Design analysis and redimensioning of prismatic parts for manufacturability

Lin, Jietong

2006

Lin, J. (2006). Design analysis and redimensioning of prismatic parts for manufacturability.
Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/6127>

<https://doi.org/10.32657/10356/6127>

Nanyang Technological University

Downloaded on 26 Apr 2023 16:20:16 SGT

DESIGN ANALYSIS AND REDIMENSIONING OF PRISMATIC PARTS FOR MANUFACTURABILITY

LIN JIE TONG

SCHOOL OF MECHANICAL & AEROSPACE ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY

2006

**Design Analysis and Redimensioning
of Prismatic Parts for
Manufacturability**

Lin Jie Tong

School of Mechanical & Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2006

ACKNOWLEDGMENT

First and foremost I would like to acknowledge with great appreciation the guidance and support of my supervisor Dr. Georg Lothar Thimm. My gratitude goes also to Dr. Graeme A. Britton, who also supported me in this research.

I also would like to thank the technical staff of Design Research Center for assistance rendered in the course of the research.

Finally, I want to make clear that this thesis would never have been completed without the encouragement, support and patience of my wife Yuna.

CONTENTS

ACKNOWLEDGMENT	1
CONTENTS	2
ABSTRACT	6
LIST OF FIGURES	9
LIST OF TABLES	10
LIST OF PUBLICATIONS	12
1 INTRODUCTION	13
1.1 Motivation	13
1.2 Objective	14
1.3 Scope	15
1.4 Contributions	15
2 LITERATURE REVIEW	17
2.1 Introduction	17
2.2 Design for Manufacturability (DFM)	19
2.2.1 Basic approach to implement DFM	19
2.2.2 Automatic generation of redesign suggestions	20
2.2.3 Measures of manufacturability	20
2.2.4 State-of-the-art DFM	21

CONTENTS

2.3	Computer Aided Tolerancing	23
2.3.1	Tolerance representation	24
2.3.2	Tolerance specification	29
2.3.3	Tolerance analysis and synthesis	29
2.3.4	Tolerance transfer	32
2.4	Efficiency Measures Regarding Dimensional Specifications	40
2.4.1	Efficiency measures for multi-finished rotational parts	40
2.4.2	Efficiency measures for multi-finished prismatic parts	42
2.4.3	Applicability of manufacturability criterion	44
2.5	Computer Aided Process Planning (CAPP)	44
2.6	Term Rewriting System and Functional Programming	45
2.6.1	Term rewriting system	45
2.6.2	Functional programming and Haskell	45
2.7	Summary	46
3	MANUFACTURABILITY REGARDING DIMENSIONING	47
3.1	Glossary	47
3.2	Critical Factors	49
3.3	Types of Design Dimensions	51
3.4	Dimension Set	52
3.5	References and Datums	52
3.5.1	Design datums	53
3.5.2	Operational datums	53
3.5.3	Locating datums	54
3.5.4	Measuring datums	54
3.5.5	Assembly datums	54
3.5.6	Principle of coincidence of datums	54
3.6	Minimizing Tolerance Stacks	55
3.6.1	Necessary tolerance stacks	55

3.6.2	Balancing process tolerances and changing operational datums . . .	55
3.7	Manufacturing Cost Considering Alternative Specifications	58
3.8	Summary	59
4	DESIGN REWRITING	60
4.1	Overall Dimension Graphs	61
4.1.1	States of dimensions	62
4.1.2	Construction of the initial overall dimension graph	64
4.2	Overall Dimension Set	66
4.2.1	Potential alternative design specifications	66
4.3	The Design Rewriting System	67
4.3.1	Definitions	68
4.3.2	Termination and confluence	70
4.3.3	Modules of the system	70
4.4	Rewriting Rules	72
4.4.1	Size dimensions	72
4.4.2	Parallelism	73
4.4.3	Perpendicularity	75
4.4.4	Angularity	76
4.4.5	Position	77
4.4.6	Over- and under-constraining a design	78
4.5	Searching Alternative Tolerance Chains	79
4.6	Example	81
4.7	Conclusion	82
5	CONSTRAINT GENERATION FOR REDIMENSIONING	83
5.1	Background	83
5.1.1	Interval arithmetic	84
5.1.2	The augmented torsor model	84

CONTENTS

5.1.3	Converting between size and geometric tolerances and augmented V-torsors	88
5.2	Tolerance Constraints	89
5.2.1	Parallelism	90
5.2.2	Size dimensions	94
5.2.3	Some augmented V-torsors	96
5.3	Examples	97
5.3.1	The prismatic part with two holes	97
5.3.2	The bracket with one hole	100
5.4	Conclusion	104
6	INTEGRATING AND IMPLEMENTING THE DESIGN REWRITING SYSTEM	106
6.1	Manufacturability Evaluation	106
6.2	The Integration of Design Rewriting System	108
6.3	Implementation of the Prototype DRS	112
6.3.1	Implementation of the rewriting module	114
6.3.2	Implementation of the constraints generation module	117
6.3.3	Implementation of the manufacturability evaluation module	118
6.3.4	Implementation of the interface module	118
6.4	An Application of the Prototype DRS	119
6.4.1	The interactive mode	121
6.4.2	The automatic mode	122
6.5	Evaluation of the Prototype DRS	123
7	CONCLUSION AND FUTURE WORK	125
7.1	Conclusion	125
7.2	Future Work	126
	BIBLIOGRAPHY	128

CONTENTS

INDEX	139
--------------	------------

ABSTRACT

To facilitate concurrent engineering, efficient and effective computer-based design assistant tools are imperative. However, tools based on current design for manufacturability approaches focus on the geometric and topological aspects of a design, like machining feature models of a design, feature approachability *etc.* Other aspects, like dimensions and tolerances, are not considered sufficiently. This thesis aims to narrow this gap by taking into account the impact of dimensioning and tolerancing on manufacturability. This is achieved by an approach which aids designers to (re)design for improved manufacturability.

A system modifying design dimension schemes through changing design datums and tolerance values of some dimensions via proper dimension chains is developed. This system mainly consists of four modules: a module for searching alternative design dimension chains; a module for constraint generation; a module for manufacturability evaluation, which assesses the manufacturability for derived design dimension schemes; as well as an interface. A prototype of the so-called Design Rewriting System demonstrates the proposed approach.

This system relies on a systematic approach investigating alternative dimensional specifications, and bases on a graph theoretical approach. Several concepts, such as explicit dimension, state of a dimension, and overall dimension graph, are discussed to facilitate the determination of alternative design dimension chains. In particular, a measure, that compares different types of geometric tolerances, is adopted as the manufacturability criterion to evaluate the manufacturability of generated designs.

Meanwhile, an augmented torsor model is adopted to mathematically represent tolerance zones and accommodate the propagation of both size and geometric tolerances.

In addition, manufacturability measures with regard to dimensioning are studied, and an

ABSTRACT

operation element based efficiency measure, as well as manufacturing cost are proposed and discussed.

The approach proposed in this thesis is validated and compared with existing methods in literature. The result of these comparisons shows that this approach maintains the functionality of the changed design and is applicable to more types of geometric tolerances than other approaches.

LIST OF FIGURES

2.1	Product development cycle in the ‘DFM wheel’ (modified from [Jo <i>et al.</i> , 1993])	18
2.2	Simplified traditional tolerance chart (adopted from [Britton <i>et al.</i> , 2001]) .	33
2.3	Drawing and 3-D model of the example part	36
2.4	Datum hierarchy tree for chart shown in figure 2.2 (modified from [Britton <i>et al.</i> , 2001])	38
2.5	A design with its design dimension tree (adopted from [Thimm <i>et al.</i> , 2004b])	40
3.1	Summary of dimensional tolerance concepts (adopted from [Bjørke, 1989])	49
3.2	Factors influencing the manufacturability of a design specification	50
4.1	Design dimension graph for the example part	62
4.2	Overall dimension graph (size dimensions)	63
4.3	Partial overall dimension graph (geometric dimensions)	64
4.4	Choices for replacing an explicit size dimension	67
4.5	The interaction diagram of the system	71
4.6	Activity diagram of searching alternative tolerance chains	80
5.1	Planes with parallelism constraints	90
5.2	Parallelism: axis to a datum plane	93
5.3	Size dimension and tolerance	94
5.4	Drawing and 3-D model of the second example part with labels and reference frames (adapted from [Xi <i>et al.</i> , 2001])	98

LIST OF FIGURES

6.1	Activity diagram of the interactive mode	109
6.2	Activity diagram of the automatic mode	111
6.3	Activity diagram of part of the automatic mode	113
6.4	Modified design dimension graph for the example part	122

LIST OF TABLES

2.1	Worst case versus root sum square (modified from [Trego, 1993])	32
2.2	Stock removal sets for figure 2.4	37
3.1	Types of design dimensions	51
4.1	The second rule set for parallelism	74
4.2	Replacing perpendicularities in the presence of two parallelisms	75
4.3	Replacing parallelism in the presence of perpendicularities	76
4.4	Replacing angularity of a line	78
5.1	Augmented V-torsors for some geometric dimensions	96
6.1	Evaluation of the prototype	123

LIST OF PUBLICATIONS

Lin Jietong and Georg Thimm. (2004b). Constraint Generation for Alternative Dimensional Specifications. *Computer-Aided Design & Applications*. 1(1-4):675-682.

Georg Thimm and Lin Jietong. (2004). Re-dimensioning Parts for Manufacturability: A Design Rewriting System. *International Journal of Advanced Manufacturing Technology*. ISSN 0268-3768 (Paper) 1433-3015 (Online).

Georg Thimm and Graeme A. Britton and Lin Jietong. (2004). Operation Element Based Efficiency for Process Plans and Designs. *International Journal of Advanced Manufacturing Technology*. 24(5-6):370-375.

Lin Jietong and G. Thimm and G. A. Britton. (2003a). Alternative geometric design specifications for manufacturability. In Qin, Y. and Juster, N., editors, *Advances in Manufacturing Technology - XVII: Proceedings of the First International Conference on Manufacturing Research (ICMR2003)*. pages 265-269. University of Strathclyde. Professional Engineering Publishing Limited, Bury St Edmunds and London, UK.

Lin Jietong and G. Thimm and G. A. Britton. (2003b). Alternative design specifications of prismatic parts for manufacturability. *Proceedings of The Institution of Mechanical Engineers Part B, Journal of Engineering Manufacture*, 217(4):569-572.

CHAPTER 1

INTRODUCTION

1.1 Motivation

In the marketplace of the 21st century, there is no place for traditional “over-the-wall” communication between designers and manufacturers. In order to “design it right the very first time”, designers must ensure that their products are functional and easy to manufacture.

Concurrent Engineering is recognized as a viable approach in which simultaneously the design of a product and its manufacturing process is taken into consideration. This ensures correspondences of the product’s structural and functional requirements with the associated manufacturing constraints [Jo *et al.*, 1993].

Four main aspects of product design and development justify the use of concurrent engineering [Stage, 1996]:

- Design decisions made early in the product development cycle can have a significant effect on the manufacturability, quality, product cost, product introduction time, and thus on the ultimate success of the product on the market.
- The more the development of a product is advanced, the more expensive it is to change its design. This means that the product designer must include manufacturing considerations as early as possible along with the structural, functional, aesthetic, and other life-cycle requirements.
- The cost of special tools and machines as well as lead time suggest that products must

be designed to fit the factory rather than changing the factory to fit the design. This is especially true for products manufactured in small batches.

- A product whose design is closely coupled with the process capacities of a factory has a reduced risk of quality defects which could be caused by uncontrollable process variations.

The transition from the sequential to concurrent engineering involves both cultural and technological changes. Organizational changes are particularly important in implementing concurrent engineering as a process, but they also have limitations. For example, when specialists from different functional areas are gathered in a cross-functional team to develop a product, there is an obvious advantage that they can contribute to the design of product and processes by early identification of potential problems and timely initiation of actions to avoid a series of costly reworks. However, in many situations it is simply not cost effective to maintain such a cross-functional team.

On the other side, as more sophisticated computer-aided design tools emerge, computer-based design for manufacturability approaches enhance the cross-functional team approach. However, current research and available computer-aided design tools focus on the influence of the geometry and topology of a design on the manufacturability, while the non-geometric aspects, like dimensioning and tolerancing, are not considered sufficiently, not to mention the benefits of redesign with respect to dimensioning and tolerancing.

This leads to the purpose of this work: to narrow this gap by developing a computer-aided redesign tool to increase the manufacturability of a design regarding its dimensional specifications.

1.2 Objective

Dimensioning and tolerancing affects both the functional quality of a design and the manufacturing processes used to convert the design into a part. Therefore, the objective of this thesis is to develop a methodology to automatically modify the dimensional scheme of a de-

1.3 Scope

sign by recommending alternative, more economical dimensional specifications, which still meet the original functional requirements.

1.3 Scope

In order to limit the thesis within a reasonable scope, the thesis focuses on:

- ◇ *Prismatic parts*, i.e. parts which consist of an arbitrary number of sets of parallel, planar surfaces and/or cylindrical surfaces.
- ◇ Only the effects of *dimensioning and tolerancing* on manufacturability are taken into account. The influence of geometric and topological aspects on manufacturability are not considered.
- ◇ The *functionality* of a part is, though considered, not directly evaluated. That is, when evaluating the manufacturability of the design or suggesting redesign options, information on the functionality of a part are assumed to be given.

1.4 Contributions

Contributions are:

1. A redesign methodology for changing dimensional scheme of a design is presented.
 - (a) Alternative dimensional specifications are investigated. A graph theoretical approach to replace dimensions by alternative dimension chains is proposed. The approach propose alternative dimension chains and generate constraints for the substitution (published in [Jietong *et al.*, 2003a]).
 - (b) As part of the design rewriting system, sets of rewriting rules for both size and GD&Ts (acronym of Geometric Dimensioning & Tolerancing) are elaborated ([Thimm and Jietong, 2004b]). Based on these rules, size and/or geometric dimensions, which reduce manufacturability, can be replaced. Meanwhile, feasible

alternative dimension chains, if any, are identified as candidates to substitute these identified problematic size and/or geometric dimensions.

- (c) A constraint generation approach, as another important component of the design rewriting system, is developed ([Jietong and Thimm, 2004]). In this approach, constraints of tolerance values for involved dimensions can be created parallel to the application of design rewriting rules.
 - (d) The whole approach of the design rewriting system is presented ([Jietong *et al.*, 2003b; Thimm and Jietong, 2004a]). In this part, four modules of this system are integrated.
2. Manufacturability measures are studied, and new criteria to evaluate the manufacturability regarding dimensioning are proposed.
- (a) A manufacturability measure, operation element based efficiency measure, is proposed ([Thimm *et al.*, 2004a]). This measure can serve as tools to estimate production cost for designers or measures of efficiency of a given process plan compared to the theoretically optimal process plan.
 - (b) As a manufacturability measure, manufacturing cost, is discussed, which can be applied when sufficient information on cost-tolerance relationship is available.
 - (c) The normalized tolerance is defined as a manufacturability measure to assess the tightness of size and geometric tolerances.
3. A prototype for a design rewriting system (DRS) is implemented in Haskell, a functional programming language. Details of the implementation and an application of the developed prototype via an actual example are presented.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Design for manufacturability (DFM) is a philosophy that considers the products' life-cycle requirements simultaneously in the early phase of the design to achieve a high product quality with reduced cost and shorter lead time [Chen *et al.*, 1995]. The DFM philosophy plays a vital role since the late 70s [Ramaswamy, 1998], and dates back to the Second World War or earlier [Ziemke and Spann, 1993].

The life-cycle issues of products include functionality, manufacturability, assemblability, process planning, testability, cost estimation, ergonomics, serviceability, recyclability, *etc.*. Refer to the 'DFM Wheel' in figure 2.1 [Jo *et al.*, 1993; Chen, 1994]. In this thesis, the integration of the design and manufacturing is focused on, excluding other life-cycle aspects of a product, *e.g.* assemblability and recyclability.

Important advances have been made in the field of DFM, and several manufacturability evaluation (ME) approach and design tools were developed, such as those reported in [Anderson, 1990; Venkatachalam *et al.*, 1993; Shankar and Jansson, 1993; Chen *et al.*, 1993; Youssef, 1994; Gupta and Nau, 1995; Hayes, 1995; Chen *et al.*, 1995; Ong and Chew, 2000; Jacquel and Salmon, 2000]. However, most current manufacturability evaluation methods have limited capabilities for analyzing the manufacturability of dimensioning and tolerancing of a design.

Tolerances are the results of the inherent variations of manufacturing processes, and it is

Figure 2.1: Product development cycle in the ‘DFM wheel’
(modified from [Jo *et al.*, 1993])

widely recognized that tolerance plays an ubiquitous role in both product design and process planning [Srinivasan, 1994], and that dimensioning and tolerancing have a strong impact on the cost and lead time of a product. However, no available ME methods provide the benefits of semi-automatically rewriting the dimensional scheme of a design to improve its manufacturability. Moreover, current ME approaches lack a clear, computationally graspable link from the design to a process plan [Thimm *et al.*, 2004a]. Our approach attempts to overcome these shortcomings.

In the following, firstly, an overview to DFM is presented. Secondly, a survey on computer-aided tolerance techniques is provided, followed by a brief review on computer aided process planning (CAPP). Finally, as the theoretical basis of the automation system proposed in the later of this thesis, the term rewriting system and functional programming is concisely introduced.

