LINKÖPING STUDIES IN SCIENCE AND TECHNOLOGY.
DISSERTATIONS, NO. 1479

Design Automation for Multidisciplinary Optimization

A High Level CAD Template Approach

Mehdi Tarkian

Linköping University

“Design Automation for Multidisciplinary Optimization – A High Level CAD Template Approach”

Linköping Studies in Science and Technology. Dissertations, No. 1479

ISBN 978-91-7519-790-6
ISSN 0345-7524

Printed by: LiU-Tryck, Linköping

Distributed by:

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

Tel. +46 13 281000
http://www.liu.se

Never send a human to do a machine's job
Agent Smith in the film "The Matrix"

ABSTRACT

In the design of complex engineering products it is essential to handle cross-couplings and synergies between subsystems. An emerging technique, which has the potential to considerably improve the design process, is multidisciplinary design optimization (MDO).

MDO requires a concurrent and parametric design framework. Powerful tools in the quest for such frameworks are design automation (DA) and knowledge based engineering (KBE). The knowledge required is captured and stored as rules and facts to finally be triggered upon request. A crucial challenge is how and what type of knowledge should be stored in order to realize generic DA frameworks.

In the endeavor to address the mentioned challenges, this thesis proposes High Level CAD templates (HLCts) for geometry manipulation and High Level Analysis templates (HLAts) for concept evaluations. The proposed methods facilitate modular concept generation and evaluation, where the modules are first assembled and then evaluated automatically. The basics can be compared to parametric LEGO® blocks containing a set of design and analysis parameters. These are produced and stored in databases, giving engineers or a computer agent the possibility to first select and place out the blocks and then modify the shape of the concept parametrically, to finally analyze it. The depicted methods are based on physic-based models, meaning less design space restrictions compared to empirical models.

A consequence of physic-based models is more time-consuming evaluations, reducing the probability of effective implementation in an iterative intensive MDO. To reduce the evaluation time, metamodels are used for faster approximations. Their implementation, however, is not without complications. Acquiring accurate metamodels requires a non-negligible investment in terms of design space samplings. The challenge is to keep the required sampling level as low as possible.

It will be further elaborated that many automated concurrent engineering platforms have failed because of incorrect balance between automation and manual operations. Hence, it is necessary to find an equilibrium that maximizes the efficiency of DA and MDO.

To verify the validity of the presented methods, three application examples are presented and evaluated. These are derived from industry and serve as test cases for the proposed methods.

SAMMANFATTNING

Vid utvecklingen av komplexa och tätt integrerade maskintekniska produkter är det viktigt att hantera gränsöverskridande kopplingar och synergier mellan olika delsystem. En ny teknik, som har potential att drastiskt förbättra konstruktionsprocessen, är multidisciplinär design optimering (MDO).

En MDO process kräver ett integrerat och parametrisk konstruktionsramverk. I detta syfte är design automation (DA) och knowledge based engineering (KBE) lovande tekniker för att stödja parametriska konstruktionsramverk. En avgörande utmaning ligger i hur och vilken typ av kunskap som bör förvaras för att förverkliga en generell DA ramverk.

Därför föreslås high level CAD template (HLCT) för geometri manipulation och high level Analysis template (HLAt) för koncept utvärderingar. Detta gör att användaren kan bygga modeller i mindre moduler som sedan monteras och utvärderas automatiskt. Grunderna kan jämföras med parametriska LEGO ® block som innehåller en uppsättning av design och analys parametrar. Dessa produceras och lagras i databaser, vilket ger ingenjörer eller en datoragent möjligheten att först välja och placera ut blocken och sedan ändra formen på dem parametriskt, för att slutligen analysera produkten. Metoderna är baserade på fysikbaserade modeller, vilket innebär mindre begränsningar jämfört med empiriska modeller.

Nackdelen med fysikbaserade modeller är tidskrävande utvärderingar, vilket gör genomförandet av dem i en iterativintensiv MDO opraktisk. För att minska utvärderingstiden införs metamodeller för snabbare approximationer. Att implementera metamodeller är dock inte utan komplikationer. Metamodeller kräver en icke försumbar investering i form av utvärderingar av fysikbaserade modeller för att nå en acceptabel approximation. Utmaningen är att hålla nivån på antalet iterationer så låg som möjligt.

Det kommer att redogöras att många samtidiga DA plattformar har misslyckats på grund av felaktig uppskattning gällande balansen mellan manuella och automatiserade operationer. Det är ytterst nödvändigt att hitta rätt balans för att maximera effektiviteten av DA och MDO.

För att verifiera giltigheten av de presenterade metoderna används tre applikationsexempel från industrin.

.

ACKNOWLEDGEMENTS

First and foremost I would like to thank all my colleagues at the Division of Machine Design. Working with you for the past 5 years has been a great experience.

Special thanks to my supervisor Prof. Johan Ölvander for all the great support and feedback provided. Your commitment as a supervisor has always been immensely appreciated. My sincere gratitude goes to our former head of division Prof. Petter Krus for believing in our, at the time, unconventional ideas and giving me the opportunity to join the group.

I want to thank my industry supervisor Dr. Xiaolong Feng for all supportive and inspiring discussions over the years. Furthermore, I would like to thank the people involved from ABB Corporate Research and ABB Robotics. Your expertise has been crucial for us to identify the challenges in industrial robot design. I also wish to express my gratitude to ABB for providing research funds for this work.

Dr. Kristian Amadori, it has been a pleasure to discuss modeling methodologies and plan optimization strategies with you. I believe that our cooperation has been rewarding, both for our research and our student course.

I would like to thank all students I have been in contact with during these years. Special thanks to all Product Modeling students, your interested attitude has been an inspiration and an important reason to further develop the course and our design methodologies.

Torbjörn Andersson, thank you for a fascinating illustration for the cover of this thesis.

Last but not least, I would like to express my gratitude to family and friends, especially my beloved Hanna, for their uncompromising support. You have all, in your own way, inspired me to do better.

Mehdi Tarkian

Linköping, September 2012

APPENDED PAPERS

The following papers are appended and will be referred to by their Roman numerals. The papers are printed in their originally published state, except for changes in formatting and correction of minor errata.

[I] Tarkian, M, Ölvander, J, Feng X, Pettersson M, Design Automation of Modular Industrial Robots, Proceedings of the ASME International Design Engineering Technical Conferences & Computers and information in Engineering Conference, San Diego, USA, Sep 2009.

[II] Tarkian, M, Ölvander, J, Feng, X, Pettersson, M, Product Platform Automation for Optimal Configuration of Industrial Robot Families, Proceedings of ICED11: International Conference on Engineering Design, Copenhagen, Denmark, Aug, 2011.

[III] Amadori, K, Tarkian, M, Ölvander, J, Krus, P, Flexible and Robust CAD Models for Design Automation, Advanced Engineering Informatics. 2012;26(2):180-95

[IV] Tarkian, M, Persson, J, Ölvander, J, Feng, X, Multidisciplinary Design Optimization of Modular Industrial Robots by Utilizing High Level CAD templates, Accepted for publication in Journal of Mechanical Design, 2012.

[V] Tarkian, M, Vemula, B, Feng X, Ölvander, J, Metamodel Based Design Automation – Applied on Multidisciplinary Design Optimization of Industrial Robots, Proceedings of the ASME International Design Engineering Technical Conferences & Computers and information in Engineering Conference, Washington, USA, Aug 2012.

The following papers are not included in the thesis but constitute an important part of the background.

[VI] Tarkian, M, Persson, J, Ölvander, J, Feng, X, Multidisciplinary Design Optimization of Modular Industrial Robots, Proceedings of the ASME 2011 International Design Engineering Technical Conferences & Computers and information in Engineering Conference, Washington, USA, Aug-Sep 2011.

[VII] Ölvander J, Tarkian M, Feng X, Multi-objective Optimization of a Family of Industrial Robots, in Multi-objective Evolutionary Optimisation for Product Design and Manufacturing, Wang L., Ng A. H.C., Deb K. (editors), pp. 189-217, Springer 2011.

[VIII] Nezhadali, V, Kayani O, Razzaq, H, Tarkian, M, Evaluation of an Automated Design and Optimization Framework for Modular Robots Using a Physical Prototype, Proceedings of ICED11: International Conference on Engineering Design, Copenhagen, Denmark, Aug 2011.

[IX] Feng, X, Wäppling, D, Andersson, H, Ölvander, J, Tarkian, M, Multi-Objective Optimization in Industrial Robotic Cell Design, Proceedings of the ASME 2010 International Design Engineering Technical Conferences & Computers and information in Engineering Conference, Montreal, Canada, Aug 2010.

[X] Safavi, E, Tarkian, M, Ölvander, J, Rapid Concept Realization for Conceptual Design of Modular Industrial Robots, NordDesign, Gothenburg, Sweden, 2010, Aug 2010.

[XI] Venkata, RCM, Tarkian, M, Jouannet, C, Model Based Aircraft Control System Design and Simulation, 27th Congress of International Council of the Aeronautical Sciences, Nice, France, Sep 2010.

[XII] Tarkian M, Ölvander J, Lundén B, Integration of Parametric CAD and Dynamic Models for Industrial Robot Design and Optimization, Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and information in Engineering Conference, New York, USA, Aug 2008.

[XIII] Tarkian, M, Zaldivar, F, Aircraft Parametric 3D Modeling and Panel Code Analysis for Conceptual Design, 26thCongress of International Council of the Aeronautical Sciences, Anchorage, USA, Sep 2008.

[XIV] Tarkian, M, Ölvander, J, Berry P, Exploring Parametric CAD-models in Aircraft Conceptual Design, 49th AIAA Structures, Structural Dynamics, and Material Conference, Schaumburg, USA, Apr 2008.

ABBREVIATIONS

AAO:	All At Once
AL:	Automation Level
BLISS:	Bi-Level Integrated Synthesis
BR:	Backward Recursive
CAD:	Computer Aided Design
CAE:	Computer Aided Engineering
CFD:	Computational Fluid Dynamics
DA:	Design Automation
DoE:	Design of Experiment
DOF:	Degrees of Freedom
DS:	Direct Sampling
FEM:	Finite Element Method
GA:	Genetic Algorithm
GeA:	Geometry Automation
HLAt:	High Level Analysis template
HLCt:	High Level CAD template
HLP:	High Level Primitive
IDF:	Individual Discipline Feasible
IDS:	Indirect Sampling
KBE:	Knowledge Based Engineering
KBS:	Knowledge Based System
MDO:	Multidisciplinary Optimization
ML:	Multi-Level
NRMSE:	Normalized Root Mean Square Error
PA:	Parametric Associative
SAND:	Simultaneous Analysis and Design
SL:	Single-Level
UML:	Unified Modeling Language

TABLE OF CONTENTS

PART I -

INTRODUCTION

Part I of the thesis presents the research domain of the conducted study. The identified challenges are presented as research questions and the research method is defined. Finally, the outline of the thesis is presented in the last chapter.

Computers are useless. They can only give you answers.
Pablo Picasso

CHAPTER 1

INTRODUCTION

With regard to the tough global market, the struggle between manufacturers is intensifying. It is becoming increasingly crucial to search for and adapt to new means to develop products with less cost and still satisfy customer requirements.

A reliable and steadily growing resource, defying the global economic trend, is computing capacity. In many fields computers and machines have replaced their human counterparts, such as time-consuming numerical processes and routine-like manufacturing processes. Undoubtedly, once a task is fully defined, computers and machines are unparalleled in executing the task repeatedly with great speed and sustained accuracy. To this end, Hopgood (2001) states "*computers have therefore been able to remove the tedium from many tasks that were previously performed manually*". The process referred to is also cited as design automation (DA) by various researchers. The key phrase here is *many* manual tasks have been removed through DA and a natural question would be, why not remove the tedium from *all* manual tasks?

The speed and accuracy of machines has been intensively explored in manufacturing where automation has successfully increased production and quality. Manufacturing automation has been an effective leverage for industrialized countries in response to the cheap labor opportunities in developing countries.

It is, however, important to take note of the recorded drawbacks in manufacturing due to automation. The hard learned lesson in manufacturing is the counter-effectiveness when establishing requirements to fully eradicate humans from the process. These measures have failed because of the principal differences between humans and machines. It can only be concluded that machines cannot replace humans in every task since, unlike humans, machines are not suitable for creative and intuitive tasks. Performing fully defined tasks is the main characteristic of machines. This is the essential source of their productivity and simultaneously the main cause of their inability to adapt to undefined deviations.

Henceforth, machines should be utilized for what they are supposed to do, in every field, including design: repeat fully defined, non-creative and iterative tasks. It has been emphasized that a non-negligible part of design is perceived as routine-like and repetitive by engineers. Automating these tasks will both speed up the design process and free time for engineers for actual creative and intuitive design.

In this thesis various design methods are proposed to automate the *design iteration process* as well as the *modeling process*. The connection between modeling and iteration cannot be ignored. For efficient design iteration, an equally efficient modeling methodology is required. Hence it is believed that these methods cannot be developed independently.

1.1 DESIGN ITERATION CHALLENGES

In the design of complex and tightly integrated mechanical engineering products it is essential to handle cross-couplings and synergies between different subsystems (Bowcutt, 2001). Typical examples of such products could be transportation vehicles like trains, automobiles and aircraft, or mechatronic machines like industrial robots. An emerging technique, which has the potential to drastically improve the design iteration process, is Multidisciplinary Optimization (MDO). Vandenbrande (2006) describes MDO as a "*systematic approach to design space exploration*", the implementation of which allows the designer to map the interdisciplinary relations that exist in a system and automatically search through the design space for optimal solutions.

Multidisciplinary design is an iterative intensive process, due to the intricate couplings between the product disciplines. Naturally, the probability of finding optimal designs increases with the number of design iterations performed. The number of design iterations possible is dependent on the evaluation time. With faster evaluations the possibility to perform more iterations naturally increases.

Evaluation time is lower in early design phases and increases throughout the design process when higher fidelity models are utilized (Ullman, 2010). The drawback with low fidelity models is the inherited uncertainties imbedded in the poor knowledge-bearing models, which results in less appropriate design decisions being made. These decisions are re-evaluated in later design phases when more knowledge becomes available. However, rectifying earlier mistakes is expensive since it involves manipulating higher fidelity models, which requires more manual operations and thus involves more engineers from multiple departments. The involvement of more departments and engineers inevitably leads to less design freedom (Ullman, 2010), see Figure 1.1.

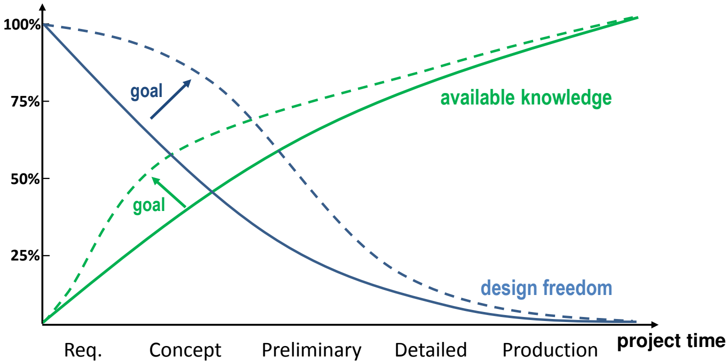

Figure 1.1 Design knowledge and freedom related to design process, adapted from Verhagen et al. (2012)

An improvement of the traditional design processes is to increase the level of knowledge in early design phases, as well as to increase the design freedom in later ones. Increasing model fidelity and introducing holistic design processes will lead to an increase of the knowledge level. However, there are many obstacles before such an approach can be realized. Simpson and Martins (2011) have pointed out several challenges for holistic MDO processes. The manuscript is based on an MDO workshop attended by 48 representatives from academia, industry, and government agencies of various nationalities. In short, Simpson and Martins outline an integrated, parametric, modular and highly reusable design framework with a centralized and parametric geometry model.

1.2 DESIGN MODELING CHALLENGES

There are numerous acknowledged methods depicting how modeling challenges stated by Simpson and Martins can be resolved. Many of the proposed methods are applied on design tools that are not primarily intended for automation purposes, such as computer aided design (CAD) and computer aided engineering (CAE) tools.

It is becoming increasingly common to use geometry from CAD tools as reference input for various types of CAE analyses such as CFD and FEM (Mawhinney, 2005). Nevertheless, the use of CAD as a design aid has been heavily debated over the years. While some have forecasted a more active role for CAD (Larsson, 2001), others state that CAD is more suitable as an automated drafting tool (Ullman, 2010).

Ullman is correct from a historical point of view since CAD was in fact first marketed predominately to reduce the cost of drafting departments (Weisberg, 2010). Creating drafts with CAD significantly decreased the lead-time. However what was first marketed as more cost efficient drafting soon began to give rise to major methodological changes, see Figure 1.2.

The first methodological change was established when drafts began to be generated semi-automatically, based on pre-defined geometry models. As a second step, CAD departments were merged with design and manufacturing departments and design engineers started to work directly in CAD. The next major change came in the late 1980s when parametric associative (PA) CAD was introduced and small geometrical changes were possible by modifying a few parameters. Nevertheless, it was not until the late 1990s that PA modeling began to have a practical methodological impact as update times and errors were considerably reduced due to both significant software and hardware improvements (Cederfeldt, 2007).

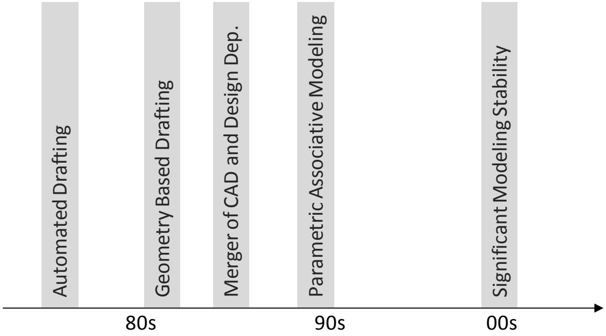

Figure 1.2 Significant CAD milestones in recent decades

Despite substantial hardware and software improvements there is still some hesitancy regarding CAD as a design aid. Ullman (2010) is somewhat correct in the assessment that too much time and detail is required in order to create CAD models. Designers thus become reluctant to abandon poor designs due to the time invested.

1.3 HIGH LEVEL TEMPLATE DRIVEN DESIGN

New methods are required in order to speed up concept generation and evaluation. The goal should be to allow engineers to work on a higher abstraction level where the use of low level and non-creative CAD functions (i.e. points, lines, sweeps and extrusions) during the concept generation and evaluation phase is minimized if not fully eradicated. The same premises holds true for CAE where lower level and non-creative functions such as mesh generation as well as boundary condition and load specifications should require comprehensively less manual operations.

To eliminate the identified non-creative work, methods for creation and automatic generation of High Level templates will be suggested in this thesis. The principles are similar to High Level Primitives (HLP) suggested by La Rocca (2009). The basics can be compared to parametric LEGO® blocks containing a set of design and analysis parameters. These are produced and stored in libraries, giving engineers or a computer agent the possibility to first topologically select the templates and then modify the shape of each template parametrically to finally evaluate the generated system with the analysis parameters.

High Level template driven design is a key enabler for integrated design frameworks such as MDO, where CAD models serve as integrators for other CAE models. Thus, a precondition for MDO is DA framework with parametric capabilities. In conclusion, geometry automation (GeA) is essential for implementation of DA in mechanical engineering design, whereas MDO necessitates DA (Figure 1.3).

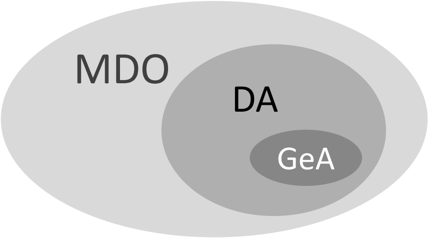

Figure 1.3 MDO necessitates DA that in turn is dependent on GeA

The design methods, which are presented in this work, are implemented and verified in three application examples; conceptual aircraft design, load frame design and multidisciplinary industrial robot design. The industrial robot example has been the main industrial driver for many of the established requirements. Consequently, to fully grasp the challenges of this domain, contemporary industrial robot design is presented in the next chapter.

CHAPTER 2

INDUSTRY AIM

The main application of this work is design and optimization of industrial robots, with a focus on the mechatronic aspects. An industrial robot constitutes a good example of a complex product, as it comprises multiple engineering domains, such as mechanics, electronics, software and control engineering.

Industrial robot design is utilized in this thesis to demonstrate the problems encountered when applying design automation on complex engineering problems. However, most of the methods and tools developed are generic and could be applied to other domains as well, as presented in the application examples in Part IV.

A design scenario for industrial robots is described in this chapter, with the aim to explain some of the existing challenges. Together with the more generic research questions, presented in CHAPTER 3, these challenges are the foundation of the contributed design methods in Part III.

2.1 INDUSTRIAL ROBOTS DESIGN

According to the International Organization for Standardization, industrial robots are "*automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes*" (ISO Standard 8373:1994).

The mechanism of an industrial robot is based upon kinematic chains, called closed or open kinematic, depending on how the chains are connected with respect to each other. If the links connected form at least one loop then the chain is called a closed loop. On the other hand, if the links are connected through only one path then the manipulator has an open kinematic structure and the robot is called a serial manipulator. The industrial robots in focus in this thesis are serial manipulators with rotational joints.

The mechanical structure of a serial industrial robot consists of a base followed by a series of structure links, as visualized in Figure 2.1. The links consist of drive-train components (precision gearing and highly dynamic AC servo motors). Major components of the robot controller are power units, rectifier, transformer, axis computers and a high level computer for motion planning and control.

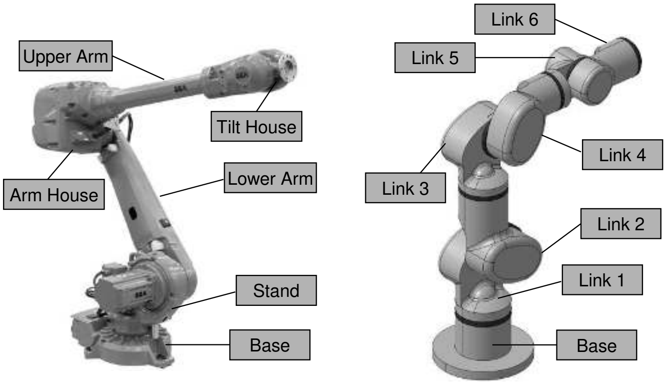

Figure 2.1 A conventional industrial robot (left) and a modular industrial robot (right)

Industrial robots can be described as typical mechatronic systems with complex dependencies between geometry, dynamic performance, structural strength and cost, see Figure 2.2. A characteristic design challenge is the forward and backward dependencies between the various links.

First, the link velocities and accelerations are iteratively computed forward recursively. When the kinematic properties are computed, the force and torque interactions between the links are computed backward recursively from the last to the first link.

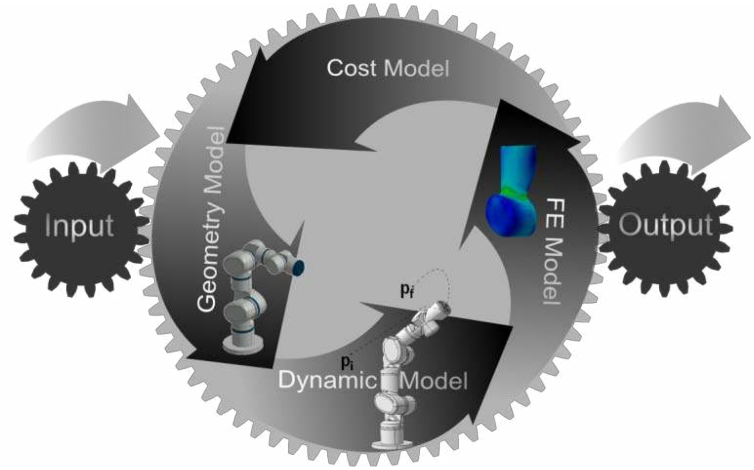

Figure 2.2 Iterative design process between various robot disciplines

In summary, when designing serial mechanical products such as industrial robots, one is to expect that applied changes affect all previous and sequent links simultaneously. A relevant example of such characteristic behavior is the scenario where a drive train is substituted. By changing a drive train a series of actions will be triggered which in turn causes other reactions in what can be perceived as a repetitive loop. The immediate effects of such a change can be described in the following three scenarios:

1. A modified drive train leads to geometry modifications on the attached links, which affects the structural strength. The structural thickness of the attached links has to be modified in order to satisfy the required structural strength limits. This in turn will

affect the mass properties as well as the dynamic behaviors. The effects of structural thickness modifications are not only local but also affect other links.

