Efficient Optimization of Complex Products

A Simulation and Surrogate Model Based Approach

Johan Persson

Linköping University
INSTITUTE OF TECHNOLOGY

Division of Machine Design
Department of Management and Engineering
Linköping University, SE-581 83 Linköping, Sweden

Linköping 2015

Copyright © Johan Persson, 2015

Efficient Optimization of Complex Industrial Products
A Simulation and Surrogate Model Based Approach

ISBN 978-91-7519-083-9

ISSN 0345-7524

Distributed by:

Division of Machine Design

Department of Management and Engineering

Linköping University

SE-581 83 Linköping, Sweden

Printed in Sweden by LiU-Tryck, Linköping 2015.

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

Abstract

This thesis investigates how to use optimization efficiently when complex products are developed. Modelling and simulation are necessary to enable optimization of products, but here it is assumed that verified and validated models of the products and their subsystems are available for the optimization. The focus is instead on how to use the models properly for optimization.

Knowledge about several areas is needed to enable optimization of a wide range of products. A few methods from each area are investigated and compared. Some modifications to existing methods and new methods are also proposed and compared to the previous methods.

These areas include

- Optimization algorithms to ensure that a suitable algorithm is used to solve the problem
- Multi-Objective Optimization for products with conflicting objectives
- Multi-Disciplinary Optimization when analyses from several models and/or disciplines are needed
- Surrogate Models to enable optimization of computationally expensive models

Modern frameworks for optimization of complex products often include more than one of these areas and this is exemplified with the industrial applications that are presented in this thesis, including the design and optimization of industrial robots and aircraft systems.

Keywords: Optimization, Surrogate Models, Product Concept Optimization, Multi-Objective Optimization, Multi-Disciplinary Optimization

Acknowledgements

This thesis could not have been written without the incredible help of numerous individuals and organizations.

I would first and foremost like to thank my supervisor, professor Johan Ölvander, for the much appreciated guidance I have received during these years. You have endured all of my notions and managed to guide me on this journey.

I would further like to thank my co-supervisor, professor Petter Krus, for all the fruitful discussions that we have had.

Thanks also goes to my industrial partners at ABB Corporate Research and SAAB AB and especially Xiaolong Feng and Sören Steinkellner. During our cooperation, we have investigated how the methods in this thesis can be used to analyse and optimize complex products from the industry.

I would also like to acknowledge the financial support that I have received from the European Union seventh framework programme Crescendo (no. 234244) and the Vinnova project IMPOz (project no. 2013-03758).

Much appreciation should also go to my colleagues at the divisions of Machine Design and Fluid and Mechatronic Systems for the excellent community and working climate that promotes inspiring work.

Last, but not least, I would like to thank my family and friends for always supporting me and being there when I needed you. This thesis and the research herein could not have been produced without you.

Papers

This thesis is based on the following five 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], the first author is the main author, whereas the second author is responsible for all parts regarding surrogate modelling. The first author has conducted the research and written papers [II] to [V].

- [I] Mehdi Tarkian, Johan Persson, Johan Ölvander, and Xiaolong Feng. “Multidisciplinary Design Optimization of Modular Industrial Robots by Utilizing High Level CAD Templates”. In: *Journal of Mechanical Design* 134 (2012). DOI: 10.1115/1.4007697.
- [II] Johan Persson and Johan Ölvander. “Comparison of different uses of metamodels for robust design optimization”. In: *51th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Paper no AIAA-2013-1039, 7-10 Jan, Grapevine, Texas*. 2013.
- [III] Johan A Persson and Johan Ölvander. “Optimization of the Complex-RFM optimization algorithm”. In: *Optimization and Engineering* (Feb. 2014), pp. 1–22. DOI: 10.1007/s11081-014-9247-9.
- [IV] Johan A Persson and Johan Ölvander. “Comparison of Different Methods for Robust Optimization in Engineering Design”. Submitted to *Optimization and Engineering*.
- [V] Johan Persson, Xiaolong Feng, Daniel Wappling, and Johan Ölvander. “A Framework for Multi-Disciplinary Optimization of a Balancing Mechanism for an Industrial Robot”. Submitted to *Journal of Robotics*.

Additional papers and publications

The following papers and publications are not part of the thesis, but constitute an important part of the background.

- [VI] Johan Persson and Johan Ölvander. “Comparison of Sampling Methods for a Dynamic Pressure Regulator”. In: *49th AIAA Aerospace Sciences Meeting, 4-7 Jan., Orlando, Florida*. 2011.
- [VII] Mehdi Tarkian, Johan Persson, Johan Ölvander, and Xiaolong Feng. “Multidisciplinary design optimization of modular Industrial Robots”. In: *Proceedings of the ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2011, 28-31 August 2011, Washington, DC, USA*. 2011, pp. 867–876.
- [VIII] Johan Persson and Johan Ölvander. “A Modified Complex Algorithm Applied to Robust Design Optimization”. In: *13th AIAA Non-Deterministic Approaches Conference, Paper no AIAA-2011-2095, 4-7 April, Denver, Colorado*. 2011.
- [IX] Johan Persson. “Design and Optimization under Uncertainties: A Simulation and Surrogate Model Based Approach”. Licenciate Thesis, Linköping University. 2012.
- [X] Johan Persson, Xiaolong Feng, Daniel Wappling, and Johan Ölvander. “Multi-Disciplinary and Multi-Objective Optimization of a Hydro-Pneumatic Balancing Cylinder for an Industrial Robot”. In: *12th ASME Biennial Conference on Engineering Systems Design and Analysis, ESDA2014, 25-27 June, Copenhagen, Denmark*. 2014.
- [XI] Johan Persson, Xiaolong Feng, Daniel Wappling, and Johan Ölvander. “Optimal Design of a Balancing System for a Serial Industrial Robot”. In: *The 14th Mechatronics Forum International Conference, 16-18 June, Karlstad, Sweden*. 2014.

Contents

1	Introduction	1
1.1	Thesis scope	3
I	Frame of Reference	5
2	Optimization Algorithms	7
2.1	Complex-RF	8
2.2	Genetic Algorithms	9
3	Multi-Objective Optimization	11
3.1	A priori methods	12
3.2	A posteriori methods	13
3.3	Interactive methods	13
3.4	Interpreting the Pareto front	13
4	Multi-Disciplinary Optimization	15
4.1	Methods for Multidisciplinary Optimization	16
4.1.1	Multidisciplinary Feasible-MDF	17
4.1.2	Bi-level Integrated System Synthesis - BLISS	18
5	Surrogate Models	21
5.1	Surrogate Model Types	22
5.1.1	Polynomial Response Surfaces	22
5.1.2	Kriging	23
5.1.3	Neural Networks	24
5.1.4	Radial basis Functions	25
5.2	Design of Experiments	27
5.2.1	Latin Hypercube Sampling	27

5.3	Accuracy estimations	28
5.3.1	Leave-k-out strategy	28
5.4	Accuracy measures	29
6	Surrogate Model Assisted Optimization	31
6.1	Surrogate Model Guided Optimization	32
6.2	Sequential Approximate Optimization	33
6.2.1	Infill criteria	33
7	Probabilistic Optimization	35
7.1	Estimating Statistical Entities	37
II	Contributions	39
8	Applications	41
8.1	Electric motorcycle	42
8.2	Overall design of an industrial robot	43
8.3	Balancing mechanism for an industrial robot	44
8.4	Dynamic pressure regulator	47
9	Accuracy of Surrogate Models	49
10	Improving the Complex-RF Method using Surrogate Models	53
10.1	Performance metric for comparison of optimization methods	56
10.1.1	Performance Index	56
11	Methods for Robust Design Optimization	59
12	MDO of a Balancing Mechanism for Industrial Robots	63
III	Conclusions	69
13	Discussion	71
14	Conclusions	75
	Bibliography	79

1

Introduction

Modelling and simulation is used extensively in the development process of complex products. It is often cheaper, faster and requires fewer resources than constructing prototypes. It is therefore desirable to create accurate models of the product and its subsystems. Accurate models can not only be used to evaluate the performance of a suggested design. They can also be used to investigate how robust and/or durable the product is and to find a product with optimal properties.

This is important in the conceptual phase of the product development process when it is desirable to know as much as possible about each possible concept. Optimizations of models of each concept can estimate the maximum potential of each concept. This information can then be used to aid selection of the promising concepts that are worth investigating more thoroughly [1].

This means that knowledge about modelling, assessment of the accuracy of a model and optimization is desirable for organizations and individuals that develop complex products.

The modelling of a system is a tedious process that depends on which property it should model. These properties can for example be performance, durability and cost, and the models can range from simple mathematical expressions to complex models that take weeks for a regular computer to evaluate once.

Models are usually undergoing a verification and validation process to assess the credibilities of their results [2]-[4]. It is usually too expensive to investigate exactly where the limits for the accuracy of the model are, which is why models should only be verified and validated for their intended purposes [5].

The modelling, verification and validation phases are outside the scope of this thesis. It is instead assumed that the accuracy of the models of the product and its subsystems has been proven. It should, however, be noted that it is important to know the assumptions that were used when the model was created and under which circumstances it is valid. There is otherwise a risk that the model is trusted to yield accurate results even when it is used outside of the range that the modeller intended.

It is often required that several objectives be evaluated when a product is optimized to ensure that a satisfactory design is obtained. These objectives might include cost, performance, durability, serviceability, manufacturability, alongside many others. These often conflicting objectives need to be handled in the optimization and this turns the problem into a Multi-objective optimization (MOO) problem [6].

Different models may be required in order to calculate the different objectives in the MOO problem. This further complicates the optimization, especially if information needs to be sent between the different models. An optimization where several disciplines need to be considered each time the objective function is evaluated is called a Multi-disciplinary optimization (MDO) [7].

Optimizations of complex products are often both multi-objective and multi-disciplinary, which makes them challenging for a computer to solve. Computer capacity has increased tremendously in recent decades, but this has partly led to models of higher fidelity. A simulation can consequently consume almost as much wall-clock time as before even though computational power has increased. This means that methods that can reduce the wall-clock times of optimizations are still in demand.

One common remedy for too computationally expensive models is to replace them with computationally efficient surrogate models [8]. These can range from simple polynomials [9] to models with numerous parameters that need to be tuned [10], but the models can be simulated in a few milliseconds. Optimization of high-fidelity models of complex products would be difficult to perform without surrogate model techniques.

Sóbester et al. [11] go as far as to claim that surrogate-based optimization is one of the most significant advances in engineering design technology. This emphasizes the importance of and the possibilities that the combination of surrogate models and optimization opens up for development of complex products.

Surrogate model techniques are recommended even more if probabilistic optimizations are to be performed. These optimizations try to find

design solutions that are insensitive to variations and/or have as high probability as possible of being successful. The statistical properties of a design need to be estimated each time the objective function value is to be calculated, which means that a probabilistic optimization is far more demanding than a regular one [12].

1.1 Thesis scope

This thesis focuses on how optimization could be used in an efficient manner to support the product development process. The increases in computer capacity in recent decades has enabled optimization of models and systems that were previously assumed to be too complex to simulate. There are, however, still many issues that need to be solved to enable optimization on an even greater scale.

The thesis has four main goals that need to be addressed.

1. Identify common methods and state-of-the-art of efficient optimization based on surrogate modelling.
2. Compare the different methods against each other to evaluate their strengths and weaknesses.
3. Modify promising methods to improve their performance.
4. Demonstrate the improvements by comparing the modified methods with their originals considering both mathematical test functions and real world engineering applications.

The comparisons of the different methods will increase understanding of their strengths and weaknesses. They will also indicate the most promising methods for performing optimizations of the complex products that are to be optimized in the context of this thesis.

Improved versions of the most promising methods are desirable since they would make optimization of complex products more feasible. The performance of the modified methods needs to be demonstrated in order to confirm that the modifications actually led to improvements.