2.2 Design for Manufacturability (DFM)

2.2 Design for Manufacturability (DFM)

2.2.1 Basic approach to implement DFM

In an application of DFM, the cross-functional team is the form to practice [Jo *et al.*, 1993]. It is human-oriented in that the team consists of designers and individuals from various related functional areas. Team members are selected for their contribution to the design of product and processes by an early identification of potential problems and a timely initiation of actions to avoid a series of costly reworks [Pennell *et al.*, 1989]. The cross-functional team is crucial for effective implementation. Continuously developed computer technologies, in both hardware and software, have given team members from different departments the ability to work on the same design to evaluate the effects of design attributes. In this context, it is widely accepted that it is compulsory to have each team member fully understand the philosophy of DFM.

While the team-based approach can be readily implemented and is widely adopted in industry, some shortcomings appear to arise [Jo *et al.*, 1993]:

1. Difficulties to manage the team effectively;
2. Team members' limited knowledge;
3. The cost of maintaining a team.

As more sophisticated computer tools emerge, the team-based methodology is being enhanced by the computer-based methodology in which the DFM philosophy is respected in CAD tools, enabling design justification or optimization with respect to the entire aspects of a product's life-cycle. The existing computer-based ME approach can be classified roughly as follows [Gupta *et al.*, 1997]:

- *Direct* approach: rules are used to identify infeasible design attributes directly from inspection of design descriptions. This approach is useful in domains such as near-net shape manufacturing. However, it is less suitable for machined or electro-mechanical

components, in which interactions among manufacturing operations can make it difficult to determine the manufacturability of a design directly from the design description.

- *Indirect* approach: the first step is to generate a manufacturing plan, and modifying various portions of the plan in order to reduce its cost. However, in the process of evaluation, only process plans are changed, no explicit changes on dimensioning are made to the original design to achieve a better manufacturability.

2.2.2 Automatic generation of redesign suggestions

To make a manufacturability evaluation approach effective, it is not always adequate to have the manufacturability rating of a design and a list of its production bottlenecks, which can result in a lower manufacturability. Since designers often are not specialists in manufacturing processes, they may not be able to rectify the problems identified by the manufacturability evaluation system. Consequently, manufacturability analysis systems have to be complemented by a system suggesting ways to redesign.

Hayes [1995] described a methodology for relaxing positional tolerances to reduce machining cost. This methodology takes into consideration the initial conditions of the stock like the surface conditions. It attempts to remove setups by relaxing positional tolerances between surfaces. However, it neither considers size and geometric tolerances other than positional tolerances, nor does it take into account the cases where initial positional tolerances should be respected (*i.e.* cannot be relaxed).

Approaches for generating redesign suggestions, as those reported in [Huh and Kim, 1991; Ishii, 1993; Das *et al.*, 1996], proposed design changes to the features (geometry) of a design, but not to dimensional schemes.

2.2.3 Measures of manufacturability

Many criteria – or combinations of criteria – were proposed on which manufacturability can be measured [Gupta *et al.*, 1997]:

2.2 Design for Manufacturability (DFM) ---

- *Binary measures.* These are the most basic kind of manufacturability rating: they simply report whether or not a given set of design attributes is manufacturable.
- *Qualitative measures.* Design are given qualitative grades based on their manufacturability by a certain production process. For example, Ishii *et al.* [Ishii, 1993] rated designs as “poor”, “average”, “good”, or “excellent”.
- *Abstract quantitative measures.* This type of schemes involve rating a design by assigning numerical ratings along some abstract scale. For example, Shankar *et al.* [Shankar and Jansson, 1993] proposed a scheme, in which each design attribute is assigned a manufacturability index between 1 and 2.
- *Time and cost.* In general, a design’s manufacturability is a measure of the effort required to manufacture the part according to the design specifications. Ratings based on time and cost present a view of the difficulty in manufacturing a proposed design. However, the cost data are site-dependent and in most cases, collecting sufficient manufacturing cost data is very hard [Chase, 1999].

2.2.4 State-of-the-art DFM

The manufacturability of a design depends strongly on the manufacturing processes selected to create it. For example, a design that has an ideal shape for casting may not be suitable for machining. Hence, approaches to computer-aided manufacturability analysis are strongly influenced by the type of manufacturing processes they address. So far, most of the approaches have limited ability to handle multiple processes [Gupta *et al.*, 1994], although some advances have been achieved in individual domains, respectively. Some manufacturing domains and representative systems are illustrated below.

Assembly

Most of the early work in the analysis of assemblability is rule-based. However, as it was found that an assembly sequence has an important influence on the assemblability, plan-

based evaluation systems were developed by Boothroyd and Dewhurst [1983], Swift [1981], Sturges and Kilani [1992], Li and Huang [1992], and so on.

DFMATM (Design for Manufacture and Assembly), developed by Boothroyd and Dewhurst, originated as a design for assembly method but was developed to include a wider range of activities in the design process [Boothroyd, 1994]. Design for assembly (DFA) still plays a major role in this method; however, it is the companion DFM methods within DFMA that help judge the contribution of DFA. The following are the stages in the DFMA method:

1. DFA with feedback to concept development stage about simplification of product structure.
2. Selection of materials, processes and early cost estimation.
3. Detailed design for manufacture, based on the best design concept.

Once DFA has been applied, a manufacturability evaluation is carried out to assess manufacturability and cost. The parts are designed and part costs are estimated at early stages using various combinations of processes and materials. A combination of processes and materials suitable for a design is chosen and this enables a thorough detailed analysis of a design.

Near-net shape manufacturing

Near-net shape processes produce parts in a way which allows the use of rules to relate design to manufacturing attributes. Hence, rule based systems are very successful in this case, such as Huh and Kim [Huh and Kim, 1991], Wozny *et al.* [Wozny *et al.*, 1993], Dessinger *et al.* [Dissinger and Magrab, 1994].

Machining

In this domain, initially, it was attempted to relate different attributes of a design to the manufacturing process in such a way to use direct rules to find the manufacturability [Venkatachalam *et al.*, 1993; Chen *et al.*, 1995; Ong and Chew, 2000]. Because of the very nature of the machining process, *e.g.* the dependencies and influences among different operations,

2.3 Computer Aided Tolerancing

it is hard to isolate instances to apply these rules. Therefore, an indirect approach is more suitable, and an accurate estimation of the manufacturability of a design must be based on a process plan.

Gupta and Nau have developed an approach for the manufacturability analysis of machined parts [Karinthi and Nau, 1992; Karinthi *et al.*, 1992; Gupta and Nau, 1995; Regli *et al.*, 1995]. Parts are interpreted as a collection of machining features, a machining feature being a portion of the workpiece achieved by a machining operation. The interpretations are mapped onto process plans and a manufacturability analysis is carried out on these plans. The approach assumes that the design is available as a solid model, including tolerances and surface finish. It is also assumed that achievable tolerances and surface finishes of available machining operations are known. The most important steps of the method are [Regli *et al.*, 1995]:

1. Generate all the possible machining features for the designed part; each feature representing a machining operation. If it is not possible to machine some features, the part is not machinable.
2. A feature based model (FBM) is created from the set of features in step 1. A FBM is a non-redundant set of machining features. An iterative procedure yields the most promising FBM, the one with the best process plan. A good process plan is one that
 - (a) does not violate best machining practices,
 - (b) is able to produce required design tolerances and surface finishes, and
 - (c) meets given production time and costs.

2.3 Computer Aided Tolerancing

Research in computer aided tolerancing can be classified in one of the following fields:

1. Tolerance representation,
2. Tolerance specification,

3. Tolerance analysis and synthesis, and,
4. Tolerance transfer.

These four distinct, but closely related, topics are elaborated in the following.

2.3.1 Tolerance representation

First, in this section, ASME Y14.5M (1994) is briefly reviewed. It is followed by the introduction to the four tolerance representation models. Then, after a summary, the tolerance representation model adopted in this thesis is specified.

Geometric tolerances and ASME Y14.5M (1994)

Conventional and geometric tolerances are usually distinguished in tolerancing. Conventional or size tolerances specify upper and lower limits of dimensions and angles; they are often referred to as plus/minus tolerances. Geometric tolerances are subdivided into *form*, *profile*, *orientation*, *location* and *runout* tolerances. The main difference between the two is that a geometric tolerance has a type and a value as well as a related datum system while conventional tolerances have only a value and no associated datum system. Fourteen types of geometric tolerances exist:

- Form includes *Straightness*, *Flatness*, *Circularity* and *Cylindricity*,
- Profile includes *Profile of a line* and *Profile of a surface*,
- Orientation includes *Angularity*, *Perpendicularity* and *Parallelism*,
- Location includes *Position*, *Concentricity* and *Symmetry*,
- Runout includes *Circular runout* and *Total runout*.

More specifically, a size tolerance is often considered to permit one degree of freedom to the target entity, *i.e.* displacement only in the direction of the dimension. For geometric tolerances, the allowed variation is defined as a *tolerance zone*, in which the target entity is

2.3 Computer Aided Tolerancing

permitted several degrees of freedom of displacement or deformation. The location of the tolerance-zone is constrained with a basic dimension; its size and internal characteristics are determined by the tolerance values, such as size, location, and orientation. The standard allows for the interactions between certain tolerances, such as location and size [Davidson and Shah, 2000].

In [ASME Y14.5M, 1994] exists a basic tolerancing principle, the envelope principle, whereby size tolerances have precedence over all other types of geometric tolerances. More precisely, the envelope principle states that where a tolerance of size is specified, the limits of size of the individual features prescribe the extent to which variations of its geometric form as well as size are allowed.

The basic tolerance principle has three exceptions: the maximum material condition (MMC), the least material condition (LMC), and the regardless of feature size (RFS).

The MMC condition, from a design point of view, is suitable for verifying the feasibility of an assembly in a worst case scenario. When the MMC condition holds, a feature of size contains the maximum amount of material within the stated limits of size. A dependence between dimensional and positional defects is assumed in the case of a MMC condition.

In case of the LMC condition, a feature of size contains the least amount of material within the stated limits of size. Commonly, a designer uses this to determine the maximum clearance in an assembly. The LMC condition also guarantees minimum thickness of material necessary to ensure proper strength and functionality of a part.

The RFS condition applies to any increment of size of the feature within its size tolerance. This means that the geometric tolerance is limited to a specified value, regardless of the actual feature size. In design practice the RFS condition are often used where balance is important; it is independent between dimensional and geometric tolerances when these apply to different technical functions.

Offset model

The offset model proposed by Requicha [1983] is one of the earliest methods to construct tolerance zones. Minkowski sum [Kaul, 1992], a mathematical morphological operation

CHAPTER 2. LITERATURE REVIEW

is used to construct non-parametrically defined zones. Requicha put forward a theory and proposed that with Minkowski sums, the tolerance zones can be elegantly constructed to contain all the objects of a variational class.

However, this approach is incompatible with existing tolerancing standards [Salomons *et al.*, 1996]. Because the individual pairs of offset surfaces are combined to obtain a composite tolerance zone of the entire solid, the individual tolerances cease to be independent.

Jayaraman and Srinivasan introduced the notion of virtual boundary requirement in an attempt to redefine the notion of tolerances from the point of view of functional tolerances [Jayaraman and Srinivasan, 1989; Srinivasan and Jayaraman, 1989], which is regarded as a specialization of Requicha's offset zone theory. Jayaraman and Srinivasan note that the real purpose of tolerances is to characterize functional requirements. Two functional constraints are cited that are believed to account for most tolerances: the maintenance of material bulk in critical locations and spatial relationships for assembly. These requirements can be captured as virtual boundary requirements. Virtual boundary requirements can be considered as a collection of virtual half spaces. The interpretation of datums in this approach is not in accordance with the standards.

Parametric space models

In these models, the variational class that represents the tolerance zone is mapped to a different parametric space which was first introduced by Turner [Pasupathy *et al.*, 2003]. This technique is considered as an extension to parametric tolerances, wherein imperfect-form with imperfect orientation of parts can be accommodated as opposed to the perfect-form with perfect orientation accommodated by parametric tolerances. Some researchers [Roy and Li, 1999; Teissandier *et al.*, 1999] have included several refinements and they consider this technique to be useful for analysis and synthesis methods.

Algebraic methods

Turner's M-space theory of form tolerances was limited to single tolerance control of a feature, such as size or position tolerance. Roy and Li [1998] consider the resultant zones from

2.3 Computer Aided Tolerancing

multiple controls, such as size and form tolerances, on a single feature.

Torsor model

The torsor concept has originated in the field of metrology [Salomons, 1995]. As in [Legoff *et al.*, 1999; Villeneuve *et al.*, 2001], the main idea of the torsor model is that the displacement of a rigid body or of a surface, except for expected movements, are assumed to be small with regarding to the nominal dimensions. Then, the small rotation can be linearized in a first order approximation. The small displacement torsor of one design element a in relation to another design element b with respect to a reference system ($\mathfrak{R}_i$) at the origin of this reference system (point O_i) is defined as (modified from [Villeneuve *et al.*, 2001]):

$$T_{a,b}|_{\mathfrak{R}_i O_i} = [\omega \ v]|_{\mathfrak{R}_i O_i} \quad (2.1)$$

where $\omega = [\omega_x, \omega_y, \omega_z]^T$ is a small rotation vector (superscript T represents vector transpose) with no change at the considered point of element a , and $v = [v_x, v_y, v_z]^T$ a small displacement vector for point O_i where the value changes with the considered point. In the following of this thesis, unless stated otherwise, the default point at which the torsor is expressed is the origin of the reference system. Therefore, $T_{a,b}|_{\mathfrak{R}_i O_i}$ is represented by $T_{a,b}|_{\mathfrak{R}_i}$.

The actual little translation $v|_{\mathfrak{R}_i M}$ of any point M of element a is obtained by [Villeneuve *et al.*, 2001]:

$$v|_{\mathfrak{R}_i M} = v|_{\mathfrak{R}_i O_i} + \omega|_{\mathfrak{R}_i} \times \overrightarrow{O_i M} \quad (2.2)$$

where $v|_{\mathfrak{R}_i O_i}$ is the small translation vector of point O_i in reference system $\mathfrak{R}_i$ as explained in equation 2.1, and $\omega|_{\mathfrak{R}_i}$ the small rotation vector in reference $\mathfrak{R}_i$ as defined in equation 2.1. “ $\times$ ” is the cross product, and $\overrightarrow{O_i M}$ is the translation of point O_i to point M in reference system $\mathfrak{R}_i$.

Two types of torsors are used [Bourdet and Ballot, 1996]:

1. *Deviation torsors* (D-torsors) represent variations between nominal and actual features and

2. *Variation torsors* (V-torsors) denote relative variations between two or more real features of a part.

In order to allow the accumulation of several deviation or variation torsors (using matrix additions), they have to be expressed in a common reference system. To this means, equation (2.3) expresses the propagation of the torsor (2.1) from its current reference frame to another reference frame [Veitschegger and Wu, 1986; Villeneuve *et al.*, 2001]:

$$T_{a,b}|_{\mathcal{R}_j} = \left[\hat{\omega}_{i \rightarrow j} \cdot \omega|_{\mathcal{R}_i} \quad \hat{\omega}_{i \rightarrow j} \cdot \left(v|_{\mathcal{R}_i} + \omega|_{\mathcal{R}_i} \times \overrightarrow{O_i O_j} \right) \right] \quad (2.3)$$

where $\overrightarrow{O_i O_j}$ describes the translation of the origin of reference frame $\mathcal{R}_i$ to the origin of $\mathcal{R}_j$ (relative to the earlier frame's origin), and the matrix $\hat{\omega}_{i \rightarrow j}$ rotates $\mathcal{R}_j$ onto $\mathcal{R}_i$.

For each size and geometric tolerance, the tolerance zone can be represented as a torsor [Salomons, 1995]. Instead of torsors, matrices can also be used to develop tolerance zones [Desrochers and Rivière, 1997].

Summary

Many techniques, like the offset model presented above, do not comply with international standards. For human designers, these techniques are not fit to replace the present tolerancing standards because they are not expressive enough; their “syntax and semantics” are too limited. Moreover, it will be hard to introduce a completely new and different tolerancing standard on a world wide scale even if it is expressive enough for human use as well as to provide for a mathematically correct model for use in computer systems. It seems more appropriate to search for a theoretical model of tolerancing, or tolerance representation scheme, that is in accordance with the international standards, like ASME Y14.5M [1994].

The torsor model is compatible with present standards and seems to be theoretically and mathematically sound [Clément *et al.*, 1997; Teissandier *et al.*, 1999]. Furthermore, this model can not only represent tolerance zones, but also handle the problem of tolerance propagation calculations in contrast to other models [Whitney *et al.*, 1994; Salomons *et al.*, 1996; Desrochers and Rivière, 1997; Lin and Zhang, 2001]. Therefore, this model is adopted in this thesis.

2.3 Computer Aided Tolerancing

2.3.2 Tolerance specification

Tolerance specification is a method to specify tolerances, *i.e.* define the tolerance types (refer to section 2.3.1) and the related tolerance values. Tolerance specification is preferably carried out in conformance with the tolerancing standards ([ISO 1101, 1983], [ASME Y14.5M, 1994], national, or company specific). Only little research has been performed on this subject. The main contributions are described in a number of publications by A. Clément et al. Clément's method of tolerance specification as presented in [Clément *et al.*, 1997; Desrochers and Maranzana, 1996] is different from most previous approaches to tolerance specification. Clément et al. proposed a method to determine tolerance *types* automatically from the assembly model, resulting in a truly functional tolerance specification. Most other approaches in computer aided tolerancing are based on manual tolerance specification, usually starting out from a single component.