2. By modifying the drive trains on at least one axis the optimal values of the internal drive train parameters such as maximum velocity and acceleration limits will be outdated and need to be re-calibrated. This will lead to new load cases between the links, which means a repetition of the previously described process.
3. Modified mass properties and load cases leads to the inevitable consequence of all drive trains being re-evaluated and possibly replaced with new ones. There are multiple aspects to take into account such as actuator lifetime and sufficient robot performance properties such as cycle time and tool center acceleration. A possible drive train modification will lead to a repetition of the described process, starting from point 1.

The depicted scenario indicates the intricate dependencies between various domains and the iterative intensive processes required to design such products.

2.1.1 Traditional Design Methods

To further illustrate the present design challenges, a scenario of the mechatronic design phases for industrial robots is presented.

A traditional design process begins with a small number of engineers generating, evaluating, and finally selecting robot concepts fulfilling pre-defined requirements, illustrated in Figure 2.3. In this phase mostly empirical and lower fidelity models are utilized in order to gain speed.

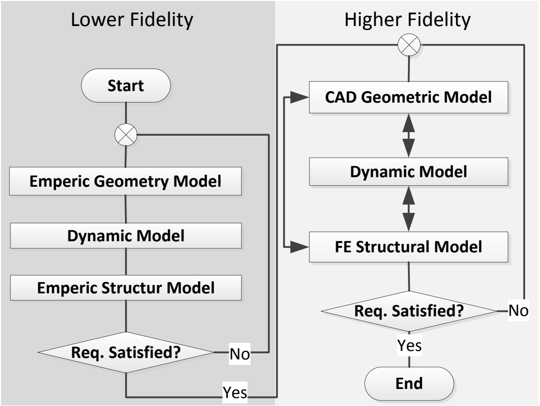

Figure 2.3 A typical manual design approach in robotic design

When the design requirements are satisfied, higher fidelity models are utilized in the following phase. The increased detail level requires the involvement of more departments, resulting not only in an increase in manual operations but also a time-consuming data exchange between the departments. The time-consuming information exchange is due to the intricate communication procedures, where many conflicting objectives have to be discussed during meetings. Ultimately, this results in a lengthy design process.

As illustrated in the previous section, even minor design changes cause multidisciplinary reactions in a time-consuming spiral. Ultimately, the iterative intensive design process together with the current time-consuming evaluations is not a suitable combination for designing optimal products.

2.1.2 Novel Design Methods

To speed up the design process, optimization procedures can be applied to automate the exhaustive manual trial and error process. In the work of Pettersson (2008) comprehensive optimization frameworks are suggested for finding suitable drive trains as well as calibrating the internal drive train parameters simultaneously. Pellicciari et al. (2011) present another approach to optimize the energy consumption of industrial robots without prior knowledge of the actuation system. These results demonstrate the possibility to successfully utilize simulation-based optimization to manage the inherited complexities of the product.

The limitations in these contributions, however, are lack of accurate mass property and structural strength estimations. With limited measures to estimate these properties, optimizing the dynamics will be of limited advantage because of the highly uncertain geometric and structural approximations. The uncertainty can be greatly reduced by implementing higher fidelity models in the optimization process. The mass properties can be generated with CAD models and the required structure thickness can be verified with FE models.

Subsequently, utilizing higher fidelity models will lead to new design challenges. The main drawback associated with these tools is speed and maintainability. Without resolving the disadvantage of slow concept generation and evaluation, higher fidelity models cannot be regarded as realistic and practical optimization enablers in industry.

The obvious design challenge is to enable fast and efficient modeling and evaluation as well as propose optimization strategies that can effectively manage the multidisciplinary nature of industrial robots. In this regard, some of the proposed design methods will be of a generic engineering nature while some will only be applicable to serial industrial robots.

CHAPTER 3

RESEARCH QUESTIONS

The research aim of this dissertation is to present novel design automation methods that are able to manage the encountered difficulties in the design of complex and multidisciplinary engineering products, such as industrial robots. The original research questions are formulated as follows:

RQ1. How to enable multidisciplinary automation and optimization processes for mechanical engineering products?
RQ2. Which types of engineering processes are suitable to automate?
RQ3. How should the identified engineering processes, suitable for automation, become automated?

When trying to address the above research questions, additional questions have naturally emerged. The following research questions have evolved over the course of the research:

RQ4. How to achieve fast design iterations?
RQ5. How to implement the proposed methods to minimize the required changes to the companies’ current design process?
RQ6. How to organize the design automation process to maximize maintainability?
RQ7. Which optimization strategies are suitable for the proposed design automation framework?

CHAPTER 4

RESEARCH METHODS

Reproducibility is an important aspect when presenting new knowledge. It is of importance for scientific progress that other researchers are able to verify the proposed knowledge. By stating the type of research method conducted, the chances for other researchers to reproduce the collected knowledge become more plausible. The degree of reproducibility is debatable for applied research where the premises of the results gathered are based on complex computer modeling. Thus, the possibility of reproducibility by other researchers also depends on the detail and quality of the modeling methodologies reported as well.

4.1 EPISTEMOLOGY PARADIGMS

Four epistemology paradigms are used to present the applied scientific field. The implemented research is a mixture of various epistemologies (Forskningsmetodik, Göteborgs Universitet 2009). These describe different methods to acquire knowledge. The four opposing branches are illustrated in Figure 4.1. The axes of the diagram consist of atomism versus holism and empiricism versus rationalism. First a general description of the scientific paradigms is given followed by short description of how they are applied in the presented work.

4.1.1 Atomism versus Holism

Atomism or reductionism is a philosophical approach that breaks down problems into their smallest components and explains the basis of these problems.

The opposite of atomism can in some cases be regarded as holism, putting weight on the whole. According to the holistic reflection, components alone are unable to describe the wider problem, hence the sum of the whole is greater than its parts.

4.1.2 Empiricism versus Rationalism

Empiricism stresses the value of experience as the only sure source of truth, thus emphasizing the role of evidence gathering through observation. Empirical studies are naturally associated with probabilistic and inductive reasoning, where, given the gathered premises, plausible conclusions are stated.

Rationalism on the other hand argues that knowledge can only be derived from common sense. In contrast to empiricism, the process of attaining knowledge is obtained through deductive derivations, leading to deterministic conclusions.

4.1.3 Epistemology Hybrids

By merging the depicted branches, various hybrid methods are derived. The following scientific methods are then utilized depending on the type of research conducted and the nature of the studied phenomena.

4.1.3.1 Atomistic-Rationalism

In atomistic-rationalism, the problem is broken down into comprehendible elements, clearly quantified through deductive processes.

In engineering this step is performed by delimiting reality to a comprehendible portion, breaking it into sub-systems and constructing models based on mathematics- and physic-based foundations.

4.1.3.2 Holistic-Rationalism

Holistic-rationalism is an approach where knowledge is absolutely and deductively outlined without any subjective measurements.

If taking into consideration the fact that virtual reality is in fact another form of reality, it can be argued that holistic-rationalism can be applied in engineering as well. Here complex system behavior can be simulated on strictly physical and mathematical premises. A system can thus be constructed and its behavior holistically described through deductive and objective measurements.

4.1.3.3 Atomic- Empiricism

In atomistic-empiricism, the problem is yet again broken down into comprehendible elements. In contrast to atomistic-rationalism, the behavior of the elements is quantified through measurements and statistics.

Hence, in engineering, this step can be compared to the procedure of measuring the behavior of a system.

4.1.3.4 Holistic-Empiricism

The core of holism is the premise of not accepting system division into smaller portions. Hence, if the system cannot be divided, it is impossible to understand how it works. However, some holistic behavior may still be noted and understood. Thus, the main purpose of holistic-empiricism is to understand the greater purpose of the system without necessarily being able to describe exactly how it works.

This approach is usually adopted in qualitative research within the humanities, where the complexity of human behavior cannot be deduced nor quantified through empirical measurements, but can still be understood.

4.1.4 Conducted Research Based on the Epistemology Branches

In engineering, single research methods are bound to be impractical. In order to utilize one epistemology branch, another one may be a necessary prerequisite. Thus, without making any empirical observations, requirements cannot be gathered and a model cannot be derived based on a rationale. Similarly, modeling and evaluating a holistic system is unrealistic if not firstly broken down into smaller sub-systems to begin with.

The adopted approach involves three of the above mentioned epistemology branches. The one not utilized is holistic-empiricism, which among other things can be useful when carrying out qualitative observations and gathering data in order to prepare system requirements.

Atomistic-rationalism is adopted to construct mathematics-based models. The models are connected and simulated following a holistic-rationalism approach. The framework is finally verified through atomistic-empiricism, see Figure 4.1.

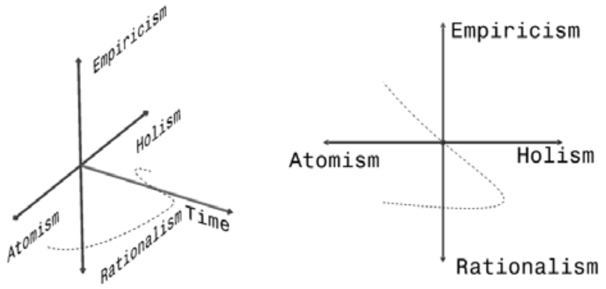

Figure 4.1 A mixture of different scientific procedures are implemented when realizing the presented work

After each emperical study new assumptions are made and the methods are further refined, resulting in new design methods. The iterative process between creation and evaluation continues iteratively as shown in Figure 4.2.

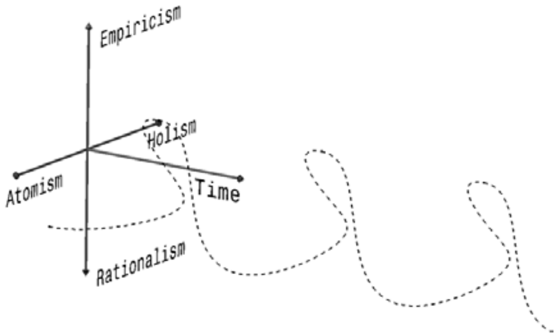

Figure 4.2 The interplay between the epistemology branches over time

4.2 RESEARCH METHODOLOGY

In order to place the conducted research in a wider context, the principles of Roozenburg and Eekels (1995) as well as Blessing and Chakrabarti (2009) are used. Roozenburg and Eekels (1995) propose a scientific approach, where the process consists of the phases *observation*, *induction*, *deduction*, *testing* and *evaluation*, see Figure 4.3.

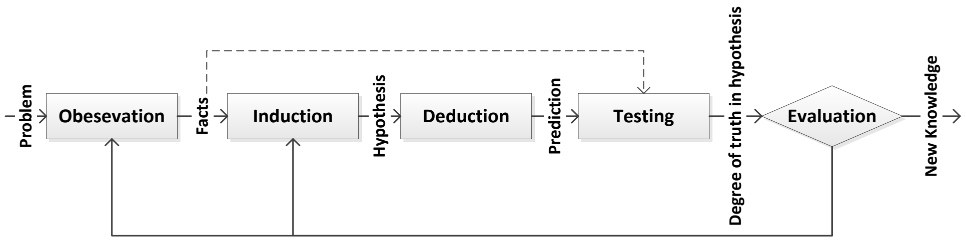

Figure 4.3 Empirical Scientific inquiry (Roozenburg and Eekels, 1995)

The research conducted in this thesis follows the process above, where based on an initial set of problems, facts are gathered through observation, followed by plausible hypothesis through induction. Through deduction, predictions of the reality can be stated. In the next step, predictions are verified by comparing them to previously defined facts. Consequently, when the predictions and the gathered facts are coherent, new knowledge has been gained.

The Design Research Methodology (DRM) suggested by Blessing and Chakrabarti (2009) also follows the same pattern. Their process consists of the four steps *criteria*, *descriptive study I*, *prescriptive study* and *descriptive study II*, as shown in Figure 4.4. The success of the research will be measured in the criteria: in descriptive study I the problem is analyzed; in prescriptive study a solution is proposed; and in descriptive study II the proposed solution is evaluated with respect to the initial criteria.

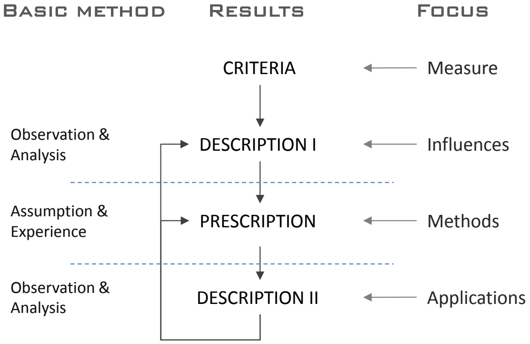

Figure 4.4 Design research methodology framework (Blessing and Chakrabarti, 2009)

4.3 VERIFICATION OF THE RESULTS

According to Buur (1990), a design theory can be verified either by *logical verification* or *verification by acceptance*. Verification by logic implies that the theory should not have any conflicts between internal parts, that it is complete, and in agreement with other theories in the field. Verification by acceptance implies that experienced users in industry or the scientific community should accept the proposed theory.

By applying the proposed methods at three fundamentally different applications it has been verified that the proposed methods are internally consistent and complete and that they are capable of solving specific problems in different fields.

The usefulness of the proposed methods is verified by the acceptance of experienced engineers working in the field. Furthermore, by applying the methods in industrial applications, the ability to address real problems has been established.

Moreover, all appended papers have been subjected to full reviews by other researchers before acceptance, which further establishes the novelty factor of the presented work.

CHAPTER 5

OUTLINE

This thesis is divided into five main parts, the first being the introduction. Each part is further divided into separate chapters.

In the second part, a literature review is presented which points out the state-of-art in multidisciplinary optimization (MDO) and design automation (DA) research. A comprehensive review of Geometry Automation (GeA) is presented to corroborate the current state of this research domain. It is established that Knowledge Based Engineering (KBE) plays a central role for DA and GeA and thus a brief review of KBE is presented. Finally the current state of MDO is elaborated and essential enablers for efficient implementation of MDO are outlined. These methods include metamodeling for faster concept generation, multi-level optimization strategies for complex product management, and various optimization algorithms for different types of engineering problems.

In the third part, the shortcomings in the literature review are addressed and new methods are proposed. For DA, a template-driven modeling methodology is proposed which is able to increase the flexibility in parametric design as well as being better suited for maintenance purposes. For MDO, a novel strategy to perform optimization on serial link manipulators is outlined. Lastly, an enabling method to generate metamodels is presented.

In the fourth part, the proposed methods are implemented and evaluated on applications examples derived from industry to verify the validity of the proposed methods. The applications include industrial robots design, aircraft conceptual design and load frame design for fork lift trucks.

The fifth part concludes the thesis and consists of a discussion, conclusions and directions for future work. Finally, prior to the appended papers, a short summary is provided that not only describes the content of each paper, but also explains how the proposed methodology in this thesis has evolved. The five papers appended describe the proposed methods and their practical implementation in detail.

PART II -

FRAME OF REFERENCE

Part II consists of a comprehensive literature review. Important theoretical concepts are introduced as a frame of reference for the upcoming contributions.

Those who cannot remember the past are condemned to repeat it.
George Santayana

CHAPTER 6

DESIGN AUTOMATION

Design automation (DA) is a field that has held great promise over the past few decades (Tomiyama, 2007). The anticipated benefits of DA have varied from previously "*let the machine do it all*" to more modest expectations. Currently, the expected profits are considerably less ambitious; "*minimize repetitive and non-creative design activities*". Even though automating tedious design processes should improve turnover and increase quality for manufacturing industries, DA is not yet widely employed. Investigating the reasons behind the current situation is thus inevitable.

In a given design project 80% of all manual design activities are routine-like and non-value adding (Stokes, 2001). It should be noted that the presented figure is based upon a somewhat limited number of studies. Nevertheless, Encanação (1990) presents similar figures where 90% of design activities are identified as variant modeling where minor design changes are made to previously established concepts, with limited creative problem solving. Comparable figures are presented for the construction industry by Elfving (2003), where approximately 90% of all design activities are identified as non-value-adding.

To categorize an overwhelming part of engineering design as non-value-adding is debatable and one could just as well argue that 100% of all design operations are in fact necessary and therefore value-adding. The actual definition of the term "non-value-adding" can hence be disputed. Nonetheless, engineering work consists of an array of operations from lower to higher levels of required creativity and ingenuity. A non-negligible part of design requires a lower level of engineering creativity. It has been shown in earlier work that these types of operations are easier to automate and usually with successful outcome, see Brewer (1996), Heinz (1996) and Cooper et al. (2001). By automating these repetitive and time-consuming operations a considerable amount of time can be freed up.

Stokes also points to a couple of industrial achievements when applying DA through Knowledge Based Engineering (KBE) to minimize routine-like tasks. The benefits in cost and lead-time cuts are significant. However, despite the recorded successes, many DA attempts ended in failure (Tomiyama, 2007). The failures could explain why DA is not widely spread in industry. According to an editorial by (Tomiyama, 2007), the main reasons behind failed "*intelligent platforms*" are:

- No room for engineering creativity
- Too small design space because of lack of generality of the model
- Limited maintenance possibilities
- Lack of integration with existing CAD system

Interestingly, most of the reasons for DA failure identified by Tomiyama are recognized by Simpson & Martins (2011) as modeling challenges for successful implementation of MDO frameworks:

- Humans have to be kept in the loop since no synthetic replacement exists
- Allowing the model and/or the set of design variables to be modified in order to allow new regions of the design space to be explored
- A component-based design approach is requested where modules can be reused in different applications
- Consistent geometric model that can be accessed by other analysis tools

6.1 DESIGN AUTOMATION DRAWBACKS

The main reason for DA failures, Tomiyama explains, are that the intelligent platforms try to do too many things at the same time, such as parametric design, optimization, data integrity management, process planning, and synthesis. Many DA attempts have therefore failed when the design platforms lost modeling flexibility by literally growing rigid and consequently limiting engineering freedom and creativity. The main drawback with intelligent platforms is revealed to be the primary intelligence requirement.

The objective of many previous DA attempts was to replace engineers with Intelligent CAD and Artificial Intelligence Systems and the quest to automate the intelligent synthetic part of design proved to be a mission impossible (Tomiyama, 2007). The same outcome has been observed in manufacturing where a fully automated plant does not necessarily equal a more profitable one (Frohm, 2008). Here, machines such as industrial robots are an important part of manufacturing, getting the job done much faster and more accurately than their human counterparts. Nonetheless, not all manufacturing is 100% machine automated.

Human operators are thus responsible for the improvising parts of manufacturing and the routine-like processes are left to the machines (Frohm, 2008). The same strategy should be adopted for DA where engineers should be responsible for the intuitive and intelligence-requiring processes. Bento and Feijó (1997) sum it all up by pointing out the main reasons for the low popularity of intelligent CAD systems has been "*the quest for full design automation rather than providing realistic active support to the design process*".

6.2 REVIVAL OF DESIGN AUTOMATION

According to Danjou and Koehler (2007), the move from 2D to 3D CAD technology during the late 1990s has brought significant potential for accelerated and more cost-efficient product development. This fact, together with the increased interest in DA in late 1990s (Danjou et al, 2008), has resulted in a revival of DA attempts with particular interest in GeA. Today, DA goals are much less ambitious and thus more realistic, which together with unprecedented software and hardware capabilities present a unique window of opportunity.

CAD, being historically thought of a tool to be used in final design phases (Brandt, 1997) is being re-established as a product development tool to be used as early as the preliminary and conceptual design phases (Ledermann, 2005). Ledermann suggests that by implementing parametric associative CAE, the overall design cost and development risks will be lowered. To this end, a new wave of DA and GeA contributions has materialized over the past decade, presenting new approaches to further cut lead-times. In the following sections these contributions are categorized and then presented.

6.3 GEOMETRY AUTOMATION AS A BRANCH OF DESIGN AUTOMATION

One major obstacle as regards to DA research from the late 1990s onward is the lack of a common scientific domain and terminology that hinders collaboration between various research communities. The situation is similar to that of the early 1990s, which Bento and Feijó (1997) described as "t*he absence of a common* terminology *affecting the settlement of a scientific community working on design automation*".

In an effort to identify the ongoing research, a comprehensive literature review has been conducted. Although many manuscripts on DA are available, very little consensus exists as to what GeA actually represents and what the status of the current research domain is and maybe even more importantly, where it is heading.

The collection of relevant manuscripts began with a search using established search engines with appropriate keywords relevant to GeA, such as "parametric CAD", "generative parametric CAD", "associative parametric CAD", "design automation", "geometry automation" and "knowledge based engineering".

Topics with only marginal contribution regarding GeA and engineering were discarded. The collected manuscripts span a wide array of various types of contribution. Therefore, to increase comprehension of the presented contributions, the reviewed manuscripts are categorized as illustrated in Figure 6.1 and listed as follows:

- **Geometry Automation Methods**: A large number of contributions present various types of GeA *methods*. The type of GeA methods is divided in two classes, *fixed* and *dynamic topology*, which will be described later.
- **Modeling Method**: A couple of publications present *modeling methods* in order to realize successful geometry automation.
- **Quality Quantification**: In order to verify the quality of the model, some manuscripts present mathematical quality quantification methods.
- **Specialized CAD**: As will be explained further on, in some manuscripts specialized CAD tools are emphasized to reach successful GeA implementation.
- **Integrated Design Process**: A branch of the reviewed manuscripts examines methods to efficiently integrate CAD and CAE models.

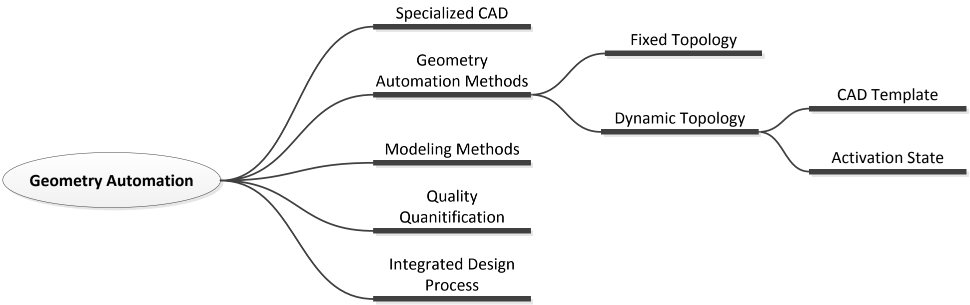

Figure 6.1 Geometry automation manuscripts branched into various categories

6.4 GEOMETRY AUTOMATION METHODS

In this section the collected manuscripts reporting GeA methods are presented. Many of these manuscripts were developed to solve specific application problems. Therefore, the comprehensiveness and generality of the contributions varies depending on the degree of the application focus

According to Sunnersjö et al (2006) the purpose of parametric models is "*to allow reuse of existing design solutions with adaptations to new specifications"*, compatibly adopted in this thesis to describe GeA. To this end, all the collected contributions present GeA frameworks with the main objective of efficiently capturing and parametrically exploring the design space.

Two main types of GeA methodologies have been identified; fixed and dynamic topology. Some manuscripts demonstrate solutions that have fixed topology and only the shape and size of the models vary while others demonstrate models with dynamic geometric features.

The dynamic topology group is further categorized into two domains; *activation state* and *CAD template*. The first manages topological changes by altering the activation state of geometric features. The CAD template domain presents a radically different automation paradigm. Here the models are divided into various templates, stored outside the CAD model, and instantiated automatically following parametric input by the user. By storing the templates and knowhow concerning the instantiation outside the model, the modeling intent becomes less imbedded in the model. As will be outlined later, these features considerably increase the possibility of reuse and automation.

6.4.1 Fixed Topology

Since the launch of the Pro-Engineer CAD tool in the late 1980s there have been aspirations to develop automated geometries representing a wide range of design variants. However, Pro-Engineer was initially an error-prone tool with update times sometimes close to an hour. Thus, the potential to produce complex geometries were limited. This was a major setback for GeA and the benefits were clearly overshadowed by the shortcomings.

From the beginning of this century, GeA has seen a revival with the introduction of tools such as CATIA V5, incorporating KBE techniques and a more reliable and stable geometry engine, resulting in a robust modeling and automation environment.

Previously, many GeA attempts were on mere part level due to poor modeling stability for complex associative products. Hence the work of Myung and Han (2001) was considered groundbreaking, demonstrating that parametric modeling techniques can be useful when frequent design changes take place. This is demonstrated with an expert system for complex associative products. Following a manual assembly of parts, an expert system is able to modify the shape of the parts parametrically. Cederfeldt (2003) proposes a similar methodology called *dimension-driven*. Cederfeldt states that the name dimension-driven indicates that only the geometry dimensions are altered and the topology remains fixed.

Fixed topology GeA is still widely reported in the literature. In order to enhance the capabilities of fixed topology models, new techniques are applied for more drastic geometry alterations. Prasanna (2010) presents generic and unitized parametric sketches. This method is applied in order to implement one algorithm for a family of shapes. Basically, even by maintaining a fixed topology completely different shapes can be generated. Rodríguez and Fernández-Jambrina (2012) present a method called “*programmed design*”, with which a wide range of ship concepts can be generated. The notion of “*one model for all concepts*” is common in this field and many researchers present design frameworks where one flexible and robust model is constructed to represent a wide range of different shapes.