The thesis is divided into three parts as follows:

- Part I - Frame of Reference. This part constitutes the scientific base and describes the methods used in the thesis and briefly presents other methods that could be used to achieve similar results.

- Part II - Contributions. The results from the appended papers are briefly presented in the contribution section and details can be found in the papers themselves.
- Part III - Conclusions. This part contains a general discussion on the methods and results in the thesis and also presents conclusions and directions for future work.

Part I

Frame of Reference

2

Optimization Algorithms

An optimization algorithm is used to automatically search the design space for an optimal solution to an optimization problem. The optimization problem can be formulated as in equation (2.1), where the aim is to minimize the objective function, $f(x)$, while ensuring that the p inequality constraints and the r equality constraints are fulfilled [6].

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & \\ g_j(x) \leq 0, \quad & j = 1, \dots, p \\ h_k(x) = 0, \quad & k = 1, \dots, r \end{aligned} \tag{2.1}$$

Numerous different optimization algorithms exist that are suitable for different types of problems. The user should typically choose an algorithm depending on available computational budget, type of problem, ease of use, availability and familiarity.

Many algorithms can handle constraints automatically, but not all. Constrained optimization problems can still be solved for most algorithms, but the constraints need to be managed by the user if the algorithm does not handle it. One example is to add a penalty to the objective function when a constraint is violated [6].

Optimization algorithms are divided into different classes depending on the operations that they perform. The gradient-based algorithms usually use gradients or Hessians to converge. These are deterministic

in the sense that they always converge to the same solution if they are run with the same settings and starting point. They usually converge fast but are prone to end up in local optima.

Population-based algorithms have been used extensively in product development in recent years and Particle Swarm Optimization (PSO) [13] and Genetic Algorithms [14], [15] in particular. These algorithms are stochastic and might therefore produce different solutions even though the optimizations are run with the same settings. Common for these methods are that they use tens of individuals that either move between each iteration or are replaced by new individuals when a new generation is created. They therefore require many objective function evaluations to converge, but the upside is that they usually do not become stuck in local optima due to their intelligence. The optimization time can be reduced significantly if tens of computer cores are available since each individual in each iteration/generation can be evaluated independently of the others.

The two optimization algorithms that are used most in this thesis are the Complex method and Genetic Algorithms. These are described in more detail in the following sections.

2.1 Complex-RF

The Complex-RF optimization algorithm [16] is a modified version of the Nelder-Mead Simplex optimization algorithm [17]. It begins by placing k points randomly in the design space, as shown in the schematic workflow in figure 2.1. These points constitute the Complex. The objective function values of the k points are evaluated and the worst point identified. The point is then reflected through the centroid of the other points, to the other side of the complex. An evaluation of the objective function value is then made and if the point is still worst, it is moved towards the middle of the Complex. When it is no longer worst, the point that was previously second worst is moved. This process continues until a stop criteria is met or the maximum number of function evaluations is reached.

The Complex can both move and contract and the algorithm has been demonstrated to be effective in optimizing simulation models [18]-[20]. An increased number of points in the Complex increases its accuracy, but this comes at the cost of more function evaluations. The standard setting of k is therefore twice the number of variables in the case of more

Figure 2.1 Flowchart for the Complex-RF optimization algorithm

than one variable.

The R and F in Complex-RF stand for randomization and forgetting factor, respectively. The first makes it harder for the algorithm to get stuck by allowing a moved point to be placed more randomly. The forgetting factor makes the latest point more important to the algorithm by gradually worsening the objective function values of older points. More details regarding Complex-RF can be found in [18] and a modified version of the Complex algorithm is presented in paper [III].

2.2 Genetic Algorithms

A genetic algorithm mimics breeding or Darwin's principle of the survival of the fittest [14]. It uses a population with a given size and

simulates the evolution of the population during numerous generations. A schematic workflow can be shown in figure 2.2.

Figure 2.2 *Flowchart for a simple genetic algorithm*

The objective function values of all individuals in the current generation are evaluated and the best individuals are selected for mating. The next generation will therefore predominantly consist of individuals that inherit properties of the best individuals from the previous generation. It is common to allow the absolute best individuals survive to the next generation to ensure that the best solution in the next generation is at least as good as the best from the previous one.

It is desirable to prevent a premature contraction of the area that the algorithm operates in. This is achieved by allowing some individuals to undergo mutation. Most of the individuals will consequently explore the area of the design space where the algorithm currently operates, while a few individuals will explore other parts of the design space [15].

3

Multi-Objective Optimization

It is important to be able to handle optimizations where several conflicting objectives are present. One example might be to maximize the speed of an industrial robot while minimizing its cost. The optimization problem then turns into a Multi-Objective optimization (MOO) problem.

The goal of an MOO is usually to find either one or many solutions that is an optimal compromise between the different objectives. These solutions are called Pareto optimal if they are not dominated by any other solutions. This means that no other solution has better values for every objective than a Pareto optimal point. This is exemplified for two objectives in figure 3.1 where the circles represent Pareto optimal points. The asterisk represents a solution that is not Pareto optimal since it is dominated by six circles.

Many methods that solve these kinds of problems have been proposed and they can be divided into a priori, a posteriori and interactive methods [6]. Their meanings and advantages and disadvantages are presented in the following sections.

Figure 3.1 Example of Pareto optimal points (represented by circles) and a dominated point (represented by an asterisk).

3.1 A priori methods

Methods that belong to the a priori category need information from the decision-maker before the optimization is begun. This might be the relative importance of the objectives or their target values. This information turns the MOO problem into one or several single objective optimization (SOO) problems and is therefore comparatively easy to solve. Examples of such methods are the utility function method [21], the lexicographic method and the goal programming method [22].

A drawback with this method is that only one or a few solutions are presented by the optimization. This means that it is important that the information given before the optimization is started is accurate. But there might be cases where a small worsening of one objective leads to larger improvements for one or more of the other objectives. This information is usually unavailable beforehand and it will probably not be gained during the optimization [23].

3.2 A posteriori methods

A posteriori methods are methods that try to find as much of the Pareto front as possible and then the decision-maker chooses solutions from the revealed front. This means that the decision-maker has a better decision support compared to a priori methods since he or she can see the trade-offs between the different objectives. The main drawback is that it is much more computationally demanding to find the whole Pareto front compared to just a few solutions.

Two common a posteriori methods are the weighted sum method [6] and the E-constraint method [24]. Both transform the MOO problem into numerous SOO problems with different objective functions. The weighted sum method creates a linear combination of the multiple objectives and alternates the weightings for each objective. The E-constraint method turns all objectives except one into constraints. Different solutions may be obtained by alternating the constraint limits and which objective to optimize for.

There also exist optimization algorithms that are tailored for MOO. Population-based optimization methods, such as genetic algorithms, are capable of identifying Pareto fronts in one single optimization run. This might be achieved by ranking individuals based on Pareto dominance, as for example in MOGA [25], NSGAII [26] and SPEA2 [27].

3.3 Interactive methods

Interactive methods have also been proposed, where the decision maker can steer the optimization and explore the Pareto front interactively. One example is Nimbus, that was proposed by Miettinen and Mäkelä [28], [29]. It begins by finding a point on the Pareto front and the decision maker can then decide in which direction the algorithm should go to find additional Pareto optimal points. An early version of this approach is the STEM method [30].

3.4 Interpreting the Pareto front

The a posteriori methods present as many points as possible on the Pareto front since they do not know which part of the front the decision-maker prefers. This means that the decision maker is recommended to

Table 3.1 *Methods that can be used to graphically interpret the Pareto front.*

No. of Objectives	Methods
2	Regular plot of f_1 versus f_2
3	Bi-objective slices, Contour plot*
4-5	Animations/scrollbar, Matrix of decision maps*
6+	Heat map*, Parallel Coordinate Plot, Scatterplot matrix

evaluate all the designs on the Pareto front in order to find the best design according to his or her preferences.

It is also important to understand the Pareto front for the interactive methods to direct the algorithms in the best directions. But it is difficult to obtain a good overview of the Pareto front from a list. It is therefore desirable to present the front graphically [31]. This is not a problem in optimization problems with two objectives, but difficult when there are more than four. Information about the design variable space is also crucial and hard to understand and visualize. Even though the number of objectives is low, the parameter space is typically high-dimensional and hard to grasp.

Lotov and Miettinen [31] have given recommendations about different graphical methods that can be used for different numbers of objectives. This information is here compiled in table 3.1, with the methods preferred by the author of this thesis marked with an asterisk. It can be noted that the author prefers graphs where as many designs and objectives as possible are shown simultaneously.

4

Multi-Disciplinary Optimization

Complex products often require analyses from different disciplines to ensure that the suggested design is appropriate. Examples of disciplines in this context are solid mechanics (like FE-models), CAD models, System dynamics models and cost models. This turns the optimization problem into an MDO problem since it is desirable to take each discipline into account in the optimization [32]. An individual optimization for each discipline is not recommended since different disciplines and subsystems usually require conflicting configurations if they are optimized without any regard to other disciplines and subsystems [33]. They therefore often yield sub-optimal solutions.

An MDO problem can be extremely demanding for a computer to solve if there are numerous disciplines involved and especially if the models are computationally expensive to solve in themselves. It is therefore imperative to create effective frameworks when MDO problems need to be solved.

The architectures can be divided into two broad categories - monolithic and distributed architectures [7]. Monolithic architectures optimize the whole problem as one optimization problem. Distributed architectures instead divide the optimization problem into multiple sub-problems that can be solved individually and yield the same results as if they would have been solved as one large problem.

The choice of architecture should depend on the human and computing environment, the available algorithms and the problem that should be solved [7]. A distributed architecture is preferred if the disciplines of

the problem easily can be run in parallel and multiple computer cores are available. This is common in the industry where different engineering groups often are responsible for their own part of the products and/or analysis, even if the companies do not perform any MDOs. Each engineering group solve their problem on their own, using the methods that they prefer, which require the groups to report to a central functional that harmonize solutions from the different groups [34].

Every discipline is analysed the same number of times in a monolithic architecture. This is normally not the case for distributed architectures if they are built properly [7]. The fastest analyses would have to wait for the slowest if each discipline is run in parallel, which would be a waste of time. This extra time can instead be used to perform several analyses of the fast disciplines while one analysis of the slowest analysis is performed. Distributed architectures therefore typically require more objective function evaluations to reach the optimum [35].

A general guideline when designing a framework is to try to place a fast evaluated analysis as early as possible. This analysis can early indicate whether the suggested design is good or not. There is no use to perform a computationally demanding analysis if the design will be inadequate anyway. The rest of the analyses can instead be omitted and the suggested design given a penalty to its objective function value to indicate to the optimization algorithm that the solution was inadequate.

It is also common to speed up the MDO process by replacing computationally demanding models with efficient surrogate models, see for example [36] and [37]. This will not change the MDO architecture since the surrogate model just takes the place of the original model in the framework [7].

4.1 Methods for Multidisciplinary Optimization

A recent survey regarding framework architectures for MDO has been made by Martins and Lambe [7]. They classify the methods and present them together with diagrams and lists of advantages and disadvantages. Some of the methods may also be referred to by other names, but this chapter follows the notations by Martins and Lambe. The following classifications are made by them:

- Early works by Shmit [32] and Haftka [38].

Monolithic architectures

- The All-at-once problem statement (AAO) [39]
- Simultaneous Analysis and Design (SAND) [40]
- Individual Discipline Feasible (IDF) [39]
- Multidisciplinary Feasible (MDF) [39]

Distributed architectures

- Concurrent Subspace Optimization (CSSO) [41]
- Collaborative Optimization [42]
- Bi-level Integrated System Synthesis (BLISS) [43]
- Analytical Target Cascading (ATC) [44]
- Exact and Inexact Penalty Decomposition (EPD and IPD) [45]
- MDO of Independent Subspaces (MDOIS) [46]
- Quasiseparable Decomposition (QSD) [47]
- Asymmetric Subspace Optimization (ASO) [48]

Details regarding the frameworks can be found in the corresponding paper. This chapter only describes two of them briefly. The two architectures that are used in the papers in the thesis are MDF [39] and BLISS [43]. MDF is used in paper [V] and BLISS in paper [I].

