances of the methods for simple numerical

problems. But, as Beyer et al. [8] points out; performance comparisons should be made with simple test functions in the beginning, since it is easier to understand the working principles of the compared methods. It is also easier to scale a simple problem by varying the number of parameters.

In an industrial case, the target is often to find the best design given certain conditions. This means that it is desirable to find a solution which is as good as possible in as short time as possible. If the optimal design is already known then there is no use to perform an optimization. Consequently, one method should be chosen and probabilistic optimizations performed until it is believed that the best design has been found.

The author would probably use much of the available computer time to draw samples of the model output. This database of samples can then be used to create several surrogate models, using the leave-one-out method. Then, probabilistic optimizations could be performed on the most promising surrogate models.

Future work includes two main tracks. The first is to implement a framework for probabilistic optimizations and apply it to design a product and then construct and evaluate physical prototypes. The aim would be to compare different design proposals by introducing uncertainties and evaluate their performances. Hence, this would ascertain the performance of the methods with regards to their intended purpose – to enable development of physical products by modeling and simulation.

The other track is to develop Complex-RFM further and improve its performance. It might be possible to replace the second order response surfaces with radial basis functions or kriging. However, it is probable that the matrices become singular since the surrogate models are fitted increasingly locally and close samples leads to singular matrices. This needs to be remedied to be able to use kriging or radial basis functions effectively.

# REFERENCES

---

- [1] Beckwith, T. G., Marangoni, R.D. and Lienhard V, J.H., *Mechanical Measurements, sixth edition*, Pearson Prentice Hall, Upper Saddle River, NJ, 2009, Chaps. 3.
- [2] Roache, P. J., *Verification and validation in computational science and engineering*. Hermosa Publishers, Albuquerque, New Mexico, 1998, pp. 23.
- [3] Sargent, R. G., *Verification and validation of simulation models*, Proceedings of the 2008 Winter Simulation Conference, 2008, pp. 157-169.
- [4] Johnson, R.T., Montgomery, D.C., Jones, B., and Fowler, J.W., *Comparing Designs for Computer Simulation Experiments*, Proceedings of the 2008 Winter Simulation Conference, 2008.
- [5] Roshan Joseph, V., Hung, Y., and Sudjianto, A., *Blind Kriging: A New Method for Developing Metamodels*, Journal of Mechanical Design, March 2008, Vol. 130.
- [6] Park, H-S., and Dang, X-P., *Structural optimization based on CAD-CAE integration and metamodeling techniques*, Computer-Aided Design, Vol. 42, 2010, pp. 889-902.
- [7] Kazemi, M. *Metamodel-Based Optimization for Problems With Expensive Objective and Constraint Functions*, Journal of Mechanical Design, Vol. 133, January 2011.
- [8] Beyer, H-G., and Sendhoff, B., *Robust Optimization – A comprehensive survey*, Computer Methods in Applied Mechanics and Engineering, Vol. 196, 2007, pp. 3190-3218.

- [9] Andersson, J., *Multi-Objective Optimization in Engineering Design – Application to Fluid Power Systems*, PhD thesis, Division of Fluid and Mechanical Engineering Systems, Linköpings universitet, Linköping, Sweden, September 2001, Dissertation No. 657.
- [10] Ericsson, L., *On Fluid Power Pump and Motor Design – Tools for Noise Reduction*, PhD thesis, Division of Fluid and Mechanical Engineering Systems, Linköpings universitet, Linköping, Sweden, January 2012, Dissertation No. 1417.
- [11] Johansson, A., Andersson, J., and Palmberg, J-O., *Optimal Design of the Cross-Angle for Pulsation Reduction in variable Displacement Pumps*, in Proceedings of Bath Workshop on Power Transmission and Motion Control, PTMC 2002, Bath, UK, September 2002.
- [12] <http://www.mathworks.se/products/simulink/>, accessed 2012-10-17.
- [13] <http://www.3ds.com/products/catia/portfolio/dymola>, accessed 2012-10-17.
- [14] Mavris, D.N., DeLaurentis, D.A., Bandte, O. and Hale, M.A., *A Stochastic Approach to Multi-Disciplinary Aircraft Analysis and Design*, AIAA 98-0912, 1998.
- [15] Duvigneau, R., and Praveen, C., *Meta-modeling for Robust Design and Multi-Level Optimization*, 42nd AAAF Congress on Applied Aerodynamics, Sophia-Antipolis, France, March 2007.
- [16] Martin, J. D. and Simpson, T. W., *A methodology to manage uncertainty during system-level conceptual design*, Proceedings of IDETC/CIE 2005 ASME 2005 International Design Engineering Technical Conferences & Computers and information in Engineering Conference, 2005, pp. 1-11.
- [17] Padulo, M., “*Computational Engineering Design under uncertainty – An aircraft conceptual design perspective*”, Cranfield University, PhD Thesis, July 2009.
- [18] Aspenberg, D., *Robust optimization of structures – Evaluation and incorporation of variations in simulation based design*, PhD thesis, Division of Solid Mechanics, Linköpings Universitet, Linköping, Sweden, August 2011, Dissertation No. 1382.