In order to increase reuse in the design process, the use of skeleton models is frequently reported in the literature. The design intent is divided into several parts, where the placement and interfacing features of the parts are stored in the skeleton model(s). Although there exists a common understanding of why skeleton models should be utilized, there is no consistent description of how and of what features a skeleton model should consist of. Neither is the number or hierarchical arrangement of skeleton models standardized

6.4.2 Dynamic Topology

Dynamic topology is an emerging standard in GeA, which is further divided into *activation state* and *CAD template* methods.

6.4.2.1 Activation State

The activation state method is based on the notion of manually modeling a surplus of geometric features for an array of concepts. By using rules, specific features are then activated while others are deactivated upon user-defined parametric input.

Lee and Lou 2002, suggest that in order to take advantage of previous design experiences, assembly configuration models should be constructed where multiple design variants are modeled within a single document. According to the authors this is a convenient way of managing families of models, where components not necessary for a certain configuration are suppressed.

Cederfeldt (2003) identifies this modeling methodology as "*generic modeling*" and states that through activation/deactivation geometries can be regenerated into several design variations. However, it is further outlined that modification of the activation state could lead to model instability if other features utilize the deactivated objects as reference. Using this method therefore increases the risk of modeling errors.

Managing model topology by modifying its activation state is still a popular and well-reported approach. Recent work based on this approach is presented by Lin and Hsu (2008) and Brujic (2010).

The basic principle of the activation state method is very similar to the fixed topology paradigm and top-down modeling and skeleton models are commonly reported to manage the complex associative relations between the parts.

6.4.2.2 CAD Templates

The CAD template paradigm was first recognized as and marketed by the tool vendors as more efficient start models, also called UDFs (user defined functions). By using UDFs ,reuse of design intent between various projects increased.

Later on UDFs became useful for automation purposes as well, where configurator tools could automatically identify the required template from a library and interactively update it with user-defined parameters. Ma et al. (2003) present a framework with object-oriented features consisting of a standard component library for mold design. Users are able to choose templates that are then fully defined with customized user inputs.

Halfawy and Froese (2005) present the paradigm of "*smart objects*", which are "*3D parametric entities*". The framework configures and modifies falsework segments. Although the instances of the smart objects are able to follow user defined paths, they are not context-dependent and the instantiated geometries are based on the original smart object. Full associative modeling is therefore not reached in this work. Similar methods are also presented by Siddique and Boddu (2005).

The next step regarding CAD template modeling is presented by Ledermann et al. (2005). Here, context-dependent instantiation of templates is proposed as an effective way to enhance the design space. Not only the shape of existing geometric objects changes, but new geometric objects can thus also be parametrically instantiated into the product. This approach is expected to be more expensive to set up initially compared to conventional design, see Figure 6.2. Nevertheless, the expenditures are predicted to drop in later design phases due to better modeling flexibility as well as improved decision-making in earlier design phases.

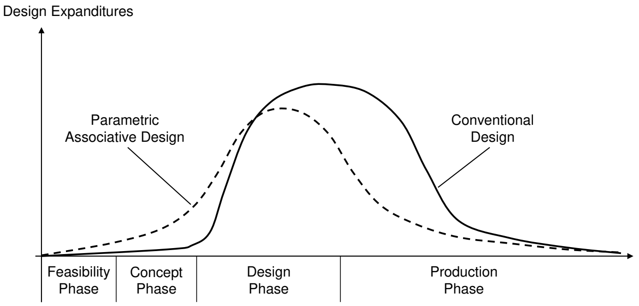

Figure 6.2 Expected variation of design expenditures over various product development phases (Ledermann et al. (2005))

The gravity of this methodology is better appreciated when previous descriptions of parametric modeling are reviewed. According to Vadenbrande et al. (2006) geometric models are required to "*form a continuous function of the input parameters*". A similar explanation regarding parametric modeling is presented by Salehi and McMahon (2011); "*having certain attributes that make modifications possible without deleting and recreating any metrical components*". In this regard, topology parameterization challenges these descriptions by enabling discrete geometric changes.

The framework utilized by Lederman et al. (2005) uses predefined CAD templates called "*Dynamic Objects*", which are automatically inserted and placed in context to generate various repetitive aircraft geometries. Interestingly, Cederfeldt (2003), proposed a similar approach with building blocks parametrically retrieved from a library. The main difference is that the building blocks proposed by Cederfeldt were assembled through Boolean operations and thus not fully associative.

Following Lederman, other research groups have either been influenced by or independently developed their own dynamic topology frameworks. Danjou et al (2008) present an approach based on UDFs as knowledge carriers, where the model is set up by instantiating the UDFs with a few input parameters. Danjou et al. state that since the UDFs are stored in a library, a fast geometric modification is possible to effectively redefine the design space. Thus, even with the proposed highly automated framework, the designer is not subjected to severe limitations.

Raffaeli et al. (2009) present automated configuration of not only detailed CAD models but also technical information such as drafts, BOM, etc. Böhnke et al. (2009) present an object oriented template based framework. UDFs are used as classes, where rules create and instantiate instances of the classes to create complete models. To create associativity, interface class instances are instantiated between the building block instances. Hürlimann et al. (2011) present a multi-layer topology modeling based on UDF libraries. The UDFs are instantiated in several steps to create the complete model, e.g. first wing template are used to create the wings, and then the flaps and slats are instantiated upon the formerly created wings.

The CAD template methodology has evolved since first introduced by Ledermann et al., where geometries were manually instantiated on manually pre-defined master models, while Hürlimann et al., and to a degree Böhnke et al., present multi-layer instantiation where the entire geometry, including the master model, is assembled by CAD templates, as illustrated in Figure 6.3.

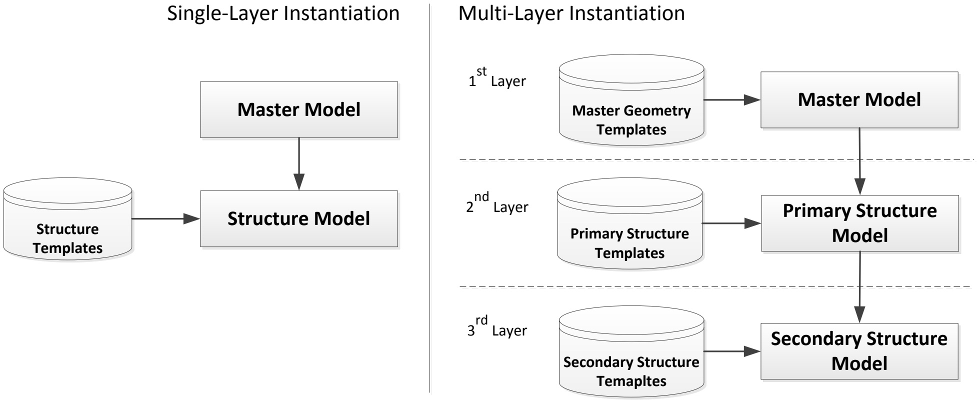

Figure 6.3 The CAD template methodology has evolved to multi-layer instantiations

Finally, a wide range of commercial tools are utilized in the above manuscripts, such as CATIA V5 (2012) from Dassault Systemes, Pro-Engineer (2012) from PTC, NX from Siemens and AutoCAD (2012) from Autodesk.

6.5 SPECIALIZED CAD

An emerging CAD paradigm is specialized and application specific CAD tools. It is argued that such tools can become much faster and easier to operate because they are made for fewer modeling objectives, making them a much lighter version of today's highly general commercial CAD tools.

The aircraft research domain is the most frequently reported field where specialized CAD tools are applied. The dominating research group with the most comprehensive design tools is from Delft University (La Rocca and Van Tooren, 2009). Although the methods presented by La Rocca and Van Tooren are not based on commercial CAD tools, they still constitute a substantial contribution for the GeA community.

La Rocca and Van Tooren suggest high level primitives (HLP) to support designers in order to produce automated aircraft geometries. The modeling methodology bears much resemblance to the work presented by Hürlimann et al. (2011). The approaches are quite similar since the suggested approach by La Rocca and Van Tooren also store all geometries and then use the own developed geometry engine for instantiation. The engineer therefore does not have to do any geometry modeling, only choose the HLPs, which are then assembled together automatically. The presented tool is said to be effective and time-saving when performing MDO, partly because the geometry is produced much faster than with a CAD tool and partly because the pre-processing activities required to feed the various analysis systems in the MDO process can be largely or fully automated. However, there are also some limitations:

- Because of the code-based nature of this application, creating new geometries or increasing fidelity of existing ones is limited and dependent on the code writer. The fact that engineers have to wait for a new HLP will ultimately slow down the design pace.
- When the design is finalized it still needs to be translated to commercial CAD tools to prepare manufacturing support. The same type of modeling therefore needs to be repeated at least once.

Similar, but in many cases less comprehensive, aircraft-specific design frameworks are presented by both Hansen et al. (2008) and Liersch and Hepperle (2009).

6.6 GEOMETRY MODELING METHODS

An effective modeling strategy is essential to decrease the risk of modeling error. Although the essence of well-executed modeling strategies is stressed in many manuscripts, generic and tool independent modeling methodologies are scarce in the literature.

One explanation for the weak interest in generic CAD modeling methods is the dissimilarity between the CAD tools. Generally, a modeling strategy is based on standardized modeling functions, something CAD vendors hold back by not offering similar functionalities. Existing modeling strategies are thus strongly coupled to specific CAD tools.

The few established and generic methods are iterative, meaning no exact guidelines to achieve a perfect CAD model. Amadori (2012) stresses that by creating a geometric model, an iterative and sequential process should be adopted. A concept or test model should be created, evaluated and discarded before the next is started, see Figure 6.4. The model version should be verified against increasingly tough objectives until a crash requires a new version to be produced.

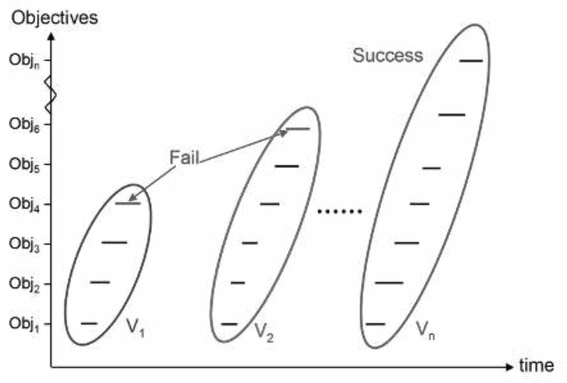

Figure 6.4 Various versions of a model are scrapped until one that fulfills all requirements is found (Amadori, 2012)

Salehi and McMahon (2011) present a modeling approach based on specification, creation and modification phases, see Figure 6.5. Here again, an iterative strategy is adopted and parameters and product associativity are tested and evaluated until all requirements are fulfilled. The modeling method is based on a survey conducted at an automotive supplier. The survey was aimed at establishing engineers' approach regarding associative and parametric CAD modeling.

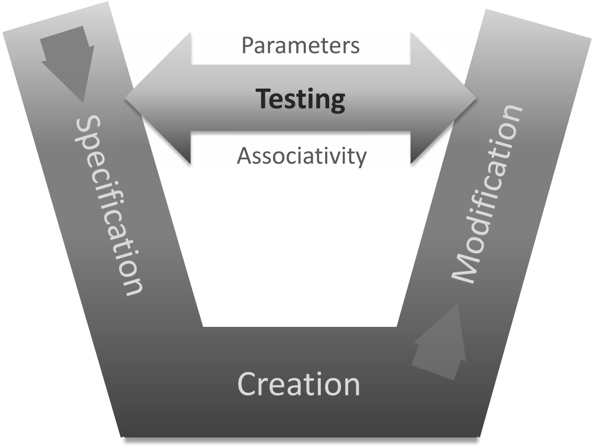

Figure 6.5 Modeling based on specification, creation and modification phases, adapted from Salehi and McMahon (2011)

6.7 GEOMETRY QUALITY QUANTIFICATION

One approach to ensure a reliable model is to measure its quality. This is an alternative approach compared to the previously described modeling methods. Since modeling methods are iterative, quality quantification methods can be used to verify the models following each iteration.

Hoffmann and Kim (2001), raise the issue that in CAD design, parametric models may fail to regenerate. The question of how to identify the parameter values leading to valid models is raised. Hoffmann and Kim present an algorithm that computes valid parametric ranges where the models will regenerate.

Similarly, Brujic et al (2010) present a method to estimate the errorless design range of a model iteratively through:

$$U = U[(1 - x) + 2 * Z * \text{Rnd} \qquad (1)$$

where U is a design variable, Z is a range and Rnd is a random number between 0 and 1. By applying the described method, the authors could define an allowable range for their model.

In Paper III various models are proposed to determine the geometry model's quality and effectiveness. To measure the effectiveness of the models, the terms flexibility and robustness are made quantifiable. Flexibility and robustness are computed and then used to objectively compare different models of the same concept in order to determine which has the highest quality.

6.8 INTEGRATED DESIGN PROCESS

One of the most common procedures utilized to achieve an integrated design process is to incorporate a product master model (MM). The product information in various disciplines on an organizational, managerial and technical level is thus integrated in this model.

Hoffman and Joan-Arinyo (1998), suggest a server-based MM architecture, available for various design disciplines in need of geometric input, see Figure 6.6. Lee (2005) presents a system, which manipulates a single master model containing all of the geometries required for CAD and CAE. Sandberg et al. (2011) suggests MM implementation by using a KBE approach. Here, every change in the geometry model automatically propagates to other model disciplines.

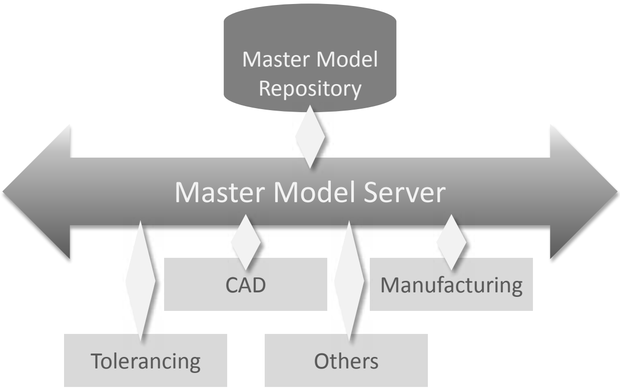

Figure 6.6 Repository distributed master model architecture with clients, adapted from Hoffman and Joan-Arinyo (1998)

Gujarathi and Ma (2011) present CAD/CAE integration by using the "*common data model*", which is a repository of the associative entities of the CAD and CAE models. These act as centralized parametric inputs. Dattoma et al. (2012) present comprehensive parametric and associative geometry modeling and automated mesh generation methods. Structural primitives are prepared, which contain various parameters that can be varied upon instantiation in the product. A mathematical description is utilized in order to assure continuity constraints are fulfilled when the mesh is modified.

A critical question not thoroughly addressed in any of the above manuscripts is the overall organizational challenges MM platforms face. The key challenges have to be recognized, especially for the contributions proposing repository- and server-based MM integration. Such a framework will be subject to constant change by numerous engineers and various departments during the whole design process. The types of challenge that could worsen the performance for such an integrated design framework have to be clarified.

6.9 CURRENT STATE OF GEOMETRY AUTOMATION

As outlined in the previous sections, one of biggest cooperation challenges of DA and GeA is the lack of common terminology. This has resulted in researchers apparently being unaware of contributions that are very similar to their own and the same kinds of GeA methodologies are reported with different terminology, resulting in a waste of valuable resources.

Most of the presented manuscripts in this chapter present various GeA methods. These are either implemented on commercial CAD tools or application-specific tools. Three types of automation methodologies are identified; fixed topology, activation state based topology and CAD template based topology.

Lack of effective methods to achieve flexible and robust model is raised in many of the manuscripts. How to solve this issue, however, is not settled. Some researchers still advocate explicit modeling methodologies to construct errorless models, while others argue that such measures are impractical due to the fundamental modeling differences between the modeling tools. Thus, in some manuscripts, numerical algorithms for quality quantification are suggested as a suitable alternative.

To efficiently extract information from the GeA platforms, a formal design integration method is required, where repository distributed MM is the frequently mentioned.

CHAPTER 7

KNOWLEDGE BASED ENGINEERING

In the previous chapter, existing DA and predominantly GeA methods were presented. Irrespective of the product development methods adopted, knowledge based engineering (KBE) is the only explicit methodology referred to when describing means to achieve GeA by reducing non-creative and repetitive design processes.

The use of KBE is explicitly reported in the majority of the manuscripts presenting a GeA framework. KBE is almost used as a synonym for design reuse and automation, where approximately half of the previously mentioned contributions do not elaborate how KBE has been utilized in their respective design frameworks. This conduct is occasionally taken to extreme levels when KBE is even portrayed incorrectly. The meaning of KBE is ambiguous, since it has been subject to many independent interpretations over the years. It is therefore important to explicitly clarify the meaning of KBE in this thesis.

7.1 DEFINITION OF KNOWLEDGE BASED ENGINEERING

KBE defines a wide range of methods and processes and can be described in several ways, depending on the application in focus. Various definitions can be found that try to highlight the multiple sides of KBE. Chapman and Pinfold (2001) refer to KBE as "*an engineering method that represents a merging of object oriented programming (OOP), artificial intelligence (AI) techniques and computer-aided design technologies, giving benefit to customized or variant design automation solutions*". According to Blount et al. (1995), KBE is a "*true integrator throughout the Product Introduction Process (PIP) supporting the ideas of concurrent engineering*". Furthermore, Verhagen et al. (2011) state that "*one of the hallmarks of the KBE approach is to automate repetitive, non-creative design tasks*", which can lead to "*significant cost savings*" and "*free up time for creativity*".

The meaning of KBE seems to be dynamic and once very tightly integrated with AI techniques is now concentrated on automating non-creative tasks. A contemporary description of KBE adopted in this thesis is:

Automating non-creative design tasks by utilizing object oriented programming.

7.2 KNOWLEDGE BASED SYSTEMS

It is not uncommon for KBE to be compared to knowledge-based systems (KBS) with a focus on engineering (Verhagen et al., 2011). Essentially there are no clear distinctions between the two, but generally KBS is pointed to when describing the system proposed to enable KBE. One of the distinct characteristics of KBS is the separation of the *knowledge base* and the functions, which make use of the knowledge, called the *inference* engine, Figure 7.1. Hence, the principal difference between KBS and conventional programs is the separation of the domain knowledge and the controlling routines, Hopgood (2001).

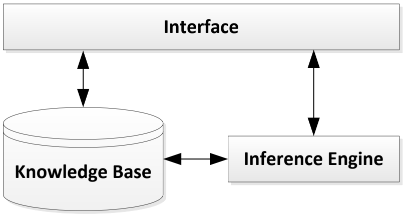

Figure 7.1 A general schematic of a knowledge based system (KBS)

A knowledge base contains explicit and declarative forms of knowledge as sets of rules and facts, enabling easier management. Moreover, the rules and facts do not follow a specific order. The inference engine is the mechanism used to trigger the knowledge stored in a knowledge base.

The separation of the knowledge base and the inference engine is believed to be a key factor for increasing maintainability of DA frameworks. Knowledge rules and facts can thus be continuously modified without affecting the overall performance of the framework. Engineers can therefore maintain the design framework without the system restricting their creativity due to lack of framework flexibility.

To execute the stored knowledge in the correct order, an inference engine is required. The two common types of inference engines are the forward- and backward-chaining.

The backward-chaining inference is goal based and basically finds the rule, which has the end result satisfying the requested action. It then searches backward recursively to end up with the starting rule that needs to be initiated and then executes the relevant rules in sequence.

In forward-chaining inference, the rules that fulfill the stated conditions are found and executed. The process continues by listing new rules that have to be executed and the process continues until the end results are realized. Consequently, the backward-chaining inference is more effective in terms of only executing the rules required to achieve the end result. Nevertheless, this approach can be impractical in cases where sought conditions depend on simulation results. Hence, not knowing which combination would yield the required results will ultimately require every possible combination to be tested first. Given this state, and considering the multidisciplinary nature of the application examples, the forward-chaining inference has been utilized in the proposed GeA framework.

CHAPTER 8

MULTIDISCIPLINARY OPTIMIZATION

A complex engineering product has to be treated as a complete system instead of developing each subsystem independently, Chapman and Pinfold (2001). Hence it is necessary to combine models from several disciplines for concurrent design. Since the subsystems in general have conflicting optimal solutions, a holistic perspective is necessary to reach a balanced global optimal design. Optimizing the subsystems separately would most likely lead to a sub-optimal system. Given this state, MDO has been recognized as a promising method to handle sub-system cross-couplings and treat the product holistically.

Giesing and Barthelemy (1998) have defined MDO as "*a methodology for the design of complex engineering systems and subsystems that coherently exploits the synergism of mutually interacting phenomena*". An MDO process can also capture relations and dependences between different subsystems. Hence, it can be expected that introducing MDO in early design phases will increase the understanding of the system, thus supporting the design process.

Depending on the type of engineering problem, the appropriate optimization algorithm and strategy differ. In the following sections a brief introduction is given concerning the methods, employed in the proposed framework.

In spite of the described benefits of MDO, being a numeric based process it is considered to be an iteration-intensive design method. This can be solved with additional computing capacity or efficient evaluations with metamodels.

The first mentioned is a more direct, brute force, method. Here the model quality is not compromised and faster evaluations are possible merely by enhancing the processing speed, by employing more or faster processors, e.g. by parallel processing. Nevertheless, more speed does not resolve all practical issues when high fidelity models are included in the design process. As discussed in CHAPTER 6, CAD and CAE tools are not traditional DA tools and consequently not frequently used in industry in such frameworks. When CAD and CAE tools are used in a DA-framework, errors might occur, either due to modeling insufficiency or an unexpected run time error. Regardless of the cause, these constitute an undesirable system behavior, which in a worst case scenario drives the optimization routine to inferior solutions. By employing global metamodels, the error-prone tools are excluded from the MDO process.

Metamodel-based MDO is far less computationally expensive. However, compromises are made as regards model accuracy and the size of the design space.

8.1 MDO WITH METAMODELING

Metamodels are numerically efficient surrogates, mimicking the behavior of a system in parts of the design space as precisely as possible (Myers et al. 2009), see Figure 8.1. Common reasons for utilizing metamodels are expansive simulations due to either lengthy virtual computation or laboratory experiments.

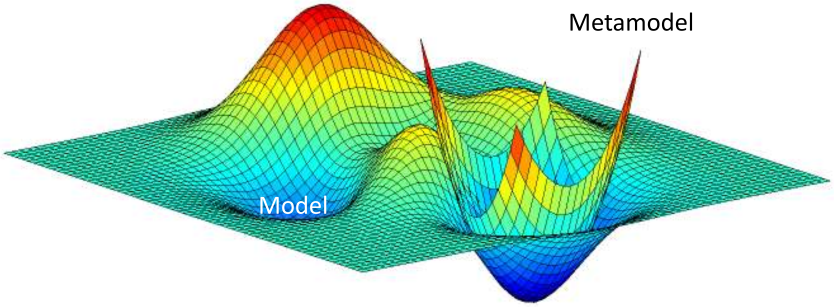

Figure 8.1 Exemplification of a Metamodel

In order to find optimal solutions engineers need to evaluate multiple design alternatives. For practical industrial problems each evaluation may take minutes or even hour to perform, making the implantation of optimization techniques impractical. As a remedy, metamodels are either constructed locally during optimization or in a pre-processing step to mimic the global behavior of the system.

Local metamodels are arguably less computationally expensive since they only approximate the required fragments of the design space during the optimization. Nevertheless, local metamodels entail indirect disadvantages, e.g. they require the complex and occasionally unstable high fidelity models during the MDO process. This detail is important to take into consideration when choosing between local and global metamodeling.

In either case, for an accurate estimation of the system behavior, efficient sampling and metamodeling techniques are necessary. These will be further elaborated in following sections.

The accuracy level of metamodels is dependent on the number and locations of the samples scattered over the design space. Various sampling methods have been developed with varying capabilities. Depending on the type of system behavior and preferred metamodel method, suitable sampling method varies. In the following section, three popular metamodel techniques are described. These are further referred to in the industrial robot application example CHAPTER 12.

8.1.1 Anisotropic kriging

Anisotropic kriging is a modified version of kriging, which originates from the field of geostatistics and calculates the value for a desired point as a function of the distance to known points (Martins and Simpson, 2005). The function is determined by analyzing how the model output varies in the design space. In anisotropic kriging these functions also depend on the variations in the function values for different directions in the design space (Pebesma and Weeseling, 1998). Since the distances to known points are important, the kriging methods benefit from points that are spread over the whole design space. The kriging methods are therefore well suited for the Latin Hypercube method (McKay et al., 1979) used for drawing the samples.