4.1.1 Multidisciplinary Feasible-MDF

The Multidisciplinary Feasible (MDF) [39] is an architecture that can be used if all analyses or disciplines are used in series. It has also been referred to as Fully Integrated Optimization [49] and Nested Analysis and Design (NAND) [50].

A simplified schematic of an MDF with three analyses is shown in figure 4.1. $\mathbf{x}$ represents the optimization variables and $\mathbf{y}_i$ the outputs from analysis i . The objective function handles the outputs from the analyses and use them to evaluate the constraints $g_j(\mathbf{x})$ and the objective function value $f(\mathbf{x})$.

The benefit with MDF is that it is the simplest of the monolithic architectures to create and solve since the analyses are performed sequentially. There is one major drawback however - the wall clock time.

Figure 4.1 Simplified schematic of a Multidisciplinary Feasible architecture with three analyses.

Every analysis needs to be performed in sequence each time the objective function value of a suggested design should be calculated since no parallelization is possible. This can however be remedied with surrogate models and/or the placement of a fast analysis early to see if there is any point in performing the expensive analyses. Additional advantages and drawbacks are presented in [7].

4.1.2 Bi-level Integrated System Synthesis - BLISS

Paper [I] uses a slightly modified version of the Bi-level Integrated System Synthesis (BLISS) [43], but the original version is presented here.

BLISS decomposes the problem into different levels. A simplified schematic of a BLISS architecture with two levels and three subsystems is shown in figure 4.2. The optimization variables, x_0 are sent to the top level by the optimization algorithm. These values are then sent as constants to the optimizer of each subsystem. Each subsystem, index k , has its own local variables, x_{ki} , and an optimization of its variables is performed by the optimizer to receive optimal properties. The out-

Figure 4.2 Simplified schematic of a Bi-level Integrated System Synthesis (BLISS) architecture with three analyses.

puts from each optimized subsystem, y_{ki} are sent to the top level where they are handled to construct the objective function, $f_0(x_0)$. This corresponds to the Collaborative Optimization [42] if the subsystems are decoupled. BLISS is distinguished from CO due to the coupled variables, c_i , which means that information needs to be sent between the subsystems.

5

Surrogate Models

Surrogate models [8] are a common name for computationally efficient mathematical models that are used to model how an entity varies when the parameters that affect the entity are changed. The models can also be referred to as metamodels [51], [52], but this name is also given to information about a model. Surrogate model is a more suitable name since the models' most common use is to replace computationally expensive models.

Surrogate models can also be used to create models of the results of experiments. One example is the Kriging surrogate model that has been used to model how the ore grade in the ground varies with the location [53]. These models are created from drill core samples that have been collected from various locations in the landscape.

A surrogate model is usually adapted as accurately as possible to a collection of data points or samples by either an optimization or estimation of its parameters. The model can then be used to predict what the outcome of an experiment with other settings would be.

An advantage with using surrogate models is that almost no assumption or knowledge of the problem is needed. One surrogate model type is naturally better suited than the others for a given problem, but the chosen surrogate model will be adapted to the sample data regardless of the mechanics behind the data. An optimization of the surrogate model's parameters using an optimization algorithm generally leads to a more accurate surrogate model, but this requires more time than an estimation.

5.1 Surrogate Model Types

Numerous types of surrogate models have been proposed and some of the most popular are presented in the following subsections. Several comparisons of surrogate model types have been made [I], [54] but no general conclusion can be drawn regarding which is generally best [51].

Both Simpson et al. [52] and Forrester et al. [8] have, however, given recommendations regarding when to use which surrogate model type and what their advantages and disadvantages are.

5.1.1 Polynomial Response Surfaces

One of the most intuitive surrogate models is the polynomial response surface (PRS) [9]. A PRS tries to estimate the value of an entity as a polynomial of arbitrary degree as shown in equation (5.1) for N variables and a second order polynomial.

$$\hat{y} = \beta_0 + \sum_{i=1}^N \beta_i x_i + \sum_{i=1}^N \sum_{j=1}^N \beta_{ij} x_i x_j \quad (5.1)$$

This can be written in matrix form according to equation (5.2)

$$\hat{y} = \mathbf{X}_v \beta \quad (5.2)$$

where

$$\mathbf{X}_v = \begin{bmatrix} 1 & x_1 & x_2 & \cdots & x_N & x_1 x_1 & x_1 x_2 & \cdots & x_N x_N \end{bmatrix},$$

$$\beta = \begin{bmatrix} \beta_0 & \beta_1 & \beta_2 & \cdots & \beta_N & \beta_{11} & \beta_{12} & \cdots & \beta_{NN} \end{bmatrix}^T$$

Here, x_i stands for the value of the i^{th} variable of the design point that is being evaluated. The coefficients β_i are the parameters that need to be tuned to use this surrogate model. The fitting of the PRS to the data means that the values of the coefficients, β_i , are determined. This can be done in a least square sense by solving the matrix system shown in equation (5.3).

$$\mathbf{X}\beta = \mathbf{y} \quad (5.3)$$

where

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_v(s_1) \\ \mathbf{X}_v(s_2) \\ \vdots \\ \mathbf{X}_v(s_m) \end{bmatrix} \mathbf{y} = \begin{bmatrix} y(s_1) & y(s_2) & \cdots & y(s_m) \end{bmatrix}^T$$

Each row in this matrix system is a realization of equation (5.1) for one of the m samples. The matrix system can be solved according to equation (5.4).

$$\hat{\beta} = (\mathbf{X}^t \mathbf{X})^{-1} \mathbf{X}^t \mathbf{y} \quad (5.4)$$

The benefit of a PRS is that the modeller can choose which degree of the polynomial should be used. A higher order polynomial means that a more advanced behaviour can be modelled. But the required number of samples in order to fit the model increases significantly with the number of variable squared. This can be seen in equation (5.5) where the number of required samples, n_{req} , is shown for a second order polynomial with N variables. This means that PRS is unsuitable for problems with many variables [55].

It is also recommended to oversample by 50% [56] or 100% [57], which means that even more samples are needed.

$$n_{req} = 1 + 2N + (N - 1) \frac{N}{2} \quad (5.5)$$

5.1.2 Kriging

Kriging is an interpolating surrogate model that also takes global trends into account. It was proposed by Daniel Krige [53] as a method to use in geostatistics and was introduced in engineering by Sachs et al. [58]

Isaaks and Srivastava [59] have written a book that introduces Kriging from the geostatistical point of view and the interested reader is encouraged to read it. Here follows a short description of Kriging in its original form.

Kriging can combine local deviations with a polynomial model of the form shown in equation (5.6) [60].

$$\hat{y} = \sum_{i=1}^k \beta_i f_i(x) + Z(x) \quad (5.6)$$

The first term is similar to a polynomial response surface and can be chosen, whereas the second term ($Z(x)$) is a stochastic process with a mean value of zero and a spatial correlation function of the form shown in equation (5.7).

$$Cov [Z(x_i), Z(x_j)] = \sigma^2 R(x_i, x_j) \quad (5.7)$$

σ^2 is the variance of the process and $R(x_i, x_j)$ is a correlation function between x_i and x_j . The parameters of the correlation function need to be determined in order to fit the Kriging model and this is a time-consuming process since an optimization is needed for good accuracy. This can for example be resolved using maximum likelihood estimators as Martin and Simpson describe [61].

One benefit of Kriging is that it is possible to estimate the uncertainty in its estimations [8]. This uncertainty usually increases as the distance to the closest sample that was used to create the surrogate model is increased, which is to be expected.

It is important to know that the correlation matrix can be singular if the samples that are used to fit the Kriging model are placed too closely together [54]. This is especially true for sequential updating schemes, such as the ones described in section 6, where new points are added to the surrogate model automatically.

Anisotropic Kriging can handle global trends in different directions in the design space [62]. It is also stationary in its original version, which indicates that it is most suitable for modelling smooth functions where the value does not vary too much in the design space. Non-stationary Kriging methods have been proposed by for example Toal and Keane [63].

5.1.3 Neural Networks

Neural Networks (NNs) were introduced by McCulloch and Pitts [10] and mimics the human brain and its processing of information. Figure 5.1 has been adapted from Aspenberg [64] and shows an NN with an input layer, an output layer and two hidden layers. Each circle is a neuron and contains a rather simple transfer function. The process of fitting an NN to samples strives to determine the shape of the transfer functions.

One commonly used transfer function is the sigmoid function that is shown in equation (5.8).

Figure 5.1 *Schematics of a simple Neural Network.*

$$f(a) = \frac{1}{1 + r^{-a}} \quad (5.8)$$

The layers and transfer functions make the NN extremely flexible, which makes it suitable to reanimate non-linear behaviour. The drawback is that the fitting of the model may take time since an optimization of its transfer functions needs to be performed.

5.1.4 Radial basis Functions

Radial Basis Functions (RBF) can be seen as a Neural Network with certain settings [65] and was introduced by Broomhead and Lowe [66].

The value of a point is interpolated from the values of known points as a function of the Euclidean distances to the points [67]. This is exemplified in equation (5.9) for a case where n points are known.

$$\hat{f}(x) = \sum_{i=1}^n \omega_i \phi_i(\|x - x_i\|) + \sum_{j=1}^m \beta_j f_j(x) \quad (5.9)$$

The last term is a polynomial response surface when there are k variables, which means that RBF can handle global trends [68]. ϕ_i is a function of the Euclidian distance between the point whose value should be estimated and the known point with index i .

The creation of an RBF means that the distance function and the weights, ω_i , and the coefficients, β_j , need to be determined. These values need to be optimized to achieve as high accuracy as possible for the RBF.

$$\phi(r) = e^{-\frac{r^2}{s^2}} \quad (5.10)$$

Numerous different distance functions that can be used exist and a Gaussian function of the form shown in equation (5.10) is used in the thesis. r is the Euclidian distance between the two points and s a parameter that needs to be determined when or before the RBF is created.

The reason for choosing the Gaussian distance function is that Nakayama [69] presents an analytical expression that can be used to approximate a value for s . This approximation is shown in equation (5.11), where d_{max} is the distance between the two points that are furthest from each other, n is the number of known points and m is the number of variables.

$$s = d_{max} (n \cdot m)^{-\frac{1}{m}} \quad (5.11)$$

An optimization of the parameters of the RBF can be avoided thanks to the fact that s can be estimated with equation (5.11). The weights, ω_i , and the coefficients, β_j , can thereby be calculated by solving the matrix system in equation (5.12) [68].

$$\begin{bmatrix} \phi & \mathbf{P} \\ \mathbf{P}^t & 0 \end{bmatrix} \begin{Bmatrix} \boldsymbol{\omega} \\ \boldsymbol{\beta} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f} \\ 0 \end{Bmatrix} \quad (5.12)$$

ϕ is a $n \times n$ -matrix that contains the distance function values for every pair of sample points. The other matrices are shown in equation (5.13).

$$\mathbf{P} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}, \quad \boldsymbol{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_n \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{bmatrix} \quad (5.13)$$

The estimation of s from equation (5.11) ensures that the RBF can be created by solving equation (5.12) since the ϕ -matrix includes no unknown parameters. This can be done rather simply with a least square estimation such as the one shown in equation 5.4. The accuracy will be lower than if the parameters of the RBF had been optimized with an optimization algorithm, but it is often convenient to speed up the process by avoiding an optimization.