- 
- [19] McKay, M.D., Beckman, R.J. and Conover, W.J., *A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code*, Technometrics, Vol. 21. No. 2 , 1979, pp. 239-245.
- [20] Box, M., J., *A New Method of Constraint Optimization and a Comparison with Other Methods*, Computer Journal, 8(1):42-52, 1965.
- [21] Nelder, J., Mead, R., *A Simplex method for function minimization*, Computer Journal 7: 308–313, 1965.
- [22] Ölvander, J., and Krus, P., *Optimizing the Optimization - A Method for Comparison of Optimization Algorithms*, proceedings of the 2nd AIAA Multidisciplinary Design Optimization Specialists Conference, Newport, RI, USA, May 1-4, 2006.
- [23] Krus, P., and Ölvander, J., *Optimizing Optimization for Design Optimization*, in Proceedings of ASME Design Automation Conference, Chicago, USA, September 2-6, 2003.
- [24] <http://www.mathworks.se/products/matlab/>, accessed 2012-10-18.
- [25] Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley Longman Publishing Co., Inc, Boston, MA, USA, 1989.
- [26] Holland, J. H., *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, The University of Michigan Press, Ann Arbor, USA, 1975.
- [27] Forrester, A. I. J., Sóbester, A. and Keane A, J., *Engineering Design via Surrogate Modelling: A Practical Guide*, John Wiley & Sons Ltd, Chichester, West Sussex, United Kingdom, 2008, Chaps. 1.
- [28] Saltelli, A., Chan, K. and Scott, E.M., *Sensitivity Analysis*, John Wiley & Sons, Chichester, West Sussex, 2000, pp. 10.
- [29] Steinkellner, S. and Krus, P., *Balancing Uncertainties in Aircraft System Simulation Models*, 2nd European Air & Space Conference CEAS, Manchester, United Kingdom, 2009.

- [30] Wang, G., and Shan, S., *Review of Metamodeling Techniques in Support of Engineering Design Optimization*, Journal of Mechanical Design, Vol. 129, No. 2, April 2007, pp. 370-380.
- [31] Myers, R.H., Montgomery, D.C. and Anderson-Cook, C.M., *Response Surface Methodology, Process and Product Optimization Using Designed Experiments, Third Edition*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2009.
- [32] Simpson, T. W., Peplinski, J., Koch, P. N., and Allen, J. K., *Metamodels for Computer-Based Engineering Design: Survey and Recommendations*, Engineering with Computers, Vol. 17, No. 2, 2001, pp. 129-150.
- [33] Redhe M., Forsberg J., Jansson T., Marklund P.-O., and Nilsson L. (2002). *Using the Response Surface Methodology and the D-optimality criterion in crashworthiness related problems*, Structural and Multidisciplinary Optimization 24(3), pp. 185-194.
- [34] Jin, R., Du, X., and Chen, W., *The use of metamodeling techniques for optimization under uncertainty*, Journal of Structural & Multidisciplinary Optimization, Vol. 25, No. 2, 2003, pp. 99-116.
- [35] Martin, J. D. and Simpson, T. W., *A Monte Carlo Simulation of the Kriging Model*, 10th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA-004-4483, 2004.
- [36] Isaaks, E. H., and Srivastava, R. M., *An introduction to Applied Geostatistics*. Oxford University Press, New York, New York, 1989.
- [37] Shao, X. Y., Chu, X. Z., Gao, L., and Qui, H. B., *A novel method to manage uncertainty for multidisciplinary design and optimization in conceptual design*, Advanced Materials Research, Vol. 44-46, 2008, pp. 225-232.
- [38] Jin, R., Chen, W., and Simpson, T. W., *Comparative Studies of Metamodeling Techniques under Multiple Modeling Criteria*, Journal of Structural & Multidisciplinary Optimization, Vol. 23, No. 1, 2001, pp. 1-13.
- [39] Nakayama, H., Arakawa, M., and Sasaki, R., *A Computational Intelligence Approach to Optimization with Unknown Objective Functions*, ICANN 2001, International Conference on Neural Networks, Vienna, Austria, pp 73-80, 2001.



- [40] Shan, S., and Wang, G., *Metamodeling for High Dimensional Simulation-Based Design Problems*, Journal of Mechanical Design, Vol. 132, May 2010.
- [41] Fritzson, P., *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*, IEEE Press, Piscataway, New Jersey, 2004.
- [42] [http://www.esteco.com/home/mode\\_frontier/mode\\_frontier.html](http://www.esteco.com/home/mode_frontier/mode_frontier.html), accessed 2012-10-17.
- [43] Krus, P., and Ölvander, J., *Performance index and meta-optimization of a direct search optimization method*, Engineering Optimization, 2012, DOI: 10.1080/0305215X.2012.725052.
- [44] Schutte, J.F., and Haftka, R.T., *Improved global convergence probability using independent swarms*, 46<sup>th</sup> AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics & materials conference, Austin, TX, April 18-21 2005.