A Metamodelling Approach Towards
Virtual Production Intelligence
Truth is ever to be found in simplicity, and not in the multiplicity and confusion of things. As the world, which to the naked eye exhibits the greatest variety of objects, appears very simple in its internal constitution when surveyed by a philosophical understanding, and so much the simpler by how much the better it is understood.

Isaac Newton
Acknowledgements

The completion of this thesis and subsequent Ph.D would not have been possible without the support of several people. It was achieved during my time as a research and teaching assistant at the Nonlinear Dynamics of Laser Manufacturing Processes department NLD of the RWTH Aachen University, together with the Modeling and Simulation department of Fraunhofer Institute of Laser Technology ILT.

First, I would like to express my sincere gratitude to my supervisor, Professor Wolfgang Schulz, for giving me the opportunity to do my doctoral study under his guidance and for allowing me to be part of the institute. Prof. Schulz is not only the leader of the institute but also my mentor and ideal. His constant assistance, precious advices, endless motivation and efforts helped me improve my critical thinking, research methodology, and understanding of different physical phenomena. For this and much more, I am forever thankful.

Second, I am sincerely grateful to the remaining members of the institute for their patience, flexibility, genuine caring and concern which enabled me to finish my dissertation. I would like to thank especially my advisor Urs Eppelt. He has always encouraged me to find the answers and greatly contributed technically and methodologically in this work. When I was lost, Urs was the person re-adjusting the tracks to push my work forward. Thank you Urs! I am also thankful to the members of the project sector B1 'Virtual Production Intelligence' of the Cluster of Excellence 'Integrative Production Technology for High-Wage Countries' for their contributions and fruitful technical discussions.

Additionally, I would like to express my gratitude to my loving and supporting wife Malak for her patience and support during my long working hours especially in the past year in which we became parents to our beautiful baby boy Ryan-Hamza whose sweet smiles and laughs motivate me every single day. A heartfelt thank you also to Rabih Tbaily, Ahmad Kheir, Mohammad Shehadeh, and Mounir El Dana for making Aachen feel like home, and for being there in both the difficult times and the more easy-going ones.

In the end, I owe whatever I have achieved so far to my family that played above all a vital role in finishing this dissertation. I would like to thank my loving and caring mother Najla Tabbara, my sisters Sara and Farah whom, regardless of the distance, were always by my side inspiring me till the end, and my father Shafik for teaching me tolerance, endurance, and self control.

Aachen, January 13, 2016

Toufik Al Khawli
Abstract

Decision making for competitive production in high-wage countries is getting strongly influenced by computer simulations in order to cope with the fast changing global market demands. These simulations offer the possibility to find better designs within a short time, predict and optimize properties that cannot be measured directly from experiments, and explore new information and parameter ranges in known processes that are not easily accessible experimentally. Virtual prototyping is being successfully introduced in production where new products are virtually designed on the computer before building expensive machine components. This led to an explosion of tools that are being unified into a new notion in production, the virtual production system. This virtual system is based on the unification of real (experiments) and virtual (simulations) worlds, bringing together the scientific knowledge and technical know-how about the process. The outcome is presented by operative tools, namely process maps and visual design apps targeted to get more skilled developers and operators.

The primary achievement of this thesis is associated with the generation of process maps and visual design apps that are capable to explore the design space spanned by solutions of complex simulation models with the aid of a fast and frugal model, known as a metamodel. Metamodeling follows Benjamin Franklin’s rule for decision making formulated in 1772 [46], where he defined "Prudential Algebra", to be one of the major ingredients that can be applied in order to obtain one algebraic Pareto-value labelling alternative decisions. Metamodeling techniques define a procedure to analyze and represent complex physical systems using easy to use, fast mathematical designs to create cheap numeric surrogates that describe cause-effect relationships between setting parameters as inputs and criteria like product quality variables as outputs.

This dissertation focuses on two main research questions and three achievements.

The first research question focuses on how to efficiently generate and validate the approximation quality of a metamodel representing a complex simulation model. The question is investigated by different sampling, interpolation, and validation techniques. The outcome is a novel iterative method that generates a high quality metamodel for outputs with continuous as well as piecewise responses- the first achievement. It basically combines the Sequential Approximate Optimization (SAO) procedure and the Radial Basis Function network (RBFN).

The second research question focuses on how to improve a complex model by replacing the simulation model by a metamodel. This question is investigated by interfacing the RBFN metamodel with several global sensitivity analysis and visualization techniques. The outcome is a "Design-Cockpit" used for Virtual Production Intelligence-
the second achievement. Its main advantage is exploring information from complex multi-dimensional computer simulation models for optimizing the process and improving the know-how as well as the model structure with respect to the relevant properties.

Finally, the concept of process maps and visual designs are applied to five different laser manufacturing applications with different parameter domain spaces ranging from two to seven dimensions of parameter space. The results show a great efficiency of process map and a visual design in supporting decision making- the third achievement. To the author’s best knowledge, no fast interactive process map is conducted in any publication of laser manufacturing process before. Although the process maps and underlying process apps are applied to laser applications, the metamodeling techniques can be applied to almost any economical, ecological, or technical process, where the process itself is described by a black box model giving scalar data.
Zusammenfassung


Radial Basis Funktion Netzwerk (RBFN) ist wesentlich, um eine große Approximationsgüte zu erreichen. **Die zweite Forschungsfrage** konzentriert sich auf die Verbesserung des grundlegenden Verständnisses eines komplexen Modells durch das Ersetzen eines Simulationsmodells durch ein Metamodell. Durch die Verknüpfung des RBFN Metamodells, Globalen Sensitivitätsanalysen sowie Visualisierungstechniken wird diese Frage detailliert untersucht. Das Ergebnis ist ein 'Design-Cockpit', das auch als die 'Virtual Production Intelligence' bezeichnet wird. Der Hauptvorteil liegt in der Systematik, Lösungen für komplexe multi-dimensionale Simulationsmodelle für die Verfeinerung von physikalischen Modellen, als auch der Optimierung des Prozesses und der Produkte zu analysieren.

# Table of Contents

1 Introduction and Overview

1.1 Scope of Virtual Production Intelligence ................................................. 1
   1.1.1 Knowledge Extraction in Complex Systems .................................. 3
   1.1.2 Experiments and Diagnostics ...................................................... 4
   1.1.3 Modeling and Simulation .............................................................. 4
   1.1.4 Reduced Modeling ........................................................................ 5
   1.1.5 Metamodeling ............................................................................. 5

1.2 State of the Art ..................................................................................... 6
1.3 Research Questions ................................................................................ 7
1.4 Goals and contributions ......................................................................... 8
   1.4.1 Discrete to continuous support ..................................................... 8
   1.4.2 Metamodel-Based Design Optimization ...................................... 9
   1.4.3 Towards knowledge exploration in simulation models .............. 9

1.5 Thesis organization ............................................................................... 10
1.6 Summary .............................................................................................. 10

2 Design of Experiments

2.1 Classical DOE ...................................................................................... 13
   2.1.1 Factorial Designs ......................................................................... 14
   2.1.2 Two-levels Fractional Factorial Designs ..................................... 15
   2.1.3 Central Composite Designs ......................................................... 16
   2.1.4 Box-Behnken Designs ................................................................. 17
## TABLE OF CONTENTS

2.2 Simulation DOE ........................................ 18
   2.2.1 Latin Hypercube Sampling ............................ 20
   2.2.2 Monte Carlo Sampling .................................. 21
   2.2.3 Hammersley Sequence Sampling ....................... 21

2.3 Adaptive DOE ........................................... 22

2.4 Summary ................................................. 23

3 Metamodelling: From Theory to Practice .......................... 25
   3.1 Metamodelling for Continuous Responses ................. 26
      3.1.1 One Shot Approach .................................. 26
      3.1.2 Iterative Approach ................................ 31

3.2 Metamodelling for Discontinuous Responses ....................... 35
   3.2.1 One Shot Approach .................................. 35
   3.2.2 Smart Iterative Approach .............................. 37

3.3 Metamodelling Validation ................................. 41

3.4 Numerical Examples ...................................... 42
   3.4.1 Two-dimensional analytical function with a continous response .................................. 42
   3.4.2 Two-dimensional analytical function with a piecewise response ................................. 44

3.5 Summary ................................................. 47

4 Analysis and Exploration ................................... 49
   4.1 Global Sensitivity Analysis .............................. 49
      4.1.1 Measures based on the Analysis of Linear Models ........................................ 50
      4.1.2 Elementary Effect ..................................... 51
      4.1.3 Variance Decomposition Method - Sobol Method ........................................ 53

   4.2 Visualization and Exploration ............................ 57
      4.2.1 memoSlice .............................................. 57
      4.2.2 High-Dimensional Visualization HDViz ................ 63

   4.3 Summary ................................................. 67
# 5 Applications of Metamodelling Techniques to Laser Manufacturing Processes

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>69</td>
</tr>
<tr>
<td>5.2</td>
<td>Laser Epoxy Cutting</td>
<td>69</td>
</tr>
<tr>
<td>5.3</td>
<td>Glass Cutting</td>
<td>72</td>
</tr>
<tr>
<td>5.4</td>
<td>Laser Drilling</td>
<td>78</td>
</tr>
<tr>
<td>5.5</td>
<td>Oxygen Cutting</td>
<td>82</td>
</tr>
<tr>
<td>5.6</td>
<td>Sheet Metal Cutting</td>
<td>91</td>
</tr>
<tr>
<td>5.7</td>
<td>Summary</td>
<td>99</td>
</tr>
</tbody>
</table>

# 6 Conclusion and Outlook

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Conclusion</td>
<td>101</td>
</tr>
<tr>
<td>6.2</td>
<td>Outlook</td>
<td>102</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Reduced Models</td>
<td>102</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Metamodelling Performance</td>
<td>103</td>
</tr>
<tr>
<td>6.2.3</td>
<td>Sensitivity Analysis and Visualization</td>
<td>103</td>
</tr>
<tr>
<td>6.2.4</td>
<td>Commercialize</td>
<td>103</td>
</tr>
</tbody>
</table>

# A The HDViz visualization tool

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Focus of the dissertation ................................. 3
1.2 Procedure of metamodeling techniques ................. 11

2.1 Full Factorial Design ................................. 15
2.2 Two-levels fractional factorial designs .................. 16
2.3 Central Composite Design ............................. 18
2.4 Box-Behnken Design ................................. 18
2.5 Latin Hypercube designs .............................. 21
2.6 Example of Space-filling designs ....................... 22

3.1 Architecture of RBFN ................................. 28
3.2 Different basis functions .............................. 29
3.3 Different widths of a Gauss and Multiquadric functions .... 29
3.4 3D surface plot of an analytical function ............... 30
3.5 Required steps for generating a one-shot metamodel .... 31
3.6 One-Shot metamodel representation .................... 31
3.7 General procedure of SAO. ............................ 32
3.8 Representation of the density function .................. 33
3.9 Flow chart for genetic algorithms ...................... 34
3.10 Flow diagram of the one-shot metamodel for piecewise functions .... 35
3.11 Flow diagram of the proposed smart sampling method. .... 37
3.12 1D Plot of Equation 3.21. ........................... 39
3.13 Smart iterative approach applied to one-dimensional analytical function . 40
3.14 Two-dimensional contour plot of the continuous Equation 3.27. . . . . . . . 43
3.15 Example of iterative sampling method of a applied to continuous response 44
3.16 Two-dimensional contour plot of a piecewise function . . . . . . . . . . . . 45
3.17 Example of iterative sampling method of a applied to discontinuous response 46

4.1 Elementary Effect sensitivity measures . . . . . . . . . . . . . . . . . . . . 52
4.2 Morris Screening graph. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
4.3 Clique Graph for the Ishigami function. . . . . . . . . . . . . . . . . . . . . 56
4.4 Screenshot of memoSlice . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58
4.5 A modified screenshot of the Scatterplot view of memoSlice . . . . . . . . . 59
4.6 A modified screenshot of the HyperSlice view of memoSlice . . . . . . . . . 60
4.7 A modified screenshot of the 3-D view of memoSlice . . . . . . . . . . . . . 61
4.8 Local optimization performed in memoSlice . . . . . . . . . . . . . . . . . . 62
4.9 Sketch of the concept of persistence in MS complex . . . . . . . . . . . . . 64
4.10 A modified screenshot of HDViz . . . . . . . . . . . . . . . . . . . . . . . . 65
4.11 Examples of MS complex in HDViz . . . . . . . . . . . . . . . . . . . . . . 67

5.1 Parameter space analysis of Epoxy Cut . . . . . . . . . . . . . . . . . . . . 71
5.2 Evolution of the two-dimensional contour plot of the Epoxy metamodel . . 71
5.3 Quality assessment of Epoxy metamodel . . . . . . . . . . . . . . . . . . . . 72
5.4 Process map of maximum free-electron density . . . . . . . . . . . . . . . . 75
5.5 RBFN Network for the electron dynamics in glass cutting . . . . . . . . . . 75
5.6 Process map of the interpolated maximum free-electron density . . . . . . 76
5.7 Quality assessment of electron density metamodel . . . . . . . . . . . . . . 77
5.8 Classification of the parameter space of the electron density metamodel . . 78
5.9 Laser drilling process . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79
5.10 Laser drilling in memoSlice . . . . . . . . . . . . . . . . . . . . . . . . . . . 81
5.11 Laser drilling in HDViz . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81
5.12 EE and Clique graph of laser drilling . . . . . . . . . . . . . . . . . . . . . 82
5.13 Schematic of the oxygen cutting process. 83
5.14 Oxygen cutting in memoSlice 85
5.15 Oxygen cutting in memoSlice 86
5.16 Parallel coordinate and Box-whisker diagram of Oxygen cutting 87
5.17 Optimization of oxygen cutting in memoSlice 88
5.18 Oxygen cutting in HDViz 89
5.19 Global sensitivity analysis of oxygen cutting 91
5.20 Schematic of conventional laser cutting process 92
5.21 Surface Roughness in sheet metal cutting 93
5.22 Architecture of RBFN for sheet metal cutting 94
5.23 Contour plots of different metamodels in sheet metal cutting 95
5.24 Parallel coordinate plot and Box whisker diagram of sheet metal cutting 97
5.25 EE of sheet metal cutting 97
5.26 Optimization through process map of metal cutting 98
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Comparison of the classical DOE and simulation DOE.</td>
<td>20</td>
</tr>
<tr>
<td>4.1</td>
<td>Sobol indeces of Ishgiami function</td>
<td>56</td>
</tr>
<tr>
<td>4.2</td>
<td>Three two-dimensional cases used as test functions in the optimization algorithms.</td>
<td>66</td>
</tr>
<tr>
<td>5.1</td>
<td>Process parameter design domain of Epoxy cut.</td>
<td>70</td>
</tr>
<tr>
<td>5.2</td>
<td>Process Parameter design space of electron density model</td>
<td>74</td>
</tr>
<tr>
<td>5.3</td>
<td>Validation measures of electron density metamodel</td>
<td>76</td>
</tr>
<tr>
<td>5.4</td>
<td>The five dimensional parameter settings required for minimizing the conicity</td>
<td>80</td>
</tr>
<tr>
<td>5.5</td>
<td>Process parameter domain of oxygen cutting.</td>
<td>84</td>
</tr>
<tr>
<td>5.6</td>
<td>Sample sizes and number of model runs in the global sensitivity analysis</td>
<td>90</td>
</tr>
<tr>
<td>5.7</td>
<td>Process design domain of sheet metal cutting</td>
<td>94</td>
</tr>
<tr>
<td>5.8</td>
<td>Quality of the metamodels A-E for sheet metal cutting</td>
<td>96</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction and Overview

1.1 Scope of Virtual Production Intelligence

Over the last decades, the global interest of industries is strongly affected by economic, ecological, and social trends such as globalization, global warming, shortage of energy resources, or even financial crises [154]. Due to these trends, several complexities rise in the production sector. For example, shorter product life cycles, a global price competition, expensive resources, and faster changing market demands put the production companies under pressure to make their work flows more efficient and more scalable as well as to provide reliable and cost-effective responses to unpredictable market changes [157].

In order to cope with these complexities, production recently moved progressively into the computational field in order to better understand, predict, and mainly control the behavior of complex, socio-technical production systems. Moreover, industries are undergoing a paradigm change by installing their own simulation competence in house. From the technical perspective of sub-systems such as manufacturing machines, the complexity can be reduced to main functional characteristics and an interaction law that can be described by physical or phenomenological models [13]. This led to an explosion of tools that are being unified into a new notion in production, the virtual production system. This virtual system is based on a unifying state of the art knowledge from the real and virtual worlds, bringing together the scientific knowledge and technical Know-How about the process. The outcome are operative tools, namely 'process maps' and 'visual design' apps aimed to get more skilled developers and operators.

Starting by the definition of a map which is a picture or representation of the earth’s surface, showing how things or locations are related to each other by direction as well as distance. Process maps represent a way of visualizing the continuous relationship between the process parameters and the output criteria. It visualizes the
actual (current process parameter configuration) as well as the target (optimized process parameter configurations) locations, and the different roots or directions to attain such an optimized parameter setting.

Implementing a virtual production system is a challenging approach. This is due to the fact that real production systems are hardly fully predictable. Production systems are constrained to different dynamical market changes and characterized by complex general and machine specific interactions, uncertainties and unknowns. For example, laser system with the same specs behave different depending on components and life-cycle (such as optics degradation).

These complications make predicting the behavior by any deterministic computer model not only inaccurate but sometimes misleading [188]. Thus, it is crucial to implement a virtual production system that supports iterative design and offers the needed knowledge exploration by the aid of intuitively operating intelligent tools. The main focus of these intelligent tools is to enhance learning capabilities. This leads to a new concept called Virtual Production Intelligence [138].

Building an intelligent virtual production system requires a basic knowledge and a comprehensive understanding of how production facilities behave at relevant levels [1,18,153]. In this work, the concept of metamodeling is introduced to virtual production systems. Here, one has to distinguish the difference between a metamodel and metamodeling. A metamodel is an approximation model that mimic the behavior of the simulation up to a required accuracy. This model, also known as surrogate model, is not only faster but also less expensive than physically motivated simulation model that is constantly questioned when running an operation on the model. However, metamodeling is defined as a fast and frugal approach to validate and iteratively improve physical models. It is an approach which indicates how to build a physical model. The main outcome is a strongly simplified error control due to fast detection of the area of applicability. Moreover, metamodels enable global exploration and optimization within a multi-dimensional parameter space. This work proposes a methodology, which is illustrated in Figure 1.1, to extract knowledge from complex processes is intended to: i) create new opportunities of computer based analysis and exploration for members of business administration, engineers or even students; ii) compromise a systematic gain and improve an understanding of the underlying behavior of the system; and iii) acquire new possibilities of knowledge transfer between experienced and inexperienced users.
Figure 1.1: The focus of the dissertation is to extract knowledge from complex systems by analyzing the data generated from real (sparse) as well as virtual world (dense).

1.1.1 Knowledge Extraction in Complex Systems

Knowledge, which is shown in the center of the Figure1.1, is the central feature of post-industrial societies, since it enables individuals to make better informed decisions in order to develop, optimize, and interpret complex applications [3]. Stephen Hawking mentioned in his book [64] that scientists became the bearers of the torch of discovery in the quest of knowledge. They agreed that in order to analyze any complex phenomena at a deeper or at a relevant level, it is not sufficient to only know how the phenomena behave, but also why it is behaving as such and where in the parameter range is the applicability of the answer [64]. The only way to learn or understand any complex process is to first introduce a theory (the why-answer) and then apply a comparative critical thinking as well as looking at crucial experiments to falsify the theory (the where-answer: range of applicability). Karl Popper, the famous 20th century philosopher, discussed that validation of theories brings no increase of knowledge, since the theory remains as before. However, the theory should be used to make a prediction that can be tested, with the possibility of initiating improvements, that is the new knowledge! [131]. The development of a knowledge exploration passes through the following consecutive stages: i) a theory (Reference I) has to be stated first. It leads to many questions that need to be answered in order to satisfy the human curiosity about the phenomenon or the process, such as why, how, where, and what if; ii) in order to answer one of the previous questions, first observations (Reference II) then hypotheses have to be formulated which are basically an educated guess regarding the answer; and iii) hypotheses define laws which become
theories after validation and verification and remain valid until they are scientifically dis-
proven or falsified. [123,125,192]. In the context of virtual production systems, data is the
fundamental ingredient for the generation of new knowledge and improved theories. It
ranges from atomistic micro levels defining for example materials properties up to global
macro levels such as the availability of resources mostly important for a holistic design
and optimization of production processes. [109,110].

1.1.2 Experiments and Diagnostics

The process of experimentation, illustrated by the gray color in Figure 1.1, involves gaining
knowledge of the process and improving the deterministic models by empirical means to
handle and cope with not yet understood or too complex behavior. In the production
sector, manufacturers mainly attach functioning parameter settings for specific pre-defined
tasks to their machines in form of catalogues or technology tables. Due to the vast
number of experiments, those technology tables, which are generated by every machine
manufacturer for their machines specifically, are expensive/labor-intensive, highly time-
consuming, and exposed to failure occasionally by non-standard system states. They also
reveal only a sparse set of potentially beneficial operating points of the machine. The
data contained in these tables are produced by numerous experimental tests performed
by appropriate Design of Experiment (DOE) techniques as well as other experience-based
procedures. When the market demands change, experts of the process might provide
alternative machine solutions, however the machine operators face a comparable challenge
for choosing an appropriate machine parameter set that can cope with any changes of
machine components [39,139].

1.1.3 Modeling and Simulation

The main component of the virtual world, illustrated by the blue color in Figure1.1,
is simulation. Simulation has positively influenced research and development (R&D) in
the industrial environment. During the 1980s, virtual prototyping was successfully intro-
duced to the engineering community. It mainly involves computer-aided design (CAD),
computer-automated design (CAutoD) and computer-aided engineering (CAE) software
that help predicting fundamental design problems as early as possible in the design pro-
cess. Thus, before going into production, new products were first designed on the com-
puter by the simulation and then prototyping different concepts and selecting the best
design were simultaneously performed [169]. This is a main contribution to the idea of
green production leading to energy efficient and low-emission production. So far, the
conventional approach in modeling and simulation of manufacturing processes relies on performing several sets of individual simulations. Each individual simulation is characterized by a set of parameters in a high-dimensional parameter space and can be bounded by the following constraints: i) an increasing time required for performing the simulation model; ii) a rising number of parameter space dimensions; or iii) a highly nonlinear complexity of the simulation model. If the preceding constraints exist, the conventional analysis tasks such as optimization, sensitivity analysis, or design space exploration will easily become unmanageable. For example, it takes Ford Motor Company about 36 to 160 hours to run one crash simulation. Thus, the total computation time required for optimizing such a task, by executing only 50 iterations, would be around 75 days to 11 months, which is in practice limiting their applicability [62].