5.2 Design of Experiments

It is desirable to achieve a surrogate model with as high accuracy as possible given a number of samples. It is therefore recommended that a sampling be used that specifies which samples should be drawn and this field is called Design of Experiments (DoE). Numerous DoEs [9] that can be used with different criteria exist, but this thesis mainly uses Latin Hypercube Sampling (LHS) [70]. LHS is also used as DoE for simulation models by many other authors [12], [57], [68].

5.2.1 Latin Hypercube Sampling

LHS divides each variable range into n parts of equal size [70]. n samples are then drawn with the requirement that only one sample should be found in each part of each variable. This is demonstrated for three samples and two variables in figure 5.2. It is shown that one sample (a cross in the figure) is placed in each row and column, which means that the requirements for LHS are fulfilled.

A diagonal sampling plan would fulfil the requirements of LHS, but it is usually unsatisfactory since no information about the other parts of the design space is gained by the samples. This can be avoided by setting additional requirements for LHS as well. One example is the maximin criterion, which maximizes the distance between the two samples that are closest to each other. The n samples will consequently be evenly spread around the design space.

A drawback with LHS is that it is difficult to add samples to an existing sampling plan without violating its current properties [71]. It is therefore recommended to double (or use another multiple) the number of samples if the properties should be preserved.

Figure 5.2 A Latin Hypercube sampling plan consisting of three samples for two variables.

5.3 Accuracy estimations

It is desirable to measure the accuracy of the surrogate model to estimate the credibility of its results. This can be done in several ways, for example by calculating how well the surrogate model is fitted to the data that was used to create it. This is usually not recommended and this is especially true for interpolating surrogate models since they will have an accuracy of 100 % at those points [68].

It is instead recommended that the accuracy of the surrogate model be assessed by comparing its predictions to another data set that was not used to fit the surrogate model. Either new samples need to be collected or the leave-k-out strategy can be used [72].

5.3.1 Leave-k-out strategy

The leave-k-out strategy removes k points from the dataset and fits a surrogate model to the remaining sets. The removed points are then used to assess the accuracy of the surrogate model. Different surrogate models are received when different points are removed and the most

promising surrogate model according to the accuracy measures can be chosen.

Meckesheimer et al. [73] perform experiments with different values of k for different types of surrogate models and present general guidelines and recommendations for the size of k .

5.4 Accuracy measures

Below follow a few accuracy measures that can be used to measure how well the predictions from the surrogate model agree with the original model or experiments. y_i is the value of the original model at point i , whereas $\hat{y}_i$ is the value predicted by the surrogate model. $\bar{y}$ is the mean value and n is the number of points in the dataset that is used for the comparison.

The measure that is used most in the thesis is the Normalized Root Mean Squared Error (NRSME), presented in equation (5.14). It is in the form of a percentage and should be as low as possible.

$$NRSME = \sqrt{\frac{\sum_{i=1}^n \left(\frac{y_i - \hat{y}_i}{y_i} \right)^2}{n}} \quad (5.14)$$

Another accuracy measure is the R square, which measures the overall accuracy of the model [57]. It can be seen in equation (5.15) and should be as close to one as possible [74].

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (5.15)$$

The relative average absolute error (RAAE) is also a metric for the overall accuracy of the surrogate model [74]. Its equation is shown in equation (5.16) and the accuracy is higher the closer the RAAE is to zero.

$$RAAE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (5.16)$$

The equation for the Relative Maximum Absolute Error (RMAE) is shown as equation (5.17) [74]. It indicates how large the largest error between the estimated and true value is. This is a local measure and should be as low as possible [57]. This metric can show a large local error even though the other metrics display high accuracies, but the other metrics are usually more important [54]. The RMAE is then just a warning that large local errors might occur.

$$RMAE = \frac{\max\{|y_1 - \hat{y}_1|, |y_2 - \hat{y}_2|, \dots, |y_n - \hat{y}_n|\}}{\sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (5.17)$$

6

Surrogate Model Assisted Optimization

The combination of surrogate models and optimization is a powerful tool when concepts of complex products are to be evaluated using modelling and simulation [11]. The most widely used method that involves surrogate models in the industry today is probably to create a surrogate model of a computationally demanding model and then perform an optimization on the surrogate model. This unfortunately puts high demands on the surrogate model since it needs to be accurate in the whole design space due to the fact that the location of the optimum is unknown before an optimization has been performed.

Recent methods instead focus on creating an initial surrogate model by using part of the computational budget, and then gradually improving the surrogate model during the optimization [8]. The benefit of these methods is that most expensive simulations of the original model are made in the parts of the design space where the optimization algorithm actually operates. This minimizes the number of unnecessary simulations and improves the accuracy of the surrogate model in the parts of interest of the design space.

The problem with these methods is that the surrogate model and thereby optimization can be misguided by particularly deceptive objective functions. This means that the initial surrogate model needs to reflect the overall appearance of the original model quite well to avoid a

misguided optimization.

These methods can be divided into two categories - Surrogate Model Guided Optimization and Sequential Approximate Optimization.

6.1 Surrogate Model Guided Optimization

Optimization methods and algorithms that belong to this category perform one optimization with a surrogate model that is updated during the optimization.

One example is the GA, proposed by DuVigneau and Praveen [67]. It begins by evaluating the fitness of the original population by simulating the original model and the 20% best individuals are selected for mating. An RBF is also fitted to the individuals from the first generation. The RBF is used to give an estimation of the fitness value of all new individuals that are created. The 30% most promising individuals in each generation are then evaluated using the original model and the 20% best individuals selected for mating. The RBF is also updated with the simulations of the original model. Ideally, this means that the accuracy of the GA is maintained, whereas the required number of simulations of the original model is reduced by approximately 70%.

Safari [75] proposes a surrogate model guided particle swarm optimization. The algorithm is called MGPSO (Meta-model guided particle swarm optimization) and uses PCA-HDMR as surrogate model. This is a surrogate model type that has been developed specifically to re-animate problems with many variables [76]. The algorithm stores the function evaluations that the PSO performs and iteratively creates surrogate models from these samples. The minimum of the surrogate model is then added to the velocity update function for PSO as an acceleration term. The updating of a particle's direction and velocity will consequently depend on the the particle's present velocity and direction, its own memory, the swarm intelligence and the minimum of the surrogate model. The algorithm will perform an equal amount of objective function evaluations as a regular PSO, but performs better thanks to the surrogate model intelligence.

There also exist optimization algorithms that use surrogate models to reduce the design space [75]. They either try to identify unpromising regions or optimization variable ranges. These can then be discarded, which reduces the design space. One example is the space exploration and unimodal region elimination algorithm (SEUMRE) [77]. It begins

by drawing samples in the design space and the algorithm then divides the design space into several regions that are ranked after how likely it is that the optimum is located in the region. Additional samples are drawn in the most promising region and a surrogate model of the region created. An optimization of the surrogate model is then performed to find the optimal solution in the region. The process can then continue with the second most promising region, and then the third, and so on until the decision maker is satisfied with the results.

6.2 Sequential Approximate Optimization

Sequential Approximate Optimization (SAO) is one of the most efficient methods to optimize computationally demanding models [8], [78]. A schematic of its basic operations is shown in figure 6.1. It begins by drawing samples according to a DoE and fits a surrogate model to the samples. An optimization is then performed to find the optimum value of the surrogate model. A simulation of the original model with parameter values according to the optimization is performed and the surrogate model updated with this sample. An optimization is then performed to find the optimum of the updated surrogate model. This process continues until a stop criterion is met.

This method is not suitable for problems that are extremely fast to evaluate since an optimization is performed between each simulation of the original model.

An optimization algorithm with high accuracy is recommended for optimizing the surrogate model, even if it requires many objective function evaluations, since all of them are performed on the surrogate model.

The ratio between how many points should be used for the initial model and the number of infill points has been investigated by for example Reischel and Lesieur [79].

6.2.1 Infill criteria

The surrogate model is generally improved as the SAO progresses. But it can still be misguided by deceptive objective functions or badly placed initial samples [78]. This can in some ways be remedied by updating the surrogate model with points not only at its optimum but also at other places to ensure that no important region is neglected.

Figure 6.1 A flowchart for a simple sequential approximate optimization method.

Both Jones [78] and Forrester et al. [8] demonstrate this for different types of infill criteria and deceptive objective functions. Below follow some proposed infill criteria that they present.

- Maximum distance from the closest known point
- Maximum uncertainty in the surrogate model
- Maximum estimated improvement of the value predicted by the surrogate model [80]

The maximum distance from the closest known point is quite an intuitive infill criterion since it is probable that the surrogate model has a low accuracy in areas where not much is known. It is also quite easy to implement and can be used with any type of surrogate model. The two last criteria can be used with Kriging since it has the capability to estimate both their uncertainty and expected improvement.

7

Probabilistic Optimization

Computer models are usually deterministic [52], but most processes and systems in real life are prone to be affected by stochastic uncertainties. Real life systems and products are affected by changing working environments, material variations, tolerances and many more types of variations and uncertainties. This means that methods are needed that can estimate how the uncertainties and variations affect the products.

The methods that handle uncertainties in optimization are usually divided into two categories depending on the goal of the optimization [64]. Robust Design Optimization (RDO) tries to find an optimum solution that is insensitive to uncertainties and variations [55], whereas Reliability-Based Design Optimization (RBDO) tries to minimize the probability of failure. This can be exemplified with figure 7.1, where A is a deterministic optimum and B a more robust optimum. C is the most reliable solution if the red line is the required performance to avoid failure and the spread in variable x of a design that is as wide as the grey field.

The objective function in RDO is usually a linear combination of the mean value, μ , and the standard deviation, σ , of the desired system characteristic [12]. This is exemplified in equation (7.1) where the weights, w_1 and w_2 , are used to balance expected performance versus robustness.

$$\min f(x) = w_1\mu + w_2\sigma \quad (7.1)$$

The randomness needs to be introduced into the analysis to enable

Figure 7.1 A deterministic, A, robust, B, and reliable, C, optimum when the function is to be minimized.

estimation of expected value, standard deviation and/or probability distributions of the desired system characteristic since the computer models are deterministic.

This can be done by assigning probability distributions to the parameters of the model to represent the variation that occurs in the real physical system [55]. Different simulations of the model will show different results when random parameter values are chosen for each simulation. Multiple simulations are then run with model parameter values sampled from the assigned distributions to estimate the distributions of the system characteristics.

It is also possible to add random noise to the output of the model to represent model uncertainty [81]. It is most common, however, to assign the uncertainty as two bounds after the statistics from the model have been calculated [82]. This creates a p -box, as exemplified in figure 7.2 [83]. The uncertainties and errors in the model mean that the possible system characteristic values from the design lie between the two dashed lines.

Figure 7.2 *An example of a p-box*

7.1 Estimating Statistical Entities

It is necessary to estimate some statistical entities to enable probabilistic optimization. The target is to either estimate the mean value and standard deviation in RDO or the probability distribution in RBDO. These entities can be estimated by methods such as FORM (first-order reliability methods) [84] or SORM (second-order reliability methods) [85], [86] or Polynomial Chaos Expansion [87].

The most renowned and used methods are the sampling-based methods, where the Monte Carlo Simulation (MCS) is the most commonly used. It resembles throwing a pair of dice numerous times to estimate the probability of rolling a six. This will converge towards the true distribution according to the law of large numbers, but the required number of samples is unrealistically large for MCS to be used in an optimization process [88], [VI].

It is instead common to use LHS (presented in section 5.2.1) to draw samples [71]. The probability distributions of the uncertainties are divided into n intervals of equal probability. One sample is then drawn from each interval.

The mean value, μ , of the samples can be estimated according to equation (7.2). y_i is the value of sample i and the number of samples is denoted n .