The tolerance specification method by Clément et al. is carried out on the basis of face associations between the different components in the assembly. This is also called the mating function [Whitney *et al.*, 1999]. By finding kinematic loops (in the graph representing the assembly), features are found on individual components which can be toleranced relative to one another [Salomons, 1995].

2.3.3 Tolerance analysis and synthesis

Tolerance analysis verifies the proper functionality of the assembly after tolerances have been specified. In tolerance analysis, all component tolerances are known or specified and resultant variations are calculated to ensure the resultant variations are within the design tolerance.

During *tolerance synthesis*, the assembly's functional tolerances are allocated to individual components, thus ensuring that the assembly requirements are met [Hong and Chang, 2002]. In contrast to tolerance analysis, in tolerance synthesis, the resultant tolerance is known from design requirement, whereas the magnitude of the component tolerances to meet these requirements are unknown.

Existing tolerance synthesis approaches can be classified into five categories: [Chase and Greenwood, 1988; Teck, 1996; Chase, 1999; Hong and Chang, 2002]

- *Proportional scaling*: the designer begins by assigning reasonable component tolerances based on process or design guidelines. Then, the related component tolerances are accumulated to see if they meet the specified design tolerance. If not, the component tolerances are scaled by a constant proportionality factor [Chase and Greenwood, 1988].
- *Weight factors*: using this algorithm, the designer assigns weight factors to each tolerance in the chain and the system distributes a corresponding fraction of the tolerance pool to each component. A larger weight factor for a given component means a larger fraction of the tolerance pool will be allocated to it [Chase, 1999].
- *Least cost optimization*: based on cost-tolerance models, the optimization algorithm varies the tolerance and searches systematically for the combination of tolerances which minimize the cost [Dong *et al.*, 1994].
- *Taguchi's approach to tolerancing* : tolerancing, in Taguchi's context, is defined as the process of establishing manufacturing tolerances (low level tolerance) based on customer tolerances (high-level tolerances). High level tolerances are always expressed as a value that is directly determined by customers. Low-level tolerances are expressed in highly technical engineering terms that are direct expressions of the physical measures that lie behind the performance the customer sees and expects [Creveling, 1997].
- *Methods based on genetic or evolutionary algorithms, simulated annealing, and/or neuro-fuzzy learning*: These relatively new techniques are applied in tolerance synthesis [Ji *et al.*, 2000; Laperriere and Kabore, 2001].

Models for size tolerance stackup

The basis for rational tolerance specification is to create an analytical model to predict the stackup of tolerances in an assembly process. In order to relate the functional tolerance (vari-

2.3 Computer Aided Tolerancing

ation of the dependent variables) to the component tolerances (variation of the independent variables), it is required that the functional relationships between the dependent and independent variables are identified and available. These functional relations are expressed as in equation (2.4) (modified from [Greenwood, 1987])

$$U = f(x_1, x_2, \dots, x_n) \quad (2.4)$$

where U is the predicated assembly dimension, and x_i are nominal component dimensions.

Common models for prediction of component tolerance stackup are the worst case model (see equation (2.5)) and the root sum square model (equation (2.6)) (modified from [Chase and Parkinson, 1991]).

$$\Delta U = \sum_{i=1}^n \left(\left| \frac{\partial f}{\partial x_i} \right| T_i \right) \leq T \quad (2.5)$$

$$\Delta U = Z \left[\sum_{i=1}^n \left(\frac{\partial f}{\partial x_i} \right)^2 \left(\frac{T_i}{Z_i} \right)^2 \right]^{\frac{1}{2}} \leq T \quad (2.6)$$

where T is the specified tolerance, x_i are still nominal component dimensions as in equation (2.4), T_i are component tolerances, ΔU is the predicted assembly variation, $\left| \frac{\partial f}{\partial x_i} \right|$ is the absolute value of $\frac{\partial f}{\partial x_i}$, Z is the number of standard deviations desired for the specified assembly tolerance, Z_i describes the expected standard deviations for each component tolerance, and f is the function described in equation (2.4).

The worst case model assumes all the components occur at their worst limit simultaneously. It is used by designers to assure that all assemblies will meet the specified assembly limit, and it is relatively easy to calculate. However, the worst case model is rather pessimistic, and as the number of parts in an assembly increases, the component tolerances must be greatly reduced in order to meet the assembly limit, requiring high production costs.

In the root sum square model, the low probability of the worst case combination occurring is taken into account statistically, usually assuming a normal or Gaussian distribution for component variations. Tolerances are commonly assumed to correspond to six standard deviations (6σ or $\pm 3\sigma$), *i.e.* $Z = Z_i = 3$. Component tolerances generally may be increased.

However, root sum square analysis assumes a normal distribution and is often too optimistic. This is due to the fact that the normal distribution is only an approximation of the

true distribution, which may be flatter or skewed. To account for these uncertainties, a more general form of the root sum square model is used (adapted from [Chase and Parkinson, 1991]):

$$\Delta U = C_f Z \left[\sum_{i=1}^n \left(\frac{\partial f}{\partial x_i} \right)^2 \left(\frac{T_i}{Z_i} \right)^2 \right]^{\frac{1}{2}} \leq T \quad (2.7)$$

where C_f is a correction factor, which accounts for any non-ideal conditions. Typically values for C_f range from 1.4 to 1.8 [Chase and Parkinson, 1991]. Table 2.1 summarizes the worst case model and the root sum square model as well as their advantages and shortcomings.

<i>Model</i>	<i>Advantages</i>	<i>Shortcomings</i>
Worst Case	Assures 100% assembly acceptance if all component tolerances are within specification. Easy to calculate.	Costly design model. Requires excessively tight component tolerances.
Root Sum Square	Permits looser component tolerances, less costly. May adjust Z and Z_i to obtain desired quality.	Usually assume normal distribution for assembly and component errors. A fraction of assembly does not meet specifications.

Table 2.1: Worst case versus root sum square (modified from [Trego, 1993])

2.3.4 Tolerance transfer

Tolerance transfer aims to transfer tolerance requirements from the design stage to the manufacturing stage, while ([Hong and Chang, 2002])

1. Process capabilities of machinery are respected;
2. The allowance given in the design is fully used, and

2.3 Computer Aided Tolerancing

3. Machining errors, set-up errors, tool wears, etc. are taken into account.

In literature, tolerance transfer is also referred to as tolerance analysis/synthesis (process planning) or operational dimensioning and tolerancing. The most classical and widely used technique for tolerance transfer is tolerance charting [Wade, 1967, 1983].

In the following, first, the traditional tolerance chart will be introduced by an example, and then, datum-hierarchy tree technique is illustrated.

Figure 2.2: Simplified traditional tolerance chart (adopted from [Britton *et al.*, 2001])

Traditional tolerance chart An example of a traditional tolerance chart is shown in figure 2.2. A sketch of the finished part and its dimensions is shown at top center of the chart. Vertical lines (shown dashed in figure 2.2) are drawn from points, lines or surfaces of the part sketch. These lines are numbered from left to right as shown.

The design dimensions and the process plan are guided by the vertical lines. A line may represent more than one surface. For instance, line number 4 has three arrowheads attached

CHAPTER 2. LITERATURE REVIEW

to it, which signifies that it represents three different surfaces (one finished and two pre-formed surfaces). The surfaces represented by each line form a *stock removal set*. Each consecutive pair of surfaces in a stock removal set is related through a stock removal, *i.e.* the difference in position between two consecutive surfaces is the stock removal allowance for the second surface in the pair [Britton *et al.*, 2001].

The process plan is shown as an ordered sequence of operations from top to bottom. An arrow represents an operation. The tail of the arrow stands for the datum surface for the operation and the head indicates the machined surface. Design dimensions are entered at the bottom left of the chart. Resultant dimensions, which can be calculated from the process plan, are placed at the bottom right of the chart. Comparisons are carried out to check whether resultant dimensions from the process plan are able to meet design dimensions.

Three types of calculation are performed using the tolerance chart [Britton *et al.*, 2001]:

1. *Calculate tolerance stackups on design dimensions.* This type of calculation guarantees the machining tolerance stackups are inferior to the design specifications. This is a design feasibility check.
2. *Calculate tolerance stackups on stock removal allowances.* This type of calculation ensures that the machining tolerance stackups between consecutive pairs of surfaces in stock removal sets do not exceed the stock removal allowances. This is a machining feasibility check.
3. *Calculate working dimensions.* Working dimensions are intermediate dimensions to be obtained by the machining process in the process sequence.

As stated in [Britton *et al.*, 2001], the limitations of the tolerance chart are:

1. Individual machined surfaces are not clearly indicated as one vertical line often represents more than one surface;
2. The datum sequence is not obvious;
3. The chaining technique for identifying operations contributing to tolerance stackups is tedious and error-prone.

2.3 Computer Aided Tolerancing

These disadvantages are overcome by the datum-hierarchy tree technique, which is introduced in the following section.

Datum-hierarchy trees

The rooted (datum-hierarchy) tree technique for process planning was developed by Whybrew and Britton [Whybrew *et al.*, 1990; Britton and Whybrew, 1997]. The datum hierarchy tree technique is based on, and augments, the traditional tolerance chart [Britton *et al.*, 2001].

Datum-hierarchy trees overcome the problems of traditional tolerance charts, and have several advantages [Britton and Whybrew, 1997]:

- They are fully compatible with existing commercial CAD software;
- They are an extremely efficient technique for representing tolerance stacks and performing tolerance stackup calculations;
- A graphical illustration of a tree clearly highlights critical datum surfaces and is an invaluable aid for planning and modifying a manufacturing sequence.

Surface labels A key feature of the datum-hierarchy technique is the surface representation scheme. In this scheme, each surface is uniquely labeled [Britton and Whybrew, 1997].

Machining removes material from a surface of the workpiece. The change in dimension before and after machining is the *stock removal allowance*, which is a nominal dimension (it has no tolerance). It is the amount of material that was left on a surface for the next machining operation on that surface. A *stock removal set* is a set of surfaces that meets the following condition: for all surfaces in the set, the difference in position between each consecutive pair of surfaces is equal to the stock removal allowance for the second surface in the pair. The first surface in the set is a newly created surface, a machined surface or a surface created by a prior pre-forming operation, such as casting or forging. The last surface in the set is a finished surface.

As in [Britton *et al.*, 2002], a reference coordinate system is arbitrarily located at the bottom left hand end of a model of the part (finished part model and its drawing, figure 2.3).

Figure 2.3: Drawing and 3-D model of the example part

2.3 Computer Aided Tolerancing

<i>Stock removal set</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
	1X0,	2X0,	3X0,	4X0,	5X0,
Surface(s) in set	1X1,	2X1,	3X1,	4X1,	5X1,
	1X2	2X2	3X2	4X2	5X2
Finished surface	1X2	2X2	3X2	4X2	5X2

Table 2.2: Stock removal sets for figure 2.4

The reference coordinate system provides a frame of reference for defining machined surfaces in three dimensional space during process planning. Each surface created during manufacturing is uniquely identified by an alphanumeric code, consisting of 3 parts:

- A non-empty sequence of numbers indicating the stock removal sets a surfaces belonging to, separated from each other by a dot;
- A letter code, which is either X, Y, or Z for surface perpendicular to the respective axis or a combination of these for inclined surfaces or holes;
- A number indicating the position of the surface in the stock removal set. This is “0” for a new surface.

Please note that:

- ◇ The above scheme is developed for three dimensional parts;
- ◇ This scheme is different from the original labels developed by Whybrew & Britton;
- ◇ A more elaborate naming scheme is possible to accommodate for other, non-flat surfaces;
- ◇ The letter code is a convenience for the planner to allow an easy identification of parallel surfaces.

That each surface is uniquely labelled is essential to the rest of the whole technique in that it permits to construct an acyclic, directed, datum hierarchy graph (a rooted tree in the case of uni-dimensioning tolerance stackup, see figure 2.4).

Figure 2.4: Datum hierarchy tree for chart shown in figure 2.2 (modified from [Britton *et al.*, 2001])

Datum-hierarchy trees & graphs A *datum-hierarchy tree* consists of nodes that correspond to surfaces (or other kind of geometric entities, for example a point as in an arc center and a line as in a hole or shaft axis), and edges corresponding to working dimensions and at the same time machining operations, where the parent node represents the datum surface and the child node the machined surface [Thimm *et al.*, 2004b] (see figure 2.4).

As stated in [Britton *et al.*, 2001], in order to obtain a dimension (that is a working dimension, a stock removal allowance, a balance dimension or a resultant dimension), the distance on the tree between the corresponding two surfaces (nodes) must be calculated. The tolerance stackup on a dimension equals the sum of the tolerances of the operations (edges) on the shortest path between corresponding two surface nodes. If the nodes are finished surfaces, the tolerance stackup is on a design dimension (resultant tolerance). Finished surfaces are the last surfaces in the stock removal sets. If the nodes are consecutive surfaces in a stock removal set, the tolerance stackup is on the stock removal [Britton *et al.*, 2001].

As an example, the tolerance stackup on the second design dimension on figure 2.2, which is represented by line 2 and 4, is calculated. The finished surface in stock removal sets 2 and 4 are 2X2 and 4X2, respectively. The path between these nodes is 2X2 – 5X2 – 4X2 (the nodes surrounded by boxes in figure 2.4), the stack contains operations 13 and 11 (figure 2.2), and the stackup is $0.025 + 0.050 = 0.075$ mm.

A datum-hierarchy tree is a special case of a datum-hierarchy graph, in which more than one datum-hierarchy tree may exist. As addressed in [Thimm *et al.*, 2004b], a specified process plan has one and only one datum-hierarchy graph representation. However, the

2.3 Computer Aided Tolerancing

reverse is not true. A specified datum-hierarchy graph represents a set of process plan; the size of these sets can be enormous [Thimm *et al.*, 2001].

Design dimension trees & graphs A *design dimension tree* is the natural result of properly dimensioning a design of a rotational part [Thimm *et al.*, 2004b].

As described in [Thimm *et al.*, 2004b], a design dimension tree does not have a specified root node, which is contrast to a datum hierarchy tree. In figure 2.5, the left hand side shows the dimension scheme and surface labeling of a design, and right hand side shows the design dimension tree. A design dimension tree is an undirected graph. One feature of the design dimension tree is that each surface has a unique label (number). In addition, dimensions not parallel to the axis of the part, are not considered. Furthermore, as stated in [Thimm *et al.*, 2004b], the graph created for the design dimensions of a rotational part is always a tree, and there is more than one way to draw the design dimension tree; the alternative trees are isomorphic.

A design dimension tree is a special case of a design dimension graph, which is the natural result of properly dimensioning a design of a general part, and not only a rotational part [Thimm *et al.*, 2004a].

Finishing, pre-finishing, pre-forming, type-1, and type-2 machining operations A *finishing machining operation* is the last operation in a stock removal set. *Pre-finishing operations* are operations needed to achieve a specified surface finish and/or tolerance or to reduce the cost of finishing operations. The order of these operations in a stock removal set are pre-forming operations, pre-finishing operations and finishing operations.

Type-1 and *type-2* machining operations are pre-finishing or finishing operations [Thimm *et al.*, 2004a]. The difference between type-1 and type-2 operations is that the datum surface for the former comes from the same stock removal set as the operation surface, whereas the datum for type-2 operations comes from a different set.

Finishing sub-graph and pre-form sub-graph The datum-hierarchy tree as well as the datum-hierarchy graph can be split into two sub-graphs: a finishing sub-graph which contains

Figure 2.5: A design with its design dimension tree
(adopted from [Thimm *et al.*, 2004b])

all operations contributing to a design tolerance stack and any pre- or finishing operations not included in the stacks, as well as a pre-form sub-graph which contains the remaining operations [Thimm *et al.*, 2004b].

2.4 Efficiency Measures Regarding Dimensional Specifications

To develop an automated technique for assessing the manufacturability of process plans and part designs, efficiency measures are proposed by Britton *et al.* [Britton *et al.*, 2001] and Thimm *et al.* [Thimm *et al.*, 2004a,b]. In contrast to other manufacturability criteria, these possess a clear, computationally graspable link from the design to a process plan [Thimm *et al.*, 2004a], and are applicable without any assumptions on the shop floor, cost of machining operations, and so on.

2.4.1 Efficiency measures for multi-finished rotational parts

Originally, the efficiency measures proposed in Britton *et al.* [2001] are for uni-finished rotational parts, and determined by comparing an actual datum-hierarchy tree and its subgraphs with an “ideal tree”. Thimm *et al.* generalized these measures to multi-finished rotational parts [Thimm *et al.*, 2004b].

The first criterion proposed for optimality aims at a “minimum number of manufacturing

2.4 Efficiency Measures Regarding Dimensional Specifications

operations”¹ [Britton *et al.*, 2001]. That is, the criterion measures technical efficiency, which is the best a process planner can achieve given the available technology level, personnel, machines, and tools (without necessarily knowing in which plant the manufacturing takes place). In terms of this criterion, the *ideal finishing sub-graph* is one that contains only finishing and pre-finishing operations, and results in minimal tolerance stacks for design dimensions [Thimm *et al.*, 2004a]. A minimal stack has no more than one type-2 machining operation per stack (and none, one or more type-1 machining operations).