1.1.4 Reduced Modeling

Direct numerical simulation of dynamical systems has been an extremely successful ways for studying complex physical phenomena [158]. However, as details are added to the system, the dimensionality (the number of degrees of freedom given by the numerical nodes or volumes) of such simulations can increase to unmanageable levels of storage and computational requirements. One approach to overcome this problem is through model reduction. Reduced models are derived to avoid any unnecessary complexity and to reduce the computational time of large-scale dynamical systems by approximations of much lower dimension that can produce nearly the same input/output response characteristics [38]. The reduced modeling procedure takes a top-down approach and starts from the original partial differential equations and derives approximated analytical solutions or a set of ordinary equations using many mathematical approaches (e.g. asymptotic analysis [84],) physical phenomenological approaches (e.g. model parameter estimation [67]), numerical approaches (e.g. Proper Orthogonal Decomposition POD [22] or spectral decomposition) or dimensional model reduction (such as Buckingham-Pi theorem [14]). It is important to recognize that the input-output relationship approach is revealed by a set of discrete data that is collected in every single simulation run.

1.1.5 Metamodelling

In order to overcome the limitations of the costly simulations and the discrete nature of data (the result of every single simulation is a discrete data set) from reduced modeling, researchers are recently using fast and frugal metamodels. Metamodels create cheap numeric surrogates that describe the cause-effect relationship between the setting param-
eters defined as the input and the product quality variables defined as the output for manufacturing processes. Metamodeling techniques rely on a preselected sampling data known as training data. The procedure to select the best coordinates for the training data is addressed by DOE techniques which are subjects of recent mathematical investigations using e.g. entropy or likelihood methods. Sampling is considered a crucial step in the metamodeling procedure, since it affects the accuracy of the metamodel. After the selection of the training data that contains the parameters-criteria pairs, a global approximation needs to be performed. The global approximation is equivalent to finding the best continuous mapping of the discrete training data. Due to the fact that the method will be applied to deterministic or phenomenological computer models, the criterion that is considered in this work is an exact match (interpolation) of the continuous function to be guessed (metamodel) and the given discrete input-output relation (simulation).

1.2 State of the Art

A survey of DOE methods can be found in [11]. The basic form is the Factorial Design (FD) where data is collected for all possible combinations of different predefined sampling levels of the full parameter space [12]. However, for high-dimensional parameter space, the size of the FD data set increases exponentially [4] with the number of parameters considered. This leads to the well-known term *curse of dimensionality* defined by Bellman [4] which means that an unmanageable number of runs is consequently conducted to sample the parameter space adequately. Many researchers confirmed that the experimental design for deterministic computer analyses should be space filling which spreads design points fairly uniformly throughout the design region. [90]. Well-known space filling designs are the Orthogonal Arrays, Latin Hypercube designs, Hemmersley sequences, and uniform designs [41,78,165,197]. Approximation techniques evolve from the DOE theory, in which polynomial functions are used as response surfaces, or metamodels. Besides the commonly used polynomial functions, Sacks et al. [144,145] proposed the use of a stochastic model, called Kriging [26] in order to treat the deterministic computer response as a realization of a random function with respect to the actual system response. Neural networks are also applied to generate the response surfaces for system approximation [128]. Other types of models include radial basis functions RBF [36,41], multivariate adaptive regression splines (MARS) [47], least interpolating polynomials [9], and inductive learning [96]. A combination of polynomial functions and artificial neural networks is archived in [176]. Concluding which model is superior to the others is a systematic procedure that depends on several properties such as complexity of the response, required simulation time, metamodeling purposes (visualization, sensitivity analysis, optimization, etc ...).
Insights are gained through a number of studies [19, 56, 73, 91, 164, 166, 176]. After a metamodel is generated, it is validated by an additional simulation data before being used as a surrogate of the computation-intensive processes. Model validation and assessment is an interesting and yet a challenging task [124, 140]. Validating a metamodel is done through two methods: 1) the use of an additional sample point set to check the predicted function value with the real ones, and 2) the cross-validation method that includes leave-one-out or leave-k-out approaches [106, 108, 185]. Once built and validated, the metamodel is used to predict the model responses at unproven design locations quickly and repetitively. The concept of metamodeling techniques is a revolutionary approach that is forcing a change in the way production planning is achieved. It is applied in a lot of manufacturing and production tasks of industrial applications. [8, 32, 34, 43, 49, 95, 127, 135, 137, 155, 165, 186].

1.3 Research Questions

After reviewing the state of the art of metamodeling in literature, the core challenges of this thesis are concerned with two main research questions:

i) **RQ1: how to efficiently generate and validate a metamodel of a simulation model within the pre-specified requirements by the domain expert or engineer?** This question mainly focuses on the generation of a metamodel from simulation data and feeding back the results to the discrete reference data from the experiments as well as the comprehensive numerical simulations. This task is subjected to three sub-questions: i) how is the training data selected? (e.g. via a one-shot-classical-design or adaptive sampling design); ii) which type of interpolation technique should be used? and iii) how to validate and assess the adequacy (fast) and quality (frugal) of the fitted metamodel?

ii) **RQ2: how to improve a basic understanding of a complex model by replacing the simulation model by a metamodel?** This research question covers a wide range of sub-questions. However, in this thesis, the main focus will be on how to extract simple rules or behaviors from the metamodel. The three sub-questions, hence, i) which input parameters are the most important? ii) how to visualize a multi-dimensional domain space? and iii) how to decompose a multi-dimensional domain space?
1.4 Goals and contributions

The research questions are answered according to the following achievements:

i) The question of why and how metamodeling is considered as the main component in the virtual production system is clarified, following the works of Schulz and Al Khawli [160], Eppelt and Al Khawli [39], and Al Khawli et al. [82]. Herein, different approaches on how to generate, validate, and integrate metamodeling techniques in virtual production systems are presented.

ii) A novel smart sampling algorithm, detailed in Al Khawli et al. [80], for generating metamodels for piecewise responses (containing a discontinuity between feasible and non-feasible regions) is proposed, elaborated, and tested. The algorithm interfaces the sequential approximation optimization (SAO) and the Radial Basis Function Network (RBFN) for generating high quality metamodel with a minimal number of simulation runs.

iii) The metamodel is interfaced with conventional global sensitivity analysis methods such as Elementary Effect and Variance Decomposition as well as visualization methods such as Morse-Smale Complex or Hyperslices. This interface, presented in Al Khawli et al. [81], allows the possibility of creating effective information structures for data aggregation, retrieval and exploitation.

The three mentioned achievements basically aim to the following contributions:

1.4.1 Discrete to continuous support

One essential contribution of metamodeling is to overcome the drawback of experimental trials generating sparse data in the multi-dimensional parameter space. Multi-dimensional interpolation in metamodeling changes the discrete training data into continuous or dense data. As a result of this continuous support, a process map is built for every process that replaces the discrete points in the technology tables. It basically supports the direct transfer of knowledge from experts in research and experiments to the machine operator in a real production environment and vice versa, via a user interactive tool. Such tool provides a landscape that can be used as an integrated, dynamic, accessible, and visible navigation system that aims for a reliable and effective decision-making procedure. This guarantees a mutual benefit, since scientific advances and economical needs are assured especially when production demands are rapidly changing or new manufacturing systems are developed.
1.4.2 Metamodel-Based Design Optimization

Unlike direct optimization where the evaluations are completed using the detailed simulation models directly, a metamodel-based design optimization technique is a very attractive approach to decrease the required computational effort. The concept of metamodeling enables the benefit rating of alternative decisions based on improvements that are achieved by iterative design optimization where additional information of the process is obtained [63].

In order to make the number of simulation as few as possible, the optimization is performed while the metamodel is generated (in parallel). This is recently addressed by the Sequential Approximate Optimization (SAO) with metamodeling techniques [30, 119]. The SAO is an iterative sampling technique that reduces the number of simulation runs and simultaneously maximizes the information gain of every sampling step. This is done, by adding appropriate sampling points which are obtained by several computation intelligence techniques until a predefined termination criterion is satisfied [85, 86, 189]. Additionally, many applications in production have multiple objectives that conflict with each other. The scalar concept of optimality in multi-objective problems does not necessarily directly apply, since an improvement of one objective can lead to a deterioration of others. The optimization problem becomes a unique solution only if a compromise between the different criteria is specified. The unique solutions of the multi-objective problems are labeled as Pareto frontier [48] that contains different optimal solutions. Since metamodeling techniques generate accurate approximation, they can identify a good approximation of the actual Pareto optimal frontier. Approaches to solve multi-objective optimization problems with a metamodel for a single objective function or by direct approximation of the Pareto frontier can be found in [101, 163, 194, 196].

1.4.3 Towards knowledge exploration in simulation models

With the aid of metamodeling techniques, the process map can be extended to an online navigating cockpit that allows performing many engineering applications. Examples of such applications are parameter identification, sensitivity analysis, multi-dimensional visualization and multi-dimensional domain decomposition. These examples are vital steps in the planning process for achieving optimal designs and gaining valuable information. Sensitivity analysis and visualization can help not only identifying the most-influential parameters and quantifying their contribution to the model output, but also reducing the model complexity and enhance the understanding of the model behavior.
1.5 Thesis organization

This dissertation consists of 6 chapters. An introductory Chapter 1, which presents the focus of this work, research questions and main contributions of the metamodeling techniques. **Chapter 2** provides a review of the field design of experiments which plays a major role in generating metamodels. The question on how to choose the sampling point coordinates for ensuring a good and balanced predictive behavior of the approximation models is addressed in this chapter. **Chapter 3** provides two approaches for generating metamodels for continuous as well as piecewise responses. A novel smart sampling algorithm for generating metamodels with piecewise responses is proposed. It basically generates accurate metamodels iteratively by adding new sampling points in order to approximate responses with discrete changes. The new sampling points are inserted in the sparse region of the feasible (continuous) domain in order to achieve a high quality metamodel and also in the vicinity of the discontinuity in order to refine the uncertainty area between the feasible and non-feasible domains. In **Chapter 4**, the theory of two global sensitivity analysis techniques (Elementary Effect and Variance Decomposition) in addition to two visualization techniques (memoSlice and HDViz) is presented and explained. The overview given in this chapter is of importance since, these techniques are the main components of an interactive tool designed specially to virtual production systems. Then, **Chapter 5** presents a number of illustrative examples of metamodeling applied to five laser manufacturing applications with different parameter dimensions ranging from two to seven dimensions. A process map is generated for each application in order to illustrate and verify the proposed method showing its applicability and efficiency. Each of the previous chapter is concluded with a summary briefly stating its outcome. Finally, a brief summary, an outlook on future developments of the presented research work and conclusions are presented in **Chapter 6**.

1.6 Summary

Due to the complexity of apparently comprehensive numerical simulation (hard to check the error) and real world processing (hard to check the parameter values), the use of metamodeling techniques became popular among scientists and engineers. However, in order to make a metamodel operative (process map, virtual design), considerable efforts and choices need to be overcome in order to generate, validate, and integrate metamodels in virtual production systems. The goal of this work is to illustrate how to integrate metamodeling techniques in order to extract knowledge from multi-dimensional physical systems. The flowchart in Figure 1.2 summarizes the suggested framework for metamod-
eling techniques towards knowledge exploration in computer simulation.
Chapter 2

Design of Experiments

In order to successfully generate an accurate metamodel, an appropriate number of useful points is needed. Each of these useful points, also called sampling points, may contain vital information of the overall behavior of the system. The sampling points enable the metamodel to correctly assess the relationship between the input and output. For example, in order to build an accurate metamodel for a highly nonlinear response, many sampling points are needed to be performed for capturing all the spikes and thresholds (for example discontinuities) in the data. The major question that arises now is which algorithm to use in order to generate the sampling set that contains both the input (design parameter settings) and the corresponding scalar output (response values)?

A relevant procedure to efficiently define the coordinates of the sampling points of the dataset is the Design of Experiments (DOE). DOE techniques or experimental designs are defined as a systematic procedure that is carried out for extracting the most relevant qualitative information from a given data set and for identifying the most important design parameters that lead to optimal design [6]. There exist two main branches of DOE: (i) a-one-shot DOE where the sampling points are generated in one go and (ii) an adaptive DOE which relies on an iterative procedure to sample. In this chapter, the main principles and characteristics of DOEs are explained.

DOE has a rich history dating back to the 18th century applied to various theoretical developments and practical processes in many fields. In 1753, Lind carried out a controlled experiment to develop a cure for the Scurvy disease [111]. In 1877, Charles S. Pierce developed a theory of statistical interference which had focused on the significance of randomization-based interference in statistics [130]. In 1935, Fisher published his innovative book 'The Arrangement of Field Experiments' [42]. His work dealt with agricultural
applications of statistical methods. In 1987, Box and Draper published an extensive work about empirical model building and response surface designs [11]. In 1986, Taguchi developed a methodology of robust design to improve the quality of products [172]. Recently, DOE has become a vital procedure in almost every industrial application that emphasizes on quality improvement.

When DOE techniques were introduced, they were specifically developed for analyzing real physical experiments that are considered to be stochastic. In these experiments, the response values were disturbed by different noise parameters. In order to generate a DOE study, three basic statistical principles are involved to cancel the effects of the noise and to increase the validity of the experiment by providing plausible information about the experiment: i) Randomization, which refers to the order of the performed trials of an experiment [77]. A randomized sequence helps eliminating the effects of unknown or uncontrolled parameters; ii) Blocking, which allows for an individual assessment of the experiments independent of noise effects especially when randomizing a parameter input is impossible or too costly. This is done by grouping out all the experiments that are expected to behave similar under the noise effect; and iii) Replication, which is based on repeating the complete experiments with the same parameter settings. The mean and variance of the sample response can be calculated in order to determine the expected response value error.

This thesis focuses on generating DOE for computer simulation models, where the data is considered to be deterministic [114]. This means that repeating the computer simulation with the same factor settings yields exactly the same response. Consequently, the statistical theory that is built up for analyzing physical experiments (randomization, blocking and replication) does not apply. In this chapter, the characteristics and applicability of several DOE methods are discussed. The contribution of this chapter to the first research question RQ1, defined in Section 1.2, of the thesis is to: i) develop a basic understanding on how a model behaves in situations where the underlying mechanisms are not well understood, and where real-world data are limited or even non-existent; ii) discuss the issues that simulation analysts should be aware of as they prepare, collect, and analyze output from a simulation model; and iii) guide the selection of appropriate designs for computer simulation models focusing on the characteristic of their simulation model.

2.1 Classical DOE

Classical DOE are mainly used in various experimental designs [12]. They tend to spread the sampling points around the boundaries and only place a few points inside the de-
2.1.1 Factorial Designs

An experimental design is called factorial design if the \( p \) parameters governing the system are varied only on a finite number \( k \) of predefined levels. A full factorial design measures the response of every possible combination of parameters and parameter levels. The total sampling size \( n \) for a full factorial design is determined by

\[
    n = k^p
\]

Figure 2.1 (a) and (b) show a full factorial design for three parameters evaluated on two and three levels respectively. The larger the value of \( k \), the better the space-filling properties of the design becomes. A full factorial DOE is not feasible when more than six parameters are being considered. As shown in Figure 2.1 (c), the sampling size grows exponentially with the number of parameters considered. For a three-levels full factorial design, 4 parameters requires 81 experiments, illustrated by the blue line in Figure 2.1 (c). While 7 parameters require 2187 experiments which may be very expensive.

![Figure 2.1: a) A two-levels full factorial design with 3 parameters requiring 8 experiments marked by the orange spheres. b) A three levels full factorial design with 3 parameters requiring 27 experiments marked by the orange spheres. c) Exponential increase of sampling size with the increase of number of parameters on a three sampling levels.](image)

Investigating parameters at many levels may result in very expensive designs. This is why, in many experimental testing, parameters are generally studied at only two levels [177].
Two-levels full factorial designs are easy to construct as they provide the smallest number of runs of a full factorial design. Additionally, it is one of the most widely used screening methods in computer simulations, since there are two levels in every factor, which makes it possible to fit a linear metamodel [88] that includes all the interaction according to:

$$f(x) = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij} x_i x_j + \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \beta_{ijk} x_i x_j x_k + \cdots$$  \hspace{1cm} (2.2)

where $\beta_0$, $\beta_i$, and $\beta_{ij}$ describe the mean of the response values, the main effects, and the interaction effects respectively. Even though two-levels factorial design can estimate all main and interaction effects, it still requires a large number of experimental runs to screen a small number of parameters [77]. However, experience suggests that high order interaction effects are not usually significant and a model including only main and low order interaction effects can be a good fit for the response function. This interpretation leads to the two-levels fractional factorial designs [87].

### 2.1.2 Two-levels Fractional Factorial Designs

The fractional factorial design is an alternative approach for the full-factorial DOE when the number of parameters increases. It consists of a chosen subset of all complete full factorial design sampling sets. If it can be assumed that certain high order interactions are negligible, then only a fraction of the two-levels factorial design is needed to get the information of the main parameters and the low-order interactions. This design is represented by generating a $2^{n-l}$ sampling set, where $l$ defines the size of the fraction of the full factorial design, and the reduced number of samples is equal to $(1/2^l)$ of the full design as shown in Figure 2.2 [87]. By using a suitable fractional factorial design,

![Figure 2.2: Two alternative fractional factorial design of type $2^{3-1}$, where the experiments are illustrated by the orange spheres.](image)
it is possible to reduce the number of experiments and decrease the computational cost while many effects are still considered [177]. The fractional factorial experiments are used for screening specified parameters and usually are performed in the early stage of an experimental process. They must be chosen carefully, since they may confound with a significant interaction effect with another effect; and therefore no information can be gained about the individual interaction effects within this confounded structure. The issue of confounding introduces the concept of resolution of a design [12]. Resolution is the ability to separate main effects and low-order interactions from one another. The most important fractional designs are those of resolution III, IV, and V: Resolution III designs focus on just finding important main effects, where main effects are not confounded with any other main effect. However main effects are confounded with two-factor interactions which may also be confounded with each other. A well-known design of Resolution III is Plackett-Burman design [114]. Moreover, Resolution IV designs are used when no main effects are confounded with any other main effect or two factor interactions but rather two factor interactions are confounded with each other [6]. Finally Resolution V designs are used when no main effect or two-factor interactions are confounded with any other main effect or two-factor interactions, but rather two-factor interactions are confounded with three-factor interactions.

2.1.3 Central Composite Designs

Due to the fact that two-levels factorials or fractional factorials generate samples at each factor at only two levels, only main and second-order interactions effects are estimated since it is not revealed what happens to the response between the parameters boundaries [94]. A solution for this is by either increasing the number of levels which requires more sampling points, or by using the central composite design (CCD) that is known for fitting a quadratic metamodel. As shown in Figure 2.3, CCD contains an embedded factorial or fractional factorial design with a center point that is augmented with a group of star points that allow the estimation of the curvature. If the distance from the center of the design space to a factorial point is 1 unit for each factor, the distance from the center of the design space to a star point is \(\pm \alpha\) with \(\alpha > 1\). The precise value of alpha as well as the number of center point runs of the design depends on certain properties desired for the design and on the number of parameters involved [94]. A central composite design contains twice as many star points as parameters in the design [88]. The star points represent new extreme values (low and high) for each factor in the design. For more information on the CCD, the reader is referred to [114].
2.1.4 Box-Behnken Designs

Other designs for fitting quadratic metamodels are the Box-Behnken designs (BBD). BBD are used when the evaluation of extremal factor settings is related to excessive costs [195]. They are constructed similar to the CCD designs but excluding the extremal factor settings or the corners of the factor space and the center point [12]. As illustrated in Figure 2.4, all the design points are on a sphere of radius $\sqrt{2}$ from the center point, and no corner points are included. Since no experiments are performed at extremal factor settings, BBD are not suited for predicting response values at the vertices of the factor space.

2.2 Simulation DOE

For a multi-dimensional parameter space, the size of the classical DOE sampling data set increases exponentially with the number of parameters considered [88]. This leads to the well-known term **curse of dimensionality** first defined by Bellman [4] which means
that an unmanageable number of runs should be conducted in order to sample the design space. When the simulation runs are extensively time consuming, the classical DOE methods as described by the previous section could be inefficient or even inappropriate. Kleijnen discusses in his work [89] that DOE is not used as widely or effectively in the practice of simulation as it should be. He supports his discussion with the following reasons: (i) Most designs are originally developed for real world experimentation that contains noises or random error rather than developed specifically for simulation models which are mainly deterministic; (ii) Many simulation analysts are not fully convinced of the benefits of DOE. Instead of using a simple experimental design, they end up making runs to measure performance for only a single system specification, or they choose to vary a handful of the many potential parameters one-at-a-time. Their efforts are focused on building- rather than analyzing- the simulation model.

Most of DOE publications applied to simulation modeling suggest modifications to improve the efficiency of specific kinds of bias in specific applications [87] rather than explaining the variety of methods available to the simulation analysts. It should be noted here, that the simulation experiments are fundamentally different from the real-world experiments. In real experiments, it is impractical to investigate many parameters: maximum ten parameters is the limit [89]. Moreover, it is also hard to experiment with parameters that have more than a few values: five values per parameter is the limit [88]. However, in simulation experiments, and with the aid of the advances in computational power, the boundaries of the real experiment constraints (such as costs, efforts, etc) become irrelevant. The computer codes may have hundreds of inputs and parameters each with many values. Consequently, a multitude of scenarios may be simulated. Sanchez et al. [150] listed some of the main differences between classical DOE and simulation DOE denoted in Table 2.1. There seems to be an agreement among scientists that a proper experimental design of sampling for deterministic models that depend on many parameters over a large design space should be space-filling [92]. These types of DOEs aim at spreading the design points within the entire design space, which is desired when the form of the metamodel is unknown and when interesting phenomena can be found in any region of the design space [88]. A design with good space-filling properties means that the analysts do not need to make many assumptions about the nature of the response surface. Such designs also provide high flexibility when estimating a large number of linear and nonlinear main and interaction effects (will be discussed in detail in Chapter 4). They also provide general bias protection when fitting metamodels of specific forms. The efficient and effective methods in space filling designs are the Latin Hypercube design, Monte Carlo designs, and Hammersley designs [92]. These designs can mainly be used in factor screening by generating a metamodel first, and second classifying the im-
Table 2.1: Comparison of the classical DOE and simulation DOE.

<table>
<thead>
<tr>
<th>Classical DOE</th>
<th>Simulation DOE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small or moderate number of parameters</td>
<td>Large number of parameters</td>
</tr>
<tr>
<td>Linear low order effects</td>
<td>Nonlinear behavior</td>
</tr>
<tr>
<td>Negligible higher-order interactions</td>
<td>Substantial higher-order interactions</td>
</tr>
<tr>
<td>Homogenous error</td>
<td>Heterogeneous error</td>
</tr>
<tr>
<td>Normal error distribution</td>
<td>Various error distributions</td>
</tr>
<tr>
<td>Black Box Model</td>
<td>Substantial expertise exists</td>
</tr>
<tr>
<td>Univariate response</td>
<td>Many performance measures of interest</td>
</tr>
</tbody>
</table>

important parameters. This will be addressed by the global sensitivity analysis in Chapter 4 in details.