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n y_i \quad (7.2)$$

The standard deviation, s , can be estimated with either equation (7.3) or 7.4. The first can be used to estimate the standard deviation of the samples, whereas the second estimates the standard deviation of the phenomena that the samples were drawn from. This is akin to the difference between estimating the standard deviation of a number of individuals in a population and estimating the standard deviation of the population that the individuals belong to.

The difference between the two estimations is negligible for large samples, but equation (7.4) is most commonly used in the context of RDO since it is desirable to estimate the standard deviation of the system characteristics for a certain design, and not the standard deviation of the samples.

$$s_n = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{\mu})^2} \quad (7.3)$$

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (y_i - \hat{\mu})^2} \quad (7.4)$$

Part II

Contributions

8

Applications

The methods presented in this thesis have been used to optimize several different mathematical functions and applications. The applications are presented in this section and are represented by models of different levels of detail. This means that some models are used for overall design of a system, whereas some are used for detailed design.

The models are as follows:

1. A MATLAB Simulink [89] model of an electric motorcycle that is used to optimize the gearbox.
2. An MDO framework that is used to optimize the overall design of an industrial robot.
3. An MDO framework that is used to optimize a balancing mechanism for an industrial robot.
4. A Dymola [90] model of a dynamic pressure regulator found in aircraft.

The first and third are examples of an optimization of a component where it is analysed together with the rest of the product to ensure that the product receives optimal properties. The second represents overall design of a product on a conceptual level and uses therefore low-fidelity models. Detailed design of a component is represented in the last model, where a high-fidelity model is used in the optimization.

The mathematical functions used in this thesis can be found in works by for example Neculai [91]. Mathematical functions have the benefits that they are fast to evaluate and can be chosen and tailored to test

different properties of the methods [55]. It is for example often possible to scale them to a desired number of dimensions/variables.

8.1 Electric motorcycle

A simple conceptual engineering problem is included in the form of a dynamic simulation of an electric motorcycle. Figure 8.1 shows a Simulink block diagram of a model of an electric motorcycle. The model is based on physical equations and consists of four main components: the battery, the electrical motor, the gearbox and the actual motorcycle. The model can be used to simulate the velocity and operational range of the motorcycle as a function of elapsed time. More details can be found in paper [III].

Figure 8.1 *A Simulink model of an electric motorcycle*

The model is used to find the optimum gear ratios of a gearbox with two gears as described in papers [III] and [IV]. The gearbox also has a velocity at which it automatically switches between first and second gear. Different gear ratios are optimal depending on what properties the motorcycle is desired to have. The mathematical formulation of the optimization problem is shown in equation (8.1). The objective is to

maximize the velocity of the motorcycle after five seconds when it is accelerated from standstill, i.e. to obtain maximum acceleration.

$$\begin{aligned}
 \min \quad & f(x) = -V_5(x) \\
 \text{s.t.} \quad & \\
 x = & [u_1 \quad u_2 \quad v_{shift}] \\
 1 \leq & u_1 \leq 20 \\
 1 \leq & u_2 \leq 20 \\
 1 \leq & v_{shift} \leq 75
 \end{aligned} \tag{8.1}$$

8.2 Overall design of an industrial robot

An MDO framework is used in paper [I] to optimize the overall design of an industrial robot consisting of CAD models, a dynamic analysis, a solid mechanics analysis based on Finite Element Models and a cost analysis. The dynamic analysis calculates how the different links and actuators move during a robot's work cycle. It also calculates the forces that are exerted on the robot components during the cycle. These forces are sent to the solid mechanics analysis, which calculates the resulting stresses in the components.

The objective is to minimize the weight of the robot's links and the variables are the thickness of the material in the links and which actuators should be used. The thickness is represented by continuous variables whereas the choice of actuators are discrete variables. This mixture of variable types makes the modelling and optimization of the problem quite demanding. The CAD model needs to be both morphological and topological to enable the continuous and discrete parameters, respectively, to be handled.

The finite element model imports the geometry from the CAD model and an example of a robot link is shown in figure 8.2. The size of its top and bottom geometries is decided by the choice of actuator whereas the material thickness variable controls the cross-sectional area of the rest of the link. This is the most time-consuming analysis and it is therefore desirable to replace it with surrogate models. This is described in section 9.

Figure 8.2 *Finite Element Models of two robot links.*

8.3 Balancing mechanism for an industrial robot

Industrial robots that carry heavy loads usually have a balancing mechanism that reduces the torque exerted on the second axis, as shown in figure 8.3.

The balancing mechanism is an example of a torque reduction. The mechanism is a cylinder containing three concentric springs. The mounting positions of the mechanism is shown in 8.4. One end is fixed while the other is attached to the second axis. The position of the piston inside the cylinder will therefore change when the angular position of the second axis changes. This means that the compression of the springs will change and thereby the axial force in the piston, which in turn results in a change in the balancing torque.

It is desirable that the balancing torque change with the position of the robot since the position of the load carried determines the torque. An example of such a torque curve is shown in figure 8.5.

It is possible to use different concepts for the balancing mechanism. It can for example be electric or a balancing cylinder filled with gas or a fluid attached to an accumulator.

Figure 8.3 An ABB IRB6640 robot, courtesy of ABB.

Figure 8.4 Schematic of mounting positions A and B for a pushing cylinder.

Figure 8.5 *Example of a function of how the torque from the balancing mechanism varies when the angle of the second axis is changed.*

8.4 Dynamic pressure regulator

The application studied in paper [IV] is from the aircraft field where the Dynamic Pressure Regulator (DPR) is used to control the pressure in the cockpit or in the pilot's g-suit. It is important that the system is filled fast enough and to the correct pressure level. The time it takes to fill the ECS and the end pressure inside it are therefore the two objectives of interest for this application.

Figure 8.6 shows a screenshot of a Dymola [90] model of the DPR. Dymola is an object-oriented software that can be used to simulate dynamic behaviour of complex systems. This is comparable to MATLAB Simulink that was used for the electrical motorcycle problem.

Figure 8.6 Two screenshots from Dymola showing the Dynamic Pressure Regulator.

9

Accuracy of Surrogate Models

As stated in section 5.1, different surrogate models are suitable for different types of problems. It is therefore important to investigate which model should be chosen to model a certain entity. Such a comparison is made in paper [I] to find the most suitable model to replace the geometrical model in the MDO framework. Additional details regarding the comparison can be found there, but some of the results are also presented here as well. The best surrogate model according to the comparison should also be a good candidate for surrogate model for the other problems.

The test case is the geometrical model of a robot link and it is tested for the continuous variable link thickness and the discrete variable actuator choice. The outputs are the link's mass, moment of inertia and centre of gravity.

The surrogate models are fitted in modeFRONTIER [92] with the same 50 samples for each surrogate model type, and their accuracies are estimated on 20 additional samples. They are tested for the accuracy measures NRSME, RAAE and RMAE. The resulting measures are compiled in table 9.1. The NRSME measures are also presented in figure 9.1, where it is shown that Anisotropic Kriging, Neural Networks and Radial Basis functions perform best.

The accuracies of these three surrogate models are also investigated when they are fitted with 100 and 150 samples to see the impact of larger samples. A comparison of this investigation is shown in figure 9.2. It is shown that additional samples improve the accuracies of the surrogate

Figure 9.1 Bar chart showing the NRSME of each surrogate model type fitted with 50 samples.

Table 9.1 Accuracy measure for the different surrogate models.

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

models, which was expected to a certain extent. The most important finding is that Anisotropic Kriging performs best and is therefore used to model the geometrical properties in the framework that optimizes the overall design of industrial robots.

Anisotropic Kriging is therefore also used in the other papers in the thesis when the time it takes to update the surrogate model is not of importance. PRS is instead chosen when it is important that the surrogate model fitting and evaluation is fast and/or only a high local accuracy is needed.

Figure 9.2 Bar chart showing the NRSME of Kriging, Neural Networks and Radial Basis Functions fitted with 50 and 100 samples.

10

Improving the Complex-RF Method using Surrogate Models

Paper [III] presents improvements of the Complex-RF optimization algorithm by modifying it so that it uses Polynomial Response Surfaces (PRSs, see 5.1.1) throughout the optimization.

The modified algorithm is called Compelx-RFM and a flowchart of the operations performed by it is shown in figure 10.1. It operates similarly to Complex-RF until enough function evaluations have been made to enable a PRS to be created. Every estimation of the objective function value is then carried out by calling the PRS. However, the PRS needs to be updated to ensure that it is accurate in the parts of the design space where the optimization algorithm is operating. A simulation of the true objective function is therefore made occasionally to update the PRS. This is handled by the "SM check" activity that is shown in figure 10.1. If the coefficients of the two latest PRSs are equal enough, it is believed that the PRS is trustworthy and the "SM check" activity omitted for a number of iterations. How equal the two PRSs need to be, ς and the number of omissions, ε , are two parameters that can be tuned to adjust the performance of Complex-RFM.

The performance of Complex-RFM is compared with Complex-RF, a Genetic Algorithm, a gradient-based method and an SAO for sev-

Figure 10.1 Flowchart for the Complex-RFM optimization algorithm

eral mathematical functions and the electric motorcycle model. The gradient-based method is *fmincon* that can be found in MATLAB [89]. The SAO use no other updating or infill criterion than the optimum of the surrogate model. The conditions of the comparison are describes in section 10.1.

The resulting performance indices are presented in table 10.1. Complex-RFM outperforms Complex-RF for every problem, which indicates that the modifications improved Complex-RF. There are two columns for Complex-RFM - one is for the algorithm when its parameters have been optimized for each problem and the other is for when it is used with its standard parameter values. These values are shown in

Table 10.1 Performance indices for the optimization algorithms and problems.

Problem	Complex-RF	Complex-RFM Optimized	Complex-RFM Standard	fmincon	SAO
Peaks	0.620	0.883	0.835	0.758	0.996
Hartmann6	1	1	0.947	1	0.999
Engine timing	0.274	0.644	0.509	0.190	0.250
PID demonstration	0.322	0.549	0.380	0.213	0.436
Electric MC	0.465	0.822	0.784	0.218	0.804

Table 10.2 Optimal parameters for Complex-RFM for each problem.

Problem	R_{fak}	γ	ε	ς
Peaks	1	0.05	25	0.01
Hartmann6	0.2	0.05	2	0.002
Engine timing	0.86	0	1	0.005
PID demonstration	0.79	0.04	3	0.001
Electric MC	0.28	0.06	2	0.008
Suggested standard	0.50	0.20	4	0.004

table 10.2.

The parameters of the optimization algorithm can be tuned by performing a meta-optimization [93], [94]. Such optimizations have been performed for each of the mathematical functions and the results are presented in table 10.2. The results have been used to give recommended values for the parameters to use as standard values.

The chosen degree of the polynomial response surface is of the second order. This is an adequate balance between precision and required number of function evaluations to fit the surrogate model. The precision needs to be good enough to reanimate the overall characteristics of the objective function, to guide the optimization, but do not need excellent accuracy over the whole design space.

The last surrogate model can also be used to estimate how the objective function behaves in the vicinity of the optimum. One example is shown in figure 10.2, where the peaks function has been optimized using Complex-RFM.

Figure 10.2 *Example of a PRS that reanimates the shape of the optimum of the peaks function.*

10.1 Performance metric for comparison of optimization methods

The ensuing chapters will present comparisons and benchmarking of different methods that can be used to perform optimizations of complex products. The conditions under which these occur are common to all of them. Each method is used 1 000 times to optimize each test problem and model in order to reduce the uncertainty due to statistical variation. The entities that are calculated are the average number of required function evaluations, the probability of finding the optimum and the mean value and standard deviation of the optimal objective function value.

Many different criteria can be used to determine whether the optimization was successful or not. These comparisons use the same criterion as Reischel and Leisutre [79], which means that an optimization is successful if the objective function value found is among the lowest per mille of all possible solutions. This number is found by performing 100,000 evenly spread simulations in the design space.