Four efficiency measures were defined in [Britton *et al.*, 2001; Thimm *et al.*, 2004b] for multi-finished rotational parts:

1. *Finishing Efficiency*: It shows the extent to which a process planner can minimize the number of pre-finishing and finishing operations [Britton *et al.*, 2001].
2. *Pre-forming Efficiency*: In practice, it is typical that pre-forming operations are required to shape the initial blank for the pre- and finishing operations. This measure can reflect the inefficiency resulting from a process planner’s choice of shape for the initial blank [Britton *et al.*, 2001].
3. *Overall Efficiency*: As described in [Britton *et al.*, 2001], the overall efficiency measure of a process plan is a combination of the finishing efficiency and the pre-forming efficiency.
4. *Tolerance Efficiency*: It is an additional measure, as a process plan, whose finishing sub-graph contains the minimum number of operations to complete a part and there are no pre-forming operations, does not guarantee the minimum total tolerance stack-up [Thimm *et al.*, 2004b].

Examples on how to calculate and apply these measures to assess the machinability of process plans are illustrated in [Britton *et al.*, 2001] and [Thimm *et al.*, 2004b].

¹Multi-surface operations and multi-tool operations are not taken into account here, which are considered in the following efficiency measure, operation element based efficiency.

2.4.2 Efficiency measures for multi-finished prismatic parts

In Thimm *et al.* [2004a], the operation element based efficiency for uni-finished prismatic parts is introduced, which extends the efficiency defined in [Thimm *et al.*, 2004b] to process plans and parts with an arbitrary number of sets of parallel surfaces. The measures serve as tools to estimate production cost for designers, or measures of the efficiency of a given process plan compared to the theoretically optimal process plan in the hands of a process planner [Thimm *et al.*, 2004a]. Furthermore, the proposed optimality criterion is the minimal number of manufacturing operations considering the influences of multi-surface operations and multi-tool operations. In the following, some minor changes are made to the definitions of machining features and operational elements, so that multi-finished prismatic parts are accommodated.

To facilitate the development of this efficiency measure, machining features and operational elements are introduced in [Thimm *et al.*, 2004a]. Both are sets of surfaces, though slightly different constraints on a set's element are defined. Either may comprise [Thimm *et al.*, 2004a]:

- a single surface,
- touching, perpendicular surfaces,
- step surfaces,
- slots, or
- any set of surfaces machined during a machining pass concurrently.

In particular, a *machining feature* is a set of pre-forming, pre-finishing, or finishing surfaces that can be machined concurrently or in any arbitrary sequence given the available technology (datum and location surfaces in the process plan stage are disregarded), and an *operational element* is defined as a subset of $\mathbb{S}$ which is a complete set of machining surfaces for a process plan, whereby machining surfaces may be pre-forming, pre-finishing, or finishing ones.

2.4 Efficiency Measures Regarding Dimensional Specifications

Moreover, a set $\mathbb{F}$ of machining features is called *feasible* if [Thimm *et al.*, 2004a]

1. The union of all N features $\mathbb{F}_i$ equals $\mathbb{S}$:

$$\bigcup_{i=1}^N \mathbb{F}_i = \mathbb{S}$$

2. A surface is part of one machining feature only:

$$\mathbb{F}_i \cap \mathbb{F}_j = \phi \quad \text{for any different } i, j \text{ with } 1 \leq i, j \leq N$$

3. A machining feature may not contain two surfaces from the same stock removal set (for the case of multi-finished surfaces).

And a set of machining features is further called *minimal*, if it is feasible, and no other feasible set with less elements exists.

An (ordered) set of operational elements $\mathbb{O}$, forms the core of a process plan and must fulfill the following [Thimm *et al.*, 2004a]:

1. All $\mathbb{O}_i$ are machining features.
2. Operational element $\mathbb{O}_1$ can be executed without any preconditions.
3. Operational element $\mathbb{O}_i$ can be executed without any preconditions, except that $\mathbb{O}_1$ to $\mathbb{O}_{i-1}$ where performed.
4. An operational element may not contain two surfaces of the same stock removal set (in the case of multi-finished surfaces).
5. The union of all N sets $\mathbb{O}_i$ equals $\mathbb{S}$: $\bigcup_{i=1}^N \mathbb{O}_i = \mathbb{S}$.
6. A surface is only machined once: $\mathbb{O}_i \cap \mathbb{O}_j = \phi$ for any different i, j with $1 \leq i, j \leq N$.

Based on the above definitions, the operation elements based efficiency is defined as [Thimm *et al.*, 2004a]

$$\text{Eff}_{\text{operational}} = \frac{||\mathbb{F}||}{||\mathbb{O}||} \quad (2.8)$$

where $||\mathbb{F}||$ is the number of elements in the minimal, feasible machining feature set $\mathbb{F}$, $||\mathbb{O}||$ is the number of elements in the operational element set $\mathbb{O}$, which is based on a tentative process plan created from the part's design automatically as shown in [Thimm *et al.*, 2004a].

2.4.3 Applicability of manufacturability criterion

As a criterion of manufacturability, efficiency measures are not aimed at pinning down the exact cost, which is hard, if not impossible, to calculate; also but not only due to its site-dependency. However, some important aspects closely related to manufacturing cost are considered. Among them are the number of tool changes, the number of machine changes, the number of set-ups, and other efficiency measures defined in section 2.4.1 and 2.4.2.

Manufacturing cost, which is further discussed in section 3.7, can only be used when sufficient cost-tolerance information is available - often requiring much more information than available at design time.

2.5 Computer Aided Process Planning (CAPP)

It is generally accepted that process planning is an important link bridging the gap between design and production [Sermsuti-Anuwat, 1992]. In the present advanced environment of computer-aided design (CAD) and computer-aided manufacturing (CAM), computer-aided process planning is an important key to the success of fully computerized manufacturing, which is often referred to as CIM (computer integrated manufacturing).

Two basic approaches of computer-aided process planning exist as follows [Alting and Zhang, 1989]:

- *Variant approach*: the variant process planning is based on a predefined processing sequences for families of parts, these being stored in a suitable form of database in the computer. This type of process planning system usually requires a coding system for identifying the part family, and therefore it is closely related to the use of group technology techniques.
- *Generative approach*: the generative process planning embodies the manufacturing logic and uses it to generate the processing sequence for a part from scratch. The limitation of the system depends only on the logic stored. Because this manufacturing logic is compiled from experiences (know-how of the process planners) and from relevant

2.6 Term Rewriting System and Functional Programming

scientific knowledge, this type of process planning system is more difficult to develop than the previous one and still under research [Huang and Zhang, 1996; Huang *et al.*, 1997; Huang, 1998; Chu *et al.*, 2000; Qiao *et al.*, 2000; Thimm *et al.*, 2001; Zhou *et al.*, 2002]. Nevertheless, the system can offer automated planning without or little human intervention.

The generative approach is more suitable for automated manufacturability analysis, since ‘only in this approach one can develop a process plan at the early design stage where manufacturability evaluation is most effective’ (citation from [Gupta *et al.*, 1997]).

2.6 Term Rewriting System and Functional Programming

2.6.1 Term rewriting system

Term rewriting is a branch of theoretical computer science which combines elements of logic, universal algebra, automated theorem proving and functional programming [Baader and Nipkow, 1998]. It is based on equational logic. In general, term rewriting applies in any context where efficient methods for reasoning with equations are required.

In this thesis, a design rewriting system (refer to section 4.3 and chapter 6), which is a specialized term rewriting system, is established. This includes a number of rewriting rules, which are applied to modify dimensional schemes of designs.

2.6.2 Functional programming and Haskell

To implement the design rewriting system, and to conduct the design rewriting process automatically, Haskell, a functional programming language, is chosen [Hal Daume, 2002].

The advantage of functional programming is that it provides tools, such as high-order functions, to make the programming more well-constructed [Hughes, 1989]. For this thesis, automatic pattern matching available in Haskell simplifies the task of finding patterns in the overall dimension graph (refer to section 4.1).

2.7 Summary

In general, current DFM approaches focus on the effects of the topology and geometry on the manufacturability of a part, and the geometric reasoning capacity has been improved vastly since the middle 1990's. For example, feature accessibility is an important problem in current ME, and it has been paid high attention [Gupta and Nau, 1995; Stage, 1996], while the non-geometric factors, such as dimension and tolerance, are only checked with minor exceptions, not considered sufficiently. This includes in particular the systematic redesign of dimensions and tolerances.

Four aspects of computer-aided tolerances are reviewed. The torsor model is identified as a good approach to model tolerance zones and accommodate the propagation of both size and geometric tolerances. Tolerance analysis (transfer) is mature for size tolerances, but not for GD&T, although great effort was and is given to it [Veitschegger and Wu, 1986; He and Gibson, 1992; Chase *et al.*, 1994; Whitney *et al.*, 1994; Rivest *et al.*, 1994; Legoff *et al.*, 1999; Villeneuve *et al.*, 2001]. In addition, five main approaches to tolerance synthesis are summarized.

The datum-hierarchy tree technique was identified as a good base for this thesis and therefore was closely examined. Also, definitions of efficiency measures and relationship between the measures and the design dimension graph are included and reviewed.

CHAPTER 3

MANUFACTURABILITY REGARDING DIMENSIONING

3.1 Glossary

The definitions in this section are mainly based on unpublished definitions related to dimensions collected by Dr. Graeme Britton.

Dimension: a necessary numerical value expressed in appropriate units of measure and used to define the size, location, geometric characteristic, or surface texture of a part or part feature [ASME Y14.5M, 1994]. As stated in the fundamental rules of ASME Y14.5M [1994], *‘each dimension shall have a tolerance, except for those dimensions specifically identified as reference, maximum, minimum, or stock (commercial stock size)’*.

Tolerance: the total amount a specific dimension is permitted to vary. The tolerance is the difference between the maximum and minimum limits [ASME Y14.5M, 1994].

Basic Dimension: a numerical value used to describe the theoretically exact size, profile, orientation, or location of a feature or datum target. It is the basis from which permissible variations are established by tolerances on other dimensions, in notes, or in feature control frames [ASME Y14.5M, 1994].

CHAPTER 3. MANUFACTURABILITY REGARDING DIMENSIONING

Datum: a theoretically exact point, axis, or plane derived from the true geometric counterpart of a specified datum feature [ASME Y14.5M, 1994]. A *datum feature* is an actual feature of a part that is used to establish a datum [ASME Y14.5M, 1994]. For more details on datums, refer to section 3.5.

Dimension Limits: the extreme limits of a dimension [Bjørke, 1989]. The largest acceptable value of the dimension is denoted *maximum dimension limit* while the smallest acceptable value is termed *minimal dimension limit* (figure 3.1).

Deviation: The *upper deviation* is defined as the difference between the maximum dimension limit and the basic size while the *lower deviation* is the difference between the minimum dimension limit and the basic size [Bjørke, 1989] (figure 3.1).

Bilateral & Unilateral Tolerance: The *bilateral tolerance* is a tolerance which is expressed as plus and minus certain values, where neither is zero, to specify allowable variations in both direction from the basic dimension.

The *equal bilateral tolerance* has the same definition as bilateral tolerance except that the positive and negative values of the deviations are equal.

The *unilateral tolerance*, on the other hand, is one that applies only in one direction from the basic dimension, and the allowance variation in the other direction is zero [Teck, 1996].

In this thesis, it is assumed that all dimensional tolerance are equal bilateral tolerances. Provided it is not an equal bilateral tolerance, a conversion is required.

Geometric Tolerance: The general term applied to the category of tolerances used to control form, profile, orientation, location, and runout [ASME Y14.5M, 1994].

Tolerance stack & stackup: A *tolerance stack* is a set of operations that contribute to the tolerance of a dimension. A *tolerance stackup* is the sum of the tolerances of the operations in a tolerance stack [Britton *et al.*, 2001].

3.2 Critical Factors

Figure 3.1: Summary of dimensional tolerance concepts (adopted from [Bjørke, 1989])

3.2 Critical Factors

Usually, dimensions and tolerances of a design are not fully determined by functional requirements. However, not all designs are optimal in the sense of manufacturability. To evaluate the manufacturability of a design with respect to its design specifications, the factors that contribute to the manufacturability must be determined first. Whether design specifications are optimal depends on the following factors:

- ◇ the functional requirements of the part;
- ◇ the geometry and topology of the part;
- ◇ the assembly constraints;
- ◇ the manufacturing constraints;
- ◇ the available resources, processes, including both manufacturing and assembly;
- ◇ cost projection;
- ◇ metrology, delivery, and so on.

An economical dimensional design specification should respect all these constraints, and designers should take all these constraints into account at the detail design stage. If a designer does not consider these constraints in this stage, he/she rarely accomplishes a design that has a high manufacturability, *e.g.* high quality, low cost and short lead time.

CHAPTER 3. MANUFACTURABILITY REGARDING DIMENSIONING

The functional requirements of a part have the deterministic priority in all these factors. Any other factors must respect the functional requirements of the part, otherwise, they have to be compromised on. In fact, assembly constraints, like clearance and fit, are determined by the functional requirements of a product.

Some of these factors are interdependent or contradictory [Weill, 1997]. For example, the manufacturing constraints depend on the available manufacturing resources, such as the availability of machines, cutters, or processes. Furthermore, from the point of view of functional requirements, tight tolerances are better, which is contrast to the fact that loose tolerances are preferred in terms of manufacturing constraints.

Figure 3.2: Factors influencing the manufacturability of a design specification

Figure 3.2 shows key factors and their main dependencies. In this figure, available resources represent among other things the capacity of a specific factory. An arrow signifies that the factor at the tail of the arrow affects the contents or quality of the factor at the head of the arrow. If the available resources set is the most advanced given the current technology level, the best dimensional design specification may be identified. One point worth noting is that when the available resources change, the optimal dimensional design specifications may change as well. In other words, whether a design specification is optimal depends heavily on the available resources and processes.

Provided the initial design specifications are not optimal with respect to available resources, there is a need to redimension it to achieve a better manufacturability. This need is

3.3 Types of Design Dimensions

fulfilled by a systematic approach presented hereafter.

3.3 Types of Design Dimensions

Not all design dimensions play the same role in a product. Some design dimensions or combinations of dimensions are critical to the functionality of the part, others are perhaps determined by the assembly requirements, or by the manufacturing processes. The cost for fulfilling these dimensions and tolerances is difficult to be reduced. However, some design dimensions are not critical for accomplishing the functionality and other requirements of the product. According to Ullman [1992], less than 20% of the tolerances on a typical component actually affect its function, and the remainder of the dimensions on a typical product could be outside the range set by their tolerances and it would still operate satisfactorily. For these dimensions, economic aspects should be considered with a bigger weighting factor. The author suggests to classify design dimensions into two different groups:

- Functional design dimension;
- Trivial design dimension.

Trivial design dimensions are those which do not have an important impact on the functionality or assemblability of a part. Therefore, the modification of this type of design dimensions will not affect greatly the design's functionality or its role in an assembly.

In order to be able to distinguish the type of dimensions or tolerances within a CAPP tool, it is assumed in the following that the designer specified the type of each design dimension.

<i>Type</i>	<i>Abbre.</i>
Functional design dimension	$\check{F}$
Trivial design dimension	$\check{T}$

Table 3.1: Types of design dimensions

Let ξ denote a size or geometric dimension, and let $\tau(\xi)$ represent the type of dimension ξ as listed in table 3.1, then $\tau(\xi) \in \{\check{F}, \check{T}\}$. In addition, all geometric dimensions as specified

CHAPTER 3. MANUFACTURABILITY REGARDING DIMENSIONING

by a designer are regarded as of type $\check{F}$, while some size tolerances are functional, and others are trivial.

As the functionality of a part must not be compromised on, modifications to design dimensions of type $\check{F}$ must result in equivalent or tighter tolerances. Also, assembly constraints and other life-cycle constraints of a product must be respected. However, for dimensions of type $\check{T}$, modification can ease the initial specifications under certain circumstances, *i.e.* relax tolerances to some extent [Jietong *et al.*, 2003a].

3.4 Dimension Set

As stated in section 3.3, ξ denotes a size or geometric dimension. A subscript distinguishes different dimensions in a version of a design, and superscripts distinguish different versions of a design. For example, ξ_i^j represents the i th dimension in the j th version of a design.

Let Ξ^j represent the set of all dimensions in the j th version of a design:

$$\Xi^j = \{\xi_1^j, \xi_2^j, \dots, \xi_i^j, \dots, \xi_n^j\} \quad (3.1)$$

Let set Ξ_T^j represent the set of the trivial dimensions, then

$$\Xi_T^j = \{\xi^j \mid \xi^j \in \Xi^j, \tau(\xi^j) \equiv \check{T}\} \quad (3.2)$$

And similarly,

$$\Xi_F^j = \{\xi^j \mid \xi^j \in \Xi^j, \tau(\xi^j) \equiv \check{F}\} \quad (3.3)$$

Ξ^j is the union of Ξ_T^j and Ξ_F^j , whose intersection is empty. Also, Ξ^j has two subsets: size dimension set and geometric dimension set.

3.5 References and Datums

In order to position a point, line or surface in a part or an assembly, references to other points, lines, *etc.*, are necessary. Common references (datums) are points, lines and surfaces. A *datum* is ‘a theoretically exact point, axis, or plane derived from the true geometric counterpart of a specified datum feature. A datum is the origin from which the location or geometric

3.5 References and Datums

characteristics of features of a part are established' (citation from [ASME Y14.5M, 1994]).