2.2.1 Latin Hypercube Sampling

Latin Hypercube Sampling (LHS) was introduced by McKay [105] for situations involving a relatively large number of parameters. It provides a flexible approach of constructing efficient designs for quantitative parameters since it has good space filling properties [12]. Let $k$ be the number of parameters and $n$ the number of design points desired with $n \geq k$, which means that $n$ levels per factor are defined. The low and high levels of factor $x_i$ are specified as 1 and $n$ respectively, and the set of specified factor levels are $1, 2, \ldots, n$. In Random LHS, each column of the design matrix is a random permutation of the factor levels. Unlike the two-levels factorial design, the LHS design provides some information about the interior of the experimental region. Figure 2.5 shows two sampling distributions of LH design for two parameters with four sampling levels. Popular measures for assessing an LH design include maximizing the minimum Euclidean distance $d$, as shown in Figure 2.5 (b), between the sampling points. This basically means that the points are spread out as much as possible in the design space [74, 116, 129].
Figure 2.5: Construction of Latin Hypercube designs with 4 sampling points in 2 dimensional design space: a) Step 1: Randomly place one sample in each cell along the main diagonal of the grid (poor distribution), Step 2: Randomly shuffle the $x_1$ and $x_2$ coordinates (good distribution when using maximin LH design, i.e. design-maximizing the minimal distance of the sampling points).

### 2.2.2 Monte Carlo Sampling

Another very widely used sampling technique is Monte Carlo. It is based on a random sampling strategy which is drawn from a specified distribution (typically normal distribution or uniform distribution). This method is used in various fields for problem solving, mostly due to the simple structure of the computation algorithm [66]. An example of Monte Carlo sampling design for a two dimensional model with 50 sampling points using a uniform distribution is plotted in Figure 2.6 (a).

### 2.2.3 Hammersley Sequence Sampling

The Hammersley sequence belongs to a group called low-discrepancy sequences. The discrepancy is a measure of the deviation from a uniform distribution. Hammersley Sequence Sampling (HSS) provides a low-discrepancy experimental design by placing $n$ points in a $p$-dimensional hypercube [78]. This provides better uniformity properties over the $p$-dimensional space than LH. Example of HSS designs with 50 sampling points in a two-dimensional parameter space is illustrated in Figure 2.6 (c). In [78], it is shown that HSS designs require significantly fewer samples to converge to the variance of a derived distribution than Latin hypercube designs and Monte Carlo sampling designs. Thus, it verifies the good uniformity properties of these types of designs in $p$ dimensions. The reader is referred to [78] for a definition of Hammersley points and an explicit procedure for generating them. The algorithm in [70] has been modified by Diwekar [33] to generate
2.3 Adaptive DOE

The previous sampling designs are executed and generated all at once. They are, hence, called one-shot DOEs. In many simulation applications, the total number of sampling points which corresponds to the total number of function evaluation is drastically limited by the computational cost. To cope with this challenge, adaptive DOE have received a great interest from the mathematical and engineering communities for its flexibility and adaptability over the one shot DOE and its effectiveness in sequential decision making under uncertainties [24, 87, 187]. A DOE is considered to be adaptive when the information from the experiments (inputs and responses) as well as the information from the metamodel are used for selecting the next sample point. An adaptive approach generally begins with an initial one shot design. The adaptive DOE approaches are classified into two branches (i) DOE for optimization and (ii) DOE for generating global high accurate metamodels [195]. Iterative sampling methods are used for optimization purposes as they try to locate the optimum by adding new samples based on the progress of the optimization, i.e., points close to the estimated optimum [151, 152]. Additionally, iterative DOE focus on generating accurate metamodels based on the estimation of cross validation (CV) errors where new samples are generated based on the maximum predicted error [15, 99, 102]. An alternative strategy applied to minimize the maximum prediction variance at each step is also available [98, 141]. It is normally applied to the Kriging metamodel which provides a prediction of the variance. Sacks et al. [145] uses this strategy to derive a sequential IMSE design using the Kriging metamodel [57, 90]. Another famous

![Diagram](image-url)
2.4 Summary

For the generation of a metamodel, an appropriate number of sampling points is needed. These points can be selected via DOE techniques to gain the maximum number of information. This is crucial for analyzing how a model behaves focusing on the characteristics of the underlying relationship between input and output. In this chapter, diverse one shot and iterative DOE techniques are presented in order to prepare the reader for the smart sampling technique presented in Chapter 3. In order to select the proper DOE technique for a simulation model, a detailed plan has to exist. It is important to know if the simulations could run in parallel or series. Additionally, it is vital to have a prior knowledge about the response to be analyzed, since the total sampling size depends on the number of the parameter space as well as the complexity of the simulation model. For example, in order to build an accurate metamodel for a highly nonlinear response, many simulations are needed to be performed for capturing all the spikes and thresholds in the data.
Chapter 3

Metamodelling: From Theory to Practice

The process of learning from simulation is a common problem that arises in many scientific disciplines. The focus is to find a rule that helps to realize or deduce information about the process from a given set of data. However, the process of learning from simulation results becomes more difficult when there is insufficient data. This happens basically when simulation time as well as the number of parameters increase [82]. To overcome these difficulties, metamodels are recently being used as fast and frugal surrogates to physical simulation models [40]. The idea of metamodels is to guess or estimate a function from some exemplary input-output pairs with little or no knowledge about the form of the response function [77]. In general, the metamodel contains functions of the form \( f : \mathbb{R}^p \rightarrow \mathbb{R}^c \) which maps \( \mathbb{R}^p \) (the parameter space) to \( \mathbb{R}^c \) (the criteria space). For simplicity, in the following work the criteria space is considered to be a univariate output, and the mapping is thus recognized by \( f : \mathbb{R}^p \rightarrow \mathbb{R} \).

In this chapter, an investigation aimed to the first research question RQ1 presented in Section 1.4, which is mainly focusing on how to generate and validate a metamodel of a simulation model is given. Different metamodeling approaches (one-shot and adaptive) for continuous as well as piecewise responses are discussed in detail. Generating a metamodel requires different aspects and trade-offs to judge if it is acceptable or not. An accepted metamodel is defined by a balance or compromise between the user predefined criteria [45, 73] which mainly are: i) **Accuracy**: the measure of deviation of the predicting metamodel from the real simulation model; ii) **Robustness**: the degree of achieving good accuracy for different designs. This metric indicates whether a modeling technique is highly problem dependent; iii) **Timing**: the measure of the computational time required
for generating the metamodel and predicting a new response of new parameter settings; iv) **Sensitivity to Parameters**: the capability of extracting information regarding the main effect and the interaction effect of the input parameters on the output criteria; and v) **Ease of implementation**: Simple methods should require easy implementation in generating the metamodels.

Additionally, a new optimization algorithm which is based on generating high quality metamodels for piecewise responses is presented [82]. The proposed approach extends the Sequential Approximate Optimization (SAO) procedure, which uses the Radial Basis Function Network (RBFN). It basically provides a global optimization and generates accurate metamodels iteratively by adding new smart sampling points, in order to approximate responses with discrete changes.

### 3.1 Metamodeling for Continuous Responses

After defining the design objectives, identifying the input parameters and output criteria of the problem, and defining the lower and upper bounds for the domain space, a metamodel can be generated by one of the two approaches: the one-shot approach or the iterative approach.

#### 3.1.1 One Shot Approach

A one-shot metamodel approach consists of three fundamental steps: 1) the creation and extraction of a training data set (sampling), 2) evaluation of a reference model, and 3) the mapping of the discrete sampling points in a continuous relationship (interpolation).

**Step 1:** The procedure to efficiently sample the parameter space is addressed by the Design of Experiments (DOE) techniques discussed in Chapter 2. It addresses mainly where to place the design point in the design space. When the sampling points are generated all at once, the metamodeling approach is called a **one-shot approach**. The sampling set that represents the input parameters and the output response is called the training data set $T$, and is defined by

$$T = \{(x_i, y_i)\}_{i=1}^n,$$

(3.1)

where $x$ is the parameter vector, and $y$ is the output, $n$ is the total number of data pairs, and $i$ is an index of the runs that ranges from 1 to $n$. 

Step 2: A reference or original model, which might be analytical functions, a reduced model, full numerical simulation, or even an experiment, is required for metamodeling. The more information available about the reference model, the better and more efficient the metamodeling generation becomes. Helpful information includes: i) the state of the model implementation (for example if stable or possible to interface with other tools or programming languages), ii) the usage requirements (licensing information), system type (deterministic or stochastic), information of the inputs (dimensionality, ranges, or sensitive parameters), and information about outputs (discontinuities, non-linearities) [61]. The main assumption of this thesis requires the use of a validated reference model with an adequate quality. A low quality reference model generates a low quality metamodel. This is summed up by the popular expression between computer scientists which says Garbage In, Garbage Out (GIGO).

Step 3: A scattered data approximation technique for mapping the discrete sampling points to a relationship is required. In scattered data approximation, which is one of the most common problems that arises in many scientific disciplines [191], the true response $y$ is replaced by an approximated response (metamodel) $f(x)$. For an arbitrary design point $x$, the relationship of the metamodel and the true response is defined by:

$$y - f(x) = \varepsilon,$$

where $\varepsilon$ is the approximation error. Many types of metamodels (Polynomial Regression [118], Moving Least Squares [97], Multivariate Adaptive Regression Splines [47], Kriging [25, 93], Artificial Neural Network [65, 121, 133], Support Vector Machines [27], Symbolic Regression [181], etc.) have been used to approximate deterministic black box models. However, the main two common questions that arise when applying metamodeling is: Is any technique superior to the others? On which basis should the various techniques be used? Recently, many authors have investigated comparative studies for the selection of metamodeling types [20, 72, 83, 100]. They agreed that there is no superior or universal model that is able to perfectly fit any kind of a true response. An appropriate metamodel choice depends on the problem itself, the metamodeling algorithm and the right selection of several model parameters which are required by these interpolation techniques [143]. In this work, the radial basis function network (RBFN) is used as the main interpolation technique. RBFN is a three-layer feed forward neural network that uses the radial basis functions. It is well known for its accuracy and its ability to generate multidimensional interpolations for complex nonlinear problems. The RBF method was introduced by Hardy [?] who originally presented the method for the multiquadric (MQ) radial basis function. The method emerged from a cartography problem, where a bivariate interpolation of sparse and scattered data was needed to represent
a given topography [17]. However, none of the existing interpolation methods (Fourier, polynomial, bivariate splines) are satisfactory because they are either too smooth or too oscillatory [8]. Furthermore, the non-singularity of their interpolation matrices is not guaranteed. In 1982, Richard Franke tested the MQ method on 32 of the most commonly used interpolation problems [45]. Franke assumed that the interpolation matrix associated with the multiquadric radial function is unconditional non-singular (invertible). This was later proven by Micchelli [112]. An RBFN, shown in Figure 3.1, consists of an input layer which is modeled as a vector of real numbers, a hidden layer that contains nonlinear basis functions and an output layer which is a scalar function of the input vector [82].

![Figure 3.1: Architecture of Radial Basis Function Network (RBFN).](image)

The output of the network $f(x)$ is given by:

$$f(x) = \sum_{i=1}^{n} w_i h_i(x),$$  \hspace{1cm} (3.3)

where $n$, $h_i$, and $w_i$ correspond to number of sampling points of the training set, the $i^{th}$ basis function, and the $i^{th}$ weight respectively. The radial basis functions are a special class of functions for which their response increases or decreases monotonically and in all directions according to a distance from the central point. Out of many radial basis functions, the ones that are applied in this thesis are listed below and are illustrated in Figure 3.2:

- **Linear Spline**

  $$h(x) = |(x - x_i)|^p, \hspace{1cm} p = 1,$$  \hspace{1cm} (3.4)

- **Multiquadric function**

  $$h(x) = \sqrt{1 + \frac{(x - x_i)^T (x - x_i)}{r^2}},$$  \hspace{1cm} (3.5)

- **Gaussian function**

  $$h(x) = \exp\left(-\frac{(x - x_i)^T (x - x_i)}{r^2}\right),$$  \hspace{1cm} (3.6)
3.1.1 One Shot Approach

![Radial Basis Functions](image)

Figure 3.2: Different Radial Basis functions in two-dimensional parameter space centered at 0 with widths $r = 1$. Showing from left to right the linear spline, multiquadric, and Gauss respectively.

where $x_i$ and $r$ represent the $i^{th}$ sampling point and the width of the basis function respectively. The shape parameter $r$ controls the width of the basis function. The larger or smaller $r$ changes, the narrower or wider the function gets. This is illustrated in Figure 3.3.

![Gauss and Multiquadric Functions](image)

Figure 3.3: (a) Gauss and (b) Multiquadric function centered at $x_i = 0$ with different widths $r$.

The learning process of the network is performed by applying the method of least squares with the aim of minimizing the sum-squared-error with respect to the weights $w_i$ of the model [126]. Thus, the learning/training is done by minimizing the cost function:

$$ C = \sum_{i=1}^{n} (y_i - f(x_i))^2 + \sum_{i=1}^{n} \lambda w_i^2 \rightarrow min, \quad (3.7) $$

where $\lambda$ is a regularization parameter which determines the relative importance of the smoothness of the function and $y_i$ is the criterion vector at point $i$. Solving Equation 3.7, leads to:

$$ w = (H^T H + \Lambda)^{-1} H^T y, \quad (3.8) $$

with
The chosen width of the radial basis function plays an important role in getting a good approximation. The $r$ value is selected according to $[?]$ and defined as

$$r = 0.81 d \quad d = \frac{1}{n} \sum_i d_i$$

(3.10)

where $d_i$ is the distance between the $i^{th}$ data point and its nearest neighbor.

In order to illustrate the steps of the one-shot metamodeling approach, an example using a two-dimensional function is considered. The analytic function, that is chosen in this test case and plotted in Figure 3.4, is defined by:

$$y(x) = \frac{1}{100} \left( (x_1^2 + x_2^2 - 62)^2 + (x_1^2 + 0.5x_1 + 0.5x_2 - 1.5)^2 \right).$$

(3.11)

![Figure 3.4: 3D surface plot of Equation 3.11.](image)

First, a four-levels full factorial design is used to generate a training set that contains 16 samples (Step1). The sampling points are represented by the gray spheres in Figure 3.5a. Second, the training set is evaluated by Equation 3.11 to obtain the response vector $y$ (Step2), shown in Figure 3.5b. Finally, the learning is done by RBFN with multiquadric basis functions that are centered on the training data. This is illustrated by Figure 3.5c. The final metamodel is shown in Figure 3.6.
3.1.2 Iterative Approach

Regardless of the metamodel use, there is always a concern of achieving high accuracy with respect to the sampling size, the sampling method and the metamodeling method. An important research issue associated with metamodeling is how to obtain a good accuracy for metamodels with a reasonable number of sampling points. This was recently addressed by the Sequential Approximate Optimization (SAO) technique [30, 120]. The SAO, shown in Figure 3.7, is an iterative sampling technique that reduces the number of simulation runs, and at the same time maximizes the information gain of every sampling step by adding appropriate sampling points until a predefined termination criterion is satisfied. In order to make the number of simulation for such systems as small as possible, the SAO approach is applied. The metamodel is constructed repeatedly by adding new sampling points in the sparse region \((x_{SR})\) of the parameter domain space which leads to...
an improvement of the quality of the metamodel [85,119].

Figure 3.7: General procedure of SAO.

In order to add infill points in the sparse region \( x_{SR} \), the density function, introduced by Kitayama [85], is constructed also by an RBFN metamodel according to the following two steps:

i) The output vector \( y \) is replaced by +1 and stored in the density output vector \( y_D \).

ii) The weights vector \( w_D \) is calculated as follows:

\[
w_D = (H_D^TH_D + \Lambda)^{-1}H_Dy_D.
\] (3.12)

In order to achieve a response that decreases monotonically between the sampling points, the Gauss function (Equation 3.6) is employed as the basis function. The width \( r \) of the basis function, chosen according to Nakayama [119], is defined by:

\[
r = \frac{d_{max}}{\sqrt{p/n}},
\] (3.13)

where \( d_{max} \) is the distance between the \( i_{th} \) data point and its farthest neighbor, \( p \) is the number of parameters, and \( n \) is the number of the training points. Figure 3.8 shows an illustrative example of the density function in one dimension. It shows the local minima which represent sparse regions, and the local maxima which represent the sampling points.

The additional point \( x_{SR} \) within the sparse region is acquired by minimizing the multidimensional density function:

\[
\min_{x} \quad D(x) = \sum_{i=1}^{n} w Dh_{Di}(x)
\] (3.14)

subject to \( l(i) \leq (x_i) \leq u(i) \).

There are several algorithms that can be used to solve Equation 3.14. On the one hand, local optimization algorithms (such as steepest descent, Newton, conjugate gradient and
Figure 3.8: Density function (a) of the corresponding one-dimensional parameter space with the corresponding sampling points (b). The maximum peaks in (a) correspond to the black dots in (b) which denote the locations of sampling points.

 simplex [179]) make use of gradient information attempts to find only a local optimum in the vicinity of a starting point that is defined by the user. As shown in Figure 3.8a, the density function has several local minima. This means that with a local optimization algorithm, it is not guaranteed that a local minimum is the actual global one, unless very specific settings are made (for example running the optimization algorithm many times while changing the location of the starting point). On the other hand, global optimization algorithms have proven to be more efficient in finding the global optimum. They are mainly categorized as: deterministic and heuristic algorithms [199]. Deterministic algorithms, such as linear programming, nonlinear programming, mixed-integer nonlinear programming, etc. take advantage of the analytical properties of the problem to generate a sequence of points that converge to a global optimal solution. However, for solving nonconvex or large-scale optimization problems, the deterministic methods may not be easy to derive a globally optimal solution within a reasonable time due to the high complexity of the problem [198]. Alternatively, the heuristic algorithms such as genetic algorithms, simulated annealing, tabu search, ant colony optimization and particle swarm optimization do not generate random sequence of points rather random variables [168]. Although the heuristic methods have the advantage of easy implementation and offer a better potential for complex problems, the obtained solution is not guaranteed to be a globally optimal solution and requires many more objective function evaluations [179].

In this thesis, the Equation 3.14 is solved by genetic algorithms due to the fact that the density function is very fast to evaluate. Genetic algorithms (GA) are the most widely known type of evolutionary algorithms (EA) which use techniques inspired by natural evolution [59]. Unlike gradient based optimization techniques, GA does not require gradient information and does not move sequentially from one point to the next one. The
core algorithm which is referred as the Canonical Genetic Algorithm (CGA) is introduced by Holland [69]. It is explained by the following steps:

![Flow chart for genetic algorithms.](image)

Figure 3.9: Flow chart for genetic algorithms.

i) Generate random population of size $n_p$ referring to the design parameter configurations (suitable solutions for the problem).

ii) Evaluate the fitness or objective function (density function $D(x)$) value of every individual of the population.

iii) Test if the termination condition is satisfied (maximum number of iterations or convergence criteria). If yes stop, and return the best solution in current population, otherwise go to the next step.

iv) Create a new population by performing:

   (a) Selection: Select three parent parameter configurations from the generated population according to a tournament selection (parents are selected according to their fitness, the better the fitness, in this case minimum density, the bigger the chance to be selected)

   (b) Crossover: According to a crossover probability $P_C$, the parents are crossed over to form a new offspring (children).

   (c) Mutation: According to a low mutation probability $P_M$, the children are mutated. Mutation changes one or more string values into a chromosome (representation of the parameter configuration in a string of binaries). This is to make sure that the algorithm does not fall in a local optimum solution.

   (d) The new offspring becomes the new population.

v) After generating the new population the algorithm, go to step ii) with new population, and repeat the process until the termination criteria is satisfied.

An example of the metamodeling iterative approach is provided in Section 3.4.1 at the end of this chapter.
3.2 Metamodelling for Discontinuous Responses

In many manufacturing processes, the functional relationship between the input parameters and the output criteria is represented mathematically by a piecewise function. Due to digital changes in the solution properties of many engineering applications (e.g. topology changes like a material cut-through), the model response is described by a piecewise continuous function. Identifying the region of the discontinuity is not explicit, since it is defined by many other parameter relationships or other physical interactions [107]. When applying metamodels to responses with discontinuity, they can provide very poor fits because metamodels are generally applied only to continuous responses, as the reason is that they mostly apply fully-steady basis functions. Similar to continuous responses there are two metamodelling approaches for responses with piecewise responses. However, additional steps need to be performed.

3.2.1 One Shot Approach

The six steps typically involved in constructing a one shot metamodel are: 1- sampling the design-space; 2- evaluating the response of the reference model and assigning a value for the discontinuity; 3- splitting the data; 4- interpolating of the feasible data; 5- classification of the domain space; and 6- merging the classification model and the interpolation model as shown in Figure 3.10

![Flow diagram of the one-shot metamodel approach for the output with a discontinuous response.](image)

Figure 3.10: Flow diagram of the one-shot metamodel approach for the output with a discontinuous response.

**Step 1:** and **Step 2:** are performed as defined in section 3.1.1, however, a discontinuity value $DiscVal$ is set by the user to represent and distinguish the discontinuity in the domain space. Thus, the domain space is classified into a feasible and non-feasible domain.
Step 3: Split the sampling data according to the DiscVal to two training data sets $T_F$ and $T_C$ such as

$$
T_F = \{(x_{Fi}, y_{Ci})\}_{i=1}^{n_F} \quad T_C = \{(x_{Ci}, y_{Ci})\}_{i=1}^{n}.
$$

where indexes $F$, $C$, $n_F$, and $n$ represent the feasible sampling data (no discontinuity values), the classification data (full data set), the number of feasible sampling data, and the number of the full data set respectively. The values of vector $y_C$ are re-scaled to either -1 that correspond to DiscVal, or 1 that correspond to feasible values.

Step 4: Construct a RBFN metamodel $f(x)$ of the feasible data set $T_F$ according to Equation 3.8.

Step 5: Perform a classification task in order to first decompose the design space into feasible and non-feasible regions and second detect the discontinuity. By applying the Cover’s theorem [23], the domain space $\Psi$ that is formed by a set of $n$ vectors $x_C$, can be split into two classes $\Psi_1$ and $\Psi_2$ by assigning a dichotomy of surfaces. This is done in Step 3 where a value of -1 is assigned to non-feasible regions and a value of 1 is assigned to feasible regions. An RBFN metamodel is used to perform a classification task. The domain space $\Psi$ is said to be separable if there exists a vector $w_C$ such that:

$$
C(x) = \sum_{i=1}^{n} w_{Ci} h_{Ci} (\|x - x_{Ci}\|) > 0, x \in \Psi_1, \quad (3.16)
$$

$$
C(x) = \sum_{i=1}^{n} w_{Ci} h_{Ci} (\|x - x_{Ci}\|) < 0, x \in \Psi_2.
$$

The discontinuity in the domain space is defined by:

$$
C(x) = \sum_{i=1}^{n} w_{Ci} h_{Ci} (\|x - x_{Ci}\|) = 0, \quad (3.17)
$$

where $w_C$ is defined similar to Equation 3.8 by considering the whole data set of size $n$, and using a first order linear spline basis which is defined according to Equation 3.4. In $C(x)$, the exact border of the discontinuity is not known due to the continuous interpolation, so results in the interval $]-1;1[$ indicate values with an uncertainty about whether the process yields a feasible result, that depends on how close the value is to either -1 or 1.