10.1.1 Performance Index

Several performance indices, η , for optimization have been proposed. They usually calculate the convergence rate of the optimization [94], but Krus and Ölvander [18], for example, use equation (10.1) instead. P_{opt} is the chance of finding the optimum by performing one optimization, also denoted hit-rate, and n_{opt} is the required number of function evaluations for the optimization algorithm to converge.

$$\eta = \frac{\text{Log}_2(1 - P_{opt})}{n_{opt}} \quad (10.1)$$

Another performance index, first presented in paper [VIII] is used in these comparisons. It is based on the assumption that if P_{opt} is the chance of finding the optimum by performing an optimization, then the probability of not finding the optimum is $1 - P_{opt}$. The probability of not finding the optimum when n optimizations are performed can be calculated with equation (10.2) and the probability of finding the optimum is then $1 -$ equation (10.2).

$$(1 - P_{opt})^n \quad (10.2)$$

This is a biased performance index since it does not consider the fact that different algorithms require different numbers of function evaluations to converge. n should therefore be replaced with the number of simulations that can be performed with a certain computational budget. The budget is here set to 100 function evaluations to obtain a numerical value for the performance index. The possible number of optimizations is then $100/n_{opt}$, where n_{opt} is the average number of function evaluations for the chosen optimization algorithm. The performance index can then be calculated with equation (10.3).

$$\eta = 1 - (1 - P_{opt})^{100/n_{opt}} \quad (10.3)$$

The performance index can be interpreted as the probability of finding the optimum if 100 simulations of the original model are performed.

11

Methods for Robust Design Optimization

Papers [II] and [IV] investigate different ways to incorporate surrogate models into RDO and more details can be found there. Complex-RF is used as the optimization algorithm and Latin Hypercube Sampling is used to estimate the statistics of each design. The first comparison investigates which entity the surrogate model should reanimate. A method without surrogate models is also included to provide reference values. The tested methods are therefore as follows:

1. A1 - Robust Design Optimization without any surrogate models.
2. A2 - Create a surrogate model of the original model before the optimization is begun and perform the RDO on the surrogate model.
3. A3 - Create a surrogate model of the original model during the RDO and perform all function evaluations on the surrogate model once it has been created.
4. A4 - Create a surrogate model of the mean value and standard deviation during the RDO and perform all estimations of these entities by calling the surrogate model once it has been created.
5. A5 - Create a surrogate model of the objective function during the RDO and call the surrogate model during the rest of the RDO once it has been created.

6. A6 - Create a surrogate model of the original model during the optimization. A deterministic evaluation of the original model is performed when the objective function value of a design is to be evaluated. The surrogate model is updated with this point and the LHS calls the surrogate model to estimate the mean value and standard deviation.

A6 is developed from the Complex algorithm in paper [II]. Paenke et al. [95] has proposed a method referred to as Evolutionary Algorithm for Robustness Optimization, here referred to as EARO, that is based on a similar idea. They both perform a deterministic evaluation and update the surrogate model before the surrogate model is used to estimate the mean value and standard deviation of the suggested design. The difference is that Paenke et al. use a genetic optimization algorithm and polynomial response surfaces as surrogate models.

The comparison was made with the conditions presented in section 10.1 and the results are presented in table 11.1. It is shown that it is better to create a surrogate model of the original function or model than to create a surrogate model of the mean value and standard deviation or the whole robust objective function. The table also shows how ineffective it is to perform an RDO without a surrogate model.

The methods denoted A1, A2 and A6 are tested again against two other methods in paper [IV]. One of the other methods is EARO [95]. The other method is Robust Sequential Optimization (RSO), presented for example by Wiebenga et al. [12] and Rehman et al. [96]

The performance indices for the methods for several problems are presented in table 11.2. These results confirm that A6 is better than A1 and A2. A6 is also better than EARO, which has a similar idea behind its mechanisms. This means that the replacement of the genetic algorithm as optimization algorithm with Complex-RF and replacing the polynomial response surfaces with Kriging improved the performance.

It is also shown that sequential robust optimization outperforms the other methods for all problems. This means that it is the recommended choice for Robust Design Optimization.

Table 11.1 *Performance indices for six different methods for Robust Design Optimization.*

Problem	Methods					
	A1	A2	A3	A4	A5	A6
Peaks	0.014	0.428	0.309	0.003	0.001	0.86
Aspenberg	0.410	0.227	0.212	0.053	0.062	0.99
DPR	0.070	0.762	0.719	0.452	0.404	0.942

Table 11.2 *Performance indices for five methods for Robust Design Optimization.*

Problem	Methods				
	A1	A2	A6	EARO	RSO
Branke3	0.026	0.076	0.117	0.002	0.338
Hart6	1	0.010	0.712	0.010	0.861
Peaks	0.022	0.606	0.655	0.352	0.999
Rehman	0.019	0.714	0.706	0.023	0.998
Beam	0.010	0.014	0.036	0.009	0.215
Electric MC	0.005	0.110	0.044	0.015	0.503

12

MDO of a Balancing Mechanism for Industrial Robots

Paper [V] performs several MDOs of the balancing mechanism presented in 8.3. A schematic of the proposed framework is shown in figure 12.1. The framework is monolithic in its present form even though the dynamic simulation and the static and weight estimations could be run in parallel. This is due to fact that the two latter can be evaluated in less than a second each and the wall-clock time of one optimization is then marginally longer when each analysis is done in a series instead of parallel.

The framework can design either a mechanical, gas or hydro-pneumatic balancing mechanism. It begins by calculating the position of the piston inside the balancing cylinder as a function of the angle of the second axis. These positions are then used to calculate the torques from the balancing mechanism according to equation (12.1), (12.4) or (12.6), depending on the chosen concept.

Equation (12.1) shows how the axial force in the piston can be calculated as a function of the spring stiffness, k_{spring} , and the compression of the springs.

$$F = k_{spring}\Delta x \quad (12.1)$$

Figure 12.1 Schematic of the proposed framework with a focus on how the information is passed between the optimization algorithm and the four tools. These are the four squares with grey background.

The spring stiffness can be calculated according to equation (12.2) as a function of the thread diameter, d , spring diameter, D , and the number of turns, n .

$$k_{spring} = \frac{Gd^4}{8nD^3} \quad (12.2)$$

The gas spring concept can calculate the axial force in the piston under the assumption that the gas pressure changes according to a polytropic expression as shown in equation (12.3).

$$C = p_{gas} V_{gas}^q \quad (12.3)$$

The axial force can then be calculated according to equation (12.4).

$$F = A_{piston} \frac{C}{(A_{piston} x)^q} \quad (12.4)$$

The hydro-pneumatic concept fills one chamber of the balancing cylinder with a fluid that is attached to an accumulator. The size of the

accumulator can be calculated according to equation (12.5).

$$V_0 = \frac{\Delta V \frac{p_1}{p_0}}{1 - \left(\frac{p_1}{p_2}\right)^{\frac{1}{n}}} \quad (12.5)$$

The axial force in the piston can then be calculated according to equation (12.6) under the assumption that the liquid is incompressible compared to the gas.

$$p_1 = \frac{p_0 V_0^q}{(V_0 - A_{piston} x)^q} \quad (12.6)$$

The torque function is sent to an input file for the dynamic simulation, which simulates all properties of the industrial robot during a given robot cycle. The input file also contains data on all links, actuators and other components. The dynamic model includes the robot mechanism, a path planner and the robot's controller. It is both an advantage and disadvantage that the dynamic simulation stops if the suggested balancing is inadequate and this needs to be handled by the optimization. The output of interest from the dynamic simulation is the time it takes for the robot to complete the cycle, CT , the energy consumption, EC , the lifetimes of the gearboxes, LT , and the dynamic torque, T .

The static estimator uses the forces and pressure inside the balancing cylinder to estimate the required cylinder wall thickness. This is then used by the weight estimator to calculate the weight of the balancing mechanism, CW .

The mathematical formulation of the optimization problem is shown in equation (12.7). It is shown that there are numerous objectives that should be optimized for. The optimization variables are the mounting positions of the balancing mechanism and a few variables that determine the relationship between piston position and force and are therefore unique to each concept. More details can be found in paper [V].

$$\begin{aligned} \min f(x) &= \{CT, EC, T_{unbal}, CW, CL, V_0\} \\ s.t. & \\ g_1(x) &= LT_{min} - LT \leq 0 \\ x_j^L &\leq x_j \leq x_j^U \end{aligned} \quad (12.7)$$

The problem is optimized with NSGA-II in modeFRONTIER [92], which results in a Pareto front. The front is shown for two variables at a

time in figures 12.2 and 12.3. The first does not give much information except that the volume of the accumulator increases as the unbalanced torque decreases. Figure 12.3, on the other hand, reveals that solutions with lower cycle times than 0.972 result in significantly longer balancing mechanisms.

Figure 12.2 The Pareto front plotted as the volume of the accumulator versus the unbalanced static torque.

It is also of interest to investigate which parameter values contribute to the Pareto front and this can be presented with a parallel coordinate chart such as that shown in figure 12.4. Each line in the chart corresponds to one Pareto optimal solution and it is shown that the solutions for α_1 and α_2 are clustered around 0.4-0.5. This information can be used to narrow the design space or to set these optimization variables to fixed values, which turns them into constants.

One solution from each concept has been selected and their properties are compiled in table 12.1. It is shown that the framework presents solutions that are better than the reference solution that exists in real life and with which they are compared. It should be noted that the Hydro-Pneumatic solution also needs to add an accumulator, which is roughly spherical and has a diameter of ten cm.

The hydro-pneumatic concept is promising according according to the

Figure 12.3 The Pareto front plotted as the robot cycle time versus the cylinder length.

Table 12.1 Percentage values of the balancing properties for three optimized concepts with respect to a reference balancing cylinder.

Property	Ref	Mech	Gas	HP
Cycle Time	100	96	98	98
Cylinder Length	100	110	115	79
Cylinder Weight	100	123	14	18
Energy Consumption	100	91	96	90
Lifetime Second Axis	100	187	135	190
Maximum Axial Force	100	139	129	119
Unbalanced Static Torque	100	92	91	88

results, even when the accumulator is added. It is important, however, to remember that the concepts that involve gas come with risks such as leakages, which means that a mechanical concept might be preferable anyway in some situations.

The dynamic simulation can be seen as confirmation whether a suggested design works or not since it returns an error if an infeasible design is suggested. The results in table 12.1 also seem credible since they are

Figure 12.4 A parallel coordinate chart showing the variable values of all Pareto optimal designs.

close to the reference design. This means that this section proves that it is possible to use this monolithic MDO framework to optimize the balancing mechanism.

Part III

Conclusions

13

Discussion

Most complex products, such as mechatronic systems, demand quite advanced models to estimate the performance, endurance and cost of each concept. It is therefore important to use efficient methods to optimize the concepts.

The methods presented in the Frame of Reference are some of the most popular for their purposes. More methods exist, but the thesis has been narrowed down to dealing with only the most referenced methods. The limitations must be drawn somewhere to avoid unrealistic long implementation and benchmarking times. The methods described are enough to give an idea of which methods exist and to benchmark new and modified algorithms.

The knowledge about existing methods for performing design optimization and their advantages and drawbacks provide good directions as to where additional research could be of interest. This includes increasing understanding of the methods and modifying them in the hope of improving them.

The most noteworthy assumption in the thesis is that the models that are optimized have computational burdens of the order of minutes. There is no need to involve surrogate models in the design process if the models are fast. The engineer can instead build the desired objective function and optimize the models with an optimization algorithm with high accuracy such as a genetic algorithm or a particle swarm optimization. If the models are too computationally expensive other techniques might be better suited.

An additional assumption is that the models studied are considered to be black box models where the derivatives of the objectives and con-

straints could not be obtained analytically. Furthermore the problems addressed in real life are typically not convex, hence the thesis has focused on non-gradient based optimization techniques.