Datums, based on their uses, can be classified into five categories [Zhang *et al.*, 1996]:

- Design datums,
- Operational datums,
- Locating datums,
- Measuring datums, and
- Assembly datums.

3.5.1 Design datums

A *design datum* is a point, a line or a surface in a part drawing, from which the position of another point, line or surface on the part is dimensioned [Wang and Li, 1991]. Several geometric entities (points, lines or surfaces) may have identical design datums. The position of an entity may be defined by several design datums.

The design datums and design dimensions on a part drawing are given by the product designer. Rationally, when the design dimensions and design datums are determined, manufacturing issues should be considered. However, due to a lack of theoretical foundations and appropriate design tools as well as experiences of the design in process planning and so on, the impact of design dimensions and datums on manufacturing processes is often not taken into account.

3.5.2 Operational datums

An *operational datum* is a geometric entity (point, line, surface, *etc.*), from which the position of a surface to be machined is dimensioned [Wang and Li, 1991]. The operational datum usually is introduced during the process planning stage. The dimension from the operational datum to the machined surface is referred to as the *working dimension*. The operational datum, working dimensions and tolerances are defined by the process planner.

CHAPTER 3. MANUFACTURABILITY REGARDING DIMENSIONING

3.5.3 Locating datums

A *locating datum* is a surface (not a point or a line) of a workpiece which defines the proper position of the workpiece in the direction of the working dimension on the workholder (fixture) or the machine table for workpiece set-up [Wang and Li, 1991]. The locating datum can be further divided into two groups: supporting locating datums and calibrating locating datums.

3.5.4 Measuring datums

A *measuring datum* is a geometric entity (point, line or surface), relative to which the position of a workpiece surface is inspected after machining [Wang and Li, 1991].

3.5.5 Assembly datums

An *assembly datum* is a geometric entity (point, line or surface), which determines [Wang and Li, 1991]

- the position of a part in an assembly, or
- the position of a sub-assembly on an assembly machine.

3.5.6 Principle of coincidence of datums

The *principle of coincidence of datums* states that, when it is practical, the operational datum should coincide with the design, locating and measuring datums [Wang and Li, 1991]. Complying with the principle avoids tolerance stackup and consequently, greatly reduces manufacturing costs. However, this coincidence is not always achievable since constraints imposed by, for example, available resources on the process plan have to be respected, like a lack of a specific fixture.

It can be inferred straightforwardly from the above principle that, whenever possible, it is better to avoid dimensioning where both design datums are lines and/or points, which cannot be used as locating datums.

3.6 Minimizing Tolerance Stacks

3.6.1 Necessary tolerance stacks

Tolerance stacks have basically two causes: poor process planning and constraints on the plan which are beyond the planner's control. Tolerance stacks are called *necessary* with respect to a design if they are present in all realistic process plans for a given design. Possible reasons for necessary stackups are:

- Direct manufacturing needs expensive equipments, tools, and so on.
- The equipment or tools required by direct manufacturing are out of the current technology level.
- The design's dimensions conflict with its geometry and topology.

In some cases, although a design dimension can be directly manufactured with no or smaller tolerance stackup, a tolerance stack is acceptable if the cost difference for the higher process cost is inferior to the cost caused by changing, for example, a setup. In general, whether a stackup is acceptable depends on its impact on the manufacturing cost.

3.6.2 Balancing process tolerances and changing operational datums

Suppose that a design with N dimensions and a process plan P are given. The set O represents all operations in process plan P , and $O = \{o_1, o_2, \dots, o_i, \dots, o_n\}$, the subscript distinguishes different operations in O . Suppose operation o_i ($i = 1, 2, \dots, n$) contains m_i cuts¹, *i.e.* $o_i = \{c_{i1}, c_{i2}, \dots, c_{im_i}\}$. Note that, only one cut exists in a uni-surface and uni-tool operation, whereby more than one cut exists in a multi-surface or multi-tool operation. One cut has one process tolerance, and it corresponds to a single line in a tolerance chart [Thimm *et al.*, 2004a].

¹As defined in [Wang and Li, 1991], a *cut* consists of one or more machining passes necessary to produce one surface of a workpiece, using the same machine, tool, and location surfaces.

CHAPTER 3. MANUFACTURABILITY REGARDING DIMENSIONING

Let $t(c_{ij})$ ($i = 1, 2, \dots, n$ and $j = 1, 2, \dots, m_i$) represent the process tolerance assigned to cut c_{ij} . Let $l(c_{ij})$ stand for the lower tolerance limit of cut c_{ij} and $u(c_{ij})$ for the upper tolerance limit achievable by the process. Then,

$$0 < l(c_{ij}) \leq t(c_{ij}) \leq u(c_{ij}), \text{ for all } i \text{ and } j, 1 \leq i \leq n, 1 \leq j \leq m_i \quad (3.4)$$

Let $\oplus$ represent the machining tolerance stackup operator. Suppose S_{ξ_j} is the *tolerance stack set* of ξ_j , representing the set of cuts that contribute to the tolerance stack for design dimension ξ_j in process plan P . Then

$$\bigoplus_{c \in S_{\xi_j}} t(c) \leq t(\xi_j), \text{ for all } j, 1 \leq j \leq N \quad (3.5)$$

Note that O and S_{ξ} are distinct: O stands for all operations in process plan P , while S_{ξ} comprises cuts that contribute to the stack of design dimension ξ in O . Thus, for all $c \in S_{\xi}$, $t(c) \leq t(\xi)$ is the least condition and if more than one element is in set S_{ξ} , $t(c)$ is truly smaller than $t(\xi)$.

Supposed that only size design dimensions are considered, the worst case stackup model as presented in section 2.3.3 can be expressed as

$$\bigoplus_{c \in S_{\xi_j}} t(c) = \sum_{c \in S_{\xi_j}} t(c) \quad (3.6)$$

where ξ_j is the j th size dimension, as presented in section 3.4. Substituting equation (3.6) into equation (3.5),

$$\sum_{c \in S_{\xi_j}} t(c) \leq t(\xi_j), \text{ for all } j, 1 \leq j \leq N_{size} \quad (3.7)$$

where N_{size} is the number of size dimensions.

Process tolerances are not uniquely defined by these unequations. Even if stock removal constraints are considered, the number of process tolerances sets that yield the same design tolerance is infinite [He, 1989]. Considering the fact that tolerances achievable by machines in practice are treated as discontinuous, this number still can be very large.

3.6 Minimizing Tolerance Stacks

For a given process plan, two ways allow in principle to change machining tolerances. The first way merely optimizes the process tolerances of individual process (balancing), without changes to the sequence of process or tolerance stacks. The optimization targets at minimizing cost according to some cost-tolerance models [Speckhart, 1972; Michael and Siddall, 1981; Lee and Woo, 1989; Dong *et al.*, 1994]. Most current approaches attempt to decrease the manufacturing cost this way. However, this approach has severe limitations as only one datum-hierarchy graph is considered (compare section 2.3.4 on page 38 for a discussion of datum-hierarchy graphs). As stated in Britton and Whybrew [1997], the major gains in cost savings come from changing operational datum surfaces for individual operations, *i.e.* changing the datum-hierarchy graph, and not from tolerance adjustments.

The second approach to optimize working dimensions and process tolerances consists of four main steps [Fathi *et al.*, 1997; Tseng and Kung, 1999]:

1. Generate alternative process plans with distinct stacks;
2. For each process plan, identify the tolerance chains for all design dimensions;
3. Balance the process tolerances;
4. Evaluate cost of individual plan and select plan(s) with least cost.

However, this improved approach limits the optimization to a fixed dimensional scheme and does not consider alternative design specifications. In particular, necessary tolerance stacks cannot be reduced by this approach.

To improve over this, alternative design dimensional schemes have to be considered in the detail design phase or redesign phase. The second approach mentioned above achieves this: the optimal dimensional scheme for a design in the sense of manufacturability with regarding to certain measures, like ones introduced in section 2.4, can be found.

3.7 Manufacturing Cost Considering Alternative Specifications

An optimal process plan can be understood as a function (in a not strictly mathematical sense) of a design specification and constraints resulting from the manufacturing environments. This applies consequently too for the manufacturing cost. The manufacturing cost function $\$()$ is a function of the dimensional scheme, the functional requirements, manufacturing constraints and quality constraints:

$$\$(\Xi^i, C_{func}, C_{manu}, C_{qual}) \quad (3.8)$$

Where,

Ξ^i is the i th dimension set of a design;

C_{func} is the set of functional requirements, which is represented by critical dimension chain(s) in an assembly;

C_{manu} is the set of manufacturing constraints, including tolerance stacks, stock removal limitations, and manufacturing process capacity constraints;

C_{qual} is the set of quality constraints.

For given Ξ^i satisfying all constraints, this function estimates the manufacturing cost. If the constraints are contradictory (*i.e.* no process plan exists), the cost is defined as infinite. Thus, the target of finding a cost effective design can be formulated as a minimization of $\$(\Xi^i, C_{func}, C_{manu}, C_{qual})$ with respect to i . The minimization of $\$()$ comprises two iteratively applied steps:

1. Redimension the design, and then
2. Vary the process datums and process tolerances to achieve the minimal manufacturing cost for the dimensional scheme being considered according to the second conventional approach presented in the previous section.

Through redimensioning, tolerance stacks, in particular, necessary stacks, can be effectively reduced or even removed totally. This directly results in a design with higher efficiency mea-

3.8 Summary

asures presented in section 2.4, simplifies the work of process planners and often decreases the manufacturing cost.

3.8 Summary

A glossary and notations used in this thesis are provided. Then, critical factors which influence the manufacturability of the dimensional scheme of a design are analyzed. Also, the principle of coincidence of datums (direct machining) is introduced, which is the potential target of an optimization of redimensioning a design. In addition, as two manufacturability measures of a dimensional scheme, tolerance stacks and manufacturing cost are discussed. In particular, two conventional approaches to optimize process tolerances are reviewed together with the benefits of considering the alternative design specifications.

CHAPTER 4

DESIGN REWRITING

Design dimensions and tolerances can be satisfied by a process plan in two ways: directly or indirectly. A dimension is achieved directly if a single machining operation exists that generates the final state of dimension using the design datum(s) of that dimension as manufacturing and location datums. For indirect machining, as opposed to a direct case, there is not a single cut which can finish a dimension applying its design datum(s). In other words, that a dimension is achieved directly means the principle of coincidence of datums (refer to section 3.5.6) is respected, and *vice versa*. In direct machining, the tolerance stack is minimal, as it consists of one operation only and the accuracy and precision of this cut alone are responsible for the compliance with tolerance constraints. Direct machining has the advantage of requiring the least possible process tolerances and simplifications of process planning.

However, initial design specifications, including both size and geometric, are often unsuitable for efficient process plans (*e.g.* direct machining). This is in particular true if the designer has no process planning experience, insufficient information on the machining environment or a complex design on hands. Process planners are consequently forced to

- either introduce tolerance stacks into the plan, which in turn require excessive high machining capabilities; or
- plan with frequent datum and fixture changes, causing high fixture and labor costs; or
- request a redesign of the part - in particular, to re-dimension it.

4.1 Overall Dimension Graphs

The later is - not considering the logistic - the better option. These disadvantages can be reduced by the presented approach, as it automates the redimensioning step. The approach aims at replacing sub-optimal dimensions by alternatives without impact the functionality of the design.

To obtain alternative design specifications, this chapter starts to define some concepts. Then, the design rewriting system is briefly introduced. It is followed by the detail illustration of design rewriting rules.

4.1 Overall Dimension Graphs

Size and geometric dimensions which are directly specified by the designer are called *explicit dimensions*. Those dimensions that can be inferred from explicit dimensions are *implicit dimensions*, which are not directly specified by the designer, but determined by the topology and geometry of a part itself. Moreover, explicit dimensions may be replaced, in a non-reversible process and under certain circumstances, by implicit dimensions, with the consequence that a new design specification is created.

For size dimensions of the example part in figure 2.3 whose design dimension graph is in figure 4.1, the explicit and implicit size dimensions are shown in figure 4.2. In figure 4.2, an edge represented by a solid line denotes an explicit dimension and an edge represented by a dashed line an implicit dimension. Additionally, in a design dimension graph, implicit geometric dimensions exist as well. For instance, between any pair of flat surfaces, are at a certain angles (0° for parallel surfaces). Refer to figure 4.3 to see implicit geometric dimensions of the example part.

Note that, as the independence of size and geometric dimensions in the modification process of dimensional schemes, and for the sake of clarity, size and geometric overall dimension graphs are drawn separately, although they coexist.

An *overall dimension graph* comprises nodes that represent all geometric entities related by dimensions (most of them are surfaces) in a design, and edges that represent explicit dimensions and/or implicit dimensions. An edge without an arrow represents a size dimen-

sion. The head of a directed edge denotes a feature controlled, while the tail corresponds to a datum. An edge labelled with B corresponds to a basic dimension, and those labelled with P , S or T stand for primary, secondary, or tertiary datum, correspondingly. It is obvious that edges in an overall dimension graph represent all possible dimensions that can be specified by a designer, and a design dimension graph is a sub-graph of an overall dimension graph. When only size dimensions are considered, a valid design dimension graph is always a forest of spanning trees of the sub-graphs corresponding to parallel surfaces. Otherwise, the design is either under-constrained or over-constrained. For a rotational part which has one dimension tree with n nodes, there are total n^{n-2} distinct spanning trees in the overall dimension graph. In this sense, a designer selects one tree from many possible ones.

Note that unless stated otherwise, a dimension is an explicit dimension.

Figure 4.1: Design dimension graph for the example part

4.1.1 States of dimensions

Often, it is the case that designers determine the dimension specifications of a part mainly from the functional point of view, and result in a sub-optimal manufacturability, whereby there is a need to rewrite the initial design dimension scheme to obtain a better manufacturability. From the description in the previous section, this can be achieved by properly rewriting an overall dimension graph and extract a new sub-graph, design dimension graph,

4.1 Overall Dimension Graphs

Figure 4.2: Overall dimension graph (size dimensions)

from it, which holds a better manufacturability.

For this purpose, the concept of state of a dimension is introduced first.

When an overall dimension graph is continuously being rewritten, there are four kinds of dimensions which should be distinguished by their states:

- Original **O**,
- Removed **R**,
- Implicit **I**,
- Created **C**.

A dimension is in the state **O**, if it was specified by the designer, and it may change into the state **R**, if it is replaced by some other dimensions. A dimension that is implicit in the original design has the state **I** and may change its state to **C** when it replaces some other dimension(s).

There are only dimensions of state **I** and **O** in an initial overall dimension graph. Along with the evolution of the overall dimension graph, some dimensions will be removed and some are created, so that dimensions of state **R** and **C** will occur, respectively.

In any stage of a rewriting process, the graph comprising all dimensions of state **O** and **C** is a correct design dimension graph, as long as the rewriting operations obey rules and constraints imposed in later sections.

Figure 4.3: Partial overall dimension graph (geometric dimensions)

4.1.2 Construction of the initial overall dimension graph

In this thesis, the initial overall dimension graph is constructed manually. However, given the solid model and dimensioning and tolerancing information of a design, the general procedure to automatically generate its initial overall dimension graph is:

1. Generate its design dimension graph, in which dimensions are of state **O**.
2. Divide planes into sets, such that planes in the same set G_i ($i = 1, 2, \dots, N$) are parallel to each other. A line is added into all sets G_i with planes parallel to the line. In cases that the axis of a hole is not parallel to any planes, create a new G_j ($j = N + 1, \dots, M$) for it. A point, like the center of a semi-sphere, is added to all sets G_i ($i = 1, 2, \dots, M$).
3. The procedures to create implicit dimensions are:
 - (a) For size dimensions:

4.1 Overall Dimension Graphs

- i. Create a set S including one arbitrarily selected size dimension of this design.
 - ii. for the remainder size dimensions of the design, if the geometric entities controlled by the considered size are parallel to those of any size dimensions in set S , it is ignored. Otherwise, it is added into S . This process is repeated until all size dimensions are taken into account.
 - iii. For each element in S , find the corresponding G_i per the planes (or axis, or point) controlled by it. Then, unless a size dimension with state **O** exists, generate implicit size dimensions for every pair of surfaces in G_i .
- (b) For parallelisms: for each set G_i ($i = 1, 2, \dots, M$), generate an implicit parallelism between every pair of geometric entities (excluding points), unless a parallelism in state **O** exists.
- (c) For perpendicularities:
- i. For perpendicularities between planes: find the perpendicularity relationship between set G_i and G_j ($i \neq j$ and $i, j = 1, 2, \dots, N$). Generate implicit perpendicularity dimensions for any pair of planes, whereby one plane is an element of G_i and the other of G_j .
 - ii. For perpendicularities between an axis and a plane: find all axes in this design. For each axis, find the sets G_i ($i = 1, 2, \dots, N$) to which it is perpendicular (zero, one or two of such sets may exist). For each set identified, establish implicit perpendicularity dimension between the axis and every element of this set, unless a perpendicularity is specified explicitly.
 - iii. For perpendicularities between two axes: find all axes in the design first. Generate implicit perpendicularity dimension between any pair of axes if they are perpendicular to each other, unless a perpendicularity of state **O** exists.
- (d) For basic dimensions (of position tolerances of holes):
- i. Find all position dimensions of state **O**.