Step 6: Merge the classification model and feasible interpolation metamodel into a final metamodel function defined by:

$$
f_D(x) = \begin{cases} 
  f(x) & C(x) > 0 \\
  DiscVal & C(x) \leq 0
\end{cases}. \quad (3.18)
$$
3.2.2 Smart Iterative Approach

Following the work of Al Khawli et al. [80], the goal of this new approach, which is shown in Figure 3.14, is to develop an adaptive sampling method that enhances the effectiveness of generating metamodels for systems that contains a discontinuity. There are two ways to enhance the accuracy of the metamodel: 1) adding infill points in the feasible sparse region \( x_{SR} \); 2) adding sampling points next to the discontinuity \( x_{DISC} \) to reduce the uncertainty region that ranges between -1 and 1. The proposed methodology involves in addition to the six major steps listed previously, two additional steps:

![Flow diagram of the proposed smart sampling method.](image)

**Step 7:** The first additional point \( x_{SR} \) within the sparse region is acquired by minimizing the multidimensional density function \( D(x) \) defined by equation 3.9. \( x_{SR} \) is a solution of the following mathematical problem defined by:

\[
\begin{align*}
\text{minimize} \quad & D(x) \\
\text{subject to} \quad & C(x) > 0 \\
& l(i) \leq x_i \leq u(i) \quad 1 \leq i \leq p
\end{align*}
\]  

(3.19)

where \( l \) and \( u \) denote the minimum and maximum ranges of every parameter, and \( p \) denotes the parameter number. The second smart point is to add a sampling point \( x_{DISC} \) close to the discontinuity to reduce the uncertainty region that ranges between -1 to 1.
3.2 Metamodeling for Discontinuous Responses

$x_{DISC}$ is the solution of the following mathematical problem:

\[
\begin{align*}
\text{minimize} & \quad D(x) \\
\text{subject to} & \quad |C(x)| \leq \text{threshold} \\
& \quad l(i) \leq x_i \leq u(i) \quad 1 \leq i \leq p
\end{align*}
\]  
\tag{3.20}

The threshold value in Equation 3.20 is set to 0.2 in order to acquire values that lie in the range of -0.2 to 0.2. The discontinuity has a higher certainty to exist within this range. Equations 3.19 and 3.20 are addressed by multi-objective optimization problems (MOOP) due to the fact that they involve more than one objective function simultaneously (in this case two functions $D(x)$ and $C(x)$). MOOP has been applied in many fields of science, engineering, finance and economics where optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives [120]. Due to the interdependence of the objectives, MOOP do not have a single solution, instead, they have a set of alternative solutions. These solutions, which are called Pareto optimal set or non-dominated solutions [143], are optimal in the sense that there exist no other solution that could improve any of the objectives without worsening at least one of the other objectives. The Non-dominated Sorting Genetic Algorithm II (NSGA-II) [29] is used to solve Equations 3.19 and 3.20. NSGA-II is a population based genetic algorithm for solving MOO problems with constraint handling based on a non-dominated sorting procedure and elitism, which is the process of keeping high fitness individuals for preserving favorable genetic information. For more information about the NSGA-II algorithm, the reader is referred to [29].

**Step 8**: The iteration number is chosen to be the termination criterion. Once the maximum iteration number, which is set by the user, is reached, the algorithm terminates. Else, additional new sampling points are added, where the algorithm is returned to **Step 2**.

To illustrate the smart iterative algorithm, the one-dimensional piecewise analytical function denoted in Equation 3.21 is considered. The one-dimensional plot is shown in Figure 3.16.

\[
y(x) = \begin{cases} 
0.01 ((x^2 - 62)^2 + (x^2 - 2x - 1.5)^2) + 10 & x \geq -3 \\
DiscVal & x < -3
\end{cases}
\]  
\tag{3.21}

The algorithm starts by generating an RBFN metamodel, a radial basis linear classifier $C(x)$, and the density function $D(x)$ to an initial training data set that contains four sampling points. The initial training experimental design set is sampled according to a
Figure 3.12: 1D Plot of Equation 3.21.

four-levels full factorial design. The resulting initial metamodel, with the corresponding classification model, and density function are plotted in Figure 3.13. The new proposed sampling points that are acquired from the multi-objective algorithm are denoted by a green horizontal line $x_{DISC}$ to refine the position of the discontinuity. The blue horizontal line which $x_{SR}$ explores the uncertainty of the feasible region. The update procedure is repeated until the predefined terminating criterion (3 iterations) is met. The results in Figure 3.13 show that the quality of the metamodel improves in every iteration.

Initial Sampling Model
Smart sampling algorithm after 1 iteration

(a) $f_D(x)$
(b) $C(x)$
(c) $D(x)$

Smart sampling algorithm after 2 iterations

(a) $f_D(x)$
(b) $C(x)$
(c) $D(x)$

Smart sampling algorithm after 3 iterations

(a) $f_D(x)$
(b) $C(x)$
(c) $D(x)$

Figure 3.13: Smart iterative approach applied to Equation 3.21. The green and blue line show the location of the next additional sampling points $x_{SR}$ and $x_{DISC}$ respectively.
3.3 Metamodeling Validation

Validating the approximation models is a crucial step in metamodeling. Model validation techniques are used to estimate the quality of the metamodel in terms of the prediction accuracy. The accuracy is strongly related on the metamodeling type as well as the quality and number of the training data set from which it is generated. Validation methods for metamodeling assessment involve the use of additional data or are based on resampling strategies [31]. Validation is defined by how good is the metamodel in mimicking the expensive or the black-box function. In other words, it determines how big is the residual $e_i$, which is defined by the difference between the simulated response $y_i$ and the metamodeling response $f(x)$. A validation procedure with small $e_i$ corresponds to an accurate metamodel. The metamodel quality is not described by a single statistical measure, instead different measures are calculated to quantify the metamodel accuracy.

In this work, several model-independent measures, listed in [185], including coefficient of determination $R^2$, the mean squared error ($MSE$), the maximum absolute error ($MAE$), the relative mean absolute error ($RMAE$), and the relative mean squared error ($RMSE$), are discussed.

$R^2$ is as measure that illustrates how well a data set fits a statistical model. It explains how much variability (spread out of a data) of a factor (metamodel responses) can be caused or explained by its relationship to another factor (simulation model responses) [73]. $R^2$ is defined by

$$ R^2 = 1 - \frac{\sum_{i=1}^{n_V} (y_i - f(x_i))^2}{\sum_{i=1}^{n_V} (y_i - \bar{y}_i)^2}, \quad (3.22) $$

where $n_V$, $y_i$, $\bar{y}_i$, and $f(x_i)$ correspond to the number of the validation data set, the output response of the validation set, the mean of the validation set, and the metamodeling response of the parameter configuration $x_i$ respectively. $R^2$ ranges between 0 to 1, where 1 indicates a perfect fit, the closer the value of gets to 1, the more accurate the metamodel is.

The additional error measures that represent the deviation of the metamodel from the real simulation model are defined according to the following equations:

$$ MSE = \frac{1}{n_V} \sum_{i=1}^{n_V} (y_i - f(x_i))^2, \quad (3.23) $$

$$ MAE = \max |y_i - f(x_i)| \quad i = 1, \cdots, n_V, \quad (3.24) $$

$$ RMAE = 100\% \times \frac{1}{n_V} \sum_{i=1}^{n_V} \left| \frac{y_i - f(x_i)}{y_i} \right|, \quad (3.25) $$
\[ RMSE = 100\% \times \frac{1}{n_V} \sum_{i=1}^{n_V} \left( \frac{y_i - f(x_i)}{y_i} \right)^2. \] (3.26)

The smaller the value of \( MSE, MAE, RMAE, \) and \( RMSE \) is, the more accurate the metamodel is. Out of the four measures, the \( MAE \) gives a local accuracy while the others give a global accuracy. A well-known validation method is the cross validation (CV) technique that allows the validation of the model without any additional sampling points. It is based on the approach of splitting the original sampling point data into training and validating sets, which is often done either randomly or by a user defined split [180]. In \( p \)-fold CV, the training data with size \( n \), is split first into \( p \) different sets. Then, a metamodel is generated \( p \) times while leaving out the \( p \)-sets from the training set. Moreover, the left out \( p \)-sets are used to evaluate the error measures. An alternative method is leave-\( k \)-out CV, where \( \binom{n}{k} \) subsets of size \( k \) are first taken out of the training set and then a metamodel is generated to the remaining set [58]. The relevant error measures are evaluated each time at the omitted sets. This method requires more computational effort than the \( p \)-fold CV. The special case of leave-\( k \)-out is when \( k = 1 \) is called leave-one-out CV, which could be efficient for metamodel with few training data sets and at the same time very expensive for metamodels with large data sets [180]. To ensure the robustness of the cross-validation error estimate and avoid redundant samples, a uniformly spaced sampling structure, such as the latin hypercubes or Hammersley technique is recommended [73].

### 3.4 Numerical Examples

#### 3.4.1 Two-dimensional analytical function with a continuous response

In the first example, the iterative smart sampling approach presented in Section 3.1.2 is applied to the two-dimensional continuous function \( y(x) \) defined by Equation 3.27.

\[ y(x) = 0.01 \left( \left( x_1^2 + x_2^2 - 62 \right)^2 + \left( x_1^2 + 0.5 x_1 - 0.5 x_2 - 1.5 \right)^2 \right) + 10, \] (3.27)

where \(-7 < x_{1,2} < 7\). The two-dimensional contour plot is shown in Figure 3.14.

The goal of this example is to create a metamodel that mimics the analytical function \( y \) which is characterized by a continuous response. The algorithm starts by generating an RBFN metamodel \( f(x) \) and the density function \( D(x) \) of an initial data set that contains 9 sampling points. The initial training experimental design set is obtained by a three-levels full factorial design. The resulting initial metamodel with the corresponding density
function are plotted in Figure 3.15. In every iteration the new sampling point \( x_{SR} \) that is acquired from the multi-dimensional global minimization algorithm (the objective function is the density function) is denoted by a white star. The update procedure is repeated and this process is continued until the predefined terminating criterion (10 iterations) is met. The results in Figure 3.15 show that the contour shapes in the metamodel become much more similar to the ones in Figure 3.14 by simply adding 10 more sampling points iteratively in the sparse regions starting by the initial DOE. The method allows the user a flexibility in determining when to stop the generation process. Validating the metamodel in every iteration allows to determine the optimal number of the required sampling points. In this example, the validation process is done through visualizing the full two-dimensional domain space, and comparing the metamodel with the exact analytical model in Figure 3.14. After 10 iterations (19 runs), the metamodel is able to detect the position of the global minimum and maximum and without the need of performing additional runs.

Initial Sampling Model
3.4 Numerical Examples

Smart sampling algorithm after 1 iteration

![Smart sampling algorithm after 1 iteration](image)

(a) $f(x)$

(b) $D(x)$

Smart sampling algorithm after 5 iterations

![Smart sampling algorithm after 5 iterations](image)

(a) $f(x)$

(b) $D(x)$

Smart sampling algorithm after 10 iterations

![Smart sampling algorithm after 10 iterations](image)

(a) $f(x)$

(b) $D(x)$

Figure 3.15: Evolution of the two-dimensional contour plot of $f(x)$ and $D(x)$ after 1, 5, and 10 iterations. The white star represents the location of the new sampling points $x_{SR}$.

3.4.2 Two-dimensional analytical function with a piecewise response

In the second example, the iterative smart sampling approach presented in Section 3.2.2 is applied to the two-dimensional piecewise function $y(x)$ defined by Equation 3.28. The
two-dimensional contour plot is shown in Figure 3.16.

\[
y(x) = \begin{cases} 
  \text{Equation 3.27} & (x_1 - 7)^2 + (x_2 + 7)^2 \geq 80 \\
  \text{DiscVal} & (x_1 - 7)^2 + (x_2 + 7)^2 < 80 
\end{cases}, 
\]  

(3.28)

where \(-7 < x_{1,2} < 7\). The function is plotted in Figure 3.16.

Figure 3.16: Two-dimensional contour plot of the piecewise function in Equation 3.28. The white area illustrates the non-feasible domain space.

The goal of this example is to create a metamodel that mimics the piecewise analytical function \(y(x)\) which contains a discontinuity (\(\text{DiscVal}\) is set to 0). The algorithm starts by generating a radial basis function, a radial basis linear classifier \(C(x)\) constructed according to Equation 3.17, and the density function \(D(x)\) constructed according to Equation 3.12 to a training data set containing 9 sampling points (obtained by a 3-levels full factorial design) as shown in the plots below.

Initial Sampling Model

(a) \(f(x)\)  
(b) \(C(x)\)  
(c) \(D(x)\)
Smart sampling algorithm after 1 iteration

(a) $f(x)$

(b) $C(x)$

(c) $D(x)$

Smart sampling algorithm after 10 iterations

(a) $f(x)$

(b) $C(x)$

(c) $D(x)$

Smart sampling algorithm after 30 iterations

(a) $f(x)$

(b) $C(x)$

(c) $D(x)$

Figure 3.17: Evolution of the two-dimensional contour plots of $f(x)$, $C(x)$, and $D(x)$ after 1, 10, and 30 iterations. The white and green stars represent the location of the new sampling points $x_{SR}$ and $x_{DISC}$ respectively. In the classification model, the gray region between the red (feasible) and blue (non-feasible) represents the uncertainty region.

The resulting initial metamodel, with the corresponding classification model, and density function are plotted in Figure 3.17. The new proposed sampling points that are acquired from the optimization algorithm are denoted by a green star $x_{DISC}$ next to the discontinuity and a white star $x_{SR}$ in the sparse feasible region. The update procedure is repeated


and this process is continued until the predefined terminating criterion (30 iterations) is met.

To resolve the discontinuity between the feasible (blue to red color map) and non-feasible (white color) infinite points are required. As shown in Figure 3.17, the results show that the algorithm spread sampling point in every iteration to refine the discontinuity with a finite number of sampling points. The gray region which represents the uncertainty in the classification model decreases while adding more sampling points next to the discontinuity. The advantages of this iterative sampling method is minimizing the number of sampling points inside the non-feasible region, once the domain space is classified, no additional simulation runs are done inside the non-feasible space. Additionally, the contour shapes in the metamodel become similar to the ones in Figure 3.16 by simply adding more sampling points in the feasible region.

The algorithm keep adding sampling points sequentially until a stopping criteria is met. The user can specify a maximum number of sampling points. Once this number is reached, the algorithm terminates. Alternatively, the user can specify a required quality measure, presented in Section 3.3, and use it as a termination criterion. In this case, a validation step is required in every iteration step. The method keeps generating sampling points till the quality measure is reached.

3.5 Summary

In many manufacturing processes, the functional relationship between the input parameters and the output criteria is represented mathematically by a piecewise function. Due to digital changes in the solution properties of many engineering applications (e.g. topology changes like a material cut-through), the model response is described by a piecewise continuous function. Identifying that the region of the discontinuity is not explicit, since it is defined by many other parameter relationships or other physical interactions. When applying metamodels to responses with discontinuity, they can provide very poor fits because metamodels are generally applied to only continuous responses, as the reason is that they mostly apply fully-steady basis functions.

The major steps for generating a metamodel are the following: i) selection of the domain space, ii) sampling, iii) interpolation, and v) validation. Unlike the one-shot approach, where sampling is generated all at once, the adaptive approach is performed by first starting with an initial sampling points and second adding appropriate sampling points until a predefined criterion is met. In this chapter the one-shot and the adaptive approaches for generating metamodels for continuous as well as piecewise responses are given. Addi-
tionally validation measures and procedures are discussed for assessing the accuracy of a generated metamodel.

This chapter includes the following contributions:

- Equation 3.9 is solved by using the LU-decomposition direct solver of the GNU Scientific Library (GSL) which shows a good numerical stability. However when the number of data exceeds 10000, the required time for achieving the solution becomes excessive.

- A novel smart sampling algorithm for generating metamodels with piecewise responses is proposed, elaborated, and tested. The RBFN, with three different basis functions (Multiquadric, Gauss, and Linear spline) is used to construct the interpolation model, the density function, and the classification model respectively. The Multiquadric basis function shows less sensitivity to the basis width. The Gauss function is used to achieve a density response that decreases monotonically between the sampling points. New sampling points are generated in the sparse feasible regions and next to the discontinuity are required to improve the quality of the interpolation.

- The metamodeling procedures are applied to one and two dimensional analytic models. The examples indicate that the proposed algorithm provides better and more efficient results than conventionally used methods.

After generating and validating the metamodel, analysis takes place. This is important step in the production tasks to understand the effect of the parameters on the criteria. Due to the fact that the metamodel is fast and frugal, analysis becomes more convenient by interfacing different conventional engineering tasks like global sensitivity analysis, robustness analysis, and visualization techniques to the metamodel. An example how to perform analysis and exploration of multidimensional models is presented in the following chapter.
Chapter 4

Analysis and Exploration

In production industries, parameter identification, sensitivity analysis and multi-dimensional visualization are vital steps in the planning process for achieving optimal designs, improving decision making, and gaining valuable information and know-how of the process. They are extremely helpful in identifying the most-influential parameters, quantifying their contribution to the model output, reducing the model complexity, and enhancing the understanding of the model behavior. However, in order to perform a feasible sensitivity analysis and precise visualization for a multi-dimensional computer model, a large number of simulations is required to cover the complete parameter space. When the simulation models are numerically complex and the number of parameter inputs increases, the two procedures require a large number of simulations, which can be both very expensive and time consuming. This chapter contributes to the second Research Question RQ2, defined in Section 1.4, which mainly focuses on how to improve a basic understanding of a complex model by replacing the simulation model by a metamodel. This can be basically achieved by generating a fast interactive computer tool or web or mobile applications that allows simulation analysts to navigate complex models within a multi-dimensional design space. The main components of this tool are sensitivity analysis and visualization. This chapter introduces different global sensitivity analysis and visualization techniques that mainly allow exploring information from complex multi-dimensional computer simulation models using a fast and frugal metamodel. All of the presented methods will be applied to real laser manufacturing applications in Chapter 5.

4.1 Global Sensitivity Analysis

Sensitivity analysis (SA) is defined by how the uncertainty in the output of a model can be apportioned to different sources of uncertainty in the model input [147]. They allow
classifying the main contributing input parameters to an output (criterion) as the non-influential inputs, or ascertain some interaction effects within the model. The sensitivity analysis not only contribute to a model simplification and parameter screening but also act as a validation and verification step to the simulation models [148]. SA methods are generally either local or global methods. Local SA methods compute or approximate the partial derivatives of model outputs with respect to individual input parameter at some nominal settings, known as the nominal value point, in the domain parameter space [146]. They are defined by

\[
\frac{\partial y}{\partial x_{x_0}}(4.1)
\]

where \(x_i, i = 1, 2, \cdots, p\) are the input parameters of size \(p\) and the subscript \(x_0\) indicates the position where the derivative is taken. The most commonly used local SA methods are one-at-a-time (OAT) techniques and differential analysis (DA) [7]. However, global sensitivity analysis methods evaluate the effects of input variations on model outputs in the entire allowable ranges of the parameter space [173]. They evaluate the effect of one parameter while varying all other parameters, thus, efficiently exploring the full multi-dimensional input space [68]. A wide range of Global SA methods are available [146] from qualitative screening methods [116, 147] which identify (as opposed to quantify) the most important input parameter using a relatively inexpensive set of simulation experiments, to quantitative techniques which apportion the output variability to individual input variabilities [28, 104, 146, 167]. Among the quantitative SA methods, variance-based methods have received the most attention as they can accommodate non-linearity and interactions in a model and its parameters. The choice of the proper sensitivity analysis technique mainly depends on: i) the number of input parameters and the consideration of interactions among them; ii) the computational cost of running the model, iii) the characteristic of the output of interest (e.g. variance of the output), iv) the features of the model (e.g. linearity), and v) the scope of the analysis [147]. The focus of this section is the following important global sensitivity analysis measures.

### 4.1.1 Measures based on the Analysis of Linear Models

If a data sample \(T\) given by:

\[
T = \{(x^i_1, y^i)\}_{i=1}^{N} = \{(x^i_1, x^i_2, \cdots, x^i_p, y^i)\}_{i=1}^{N}, \tag{4.2}
\]

where \(N\) is the data size, and \(p\) is the number of inputs defined by vector \(x\) and one output defined by scalar \(y\) exist, then it is possible to fit a linear model explaining the behavior of \(y\) given the values of \(x\), provided that the sample size \(N\) is sufficiently large \((N > p)\) [146].
The two important sensitivity measures for linear models are: the pearson’s correlation coefficient and the standard regression coefficient. The pearson’s correlation coefficient is a quantitative estimate of linear correlation that can be determined by calculating a correlation coefficient on the parameter values of input and output. It is recommended to use correlation coefficients, derived from Monte Carlo simulations, as a reasonable way to rank model parameters according to their contribution to the outputs [104].

Pearson’s correlation coefficient $\rho(x_i, y)$ is defined by:

$$
\rho(x_i, y) = \frac{\sum_{j=1}^{n}(x_i - E(x))(y - E(y))}{\sqrt{\sum_{j=1}^{n}(x_i - E(x))^2(y - E(y))^2}}, \quad i = 1, \cdots, p, \quad |\rho| < 1 \quad (4.3)
$$

where $E(.)$ denotes the expected value. $\rho$ can be seen as a linearity measure between $x_i$ and $y$. It equals 1 or -1 if the tested input parameter has a linear relationship with the output. The larger the absolute value of $\rho$ is, the stronger the degree of linear relationship between the input and output values is. A negative value of $\rho$ indicates that the output decreases when the input parameter increases and vice versa. If $x_i$ and $y$ are independent, the correlation equals zero.

The standard regression coefficient $SRC(x_i)$ represents a share of variance $V(.)$ if the linearity hypothesis is confirmed and is defined by:

$$
SRC(x_i) = \beta_i \sqrt{\frac{V(x_i)}{V(y)}}, \quad i = 1, \cdots, p, \quad (4.4)
$$

where $\beta_i$, given by Equation 2.2, is the linear regression coefficient associated to $x_i$. In case of nonlinearity, these coefficients fail to represent the response sensitivities properly. Because of that, different global sensitivity analysis techniques (Elementary Effect and Variance Decomposition) have been developed.

### 4.1.2 Elementary Effect

The Elementary Effect method, also called Morris method, has been proposed as a screening method to identify a subset of inputs that have the greatest influence on the outputs [115]. It is a statistically motivated procedure for predicting the output of a computational model for unexecuted runs thereby reducing the number of runs [16]. The elementary Effect method provides two sensitivity measures with the aim of determining which input parameters could be considered to have effects which are (a) negligible, (b) linear and additive, or (c) nonlinear or involved in interactions with other parameters. It is a simple but effective way of screening a few important input parameters among the many that can be contained in a model, which is based on a replicated and randomized one-at-a-time (OAT) design. The main focus of this method is to identify and screen a
subset of input parameters that have the greatest influence on the outputs variations in a given hyper-cube of parameter ranges. The elementary effect $EE_i$ of a parameter $x_i$ under investigation is defined by:

$$EE_i = \frac{y(x_1, x_2, \cdots, x_{i-1}, x_i + \Delta, \cdots, x_p) - y(x_1, x_2, \cdots, x_i, \cdots, x_p)}{\Delta},$$

(4.5)

where $x = (x_1, x_2, \cdots, x_p)$ is input parameters vector of size $p$ and $\Delta$ is the sampling width of the parameter space. The value of $\Delta$ is a value in the range of $\{\frac{1}{k-1}, \cdots, 1 - \frac{1}{(k-1)}\}$ where $k \geq 2$ is the sampling level of the domain space. The domain space is normalized to a unit hypercube $(x_i \in [0, 1])$ [115]. For example if four sampling levels $(k=4)$ are considered on a one dimensional parameter space $x_1 \in [0, 1]$, the corresponding $\Delta$ is equal to either $\frac{1}{3}$ or $\frac{2}{3}$. Campolongo [16] suggests sampling the parameter space with $r$ different random trajectories $y(x_i)$ and computing the $EE_i$ of each trajectory individually [115]. A trajectory is defined by a sampling interval $[x_{i0}, x_{i1}]$ of the parameter $x_i$. The measures $\mu$ and $\sigma$, are estimated according to:

$$\mu_i = \frac{1}{r} \sum_{j=1}^{r} \left| EE_i^j \right|,$$

(4.6)

$$\sigma_i = \frac{1}{r-1} \sum_{j=1}^{r} (EE_i^j - \mu_i)^2,$$

(4.7)

where $r$ is the number of trajectories. $\mu$ assesses the overall influence of the input parameter on the output. If $\mu_i$ is large, the parameter $x_i$ has the biggest influence on the (scalar) output and $\sigma$ measures the nonlinearity or interaction effect. Additionally, if $\sigma_i$ is large, then the parameter $x_i$ has either a nonlinear effect on the output or an interaction effect with other inputs.