8.1.2 Neural Networks

Artificial Neural Networks (NN) reanimate the way the human brain processes information. They are generally used as black box models for modeling high-dimensional, non-linear data (Myers et al. 2009). An NN consists of several layers - an input layer, an output layer and one or more hidden layers. The variables in the layers are called nodes and a node uses a linear combination of the outputs of the nodes from the previous layer. To use an NN, the weights of those linear combinations need to be determined, which is called training. An NN is a flexible metamodel and consequently can reanimate most systems or models accurately.

8.1.3 Radial Basis Functions

Radial Basis Functions (RBF) are analytical functions whose values depend on the Euclidian distance from the origin (Shan and Wang, 2010). As metamodels they estimate the value of new points by interpolating the values from the previous points. Common choices for RBF for metamodeling are linear, cubic and Gaussian functions (Park and Dang, 2010).

8.2 OPTIMIZATION METHODS

Evaluating different design alternatives for a given design variant or concept can be viewed as one of the less intuition-requiring processes since many repetitions occur. The actual engineering challenge lies in generating novel concepts. Re-evaluating the concept with varying parametric settings requires less engineering intuition and can be regarded as a routine-like process. Therefore, the design trials needed to optimize the concept should be guided by more scientific techniques, viz. optimization algorithms, in contrast to the trial-and-error procedures guided by engineers.

Engineering optimization presents a structured and efficient way of addressing engineering problems (Pettersson, 2008). Optimization algorithms can effectively automate the iterative and time-consuming process of design that involves finding a suitable tradeoff.

Numerical optimization methods could be characterized based on the order of the derivatives used in solving the problem, i.e. second, first or zero order methods, see Figure 8.2. Zero order methods, also referred as non-gradient based methods, do not use any derivatives. Generally speaking, non- gradient, or zero order, methods are applicable to a broader range of problems, as they do not rely on assumptions about the properties of the objective function such as differentiability, and continuity, etc. However they are more computationally expensive than gradient-based methods. In this work three non-gradient methods have been used, viz. a genetic algorithm, Simplex and a discrete version of the Complex method. These are further described in following sections.

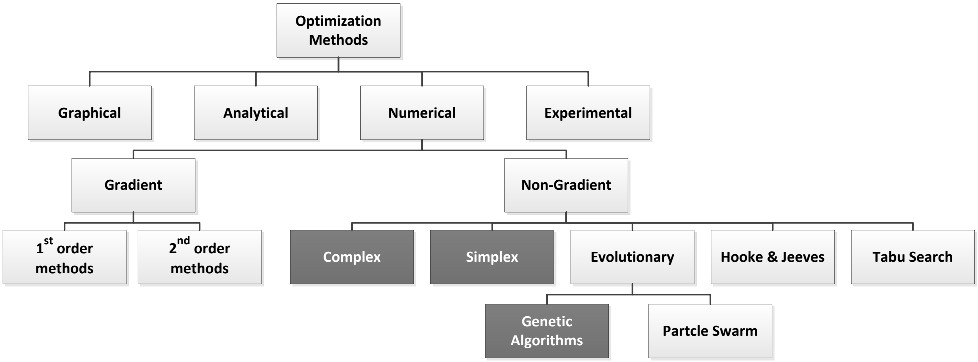

Figure 8.2 Categorization of optimization algorithms, with those utilized in this work colored dark grey

8.2.1 Genetic Algorithms

The basic idea of genetic algorithms is the mechanics of natural selection (Goldberg, 1989). Each optimization variable is coded into a gene, for example as a real number or a string of bits. The corresponding genes for all parameters form a chromosome, which describes each individual. A chromosome could be an array of real numbers, a binary string, or a list of components in a database, all depending on the specific problem. Each individual represents a possible solution, and a set of individuals forms a population. In a population, the fittest individuals have the highest probability of being selected for mating. Mating and thus crossover is achieved by combining genes from different parents to produce a child. Then there is also the possibility that a mutation might occur. Finally the children are inserted into the population to form a new generation.

The presented application examples, in Part IV, contain both discrete and continuous variables, and the objectives and constraints are represented by non-linear functions where no analytical derivatives are available. Genetic Algorithms have therefore been utilized to solve many of the problems addressed in this thesis. Moreover, in the addressed problems there are also multiple objectives that need to be considered, e.g. to identify the trade-off between the weight of the industrial robot and the cycle time. Therefore, the algorithm used should preferably generate a set of Pareto optimal solutions that visualizes the trade-off between the competing objectives. Optimization methods that can handle this type of problem in general are Genetic Algorithms, specifically Multi-Objective Genetic Algorithms (MOGA) (Fonseca and Fleming, 1998).

8.2.2 Simplex

The Simplex algorithm is an iterative non-gradient-based algorithm (Nelder and Mead, 1998). The number of points in the Simplex is *n+1*, where *n* is the number of variables. The starting points are generated using random numbers and the objective function is evaluated at each point. The method progresses by expanding and contracting the Simplex until the optimal solution is found.

8.2.3 Complex

In the Complex method, the word *complex* refers to the geometric shape with k ≥n+1 points in an n-dimensional space. These k points are known as vertices of the complex. The starting points are generated using random numbers and the object function is evaluated at each point. The method progresses by replacing the worst point by a new point obtained by reflecting the worst point through the centroid of the remaining points by a factor α. It has been shown that a useful value of α is 1.3 (Box, 1965). If a point repeats as the lowest value on consecutive trials, it is moved one half the distance towards the centroid. The original complex method has been further developed (Krus and Andersson, 2003), using a randomization factor and a forgetting principle, yielding the Complex- RF method. The Complex-RF method has also been modified in order to handle discrete variables (Pettersson et. al, 2005).

8.3 SINGLE AND MULTI-LEVEL OPTIMIZATION STRATEGIES

Intricate dependencies between various disciplines, together with a large number of optimization variables and constraints, can represent a rather hard optimization problem to solve. It has been identified that multi-level (ML) strategies can efficiently manage optimization of complex engineering products. Unlike single-level (SL) strategies, multi-level strategies are divided into several optimization processes. The engineer is thus able to choose a different optimization algorithm that is most suiting for each particular process.

SL strategies generally have a single optimizer. Yi et al. (2007) showed that the two most efficient SL strategies are individual-discipline-feasible (IDF) and all-at-once (AAO) (Cramer, 1993). The AAO strategy treats the MDO design cycles as a single large optimization problem. Such an approach is also referred to as Simultaneous Analysis and Design (SAND) (Haftka, 1985) in the literature.

ML strategies generally require more evaluations to reach optimum compared to the SL routines (Yi et al., 2008). Nevertheless, by utilizing ML strategies, complex engineering problems can be efficiently managed when divided into hierarchical layers with dedicated optimizers, effectively decreasing the number of design variables, constraints and objective functions for each layer. It is reported that ML strategies are efficient for optimization of complex engineering products as presented by Fujita and Yoshida (2004), Ferguson et al. (2008), McAllister and Simpson (2003) and Venter and Sobieszczanski-Sobieski (2004). In the benchmarking study made by Yi et al. (2008), the ML optimization strategy, Bi-Level Integrated Synthesis (BLISS), see Figure 8.3, required a relatively low number of evaluations in order to find optimal solutions. The BLISS strategy is adapted in CHAPTER 10 for a customized ML strategy suitable for serial link manipulators.

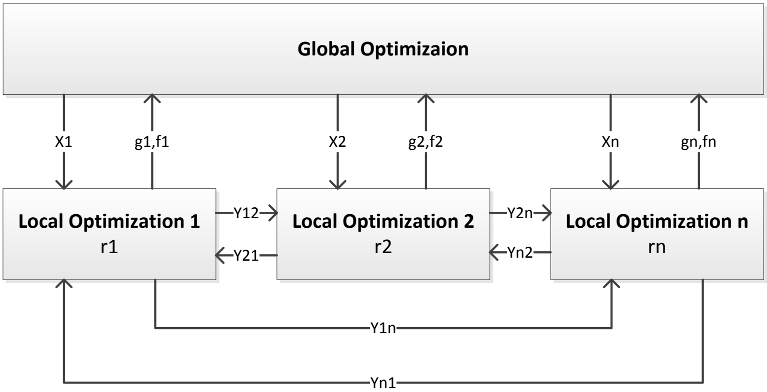

Figure 8.3 General schematics of the BLISS ML strategy

BLISS decomposes the optimization problem into upper system level and lower disciplinary levels (Sobieszczanski-Sobieski and Kodiyalam, 2001). The lower levels contain unique design variables, **r**, while the common variables, **X**, are made into constants. After each completed local optimization cycle, the constraints, **g**, and objectives, **f**, are sent to the system level optimizer. The common variables, **X**, in the local levels are used as design variables in the upper level. Furthermore, coupling variables, **Y**, may be stated between the local optimizers.

In this thesis both SL and ML optimizations are performed and evaluated in CHAPTER 12. For the SL approach the AAO method is used, whereas a novel strategy based on the BLISS is presented in CHAPTER 10 as well as Paper IV.

PART III –

CONTRIBUTIONS

Part III of the thesis presents the theoretical contributions of the research. Based on the literature review, six requirements for successful DA and MDO are established. The following three chapters summarize the theoretical contributions that address the identified requirements.

Don't repeat yourself.
Andy Hunt and Dave Thomas

CHAPTER 9

DESIGN AUTOMATION

State-of-the-art design automation (DA) and multidisciplinary optimization (MDO) contributions have been discussed in the previous chapters. The research challenges and gaps have been raised and formalized in the following requirements:

R1. Only automate non-creative and routine like processes of design
R2. Implement a modular modeling architecture with highly reusable models
R3. Do not limit engineering creativity by allowing users to add, modify and remove models
R4. The modeling methodology should support easy maintenance of the system
R5. Apply suitable optimization techniques, i.e. MDO, to search the design space efficiently
R6. Minimize evaluation time to allow time-efficient optimization processes

Essentially, all of the above requirements have, to some degree, been addressed in the literature. However, none of the manuscripts have fulfilled all of the requirements simultaneously. In this chapter, the first three requirements are addressed, and the latter three are taken into consideration in the following two chapters.

In the following sections an extensive evaluation of current modeling methods is presented and the capabilities of each outlined. Subsequently, how the outlined methods should be realized is described. Finally the most suitable methods, based on the above requirement, are suggested.

9.1 GEOMETRY AUTOMATION LEVELS

Two main geometry automation approaches are presented in the literature review. These approaches are visualized in three automation levels (ALs) based on their associative and transformation capabilities, as shown in Figure 9.1; fixed-topology (AL_1); activation state (AL_2); CAD template (AL_3).

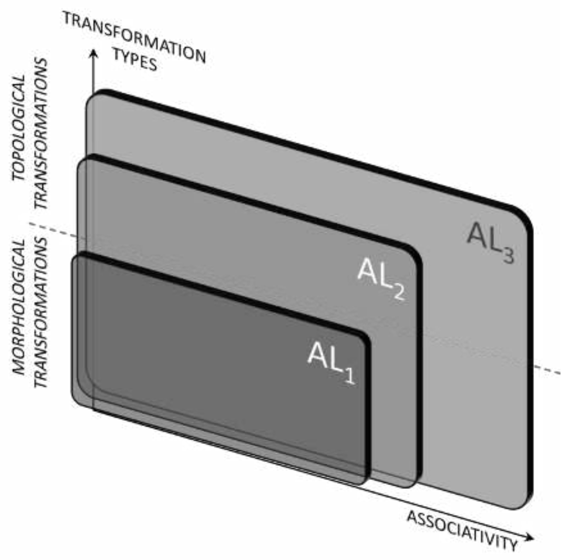

Figure 9.1 Variation of transformation capability together with the model's associativity depending on the automation level (AL)

Basically, geometry transformations are either morphological or topological. Morphological transformations occur within the same instance of a given class, i.e. it is enough to re-evaluate the instance. The transformation degree is increased with topological transformations since the design space can be radically modified when parametrically adding or removing geometric features, each containing a unique set of morphological characteristics.

Associativity is defined as the degree to which a model can handle parametric variations by automatically adapting to new situations. The possibility to alter the model's topology increases the means to adapt to changes. Thus, if a certain part cannot adapt, it can be removed (deactivated) or replaced with another part parametrically, hence increasing the associativity.

9.1.1 Fixed Topology – AL_1

The fixed topology method will be placed in the first level of automation (AL_1). It has been shown in the literature that the method is certainly capable of modeling advanced and complex geometries. However, the fact remains that the design space is restricted to the geometric features modeled and manually assembled beforehand. Also, by increasing the level of geometric features and fidelity, the complexity will grow and likely lead to reduced model flexibility. If the model has to be replaced, then the level of reuse is mainly restricted to what can be copied manually, which can be a demanding task since it requires handling of the intricate knowledge and relations embedded within the model.

When applying KBE to achieve AL_1, the inference engine will be directly connected to the pre-defined model as illustrated in Figure 9.2. As elaborated in CHAPTER 7, KBS enhances the maintainability of the GeA framework because of the separation between the knowledge base and inference engine.

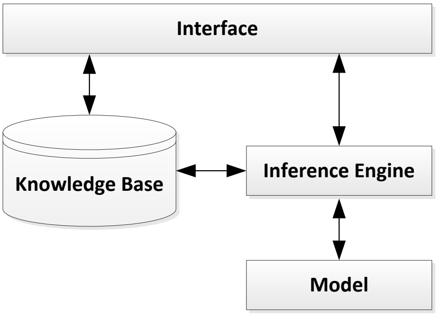

Figure 9.2 By applying KBS, the knowledge base and the inference engine are separated

9.1.2 Dynamic Topology – AL_2

By being able to manipulate the activation state of the geometric features, the transformation level is raised, bringing us to the second level of automation (AL_2). The associativity level is also increased with the possibility to deactivate specific features in case of possible update errors. It is worth mentioning that despite the enhanced capabilities, AL_2 follows the same fundamental KBE principles as AL_1, illustrated in Figure 9.2.

9.1.3 Dynamic Topology – AL_3

Template driven design brings CAD modeling and the KBE paradigm one step closer. Templates, like programming objects, carry processes required to adapt in response to the context they are passed along, according to their inherited properties and behaviors. Being able to remove geometric features parametrically as well as being able to add any number and any type of templates, brings the AL_3 to the highest transformation level. Moreover, the level of associativity is raised with the possibility to adapt to new situations by managing the number and type of geometries parametrically.

The core element of template-driven designs is that they are first produced and stored in libraries, then selected parametrically to be inserted and placed in context automatically. Hence, generating a concept with the depicted procedure does not require lower-level geometric functions such as points, curves and splines. The engineer is thus introduced to higher-level functions. This level of automation, AL_3, is referred to as High Level CAD template (HLCt) modeling, because of enabling a higher-level modeling approach.

By incorporating the HLCt methodology, a fundamental change occurs in the KBS, shown in Figure 9.3. In AL_1 and AL_2, the knowledge regarding the connectivity and relations between various geometries are directly stored in the model. In AL_3, these are stored in the knowledge base, resulting in a more structured way of storing information compared to the complex and intricate approach in CAD tools. Storing know-how structured and externally will therefore increase the likelihood of reuse when the model has to be replaced for any reason.

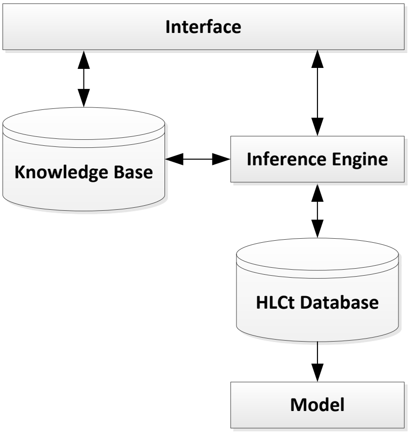

Figure 9.3 Topological variations instantiated from the HLCt database

9.2 PARAMETRIC GEOMETRY TRANSFORMATION

In the previous section the levels of automation were described. Here, methods required to facilitate these automation levels are presented. Figure 9.4 visualizes various types of parameterization required for the different transformation types. As noted previously, geometric transformations are divided into morphology and topology levels.

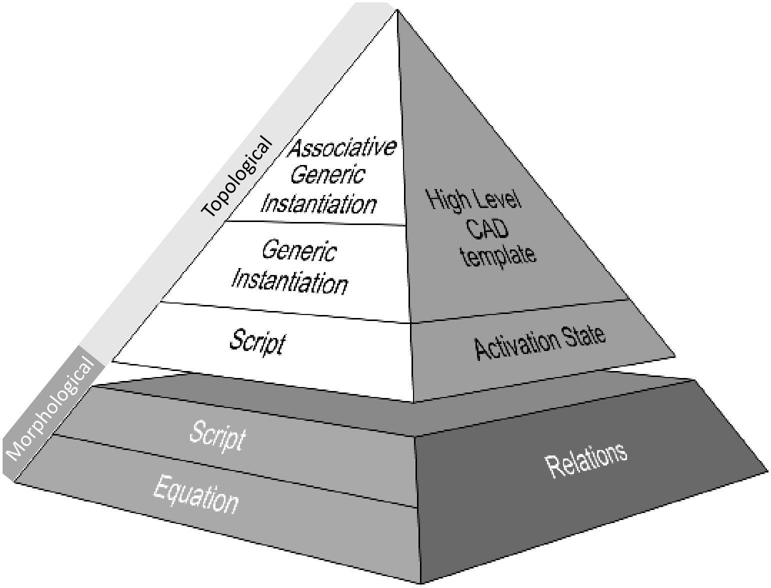

Figure 9.4 The geometric transformation pyramid, categorized in four transformation steps

The morphological and topological transformations can be categorized in the following 4 steps:

T1. Equation Based Relations (morphological): An effective way to decrease the number of input parameters is to set up relations between a model's geometric features. This can be done strictly mathematically, referred here as equation based relations.

T2. Script Based Relations (morphological and topological): With this step, both the morphology and the topology of a model are manageable:
   a) Using various programming languages, either within the CAD system, or in a third party system, relations are created. These are mainly utilized to reduce the number of input parameters as depicted for T1.
   b) Script-based relations allows logic reasoning to further enhance the automation capabilities. With script-based relations it is possible to e.g. modify the activation states of geometric objects and thus achieve topological parameterization.

T3. Generic Instantiation (topology): In this step, HLCts are used. Instantiation is achieved when pre-defined functions can automatically generate or delete instances depending on the user input. The placement and morphology is then changed parametrically.

T4. Associative Generic Instantiation (topology): The principles are similar to T3, however the instance associativity is improved, which is further elaborated in the coming sections.

The major difference between T3 and T4 is how the HLCt is created and instantiated. In T3 the template is instantiated in context by constraining the geometric elements of the instance to the context. In Figure 9.5 an example is visualized where point **p_1** on a rectangle HLCt instance is constrained on **S_{c1}** spline of the context. Following the instantiation, the user can change the parameters in order to fit the instances in-between the two splines by modifying **w_1**, **w_2** and **b_1**, **b_2**.

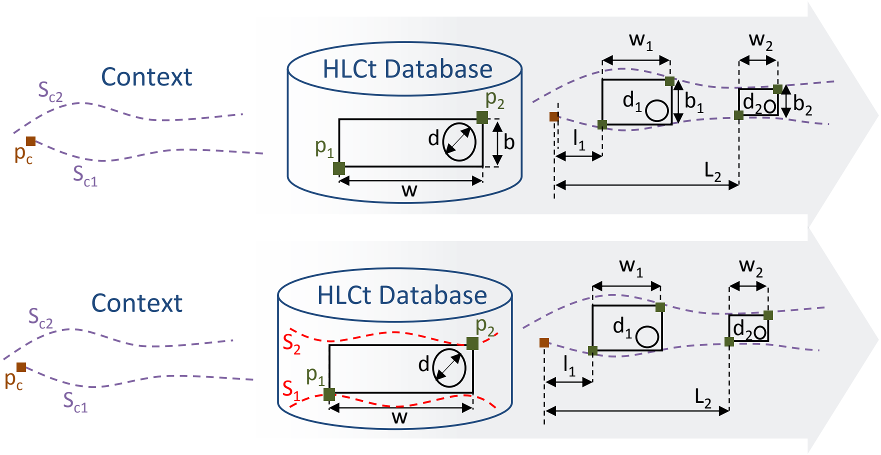

Figure 9.5 Instantiation principles illustrated for T3 (above) and T4 (below)

In T4, the HLCt is modeled according to a specific set of geometric references, illustrated as **S_1** and **S_2** in Figure 9.5. During the instantiation process, the HLCt is instantiated in context by replacing the imbedded references **S_1** and **S_2** with **S_{c1}** and **S_{c2}**. The imbedded references are geometric interfaces, which are replaced with new ones of the same type during instantiation. This process considerably simplifies the instantiation process, since additional constraints are not required after the instantiation.

The given examples are deliberately simplified to highlight the basic principles of the methodologies. Naturally, when applying the methods in a real case scenario, all the features involved in the topological transformation are typically more complex.

9.3 High Level CAD Template Modeling

Traditionally, CAD design has been divided into top-down and bottom-up approaches (Mäntylä, 1990). These design strategies have their roots in software development and can be associated to analysis and synthesis respectively. Wirth (1971) and Mills (1976) were among the first to suggest the adoption of top-down design. Mills (1976) suggests that top-down strategies requires "*thinking and problem solving before integration, rather than afterwards*". Hence, in the top-down approach, the critical information is placed on a hierarchal top level and branches down to all lower component levels in the product. The holistic representation of the product is thus in focus and as a result the complexity is managed and the possibility to revise the product structure and parametrically modify the morphology of the geometry is improved. Conversely, in the bottom-up approach, all base elements are modeled separately in detail and finally assembled into larger sub-assemblies. Since there is no context dependency between the parts, the final geometry may be difficult to modify. This approach is therefore less suitable for design automation purposes.

Although the top-down approach is a well-proven method, it has to be modified in order to enable topological automation of the geometry. In a truly flexible geometry models, the shape, placement and number of the CAD components should be parametrically modifiable. This in turn generates a new means of CAD modeling, referred here as HLCt modeling. The critical information on how the HLCt should be instantiated is stored in the knowledge base and triggered by the inference engine. From an initial model, the user starts by determining the number of instances needed from each HLCt database through a user interface. Various HLCts can be attached dynamically to the model and their shape altered by the inherited design variables.

Clearly, the needed HLCts can be designed already in the early modeling processes. It may thus appear that this approach resembles the bottom-up strategy. The fundamental difference is that the associativity between geometric elements and the hierarchical structure in the model must be carefully planned and stored in the knowledge base before the modeling phase can begin.

9.4 HIGH LEVEL ANALYSIS TEMPLATES

Preparing high fidelity analysis models such as FEM and CFD require a large number of manual operations; the geometries have to be imported into the dedicated tool; the boundary- and load conditions have to be set; and the mesh has to be specified. Preparing the analysis models is a time-consuming procedure. When the process is repeated, it soon becomes well defined and requires only a lower level of engineering creativity to perform, and is hence suitable for automation.

It is therefore suggested that High Level Analysis templates (HLAts) be created and stored in a database for faster concept evaluations (Paper V). The references required to generate the boundary conditions, loads and mesh are all specified in the knowledge base which points to specific geometric references on the HLCt, see Figure 9.6. Hence, the presented approach is derived from the master model (MM) methodology described in CHAPTER 6, with the CAD model having the role as the central integrator.

The references needed to generate an HLAt are stored as geometric features in an HLCt. By storing the name of these references in the knowledge base, the inference engine is able to link the geometric references to various processes:

- Generating the mesh, with pre-specified mesh type and density, by importing geometric references through the function GetMeshRef.
- GetBCRef imports the required references to define the boundary conditions
- GetLoadRef imports the necessary references to establish the applied loads.
- Finally the HLAt is generated and stored in the HLAt database.

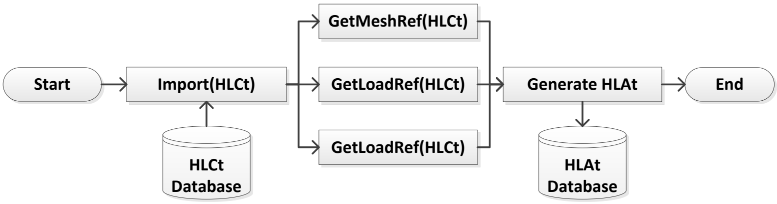

Figure 9.6 Automated generation of HLAts

9.5 PROSPECTS FOR HIGH LEVEL TEMPLATE MODELING

In summary, it has been demonstrated that high level template methods such as HLCt and HLAt are able to further decrease non-creative and routine-like processes, by allocating additional reusable information in the knowledge base as well as geometric and analytical databases.