- ii. For each position tolerance,
 - A. For its primary datum¹, identify the set G_j to which it belongs. Note that in cases the primary datum is an axis, there may exist more than one such set. For each set identified, generate implicit primary basic dimensions between every element of this set and the feature controlled by this position tolerance, unless already specified in the design.
 - B. For its secondary and tertiary datum (if any), the similar procedure described in step 3(d)iiA applies, except that implicit secondary and tertiary basic dimensions are generated, respectively.

The procedure presented above deals with that which edges, especially those for implicit dimensions, will appear in the initial overall dimension graph. Other information, like extreme points of geometric entities in the design, relative geometric position and orientation among geometric entities in the design, is stored in the corresponding node or edge labels, which is addressed in detail in section 6.3.1 on page 115.

4.2 Overall Dimension Set

An *overall dimension set*, Ω , consists of all dimensions of all states in an overall dimension graph. In the initial overall dimension set, Ω^0 , the explicit dimensions with state **O** are the initial design specifications given by the designer, and the implicit dimensions with state **I** are generated according to those initial explicit dimensions as addressed in section 4.1.2.

4.2.1 Potential alternative design specifications

In general, many implicit design specifications can potentially be used to substitute an explicit specification and rewrite a design. As the overall dimension graph for a rotational part considering only dimensions along the axis of the part shown in figure 4.4, for example, the size dimension between surfaces **k** and **m** is removed and an alternative to be chosen. The

¹In cases no primary datum, this position tolerance is ignored, and the next one, if any, is considered.

4.3 The Design Rewriting System

Figure 4.4: Choices for replacing an explicit size dimension

initial design dimension tree is the graph connected with solid lines in figure 4.4a. In the figure 4.4b, set **A** has i nodes and set **B** has $(n-i)$ nodes. In order to achieve again a valid design dimension graph, the newly added dimension must relate a node from set **A** with another from set **B**. The number of possible choices is $i \cdot (n-i)$. Figure 4.4c shows a possible choice, which includes the edge $p - q$ into the new design dimension graph.

4.3 The Design Rewriting System

In this section, the concepts of the term and rewriting rule are defined. It is followed by the property of the design rewriting system in terms of termination and confluence. Finally, the interaction among modules of the system is discussed.

The theoretical base of the presented design rewriting system is the term rewriting system that is introduced in section 2.6. The later are sets of rewriting rules, which transform a term (in this case a design, therefore the name design rewriting system) in a specific manner [Baader and Nipkow, 1998].

4.3.1 Definitions

Terms

In general, *terms* consist of *variables*, *constant symbols*, and *functional symbols*. In the here presented design rewriting system, usually points, lines and surfaces of a design are variables which are represented by $a, b, c, \dots$. The constants **O**, **I**, **R** and **C** designate the state of a dimension (see section 4.1.1). $\perp$, $\parallel$, $\oplus$, $\angle$, and $\mathcal{S}$ are functional symbols, where $\perp$ stands for perpendicularity, $\parallel$ for parallelism, $\oplus$ for position, $\angle$ for angularity, and $\mathcal{S}$ for size tolerance. The first parameter of all functions is a state, and these functions have two more parameters, where, with the exception of the size function, the first one is a geometric entity which is the feature controlled by the dimension, and the second one is a geometric entity or a datum frame made up of geometric entities. However, the size function is symmetric with respect to its last two parameters, *i.e.* $\mathcal{S}(s, a, b) \equiv \mathcal{S}(s, b, a)$.

For example, the term $\parallel(s, a, b)$ denotes that a parallelism exists between feature² a and b , with state s , where a is the feature controlled and b is the datum. As for the position $\oplus(s, a, [b, c, d])$, a is the geometric entity to be positioned and $[b, c, d]$ represents the frame where b is the primary datum, c the secondary datum, and d the tertiary datum. Moreover, $[p, q]$ denotes a datum frame which is composed of two datum features, where p the primary datum and q secondary.

As other ASME geometric tolerances are not expected to contribute to the decisions made during operation sequencing, set-up planning and fixture planning [Hong and Chang, 2003], they are not further considered here.

²The definition of feature is: the general term applied to a physical portion of a part, such as a surface, pin, tab, hole or slot [ASME Y14.5M, 1994].

4.3 The Design Rewriting System

Rewriting rules

A *rewriting rule* is an identity of two terms l and r , where l is not a variable and all variables occurring in r must also be in l and *vice versa*. The rule $l \xrightarrow{\text{cond}} r$ means the left-hand side is replaced by right-hand side if the optional condition cond is true. Implicitly in the following, all variables that are arguments to the same function must assume different values (in other words, a dimension cannot establish a reflective relation for any feature). A condition may make use of the following predicates: $\mathcal{P}(a, b, \dots)$, signifying that a, b are planes and $\mathcal{L}(a, b, \dots)$, constraining $a, b, \dots$ to be lines.

By convention, the first parameter of a function is its state as introduced in section 4.1.1. Apart from state transfers described in the above mentioned section, no other state changes are allowed. By convention, variables s and t designate states.

This restriction on the state changes is motivated by the reflection that the problematic original explicit dimensions should be removed from the design dimension graph by labelling its state as removed and should not be reinserted again. On the other side, the dimensions replacing them should not be replaced, as this would increase tolerance stacks further. This is an important feature of the approach, as it avoids a reinsertion of dimensions (with potentially decreased tolerance) and ensures that design specifications remain sensible as compared to the original design.

Furthermore, the design, as well as the left and right side of the rewriting rules, have to be understood as sets, implying that their sequence is insignificant.

When a left side of a rule is matched with the design, this is done function-wise. If a symbol of a design is matched with a function of the left side of a rule, the function symbol, all constants and bound variables must be the same. If a function still has a free variable, it is bound to the corresponding value of the matching function, before the next function is considered.

4.3.2 Termination and confluence

The presented design rewriting system is finite, *i.e.* the rewriting process will terminate after a finite number of steps, which is determined by the allowed state transfer mechanism stated in section 4.1.1. However, this system is not confluent. In other words, the sequence in which the rules are applied and location of their application may change the result.

4.3.3 Modules of the system

The reader will note that the rules do not take into account any tolerance, nor do they define any. The determination of the tolerances requires a second mechanism which is illustrated in detail in chapter 5. The basic idea is that whenever a rule is applied, a corresponding constraint on the value of the new dimensions based on the removed dimension is added to a set of constraints, which also includes any machining constraints³. Once the design is transformed, a solution to these unequations are the tolerances for the changed design. This may occur very complicated, but has several main advantages:

- the balancing of tolerances can include previously existing dimensions,
- the tolerance stackup model can be changed, and
- the approach is split into fairly independent parts.

Furthermore, to automatically generate redesign suggestions, a manufacturability evaluation module is required. It is used to automatically rate the manufacturability of a newly derived design and offer redesign options to designers.

Therefore, a design rewriting system is composed of four modules:

1. The rewriting module, which aims to search feasible alternative dimension chains for given dimensions (without considering tolerance values),
2. The constraint generation module, which is designed to create restrictions for the real-location of tolerance values,

³These constraints and tolerances could be included into the rewriting system, though this would unnecessarily complicate the notation.

4.3 The Design Rewriting System

3. The manufacturability evaluation module, which is used to automatically assess the manufacturability of a design, and
4. The user interface, where users can enter the dimensions desired to be removed from a design dimension graph.

Figure 4.5: The interaction diagram of the system

The interaction of the system is depicted in figure 4.5. Arrows represent messages dispatched among them. With the collaboration of rewriting module and constraint generation module, alternative design dimension graphs satisfying all available requirements are created, without manufacturability evaluation yet. In this system, two modes exist: in the inter-

active mode, it is up to the human judgment to determine which dimension chain is chosen; while in the automatic mode, it depends on the manufacturability evaluation module to assess the manufacturability of designs, and offer a list of sorted designs with respect to the manufacturability measure.

Theoretical considerations for rewriting rules are presented in the remaining of this chapter. Chapter 5 discusses theoretical issues of constraint generation. Then, the module of manufacturability evaluation, the integration of modules, and the implementation of the whole system are proposed in chapter 6.

4.4 Rewriting Rules

Given the definitions in section 4.3.1, it is possible to define the core of the approach: the design rewriting rules. The rules' main purpose is to change the state of implicit and original dimension(s) in the presence of others.

Rewriting rules are grouped into sets of similar rules for clarity, though, in an implementation no separation is necessary. Furthermore, the rules only permit a maximal tolerance stack of three dimensions. As it is, the rule set is - for the sake of the length of the thesis - not detailed in its full extent. The total amount of rules is approximately 100, and in this thesis, only around 50 rules are detailed.

4.4.1 Size dimensions

A size dimension can be substituted by others, if they link features, which are all parallel to the same (virtual) plane. This constraint is indicated as $//(\cdot, \cdot, \cdot)$ ($\cdot$ represents a feature), and a point is regarded as parallel to any plane or line. Equations (4.1) and (4.2) replace a dimension by a chain of length two and three, respectively (where only one is allowed to be implicit in order to prevent over-constraining). The equations (4.3) allows for arbitrary long

4.4 Rewriting Rules

tolerance chains, though this is usually not desirable.

$$\mathcal{S}(\mathbf{O}, a, b) \mathcal{S}(s, b, c) \mathcal{S}(\mathbf{I}, c, a) \xrightarrow[s \in \{\mathbf{O}, \mathbf{C}\} \wedge // (a, b, c)]{} \quad (4.1)$$

$$\begin{aligned} & \mathcal{S}(\mathbf{R}, a, b) \mathcal{S}(s, b, c) \mathcal{S}(\mathbf{C}, c, a) \\ & \mathcal{S}(\mathbf{O}, a, b) \mathcal{S}(s, b, c) \mathcal{S}(t, c, d) \mathcal{S}(\mathbf{I}, d, a) \xrightarrow[s, t \in \{\mathbf{O}, \mathbf{C}\} \wedge // (a, b, c, d)]{} \end{aligned} \quad (4.2)$$

$$\begin{aligned} & \mathcal{S}(\mathbf{O}, a_1, a_2) \mathcal{S}(s_2, a_2, a_3) \dots \mathcal{S}(s_{n-1}, a_{n-1}, a_n) \mathcal{S}(\mathbf{I}, a_n, a_1) \\ & \xrightarrow[s_2, \dots, s_{n-1} \in \{\mathbf{O}, \mathbf{C}\} \wedge // (a_1, a_2, \dots, a_n)]{} \end{aligned} \quad (4.3)$$

$$\mathcal{S}(\mathbf{R}, a_1, a_2) \mathcal{S}(s_2, a_2, a_3) \dots \mathcal{S}(s_{n-1}, a_{n-1}, a_n) \mathcal{S}(\mathbf{C}, a_n, a_1)$$

4.4.2 Parallelism

Two simplest rules are presented first. They replace a parallelism constraining planes or lines by two others (where at lease one is implicit)⁴:

$$//(\mathbf{O}, a, b) //(\mathbf{I}, a, c) // (s, c, b) \xrightarrow[s \in \{\mathbf{I}, \mathbf{C}\} \wedge (\mathcal{P}(b, c) \vee \mathcal{L}(a, b, c))]{ } //(\mathbf{R}, a, b) //(\mathbf{C}, a, c) //(\mathbf{C}, c, b) \quad (4.4)$$

$$//(\mathbf{O}, a, b) //(\mathbf{O}, a, c) //(\mathbf{I}, c, b) \xrightarrow[\mathcal{P}(b, c) \vee \mathcal{L}(a, b, c)]{} //(\mathbf{R}, a, b) //(\mathbf{O}, a, c) //(\mathbf{C}, c, b) \quad (4.5)$$

Furthermore, a parallelism can be converted into four perpendicularities in the presence of two suitable surfaces. A simple example for this rule is a cube, with the top and bottom surfaces (a and b) constrained to be parallel. In the modified scheme, both surfaces are constrained to be perpendicular to the same two, non-parallel side surfaces (c and d). This rule set is summarized in table 4.1. Each row represents a rewriting rule. The columns correspond to dimensions. The arrows in the table indicate a state change of the dimension at the top of the column. For example, $\mathbf{I}, \mathbf{C} \rightarrow \mathbf{C}$ mean that if the dimension is either in the state $\mathbf{I}$ or in the state $\mathbf{C}$, the rule can be applied and the new state is $\mathbf{C}$. The notation $\mathbf{O}, \mathbf{C} \rightarrow \mathbf{O}, \mathbf{C}$ means that the rule can be applied if the dimension in state $\mathbf{O}$ or $\mathbf{C}$, and the state remains unchanged. The constraint applies to all rules in this table.

⁴If b and c are planes, a can be a line or a plane.

$//(\cdot, a, b)$	$\perp(\cdot, a, c)$	$\perp(\cdot, a, d)$	$\perp(\cdot, c, b)$	$\perp(\cdot, d, b)$	Constraints
$O \rightarrow R$	$I \rightarrow C$	$I \rightarrow C$	$I \rightarrow C$	$I, C \rightarrow C$	$\mathcal{P}(a, b, c, d)$ and $\neg //(\cdot, c, d)$.
$O \rightarrow R$	$I \rightarrow C$	$I \rightarrow C$	$C \rightarrow C$	$C \rightarrow C$	
$O \rightarrow R$	$C \rightarrow C$	$I \rightarrow C$	$I \rightarrow C$	$C \rightarrow C$	
$O \rightarrow R$	$C \rightarrow C$	$I \rightarrow C$	$C \rightarrow C$	$I, C \rightarrow C$	
$O \rightarrow R$	$O \rightarrow O$	$I \rightarrow C$	$I, C \rightarrow C$	$I, C \rightarrow C$	
$O \rightarrow R$	$O \rightarrow O$	$C \rightarrow C$	$I \rightarrow C$	$I, C \rightarrow C$	
$O \rightarrow R$	$O \rightarrow O$	$C \rightarrow C$	$C \rightarrow C$	$I \rightarrow C$	
$O \rightarrow R$	$O \rightarrow O$	$O \rightarrow O$	$I \rightarrow C$	$I, C \rightarrow C$	
$O \rightarrow R$	$O \rightarrow O$	$I \rightarrow C$	$O \rightarrow O$	$I, C \rightarrow C$	
$O \rightarrow R$	$O \rightarrow O$	$I \rightarrow C$	$I, C \rightarrow C$	$O \rightarrow O$	
$O \rightarrow R$	$O \rightarrow O$	$I \rightarrow C$	$O \rightarrow O$	$O \rightarrow O$	

Table 4.1: The second rule set for parallelism

For example, the first row in table 4.1 represents the rule

$$//(\mathbf{O}, a, b) \perp (\mathbf{I}, a, c) \perp (\mathbf{I}, a, d) \perp (\mathbf{I}, c, b) \perp (s, d, b) \xrightarrow[\mathcal{P}(a,b,c,d)]{s \in \{\mathbf{I}, \mathbf{C}\} \wedge \neg //(\cdot, c, d) \wedge} \quad (4.6)$$

$$//(\mathbf{R}, a, b) \perp (\mathbf{C}, a, c) \perp (\mathbf{C}, a, d) \perp (\mathbf{C}, c, b) \perp (\mathbf{C}, d, b)$$

To complement rules summarized in table 4.1, instead of replacing the parallelism by perpendicularities, the parallelism can replace two of the perpendicularities.

$$//(\mathbf{I}, a, b) \perp (\mathbf{O}, a, c) \perp (\mathbf{O}, a, d) \perp (s, b, c) \perp (s, b, d) \xrightarrow[\mathcal{P}(a,b,c,d)]{s \in \{\mathbf{O}, \mathbf{C}\} \wedge \neg //(\cdot, c, d) \wedge} \quad (4.7)$$

$$//(\mathbf{C}, a, b) \perp (\mathbf{R}, a, c) \perp (\mathbf{R}, a, d) \perp (s, b, c) \perp (s, b, d)$$

4.4 Rewriting Rules

4.4.3 Perpendicularity

Two parallel surfaces or lines may be linked by a perpendicular feature, where either a perpendicularity or the parallelism is replaced, as shown in equations (4.8) to (4.12).

$$//(\mathbf{O}, a, b) \perp (\mathbf{I}, a, c) \perp (s, c, b) \xrightarrow[\mathcal{L}(a,b) \wedge \mathcal{L}(c)]{s \in \{\mathbf{O}, \mathbf{C}\} \wedge (\mathcal{P}(a,b) \vee)} \quad (4.8)$$

$$//(\mathbf{R}, a, b) \perp (\mathbf{C}, a, c) \perp (s, c, b)$$

$$//(\mathbf{O}, a, b) \perp (\mathbf{I}, a, c) \perp (\mathbf{I}, c, b) \xrightarrow[\mathcal{L}(a,b) \wedge \mathcal{L}(c)]{(\mathcal{P}(a,b) \vee)} \quad (4.9)$$

$$//(\mathbf{R}, a, b) \perp (\mathbf{C}, a, c) \perp (\mathbf{C}, c, b)$$

$$//(\mathbf{I}, a, b) \perp (\mathbf{O}, a, c) \perp (s, b, c) \xrightarrow[\mathcal{L}(a,b,c) \vee \mathcal{P}(a,b,c)]{s \in \{\mathbf{O}, \mathbf{C}\} \wedge ((\mathcal{P}(a,b) \wedge \mathcal{L}(c)) \vee)} \quad (4.10)$$

$$//(\mathbf{C}, a, b) \perp (\mathbf{R}, a, c) \perp (s, b, c)$$

$$//(s, a, b) \perp (\mathbf{O}, a, c) \perp (\mathbf{I}, b, c) \xrightarrow[\mathcal{L}(a,b,c) \vee \mathcal{P}(a,b,c)]{s \in \{\mathbf{I}, \mathbf{C}\} \wedge ((\mathcal{P}(a,b) \wedge \mathcal{L}(c)) \vee)} \quad (4.11)$$

$$//(\mathbf{C}, a, b) \perp (\mathbf{R}, a, c) \perp (\mathbf{C}, b, c)$$

$$//(s, b, c) \perp (\mathbf{O}, a, c) \perp (\mathbf{I}, a, b) \xrightarrow[\mathcal{L}(a,b,c) \vee \mathcal{P}(a,b,c)]{s \in \{\mathbf{I}, \mathbf{C}\} \wedge ((\mathcal{P}(a,b) \wedge \mathcal{L}(c)) \vee)} \quad (4.12)$$

$$//(\mathbf{C}, b, c) \perp (\mathbf{R}, a, c) \perp (\mathbf{C}, a, b)$$

A similar replacement can be performed in the presence of two pairs of parallel planes in a perpendicular orientation, as shown in table 4.2 and 4.3. Longer chains of parallel surfaces are imaginable, but this implies ever longer tolerance chains - impairing the machinability of the dimensions in the replacing tolerance chain, as ever higher machining capabilities are required.