Figure 4.1: The estimated sensitivity measures: mean ($\mu$) and standard deviation ($\sigma$) for (a) $y(x_1) = a$, (b) $y(x_2) = ax_2$, and (c) $y(x_3) = ax_3^2$.

Figure 4.1 shows three different graphs from left to right with three different effects of the parameter $x$ on the criterion $y$ where $x$ has a negligible $y(x_1) = 0$, linear $y(x_2) = ax_2$, and nonlinear effect on $y(x_3) = ax_3^2$ respectively. Five trajectories, represented by the
green dashes, are considered for sampling. In Figure 4.1 (a) the EE value for the five trajectories equals to zero. Correspondingly, \( \mu_1 \) and \( \sigma_1 \) are equal to zero. However, in Figure 4.1 (b) the EE value of the trajectories sampled on the linear effect of parameter \( x_2 \) equals to \( a \). Thus, \( \mu_2 \) will be equal to the given slope \( (a) \) and \( \sigma_2 \) will be also equal to zero.

The EE screening measures can be represented graphically by screening plots. It provides two sensitivity measures: i) the x-axis as the mean value \( \mu \) and ii) the y-axis as the standard deviation \( \sigma \) which describes the higher order effects such as nonlinearity or interactions between inputs. The Morris screening plot for the three previous effects is shown in Figure 4.2. The effect of each parameter \( x_i \) is represented graphically on the screening plot by a point with the coordinate \( (\mu_i, \sigma_i) \).

![Figure 4.2: Morris Screening graph.](image)

### 4.1.3 Variance Decomposition Method - Sobol Method

The variance-based methods are used to study how the variance of the output is apportioned across different inputs as well as across the interactions between them [68, 103]. They are mainly used to: i) identify what is/are the most deserving parameter(s) that should be fixed to achieve the greatest reduction in the uncertainty of the output, ii) screen the input parameters by identifying parameters or sets of parameters that are non-influential, i.e. the parameters that can be fixed at any value over the corresponding range of uncertainty without significantly reducing the output variance, and iii) identify what is the minimal subset (interaction) of parameters that should be fixed to achieve a prescribed reduction in the uncertainty of the output, i.e. the reduction of the output variance to below a given tolerance [149]. Variance gives a measure of how the data distributes itself around the mean or expected value [167]. For an arbitrary function \( y(x) \),
the expected value \(E[y]\) and variance \(V(y)\) of this function is given by:

\[
E[y] = \int_{-\infty}^{\infty} yp(x)dx, \tag{4.8}
\]

\[
V(y) = E[(x - \mu)^2] = \int_{-\infty}^{\infty} [y - \mu]^2 p(x)dx, \tag{4.9}
\]

where \(p(x)\) denotes the probability density function PDF [147]. In this work, the uniform distribution is chosen to provide the effects of each individual variable across the whole domain of parameter space restricted by its lower and upper bounds. For a random variable \(x\) that is distributed uniformly in the interval \([a, b]\), \(p(x)\) is defined by

\[
p(x) = \begin{cases} 
\frac{1}{b-a} & x \in [a, b] \\
0 & x \notin [a, b].
\end{cases} \tag{4.10}
\]

The Sobol method [167] is a global variance Variance-based sensitivity analysis. It involves the computation of the Sobol indices which give a sensitivity measures that do not need a linear or an additive model behavior. The Sobol method is based on analysis of variance (ANOVA) decomposition [147]. For example, for a square-integrable function \(f(x_i)\) with parameter vector \(x_i = [x_1, \cdots, x_p]\) of size \(p\) where \(x_i \in [0, 1]\), the ANOVA decomposition is represented as follows [68, 167]:

\[
f(x) = \text{const} + \sum_i f_i(x_i) + \sum_{i<j} f_{ij}(x_i, x_j) + \cdots + f_{1,2,\ldots,p}. \tag{4.11}
\]

Sobol proved that if each term in the expansion above has zero mean, e.g. \(\int f_i(x_i)dx_i = 0\), then all terms of the decomposition are orthogonal in pairs, e.g. \(\int f_i(x_i)f_j(x_j)dx_i dx_j = 0\) [147]. As a consequence, the definitions of the terms of the functional decomposition in terms of conditional expected values are given according to:

\[
f_0 = \text{const} = E(y), \tag{4.12}
\]

\[
f_i = f_i(X_i) = E(y|x_i) - E(y), \tag{4.13}
\]

\[
f_{ij} = f_{ij}(x_i, x_j) = E(y|x_i, x_j) - f_i - f_j - E(y), \tag{4.14}
\]

where \(y\) is the output, \(f_i\) are referred to as main effects of \(x_i\), and the \(f_{ij}\) are two-way interactions between the pairs \((x_i, x_j)\), etc. By squaring and integrating Equation 4.11 we get

\[
\int f^2(x)dx - f_0^2 = \sum_{s=1}^{p} \sum_{1 \leq i_1 < \cdots < i_s} \int f_{i_1\cdots i_s}^2 dx_{i_1} \cdots dx_{i_s}. \tag{4.15}
\]

In Equation 4.15 the left hand side corresponds to the variance of \(y\) and the right hand side corresponds to variances decomposed according to parameters \(x_i\). This leads to the variance expression defined by

\[
V(y) = \sum_i V_i + \sum_{i<j} V_{ij} + \cdots + V_{1,2,\ldots,p}. \tag{4.16}
\]
where,

\[ V_i = V[f_i(x_i)] = V[E(y|x_i)], \]  
(4.17)

\[ V_{ij} = V[f_{ij}(x_i, x_j)] = V[E(y|x_i, x_j) - f_i(x_i) - f_j(x_j)], \]  
(4.18)

\[ = V[E(y|x_i, x_j)] - V_i - V_j. \]

After considering the variances, the Sobol indices can be calculated. The Sobol indices are sensitivity measures that describe the contribution of each input or group of inputs to the variance of the output. They include the main effects as well as the interaction effects for the inputs. The main or first order effect of \( x_i \) on \( y \), which is the expected amount of variance \( V_i = V[E(y|x_i)] \) that would be removed from the total unconditional variance, is defined according to Sobol [167] by

\[ S_i = \frac{V[E(y|x_i)]}{V(y)}, \quad S_i \in [0, 1], \]  
(4.19)

where \( S_i \) is a number that lies in the range of 0 to 1. When \( S_i \) is equal to 0, this implies that \( y \) and \( x_i \) are independent. When \( S_i \) is close to 1, this implies that \( y \) might depend only on \( x_i \) or an interaction of \( x_i \) with other parameter combinations.

The interactions effects occur when extreme values of the output \( y \) are uniquely associated with particular combinations of model inputs, in a way that is not described by the first-order effects \( S_i \). The second order sensitivity indices are defined by:

\[ S_{ij} = \frac{V_{ij}}{V(y)}, \quad S_{ij} \in [0, 1], \]  
(4.20)

where \( V_{ij} \) is the joint effect of \( x_i \) and \( x_j \) minus the first-order effects for the same parameters. Analogous formulas can be written for higher orders, enabling the analyst to quantify the higher-order interactions [147]. For example, the third order sensitivity indices (also called three-way interactions) are defined by:

\[ S_{ijl} = \frac{V_{ijl}}{V(y)}, \quad S_{ijl} \in [0, 1], \]  
(4.21)

where \( V_{ijl} \) is the joint effect of \( x_i, x_j, \) and \( x_l \) minus the second-order effects for the same parameters. It is defined by:

\[ V_{ijl} = V[f_{ijl}(x_i, x_j, x_l)] = V[E(y|x_i, x_j, x_l)] - V_{ij} - V_{il} - V_{jl}. \]  
(4.22)

In this work, the Sobol indices are presented by a form of a clique graph. A clique graph is a graph \( G = (C, L) \) with a finite set of circles \( C \) that are connected by a set of lines \( L \). The input parameters are represented by the graph’s circles \( C_i \), the edge thickness of the circle \( i \) reflects the main effect Sobol indices of the parameter \( x_i \). The lines \( L_i \) represents the interaction between two parameters that connect them. The line thickness reflects
the interactions effects of the Sobol indices of the circles it combines. For illustrating the calculation of the Sobol indices and demonstrating the clique graph, the three-dimensional Ishigami function which exhibits strong nonlinearity and non-monotonicity is considered:

\[ y(x) = \sin(x_1) + 7\sin^2(x_2) + 0.1x_3^4\sin(x_1), \quad x = \{x_1, x_2, x_3\}, \quad (4.23) \]

where \( x_{1,2,3} \in [-\pi, \pi] \). In order to estimate the Sobol indices \( S_i \) and \( S_{ij} \) numerically, a Monte Carlo simulation is executed by first considering \( y \) as a black box model and then generating 7000 sample points from a uniform distribution. A tolerance value equals to 0.1 is taken as an indicator to the effect. If the Sobol indices are less than the tolerance value, they are assigned a zero value. The resulting values are listed in Table 4.1, and the corresponding clique graph is shown in Figure 4.3.

Table 4.1: Sobol indeces of Ishigami function

<table>
<thead>
<tr>
<th></th>
<th>( S_1 )</th>
<th>( S_2 )</th>
<th>( S_3 )</th>
<th>( S_{12} )</th>
<th>( S_{13} )</th>
<th>( S_{23} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.34</td>
<td>0.43</td>
<td>0.01</td>
<td>0</td>
<td>0.189</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 4.3: Clique Graph for the Ishigami function.

From the clique graph, in Figure 4.3, it is found that parameter \( x_3 \) is the least sensitive parameter and \( x_2 \) is the most sensitive parameter as the circle thickness is greater than that of \( x_1 \) and \( x_3 \). The red line illustrates a high second order interaction effect between \( x_1 \) and \( x_3 \). This final result matches completely the sensitivity properties of Equation 4.3.
4.2 Visualization and Exploration

Visualization is important for analyzing and understanding multi-dimensional metamodels, since it makes use of human pattern recognition capabilities that do not apply to abstract representations like equations or tables. The state of the art of visualizing multi-dimensional data with conventional mathematical software is performed by projecting plots onto one, two, or three dimensional subspaces, graphs including scatter plots, surface responses, and color labels which correspond to the projected data within this subspace.

The analysis of multi-dimensional data has been a core area of visualization for the past twenty years. In 1990, Tufte [175] presented a book that provides a practical advice about how to explain complex material by visual means, with extraordinary examples to illustrate the fundamental principles of information displays. In the 1993, Van Wijk [193] introduced the HyperSlice visualization concept to facilitate the discovery of relationships between multiple input variables. In 2001, Meyer and Johnson [111] proposed a tool to visualize and explore a large amount of data from simulation as well as experiments. A recent survey on visualization techniques, which covers both multi-dimensional data, has provided recently by Kehrer and Hauser [79]. In this chapter, two tools called memoSlice, extended HyperSlice, and HDViz, a global steepest gradient representation, which allow exploring and visualizing predicted outputs of metamodels for multi-dimensional simulation models are introduced. Both tools provide in addition to the online exploration, additional tasks such as optimization, online exploration, and sensitivity analysis.

4.2.1 memoSlice

MemoSlice is an innovative visualization tool with a high level of user interactivity that is developed by Gebhardt [50] in the second phase of the Cluster of Excellence of RWTH Aachen "Integrative Production Technology for High-Wage Countries" in the research area "Virtual Production Intelligence (VPI)". MemoSlice is mainly based on a multi-view visualization approach for the exploratory analysis of multi-dimensional RBFN metamodels. MemoSlice enables users to improve the understanding of a metamodel and optimize the process by allowing fast and user-friendly navigation through the full multi-dimensional domain space. This tool has the following contributions: i) optimization by identification of improved parameter configurations, ii) sensitivity analysis and validation of the simulation model by embedding experimental evidence into the parameter space, iii) comparison and assessment of different technical designs, iv) prediction of the behavior under uncertainties, and v) provision of high degree of interactivity by means of specific parallelization techniques.
In order to describe the application of memoSlice, a three-dimensional analytical function is used as a test case for the visualization process.

\[
y(x) = \sum_{i=1}^{3} -10 \exp \left( -0.2 \sqrt{x_i^2 + x_{i+1}^2} \right), \quad -5 \leq x_i \leq 5, \quad 1 \leq i \leq 3 \quad (4.24)
\]

First, a metamodel is constructed by generating 1000 training points from the Hammersley sequence design presented in Section 2.2.3. Second, the RBFN with multiquadric basis functions (defined in Section 3.2.1) is used to map the discrete function values at the sampling points to a continuous metamodel function \( f(x) \) as an approximation of the test function \( y(x) \) defined in Equation 4.24.

As shown in Figure 4.4, memoSlice is composed of the following three basic components:

**Scatterplot matrix view**

Scatterplots are plots that reveal relationships between each pair of variables. They show how much one variable is affected by another [21]. The relationship between two variables is called correlation. The properties of correlations, which are the strength
and direction, are determined by the pattern of points of the scatterplot of the two variables. In memoSlice, the scatterplot matrix, illustrated in Figure 4.5, contains one scatterplot for every possible pair of the parameter-criterion combinations which gives an overview of the distribution of values. The scatterplot data is created by performing a Monte Carlo simulation using the RBFN metamodel on-the-fly. The parameter samples are generated based on user-defined rules (sampling either from a uniform distribution or normal distribution). For a flexible analysis, memoSlice allows the user to define a specific region of interest inside the parameter space by locking not only the parameters ranges but also the criterion to a fixed value or to a defined range with specific minimum and maximum values. MemoSlice offers the possibility of exporting the scatterplot data to a file which can be imported to different data mining or statistical software or web platforms.

![Figure 4.5: A modified screenshot of the Scatterplot view. The lower row represents the scatterplots for the three pairs of the parameters with $f(x)$ and the upper row represents the scatterplots for the three pairs of the parameters with $f(x)$ density function $D(x)$.](image)

**HyperSlice view**

The HyperSlice view is created for visualizing the metamodel (multi-dimensional parameter and criteria) within the parameter space. HyperSlice is a matrix of axis-aligned one-dimensional (diagonal) and two-dimensional (upper left and lower right) slices of the parameter space. The one-dimensional slice is described by an x-axis which corresponds to the parameter labeled on the bottom of the matrix and a y-axis which corresponds to the selected metamodel criterion (upper or lower). The two-dimensional slices are surface plots which are graphical representations of the relationships among three numerical variables in two dimensions. The x-axis and y-axis correspond to two parameters labeled on the bottom of the matrix and on the left of the matrix respectively and a third variable,
which corresponds to the selected criterion, for determining the surface color. The user has the possibility to select which criterion to visualize, and configure its corresponding color map via the extended pie menus described in [51]. The RBFN metamodel $f(x)$ is illustrated in Figure 4.6 by the blue-to-red color map where blue and red corresponds to the global minimum and the global maximum respectively.

![Figure 4.6: A modified screenshot of the HyperSlice view (with respect to the text labeling compared with the ones in Figure 4.4). The white sphere illustrates the location of the seed point that is selected by the user. The diagonal represents the one-dimensional relationship of each parameter on $f(x)$ at the seed point. The upper left represents the relationships between the parameters and $f(x)$ in two dimensions. The lower right represents the relationships between the parameters and density function $D(x)$ which reflects the position of the training points and consider as a measure of the metamodel quality of interpolation.]

**3-D view**

In order to visualize an additional dimension, memoSlice offers a 3D view, including direct volume rendering. The 3-D plot is a graphical representation of the relationship among four numerical variables in three dimensions. The x-axis, y-axis, and z-axis correspond to the first three variables that represent the parameters which are freely selected by the
user, and a fourth variable, which corresponds to the selected criterion, determines the rendered color map. A snapshot of the 3D plot for $f(x)$ and the density function $D(x)$ is illustrated in Figure 4.7

![Figure 4.7](image)

Figure 4.7: A modified screenshot of the 3-D view applied to the three-dimensional (a) RBFN metamodel of $f(x)$ and (b) Density function $D(x)$ with their corresponding color legends. The seed point is represented by the intersection of the white planes shown in the two plots.

The light gray spheres in the upper and lower HyperSlice matrices of Figure 4.6 and Figure 4.7 illustrate the locations of training points of the metamodel. For analysts, the location of the training points is of huge interest when visualizing a metamodel since it reflects the exact parameter settings where simulations are executed as well as it provides an assessment of the behavior of the metamodel between training points (uncertainty). The size of these light gray spheres is scaled according to the distance of the training point from the selected view. The point radius decreases when the distance from the view increases until it vanishes when a threshold maximum distance is reached [50] as the distance is negative. The radius of any sampling point is defined by:

$$r = r_{max} \left(1 - d_\perp \sqrt{n}\right),$$

(4.25)

where $r$ is the final radius, $r_{max}$ is the maximum point radius, $d_\perp$ is the perpendicular distance of the point to the plane generated by the seed point, $p$ is the number of dimensions of the parameter space and $n$ is the total number of training points.

In addition to the light gray spheres, there exist in Figure 4.6 a large white sphere and a cross in Figure 4.7 which illustrate the current parameters configuration. This point, which is called seed point, reflects the current settings of the metamodel to be displayed for the single parameter configurations. This means that for the three-dimensional RBFN metamodel $f(x)$, every one-dimensional graph is varied according to its corresponding parameter range while keeping the two other parameters fixed at the seed point. This
is similar to the two-dimensional graphs, where the two parameters are varied (in every corresponding slot) and one parameter is fixed at the seed point. When the seed point is changed, the affected plots are triggered which execute new metamodel evaluations. A snapshot of the training points and the seed point of the RBFN metamodel is illustrated in Figure 4.8 (a).

Figure 4.8: Modified screenshot of memoSlice applied to the three-dimensional RBFN metamodel of \( f(x) \). (a) The training points are represented by the small gray spheres and the seed point is represented by the big white sphere. Local optimization of the metamodel via the gradient trajectory from a local minimum (b) to a local maximum (c).

MemoSlice provides an additional measure to optimize the process in the vicinity of the seed point. The measure is the gradient trajectory which is of key importance for the understanding of scalar functions and its local sensitivity. Starting from the seed point, the gradient trajectory is extended between the next local minima and maxima labeled by the blue and red color respectively. The gradient of RBFN is computed analytically according to

\[
\nabla f(\bar{x}) = \sum_{m \in M} \bar{e}_m \sum_{i \in N} 2 w_i (x_m - x_{i,m}) h' \left( \|x_m - x_{i,m}\|^2 \right) \tag{4.26}
\]

where \( M \) is the number of parameter dimensions, \( N \) is the number of training points, \( \bar{e}_m \) is the orthogonal unit basis vector, and \( h'(x) = \frac{dh}{dx} \). In order to generate the gradient trajectory in the vicinity of the training point, the steepest descent and ascent are constructed by:

\[
\bar{x}_{k+1} = \bar{x}_k \mp \gamma_k \nabla f(\bar{x}_k), \tag{4.27}
\]

with \( \nabla f \) being the gradient described in Equation 4.26, and \( \gamma \) is the step size. The gradient trajectory of the metamodel is illustrated by the colored trajectory in Figure 4.8.

An additional feature of memoSlice is visualizing the non-feasible region of the metamodel that includes a discontinuity (feasible/non-feasible). Non feasible areas, which represent
working limits or physical limits, are grayed out in all visualizations. The second meta-
model (classification metamodel $C(x)$), described in Chapter 3, is loaded in memoSlice
for predicting a feasible as well as a non-feasible region. MemoSlice allows the user a func-
tionality to not only choose the color of the discontinuity but also blending options of the
color that vary between -1 and 1. In the application chapter, visualizing the non-feasible
regions will be emphasized by different applications of laser manufacturing processes. For
further details on memoSlice, the reader is referred to [50].

### 4.2.2 High-Dimensional Visualization HDViz

In order to understand the behavior of a model based on a sample data, an alternative
visualization approach which combines topological and geometric techniques is proposed
by Gerber [53]. Gerber provided an open source package called HDViz [55] to generate
a discrete Morse-Smale (MS) complex approximation based on the k-Nearest Neighbor
(kNN) graph for a multi-dimensional function that is characterized by input parameters
$x \in \mathbb{R}^p$ and output criterion $y \in \mathbb{R}$. Unlike memoSlice which need an RBFN interpolation
model, the Morse Smale complex analyzes and visualizes the model based on a set of
discrete samples. The MS complex [117] provides a decomposition of the parameter space
into piecewise monotonic parameter domain called crystals. Within each crystal, a regres-
sion curve of the system parameters with respect to the output connecting the minimum
and maximum is generated. Moreover, additional information such as local and global
shape, width, length, and sampling densities are provided in every crystal. The result is
a simplified geometric representation of the MS complex in the multi-dimensional input
domain [52]. Finally, the geometric representation is embedded in 2D in order to preserve
the important information about the high dimensional scalar fields, using dimension re-
duction techniques such as principal component analysis [75], to provide a visualization
platform. HDViz has following contributions: i) understand the extreme output values
and the geometry of the regions connecting them, and ii) define an inverse relationship
that indicates which parameter configuration corresponds to a specified output.

HDViz is based on the Morse theory [113] which is defined as the following: let $\mathcal{M}$ be
a smooth compact $p$ dimensional manifold and $f : \mathcal{M} \rightarrow \mathbb{R}$ a smooth function. If the
gradient of $f$ is zero at some point $x \in \mathcal{M}$, then $f$ has a critical point at $x$. The Hessian
matrix plays an important role in Morse theory, because it allows the classification of the
critical points. If the determinant of the Hessian matrix $\mathcal{M}$ of $f$ is zero then $x$ is called a
non-Morse critical point of $f$ (for example saddle points). Otherwise it is called a Morse
critical point of \( f \). From any point \( s \), the integral line \( \lambda \), which is defined by:

\[
\frac{d\lambda}{ds}(s) = \nabla f(\lambda(s)),
\]

is constructed between the maximum and minimum critical points. An ascending or descending manifold of a critical point is defined by the points whose integral line start and end at the critical points. The descending and ascending manifolds determine the MS complex of \( f \) (increase) and \( -f \) (decrease) respectively [53].