Furthermore, since the building blocks of this modular framework are stored in databases, engineers can further modify and increase the number of templates without any evident restriction. In summary, only non-creative processes have been automated fulfilling the first requirement (R1). Moreover a modular architecture with highly reusable models is utilized, with no direct limitation on engineering creativity, fulfilling R2 and R3.

MULTIDISCIPLINARY DESIGN PROCESS

In this chapter the fourth and fifth requirements from CHAPTER 9 are addressed:

R4. The modeling methodology should support easy maintenance of the system

R5. Apply suitable optimization techniques, i.e. MDO, to search the design space efficiently

The methods described in the first section concerns the maintenance aspects of the proposed framework. In the second section, the fifth requirement is addressed and a novel optimization strategy for serial manipulators is presented.

10.1 PROPOSED MULTIDISCIPLINARY DESIGN PROCESS

A straightforward approach for achieving a multidisciplinary design process for complex mechanical product is to integrate the various analysis disciplines with an AL_1 or Al_2 type of parametric CAD model. Here the models are firstly created and assembled and then manually integrated. The process in Figure 10.1, illustrates an example of a general and conventional MDO process.

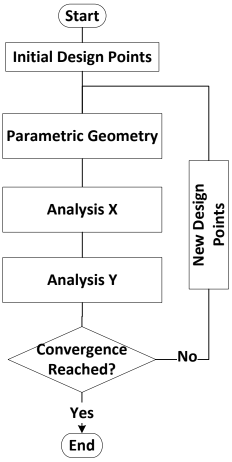

Figure 10.1 A general MDO process

The amount of design reuse possible with such an integrated framework is however restricted to the design space generated by the geometry model. If the geometry model has to be replaced, then the amount of reuse is restricted to the knowledge stored in the knowledge base as described in CHAPTER 6.4. The integration between the CAD and CAE models has to be re-established. Furthermore, many non-creative procedures have to repeated when a new CAE model is constructed as explained in CHAPTER 9.4.

A high-level template approach will have a larger portion of the design intent stored in the knowledge base and the reuse of knowledge will be less dependent on the models and more so on the information stored in the knowledge base. The proposed approach will be divided in the following phases (see Figure 10.2):

- Manual concept selection
- High level template generation
- Metamodel generation
- Automated concept selection

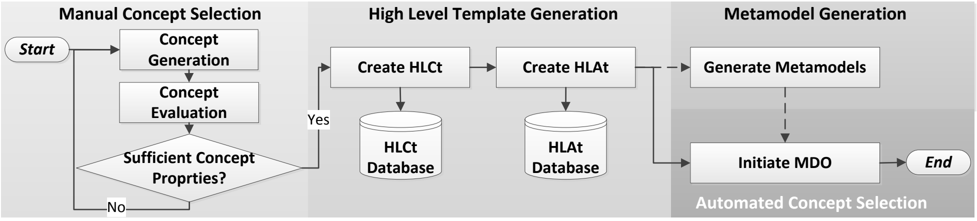

Figure 10.2 The proposed design process, decoupled into four packages

The process begins with a manual design process where a generation, evaluation and concepts selection take place. A concept that indicates promising characteristics is selected.

In the next phase the HLCt and HLAt models are generated. This process is more explained in CHAPTER 9. If the evaluation time is too time-consuming then a metamodel can be generated.

In the final phase, the MDO framework can be setup and initiated. In recent years many design tools, such as modelFRONTIER (2012), iSIGHT (2012) and ModelCenter (2012), have been introduced, greatly enhancing the possibilities to set up MDO processes efficiently. This is realized by using graphical representations of various models and components. Moreover, these tools standardize the pre- and post-processing procedures. This enables an easier collaborative approach compared to previous hard-coded alternatives, implemented in tools such as MATLAB (2012), Excel (2012) or in-house developed tools managed by a few experts.

The direct shortcoming of the presented process is the limited possibilities to treat the concepts holistically in the *manual concept selection* phase, since an integrated multidisciplinary approach is not adapted initially. It should be recapped that the reason for the presented approach is not to propose a radically new design process, but rather one able to automate repetitive and non-creative processes. Starting the design process with a DA framework will severely restrict the creativity of the engineers.

Hence the presented DA approach is only utilized at a time where a good understanding of the concepts is established and the iterative phase to optimize them has initiated. This is the design phase where the substantial portion of non-creative and iterative operations occurs.

10.2 MULTI-LEVEL OPTIMIZATION STRATEGY FOR SERIAL MANIPULATORS

As presented in CHAPTER 8.3, MDO processes are divided into SL and ML strategies, where the former generally requires less iteration to reach an optimal solution. On the other hand, ML strategies are suitable to manage a complex engineering project more efficiently by dividing the disciplines into several domain specific optimization layers.

The ML strategy proposed in this section has been specifically developed for serial manipulators. The improvements compared to conventional SL strategies will be further elaborated in CHAPTER 12.5.2. The main principles of the proposed method are adapted from the BLISS ML strategy depicted in CHAPTER 8.3.

The main motivation behind the proposed method is derived from the Newton-Euler formulation. Therefore, to gain a better understanding of how and why the strategy works, a clear overview of the Newton-Euler formulation is required (Sicilano, 2001). In this formulation, the link velocities and acceleration are iteratively computed, forward recursively.

$$\begin{aligned} a_{e,i} &= (R^{i}_{i-1})a_{e,i} + \alpha_i \times r_{i,i+1} + \omega_i(\omega_i \times r_{i,i+1}) \\ a_{c,i} &= (R^{i}_{i-1})a_{e,i} + \alpha_i \times r_{i,ci} + \omega_i(\omega_i \times r_{i,i+1}) - (R^{i}_{0})^{T}\mathrm{G}_0 \end{aligned} \tag{2}$$

When the kinematic properties are computed, the force and torque interactions between the links are computed backward recursively from the last to the first link.

$$\begin{aligned} f_i &= R_i^{i+1} f_{i+1} + m_i a_{c,i} \\ \tau_i &= R_i^{i+1}\tau_{i+1} - f_i \times r_{i,ci} + (R_i^{i+1} f_{i+1}) \times r_{i+1,ci} + I_i\alpha_i + \omega_i \times (I_i\omega_i) \end{aligned} \tag{3}$$

where ω is the angular velocity, and α angular acceleration, and a_e and a_c describe the acceleration at the end and at the center of each link respectively. The mass of each link is defined as m. f and τ describe the force and torque between each link correspondingly. R is the rotational matrix, I the mass inertia and g_0 the gravity acceleration. $r_{i,i+1}$ is the positional vector from axis$_i$ to axis$_{i+1}$ and $r_{i,ci}$ the central gravity of link$_i$.

The characteristic of computing the force and torque backward recursively enables the optimization of each link to be partitioned in sequential single optimization routines, starting with the last link. This results in the possibility to optimize the weight of each link separately. After completing the first local optimization on link^n, then optimization on link^{n-1} is initiated and the current weight properties, **Y**, are transferred to the subsequent local optimization routine. Compared with the BLISS formulation the coupling variables are transferred single directional and transferred to the next optimization routine as constant, as visualized in Figure 10.3.

The global system level optimizer consists of system variables, **X**, which serve as constraints on the local level optimizers with the optimization variables **r**. The **X** variables can include drive train variables such as type of drive train and maximum velocity and acceleration limits. The local variables **r** include the variables affecting the geometric weight as well as structural stress. Hence, in the local optimization levels the weight (**f**) of the link is minimized without exceeding the structural stress constraints (**g**).

Due to the principle layout of the proposed optimization routine it is henceforth referred as the backward recursive (BR) ML strategy.

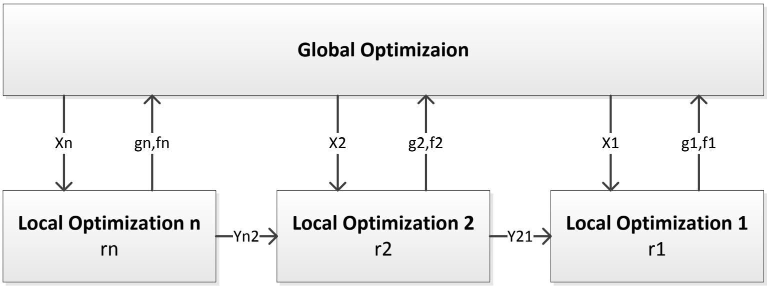

Figure 10.3 The proposed backward recursive multi-level strategy for serial manipulators

CHAPTER 11

METAMODEL SAMPLING

The last remaining requirement (presented in CHAPTER 9) is stated in order to facilitate fast multidisciplinary optimization (MDO):

R6. Minimize evaluation time to allow time-efficient optimization processes

Metamodeling is a widely accepted methodology to reduce evaluation time. However, effective distributions of samples is required to achieve accurate metamodels.

A rule of thumb is that increasing the numbers of design variables give rise to a more complex system behavior, which in turn requires more samples for accurate approximations. For this reason, various sensitivity analysis techniques have been proposed which can determine the significance factor of the design variables through numerical analysis. The variables pointed as considerably less significant can thus be excluded. This will reduce the complexity of the system and result in fewer samples being required for an accurate approximation.

Sensitivity analysis approaches, however, are restricted to the prerequisite that some design variables can in fact be excluded. In many cases this possibility is limited and essentially not practical. In this chapter an alternative sampling approach is presented.

11.1 INDIRECT SAMPLING

A well-distributed set of samples is essential for generating accurate metamodels. The sampling procedure is generally quite straightforward where the samples are generated based on the number and range of the input variables specified by the user. The generated Design of Experiments (DoEs) are then evaluated and an array of output is generated before a metamodel can be generated, see Figure 11.1.

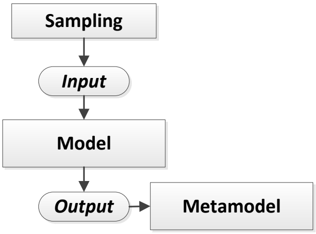

Figure 11.1 A typical sampling procedure, utilized to generate metamodels

In the literature, there are many sampling methods available that generate well-distributed DoEs (Myers et al. 2009). A perquisite for a correct representation of the design space, however, is the user-specified input range. This is usually not a difficult task, but becomes more complicated when the inputs depend on the outputs generated by other models, as shown in Figure 11.2, and the difficulty increases with the number of preceding models in sequence. As the number of previous models increases, the more unpredictable the input range of the last model in the sequence will be. For the design of a complex mechatronic product, such as an industrial robot, there may be many different models involved, e.g. the FE-model (which calculates the stress) will need input from both the Geometric CAD model and a dynamic simulation (e.g. the load).

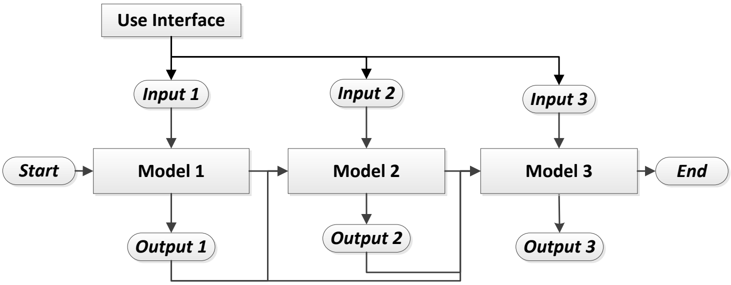

Figure 11.2 An integrated design process

A common praxis is to first identify the minimum and maximum output values generated by the previous models and then apply these values as input ranges, as illustrated to the left in Figure 11.3. This sampling method will henceforth be referred to as direct sampling (DS). The disadvantage here is that in the obtained design space, a large portion of the design alternatives will never occur in practice, as they are not physically feasible.

An alternative approach to the DS method is to generate the samples based on the input ranges of the previously analyzed models, thus having full control of the size and shape of the design space, see Figure 11.3 right. This sampling method will be referred to as indirect sampling (IDS).

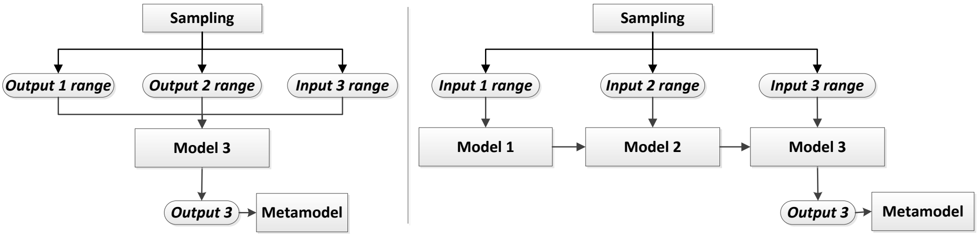

Figure 11.3 Direct sampling (left) and indirect sampling (right)

Comparing the samples generated with both methods for the industrial robot application is visualized in Figure 11.4. In this 2D visualization of the design space, it is clear that the IDS approach is able to capture the actual shape of the design space, while DS generates well-distributed DoEs within the input ranges. Hence IDS will yield a more effective use of the computational resources.

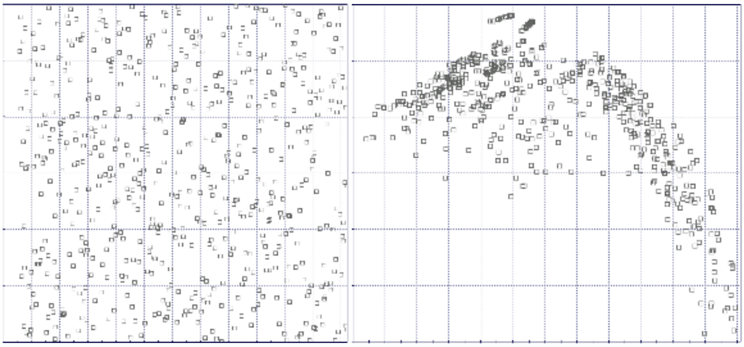

Figure 11.4 Comparison of the DoEs generated with DS (left) and IDS (right)

However, the samples generated with IDS are not evenly distributed. To remedy this shortcoming, IDS has to be complemented with a clustering algorithm to remove samples that are too close to each other. This sampling technique has been applied to the industrial robot application as will be described in CHAPTER 12.4, and further details can be found in Paper V.

PART IV –

APPLICATION EXAMPLES

In this section three application examples are presented to verify the previously presented contributions. With the results derived in this chapter the established hypothesis can finally be verified.

In the first example, a MDO framework for industrial robots will be presented. Here, the entire range of the presented methods is utilized.

In the second example, a multidisciplinary design framework for aircraft conceptual design will be presented.

In the final example, a design automation framework for a load frame configurator will be presented.

Design is to design a design to produce a design.
John Heskett

MULTIDISCIPLINARY OPTIMIZATION OF INDUSTRIAL ROBOTS

To improve the current Industrial robot design process, the methods proposed in Part III will be employed in the following sections. The presented results will support the verification of the modeling framework presented in Part IV.

Designing an industrial robot is a complex process involving tremendous modeling and simulation efforts. Major steps in robot manipulator design are; kinematics design, dynamics design, thermal design, and stiffness design, see Figure 12.1. In addition, the design of a robot manipulator is an iterative process due to the following complex issues: serial connection of robot links, configuration dependent robot performance, multiple domain nature of the robot system including mechanical, electrical, software, and control sub-systems.

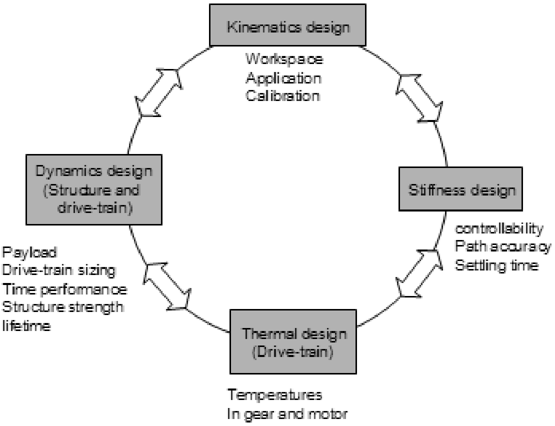

Figure 12.1 Workflow for industrial robot design process

The work presented in this thesis focuses on Kinematics and Dynamics design, and hence performance characteristics such as the size and shape of the workspace, the payload capacity, the weight and cost of the manipulator, structural stress, and lifetime of drivetrain components are of vital importance. The design variables at this stage are parameters such as choice of actuators, acceleration and speed limits for the motion, and geometric parameters for the links such as shape, length, thinness etc. A design automation (DA) framework has been developed to enable multi-disciplinary optimization (MDO) comprising the following main components:

- The CAD model forming the base for all geometric data needed for dynamics kinematics and FE analysis.
- The dynamic simulation model, which calculates the dynamic performance of the robots, the lifetime of drive train components, and the forces and torques acting on the structural elements.
- The FE-model which calculates the stress in each of the structural parts of the robot.

In order to set up this MDO framework the proposed methods have been utilized in the following manner; efficient generation of geometric models based on HLCts (Paper I and Paper III), automation of stress analysis utilizing HLAts (Paper V), evaluation and implementation of efficient metamodeling techniques (Paper IV and Paper V), and finally implementation and execution of an efficient multi-level MDO framework (Paper IV). Each of these steps will be described in the following sections. Only the most significant contributions of the appended papers are included in this chapter.

12.1 DYNAMIC MODEL

The objective of performing dynamic simulation of a robot is to evaluate system performance, such as predicting acceleration, time performance, loads on each actuated axis, and actuator lifetime. The dynamic model is developed in Dymola (Figure 12.2). The model includes a 7-axis robot arm based on the Modelica Standard library (Elmqvist et al., 1998).

Figure 12.2 The dynamic model of a 7-axis robot

The first component, a *trajectory planner*, computes the trajectory for the robot in joint space. The joint space trajectory is then sent to the *drive train* consisting of electrical and mechanical models of the motors and gears. The output from the drive train component is then used to generate motion on the rigid body model, containing mechanical structure of the links.

Each axis is actuated by a drive train comprising a controller and an actuator. The drive train also includes weight properties since the weight of the drive train changes for different designs, which is important to consider for the robot performance. To facilitate comparison of different conceptual designs, the model also includes a simplified trajectory planner and controller in order to be able to run complete robot motion simulations.

The estimated lifetime of each actuator is computed in the drive train components. The lifetime is dependent on the rated performance values of the actuator as well as the actual values in terms of torque and speed during the motion. The definition of lifetime is:

$$L_{10h} = K \frac{N_0}{N_m} \left(\frac{T_0}{T_m} \right)^{\frac{10}{3}} \tag{4}$$

Where L_{10h} is the predicted lifetime in hours and K is the converting factor. N_0 is the rated output speed and T_0 is the rated output torque. N_m is the average output speed and T_m is the average output torque for a specific robot cycle. T_m is the average torque of the actuator and is calculated as:

$$T_m = \left(\frac{\int n(t) \cdot T(t)^{\frac{10}{3}} dt}{\int n(t) dt} \right)^{\frac{3}{10}} \qquad (5)$$

Where *n(t)* is the actual time varying gearbox speed during a robot cycle, and *T(t)* is the corresponding gearbox torque. The dynamic model calculates these critical characteristics in line with the simulation for every concept evaluated and for each motion simulation.

12.2 AUTOMATED DESIGN GENERATION

The robot geometry is constructed with pre-saved building blocks, stored in an HLCt database (see CHAPTER 9.3). The names of the reference for each HLCt instantiation are stored in the knowledge base, which is searched through by the inference engine before instantiation. In Paper V, pseudocode examples describing how the references are retrieved and stored in the knowledge base, are presented.

The process starts with the user defining the number of Degrees of Freedom (DOF) of the robot (see Figure 12.3) and is repeated until the number of axes (i) is equal to the user defined DOF and all structure HLCts are instantiated. In the second stage, all drive train components are retrieved and instantiated in locations pre-defined in the knowledge base. Lastly, the robot structure's internal parameters are morphologically modified to fit the selected components.

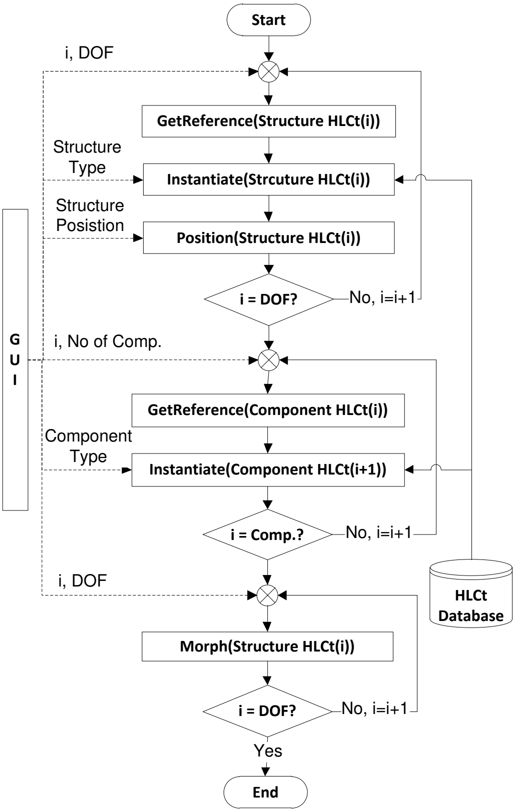

Figure 12.3 The HLCt instantiation process of the industrial robot

12.3 AUTOMATED DESIGN EVALUATION

The proposed HLAt process (CHAPTER 9.4) is adopted for the automated FE process, which begins with the user choosing the HLCt to be inserted into the FEM tool. The mesh, boundary and load conditions are imported from the knowledge base. When the selected geometric elements have been inserted into the FE-model, the mesh is created and the analysis performed.

Figure 12.4 Generating High Level Analysis templates

GetLoadRef imports the flange faces where the actuators are mounted as load references, as shown in Figure 12.4. Since the stress concentration will be highest in these areas, the mesh is pre-set to be denser, defined through GetMeshRef.

12.4 GEOMETRY AND STRUCTURAL METAMODELS

Figure 12.5 illustrates the dependencies between various disciplines involved in industrial robot design. The geometry model, HLCts, provides the analyses tools with geometric input. The dynamic model requires the mass properties. The FE model, HLAts, requires 3D geometry of the HLCts, before receiving the forces and torques interactions from the dynamic model.

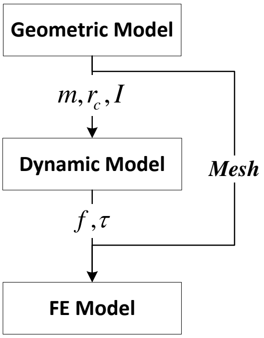

Figure 12.5 Multidisciplinary dependencies between various disciplines

Considering that each discipline is fairly time-consuming, the total optimization time may amount to more than a month for thousands of evaluations. This shortcoming is addressed by applying metamodels for the FE and the CAD models.

To accomplish this, appropriate metamodeling methods are required. In the following sections a couple of different metamodeling methods will be evaluated to identify the most suiting ones. Since the input range of the FE model is dependent on the outputs of the geometric and dynamic models, the IDS will be utilized to define the design space and thus reducing the number of samples required.

12.4.1 IDS Sampling

When creating the metamodels for the FE models, the straightforward DS strategy is to sample the τ and the f variables within the maximum and minimum values obtained from the dynamic simulations. With DS, many of the sampled configurations are not likely to occur in reality, as they are bad representations of the τ and the f, which are not an outcome of the dynamic simulation.

With the IDS approach (CHAPTER 11.1), the gear (G), motor (M), thickness (t), angular velocity (ω) and angular acceleration (α) variables are sampled and the torque (τ) and force (f) are generated as output from the dynamic model as shown in Figure 12.6.

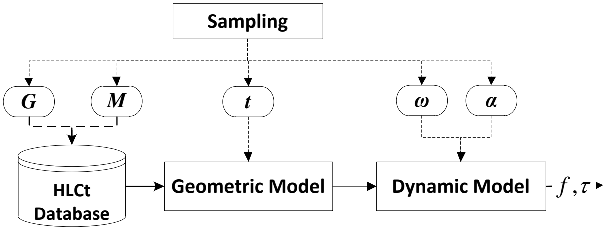

Figure 12.6 Utilizing IDS to sample force and toque variables

Further investigation of the τ and the f relationships in the dynamic model and the generated IDS suggests that many of the load parameters follow the same pattern, see Figure 12.7.