$\perp(\cdot, a, b)$	$\perp(\cdot, c, d)$	$//(\cdot, a, c)$	$//(\cdot, d, b)$	Constraints
$\mathbf{O} \rightarrow \mathbf{R}$	$\mathbf{I} \rightarrow \mathbf{C}$	$\mathbf{I}, \mathbf{C} \rightarrow \mathbf{C}$	$\mathbf{I}, \mathbf{C} \rightarrow \mathbf{C}$	$\mathcal{L}(a, b, c, d) \vee$
$\mathbf{O} \rightarrow \mathbf{R}$	$\mathbf{O}, \mathbf{C} \rightarrow \mathbf{O}, \mathbf{C}$	$\mathbf{I} \rightarrow \mathbf{C}$	$\mathbf{I}, \mathbf{C} \rightarrow \mathbf{C}$	$\mathcal{P}(a, c) \wedge \mathcal{L}(b, d) \vee$
$\mathbf{O} \rightarrow \mathbf{R}$	$\mathbf{I} \rightarrow \mathbf{C}$	$\mathbf{O} \rightarrow \mathbf{O}$	$\mathbf{I}, \mathbf{C} \rightarrow \mathbf{C}$	$\mathcal{P}(a, b, c, d)$
$\mathbf{O} \rightarrow \mathbf{R}$	$\mathbf{O} \rightarrow \mathbf{O}$	$\mathbf{O} \rightarrow \mathbf{O}$	$\mathbf{I} \rightarrow \mathbf{C}$	

Table 4.2: Replacing perpendicularities in the presence of two parallelisms

$\perp(\cdot, a, b)$	$\perp(\cdot, d, c)$	$\parallel(\cdot, c, a)$	$\parallel(\cdot, d, b)$	Constraints
$\mathbf{I} \rightarrow \mathbf{C}$	$\mathbf{O}, \mathbf{C} \rightarrow \mathbf{O}, \mathbf{C}$	$\mathbf{O}, \mathbf{C} \rightarrow \mathbf{O}, \mathbf{C}$	$\mathbf{O} \rightarrow \mathbf{R}$	$\mathcal{P}(a, c) \wedge \mathcal{L}(b, d)$ $\vee \mathcal{L}(a, b, c, d)$
$\mathbf{I} \rightarrow \mathbf{C}$	$\mathbf{I} \rightarrow \mathbf{C}$	$\mathbf{O}, \mathbf{C} \rightarrow \mathbf{O}, \mathbf{C}$	$\mathbf{O} \rightarrow \mathbf{R}$	
$\mathbf{I} \rightarrow \mathbf{C}$	$\mathbf{O}, \mathbf{C} \rightarrow \mathbf{O}, \mathbf{C}$	$\mathbf{I} \rightarrow \mathbf{C}$	$\mathbf{O} \rightarrow \mathbf{R}$	
$\mathbf{I} \rightarrow \mathbf{C}$	$\mathbf{I} \rightarrow \mathbf{C}$	$\mathbf{I} \rightarrow \mathbf{C}$	$\mathbf{O} \rightarrow \mathbf{R}$	

Table 4.3: Replacing parallelism in the presence of perpendicularities

4.4.4 Angularity

Depending on the respective values, elements in a chain of angularities may effectively be parallel or at a right angle to each other. This offers the theoretical possibility that a parallelism or a perpendicularity is replaced by a chain of almost arbitrary angularities. However, as machining to an arbitrary angle is usually more difficult than machining to parallelism or to a right angle, only rules, which replace an angle, are given here. Note, that whether angles add up to a parallelism, perpendicularity or an arbitrary angle, is established when the overall design dimension graph is calculated (only in this case, the two angle “terms” exist in the design). If the two angles result into a parallelism, the following rules can be applied:

$$\angle(\mathbf{O}, a, b) \angle(\mathbf{I}, a, c) \parallel(s, c, b) \xrightarrow[\mathcal{P}(b, c) \vee \mathcal{L}(a, b, c)]{s \in \{\mathbf{I}, \mathbf{C}\} \wedge} \angle(\mathbf{R}, a, b) \angle(\mathbf{C}, a, c) \parallel(\mathbf{C}, c, b) \quad (4.13)$$

$$\angle(\mathbf{O}, a, b) \angle(\mathbf{C}, a, c) \parallel(\mathbf{I}, c, b) \xrightarrow[\mathcal{P}(b, c) \vee \mathcal{L}(a, b, c)]{} \angle(\mathbf{R}, a, b) \angle(\mathbf{C}, a, c) \parallel(\mathbf{C}, c, b) \quad (4.14)$$

$$\angle(\mathbf{O}, a, b) \angle(s, a, c) \parallel(\mathbf{I}, c, b) \xrightarrow[\mathcal{P}(b, c) \vee \mathcal{L}(a, b, c)]{s \in \{\mathbf{O}, \mathbf{C}\} \wedge} \angle(\mathbf{R}, a, b) \angle(s, a, c) \parallel(\mathbf{C}, c, b) \quad (4.15)$$

$$\angle(\mathbf{O}, a, b) \angle(\mathbf{I}, a, c) \parallel(\mathbf{O}, c, b) \xrightarrow[\mathcal{P}(b, c) \vee \mathcal{L}(a, b, c)]{} \angle(\mathbf{R}, a, b) \angle(\mathbf{C}, a, c) \parallel(\mathbf{O}, c, b) \quad (4.16)$$

Similar rules apply for the case when the sum of the two angles results in a perpendicularity (the $\parallel$ functions have to be replaced by $\perp$ functions in equations (4.13) to (4.16)). An

4.4 Rewriting Rules

angularity may also be substituted by an angularity and/or two parallelisms.

$$\angle(\mathbf{O}, a, b) \angle(\mathbf{I}, c, d) // (s, a, c) // (t, d, b) \xrightarrow[\vee \mathcal{P}(a, b, c, d) \vee \mathcal{L}(a, b, c, d)]{s, t \in \{\mathbf{I}, \mathbf{C}\} \wedge (\mathcal{L}(a, c) \wedge \mathcal{P}(d, b))} \quad (4.17)$$

$$\begin{aligned} & \angle(\mathbf{R}, a, b) \angle(\mathbf{C}, c, d) // (\mathbf{C}, a, c) // (\mathbf{C}, d, b) \\ & \angle(\mathbf{O}, a, b) \angle(\mathbf{C}, c, d) // (\mathbf{I}, a, c) // (s, d, b) \xrightarrow[\vee \mathcal{P}(a, b, c, d) \vee \mathcal{L}(a, b, c, d)]{s \in \{\mathbf{I}, \mathbf{C}\} \wedge (\mathcal{L}(a, c) \wedge \mathcal{P}(d, b))} \end{aligned} \quad (4.18)$$

$$\begin{aligned} & \angle(\mathbf{R}, a, b) \angle(\mathbf{C}, c, d) // (\mathbf{C}, a, c) // (\mathbf{C}, d, b) \\ & \angle(\mathbf{O}, a, b) \angle(\mathbf{O}, c, d) // (\mathbf{I}, a, c) // (s, d, b) \xrightarrow[\vee \mathcal{P}(a, b, c, d) \vee \mathcal{L}(a, b, c, d)]{s \in \{\mathbf{I}, \mathbf{C}\} \wedge (\mathcal{L}(a, c) \wedge \mathcal{P}(d, b))} \end{aligned} \quad (4.19)$$

$$\begin{aligned} & \angle(\mathbf{R}, a, b) \angle(\mathbf{O}, c, d) // (\mathbf{C}, a, c) // (\mathbf{C}, d, b) \\ & \angle(\mathbf{O}, a, b) \angle(\mathbf{I}, c, d) // (\mathbf{O}, a, c) // (s, d, b) \xrightarrow[\vee \mathcal{P}(a, b, c, d) \vee \mathcal{L}(a, b, c, d)]{s \in \{\mathbf{I}, \mathbf{C}\} \wedge (\mathcal{L}(a, c) \wedge \mathcal{P}(d, b))} \end{aligned} \quad (4.20)$$

$$\begin{aligned} & \angle(\mathbf{R}, a, b) \angle(\mathbf{C}, c, d) // (\mathbf{O}, a, c) // (\mathbf{C}, d, b) \\ & \angle(\mathbf{O}, a, b) \angle(\mathbf{I}, c, d) // (\mathbf{O}, a, c) // (\mathbf{O}, d, b) \xrightarrow[\vee \mathcal{P}(a, b, c, d) \vee \mathcal{L}(a, b, c, d)]{\mathcal{L}(a, c) \wedge \mathcal{P}(d, b)} \end{aligned} \quad (4.21)$$

$$\begin{aligned} & \angle(\mathbf{R}, a, b) \angle(\mathbf{C}, c, d) // (\mathbf{O}, a, c) // (\mathbf{O}, d, b) \\ & \angle(\mathbf{O}, a, b) \angle(\mathbf{O}, c, d) // (\mathbf{O}, a, c) // (\mathbf{I}, d, b) \xrightarrow[\vee \mathcal{L}(a, c) \wedge \mathcal{P}(d, b)]{\mathcal{P}(a, b, c, d) \vee \mathcal{L}(a, b, c, d)} \end{aligned} \quad (4.22)$$

$$\angle(\mathbf{R}, a, b) \angle(\mathbf{O}, c, d) // (\mathbf{O}, a, c) // (\mathbf{C}, d, b)$$

A similar idea motivates the rules in table 4.4.

4.4.5 Position

Under certain circumstances, the datum frame of an explicit original position can be changed (effectively represented by a replacement of the position). Only two example rules are presented for the length of the thesis.

$$\oplus(\mathbf{O}, a, [b, c, d]) \oplus(\mathbf{I}, a, [e, f, g]) // (\mathbf{C}, e, b) // (\mathbf{I}, f, c) // (\mathbf{I}, g, d) \xrightarrow[\mathcal{P}(b, c, d)]{\mathcal{L}(a) \wedge} \quad (4.23)$$

$$\begin{aligned} & \oplus(\mathbf{R}, a, [b, c, d]) \oplus(\mathbf{C}, a, [e, f, g]) // (\mathbf{C}, e, b) // (\mathbf{C}, f, c) // (\mathbf{C}, g, d) \\ & \oplus(\mathbf{O}, a, [b, c]) \oplus(\mathbf{I}, a, [b, d, e]) \oplus(\mathbf{O}, c, [b, d, e]) \xrightarrow[\wedge // (a, c)]{\mathcal{L}(a, c) \wedge \mathcal{P}(b, d, e)} \end{aligned} \quad (4.24)$$

$$\oplus(\mathbf{R}, a, [b, c]) \oplus(\mathbf{C}, a, [b, d, e]) \oplus(\mathbf{O}, c, [b, d, e])$$

$\angle(\cdot, a, [b, c])$	$\angle(\cdot, a, [d, e])$	$\parallel(\cdot, d, b)$	$\parallel(\cdot, e, c)$	Constraints
$O \rightarrow R$	$I \rightarrow C$	$I, C \rightarrow C$	$I, C \rightarrow C$	$\mathcal{L}(a)$
$O \rightarrow R$	$I \rightarrow C$	$O \rightarrow O$	$I, C \rightarrow C$	
$O \rightarrow R$	$I \rightarrow C$	$I, C \rightarrow C$	$O \rightarrow O$	
$O \rightarrow R$	$C \rightarrow C$	$I \rightarrow C$	$I, C \rightarrow C$	
$O \rightarrow R$	$C \rightarrow C$	$I \rightarrow C$	$O \rightarrow O$	
$O \rightarrow R$	$C \rightarrow C$	$C, O \rightarrow C, O$	$I \rightarrow C$	
$O \rightarrow R$	$O \rightarrow O$	$I \rightarrow C$	$I, C \rightarrow C$	
$O \rightarrow R$	$O \rightarrow O$	$I \rightarrow C$	$O \rightarrow O$	
$O \rightarrow R$	$O \rightarrow O$	$C, O \rightarrow C, O$	$I \rightarrow C$	

Another constraint added to rules in this table and subsequent rules using datum frames is that: the dimension of replacing features must be equal to or higher than ones of the original features in a datum frame. For instance, a surface can be used to substitute a line or a point, but not vice versa.

Table 4.4: Replacing angularity of a line

Note that, rule (4.23) is used in an example in section 4.6, and rule (4.24) is applied in section 5.3.1.

4.4.6 Over- and under-constraining a design

In general, applying a design rewriting rule changes the states of some dimensions such that an original dimension is effectively eliminated, and, in the meantime, one or more created dimensions are introduced. This rises the question whether the new specification over- or under-constrains the design. As the question whether size dimensions properly constrain a part can be answered independently from the question whether the geometric dimensions do so, these two problems are addressed separately.

By the construction of the rules, an explicitly given dimension is replaced by a dimension

4.5 Searching Alternative Tolerance Chains

chain. Consequently, under-constraining a design is no issue. On the other hand, assumed that the designer-given geometric dimension and tolerance scheme is valid, then the resulting design is valid if only one dimension changes its state to C. This can be shown by contradiction: suppose the resulting design is over-constrained, but not the original. This can only be the case if the dimension d , whose state was changed to C, was constrained by some chain of dimensions c_1 . However, in the original design, the chain of dimensions c_2 , which matches with the left side of the applied rewriting rule, also constrains d . Consequently, in the original design c_1 and c_2 constrain both the (implicit) dimension d , which contradicts the assumption. Therefore, the rewriting rules for size dimensions (see equations (4.1) to (4.3)) maintain a well-formed set of design specifications, as only one dimension modifies its state to C.

However, when more than one dimension changes its state from I to C, the resulting design may become over-constrained. Whether a rule will over-constrain a design depends on the existing design dimension scheme, *i.e.* the context of the replacement. To detect those cases, existing approaches, as for example [Kandikjan *et al.*, 2001] can be referred to.

4.5 Searching Alternative Tolerance Chains

In this section, the procedure to apply rules proposed in previous sections to find alternative tolerance chain is illustrated. As for the strategy to apply these rules, no “best” scheme is yet identified. The here employed procedure is as follows:

Search-alternative algorithm (see figure 4.6)

1. Specify the dimension and its type to be replaced;
2. Check whether this dimension is of state O, if yes, proceed to the next step, else, step back to 1;
3. Identify applicable rule sets according to the type of this dimension;
4. For each applicable rule,

Figure 4.6: Activity diagram of searching alternative tolerance chains

- (a) Search all paths starting from the datum node and ending at the target node;
- (b) For each path found, check whether it satisfies the feature type constraints and dimension state constraints imposed by the rule being considered;
- (c) Output the feasible paths of this rule being considered;

5. Repeat until all applicable rules are taken into account, then, return all feasible paths.

4.6 Example

4.6 Example

The application of the design rewriting rules is illustrated by the means of the part in figure 2.3. Assume that the part is cut to shape except for the slot, the top surface, the step and the hole. Then, in practice, the machine tool is usually set such that the step surfaces 12Y and 4Z, as well as top surface 1Z are milled parallel the control surface 5Z defined by a 3-point location surface. Surface 10X is adopted as the 2-point location surface to meet dimension 10X–9X. Surface 13Y is used as the 1-point location surface in order to achieve size dimension 13Y–12Y. Then, for cost reasons, the same datum frame is selected for drilling the hole 14XY.

This set-up results into a tolerance stack for the geometric dimension between surfaces 8X–4Z, and the position tolerance for the hole 14XY, due to both machining and setting variations. The design is such that any single set-up causes tolerance stacks.