The MS complex introduces a parameter called persistence that gives a measure of the importance of each critical point. It is defined as the measure of the amount of change in the function \( f \) required to remove a critical point which correspondingly merges partitions. Persistence is defined mathematically as

\[
p(x_i) = \| f(x_i) - f(n(x_i)) \|,
\]

where \( x_i \) is a critical point, \( n(x_i) = \text{argmin}_{x_j \in s(x_i)} \| f(x_i) - f(x_j) \| \) and \( s(x_i) \) is the set of critical points that have a direct integral line connecting to \( x_i \) [55]. A sequential Morse-Smale complex filtering or simplification is done when the critical point with minimal persistence is removed until a single partition is reached [37]. At the highest persistence level, the MS complex only consists of the highest maximum and lowest minimum, and the segmentation corresponds to the entire domain. With decreasing persistence levels, more extremal points and corresponding partitions are introduced based on their persistence level. Thus, persistence introduces a notion of scaling at which the MS complex of \( f \) is considered [54].

Figure 4.9 illustrates the concept of persistence on a one-dimensional test function.

Figure 4.9: Sketch of the concept of persistence in the MS complex applied to a one-dimensional test function \( y = f(x) \). (a) The MS complex is constructed by the finest persistence level \( p_2 \) resulting partitions \((A,B), (B,C), (C,D), (D,E)\). (b) The change of the Morse-Smale complex with a \( p_1 \) persistence which required to remove the critical points C and D resulting partitions \((A,F), (F,E)\) is illustrated.

Since there is no modification in this work to the HDViz tool, the steps of generating the MS Complex in HDViz are listed in Appendix A. For understanding how the MS approximation is visualized, the two-dimensional four peak analytical function defined
by:

\[
y(x) = \frac{1}{2} \left( e^{-\frac{(x_2-0.25)^2}{0.3^2}} + e^{-\frac{(x_2-0.25)^2}{0.3^2}} + e^{-\frac{(x_1-0.75)^2}{0.3^2}} + e^{-\frac{(x_1-0.75)^2}{0.1^2}} \right)
\]

(4.29)

is used as an example for visualization. 2000 sampling points are uniformly sampled on \(x_1, x_2 \in [0, 1]\) and loaded in HDViz, the A screenshot of HDViz visualization is shown in Figure 4.10.

Figure 4.10: Visualization of the approximate Morse-Smale complex of Equation 4.29 from 2000 sampling points. The analytical function is plotted on the bottom left corner. The additional information inside the box of the location slider gives an insight to the decomposed regions by providing the value of the function, the standard deviation, the sampling density, and the function relationship of the parameters on the output.

As shown in Figure 4.10, the function has four maxima, nine minima, and 16 crystals determined by the regression curves. In each portion, the function has one local minimum and maximum located on the boundary. The visualization in Figure 4.10 describes, in addition to the topological representation, additional information boxes in each regression curve on a local position marked by the location slider. The additional information is the value of the function and its corresponding parameter configuration (from the inverse problem) with the corresponding standard deviation and sampling density. In order to explain the concept of persistence, three different two-dimensional analytical functions
with different scalar field complexities are considered as examples. All three functions, which are used as test functions in the optimization algorithms, are listed with their parameter space in Table 1. 2000 sampling points were uniformly sampled from every use case on the parameter domain and two simplifications (coarsest persistence level and finest persistence level) of the MS complex are performed. The visualization of the topology of the approximation of the MS complex with the coarsest level for the three functions is shown in Figure 4.11b. Alternatively, the visualization of the topology of the approximation of the MS complex with the finest level is shown in Figure 4.11c.

Table 4.2: Three two-dimensional cases used as test functions in the optimization algorithms.

<table>
<thead>
<tr>
<th>Use Cases</th>
<th>Ranges</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-StyblinskiTang</td>
<td>$-5 \leq x_1, x_2 \leq 5$</td>
<td>$y(x) = 0.5 \sum_{i=1}^{2} x_i^4 - 16x_i^2 + 5x_i$</td>
</tr>
<tr>
<td>2-Rosenbrock</td>
<td>$-1.5 \leq x_1 \leq 1.5, -1 \leq x_2 \leq 3$</td>
<td>$y(x) = \sum_{i=1}^{2} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$</td>
</tr>
<tr>
<td>3-Ackley</td>
<td>$-5 \leq x_1, x_2 \leq 5$</td>
<td>$y(x) = -20 \exp(-0.2 \sqrt{\frac{1}{2} \sum_{i=1}^{2} x_i^2})$ - $\exp(\frac{1}{2} \sum_{i=1}^{2} \cos(2\pi x_i)) + 20 + e$</td>
</tr>
</tbody>
</table>

StyblinskiTang function

![StyblinskiTang function](image)
Rosenbrock function

![Rosenbrock function graph]

Ackley function

![Ackley function graph]

Figure 4.11: Visualization of the approximate MS complex of the test functions listed in Table 4.2 with different persistence levels using 5000 sampling points. (a) Two-dimensional plot of the function (b) the visualization of MS approximation at the persistence level (c) the visualization of the MS approximation at the finest persistence level.

The results show that the crystals of the simplified MS-complex will not be monotonic with respect to the original function, but they will be monotonic on filtered versions of $f$. At the coarsest level, only a single maximum and minimum remain and the method is equivalent to a multivariate regression analysis. At finer levels, more detailed information about the topology is represented.

4.3 Summary

The possibility to analyze, visualize, and explore multi-dimensional models leads to create an online analysis applications that allows users "on the spot" to i) analyze and validate complex simulation models, ii) explore uncertainties, and iii) effectively identify the important model parameters. In the following chapter, the presented sensitivity analysis and visualization techniques are applied to real manufacturing processes.
In order to bring the virtual metamodel to reality, it is crucial to interface it to conventional engineering applications. The three typical applications of a metamodel are: the optimization techniques where the metamodel is used as a predictor, the global sensitivity analysis where the metamodel is used as an approximation of the model response in order to derive the sensitivity measures, and the visualization techniques where the metamodel is used as an emulator.

In this chapter, two global sensitivity analysis methods (the Elementary Effect EE and Variance Decomposition Method) are introduced. The screening EE method offers the possibility to identify a subset of parameters that have the greatest influence on the output criteria. The Variance Decomposition method analyze how the variance of a criterion is apportioned across different parameters as well as across the interactions between them.

To the author’s knowledge, the methods and algorithms for global sensitivity analysis are directly applied to continuous functions. However, as explained in Chapter 3, metamodels are indeed discontinuous due to existing physical limits. Therefore, two metamodels need to be created: One for predicting the response and one for checking the feasibility.

This chapter includes the following contributions:

- The Elementary Effect and the Variance Decomposition methods were implemented as tools in MATLAB. The tools are based on a random sampling approaches, which take the metamodels as inputs and calculate sensitivity measures as an output. During the analysis, the data are classified to feasible and non feasible, where the sensitivity estimations are applied to the feasible space only.

- The memoSlice visualization tool is currently under daily development. The inputs of the tool are metamodels with an ASCII files format generated by the author. The author main contribution is focusing on the conceptual development and the live demonstration to make the tool operative in real industrial applications.

- The concept of decomposing the multi-dimensional domain space by the Morse-Smale complex with different persistence level is presented. Different analytical examples are interfaced to the open source library HDViz. The results show that Morse-Smale complex is a promising method for visualizing and exploring the domain space.
Chapter 5

Applications of Metamodelling Techniques to Laser Manufacturing Processes

5.1 Introduction

In this chapter, several case studies of metamodelling applied to laser manufacturing are presented. These studies illustrate why and how a metamodel is used. The main goal here is to highlight the importance of using the proper metamodelling technique in order to generate a specific metamodel for every application with different objectives. The emphasis of the cases is how to easily implement a problem formulation that allows visualization, optimization, and sensitivity analysis. Here, all of the case studies are based on the RBFN metamodelling procedure introduced in Chapter 3. The five different manufacturing processes are: i) laser epoxy cutting (two-dimensional parameter space); ii) glass cutting (two-dimensional parameter space); iii) oxygen cutting (seven-dimensional parameter space), iv) laser drilling (five-dimensional parameter space); and v) laser sheet metal cutting (five-dimensional parameter space).

5.2 Laser Epoxy Cutting

In this test case, the superior performance of the proposed smart sampling algorithm for responses with discontinuity, presented in Section 3.2.2, is demonstrated. One of the challenges in cutting glass fiber reinforced plastics using a pulsed laser beam is to estimate the achievable cutting quality. An important factor for the process improvement is first
to detect the cutting limits and then to minimize the damage thickness of the epoxy-glass material. A simulation tool called 'EpoxyCut', which is a tool developed by the Chair Nonlinear Dynamics of Laser Processing (NLD) of RWTH Aachen University, is a reduced model that calculates the upper and lower cutting width, in addition to other criteria like melting threshold, time required to cut through, and damage thickness [160]. For details on the mathematical analysis, the reader is referred to [161]. The goal of this test case is to: i) minimize the lower cutting width while minimizing the laser power; and ii) efficiently generate an accurate metamodel. The laser beam is modeled as a Gaussian beam. The material thickness, the focal position, the beam radius, and the Rayleigh length are kept constant at 1 mm, 0 mm, 70 µm, and 1.28 mm respectively. The parameter space of the metamodel is spanned by the process parameters pulse duration and laser power. The criterion is the cutting width at the bottom of the material $W_B$.

The two-dimensional design space is sampled according to a fifty-level full factorial design. The process parameter design space is listed in Table 5.1.

<table>
<thead>
<tr>
<th>Process Parameters</th>
<th>Minimum</th>
<th>Maximum</th>
<th>Sampling Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pulse Duration $t_P [\mu s]$</td>
<td>10</td>
<td>1000</td>
<td>50</td>
</tr>
<tr>
<td>Laser Power $P_L [W]$</td>
<td>10</td>
<td>5000</td>
<td>50</td>
</tr>
</tbody>
</table>

With the help of the first-order polynomial regression metamodel, a process map on a 50 × 50 grid (2500 simulations to capture the cutting limits) is generated as shown in Figure 5.1 (a), illustrating the EpoxyCut model, and Figure 5.1 (b), illustrating the RBFN metamodel with only 69 simulations. The white color, shown in Figure 5.1, illustrates the discontinuity (non-feasible region which is the no-cut region) in the process domain. The physical interpretation is plausible. If the laser power or the pulse duration is not large enough to melt a specific material volume, cutting through does not occur. The relationship between the laser intensity and the cutting width bottom is discontinuous. If the intensity exceeds the energy material threshold, ablation occurs, and the cut depth increases continuously and finally, the response between the laser intensity and the cutting width becomes continuous. For generating an accurate metamodel for the cut width $W_B$ at the bottom of the material using the minimal number of simulation runs, the proposed smart sampling algorithm, introduced in Section 3.2.2, is applied with a maximum of 30 iterations starting with a three-levels full factorial design (9 sampling points).
Figure 5.1: Parameter space analysis of the cutting width at the bottom on laser power and pulse duration. The plots, generated using 2500 sampling points represent the process map of (a) the numerical model EpoxyCut and (b) the RBFN Metamodel using only 69 sampling points. The black points in the metamodel represent the sampling points generated by the smart sampling algorithm. The white region illustrates the no-cut region.

The evolution of the two-dimensional contour plot of the metamodel after 1 iteration, 10 iterations, and 20 iterations is illustrated in Figure 5.2.

Figure 5.2: Evolution of the two-dimensional contour plot of the metamodel after (a) 1 iteration, (b) 10 iterations, and (c) 20 iterations. The green and white stars represent the location of the new sampling points on the discontinuity and in the sparse region of the feasible domain respectively.

In each iteration step, the relative mean squared error $RMSE$ defined by Equation 3.26, and the coefficient of determination $R^2$ defined by Equation 3.22, of the cutting width are calculated and plotted in Figure 5.3. The results show that the quality of the metamodel improves (decrease of $RMSE$ and increase of $R^2$) when more training points are added till a convergence is reached. The advantage of the iterative technique over the one-shot classical approach is the control of the number of model evaluation runs and the location of the model evaluation. In Figure 5.3, the blue line represents the validation
measures of a one-shot metamodel generated from ten-levels full factorial design. Figure 5.3 shows that the same quality of a 100 sampling points from full factorial design can be achieved by 69 sampling points generated by the proposed smart sampling algorithm. The algorithm is controlled by the user. For example if a maximum iteration number or a required quality is reached, the algorithm is terminated, and there is no need for further simulation evaluations.

The main advantage of using a metamodel in this process over the full scale simulation is the minimization of the time required to generate the process map. The full data set is 2500 samples in total. It takes EpoxyCut around 5 seconds to estimate the cutting width of one parameter set. Thus, the total computation time required for generating such a process map will be around 3.5 hours. However, by using the fast metamodel, it takes the metamodel (generated after 30 iterations with 69 sampling points) about 0.003 seconds to evaluate one run. Thus, for the same grid, the process map requires around 7.5 seconds only.

![Graphs showing RMSE and R² over iterations](image)

Figure 5.3: The statistical measures (a) RMSE and (b) R² are generated to validate the quality of the metamodel. The blue line represents the validation of a one shot metamodel generated from ten-levels full factorial design. The plots show that the same quality of a 100 sampling points from full factorial design can be achieved by 60 sampling points generated by the proposed smart sampling algorithm.

### 5.3 Glass Cutting

Manufacturers of smart-phones and tablet computers are searching daily for faster and crack free production technologies to cope with the fast growing market demands. One of the core challenges for the display manufacturers is to use thinner and lighter glass sheets that can provide the same damage resistance at less weight. Due to the fact that the mechanical properties of the glass sheet become more delicate, processing and machining
of such sheets become very complex especially when using the traditional mechanical manufacturing methods. Currently, around 30% to 40% of waste is produced because of non-automated processes [142]. Therefore, it is absolutely necessary to establish more reliable and green manufacturing processes. One of the promising solutions offered by the laser technology for cutting thin glass is laser ablation with ultra-short pulses. It shows the potential to improve the cutting quality and reduces the number of processing steps [200]. There are, however, gaps in understanding the dynamics of the process, especially with regard to issues related to cracking and ablation (stops at finite cut depth) [40]. Since every kind of glass or dielectrics is responding differently, choosing the right process parameters to different types of glass, type of optics, and the thickness of the glass material becomes a challenging task that requires experience and understanding.

In order to obtain a better know-how of the cutting of transparent dielectrics based on ultra-short-pulsed laser ablation, a reasonable numerical model is necessary to investigate the laser ablation mechanism in glass cutting with ultrashort laser pulses. GlassCut [40], which is a simulation tool developed by the Chair Nonlinear Dynamics of Laser Processing (NLD) of RWTH Aachen University, is a reduced model that describes laser ablation and laser damage in glass. It is based on beam propagation and nonlinear absorption of photons in dielectrics. Unlike metals, the photon energy in wide-band-gap dielectrics such as glass and water is insufficient to excite electrons from the valence band to the conduction band by linear absorption. When the dielectrics is irradiated by an ultrashort laser pulse, the electrons in the valence band can be excited to the conduction band by multi photon ionization (MPI) (so called free-electrons) and then are heated to higher energy levels by the inverse Bremsstrahlung absorption. Recombination and diffusion processes are the losses terms for electron density. The laser beam generates a free-electron density $\rho$ at a given spatial position in the glass. $\rho$ is calculated using a rate equation in a generic form [171]:

$$\frac{\partial \rho}{\partial t} = \sigma I^k + \alpha_c I \rho - \eta_{rec} \rho^2 - \eta_{diff} \rho,$$

where the terms on the right-hand side are the MPI rate, cascade ionization rate, recombination rate and diffusion rate, respectively, and $\sigma$, $\alpha_c$, $\eta_{rec}$ and $\eta_{diff}$ are their corresponding coefficients. For further details on the mathematical physical analysis, the reader is referred to [170].

In this example, the focus is on one of the sub-models of GlassCut which is the electron dynamics that is described by Equation 5.1 to determine the maximum free-electron density (criterion) which is a function of fluence $F$. It can be calculated from the pulse shape leading to a parametric dependence on peak intensity $I_0$ and pulse duration $t_p$. The process parameters with their corresponding ranges are listed in Table 5.2.
### Table 5.2: Process Parameter design space of electron density model

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Units</th>
<th>Symbol</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluence/Intensity</td>
<td>$[J/cm^2]/[W/m^2]$</td>
<td>$F/I$</td>
<td>$10^{-4}/10^{10}$</td>
<td>$10^{3}/10^{20}$</td>
</tr>
<tr>
<td>Pulse duration</td>
<td>$[s]$</td>
<td>$t_p$</td>
<td>$10^{-13}$</td>
<td>$10^{-8}$</td>
</tr>
</tbody>
</table>

The laser pulse shape at any point in the material is a Gaussian profile is defined by:

$$I(t) = I_0 \cdot \exp \left( a \left( \frac{t}{t_p} \right)^2 \right), \quad a = 4 \log 2,$$

where $t_p$ is the pulse duration and the peak intensity is given by $I_0$. The fluence $F$ is defined by integrating the intensity with respect to time. For a Gaussian profile, the fluence is defined by:

$$F = I_0 \cdot \sqrt{\frac{\pi}{4 \log 2} t_p}.$$

and the maximum free-electron density is defined by:

$$\rho_{\text{max}} = \max (\rho(I, t_p, t)).$$

In order to visualize the effect of these parameters characterizing the laser pulse on the maximum free-electron density, the metamodeling concept is applied. A metamodel is first generated by a fifty-levels full factorial sampling design which requires 2500 simulation runs. Second, a first order polynomial regression model is used to map the discrete sampling points to a continuous function. The results are plotted in form of two process maps shown in Figure 5.4.

In order to investigate the influence of the number of simulations on the process map, an additional metamodel is constructed. The metamodel is constructed by first generating 50 training points from an optimized latin hypercube design algorithm (minmax) [197] (46 points) augmented with a two-levels full factorial sampling design (four corner points). The simulation GlassCut is considered as a black box model that has two parameters which are the pulse duration and the peak intensity, and one criterion which is the free electron density. The training set is illustrated by the gray circular spheres shown in Figure 5.6. Additionally an RBFN with multiquadric basis functions, introduced in Section 3.1.1, is used to map the discrete sampling points to a continuous function $f_{\rho_{\text{max}}}(x)$. A schematic for the architecture of the RBFN for this process is shown in Figure 5.5.
Figure 5.4: Process map of maximum free-electron density $\rho_{\text{max}}$ on (a) pulse duration $t_p$ and fluence $F$ and (b) pulse duration $t_p$ and intensity $I$.

Figure 5.5: Architecture of the Radial Basis Function Network (RBFN) for the electron dynamics in glass cutting.

In order to assess and estimate the metamodel quality, 2500 points from the full factorial model were used as the validation data set to calculate the relative absolute error $RMAE$ defined by:

$$RMAE = 100\% \cdot \left( \frac{1}{n_V} \sum_{i=1}^{n_V} \left| \frac{\rho_{\text{max}_i} - f_{\rho_{\text{max}_i}}}{\rho_{\text{max}_i}} \right| \right), \quad n_V = 2500,$$

and the coefficient of determination $R^2$, defined by:

$$R^2 = 1 - \frac{\sum_{i=1}^{n_V} \rho_{\text{max}_i} - f_{\rho_{\text{max}_i}}}{\sum_{i=1}^{n_V} \rho_{\text{max}_i} - \rho_{\text{max}_i}}, \quad n_V = 2500.$$

The results of $RMAE$ and $R^2$ are listed in Table 5.3. The main advantage of the metamodeling techniques is generating a fast and cheap metamodel that resembles the simulation model according to a controlled accuracy. In this test case, the technique is able to create
a metamodel that looks like the simulation model by only running 50 simulations instead of 2500.

Table 5.3: Validation measures of electron density metamodel

<table>
<thead>
<tr>
<th>Metamodel</th>
<th>$R^2$</th>
<th>$RMAE$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_{\rho_{\text{max}}}(F, tp)$</td>
<td>0.999457</td>
<td>0.248%</td>
</tr>
<tr>
<td>$f_{\rho_{\text{max}}}(I, tp)$</td>
<td>0.99862</td>
<td>0.768%</td>
</tr>
</tbody>
</table>

Figure 5.6: Process map of the interpolated maximum free-electron density $f_{\rho_{\text{max}}}$ on pulse duration $tp$ and fluence $F$ (a) and intensity $I$ (b). The gray points represent the training points.

From the validation study, the maximum local $RAE$, shown in Figure 5.7, is found to be less than 5% and a global average of less than 1% as shown in Table 5.3. Also, the advantage of generating a metamodel helps to detect, visualize, and explore the released high-density free-electrons. Once defined, the material near the ablated-crater wall can be modified according to this energy. For example, building up free electrons is necessary in order to initialize the laser absorption and consequently the ablation of dielectrics. Free-electron density is increased until reaching a threshold criterion $\rho_{\text{crit}}$ (critical electron density) where the breakdown of dielectrics occur (ablation threshold) [40]. $\rho_{\text{crit}}$ is defined by:

$$\rho_{\text{crit}} = \omega^2 \left( \frac{m_e \varepsilon_0}{e^2} \right),$$  \hspace{1cm} (5.7)
where $\omega$, $m_e$, $e$ and $\varepsilon_0$ are the laser angular frequency, the effective mass of a quasi-free electron, the electron charge and the vacuum dielectric permittivity respectively. The material is ablated when the maximum value of the time-dependent free-electron density exceeds $\rho_{\text{crit}}$. Additionally, from experiments and diagnostics another interesting threshold criterion $\rho_{\text{damage}}$ (damage electron density where damage of the material sets in) can be introduced. $\rho_{\text{damage}}$ is a material dependent quantity and for transparent dielectrics it is determined empirically according to [170]

$$\rho_{\text{damage}} = 0.025\rho_{\text{crit}}$$

The limits $\rho_{\text{crit}}$ and $\rho_{\text{damage}}$ correspond to the electron density thresholds for ablation and damage respectively. Therefore, the parameter space can be classified into an ablation region ($\rho_{\text{max}} > \rho_{\text{crit}}$), a damage region lies between damage ($\rho_{\text{damage}} < \rho_{\text{max}} < \rho_{\text{crit}}$) and neither ablation nor damage region ($\rho_{\text{max}} < \rho_{\text{damage}}$). This can be illustrated by taking a laser wavelength of 532nm which leads to a critical electron density value of $3.95 \times 10^{21} cm^{-3}$ and $\rho_{\text{damage}}$ value of $0.09875 \times 10^{21} cm^{-3}$. The ablation region, the damage region, and the no-ablation/damage region are shown in Figure 5.8 by the green region, yellow region, and red region respectively.
5.4 Laser Drilling

Laser beam drilling is one of the most widely used thermal-energy-based non-contact type of advanced machining processes which is applied for a wide range of materials. Depending on the application and the requirements regarding quality and efficiency, laser drilling is often the only applicable technique for drilling holes for cooling channels with high inclination angles in turbine blades or drilling very small cavities in injection tubes. However, there are gaps in understanding the dynamics of the drilling process, especially with regard to issues related to the hole shape. Numerical modeling and simulation of the laser drilling process improves the understanding without the need for executing a lot of experiments in the real world. Modeling and simulation of laser drilling is a complex process as it is described as a continuum physical problem for mass, momentum (liquid flow) and energy (heat conduction/temperature distribution within the solid and liquid) that is applied separately to the three different phases of solid, liquid and gas [159].