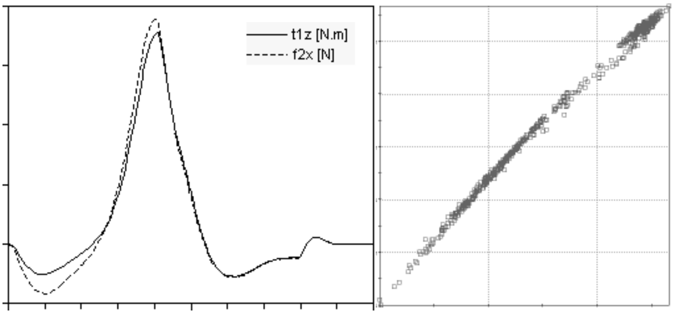

Figure 12.7 Relation between t_{1z} and f_{2x} during a dynamic simulation cycle (left) and 600 IDS of the maximum t_{1z} and f_{2x} values (right)

This finding is not unexpected since the load parameters are coupled both backward and forward recursively according to the Newton-Euler formulation. It is therefore possible to exclude a total of 6 input parameters, which follow the same pattern, when creating the metamodels. The total number of input parameters can be decreased to 9 compared to the original 15.

The outlined assumptions are verified when FE metamodels are created for each arm and benchmarked by generating a set of 50 new samples. As can be seen in Figure 12, the Normalized Root Mean Square Error (NRMSE) for 600 samples with IDS is far superior to that achieved with DS, which has an NRMSE of ~40%. Moreover, the NRMSE decreases even further when the number of input variables is decreased to 9, see Figure 12.8.

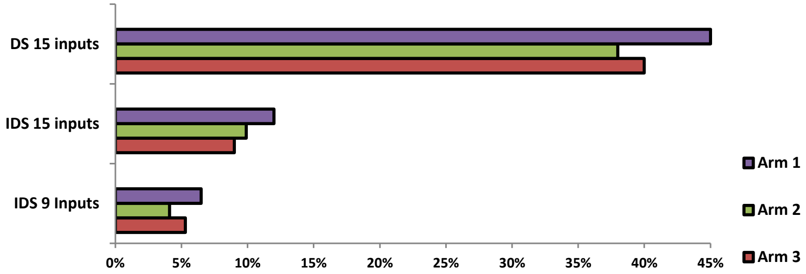

Figure 12.8 NRMSE of the DS and IDS methods for three different arms

12.4.2 Suitable Metamodels

To identify the most suitable type of metamodel for the outlined problem, a range of metamodels types are created and evaluated using 50 samples. The precision of each metamodel is compared with the values of the original model with 20 new samples. The comparison is made using the Relative Average Absolute Error (RAAE) and Relative Maximum Absolute Error (RMAE) as specified by Shan and Wang (2011), as well as the NRMSE, calculated as seen in (6). All precision metrics are desired to be as low as possible, since low values mean that the metamodel is accurate.

$$NRSME = \sqrt{\frac{\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{y_i}\right)^2}{n}} \qquad (6)$$

The resulting precision metrics can be seen in Paper IV and the general conclusion is that Anisotropic Kriging, Neural Networks and Radial Basis Functions are the most promising metamodels (see CHAPTER 8.1). To investigate the impact of increasing numbers of samples, additional metamodels of these three are fitted using 100 samples. The results have been compiled in Paper IV. The resulting NRMSEs for 50 and 100 samples for Anisotropic Kriging, Neural Networks and Radial Basis Functions can be seen in Figure 12.9. The figures inside the parentheses indicate the number of samples used to fit the metamodels.

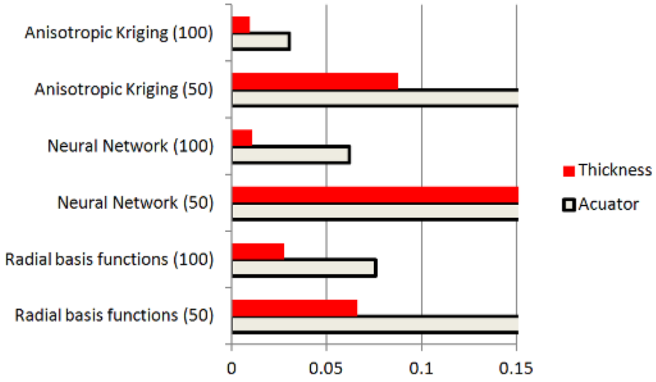

Figure 12.9 Graph of the NRMSEs for different metamodels, fitted using 50 and 100 samples.

According to Figure 12.9 Anisotropic Kriging outperforms the other metamodels and the doubling of the number of samples used for fitting the metamodel increase the precision dramatically.

The input variables for the geometric metamodels are the morphological variables *thickness* and *link height* as well as a topological variable *actuator type*. The outputs of the metamodels are mass *m*, Inertia *I,* and center of gravity $r_{i,ci}$,

For generating a FE metamodel, anisotropic kriging is also proven to be the most accurate method. Here, one metamodel is created for each link of the robot. Inputs are thickness, actuators, force and torque. The output for each metamodel is maximum stress (MS).

12.5 AUTOMATED DESIGN SELECTION

The design objective here is to search for optimal actuators by taking mainly cost and performance into account. The performance objective is cycle time (*CT*), while cost is represented by the total weight (*W*) of the robot. Weight is a reasonable cost estimator based on the rationale that increased material cost for the robot structure and bigger and heavier actuators lead to a higher cost.

In this design study, the kinematic structure of the robot is predetermined, defining a small robot with a payload of 5 kg and representative duty cycles. The following variables are varied during the optimization, as visualized in Figure 12.10:

1. Type of actuators (ac_i) for each axis, i, represented by discrete decision variables.
2. Acceleration (α_i) and velocity (ω_i) limits for each axis i (continuous variables), which influence the cycle time but also the load on the actuators.
3. Wall thickness (t_i) for each link i (continuous variables). Note that *t* is a driving variable which defines only the minimum thickness of the links, i.e. actuator attachments parts are thicker compared to other link parts.

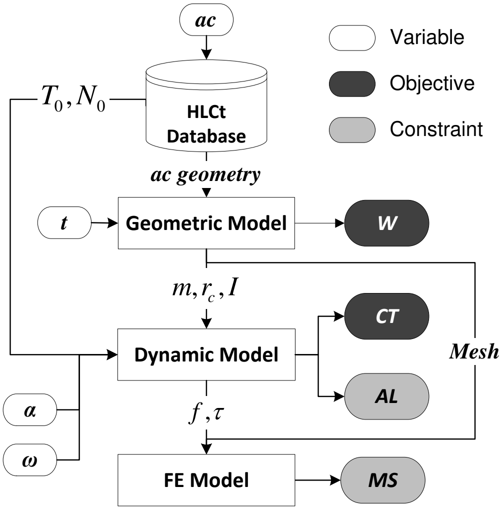

Figure 12.10 Relationships between variables (*ac*, *α* and *ω* & *t*), objectives (*W* & *CT*) and constraints (*AL* & *MS*) in the integrated design framework.

As can be seen from Figure 12.10, intricate dependencies exist between the various disciplines, which are further explained below:

The thickness variable *t* has strong couplings with all objectives and constraints. The actuator type (*ac*) affects the mass properties of the robot. This will in turn affect all objectives and constraints. It is worth noting that the actuator life (*AL*) is influenced both because of the modifications of mass properties and the fact that each *ac* has unique rated torque (T_0) and speed (N_0) values, which affects *AL*, see equation (4).

The α and *ω* variables for a specific axis affects the *CT*. However *AL* for all axes is also influenced. This is due to the backward and forward dependencies of the links and both nominal torques (T_m) and speeds (N_m) are affected, see equations (2)-(5). The maximum stresses (*MS*) of the links are also affected since the force and torque interactions vary with the α and ω limits of each axis.

To conclude, the properties of the robot are tightly coupled, which further strengthens the necessity for a holistic MDO approach. Trying to search through the design space in any other fashion would be an overwhelming task.

12.5.1 Single-level strategy

In the SL strategy (see CHAPTER 8.3), the objective is to minimize the cycle time and weight while ensuring a prescribed lifetime and maximum allowed structural stress, see equations (4). The optimization problem considers all seven axes of the robot and hence the problem is formulated as:

$$\begin{aligned}
&\min f_1(\mathbf{x}) = CT(\mathbf{x}) \\
&\min f_2(\mathbf{x}) = \lambda_1 \sum W_i\ (\mathbf{x}) \\
&g_a(\mathbf{x}) = \frac{AL_a(\mathbf{x})}{AL_{req}} - 1 \le 0, \quad \mathrm{a} = 1:7 \\
&g_s(\mathbf{x}) = \frac{MS_l(\mathbf{x})}{MS_{req}} - 1 \le 0, \quad \mathrm{s} = 8:13 \\
&x_i \in \{1, 2, \ldots, n_{servo}\}, i = 1:7 \\
&x_j^{low} \le x_j \le x_j^{up}, j = 8:27
\end{aligned} \tag{7}$$

The optimization variables are 7 discrete parameters of actuators, 27 continuous parameters for *t*, and α and *ω* limits. The required *AL* is set to 20,000 hours and *MS* limit to 70 MPa. The available number of servo actuators (n_{servo}) is 30 for each axis.

Because of the mixture of continuous and discrete variables, as well as the non-linear functions, MOGA is chosen as the optimization algorithm.

Following the initial population generation, the process starts by calling the geometric metamodels to generate the needed outputs to evaluate the static performance of the robot, see Figure 12.11.

Static simulation includes a range of robot workspace positions, evaluating whether the chosen actuators are strong enough to withstand the gravitational forces. If the configuration does not meet the gravitational loads, then the performance objective is given a penalty value and the dynamic simulation will not be initiated, thus reducing the computational burden.

If the static simulation is successful, the equation of motion for the robot is then calculated using the dynamic model. For the motion performance, each robot configuration is evaluated using a set of representative duty cycles and the objectives and constraints are calculated based on simulation of these cycles. The cycle time is obtained by summarizing the cycle time for all cycles, whereas the *AL* constraints are evaluated for each cycle separately.

Following the motion simulation, maximum stress values are computed using the FE metamodels for all links. The objectives and constraints are evaluated and if convergence is not reached then a new population is generated.

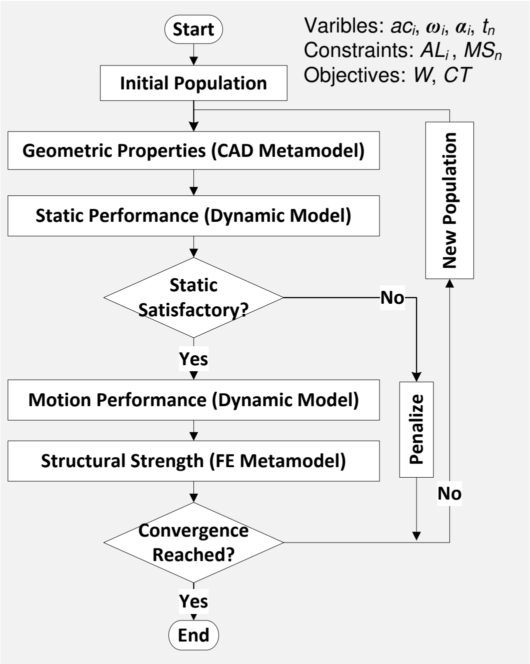

Figure 12.11 The SL optimization process.

12.5.2 Multi-level strategy

The BR ML strategy (see CHAPTER 10.2) consists of one global and six local optimizations, see Figure 12.12. The global procedure contains 21 common variables (*ac*, α and *ω*) and 7 *AL* constraints. The objective once again is to minimize *CT* and *W*. The optimization problem for the global level can be outlined as:

$$
\begin{aligned}
&\min f_1(\mathbf{x}) = CT(\mathbf{x}) \\
&\min f_2(\mathbf{x}) = \sum f_y(\mathrm{t}) \\
&g_a(\mathbf{x}) = \frac{AL_a(\mathbf{x})}{AL_{req}} - 1 \le 0, \quad \mathrm{a} = 1:7 \qquad\qquad (8) \\
&x_i \in \{1,2,\ldots,n_{servo}\}, i = 1:7 \\
&x_j^{low} \le x_j \le x_j^{up}, j = 8:21
\end{aligned}
$$

The weight of each link is then optimized in six single objective routines with the unique single design variable, thickness. Each local optimization has one *MS* limit to satisfy and the common variables of the global level are set as constraints. The optimization algorithm used for the thickness optimization is Nelder-Mead Simplex (CHAPTER 8.2.2).

$$
\begin{aligned}
&\min f_y(\mathrm{t}) = W_i\ (\mathrm{t}), \quad \mathrm{y} = 3:8 \\
&g_s(\mathrm{t}) = \frac{MS_j(\mathrm{t})}{MS_{req}} - 1 \le 0, \quad \mathrm{s} = 8:13 \qquad\qquad (9) \\
&\mathrm{t}_j^{low} \le \mathrm{t}_\mathrm{j} \le \mathrm{t}_j^{up}, j = 1:6
\end{aligned}
$$

The optimization of each link is partitioned in six sequential single optimization routines. It is important to note that the proposed decoupling will not lead to a sub-optimized system. The lighter link_n becomes without violating the stress constraint, the less torque and force interaction with $\mathrm{link}_{\mathrm{n-1}}$

is to be expected. Convergence is quickly reached and the dynamic model is called only once prior to each local optimization, see Figure 12.12.

The optimization process starts by the global optimization routine selecting an initial population, see Figure 12.12. The global layer, which uses the MOGA algorithm (CHAPTER 8.2.1), has the same process flow as the *SL* strategy until the completion of the motion simulation.

Prior to the first local optimization, the torque and force interaction of the last link (Link_n) and the payload are computed by evaluating the dynamic model, see Figure 12.12. The local optimization routines are then initiated. The FE metamodel is called to analyze the structural strength for a set of load cases and the geometric metamodel computes the new mass properties. After convergence is reached, the dynamic model finds new force and torque interactions between Link_n and Link_{n-1} taking the new mass properties of Link_n into consideration. This process then continues to the first link.

When the local optimizations are finalized, the dynamic model is called one last time and all constraint and objective values are updated. If convergence is reached then the process ends, otherwise a new population is generated.

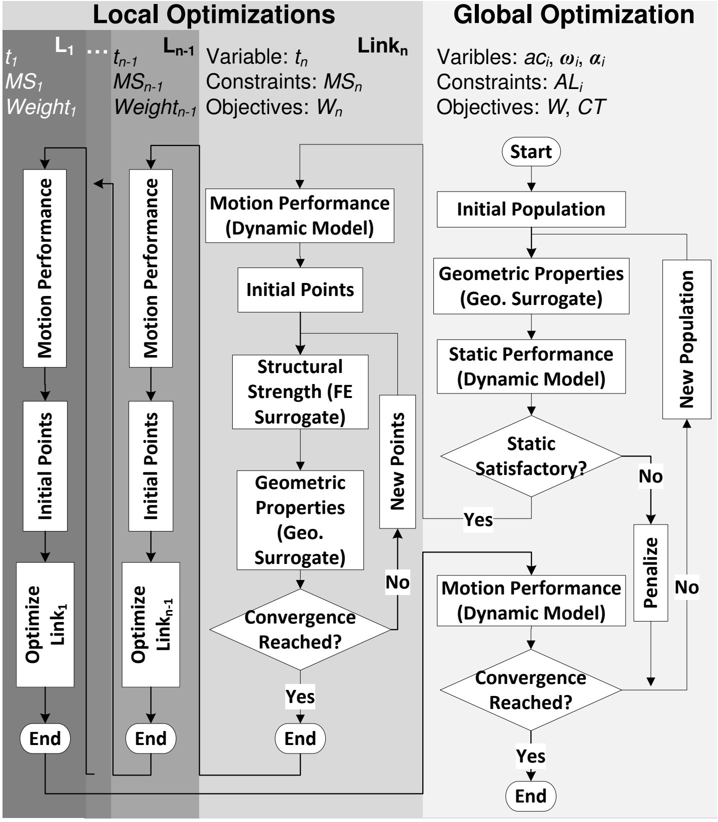

Figure 12.12 The ML optimization process with one global and six local optimizations in sequence.

12.5.3 Results

Population size for the BR ML approach has been calibrated to 70 and the Pareto frontiers for 130 generations are shown in Figure 12.13.

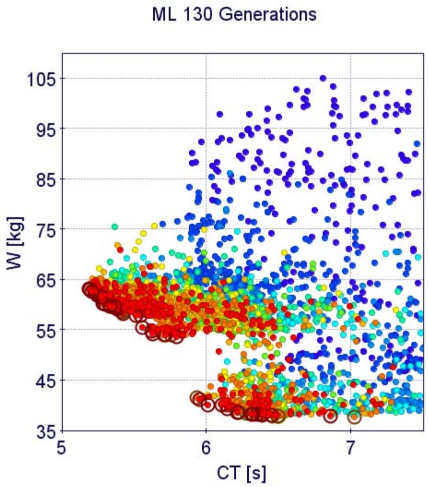

Figure 12.13 Pareto frontier (marked) for the ML strategy

The population size of the SL strategy has been calibrated to 150. The SL strategy runs 270 generations in order to reach the same CPU usage as the ML. The obtained Pareto frontiers are visualized in Figure 12.4.

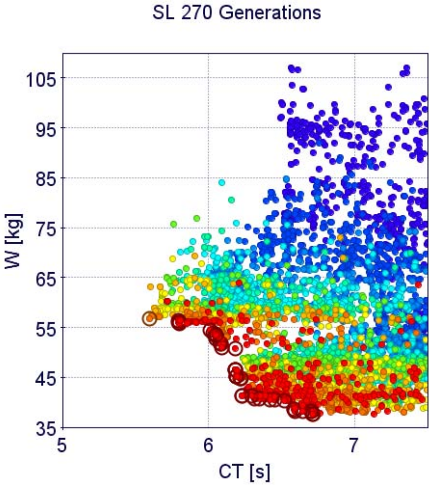

Figure 12.14 Pareto frontier (marked) for the SL strategy

The 1st order Pareto frontiers of both strategies are visualized in Figure 12.15. As can be seen, the BR ML strategy is superior at finding more optimal solutions. The SL routine is still inferior when running up to 540 generations, using twice the amount of CPU compared to the ML strategy.

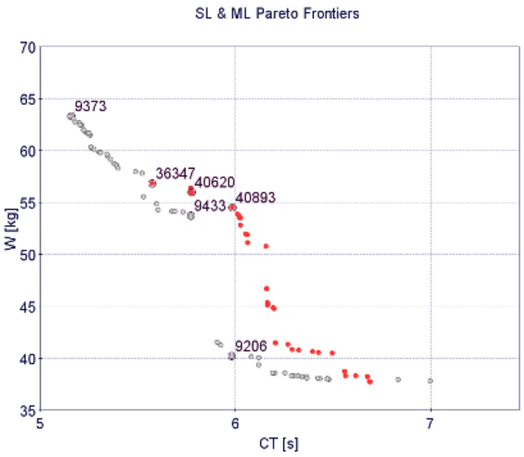

Figure 12.15 Pareto frontiers of the SL (red) and the ML (grey) strategies.

The better result of the BR ML strategy is explained by the fact that the thickness of each evaluated robot is optimal because of the local optimization routines. The global optimization has therefore only to search for optimal actuators, speed and acceleration limits. To exemplify this, the marked design points in Figure 12.15 will be further investigated and the corresponding robot geometries visualized in Figure 12.16.

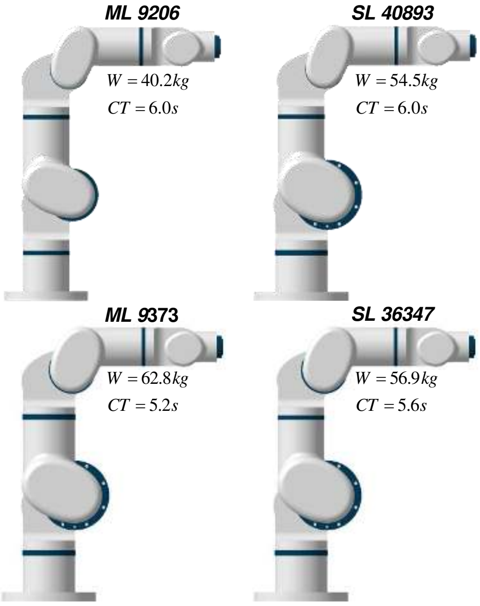

Figure 12.16 Optimal robot variants from the ML and SL Pareto frontiers.

Design points *9206* from the BR ML strategy and *40893* from the SL strategy have approximately the same CT at 6 seconds. However, the mass of each robot differs considerably. This has to do with the fact that heavier types of actuators are chosen for axes 1 and 2 for variant *40893*, see Figure 12.16. The variables in the BR ML strategy have been chosen such that both stress and life time constraints are very close to their limits, in contrast to the ones in the SL strategy. Ultimately, the thickness of each link for robot *9206* has been reduced to the point where it paves the way for a smaller type of actuator on axis 2, without violating any of the constraints. A smaller actuator on axis 2 has in turn made it possible to also have a lighter actuator on axis 1.

Variants *9373* and *36347* both represent the fastest CT design alternatives for the BR *ML* and *SL* routines respectively. The main difference here is that robot *9373* has a heavier actuator type on axis 3 compared to *36347*, which allows it to perform cycles faster without violating the actuator life time constraint.

Variants *9433* and *40620* have the same actuator types on all axes. Robot *9433*, however, is lighter, which is due to the slightly smaller thickness values for each link, which in turn are due to the local optimization routines.

CHAPTER 13

MULTIDISCIPLINARY AIRCRAFT DESIGN

In the field of transport aircraft design, as with other complex products, automatic geometry generation represents an important enabler to achieve MDO. The geometry model, presented in this section, has the central master model (MM) role for other analyses tools in the framework by providing required geometry data. In the presented example, the design framework of which a wide spectrum of different aircraft configurations can be generated will firstly be presented, followed by the multidisciplinary design framework involving aerodynamic and structural analysis.

Development of the aircraft model began in late 2006 and the latest version (Paper III) discussed in these pages is an evolution of the work by Tarkian and Tessier (2008) and Amadori et al. (2007). The CAD and the FE models are developed in CATIA V5 and the aerodynamic analysis is performed in PANAIR (Epton, 1981).

13.1 AUTOMATED DESIGN GENERATION

The aircraft model is generated in a multi-layered instantiation approach (CHAPTER 6.4.2.2) to facilitate a wide range of different concepts, see Figure 13.1. The knowledge-based system (KBS) is capable of representing several different types of configurations, including less conventional blended wing bodied aircraft.

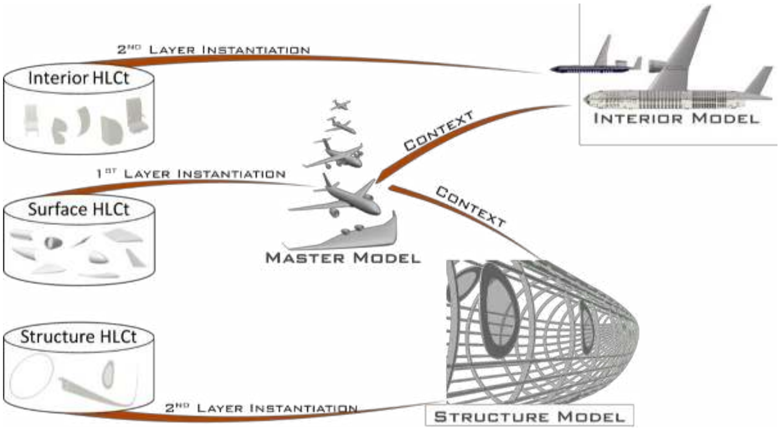

Figure 13.1 The aircraft geometry instantiation sequence is multi-layered

The power of the presented approach derives from the possibility to parametrically determine the number, placement and size of the geometrical elements existing in the framework, see Figure 13.1. The system is then able to automatically create an aircraft model that incorporates all the entered choices. Through the same interface the user can also control detailed structure and interior models. The KBS also includes a dedicated meshing tool that prepares and formats the aero-data for aerodynamic analysis tools, see Figure 13.2. The load data from the aerodynamic analysis is then sent to an FE model, for further structural analysis.

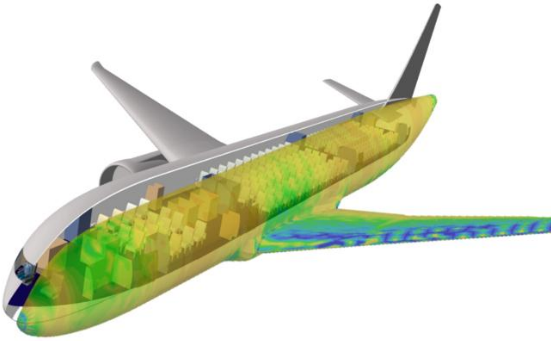

Figure 13.2 A dual view of the interior configuration and the distributed aerodynamic loads

In the aircraft model, the reference model is placed at the highest hierarchical level. It contains references (i.e. planes, points, lines and surfaces) that are required to control and instantiate all remaining models.