Another assumption is that the models that are used in the thesis are accurate reanimations of the real world. This means that the results from them are considered credible and consistent. Models are usually not very accurate for more than a small range of parameter values and this might affect the results, but methods for model validation and verification is outside the scope of the thesis.

The methods are intended to be used in the industry and it is therefore imperative that they are evaluated for applications that are as close to real products as possible. It is impossible to prove without doubt that the methods work for every conceivable product. The comparisons therefore strive to use several applications, with different model fidelity, to demonstrate the performances of the methods under different conditions. The models in the thesis are taken from the vehicle, aeronautical and robotics domains and should therefore be representative for complex products.

The results from the optimizations should ideally be confirmed with physical experiments, but those are expensive, especially when a whole complex product such as an industrial robot needs to be constructed. The methods and suggested optimal designs are therefore compared with models and simulations with parameter settings that represent existing products. This is for example the case for the balancing mechanism in paper [V] where the obtained solutions are compared with an existing design.

The optimizations are carried out with a standard computer with 16 GB RAM-memory and a 3.1 GHz processor without consuming more wall-clock time than a day or two. This, together with credible results from the optimizations, suggest that the methods can be applied in industry.

It should however be pointed out that it is only possible to analyse the things that have been modelled for a specific design problem. The unknown characteristics that have been neglected or omitted in the computer analysis might affect the results in real life. This is another rationale for performing physical experiments to confirm the results from the computer analyses.

The comparison of the surrogate model types in paper [I] only tested the surrogate models for one problem. But it revealed that Kriging,

Neural Networks and Radial Basis Functions are the most accurate surrogate model types and this agrees with the findings and suggestions by other authors. One of them should generally be used and which one to use depends on personal preferences or availability. A comparison of the surrogate models should however ideally be made for each model or phenomena that is animated with a surrogate model to find the most suitable model.

Furthermore, Complex-RFM outperforms Complex-RF for every problem in the comparison. It should be possible to find mathematical problems where Complex-RF outperforms Complex-RFM, but the latter is generally better. It is therefore recommended to use Complex-RFM instead of Complex-RF for objective functions with few variables that consumes wall-clock times in the size of minutes.

It is no surprise that the results in the thesis confirm that surrogate models are efficient when computationally demanding models are optimized. But the results clearly show the potential of the, to most of the industry, relatively unknown Sequential Approximate Optimization.

14

Conclusions

The thesis presents methods that can be used to optimize complex products. Several of the methods in each step of the optimization process have been compared with each other to evaluate their performances. This increases the understanding of the methods and can aid the selection of which methods to use. It also answers the research question regarding identifying popular methods for design optimization of complex products.

Some of the methods have been modified to improve their performance and the benchmarking confirms that the modified methods perform better than the original methods. This answers the research question regarding whether it is possible to modify a method to improve its performance.

The comparison of surrogate model types in paper [I] reveals that Anisotropic Kriging, Neural Networks and Radial Basis Functions are most accurate. The geometrical model that they reanimate has both continuous and discrete variables and should therefore be an appropriate test case. Anisotropic Kriging has the highest accuracy and is therefore used as surrogate model throughout the thesis, except when the fitting time is of importance or only local estimations are needed. Complex-RFM fulfils both these criteria and therefore uses Polynomial Response Surfaces instead.

A modified version of Complex-RF, Complex-RFM, is presented in the thesis. The modified algorithm outperforms its original for every problem in the comparison that does not have a one hundred percent hit-rate. This means that the new algorithm is better than Complex-RF and is preferred when problems with few variables are optimized.

Furthermore, the results indicate that a sequential approximate optimization is the best way to handle optimization of computationally demanding models. This is also one of the most complicated methods of the ones presented here since numerous optimizations are performed and the surrogate models are updated iteratively. The effort seems to be worth it, however, since it clearly performs better than the other methods for all benchmarking problems.

The results also indicate that it is almost impossible to perform optimization, and especially probabilistic optimizations such as Robust Design Optimization, of computationally demanding models without involving surrogate models. Several thousand simulations of a demanding model are clearly unrealistic.

One of the greatest challenges over the coming years is to increase usage of the methods for surrogate model assisted optimization in industry. They are the ones that stand to benefit from better support for modelling, simulation and optimization of complex methods.

The problem is that it requires more knowledge to include surrogate models in the optimization compared to using the original models. It is already challenging for many companies to set up MDO frameworks that can be used to optimize their products. Methods must therefore be easy to use and maintain if they should be used in industry.

An increasing number of tools for MDO are developed and some of them are rather good. Further improvements are needed however. Sequential Approximate Optimization is for example not implemented in any of the most popular tools as far as the author know.

Most of the methods that are presented in the frame of reference of the thesis originate from the 20th century. Recent research have instead focused on how the methods should be used and combined effectively. This can be seen by the numerous papers regarding MDO frameworks and surrogate model assisted optimization algorithms that have been published recently.

The most desired breakthrough according to this author in the field of efficient optimization of complex products would however be a new surrogate model type that is accurate, fast to create and has easily understood mathematics behind it.

Other areas for future work includes application of the proposed methods to more engineering problems and potentially also in real industrial settings. Furthermore, even though the field of surrogate assisted optimization has been growing recently there are still improvements to

be made in order to identify the optimal way in which to spend the expensive and time consuming calls of the high fidelity models.

Bibliography

- [1] Johan Andersson. “Multiobjective optimization in engineering design applications to fluid power systems”. PhD thesis. Linköping University, 2001.
- [2] Patrick J Roache. *Verification and validation in computational science and engineering*. Hermosa, 1998.
- [3] William L Oberkampf and Timothy G Trucano. “Verification and validation benchmarks”. In: *Nuclear engineering and Design* 238.3 (2008), pp. 716–743.
- [4] Osman Balci. “Verification validation and accreditation of simulation models”. In: *Proceedings of the 29th conference on Winter simulation*. IEEE Computer Society. 1997, pp. 135–141.
- [5] Robert G Sargent. “Verification and validation of simulation models”. In: *Proceedings of the 37th conference on Winter simulation*. Winter Simulation Conference. 2005, pp. 130–143.
- [6] Johan Andersson. *A survey of multiobjective optimization in engineering design, Tech. Rep. LiTH-IKP-R-1097*. Tech. rep. Department of Mechanical Engineering, 2000.
- [7] Joaquim RRA Martins and Andrew B Lambe. “Multidisciplinary design optimization: a survey of architectures”. In: *AIAA journal* 51.9 (2013), pp. 2049–2075.
- [8] Alexander Forrester, András Sóbester, and Andy Keane. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.

- [9] Douglas C Myers Raymond H; Montgomery and Christine M Anderson-Cook. *Response surface methodology: process and product optimization using designed experiments*. Vol. 705. John Wiley & Sons, 2009.
- [10] Warren S McCulloch and Walter Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.
- [11] András Sóbester, Alexander IJ Forrester, David JJ Toal, Es Tresidder, and Simon Tucker. “Engineering design applications of surrogate-assisted optimization techniques”. In: *Optimization and Engineering* 15.1 (2014), pp. 243–265.
- [12] JH Wiebenga, AH Van Den Boogaard, and G Klaseboer. “Sequential robust optimization of a V-bending process using numerical simulations”. In: *Structural and multidisciplinary optimization* 46.1 (2012), pp. 137–153.
- [13] Russ C Eberhart and James Kennedy. “A new optimizer using particle swarm theory”. In: *Proceedings of the sixth international symposium on micro machine and human science*. Vol. 1. New York, NY. 1995, pp. 39–43.
- [14] John H Holland. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press, 1975.
- [15] David E Goldberg and John H Holland. “Genetic algorithms and machine learning”. In: *Machine learning* 3.2 (1988), pp. 95–99.
- [16] MJ Box. “A new method of constrained optimization and a comparison with other methods”. In: *The Computer Journal* 8.1 (1965), pp. 42–52.
- [17] John A Nelder and Roger Mead. “A simplex method for function minimization”. In: *The computer journal* 7.4 (1965), pp. 308–313.
- [18] Petter Krus and Johan Ölvander. “Optimizing optimization for design optimization”. In: *Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2003*. ASME Press. 2003.

-
- [19] M Petterson, Johan Andersson, and Petter Krus. “Methods for discrete design optimization”. In: *Proceedings of ASME DETC05, Design Automation Conference, 24-28 Sep., Long Beach, California*. 2005.
 - [20] Liselott Ericson, Johan Ölvander, and Jan-Ove Palmberg. “On optimal design of hydrostatic machines”. In: *Proceedings of the 6th International Fluid Power Conference, IFK, Vol WS*. 2008.
 - [21] RL Keeney and H Raiffa. “Decisions with multiple objectives. 1976”. In: *John Wiley & Sons, New York* (1976).
 - [22] Abraham Charnes, William W Cooper, and Robert O Ferguson. “Optimal estimation of executive compensation by linear programming”. In: *Management science* 1.2 (1955), pp. 138–151.
 - [23] Kaisa Miettinen. *Nonlinear multiobjective optimization*. Vol. 12. Springer Science & Business Media, 1999.
 - [24] Chankong Vira and Yacov Y Haimes. *Multiobjective decision making: theory and methodology*. 8. North-Holland, 1983.
 - [25] Carlos M Fonseca and Peter J Fleming. “Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation”. In: *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 28.1 (1998), pp. 26–37.
 - [26] Kalyanmoy Deb, Sameer Agarwal, Amrit Pratap, and Tanaka Meyarivan. “A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II”. In: *Parallel Problem Solving from Nature PPSN VI* (2000), pp. 849–858.
 - [27] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. *SPEA2: Improving the strength Pareto evolutionary algorithm*. 2001.
 - [28] K Miettinen and MM Mäkelä. “Interactive bundle-based method for nondifferentiable multiobjective optimization: nimbus §”. In: *Optimization* 34.3 (1995), pp. 231–246.
 - [29] Kaisa Miettinen and Marko M Mäkelä. “Interactive multiobjective optimization system WWW-NIMBUS on the Internet”. In: *Computers & Operations Research* 27.7 (2000), pp. 709–723.

- [30] R Benayoun, J De Montgolfier, Jo Tergny, and O Laritchev. “Linear programming with multiple objective functions: Step method (STEM)”. In: *Mathematical programming* 1.1 (1971), pp. 366–375.
- [31] Alexander V Lotov and Kaisa Miettinen. “Visualizing the Pareto frontier”. In: *Multiobjective optimization*. Springer, 2008, pp. 213–243.
- [32] Lucien A Schmit. “Structural design by systematic synthesis”. In: *Proc. of the Second ASCE Conference on Electronic Computation*. 1960, pp. 105–122.
- [33] Olivier de Weck, Jeremy Agte, J Sobieski, Paul Arendsen, Alan Morris, and Martin Spieck. “State-of-the-art and future trends in multidisciplinary design optimization”. In: *Proceedings of the 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA*. Vol. 1905. 2007.
- [34] Ilan Kroo. “MDO for large-scale design”. In: *Multidisciplinary design optimization: state-of-the-art*. SIAM, Philadelphia (1997), pp. 22–44.
- [35] Sang-Il Yi, Jung-Kyu Shin, and GJ Park. “Comparison of MDO methods with mathematical examples”. In: *Structural and Multidisciplinary Optimization* 35.5 (2008), pp. 391–402.
- [36] Ian P Sobieski and Ilan M Kroo. “Collaborative optimization using response surface estimation”. In: *AIAA journal* 38.10 (2000), pp. 1931–1938. DOI: 10.2514/2.847.
- [37] Parviz M Zadeh, Vassili V Toropov, and Alastair S Wood. “Metamodel-based collaborative optimization framework”. In: *Structural and Multidisciplinary Optimization* 38.2 (2009), pp. 103–115. DOI: 10.1007/s00158-008-0286-8..
- [38] Raphael T Haftka. *Automated procedure for design of wing structures to satisfy strength and flutter requirements*. Tech. rep. Tech. Rep. TN D-7264, NASA Langley Research Center, Hampton, VA, 1973.
- [39] Evin J Cramer, JE Dennis Jr, Paul D Frank, Robert Michael Lewis, and Gregory R Shubin. “Problem formulation for multidisciplinary optimization”. In: *SIAM Journal on Optimization* 4.4 (1994), pp. 754–776.