In this section, a simple but powerful heuristic approach for laser drilling is proposed. According to U. Eppelt (personal communication, May, 2015), an asymptotic model is generated based on the concept of an ablation threshold. The model has an ablation threshold characterized by the threshold fluence $F_{TH}$, which is material specific and has to be determined in the model. For laser ablation, the applied fluence $F$ is a fundamental process parameter which specifies the asymptotic shape of the drill wall. As shown in Figure 5.9 (a), the asymptotic shape of the hole is characterized by a local angle of incidence $\theta$. The local angle of incidence $\theta$ is given by:

$$\cos(\theta) = \frac{F_{TH}}{F}, \quad F = F(x), \quad \theta = \theta(x), \quad x \in (0, \infty), \quad (5.9)$$
which leads to the equation of the asymptotic local slope \( \frac{dz(x)}{dx} \) of the asymptotic drilling wall defined by:

\[
\frac{dz(x)}{dx} = \begin{cases} \sqrt{\left( \frac{F(x)}{F_{TH}} \right)^2 - 1} & \text{if } F(x) \geq F_{TH} \\ 0 & F(x) < F_{TH} \end{cases}, \quad z(x) \bigg|_{x \to \infty} = z_{\infty}, \quad (5.10)
\]

where \( z(x) \) is the depth of the drilled wall and \( x \) is the lateral coordinate with respect to the laser beam axis. In order to estimate \( F_{TH} \), one single simulation run is executed and the width of the drill at the bottom is fitted with one single experimental observation. A cross section of the simulation and experiment is shown in Figures 5.9 (b) and (c) respectively. The absolute relative error of the entry and exit diameter between the simulation and experiment is between 3.58% and 5.8%.

Figure 5.9: Process sketch (a). Simulation (b). Simulation (solid curve) with a drilling experiment (c).

One of the challenges in laser drilling applications is to achieve a straight parallel shape of the drill walls. In the laser drilling community, this challenge is referred to as conicity. The objective in this section is to minimize the conicity which is described by:

\[
\text{conicity} = \frac{\text{Width}_{Top}}{\text{Width}_{Bottom}} \in (a, b), \quad a < 1, \quad b > 1 \quad (5.11)
\]

where \( \text{Width}_{Top} \) is the width at the top of the workpiece and \( \text{Width}_{Bottom} \) is the width at the bottom. Since it not practical to do the measurement directly on the material surface, the \( \text{Width}_{Top} \) is measured at the width at the 15% of the workpiece thickness \( d \), with \( d \) set to 1.8mm. For metamodeling, the reduced model is considered as a black box model that have five parameters, listed in Table 5.4, and one criterion which is the conicity given by Equation 5.11.
For understanding the effect of the parameters on the criterion, the metamodeling technique is applied. The goal is to generate a fast and frugal surrogate that replace the numerical asymptotic model. The metamodel is generated first with 8000 training points using Hammersley sampling design presented in Section 2.2.3. Second, an RBFN with multiquadric basis functions is used for interpolation. In order to assess and estimate the metamodel quality, an additional data set with 2000 simulations is generated to calculate the following statistical measures: the relative mean squared error \( RMSE \) is 0.5\% and the coefficient of determination \( R^2 \) is 0.918.

Table 5.4: The five dimensional parameter settings required for minimizing the conicity

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Ranges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pulse Duration ( tP ) [ms]</td>
<td>0.1 - 1.5</td>
</tr>
<tr>
<td>Laser Power ( PL ) [kW]</td>
<td>3 - 10</td>
</tr>
<tr>
<td>Focal position ( z_0 ) [mm]</td>
<td>-8 - 8</td>
</tr>
<tr>
<td>Beam Radius ( w_0 ) [( \mu )m]</td>
<td>50 - 350</td>
</tr>
<tr>
<td>Rayleigh length ( zR ) [mm]</td>
<td>3 - 35</td>
</tr>
</tbody>
</table>

Figure 5.10 shows a modified screenshot of memoSlice with the five-dimensional laser drilling metamodel. A HyperSlice view with superimposed gradient trajectories guides the user to the next local minimum (blue) and maximum (red) in the vicinity of the selected parameter settings. Two criteria are shown: the conicity in the upper part of the HyperSlice and density of the training points in the lower part. On the right of the figure, a corresponding snapshot from one single simulation with parameter settings located at the seed point indicated by the red star.

Figure 5.11 shows a snapshot of the HDViz tool of the laser drilling metamodel. One trace is detected by the colored line where blue is the global minimum and red is the global maximum. The geometric representation is embedded in two dimensions by projecting the extremal points and the regression curves onto its first two principal components. These traces can be followed visually through the high-dimensional parameter space revealing the technical parameters or physical reasons for any deviation from the optimum performance; the global minimum (Figure 5.11 (a)) and the global maximum (Figure 5.11 (c)) with their corresponding simulation). The black arrow in the middle plot represents a strong
change of the relationship of the Rayleigh length $z_R$ on the conicity (the effect of large values of $z_R$ become insensitive on the conicity).

In order to detect the behavior of the parameters on the conicity, the Rayleigh length ($z_R$) is fixed to a small value (3mm) and to a large value (30mm). The analysis is then performed on a four-dimensional parameter space. Results in Figure 5.12 show a clique graph, presented in Section 4.1.3, where the red circles represent the main effects and the blue lines represent the interaction effect. The widths of the circles and lines represent the value of the effect, the thickest the width is, the highest the effect is. In the EE screening
plot, presented in Section 4.1.2, $\mu$ reflects the biggest influence of the parameters on the output and $\sigma$ describes the higher order effects such as non-linearity or interactions between inputs. Figure 5.12 (a) illustrates the global sensity analysis measures with a large $zR = 30mm$, while Figure 5.12 (b) illustrates the global sensity analysis measures with small $zR = 3mm$.

Figure 5.12: The EE screening plots and Clique graphs for: (a) large $zR$ (30mm), and (b) small $zR$ (3mm).

The results for the laser drilling model can be summarized as follows: i) within the five parameter domain space, defined in Table 5.4, and for a large Rayleigh length $zR >$, the pulse duration $t_P$ and the laser power $PL$ have the highest main effect for having a straight drill walls, and ii) the focal position has a negligible effect. However, for a small Rayleigh length $zR$, the duration of the pulse $tP$ and the focal position $z0$ play the highest main effect.

5.5 Oxygen Cutting

Oxygen cutting or flame cutting is a combustion process used for separating and shaping mild and low-alloyed components with a thicknesses up to two meters [132]. As shown in Figure 5.13, the process involves an oxygen and fuel gas being burned to heat the metal to the ignition temperature while at the same time feeding a jet of high purity oxygen centered on to the material to be cut [132]. It is carried out using a torch where oxygen and fuel gases (for example acetylene, propane) are passed. The main cutting tool, which is the cutting jet, is described by the oxygen gas in a central part surrounded by a fuel gas jet in the outer ring. The choice of fuel gas depends on the cutting thickness which affects the cutting quality as well as the required time used for preheating [10]. After ignition, the energy dissipated from the flame and the heat of the cutting zone is transformed into a process temperature of approximately 1900K [190]. At this high temperature, combustion and rapid oxidation of the steel takes place and a large amount of heat is generated that
melts both the oxides and the underlying metal layers. The resultant liquid oxides and molten metal are blown away by the oxygen jet from the cut cavity. There exist gaps in understanding the dynamics of this process, especially with regard to issues related to the role of oxygen in the cutting zone. Choosing and configuring the right process parameters becomes a very challenging task that requires experience and understanding. This is due to the fact that the cutting process responds differently for different kinds of jet designs, fuel gases, oxygen purity, material geometry and material thicknesses.

![Schematic of the oxygen cutting process.](image)

**Figure 5.13:** Schematic of the oxygen cutting process.

In order to obtain a better know-how of the oxygen cutting process and minimize the material waste of plate material being cut, a comprehensive and reasonable numerical model is necessary to investigate the effect of the exothermic reaction of oxygen with iron that provides a considerable thermal energy to the cutting process. OxyCut, which is a tool developed by the Chair Nonlinear Dynamics of Laser Processing (NLD) of RWTH Aachen, is a reduced model that determines the relationship between the parameters of oxygen cutting and the physical processes of heat and mass transfer in the area of the cut surface as well as the chemical reaction that takes place on the melt surface when iron is transformed to iron oxide. OxyCut calculates the maximum and minimum cutting velocity in addition to other criteria like cutting limits, gas velocity, burn up, melting thickness, oxide thickness, etc. For further details on the numerical model, the reader is referred to [44].

In this example, the main focus is to: i) optimize the cutting process by considering the maximum cutting velocity as the objective function (maximizing); and ii) understand the relationship between the applied energy that is generated by the process parameters and
the maximum cutting speed. The maximum velocity $V_{\text{max}}$ is defined by:

$$V_{\text{max}} = f(p, XO2P, N_{\text{diam}}, T_{O2}, T_{\text{mat}}, d_{\text{mat}}, cw_{\text{upper}}, a_{\text{front}}),$$  \hspace{1cm} (5.12)

where the process parameters $p$, $XO2P$, $N_{\text{diam}}$, $T_{O2}$, $T_{\text{mat}}$, and $d_{\text{mat}}$ are the gas pressure, oxygen purity, nozzle diameter, gas temperature, material temperature, and material thickness respectively. In addition to two model parameters $cw_{\text{upper}}$ and $a_{\text{front}}$, which correspond to the cutting width and the cutting inclination angle respectively. In order to construct the process map for $V_{\text{max}}$ within the parameters design space, which is listed in Table 5.5 in details, the OxyCut function $f(x)$ (the reduced physical model) is considered as a black box model that has 8 parameters and 1 criterion.

Table 5.5: Process parameter domain of oxygen cutting.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Units</th>
<th>Minimum</th>
<th>Maximum</th>
<th>Seed point</th>
</tr>
</thead>
<tbody>
<tr>
<td>gas pressure $p$</td>
<td>[bar]</td>
<td>8</td>
<td>30</td>
<td>19</td>
</tr>
<tr>
<td>oxygen purity $XO2P$</td>
<td>[%]</td>
<td>90</td>
<td>100</td>
<td>95</td>
</tr>
<tr>
<td>nozzle diameter $N_{\text{diam}}$</td>
<td>[mm]</td>
<td>1</td>
<td>15</td>
<td>8</td>
</tr>
<tr>
<td>gas temperature $T_{O2}$</td>
<td>[K]</td>
<td>300</td>
<td>1000</td>
<td>650</td>
</tr>
<tr>
<td>material temperature $T_{\text{mat}}$</td>
<td>[°C]</td>
<td>20</td>
<td>1000</td>
<td>510</td>
</tr>
<tr>
<td>material thickness $d_{\text{mat}}$</td>
<td>[mm]</td>
<td>100</td>
<td>2000</td>
<td>1050</td>
</tr>
<tr>
<td>cutting width $cw_{\text{upper}}$</td>
<td>[mm]</td>
<td>1</td>
<td>22.5</td>
<td>11.75</td>
</tr>
<tr>
<td>Cutting front inclination angle $a_{\text{front}}$</td>
<td>[°]</td>
<td>1</td>
<td>10</td>
<td>5.5</td>
</tr>
</tbody>
</table>

A metamodel is constructed by first generating 10000 training points from a Hammersley design [78] (9872 points) augmented with a two-levels full factorial sampling design (corner points of the parameter hypercube-128 points). These training points are illustrated in Figure 5.14 by the small gray spheres shown in the upper left hyper slices matrix. The big white sphere illustrates the seed point which is located at the parameter settings listed also in Table 5.5. The lower right part of the hyper slices matrix in Figure 5.14 reflects the density function of the training points. The density function, introduced in Section 3.1.2, is equal to one when there is exactly a training point (simulation point), and decreases to zero in a sparse region far away from the training points. Since the
model contains a discontinuity (Cut/No-Cut Region), the one-shot-approach from Section 3.2.1 for discontinuous responses is applied and a metamodel function is used to map the discrete sampling points to a continuous function $f_{V_{\text{max}}}$. The process map of the metamodel is shown in Figure 5.15.

Figure 5.14: The illustration of the 8-dimensional discrete training dataset from OxyCut model visualized in memoSlice. The training points are illustrated by small gray spheres shown in the upper left part of the hyperSlice matrix. The density function which reflects the position of the sampling points is illustrated by the lower right part of the hyperSlice matrix.

In order to validate the metamodel, additional 3000 points from a Latin hypercube design are generated from OxyCut and used as a validation data set to calculate the relative mean absolute error ($RMAE$) given by:

$$RMAE = 100\% \cdot \left( \frac{1}{n_V} \sum_{i=1}^{n_V} \left| \frac{V_{\text{max},i} - f_{V_{\text{max},i}}}{V_{\text{max},i}} \right| \right), \quad n_V = 3000,$$

and the coefficient of determination $R^2$, defined by:

$$R^2 = 1 - \frac{\sum_{i=1}^{n_V} (V_{\text{max},i} - f_{V_{\text{max},i}})^2}{\sum_{i=1}^{n_V} (V_{\text{max},i} - \bar{V}_{\text{max}})^2}, \quad n_V = 3000.$$

The values of $RMAE$ and $R^2$ are calculated to be 10.14% and 0.91 respectively which reflects a high accuracy and determination.
5.5 Oxygen Cutting

Figure 5.15: Visualizing the 8-dimensional RBFN metamodel $f_{V_{\text{max}}}$ in memoSlice. The RBFN is illustrated by the blue to red continuous function in the upper left hyper slices matrix. The gray areas illustrate the no-cut region. A classification RBF model function is illustrated by the lower right hyper slices matrix. In the classification model, the white region between the green (cut) and red (no-cut) colors represent the uncertainty region. The diagonal hyper slices matrix illustrates the impact of each individual parameter on $f_{V_{\text{max}}}$.

In the context of decision-making and understanding the effect of the model parameters on the maximum cutting velocity, the material is set to a fixed thickness ($d_{\text{mat}}=600$ mm) which makes the dimensionality of the metamodel decrease from 8-dimensions to 7-dimensions. A Monte Carlo simulation is performed by generating 1000 random points according to a uniform distribution on the parameter range. The data is selected such that the criterion $f_{V_{\text{max}}}$ falls within the range of 0.40 m/min and 0.52 m/min. The resulting data are visualized by the parallel coordinate plot shown in Figure 5.16 (a) and the box-whisker diagram shown in Figure 5.16 (b). The x-axis of these plots represents the parameters of the domain space and the y-axis represents the normalized value of the parameters (all normalized to a range between 0 and 1). The parallel coordinates plot is a visualization technique pioneered in the 1970s which has been applied to a diverse set of multi-dimensional problems [71]. Each parameter corresponds to an axis, and the
7 axes are located on equidistant vertical lines. A data element in 7-dimensional space is displayed as a line that connects a set of points, each located on one axis. The total sets of lines perceive the final image structure. The box whisker diagram is similar to the parallel plot, however each box in the diagram is divided into lower quartile (25%), median, upper quartile (75% and above), and rectangle boxes that reflect the significance of the parameter on the criterion [5]. It is shown that the material temperature $T_{mat}$ plays the highest effect for maximizing the maximum cutting velocity $V_{max}$. The results show that in order to achieve a high cutting velocity between (0.40 and 0.52 m/min), $T_{mat}$ should be equal to a value that lies in the normalized range of 0.8 and 1, which corresponds to a value between 804 and 1000 °C. The second and third sensitive parameters are the nozzle diameter $N_{diam}$ and gas temperature $T_{O2}$ respectively.

Figure 5.16: (a) Parallel coordinate plot and (b) Box-whisker Diagram for 1000 multi-dimensional data set where $V_{max}$ lies in the range of 0.40-0.52 m/min. The green arrows determine the shift or the route to guide the analyst from a selected seed point or current status (here a parameter configuration that lies in the middle of every parameter range) to the global optimum or required target.

For optimizing the process using the process map, the memoSlice visualization tool introduced in Section 4.2.1 is considered. The upper left part of the HyperSlice plot denotes the $f_{V_{max}}$ with the corresponding combinations of parameters slices. The exact value of $f_{V_{max}}$ using the parameters configuration listed in Figure 5.17 (a) is represented by the upper black arrow (0.198 m/min). The lower right matrix represents the density function which reflects the position of the training points. The exact value of the density function of the parameters configuration point is represented by the black arrow on the right (0.17). By following the steepest ascent, a new configuration point is acquired and listed in Figure 5.17 (b). The $f_{V_{max}}$ is maximized from 0.198 m/min to 0.454 m/min.
5.5 Oxygen Cutting

Figure 5.17: Optimizing the 7-dimensional RBFN metamodel $f_{V_{\text{max}}}$ in memoSlice by changing the position of the seed point according to three different parameter configurations: (a) configuration listed in Table 5.5 (b) configuration shown in Figure 5.16 (c) minimizing the density function in the vicinity (coming closer to a training point) of the target configuration shown in Figure 5.16.

The value of the density function of the new seed point, which is equal to 0.01, ensures that the configuration settings of the parameter list lie in a sparse region that reflects an interpolated value of $f_{V_{\text{max}}}$ that might be inaccurate. In this uncertain region, it is always important after the analysis to validate the value by running additional simulations. However, since in this work the focus is on tendencies and not on single values, the density function is maximized to 1 (reflects the position of an exact simulation point). The new parameter configuration illustrated by Figure 5.17 (c) denotes a maximum of $f_{V_{\text{max}}}$ that match directly a simulation point in the training set.

For demonstrating the decomposition of the parameter space by Morse Smale approach [117], the OxyCut metamodel data is loaded by HDViz tool [53]. The loaded data set consists of 22,000 random samples of the metamodel that contains 8 columns (the 7 input parameters and their corresponding $f_{V_{\text{max}}}$). The data are normalized between 0 and 1 according to the maximum and minimum of every column. The 22000 points are generated only in the feasible (cut) region (Morse-Smale does not succeed with responses that contain discontinuity). Figure 5.18 shows the Morse Smale graph at the coarsest level of detail (highest persistence) with a single maximum and minimum. Since the tool offers an inverse problem (obtaining parameters from the criterion), the three obtained values of $f_{V_{\text{max}}}$ in Figure 5.17 (a), (b), and (c) which are equal to 0.198 $m/min$, 0.454 $m/min$, and 0.41 $m/min$ are normalized between 0 and 1, and hence correspond to, 0.37, 0.89, and 0.79 respectively. This is illustrated in Figure 5.17 (a), (b), and (c) respectively. The colored regression line is varied from blue (the global minimum) to red (the global
Figure 5.18: Visualization of the 22000 OxyCut metamodel data set at coarsest level of detail (0 global minimum and 1 global maximum). The black arrow denotes the inverse relationship between the $f_{V_{\text{max}}}$ criterion and the 7 model parameters all normalized between 0 and 1 according to their corresponding maximum and minimum ranges. (a), (b), and (c) corresponds to 0.198, 0.454, and 0.41 respectively (same three values of $f_{V_{\text{max}}}$ shown in Figure 5.17) normalized between 0 and 1. The shaded red area illustrates an area in the vicinity of the no-cut region (no points are given).

maximum) and the geometric representation is embedded in two-dimensional by projecting the extremal points and the regression curves onto its first two principal components. The relationship of the parameters on $f_{V_{\text{max}}}$ in the form of a single regression curve is observed by the plots shown in Figure 5.18. It is visible that a relative increase in the gas pressure, oxygen purity, nozzle diameter, gas temperature, and material temperature lead to an increase of the maximum cutting velocity. The shaded red area in Figure 5.18 denotes an area close to the discontinuity between the cut and no-cut regions. Since the non-feasible data-points are left out of the loaded 10000 data-points, the analysis values are extrapolated. This corresponds to a region that might be misleading to the user or operator.

The two global sensitivity analysis methods (Elementary Effect EE and Variance Decom-
position VD) are conducted in order to: i) screen the parameters that have the greatest influence on the maximum cutting velocity, and ii) detect and evaluate in addition to the main effect, the first order interaction effects. For investigating the convergence, the sample sizes and the corresponding number of model evaluations required for both EE and VD are varied according to the values listed in Table 5.6.

Table 5.6: Sample sizes and number of model runs performed for each of the sensitivity analysis methods.

<table>
<thead>
<tr>
<th>Sample Size</th>
<th>Model Evaluations</th>
<th>Required Evaluation Time [hours]</th>
</tr>
</thead>
<tbody>
<tr>
<td>VD-8000</td>
<td>581,752</td>
<td>12.28</td>
</tr>
<tr>
<td>VD-9000</td>
<td>653,895</td>
<td>13.47</td>
</tr>
<tr>
<td>VD-10000</td>
<td>727,365</td>
<td>15.09</td>
</tr>
<tr>
<td>EE-500</td>
<td>11,423</td>
<td>2.38</td>
</tr>
<tr>
<td>EE-1000</td>
<td>23,316</td>
<td>4.88</td>
</tr>
<tr>
<td>EE-1500</td>
<td>34,667</td>
<td>7.01</td>
</tr>
</tbody>
</table>

For the EE method, a 30 levels grid is fixed in the domain space and 500, 1000, and 1500 trajectories are generated randomly for the analysis. The three different studies require 11423, 23316, and 34667 model evaluations respectively. It is important to mention that the trajectories that contain limit values (no-cut) were filtered out from the analysis. For the VD method, the sampling distributions for all parameters were chosen to be uniform. The three different sample sizes of 8000, 9000, and 10000 required 581752, 653895, and 727365 model evaluations respectively. The required time to calculate the sensitivity indices for both methods on a quad-core Intel i5 processor (3.2 GHz) is listed in Table 5.6. If the OxyCut takes around 5 seconds to estimate the $f_{V_{max}}$ for one parameter configurations settings, then a sensitivity analysis run by using OxyCut evaluations requires around 15.8 hours to complete. However when using a metamodel, the sensitivity analysis run requires 2.48 hours, and thus attaining an increase in the computational efficiency up to 80%. The three different sampling sizes for the two methods show a good convergence. This is expected due to the high number of the varied sampling size. The outcome of the EE and VD methods is shown graphically by the screening plot illustrated in Figures 5.19 (a) and (b) respectively.
Figure 5.19: Outcome of the EE and VD global sensitivity analysis methods of the oxygen cutting process: (a) the EE screening plot (b) and the Clique graph. The EE plot screens the parameters that have linear and very small main effect on \( V_{\text{max}} \) (the gas pressure and the cutting front inclination angle).

In Figure 5.19 (a), \( \mu \) assesses the overall influence of the input factor on the output. If \( \mu_i \) is large, the parameter \( x_i \) has the biggest influence on the (scalar) output. In addition, \( \sigma \) measures the nonlinearity or interaction effect. If \( \sigma_i \) is large, then the parameter \( x_i \) has either a nonlinear effect on the output or an interaction effect with other inputs [115]. The outcome of the VD method with the sample size of 10000 is plotted in Figure 5.19 (b) in the form of a clique graph. The results from the global sensitivity analysis methods indicate that the cutting width \( cw_{\text{upper}} \), the nozzle diameter \( N_{\text{diam}} \), as well as the material temperature \( T_{\text{mat}} \) are sensitive and have an interaction and high nonlinear effect on the maximum velocity. Less sensitive parameters are the gas temperature \( T_{O2} \) (interacted with nozzle diameter and material temperature) and the oxygen purity \( XO2P \). The gas pressure \( p \) and the cutting front inclination angle \( a_{\text{front}} \) have very small sensitivity and a linear effect on the criteria. It is very essential to mention that the sensitivity analysis results are determined by the minimum and maximum ranges of the model parameters.