The instantiation procedure begins by identifying the HLCts for creating the overall aircraft MM, which consists of both surface and datum features. The surface and datum HLCts are instantiated in distinct sub-products to define a clear associative data flow. Surface HLCts include geometries such as fuselage and wing sections, and horizontal and vertical tail sections. The HLCts contain contextual geometries, making it possible for the inference engine to assemble them correctly. For instance, the interface geometries necessary to connect the mid-fuselage section to the fore and aft fuselage sections are *end-planes* and *contours* of the cross-sections.

When the aircraft surfaces are complete, the structure and interior HLCts can be instantiated and then positioned as illustrated in Figure 13.1. To instantiate these, the inference engine locates context geometries stored in the knowledge base. As an example, for a fuselage frame instance, it is necessary to identify the fuselage surface and the plane defining the frame’s location.

13.2 AUTOMATED DESIGN EVALUATION

Aircraft are true multidisciplinary products. Modifying the external shape of the aircraft influences multiple disciplines such as the aerodynamics and the structural aspect. The objectives of the involved disciplines might very well be conflicting. In this regard, a leaner wing profile might be aerodynamically advantageous, but will likely cause higher stress on the aircraft structure. An iterative interaction arises between these disciplines considering that the aerodynamic loads generated affect the structural stress and strain, which in turn affect the displacements of the geometries. The displacements are then feedback through the geometry model to the aerodynamic model, as illustrated in Figure 13.3.

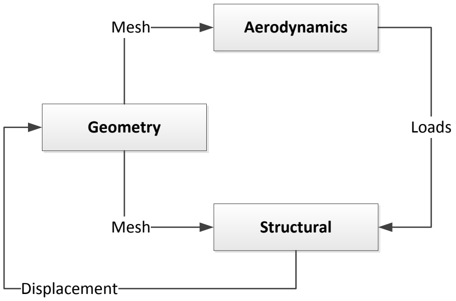

Figure 13.3 Multidisciplinary interaction between aerodynamics and structural aspects of the aircraft

By enabling a holistic design approach, the probability of capturing multidisciplinary couplings significantly increases. Furthermore, an integrated DA framework is a necessity for a possible MDO process. Nonetheless, before achieving an automated and integrated multidisciplinary approach, manually performed pre-processing activities have to be automated. One of the more time-consuming manual processes is related to mesh quality verification.

For the aerodynamic analysis, quadrilateral panels are required. A dedicated mesh model is created in CATIA V5. The quadrilateral panels have to be perfectly matched and it is therefore crucial to minimize eventual gaps between the nodes in order to achieve accurate results. Nevertheless, for geometries with complex curvatures like the curved leading edge of a wing or the union between a wing and a fuselage, there are bound to be gaps between the panels, see Figure 13.4.

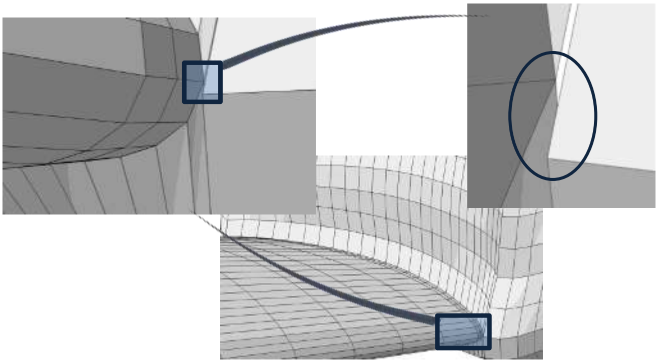

Figure 13.4 Identifying and eliminating gaps between mesh nodes

Eliminating the gaps between the nodes is a tedious process, with arguably no engineering creativity required. It is likewise a task that generates many errors when performed manually. Consequently, since the knowledge to perform this task is well-defined, the manual node connectivity operations can be translated to formal rules. By automating this tedious task, fast and accurate, aerodynamic analysis can be performed and the loads generated can be transferred to the FE model in a seamless fashion.

Finally, the presented framework has been verified by comparing the aerodynamic results with existing wing tunnel tests (Tarkian and Tessier, 2008).

CHAPTER 14

LOAD FRAME DESIGN AUTOMATION

A load frame is a device mounted on a forklift for operations requiring elevation of personnel. A load frame may have different sizes and various door configurations.

Although seemingly a simple design it consists of about 200 unique parts. Load frames are offered in a variety of different sizes and variants. For each order a new 3D CAD model with the 2D drafts as well as bill of material (BOM) are procured manually. According to provided figures, an average of approximately 40 hours are invested for each load frame. This is in spite of the variants being derived from the same load frame concept. Furthermore, no explicit standardization has been established and each load frame variant has therefore been fitted with various unique features as shown Figure 14.1.

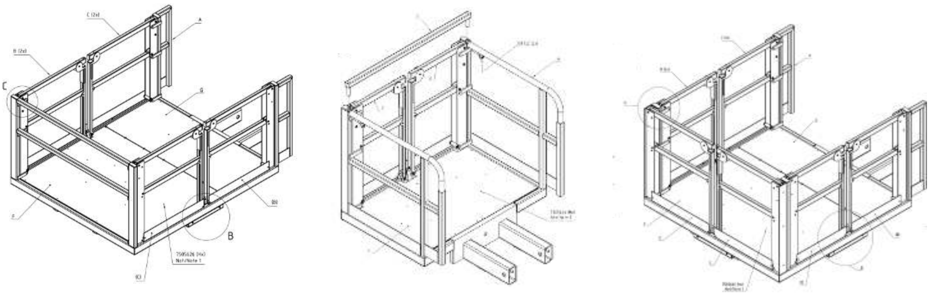

Figure 14.1 Three previous load frame variants

The main morphological change is the floor size of the load frame, which also affects the gate sizes. Topological changes include gates, lock mechanism and pillars. Given the vast number of repetitive manual processes and the varying topology and morphology of the load frame, it constitutes a suitable benchmarking product. In the following sections, a GeA framework based on HLCts, for load frames is presented.

14.1 AUTOMATED DESIGN GENERATION

Prior to the design automation framework, namely the load frame configurator, a careful study and review of the load frame design, assembly sequence and product flora were carried out. During this initial phase it could be assessed that a number of variants and customizations had been made over the years to accommodate singular customer wishes.

As an important part of the project, a set of standardized components are selected or developed, discarding unnecessary variants. The identified standard components are also included as the principle HLCts in the configurators HLCt database, see Figure 14.2. By combining them, a family of standard load frames can be generated.

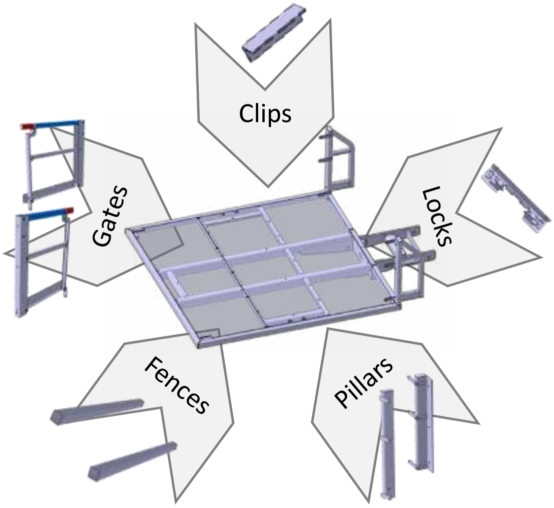

Figure 14.2 The load frame consists mainly of five types of HLCts

The solution implemented at the company includes a Graphic User Interface (GUI) where the user can select the dimensions of the load frame (width and length) and the number of gates. The allowed dimensions are between 1000mm x 1000mm and 2500mm x 2500mm, in 100mm steps. The number of gates (composed by a right and a left half) can be either two (one on each side of the frame) or three (one gate per side). This limited number of inputs is sufficient to generate a complete model of the load frame through the process described in Figure 14.3.

The first operation that is carried out is to update the dimensions of the base of the frame, which includes the following elements:

- The peripheral beams
- The floor plates
- The beams forming the central cross
- The two pillars for supporting the gates
- Brackets to attach and support the floor plates.

Once the base is updated, the remaining two pillars are instantiated. These can be of two types depending on the number of gates selected. With two gates the front of the load frame will be closed by a fixed fence, while if three gates are desired, the instantiated pillars need attachments for the third pair of gates in the front. The gates themselves are then instantiated. The gates HLCts differ from others because each instance must adapt its length (morphological transformation) to fit the length of the load frame side where it is instantiated. Also, the gates include a kinematic model that allows the designer to try opening the gates to verify that no clash occurs when maneuvering them. Finally a locking mechanism for each gate pair is instantiated below the floor.

As explained, pillars, gates and locking mechanisms are stored in an HLCt database, so that only the needed ones are added to the load frame assembly, following predefined instructions saved in the knowledge base.

Once all components are instantiated and the load frame is updated according to the user inputs, the weight of the complete assembly can be evaluated and presented to the designer in the GUI.

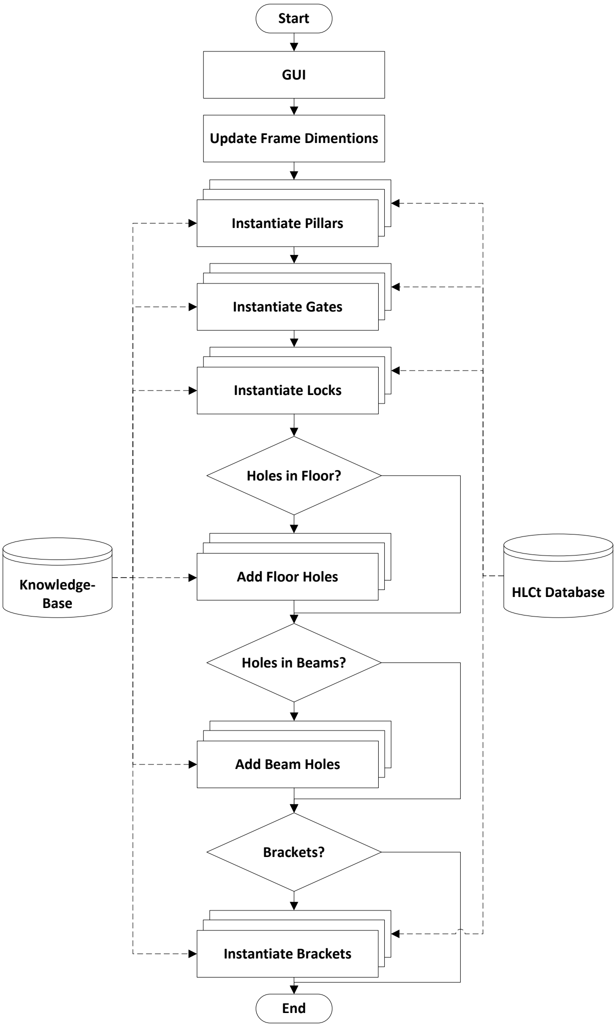

Figure 14.3 The HLCt instantiation process of the load frame

It is interesting to observe that different automation level (AL) approaches have been used when generating the models. The parts that constitute the frame base assembly are updated using AL_2, for instance the activation state of number of holes in the beams. AL_3 is utilized to automatically instantiate components from the database (i.e. gates, pillars and locks).

The development of the load frame configurator offered the opportunity to revise all drafts that are used as part manufacturing and assembly documentation. The rules stored in the knowledge base were employed to generate design tables collecting all possible sizes for each part or sub-assembly. Hence, these tables could be added to the drafts so that all documentation would be ready for any load frame variant within the designed family.

14.2 RETURN ON INVESTMENT

The implemented load frame configurator has had a dramatic reduction in design lead time. Initially, an expert load frame designer would need up to a week to redesign a load frame to reflect the customer's requirements. With the configurator the same operation can be carried out in a matter of minutes. Specifically, a development time of 10 minutes has been reported for a new load frame design with all necessary parts listed and drawings. This figure is less than 0.5% of the time required before this project was initiated.

These types of figures are not uncommon in the DA field (CHAPTER 6). However, creativity restrictions on engineers when they want to implement new models is a question not yet resolved. To this end, it has been verified that designers are in fact able to modify and replace previous HLCts with new ones, just by adding new HLCts to the database. The framework can thus be continuously adapted for newer product variants, fulfilling the requirements established in CHAPTER 9. This task would be more challenging to accomplish with an AL_1 or Al_2 modeling approach since the associative relations between the parts, need to be comprehended before changes to the product structure is made. These relations are intricately imbedded in the model structure; thus, if a new component is to be added then, then the relations have to be identified, the old component deleted, the new one added, and finally the relations with the new component established. It is therefore hard for the designer to modify the parametric model without accidently damaging the model's associativity.

The development, testing and validating of the configurator took a total of 240 man-hours. For variant design, the return on investment is reached with the sixth configured load frame.

PART V -

DISCUSSION AND CONCLUSION

Part IV concludes the thesis. Discussion topics related to the contributions and applications are presented. Conclusions are drawn both in general and in relation to the research questions that are formulated. Directions for future research are also described.

Machines will be capable, within twenty years, of doing any work that a man can do.
Herbert Simon, 1965

CHAPTER 15

DISCUSSION

In this chapter the presented topics are briefly summarized, discussed and related to previous work. Moreover, the proposed methods are generalized and the limitations determined.

15.1 AUTOMATED DESIGN GENERATION AND EVALUATION

The anticipated benefits of DA have over the decades become considerably more realistic; "*minimize repetitive and non-creative design activities*". Considering the fact that numerous researchers have established 80-90% of all design activities to be routine-like and repetitive (CHAPTER 6), minimizing these manual tasks would still constitute a great improvement to current design processes. Consequently, the contemporary DA objectives should still manage to reduce the product development time considerably.

A valid question is then why DA has failed to materialize in a broader industrial perspective. There are plenty of successful small-scale DA applications reported and applied in industry (CHAPTER 6). The figures from these studies indicate that it is possible to eliminate 70-90% of certain tedious design activities.

As usual, there is no simple and single answer to why this promising technology is not implemented as an industrial standard. On the other hand, many different factors are responsible for the current state. A review of relevant publications, CHAPTER 6.3, suggests that the research community is split, with limited collaboration between different research settings. This has resulted in limited methodological and terminological standardizations (CHAPTER 6.9). However, this does not imply that progress in DA and GeA has not been made. Lack of collaboration may have reduced research development speed but significant improvements in DA and GeA have nonetheless been realized over the past decade.

15.1.1 High Level CAD templates

A recent methodological leap in GeA is the modeling methods falling into the HLCt sphere (see CHAPTER 6.4.2.2). One of the great benefits of HLCt modeling is its modular architecture, sought after in many product development processes in order to increase design reuse as a means to increase efficiency.

After reviewing relevant GeA manuscripts, it is concluded that it can be branched into three automation levels (AL). Depending on the level of automation the complexity of transformation and model associativity varies (CHAPTER 9). Previous GeA attempts such as AL_1 and AL_2 (CHAPTER 9.1.1-9.1.2) have been centered on the concept of a *one-model-for-all* technique, where one model represents a set of product variants. It has been verified in the literature that these attempts have in fact been successful, at least to a certain extent. Although successful study cases have been formalized, the probability to sustain such an approach in a broader perspective might prove to be unfeasible. Increasing detail and complexity will eventually reduce the model's flexibility as well as making effective maintenance more difficult. Furthermore, the amount of reuse is restricted, since a great deal of knowledge is intertwined in the model architecture and thus difficult to extract.

On the other hand, AL_3, which constitutes HLCt modeling, is well suited for maintainability and reuse due to its modular architecture. In this approach, both the knowledge and geometries are stored and instantiated upon request. Hence the complexity of neither the knowledge base nor the HLCt database increases with increasing modeling detail.

HLCt modeling is in one aspect quite similar to current bottom-up modeling strategies where models are generated out of context. This trait has been recognized as the main reason why bottom-up models lack product associativity. Although HLCts are likewise built out of context, they still possess specific input and output geometries that other HLCts can attach to according to the logic stored in the knowledge base. Nevertheless, being similar to bottom-up modeling is beneficial in terms of easier implementation in the company.

The main limitation of HLCt modeling is that although the methods are well established, no tool fully supports it. Various CAD tools have certain functions such as *powercopy* and *knowledge pattern* in CATIA V5, where templates can be defined. These functions facilitate easier implementation of the transformation step T4, where geometric reference-features are required to achieve instantiation. Despite easier implementation, a substantial quantity of in-house scripting is still necessary to completely setup up a GeA framework based on HLCts. In-house scripts are generally undesirable in industry, requiring expert knowledge and therefore limiting the number of engineers able to manage and maintain the framework.

For HLCt method to reach full potential, it is believed that a graphical user interface is necessary where the HLCts can be defined, stored, modeled and connected to each other in a less script demanding environment. Tool standardization for HLCt modeling is an essential precondition before it can be widely accepted and applied in industry.

15.1.2 High Level Analysis templates

As emphasized by several researchers (CHAPTER 6.8), geometry models are essential integrators in multidisciplinary design frameworks. Being rich information carriers, CAD models are suitable as the input source for many CAE models. Many formalized methods have been presented to reach and manage data flow between CAD and CAE (CHAPTER 6.8).

In order to sustain the modular architecture of the presented MDO process, CAE analysis is realized by utilizing HLCts to automatically generate HLAts, which are then placed in a database for further analysis. Hence, during design evaluation, the type of HLAt is first topologically selected followed by modifications on the internal input variables. With the proposed method, time-consuming processes such as applying mesh, boundary condition and loads is automated, minimizing the tedious operations associated with CAE analysis.

Nevertheless, like HLCt modeling, the main limitation for effective industrial realization is lack of standard implementation and maintenance. Although this approach has been successfully applied and verified on the industrial robot application, a great deal of scripting within the dedicated tool, ANSYS, is required. Moreover, although the proposed approach is predominantly stable, some implications in terms of sudden tool and model error and crashes do occur. These types of implications significantly reduce the likelihood of the proposed methods being accepted in industry, especially if applied on iteration intensive processes where the probability of crashes increases.

15.1.3 Global Metamodels in Multidisciplinary Optimization

As a means to increase evaluation speed, global metamodels are introduced as replacements for the computationally expensive HLCts and HLAts. The challenge is to realize a metamodel with high accuracy, without having to drastically increase the number of expensive samples. It is thus vital to more efficiently capture the design space.

To do so, the indirect sampling (IDS) method is proposed (CHAPTER 11.1). The IDS approach is suitable for models requiring inputs from sequentially preceding models. With IDS, the sampling is performed on the inputs of all the models in sequence. Hence when generating the sample the entire integrated design framework has to be employed. IDS is therefore a somewhat more complicated sampling procedure compared to direct sampling (DS). On the other hand, with IDS the quality of the obtained metamodel is much improved.

Another benefit of global metamodels, rarely mentioned in the literature, is that the risk of sudden errors and crashes during design and optimization being significantly reduced.

The main implication of metamodeling is that yet another step in the design process is required. New metamodels have to be frequently created as new HLCts and HLAts are generated, which is undeniably a time-consuming task. However, this step is a necessity as long as the concerns of modeling stability and lengthy simulation times for CAD and CAE tool are not resolved.

15.1.4 Multidisciplinary Optimization

One of the stated research questions in this thesis considers efficient optimization strategies. In this regard MDO methods are classified into single-level (SL) and multilevel (ML) strategies. Generally, SL strategies are considered more efficient. Nonetheless, in the case of industrial robot MDO, the backward recursive (BR) ML strategy clearly outperforms the all-at-once (AAO) strategy, contradicting the previously, generally established perceptions. For not being in agreement with established theories, further disclosure of the BR strategy is necessary.

First, it should be emphasized that the iteration efficiency of SL strategies is not challenged in this thesis; it is actually further verified. What is claimed is that less processing time is required to reach optimal solutions with the proposed ML strategy.

The complete serial link manipulator is partitioned in link-specific local optimization by implementing the rationale behind the Newton-Euler formulation. A very important difference with other ML strategies, such as BLISS, is the coupling variables between the optimization levels being single directional. This fact alone significantly decreases the number of required iterations. The total number of iterations generated in the BR ML strategy is also greater than in the AAO SL strategy. The superior processing efficiency is thus explained by not necessitating execution of the expensive dynamic model in the local optimizations. Evidently, exploiting the complex multidscipinary characteristics of the concept under investigation might therefore be beneficial when choosing between different modeling and optimization strategies.

15.2 APPLICATION EXAMPLES

In this section relevant attributes gathered from the application examples are presented and further discussed.

15.2.1 Multidisciplinary Optimization of Industrial Robots

This particular application example has a central role in this thesis, and has therefore been more rigorously evaluated compared to the other two application examples. Hence, many of the relevant results are discussed in other sections of this chapter. It is therefore sufficient to mention that the proposed methods have been implemented in a framework consisting of CAD/CAE tools used in industry. Even though the framework has not been utilized directly in an industrial product development process, engineers from industry have stated the demands on the framework and assessed its applicability for industrial usage. Moreover, different CAD tools are used in the design cases presented in Paper I and V, further supporting the generality of the proposed HLCt method.

15.2.2 Multidisciplinary Aircraft Design

By dividing the modeling elements into a multi-layered instantiation process, such as external surfaces, internal structures and interior HLCts, it is clearly demonstrated that detail modeling on specific features can be conducted without restricting the user from making radical design modifications.

A hypothetical scenario is if a drastic design change would require the exterior of the aircraft to be completely redefined. In AL_1 and AL_2 approaches, such a design change would necessitate scrapping the model. In that case, all associative features such as internal structures have to be removed and most likely re-modeled for the new exterior geometry. On the other hand, with HLCt modeling, the structure and interior HLCts are stored in databases and can therefore be automatically instantiated on any generated exterior geometry, as long as the correct geometric inputs are provided.

Thus by storing both the HLCts as well as the requiring knowledge base, the prospect for reuse and to adapt to radical design changes is significantly enhanced.

15.2.3 Load Frame Design Automation

HLCt based modeling has previously been successfully implemented in industrial settings by other researchers, as elaborated in CHAPTER 6. Nonetheless, maintainability is still an unresolved issue and one remaining question is whether engineers can independently maintain the HLCt based design framework.

As presented in CHAPTER 14.2, after applying the GeA framework, the speed of retrieving new load frame configurations is greatly increased. After evaluating the system for nearly a year it is reported that the engineers are able to add and replace various HLCts to continuously update the configurator framework. Hence, the preliminary reports indicate that the engineers are in fact able to maintain the system independently.

15.3 GENERALITY OF THE PROPOSED METHODS

Perhaps the most relevant question remaining is how generic the suggested methods are. Applying the methods on three diverse applications does not automatically make the methods generic and a discussion concerning generality is necessary.

15.3.1 High Level CAD template

The HLCt method has been successfully implemented in three design applications. The common denominator is the need to modify the number, shape and type of geometric features.

In all three cases the HLCt method has provided an effective solution, fulfilling the requirements. Does this mean that the method can be applied on any product where dynamic topology occurs? Simply put, the answer is no. The HLCt method should only be applied where well-defined topological modifications occur. Essentially this means the context in which the HLCt is instantiated upon should not change drastically in terms of type and number of interface features.

As far as generality is concerned regarding modeling tools, the possibility to apply HLCt modeling on different CAD tools has been reported (CHAPTER 6.4 and Paper I and Paper V). It is therefore safe to assume that HLCt modeling can be applied on most common CAD tools.

15.3.2 High Level Analysis template

The HLAt method has been applied in the industrial robot application with adequate results. To apply the HLAt in other applications, the following generic pre-conditions have to be fulfilled: well-defined topological modifications and no drastic changes in terms of type and number of mesh, boundary conditions and loads. Nonetheless, the fact remains that the HLAt method has only been verified in one case study and thus the possibility to generalize the results on a wider scientific context is limited.

The proposed HLAt method has so far only been applied on the dedicated CAE tool ANSYS. The same outcome is not verified on other high-end CAE tools. However, it is highly probable, since the possibility to add user-defined scripts to automatically generate pre- and post-processing operations is provided by most tool vendors.

15.3.3 Indirect Sampling

Although only applied in the industrial robot application (CHAPTER 12.4.1), it can be argued that the IDS method is in fact generic and able to improve NRMSE for models requiring input from preceding sequential models. The utilized application example is thus trivial, due to the essential IDS rational of only including samples that are realistic representations of the design space.

Eventual acceptance of the IDS method in industry has not been properly addressed in this thesis. Although sampling efficiency is established, the number of steps required to reach this goal is definitely a drawback in terms of possible acceptance by industry. Therefore, there is no question that the IDS method is well-suited in terms of design space approximation for multidisciplinary products. However, whether the required pre-processing time to enable the IDS methods can be accepted is not fully addressed.

15.3.4 Multi-Level MDO for Serial Link Structures

The BR ML strategy has been specifically developed for MDO of industrial robots. This approach has a relatively narrow applicability and is only suited for products with open kinematic chains such as serial link manipulators and cranes.