-
- [40] Raphael T Haftka. “Simultaneous analysis and design”. In: *AIAA journal* 23.7 (1985), pp. 1099–1103. DOI: 10.2514/3.9043.
- [41] CL Bloebaum, P Hajela, and J Sobieszczanski-Sobieski. “Non-hierarchical system decomposition in structural optimization”. In: *Engineering Optimization+ A35* 19.3 (1992), pp. 171–186.
- [42] Robert David Braun. “Collaborative optimization: an architecture for large-scale distributed design”. In: (1996).
- [43] Jaroslaw Sobieszczanski-Sobieski, J Agte, and R Sandusky. “Bi-Level Integrated System Synthesis (BLISS)”. In: *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO*. 1998.
- [44] Simon Tosserams, LFP Etman, and JE Rooda. “Augmented Lagrangian coordination for distributed optimal design in MDO”. In: *International journal for numerical methods in engineering* 73.13 (2008), pp. 1885–1910.
- [45] Victor DeMiguel and Walter Murray. “A local convergence analysis of bilevel decomposition algorithms”. In: *Optimization and Engineering* 7.2 (2006), pp. 99–133.
- [46] Moon-Kyun Shin and Gyung-Jin Park. “Multidisciplinary design optimization based on independent subspaces”. In: *International Journal for Numerical Methods in Engineering* 64.5 (2005), pp. 599–617.
- [47] Raphael T Haftka and Layne T Watson. “Multidisciplinary design optimization with quasiseparable subsystems”. In: *Optimization and Engineering* 6.1 (2005), pp. 9–20.
- [48] Ian R Chittick and Joaquim RRA Martins. “An asymmetric suboptimization approach to aerostructural optimization”. In: *Optimization and Engineering* 10.1 (2009), pp. 133–152.
- [49] Natalia M Alexandrov and Robert Michael Lewis. “Analytical and computational aspects of collaborative optimization for multidisciplinary design”. In: *AIAA journal* 40.2 (2002), pp. 301–309. DOI: 10.2514/2.1646.
- [50] Richard J Balling and Jaroslaw Sobieszczanski-Sobieski. “Optimization of coupled systems-a critical overview of approaches”. In: *AIAA journal* 34.1 (1996), pp. 6–17. DOI: 10.2514/3.13015.

- [51] G Gary Wang and S Shan. “Review of metamodeling techniques in support of engineering design optimization”. In: *Journal of Mechanical Design* 129 (2007), p. 370.
- [52] Timothy W Simpson, JD Poplinski, Patrick N Koch, and Janet K Allen. “Metamodels for computer-based engineering design: survey and recommendations”. In: *Engineering with computers* 17.2 (2001), pp. 129–150.
- [53] DG Krige. “A statistical approach to some basic mine valuation problems on the Witwatersrand”. In: *Journal of Chemical, Metallurgical, and Mining Society of South Africa* (1951).
- [54] Ruichen Jin, Wei Chen, and Timothy W Simpson. “Comparative studies of metamodeling techniques under multiple modelling criteria”. In: *Structural and Multidisciplinary Optimization* 23.1 (2001), pp. 1–13.
- [55] Hans-Georg Beyer and Bernhard Sendhoff. “Robust optimization—a comprehensive survey”. In: *Computer methods in applied mechanics and engineering* 196.33 (2007), pp. 3190–3218.
- [56] Marcus Redhe, Jimmy Forsberg, T Jansson, Per-Olof Marklund, and Larsgunnar Nilsson. “Using the response surface methodology and the D-optimality criterion in crashworthiness related problems”. In: *Structural and multidisciplinary optimization* 24.3 (2002), pp. 185–194.
- [57] Ruichen Jin, Xiaoping Du, and Wei Chen. “The use of metamodeling techniques for optimization under uncertainty”. In: *Structural and Multidisciplinary Optimization* 25.2 (2003), pp. 99–116.
- [58] Jerome Sacks, William J Welch, Toby J Mitchell, and Henry P Wynn. “Design and analysis of computer experiments”. In: *Statistical science* (1989), pp. 409–423.
- [59] Edward H Isaaks and R Mohan Srivastava. *Applied geostatistics*. Oxford University Press, 1989.
- [60] Xin Yu Shao, Xue Zheng Chu, Liang Gao, and Hao Bo Qiu. “A novel methodology to manage uncertainty for multidisciplinary design and optimization in conceptual design”. In: *Advanced Materials Research* 44 (2008), pp. 225–232.

- [61] Jay D Martin and Timothy W Simpson. “A Monte Carlo simulation of the Kriging model”. In: *10th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. 2004, pp. 2004–4483.
- [62] JB Boisvert, JG Manchuk, and CV Deutsch. “Kriging in the presence of locally varying anisotropy using non-Euclidean distances”. In: *Mathematical Geosciences* 41.5 (2009), pp. 585–601.
- [63] David John James Toal and Andy J Keane. “Non-stationary kriging for design optimization”. In: *Engineering Optimization* 44.6 (2012), pp. 741–765.
- [64] David Aspenberg. “Robust optimisation of structures: Evaluation and incorporation of variations in simulation based design”. PhD thesis. Linköping University, 2011.
- [65] David Aspenberg, Johan Jergeus, and Larsgunnar Nilsson. “Robust optimization of front members in a full frontal car impact”. In: *Engineering Optimization* 45.3 (2013), pp. 245–264.
- [66] David S Broomhead and David Lowe. “Multi-variable functional interpolation and adaptive networks”. In: *Complex Systems* 2 (1988), pp. 321–355.
- [67] R Duvigneau and C Praveen. “Meta-modeling for robust design and multi-level optimization”. In: *42e Colloque d’Aérodynamique Appliquée, Couplages et Optimisation Multidisciplinaires, INRIA Sophia Antipolis, 19-21 March*. 2007.
- [68] Hong-Seok Park and Xuan-Phuong Dang. “Structural optimization based on CAD–CAE integration and metamodeling techniques”. In: *Computer-Aided Design* 42.10 (2010), pp. 889–902.
- [69] Hirotaka Nakayama, Masao Arakawa, and Rie Sasaki. “A computational intelligence approach to optimization with unknown objective functions”. In: *Artificial Neural Networks-ICANN 2001*. Springer, 2001, pp. 73–80.
- [70] Michael D McKay, Richard J Beckman, and William J Conover. “Comparison of three methods for selecting values of input variables in the analysis of output from a computer code”. In: *Technometrics* 21.2 (1979), pp. 239–245.

- [71] Cedric J Sallaberry, Jon C Helton, and Stephen C Hora. “Extension of Latin hypercube samples with correlated variables”. In: *Reliability Engineering & System Safety* 93.7 (2008), pp. 1047–1059.
- [72] Jay D Martin and Timothy W Simpson. “Use of adaptive meta-modeling for design optimization”. In: *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. 2002, pp. 4–6.
- [73] Martin Meckesheimer, Andrew J Booker, Russell R Barton, and Timothy W Simpson. “Computationally inexpensive metamodel assessment strategies”. In: *AIAA journal* 40.10 (2002), pp. 2053–2060.
- [74] Songqing Shan and G Gary Wang. “Metamodeling for high dimensional simulation-based design problems”. In: *Journal of Mechanical Design* 132 (2010), p. 051009.
- [75] Amir Safari. “Numerical Simulation Driven Product Design Optimization for Improved Performance”. PhD thesis. University of Stavanger, 2014.
- [76] Kambiz Haji Hajikolaie and G Gary Wang. “High Dimensional Model Representation With Principal Component Analysis”. In: *Journal of Mechanical Design* 136.1 (2014), p. 011003.
- [77] Adel Younis, Kerem Karakoc, Zuomin Dong, Edward Park, and Afzal Suleman. “Application of SEUMRE global optimization algorithm in automotive magnetorheological brake design”. In: *Structural and Multidisciplinary Optimization* 44.6 (2011), pp. 761–772.
- [78] Donald R Jones. “A taxonomy of global optimization methods based on response surfaces”. In: *Journal of global optimization* 21.4 (2001), pp. 345–383.
- [79] Patrick H Reisenthel and Daniel J Lesieutre. “A Numerical Experiment on Allocating Resources Between Design of Experiment Samples and Surrogate-Based Optimization Infills”. In: *13th AIAA Non-Deterministic Approaches Conference*. 2011.
- [80] Harold J Kushner. “A new method of locating the maximum point of an arbitrary multippeak curve in the presence of noise”. In: *Journal of Fluids Engineering* 86.1 (1964), pp. 97–106.

-
- [81] Magnus Carlsson. “Methods for Early Model Validation - Applied on Simulation Models of Aircraft Vehicle”. Licenciate Thesis, Linköping University. 2013.
- [82] Christopher J Roy and William L Oberkampf. “A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing”. In: *Computer Methods in Applied Mechanics and Engineering* 200.25 (2011), pp. 2131–2144.
- [83] William L Oberkampf and Christopher J Roy. *Verification and validation in scientific computing*. Cambridge University Press, 2010.
- [84] Jay D Martin and Timothy W Simpson. “A methodology to manage system-level uncertainty during conceptual design”. In: *Journal of Mechanical Design* 128.4 (2006), pp. 959–968.
- [85] Yan-Gang Zhao and Tetsuro Ono. “A general procedure for first/second-order reliability method (form/sorm)”. In: *Structural safety* 21.2 (1999), pp. 95–112.
- [86] Timothy W Simpson, Timothy M Mauery, John J Korte, and Farrokh Mistree. “Kriging models for global approximation in simulation-based multidisciplinary design optimization”. In: *AIAA journal* 39.12 (2001), pp. 2233–2241.
- [87] Norbert Wiener. “The homogeneous chaos”. In: *American Journal of Mathematics* (1938), pp. 897–936.
- [88] Dongbin Xiu. *Numerical methods for stochastic computations: a spectral method approach*. Princeton University Press, 2010.
- [89] Mathworks. <http://www.mathworks.se/products/matlab/>. Accessed 09.05.14 2014.
- [90] Dymola. <http://www.3ds.com/products-services/catia/capabilities/modelica-systems-simulation-info/dymola>. Accessed 04.03.15 2015.
- [91] Andrei Neculai. “An unconstrained optimization test functions collection”. In: *Adv. Model. Optim* 10.1 (2008), pp. 147–161.
- [92] ESTECO. <http://www.esteco.com/modelfrontier>. Accessed 04.03.15 2015.
- [93] Robert E Mercer and JR Sampson. “Adaptive search using a reproductive meta-plan”. In: *Kybernetes* 7.3 (1978), pp. 215–228. DOI: 10.1108/eb005486.

- [94] P Krus and J Ölvander. “Performance index and meta-optimization of a direct search optimization method”. In: *Engineering Optimization* 1 (2012), pp. 1–19. DOI: 10 . 1080 / 0305215X.2012.725052.
- [95] Ingo Paenke, Jürgen Branke, and Yaochu Jin. “Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation”. In: *Evolutionary Computation, IEEE Transactions on* 10.4 (2006), pp. 405–420.
- [96] Samee Rehman, Matthijs Langelaar, and Fred van Keulen. “Efficient Kriging-based robust optimization of unconstrained problems”. In: *Journal of Computational Science* (2014). DOI: 10 . 1016/j . jocs.2014.04.005.

Papers

The articles associated with this thesis have been removed for copyright reasons. For more details about these see:

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-115939>