5.6 Sheet Metal Cutting

Laser cutting is a thermal separation process that is widely used in several areas of manufacturing such as trimming, processing of flat metal sheets for shaping, and contour cutting applications. In many sectors of production, laser technology has already become
a state of the art technology, where laser cutting is definitely the most established one [2]. The laser beam permits tool-free machining with active heat energy, thus allowing the process to be a fast and accurate technology that is not exposed to any wear. During the fusion cutting process, a high energy density laser beam is focused on a work surface. The thermal energy is absorbed which heats and transforms the work volume into a molten, vaporized, or plasma state that can easily be removed by the flow of high pressure assisting gas jet [136], [35]. The schematic of the laser cutting process is shown in Figure 5.20.

![Figure 5.20: Schematic of conventional laser cutting process](image)

Important degradation of the quality is due to the onset of unevenness and roughness of the cut edges, the appearance of adherent dross, as well as other properties like gas consumption and robustness with respect to sensitive parameters such as nozzle standoff distance and others. These defects have in common that they originate from the dynamical behavior of the cutting process [161].

There are gaps in understanding the dynamics of the process, especially with regard to issues related to the cut quality. Numerical modeling and simulation of the laser cutting improved the understanding of the process without the need of executing numerous experimental tests [162]. The three elements involved in laser cutting are the gas jet, the laser beam and the material to be cut. Therefore, the modeling of the cutting gas flow, the radiation propagation and the ablation of the material (in fusion cutting: removal by melt ejection) has to be accomplished as well as the numerical solvers of these models have to be implemented. One of the current challenges in R&D is to design beam-shaping optics such that the ripple structures on the cutting kerf surface stay minimal, as shown in Figure 5.21.

QuCut, a tool also developed at the institute of Nonlinear Dynamics of Laser Processing (NLD) of RWTH Aachen, is a numerical model for continuous wave laser cutting that takes into account the spatially distributed laser radiation. QuCut reveals the occurrence
Figure 5.21: Smooth (left) and non-smooth (right) cutting surface with ripple structures

of ripple formation at the cutting front and defines a measure for the roughness on the
cutting kerf surface. The tool is based on a numerical model which involves two coupled
nonlinear partial differential equations. It is important to mention here, that there is no
explicit function of the roughness in terms of the beam optics parameters. The model is
fully numerical and will be considered in this chapter as a black-box model. For further
details on the mathematical analysis, the reader is referred to [183], [182].

Production with laser cutting systems is facing a big challenge when choosing the appro-
priate machine parameter set (comprising parameters for the laser optical system, nozzle
design parameters, and others). For this reason, laser cutting machine manufacturers
are attaching working parameter settings for specific pre-defined cutting tasks to their
machines and documenting them in forms of catalogues or technology tables. These tech-
nology tables, which are generated by every laser cutting machine manufacturer for their
machines specifically, are expensive/labor-intensive, highly time-consuming (due to the
vast number of cutting experiments) and finally revealing only a discrete set of poten-
tially beneficial operating points of the machine. The data contained in these tables is
produced by numerous experimental tests performed by appropriate Design of Experi-
ment (DOE) techniques as well as other experience-based procedures. As a beam shape
optimization cannot be done with one single simulation only, an expert has to explore or
visualize scalar quality criteria for different parameter design sets. It takes QuCut about
6 minutes to estimate the roughness for one parameter set. Thus, the total computation
time required for optimization, or sensitivity analysis, by executing only 1000 iterations,
would be around 3 days. This is also impossible to perform an online exploration. Thus,
with the help of a fast metamodel, a process map is generated. Due to the experimental
costs and effort, the metamodel is generated from a discrete data set of the simulation.
It provides the operator with a continuous relationship between the quality (roughness)
and the optic beam parameters. The goal of this use case is to find the optimal param-
eters configuration of certain laser optics that result in a minimal ripple structure (i.e.
roughness). The 5 laser optics design parameters investigated here are the beam quality,
the astigmatism which is a measure of the shift between the horizontal and vertical cross

sections of the beam focus along the beam path, the focal position, and the beam radius in x and y directions (as an elliptical laser beam was used). The properties of the sampling design are listed in Table 5.7.

Table 5.7: Process design domain of sheet metal cutting

<table>
<thead>
<tr>
<th>Beam Parameters</th>
<th>Minimum</th>
<th>Maximum</th>
<th>Sampling Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beam Quality $M^2$</td>
<td>7</td>
<td>13</td>
<td>7</td>
</tr>
<tr>
<td>Astigmatism $Ast[m]$</td>
<td>-0.0150</td>
<td>0.0250</td>
<td>9</td>
</tr>
<tr>
<td>Focal position $fp[m]$</td>
<td>-0.0080</td>
<td>0.0020</td>
<td>11</td>
</tr>
<tr>
<td>Beam Radius x-direction $w_x[m]$</td>
<td>0.0001</td>
<td>0.00023</td>
<td>6</td>
</tr>
<tr>
<td>Beam Radius y-direction $w_y[m]$</td>
<td>0.0001</td>
<td>0.00023</td>
<td>6</td>
</tr>
</tbody>
</table>

The selected criterion is the surface roughness ($R_s$ in $\mu m$) simulated at a 7 $mm$ depth of an 8 $mm$ workpiece thickness. The full data set is 24948 samples in total. A schematic for the architecture of the RBFN for this process is shown in Figure 5.22.

Figure 5.22: Architecture of Radial Basis Function Network (RBFN) for sheet metal cutting.

In order to assess the quality of the mathematical interpolation, 5 different RBFN metamodels are generated according to 5 randomly selected sample sets of size 1100, 3300, 5500, 11100 and 24948 data points from the full dataset. As shown in Figure 5.23, the metamodels are denoted by Metamodel A to E. Metamodel F, which is used as a reference for comparison, is a two dimensional metamodel with a finer sampling points denoted by the blue points.

In Figure 5.23, the contour shapes of the metamodels A to E (five-dimensional), become similar to the ones in Metamodel F (two-dimensional) by simply adding more sampling
Figure 5.23: two-dimensional contour plots of different metamodels at $M^2 = 10$, $Ast = 25\, \text{mm}$, $w_y = 134\, \mu\text{m}$. The polynomial linear regression metamodel (F) on the bottom right contains more sampling points and is shown here for evaluation of the metamodel quality (A-E).

Points to the metamodel. Additionally, the reddish-orange region (resembling large roughness regions) at a focal position corresponding to $8\, \text{mm}$ and for a beam radius between $100\, \mu\text{m}$ and $140\, \mu\text{m}$ becomes progressively weaker from A to E. This is a consequence of using more sampling points in the vicinity of the slice (Beam Radius x-direction, focal position). The star-shaped marker, denoting the seed point of the investigation, represents the current cutting parameter settings and the arrow trajectory shows how an improvement in the cut quality is achieved. The results show that in order to minimize the cutting surface roughness in the vicinity of the seed point, the beam radius in the feed x-direction should be decreased and the focal position should be increased. In this test case, the minimum number of sampling points with an RBFN model is already a good choice for giving an optimized working point for the laser cutting process. This is due to the fact that eventhough these metamodels have different accuracy values, they show similar tendencies that support the developer with his decision making step.

The advantage of using a metamodel in this process over the full scale simulation is denoted in Table 5.8. As stated before, a single evaluation run requires 6 minutes using QuCut, and less than 1 second on all of the above generated metamodels. So generating a two-dimensional $10 \times 10$ grid (100 evaluations) contour plot, similar to what is shown in Figure 5.23, would require around 30 seconds by using a fast metamodel or 10 hours
by using the simulation model.

Table 5.8: Quality of the metamodels A-E for sheet metal cutting

<table>
<thead>
<tr>
<th>Metamodel</th>
<th>MAE[μm]</th>
<th>R$^2$</th>
<th>Generation Time</th>
<th>Single evaluation time[s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>19.69</td>
<td>0.89</td>
<td>∼ 30 seconds</td>
<td>0.032</td>
</tr>
<tr>
<td>B</td>
<td>17.89</td>
<td>0.90</td>
<td>∼ 50 minutes</td>
<td>0.044</td>
</tr>
<tr>
<td>C</td>
<td>15.50</td>
<td>0.92</td>
<td>∼ 1 Day</td>
<td>0.076</td>
</tr>
<tr>
<td>D</td>
<td>14.91</td>
<td>0.93</td>
<td>∼ 3 Days</td>
<td>0.143</td>
</tr>
<tr>
<td>E</td>
<td>13.63</td>
<td>0.94</td>
<td>∼ 1 Week</td>
<td>0.319</td>
</tr>
</tbody>
</table>

The Mean Absolute Error (MAE) of the criterion surface roughness and the coefficient of determination ($R^2$) are calculated and compared to each other. As mentioned earlier in Chapter 3, the smaller the value of $MSE$ is, the more accurate the metamodel becomes. Additionally, the closer the value of $R^2$ gets to 1, the more accurate the metamodel is. The results show that the quality of the metamodel naturally depends on the number of sampling points; the quality is improved when the number of training points is increased. However this increase the metamodel generation time as shown in Table 5.8. This is due to the time required to calculate the inverse of the dense matrix

$$\left(H^TH + \lambda\right)^{-1}$$  \hspace{1cm} (5.15)

found in Equation 3.8. Starting with the analysis on Metamodel E, a Monte Carlo simulation is performed by generating 1000 random points according to a uniform distribution on the parameter range. The data is selected such as the criterion $f_{R_x}(x)$ falls within the range of $10 \mu m$ and $30 \mu m$. A statistical test is run on the metamodel by generating a parallel coordinate plot shown in Figure 5.24 (a) and a box-whisker diagram shown in Figure 5.24 (b). The plots provide a new parameter configuration (denoted by the green arrows) for minimizing the roughness.

Additionally, the Morris method is then applied to identify the subset of beam parameters that have the greatest influence on the cutting roughness. The Morris method is applied to the Metamodel E, since it showed the highest accuracy, using 40 sampling levels and 300 trajectories. As shown in Figure 5.25, astigmatism is the most valuable factor, the beam radius in the feed x-direction and the focal position are the second important factors, and the remaining parameters ($M^2, w_y$) are relatively less sensitive to the surface roughness.
Figure 5.24: (a) Parallel coordinate plot and (b) Box-whisker Diagram for 1000 multi-dimensional data set where $f_{R_{c}}(x)$ lies in the range of 10-30 $\mu m$. The green arrows determine the shift from the seed point to the global optimum.

Figure 5.25: Representations of each input parameter on the screening plot by a point with coordinates $(\mu, \sigma)$.

From fundamental research in modelling and simulation it is proposed that increasing the beam diameter in feed direction should be beneficial to the surface quality in laser cutting, as it should reduce the ripple-forming instability on the cut front and thereby decrease the roughness amplitude on the cut edges. This idea inherently lead to the concept of an elliptic beam, i.e. a beam with elliptic cross-section. However, this concept is needed to be quantified in order to get tested in the real-world. So the question is, which minimal diameters the beam should have and at which position with respect to the workpiece the focuses corresponding to these minimal beam diameters should be placed. Together with the beam quality giving a measure for the beam divergence this makes five parameters
to choose beneficially, a perfect task for accomplishing with the process map presented in previous chapters, where the metamodelling concept can prove its strengths.

Looking at the process map for laser cutting, shown in Figure 5.26, the star-shaped marker denoting the seed point of the investigation represents the current cutting parameter settings and the arrow trajectory shows how an improvement in the cut quality is achieved. The results show that in order to minimize the cutting surface roughness $R_z$ in the vicinity of the seed point, astigmatism $Ast$ should be increased, the beam radius in the feed direction $w_x$ should be decreased and the focal position should be raised $fp$. The fact that the minimal beam radius in x-direction is decreased to get a lower roughness value is actually no contradiction to the premise of increasing the beam radius in feed direction as the focal position such that the beam radius in feed direction at the position of the workpiece is actually increased.

Figure 5.26: Use of the metamodel and process map in the design of a new focusing optics (for elliptic beams) with the goal of reducing roughness in laser cutting; the blue color indicates regions in the domain space where smaller roughness values could be expected.
5.7 Summary

In this chapter, fast metamodels are used to mimic the response behavior of the physical numerical models of different laser processing applications. The small reduction in the computational time lead to a significant decrease in the total evaluation time required performing the sensitivity analysis, optimization, and visualization run times since numerous numbers of evaluations are required. Generating a metamodel is a user demanding procedure that involves compromises between many criteria since the metamodel with the greatest accuracy is not necessarily the best choice for a metamodel. The proper metamodel is the one which fits perfectly to the developer needs. The needs have to be prioritized according to some characteristics or criteria which are accuracy, speed, storage, visual aspects, sensitivity to parameters and ease of implementation [45]. The additional information allow a high flexibility to decision-makers in setting an appropriate parameter configuration to the process by taking into consideration the actual physical limitation (if cut occurs for example) and design limitations (manufacturing costs for example) of the process parameters.
Chapter 6

Conclusion and Outlook

6.1 Conclusion

In this thesis, the concept of metamodeling towards Virtual Production Intelligence is discussed. The two primary objectives are: (1) generating and validating a fast and frugal metamodel of complex simulation models; and (2) improving a basic understanding of a complex model by replacing the simulation model by a metamodel.

Since the process of learning from simulation results becomes very challenging when the simulation time and the number of parameters increase, a fast and frugal metamodel is used as a surrogate of the complex simulation. It allows much faster analysis of the simulation model with a controlled accuracy. Metamodels offer an excellent possibility to describe the process behavior of technical systems. Metamodeling approach defines a procedure to analyze and simulate involved complex physical models using fast mathematical models. They mimic the real behavior of the simulation model by considering only the input-output relationship in a simpler manner rather than the full simulation. Once a fast metamodel is generated with a moderate number of computer experiments, it offers making predictions at additional untried inputs, and thus the conventional engineering tasks such as optimization, sensitivity analysis, or design space exploration become easily possible due to the numerous number of evaluation runs that could be performed.

First, two approaches (one-shot and adaptive) for generating metamodels for continuous as well as piecewise responses are discussed in Chapter 3. Herein, a novel smart sampling algorithm for generating metamodels with piecewise (processes where the response is categorized to either feasible or non feasible) responses is proposed. The RBFN with multiquadric, gauss, and linear splines basis functions is used to construct the interpolation model, the density function of sampling points, and the classification model (required to
identify to which region in the domain space, i.e. feasible or non feasible, a new parameter configuration belongs) respectively. As expected, different numerical examples show that the quality of the metamodel is dependent on the number of sampling points since the quality and accuracy of the metamodel is improved when the number of training points increases. However, the accuracy does not necessarily mean that the best metamodel is achieved, since choosing a metamodel is a decision making procedure that involves a compromise between many other criteria like speed, visualization, complexity, storage, ease of the methods etc. in addition to accuracy.

Second, the metmodel is used as an emulator to perform different conventional engineering tasks such as optimization and sensitivity analysis. These tasks, which are presented in Chapter 4, are the main component of a navigation tool or Cockpit that can be used for R&D purposes as well as for production planning. They offer high flexibility to better: i) analyze and validate complex simulation models; ii) explore uncertainties; and iii) effectively identify the important/sensitive model parameters that contribute most to the process.

Finally, the metamodeling techniques are applied to five different laser manufacturing processes with different parameter dimensions ranging from two to seven dimensions. A process map is generated for each application in order to illustrate and verify the proposed method showing its applicability and efficiency. With the help of process maps, the knowledge about the process behavior is operative for production planning process. In each application, a new optimal working point in machine parameter configuration is found. Although the metamodeling techniques are applied to laser manufacturing applications, the goal of this work is to convince the simulation analysts to use the metamodeling techniques in their multi-dimensional applications. The technique is applicable to almost any economical, ecological, or technical process, where the process itself can be described by models giving scalar values representing the criteria.

6.2 Outlook

6.2.1 Reduced Models

Model reduction is the key ingredient to get an insight into the structure of the solution. Therefore, further work will focus on enhancing and finding more reduced physical, numerical, or even phenomenological models that are able to provide the solution properties of any production system much faster than the full numerical solution. For example,
providing the discontinuity (between feasible and non-feasible regions) in a form of an additional explicit (analytic) form would allow engineers to perform better analysis when extracting hidden information in special regions of the parameter space.

### 6.2.2 Metamodelling Performance

Additionally, further progress will focus on improving the performance of generating metamodels. In the RBFN for example, especially when the training data set becomes large (> 10000 sampling points), the time required to generate a metamodel becomes very costly (inversion of the dense matrix). Alternative parallelized solvers and even different metamodelling techniques will have to be tested. This will require testing additional metamodelling techniques such as fuzzy neural networks, symbolic regression, etc... Moreover, in order to improve the performance of the iterative approach, additional smart points will be considered such as the global optimum as well as the Expected Improvement (via Kriging-method) in the feasible domain. A very important issue in the future work is to focus not only how to achieve an optimal solution but also a robust one.

### 6.2.3 Sensitivity Analysis and Visualization

In the context of sensitivity analysis, further work will focus on: i) decomposing the domain space (via Morse-Smale-Complex for example) and ii) applying global sensitivity analysis, advanced data mining and machine learning techniques on the decomposed regions. This will provide better understanding and analysis of the full parameter space. Additionally, further work in visualization will be focus on creating a tool where the user can annotate certain interesting points (seed points for example) in the multi-dimensional parameter space (this might be a combination of memoSlice and HDViz).

### 6.2.4 Commercialize

The realization of the proposed examples is still impeded by the lack of a global interface between all of the visualization, optimization, and global sensitivity analysis. Therefore, further plans will focus on industrialization of the developed concepts and (numerical) tools by adapting their usability according to the specific needs of the users. This is done by either selling the developed tool to industrial clients which require the elaboration of licensing concept, or integrating the proposed concept in already known commercial software.
Literature


Appendix A

The HDViz visualization tool

The first step of HDViz consists of approximating the Morse-Smale complex by the k-nearest-neighbor graph algorithm [178] which traces the paths of the steepest descent and ascent based on the connectivity of the graph. At each point \( x_i \), the adjacent points \( \text{adj}(x_i) = \{ x_j : x_i \in \text{knn}(x_j), x_j \in \text{knn}(x_i) \} \) are calculated. The adjacent points which have lower and higher functional values are used to approximate the integral line which is traced out by the steepest descent according to

\[
p_{d}(x_i) = \arg\max_{x_j \in \text{adj}(x_i)} |f(x_i) - f(x_j)|
\]

and the steepest ascent according to

\[
p_a(x_i) = \arg\max_{x_j \in \text{adj}(x_i)} |f(x_j) - f(x_i)|
\]

with \( p_a(x_i) = x_i \) and \( p_d(x_i) = x_i \). The resulting complex contains a region for each local extremum that represents the ascending as well as descending manifolds. The outcome of the MS complex approximation is a set of k-partitioned crystals \( C = C_1, \cdots, C_k \). Each crystal \( C_i \) contains a set of points for the partition \( i \) [178].

The second step of HDViz consists of generating an inverse regression procedure that gives a curve in the \( p \)-dimensional domain of \( f \). For every crystal \( C_i \) with samples \( (x_i, y_i) \), the MS complex consists of a curve \( r_i : [\min_{x \in C_i} f(x), \max_{x \in C_i} f(x)] \mapsto \mathbb{R}^p \) which is modeled by the conditional expectation \( r_i(y) = E[X \in C_i | Y = y] \) which is predicted with a linear regression according to:

\[
r_i(y) = \left( \bar{Y}_i W(y) \bar{Y}_i^T \right)^{-1} \bar{Y} W(y) X_i^T \cdot u_1,
\]

where \( \bar{Y}_i = (1, Y_i) \) is a matrix with the first row all ones and the second row the function values of \( C_i \), \( u_1 = (1, 0)^T \) and \( W(y) \) is a \( n_i \times n_i \) diagonal matrix with \( W_{k,k} = K(y, y_k) \) where \( K \) is a Gauss kernel function defined by:

\[
K(y, y_k) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left( -\frac{|y-y_k|^2}{2\sigma^2} \right),
\]

where \( \sigma \) is the kernel bandwidth sigma that is adapted according to the data.
The shape of every crystal is resolved by the average distance of the data to the curve, which is the standard deviation. The standard deviation in a crystal \( C_i \) is a function of \( y \) and defined by:

\[
s_i(y) = \sqrt{\left( \frac{\sum_j^n K(y, y_i)d_i(x_j)}{\sum_j^n K(y, y_i)} \right)}.
\]  

(A.3)

where \( d_i(x_j)_m = \| r_i(y_j) - x_j \|^2 \). Another important measure that gives information about the number of points used in the computation of the regression curve point at \( y \) as well as indication of how the crystal in a specific region is densely sampled is the density function which is defined by:

\[
\rho_i(y) = \frac{1}{|X|} \sum_j^n K(y, y_j).
\]  

(A.4)

The third step of HDViz is visualizing the regression edges with their corresponding vertices (critical points) in a two-dimensional graph. High-dimensional (three-dimensional and higher) curves are embedded in a two-dimensional graph by reducing the dimension using the principle component analysis.

When samples are correlated and data variability is considered to be produced by the investigated process, the data can be restricted to a lower dimensionality. Principal components analysis (PCA) aims to find the structure in the data that hold the most variance [134]. PCA is an orthogonal linear transformation, converting the data to a new coordinate system. The greatest variance by any projection of the data lies on the first coordinate (principal component), and so on [53]. The transformation of the data into this new coordinate system is given by:

\[
Y^\top = X^\top W,
\]  

(A.5)

where \( X \) is the data matrix and \( W \) is the transformation (rotation) matrix, which can be obtained by singular value decomposition (SVD), multidimensional scaling (MDS) or an adaptive PCA method [156]. By using Singular value decomposition, the factorization of \( X \) can be achieved efficiently [60] and is defined by:

\[
X = W\Sigma V^\top,
\]  

(A.6)

where \( W \) is an \( m \times m \) matrix, \( \Sigma \) is an \( m \times n \) diagonal matrix, and \( V^\top \) is an \( n \times n \) matrix. The columns of \( W \) are the left singular vectors, the rows of \( V^\top \) contain the right singular vectors, and the elements of \( \Sigma \) are the singular values (only different from zero on the diagonal). If the rows of \( X \) are centered, the matrix \( W \) can be fitted in the PCA equation, this explanation is further developed in [184].
In the HDViz, the high dimensional graph is embedded into a plane that preserves the spatial relation among the extrema and the geometry of the partitions that connect them as good as possible using a three step approach [53]. First, the vertices are embedded into two dimensions using the principal components of the corresponding point set. Specifically, the extremal points $E = [e_1, \cdots, e_k]$ are projected onto their first two principal components $C_e = [c_1, c_2]$ with the projections $P_e = C_e E$. Second, the edges are embedded individually by projecting the polylines $L = l_1, \cdots, l_{nsamples}$ with $l_j = r_i(s_j)$ by $P_{L_i} = C_{L_i} r_i$, onto their first two principal components $C_{L_i}$ of $L_i$. Finally, the two-dimensional curves are connected to the projected vertices through affine transformations [122].