CHAPTER 16

CONCLUSION

Automating non-creative and iterative manual tasks has been recognized as an effective approach to cut down lead time and thus increase competitiveness. The question is *what* is iterative and non-creative and *how* can it be automated. In this thesis two non-creative processes have been identified and addressed: the modeling process and the design iteration process.

Regarding the modeling process, it is established that topology is equally important as morphology to take into consideration for effective geometry parameterization. To facilitate topology parameterization, HLCt modeling is proposed. With HLCt modeling less restriction is put on the design space compared to earlier GeA contributions. Furthermore, HLCt facilitates automated concepts evaluations with HLAt, eliminating many tedious pre-processing activities.

Before utilizing the proposed methods in a design iteration process, the poor stability and evaluation time of the models have to be considered. As a remedy, global metamodels are utilized as effective tools to reduce evaluation time and thus improve the design iteration process.

The impact of an automated approach based on the proposed methods is obvious: first, a significantly increased number of design iterations becomes possible in contrast to the manual design iteration; second, design trials can be guided by more scientific techniques, viz. optimization algorithms, in contrast to trial-and-error guided by engineers; and third, the product may be treated holistically as a complete system in order to avoid sub-domain optimization by engineers of each discipline.

In the following section, the scientific contributions of this thesis are summarized as answers to the previously stated research questions.

RQ1. How to enable multidisciplinary automation and optimization processes for mechanical engineering products?

To enable MDO for mechanical products where the geometric properties are of importance, DA platforms are required. However setting up DA platforms is a time-consuming process. For the MDO process to be practical, the DA platform should be adaptable to new design changes and requirements. Prominent researchers have thus recognized modular architecture methods for efficient DA platforms.

RQ2. Which types of engineering processes are suitable to automate?

Generally, repetitive and non-creative engineering processes are considered to be suitable manual processes for automation. Specifically, the following engineering operations have been recognized as appropriate for automation:

- CAD transformations
    - Morphology:
        - Relations between geometric features
    - Topology
        - Assembly operations
        - Establishing contextual links between geometries
- CAE pre-processing:
    - Importing CAD parts
    - Applying mesh
    - Applying pre-conditions such as boundary conditions and loads

In this thesis the above operations have been automated for a set of different engineering applications.

RQ3. How should the identified engineering processes, suitable for automation, become automated?

In the literature, three different automation levels (ALs) are found, of which the HLCt based AL_3 is considered to be the most suitable. A high level template-based method is likewise proposed for CAE analysis. Thus, suitable automation procedures for concept generation and evaluation are:

- HLCt for CAD transformations.
- HLAt for CAE analysis.

RQ4. How to achieve fast design iterations?

For an MDO process to be applicable in industry, fast evaluations are a necessity. In the presented work this is achieved by:

- Metamodeling for faster design evaluations.
- Indirect Sampling (IDS) for efficient and better approximation of the design space.

RQ5. How to implement the proposed methods to minimize the required changes to the companies' current design process?

In order for the proposed methods to be applicable in industry, compatibility with the tools and methods used in industry is required. Furthermore, a modular framework structure and the usage of metamodels facilitate implementation. The usage of metamodels allows a more flexible execution of the framework, as it is less error-prone.

RQ6. How to organize the design automation process to maximize maintainability?

Early attempts to implement DA in industry failed to revolutionize the design process, due among other things to too broad scopes with intelligent platforms trying to accomplish too many things. With the proposed decoupled MDO process, each independent DA package can be more easily maintained and further developed.

RQ7. Which optimization strategies are suitable for the proposed design automation framework?

For an MDO framework to be applicable in industry it needs to be efficient and flexible. The thesis presents a flexible framework, where efficiency relies on appropriate optimization algorithms, and suitable problem formulations.

Generally, Single-Level (SL) strategies are reported to be more effective and less iteration-intensive compared to Multi-Level (ML) strategies. Specifically for the industrial robot application, the proposed backward recursive (BR) ML strategy clearly outperforms the traditional all-at-once (AAO) SL approach. The BR ML strategy is therefore suggested as the ideal method when conducting holistic MDO on serial link manipulators.

16.1 FUTURE WORK

For future work studies three subjects stand out. The first study involves further applications in real industrial settings. Second, the identified limitation of high level template modeling is addressed, whereas the third study has more of a spin off character regarding metamodeling.

16.1.1 Application in Industrial Projects

Even though the tools and methods proposed in this thesis have been developed in co-operation with industry, more rigorous tests in industrial development processes are required. A natural continuation of this work is to apply the proposed tools and methods in other areas and in real development processes. It would also be interesting to conduct qualitative studies to estimate the benefits of the proposed framework.

16.1.2 Tools to enable High Level Template Modeling in Industry

As discussed in CHAPTER 15, the main remaining challenge to support HLCt and HLAt in industry is a standardized modeling functionality. For the time being, facilitating high level template modeling requires a non-negligible amount of dedicated scripting. The main feature of high level template modeling is the modular architecture, where new templates can be introduced continuously. However, setting up the architecture or introducing any major changes or updates will still require expert know-how. To this end a standardized modeling functionality will allow companies to plan and execute these necessary tasks more independently.

16.1.3 Metamodels as a Cost-efficient Collaborative Tool

In this work, global metamodels have been applied to enable evaluation and integration for effective MDO. However, the possible benefits of global metamodels stretch further than the traditional reasons and are rarely discussed in the literature. First, unlike CAD and CAE models, metamodels do not need any expert knowledge to operate, since they can be easily implemented on e.g. spreadsheets. Second, the complexity of model management between different departments is considerably reduced since global metamodels can function independently and do not need to be integrated with other models to provide adequately accurate output. Furthermore, they do not require expensive licenses in order to be executed. The possibility to adopt metamodels as a cost efficient collaborative tool is thus a subject that requires future consideration.

CHAPTER 17

SUMMARY OF PAPERS

In this chapter short summaries of the appended papers are provided, with the aim to explain the role of each paper.

PAPER [I]

DESIGN AUTOMATION OF MODULAR INDUSTRIAL ROBOTS

In this paper the notion of topology and morphology parameterizations is introduced. Thus, a novel approach for design of modular industrial robots is proposed where not only the shape but also the type of the structure and actuators are parameterized. By utilizing the presented method, it is illustrated how radically different concepts can be modeled and analyzed parametrically.

To further verify the presented design methodology, an integrated analysis tool for industrial robots is developed combining dynamic and geometric models in a parametric design approach. An optimization with mixed discrete and continuous variables is performed to demonstrate the capabilities of the proposed framework by optimizing a modular industrial robot concept. Furthermore, the need to cut evaluation time is raised and therefore geometric properties are saved in a database and re-used during the optimization, leading to fewer time consuming CAD evaluations.

PAPER [II]

PRODUCT PLATFORM AUTOMATION FOR OPTIMAL CONFIGURATION OF INDUSTRIAL ROBOT FAMILIES

The modular design framework proposed in Paper I, is well suited for product family and platform design. In Paper II a quantitative product family approach based on modular high fidelity models is presented. The product family design is formally stated as a multi-objective optimization problem, which is solved using a multi-objective genetic algorithm. The results suggest that by generating a Pareto front, a wide range of optimal product families can be obtained. By presenting a set of optimal solutions with varying balance between commonality and performance, critical decisions can be made later in the design process when more knowledge considering the concepts is available.

The need to cut evaluation time is further stressed in this paper. To achieve faster optimization results, the concept of re-using former evaluated data during optimization is once again applied. Nonetheless, the need for more efficient methods such as metamodeling and multi-layer optimization is raised.

PAPER [III]

FLEXIBLE AND ROBUST CAD MODELS FOR DESIGN AUTOMATION

In this paper generic modeling methods are stated to facilitate MDO of complex engineering products. The proposed methods are based on several research projects and applied on three different application examples. To realize MDO, KBE is adopted with the aim of achieving design reuse and automation.

The concept of High Level CAD templates (HLCt) is proposed to automatically construct flexible and robust CAD models. Furthermore, a quantification method for the terms flexibility and robustness is presented, providing a means to measure the quality of the CAD models.

PAPER [IV]

MULTIDISCIPLINARY DESIGN OPTIMIZATION OF MODULAR INDUSTRIAL ROBOTS BY UTILIZING HIGH LEVEL CAD TEMPLATES

This paper presents a design framework that integrates HLCts and physic-based models for automated geometry manipulation, dynamic simulation, and structural strength analysis. To realize such an evaluation expensive framework, a novel multi-level optimization strategy as well as metamodeling for CAD and FEM is employed to significantly speed up the design optimization process.

It is established that CAD models can be effectively replaced with global metamodels, requiring a rather negligible number of samples. FE models, on the other hand require a substantial amount of samples, necessitating a novel sampling approach in order to reach an acceptable precision.

PAPER [V]

METAMODEL BASED DESIGN AUTOMATION – APPLIED ON MULTIDISCIPLINARY DESIGN OPTIMIZATION OF INDUSTRIAL ROBOTS

Paper V utilizes HLCt and, for first time, introduces the high-level template approach for FE analysis, effectively eliminating many repetitive tasks.

Furthermore, the indirect sampling (IDS) method is presented to drastically decrease the number of samples required to create precise FE metamodels.

Thus, with the contribution of this paper, a complete MDO design process is presented, which is able to parametrically generate and evaluate diverse concepts, in a time efficient manner. To increase the level of generality another type of industrial robot concept is studied in this paper compared to Paper I, Paper II and Paper IV. Additionally, a new set of CAD and CAE tools is applied, to further strengthen the generic features of the high level template methods.

CHAPTER 18

REFERENCES

Amadori, K, Geometry Based Design Automation: Applied to Aircraft Modelling and Optimization, Doctoral thesis, Linköping University, 2012.

Amadori, K, Johansson, B, Krus, P, Using CAD Tools and Aerodynamic Codes in a Distributed Conceptual Design Framework, 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, USA, Jan. 2007.

AutoCAD, 2012, http://usa.autodesk.com/autocad/, last accessed on 2012-08-30.

Bento J, Feijó B, An agent-based paradigm for building intelligent CAD systems, Artif Intell Eng. 1997;11(3 SPEC. ISS.):231-44.

Blessing, LTM, Chakrabarti, A, DRM, a Design Research Methodology, Springer, 2009.

Blount, GN, Kneebone, S, Kingstone, MR, Selection of Knowledge-based Engineering Design Applications, Journal of Engineering Design, Vol. 6, No. 1, 1995, pp. 31-38.

Bowcutt, KG, A Perspective on the Future of Aerospace Vehicle Design, AIAA 2003-6957, 12th AIAA International Space Planes and Hypersonic Systems and Technologies, Norfolk, VA, USA, Dec. 2003.

Box MJ, A new method of constrained optimization and a comparison with other methods, Computer Journal, 8, pp. 42-52. 1965.

Brandt, SA, Stiles, RJ, Bertin, JJ, Whitford, R, Introduction to Aeronautics: A Design Perspective, AIAA Educational Series, Reston, VA, USA, 1997.

Brewer, H, Automated tool design: age forming tool for aerospace panels, International ICAD Users Group Conference Proceedings 1996.

Brujic D, Ristic M, Mattone M, Maggiore P, De Poli GP, CAD based shape optimization for gas turbine component design, Structural and Multidisciplinary Optimization. 2010;41(4):647-59.

Buur, J, *A theoretical approach to mechatronics design*, Doctoral thesis, Institute for engineering design, Technical University of Denmark, Lyngby, Denmark, 1990.

Böhnke D, Reichwein A, Rudolph S, Design language for airplane geometries using the unified modeling language, Proceedings of the ASME international design engineering technical conferences and computers and information in engineering conference 2009, DETC2009; 2010. p. 661-70.

CATIA V5, 2012, http://www.3ds.com/products/catia, last accessed on 2012-08-30.

Cederfeldt, M, Planning Design Automation – A Structured Method and Supporting Tools, Doctoral Thesis, Chalmers University of Technology, Göteborg, Sweden, 2007

Cederfeldt, M, Sunnersjö, S, Solid Modeling With Dimensional and Topological Variability, Proceedings of ICED03: International Conference on Engineering Design, August 19–21, Stockholm, Sweden, 2003.

Chapman, CB, Pinfold, M, The application of knowledge based engineering approach to the rapid design and analysis of an automotive structure, Journal of Advances in Engineering Software, Vol. 32, Issue 12, Dec. 2001, pp. 903-912.

Cramer EJ, Dennis JE, Frank PD, Lewis RM, Shubin GR, Problem formulation for multidisciplinary optimization, Center for Research on Parallel Computation Rice Univ., CRPC-TR93334, 1993.

Cooper, S, Fan, I, and Li, G, Achieving Competitive Advantage Through Knowledge Based Engineering, -A Best Practice Guide, White Paper for the Department of Trade and Industry, Cranfield, UK, 2001.

Danjou, S, Koehler, P, Challenges for design management, Computer-Aided Design and Applications. 2007;4(1-6):109-16.

Danjou, S, Lupa, N, Koehler P, Approach for automated product modeling using knowledge-based design features, Computer-Aided Design and Applications. 2008;5(5):622-9.

Dattoma V, De Giorgi M, Giancane S, Manco P, Morabito AE. A parametric-associative modelling of aeronautical concepts for structural optimization, Adv Eng Software. 2012;50(1):97-109.

Elfving, JA, Exploration of Opportunities to Reduce Lead Times for Engineered-to-Order Products, A dissertation for Engineering-Civil and Environmental Engineering, Graduate Division of the University of California, Berkeley, 2003.

Elmqvist H, Mattson SE, and Otter M, Modelica – An International Effort to Design an Object-Oriented Modeling Language, in Proceedings of Summer Computer Simulation Conference, SCSC'09, Nevada, USA, 1998.

Encanação, JL, Lindner, R and Schlechtendahl, EG, Computer Aided Design – Fundamentals and System Architectures, Springer-Verlag Ltd., Berlin, Germany, 1990.

Epton, MA, PANAIR- A Computer Program for Predicting Subsonic or Supersonic Linear Potential Flows About Arbitrary Configurations Using a Higher Order Panel Method Volume I (version 3.0), Boeing computer services company, Seattle Washington, USA, 1981.

Excel, 2012, http://office.microsoft.com/en-us/excel/, last accessed on 2012-08-30.

Ferguson, SF, Kasprzak, EM, and Lewis, K, Design of a Family of Reconfigurable Vehicles Using Multilevel Multidisciplinary Design Optimization, Structural and Multidisciplinary Optimization Journal, 39(2), pp171-186, 2008.

Fonseca, CM, Fleming, PJ, Multiobjective optimization and multiple constraint handling with evolutionary algorithms—Part I: A unified formulation. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 28(1) 26–37, 1998.

Frohm, J, Levels of Automation in production systems, in Department of production system. vol. Doctorial Gothenburg: Chalmers University of technology, 2008.

Fujita K, Yoshida H, Product Variety Optimization Simultaneously Designing Module Combination and Module Attributes, Concurrent Engineering: Research and Applications, 12(2), pp105-118, 2004.

Giesing, JP, Barthelemy, J-FM, A summary of industry MDO applications and needs, AIAA 1998-4737, AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO, USA, Sep. 1998.

Goldberg, D, Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, 1989.

Gujarathi GP, Ma Y-, Parametric CAD/CAE integration using a common data model. J Manuf Syst. 2011;30(3):118-32.

Haftka RT, Simultaneous analysis and design, AIAA J 23(7):1099-1103, 1985.

Halfawy M, Froese T, Building integrated architecture/engineering/construction systems using smart objects: Methodology and implementation, J Comput Civ Eng. 2005;19(2):172-81.

Heinz, A, 777 Rule Based Design - Integrated Fuselage System, International ICAD Users Group Conference Proceedings 1996.

Hansen LU, Heinze W, Horst P, Blended wing body structures in multidisciplinary pre-design, Structural and Multidisciplinary Optimization. 2008;36(1):93-106

Hoffmann CM, Kim K-, Towards valid parametric CAD models, CAD Computer Aided Design. 2001;33(1):81-90.

Hoffman CM, Joan-Arinyo R, CAD and the product master model. CAD Computer Aided Design. 1998;30(11):905-18.

Hopgood, AA, Intelligent Systems for Engineers and Scientists, Second Edition, CRC Press LLC., FL, USA, 2001.

Hürlimann F, Kelm R, Dugas M, Oltmann K, Kress G. Mass estimation of transport aircraft wingbox structures with a CAD/CAE-based multidisciplinary process, Aerospace Science and Technology. 2011;15(4):323-33.

Institutionen för Medicin, Göteborgs Universitet, (2009), Forskningsmetodik, (Electronic), Available: http://www.infovoice.se/fou/bok/10000025.shtml, last accessed on 2012-08-30.

iSIGHT, 2012, http://www.3ds.com/products/simulia/portfolio/isight-simulia-execution-engine/overview, last accessed on 2012-08-30.

ISO Standard 8373:1994, Manipulating Industrial Robots – Vocabulary.

Krus P, Andersson J, Optimizing optimization for design optimization, Proceedings of DETC'2003, ASME Design Automation Conference, Chicago, 2003.

La Rocca G, Van Tooren MJL, Knowledge-based engineering approach to support aircraft multidisciplinary design and optimization. J Aircr. 2009;46(6):1875-85.

Larsson, T, Multibody dynamic simulation in product development, Doctoral thesis, Luleå University of Technology, 2001.

Lee SH, A CAD-CAE integration approach using feature-based multi-resolution and multi-abstraction modelling techniques, CAD Computer Aided Design. 2005;37(9):941-55.

Lee KS, Luo C, Application of case-based reasoning in die-casting die design, Int J Adv Manuf Technol. 2002;20(4):284-95.

Liersch, M, Hepperle, A Unified Approach for Multidisciplinary Preliminary Aircraft Design, CEAS European Air and Space Conference, 26-29 October 2009, Manchester.

Ledermann C, Hanske C, Wenzel J, Ermanni P, Kelm R, Associative parametric CAE methods in the aircraft pre-design, Aerospace Science and Technology. 2005;9(7):641-51.

Lin B-, Hsu S-, Automated design system for drawing dies, Expert Syst Appl. 2008;34(3):1586-98.

Ma Y-, Tor SB, Britton GA, The development of a standard component library for plastic injection mould design using an object-oriented approach, Int J Adv Manuf Technol. 2003;22(9-10):611-8.

Martins, JD and Simpson, TW, A Method to Manage Uncertainty During System-Level Conceptual Design, ASME DETC2005-84984, ASME Design Engineering Technical Conference – Design Automation Conference, Long Beach, California, September 24-28, 2005.

MATLAB, 2012, http://www.mathworks.se/products/matlab/

Mawhinney P, Price M, Curran R, Benard E, Murphy A, Raghunathan S., Geometry-based approach to analysis integration for aircraft conceptual design, AIAA-2005-7481, 5th annual aviation technology, integration, & operations (ATIO) forum, Washington, DC, September 2005.

McAllister CD, Simpson TW, Multidisciplinary robust design optimization of an internal combustion engine, ASME - Journal of Mechanical Design, 125(1):124-30, 2003.

McKay, MD, Beckman, RJ and Conover, WJ, A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code, Technometrics Vol. 21, No. 2, May 1979.

Mills, HD, Software Development, IEEE Transactions on Software Engineering, Vol.SE-2, No. 4, December 1976, pp.265-273.

modeFRONTIER, 2012, http://www.modefrontier.com/homeMF.html, last accessed on 2012-08-30.

ModelCenter, 2012, http://www.phoenix-int.com/software/phx-modelcenter.php, last accessed on 2012-08-30.

Myers, RH, Montgomery, DC and Anderson-Cook, CM, Response Surface Methodology, Process and Product Optimization Using Designed Experiments, Third Version, John Wiley & Sons, Inc., Hoboken, New Jersey, 2009.

Myung S, Han S, Knowledge-based parametric design of mechanical products based on configuration design method, Expert Syst Appl. 2001;21(2):99-107.

Mäntylä, M, A Modeling System for Top-Down Design of Assembled Products, IBM Journal of Research and Development, Vol.34, No. 5, September 1990, pp.636-659.

Nelder, J, Mead, R, A simplex method for function minimization, Computer Journal 7: 308–313, 1965.

NX, 2012, http://www.plm.automation.siemens.com/en_us/products/nx, last accessed on 2012-08-30.

Park, H-S, and Dang, X-P, Structural optimization based on CAD-CAE integration and metamodeling techniques, Computer-Aided Design, Vol. 42, 2010, pp. 889-902.

Pebesma, EJ, and Weeseling, CG, GSTAT: A Program for Geostatistical Modelling, Prediction and Simulation, Computers & Science Vol. 24, No. 1, pp. 17-31, 1998.

Pellicciari, M, Berselli, G, Leali, F, Vergnano, A., Lennartson, B., Object-oriented modeling of industrial manipulators with application to energy optimal trajectory scaling, ASME IDETC 2011, Washington DC, USA, 2011.

Prasanna L, Guhanathan S, Agrawal R, Automated approach for designing components using generic and unitized parametric sketches, Computer-Aided Design and Applications. 2010;7(1):109-23.

Pettersson, M, Design Optimization in Industrial Robotics – Methods and Algorithms for Drive Train Design, Doctoral thesis, Linköping University, 2008.

Petterson M, Andersson J, Krus P, Methods for Discrete Design Optimization, in proceedings of ASME DETC'05, Design Automation Conference, Long Beach, California, USA, September 24-28, 2005.

Pro-Engineer, 2012, http://www.ptc.com/product/creo/parametric, last accessed on 2012-08-30.

Raffaeli R, Mengoni M, Germani M, Mandorli F, An approach to support the implementation of product configuration tools. Proceedings of the ASME international design engineering technical conferences and computers and information in engineering conference 2009.

Rodríguez A, Fernández-Jambrina L, Programmed design of ship forms. CAD Computer Aided Design. 2012;44(7):687-96.

Roozenburg, NFM, Eekels, J, Product Design: Fundamentals and Methods, John Wiley & Sons, 1995.

Salehi V, McMahon C, Development and application of an integrated approach for parametric associative CAD design in an industrial context, Computer-Aided Design and Applications. 2011;8(2):225-36.

Sandberg M, Tyapin I, Kokkolaras M, Isakasson O, Aidanpää J-, Larsson T, A knowledge-based master-model approach with application to rotating machinery design, Concurrent Engineering Research and Applications. 2011;19(4):295-305.

Simpson, TW, and Martins, JD, Multidisciplinary Design Optimization for Complex Engineered Systems: Report from a National Science Foundation Workshop, ASME - Journal of Mechanical Design, 133(10), 2011.

Shan, S, Wang, GG, M, Metamodeling for High Dimensional Simulation-Based Design Problems, ASME - Journal of Mechanical Design, 132(5), 2010.

Sicilano B., Modeling and Control of Robot Manipulators, Springer Verlag, 2001.

Siddique Z, Boddu K, A CAD template approach to support web-based customer centric product design, Journal of Computing and Information Science in Engineering. 2005;5(4):381-7.

Sobieszczanski-Sobieski J, Kodiyalam S, BLISS/S: A new method for two-level structural optimization. Structural and Multidisciplinary Optimization, 21(1):1-13, 2001.

Stokes, M, Managing Engineering Knowledge; MOKA: Methodology for Knowledge Based Engineering Applications, Professional Engineering Publishing, London, 2001.

Sunnersjö S, Cederfeldt M, Elgh F, Rask I, A transparent design system for iterative product development, Journal of Computing and Information Science in Engineering. 2006;6(3):300-7.

Tarkian, M, Zaldivar, F, Aircraft Parametric 3D Modeling and Panel Code Analysis for Conceptual Design, 26thCongress of International Council of the Aeronautical Sciences, Anchorage, USA, Sep, 2008.

Tomiyama, T, Intelligent computer-aided design systems: Past 20 years and future 20 years, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 21, Issue 1, pp. 27-29, 2007.

Ullman, DG, The mechanical design process, fourth edition, McGraw Hill, 2010.

Vandenbrande, JH, Grandine, A, Hogan, T, The search for the perfect body: shape control for multidisciplinary design optimization, AIAA 2006-928, 44th AIAA Aerospace Science Meeting and Exhibit, Reno, NV, USA, Jan. 2006.

Venter G, Sobieszczanski-Sobieski J, Multidisciplinary optimization of a transport aircraft wing using particle swarm optimization, Structural and Multidisciplinary Optimization, 26(1-2):121-31, 2004.

Verhagen, WJC, Bermell-Garcia, P, van Dijk, REC, Curran, R, A Critical Review of Knowledge-Based Engineering: An Identification of Research Challenges, Advanced Engineering Informatics, 2011.

Weisberg, D, The Engineering Design Revolution, E-Book, www.cadhistory.net, 2010.

Wirth, N, Program Development by Stepwise Refinement, Communication of the ACM, Vol.14, No. 4, April 1971, pp.221-227.

Yi SI, Shin JK, Park GJ, Comparison of MDO methods with mathematical examples. Structural and Multidisciplinary Optimization, 35(5):391-402, 2008.