Direct machinability of these three features in a single setup can be achieved by first applying rule (4.12) with $a := 8X$, $b := 5Z$ and $c := 4Z$ to the partial overall dimension graph in figure 4.3:

$$\begin{aligned} \perp(\mathbf{O}, 8X, 4Z) \perp(\mathbf{I}, 8X, 5Z) //(\mathbf{I}, 5Z, 4Z) \longrightarrow \\ \perp(\mathbf{R}, 8X, 4Z) \perp(\mathbf{C}, 8X, 5Z) //(\mathbf{C}, 5Z, 4Z) \end{aligned}$$

Direct machining can be accomplished for the hole 14XY by first applying rule (4.23) with $a := 14XY$, $b := 4Z$, $c := 10X$, $d := 13Y$, $e := 5Z$, $f := 10X$ and $g := 13Y$ to modify the position tolerance and then applying the rule (4.12) with $a := 14XY$, $b := 5Z$, $c := 4Z$.

$$\begin{aligned} \oplus(\mathbf{O}, 14XY, [4Z, 10X, 13Y]) \oplus(\mathbf{I}, 14XY, [5Z, 10X, 13Y]) //(\mathbf{C}, 5Z, 4Z) \longrightarrow \\ \oplus(\mathbf{R}, 14XY, [4Z, 10X, 13Y]) \oplus(\mathbf{C}, 14XY, [5Z, 10X, 13Y]) //(\mathbf{C}, 5Z, 4Z) \quad (4.25) \end{aligned}$$

$$\begin{aligned} \perp(\mathbf{O}, 14XY, 4Z) \perp(\mathbf{I}, 14XY, 5Z) //(\mathbf{C}, 5Z, 4Z) \longrightarrow \\ \perp(\mathbf{R}, 14XY, 4Z) \perp(\mathbf{C}, 14XY, 5Z) //(\mathbf{C}, 5Z, 4Z) \quad (4.26) \end{aligned}$$

As argued in the first paragraph of section 4.4, these three replacements are all complemented by unequations limiting the tolerances of the dimensions in states **O** and **C** on the right side

of the rules. For example, in the first replacement, the tolerance of dimension $8X-4Z$ has to be superior or equal to the composite tolerance of $8X-5Z$ and $5Z-4Z$. In these replacements, three specifications are removed from the design dimension graph while five are added. Although the resulting design has consequently two more geometric constraints than the original, it has now the advantage of a common setup for direct machining of three machining features.

4.7 Conclusion

To resolve the problem of automatically searching alternative dimensional specifications, some basic concepts, like overall dimension graphs and the state of a dimension, are defined. Based on the foregoing concepts, the design rewriting system is elaborated. In particular, definitions related to the system, properties of the system, and the interaction of its four modules are presented. Then, as the core of the design rewriting system, design rewriting rules are proposed to modify design dimensional schemes. Finally, the procedure to apply design rewriting rules to inspect possible alternative dimensional specifications is developed.

CHAPTER 5

CONSTRAINT GENERATION FOR REDIMENSIONING

In the previous chapter, many rules suitable to substitute perpendicularity, parallelism, angularity, position and size tolerances are presented. However, the rewriting rules alone are insufficient to maintain the design functionality, as they do not determine the tolerance of dimensions in state C. To complete a redesign, constraints for tolerance values of involved dimensions have to be generated parallel to the use of design rewriting rules. Tolerance values, however, are not directly determined. The reason is that, to achieve an optimal tolerance allocation with respect to certain criteria, a tolerance synthesis approach like, for example, the one presented in [Dong *et al.*, 1994] is needed, and it is not the focus of this work. However, constraints are necessary to restrict the resulting tolerance values during tolerance synthesis in order to maintain the initial design intent. Therefore, a mathematical method to generate such constraints is proposed in this chapter.

5.1 Background

In this section, the interval arithmetic and the augmented torsors used to model design tolerances are briefly introduced.

5.1.1 Interval arithmetic

As tolerances are here represented by torsors comprising intervals, relevant interval arithmetic is shortly reviewed. For more information on interval operations, refer to [Moore, 1966].

An interval is a set of numbers defined as $[p, q] = \{x | p \leq x \leq q\}$, where p is the lower bound, and q the upper bound. The addition of two intervals is defined as

$$[p, q] + [r, s] = [p + r, q + s] \quad (5.1)$$

and the multiplication of an interval with a constant α as

$$\alpha \cdot [p, q] = \begin{cases} [\alpha p, \alpha q] & \text{if } \alpha \geq 0, \\ [\alpha q, \alpha p] & \text{otherwise.} \end{cases} \quad (5.2)$$

5.1.2 The augmented torsor model

A tolerance zone is a region of the Euclidean space, in which a fabricated surface is positioned. This zone is expressed relative to the ideal design surface [Teissandier *et al.*, 1999]. A tolerance zone can be defined mathematically by a V-torsor (see section 2.3.1) combined with six interval constraints for components of the torsor [Teissandier *et al.*, 1999]. As shown in equation (5.3), the result of this combination is an *augmented torsor* with 12 elements, in which each element of the original torsor is replaced by an interval.

$$\mathbf{T}_{a,b} |_{\mathfrak{R}_i} = \left[\begin{array}{cc|cc} [r_x] & [R_x] & [t_x] & [T_x] \\ [r_y] & [R_y] & [t_y] & [T_y] \\ [r_z] & [R_z] & [t_z] & [T_z] \end{array} \right] \Bigg|_{\mathfrak{R}_i} \quad (5.3)$$

where a, b as well as $\mathfrak{R}_i$ signify the same as in equation (2.1), respectively. In this augmented torsor, the first two columns are the extremes of the rotations of the real feature in either direction from the ideal position as given by the reference system. The third and fourth column are the like for the translation. In the following, for simplicity, square brackets in equation (5.3) are omitted.

5.1 Background

In the torsor model, a substituting feature is assumed to preserve the topology of the nominal feature, and is used to represent the real feature. In other words, a substituting feature has a perfect form. Meanwhile, the nominal feature has a perfect form by definition. Therefore, the torsor model alone does not support form tolerances [Whitney *et al.*, 1994; Pasupathy *et al.*, 2003].

An augmented torsor comprises of six intervals, which define the small displacement limits between two features with perfect form in a tolerance zone. In an augmented D-torsor, one of the two such features is the substituting feature and the other is the nominal feature, while in an augmented V-torsor, they are both substituting features, whereby one is for the feature controlled by the tolerance, and the other for the datum feature(s). Due to the six intervals, an augmented torsor characterizes an infinite number of torsors, which can formalize any relative position between two features with perfect form.

For the size and geometric tolerances considered in this thesis, corresponding augmented torsors can be defined to mathematically model their zones.

According to the *degree of invariance* concept proposed in [Bourdet and Ballot, 1996], elements of the variation and deviation torsors are undetermined as feature-specific rotations or translations leave the feature unchanged. These undetermined elements are denoted as U in this work, though they are handled in a slightly different way as by Bourdet and Ballot. The following properties are defined ($\mathbb{A} \subset \{U\} \cup \mathbf{R}$) [Legoff *et al.*, 1999]¹:

$$a + U = U \quad \text{for all } a \in \mathbf{R} \quad (5.4)$$

$$a \cdot U = U \quad \text{for all } a \in \mathbf{R} \quad (5.5)$$

$$U + U = U \quad (5.6)$$

$$U \leq U \text{ is true} \quad (5.7)$$

$$[a, b] < [U, U] \text{ is true} \quad \text{for all } a, b \in \mathbf{R}^2, a \leq b \quad (5.8)$$

¹Only equation (5.4) is adopted from [Legoff *et al.*, 1999]

$$\min(\mathbb{A}) = \begin{cases} U & \text{if } U \in \mathbb{A}, \\ x & \text{otherwise. (} x \text{ is the minimal real element of } \mathbb{A} \text{)} \end{cases} \quad (5.9)$$

$$\max(\mathbb{A}) = \begin{cases} U & \text{if } U \in \mathbb{A}, \\ y & \text{otherwise. (} y \text{ is the maximal real element of } \mathbb{A} \text{)} \end{cases} \quad (5.10)$$

Propagation of augmented torsors

To propagate an augmented torsor in equation (5.3), equation (2.3) for a torsor has to be altered. The six elements of an *extreme torsor* of an augmented torsor are interval bounds of the corresponding variations. For an augmented torsor, there exists at most $2^6 = 64$ different extreme torsors, three of which, for example, are

$$\begin{bmatrix} r_x & t_x \\ r_y & t_y \\ r_z & t_z \end{bmatrix} \Big|_{\mathbb{R}_i} \quad \begin{bmatrix} R_x & t_x \\ r_y & t_y \\ r_z & t_z \end{bmatrix} \Big|_{\mathbb{R}_i} \quad \begin{bmatrix} R_x & T_x \\ R_y & T_y \\ R_z & T_z \end{bmatrix} \Big|_{\mathbb{R}_i} . \quad (5.11)$$

Equation (2.3) is applied to all extreme torsors of an augmented torsor respectively. From these transformed torsors, the transformed augmented torsor is obtained as

$$\mathbf{T}_{a,b}|_{\mathbb{R}_j} = \begin{bmatrix} \min(\mathbb{R}_{xj}) & \max(\mathbb{R}_{xj}) & \min(\mathbb{T}_{xj}) & \max(\mathbb{T}_{xj}) \\ \min(\mathbb{R}_{yj}) & \max(\mathbb{R}_{yj}) & \min(\mathbb{T}_{yj}) & \max(\mathbb{T}_{yj}) \\ \min(\mathbb{R}_{zj}) & \max(\mathbb{R}_{zj}) & \min(\mathbb{T}_{zj}) & \max(\mathbb{T}_{zj}) \end{bmatrix} \Big|_{\mathbb{R}_j} , \quad (5.12)$$

where $\mathbb{R}_{xj}$ is the set of elements representing the rotation around x-axis of all transformed extreme torsors, and similar for other sets. Function min and max are defined in equation (5.9) and (5.10), respectively.

Note that, in actual cases, smaller amount (less than 64) of extreme torsors are needed to calculate the transformed augmented torsors by considering coefficients of relevant functions.

5.1 Background

Operators for augmented torsors

Three operators for the augmented torsors are defined hereafter. Suppose $\mathbf{T}_{a,b}|\mathfrak{R}$ and $\mathbf{T}_{c,d}|\mathfrak{R}$ are two augmented torsors.

$$\mathbf{T}_{a,b}|\mathfrak{R} = \left[\begin{array}{cccc} r_x & R_x & t_x & T_x \\ r_y & R_y & t_y & T_y \\ r_z & R_z & t_z & T_z \end{array} \right] \Big|_{\mathfrak{R}} \quad \mathbf{T}_{c,d}|\mathfrak{R} = \left[\begin{array}{cccc} r'_x & R'_x & t'_x & T'_x \\ r'_y & R'_y & t'_y & T'_y \\ r'_z & R'_z & t'_z & T'_z \end{array} \right] \Big|_{\mathfrak{R}} \quad (5.13)$$

Definition 1 (Addition) *The augmented torsor addition is a normal element-wise matrix addition for two torsors expressed in the same reference system.*

$$\mathbf{T}_{a,b}|\mathfrak{R} + \mathbf{T}_{c,d}|\mathfrak{R} = \left[\begin{array}{cccc} r_x + r'_x & R_x + R'_x & t_x + t'_x & T_x + T'_x \\ r_y + r'_y & R_y + R'_y & t_y + t'_y & T_y + T'_y \\ r_z + r'_z & R_z + R'_z & t_z + t'_z & T_z + T'_z \end{array} \right] \Big|_{\mathfrak{R}} \quad (5.14)$$

If two augmented torsors to be added are not in the same reference system, formula (5.12) should be applied to convert them onto a common reference.

Definition 2 (Multiplication with a constant) *Given a real number α , if α is positive, the product of $\alpha\mathbf{T}_{a,b}$ is obtained by multiplying each element of $\mathbf{T}_{a,b}$ with α . Otherwise, $\alpha\mathbf{T}_{a,b}$ is defined as $|\alpha|\mathbf{T}_{b,a}$ ($\alpha < 0$ and $|\alpha|$ is the absolute value of α):*

$$\alpha\mathbf{T}_{a,b}|\mathfrak{R} = \begin{cases} \left[\begin{array}{cccc} \alpha r_x & \alpha R_x & \alpha t_x & \alpha T_x \\ \alpha r_y & \alpha R_y & \alpha t_y & \alpha T_y \\ \alpha r_z & \alpha R_z & \alpha t_z & \alpha T_z \end{array} \right] \Big|_{\mathfrak{R}} & \text{if } \alpha > 0, \\ \left[\begin{array}{cccc} \alpha R_x & \alpha r_x & \alpha T_x & \alpha t_x \\ \alpha R_y & \alpha r_y & \alpha T_y & \alpha t_y \\ \alpha R_z & \alpha r_z & \alpha T_z & \alpha t_z \end{array} \right] \Big|_{\mathfrak{R}} & \text{otherwise.} \end{cases} \quad (5.15)$$

Note that, $\mathbf{T}_{b,a}$ is derived from $\mathbf{T}_{a,b}$ by exchanging column 1 with column 2, exchanging column 3 with column 4, and at the same time, multiplying -1 to each element. In this thesis, only 1 and -1 are multiplied to augmented torsors.

 CHAPTER 5. CONSTRAINT GENERATION FOR REDIMENSIONING

Definition 3 (Comparison) Torsor $\mathbf{T}_{a,b}|_{\mathcal{R}}$ is inferior to torsor $\mathbf{T}_{c,d}|_{\mathcal{R}}$ if each interval of $\mathbf{T}_{a,b}|_{\mathcal{R}}$ is inferior to the corresponding interval of $\mathbf{T}_{c,d}|_{\mathcal{R}}$. Suppose $\mathbf{T}_{a,b}(l, m)$ is the element of the torsor at the l th row and m th column, then

$$\mathbf{T}_{a,b} \leq \mathbf{T}_{c,d} \text{ is true if and only if } \mathbf{T}_{c,d}(l, m) \leq \mathbf{T}_{a,b}(l, m) \leq \mathbf{T}_{a,b}(l, m+1) \leq \mathbf{T}_{c,d}(l, m+1) \\ \text{for all } l \in \{1, 2, 3\} \text{ and } m \in \{1, 3\}. \quad (5.16)$$

where $\mathbf{T}_{a,b}$ and $\mathbf{T}_{c,d}$ are expressed in the same reference system.

5.1.3 Converting between size and geometric tolerances and augmented V-torsors

Given a size or geometric tolerance, the procedure to determine bounds of intervals of an augmented V-torsor expressed in the local reference of the feature controlled by the tolerance is:

1. Find out the degree(s) of invariance as per the tolerance and the feature controlled by the tolerance, which are represented by U .
2. For those left unknown bounds of intervals, choose several extreme points p_i , $i = 1, 2, \dots, N$ of the feature.
3. For each point p_i , unequations are determined according to equation 2.2 to limit its displacement within the tolerance zone.
4. All unequations are linear, therefore, maximal and minimum bounds of the variables in the generated unequations can be determined by the *simplex*-algorithm, depending on the original tolerance specification as well as the size and shape of the feature controlled by the tolerance (for more details, see [Teissandier *et al.*, 1999]).

On the other hand, given an augmented V-torsor, $\mathbf{T}_{a,b}|_{\mathcal{R}_a}$, representing a known type of tolerance, size or geometric, the feature, a , which is controlled by the tolerance, its geometric size, as well as the datum frame, b , the general procedure to convert it into a tolerance value, t , as defined in ASME Y14.5M is:

5.2 Tolerance Constraints

1. Suppose all elements of the augmented torsor are not U , and the augmented torsor is

$$\mathbf{T}_{a,b} |_{\mathcal{R}_a} = \left[\begin{array}{cccc} r_x & R_x & t_x & T_x \\ r_y & R_y & t_y & T_y \\ r_z & R_z & t_z & T_z \end{array} \right] |_{\mathcal{R}_a} \quad (5.17)$$

where all symbols are defined as in equation 5.3, except that $\mathcal{R}_a$ is the local reference for feature a . If U elements are included, they are ignored.

2. Extreme points p_i ($i = 1, 2, \dots, N$) of the feature a are chosen.
3. For each extreme point p_i and for each non- U element in the augmented V-torsor, the maximum displacement in the normal direction of the feature according to equation 2.2 is calculated, for example, v_{i,r_x} , the maximum small displacement of point p_i due to r_x . In other words, for each point p_i , v_{i,d_m} are calculated, where $d = r, R, t$ or T , and $m = x, y$ or z .
4. For each point p_i , $|v_{i,R_m} - v_{i,r_m}|$, and $|v_{i,T_m} - v_{i,t_m}|$ are calculated, where $m = x, y$ or z , and $|\cdot|$ is the absolute value.
5. The minimal value from $|v_{i,R_m} - v_{i,r_m}|$, and $|v_{i,T_m} - v_{i,t_m}|$ ($i = 1, 2, \dots, N$ and $m = x, y$ or z) is assigned to t .

5.2 Tolerance Constraints

In the following, the general method to create constraints for redimensioning is introduced, followed by detailed descriptions on how to define tolerances for parallelism and size dimensions, both functional and trivial (see section 3.3). Other types of dimensions, like perpendicularity, angularity, and position, can be handled in a similar manner. Note that, only regardless-of-feature-size cases are considered, MMC as well as LMC (see section 2.3.1) are not taken into account at this moment.

Given a design or a solid model with design specifications conforming to ASME Y14.5 [ASME Y14.5M, 1994], an unequation for a rewriting rule is created as follows:

Constraint-generation algorithm

1. Identify the dimension chain designated by the rewriting rule and features F_i involved in this chain;
2. Suppose the features controlled by the original specification are F_1 (target) and F_n (datum). Then, define the augmented V-torsor $\mathbf{T}_{1,n}$ representing the tolerance zone of the original specification;
3. Define augmented V-torsor $\mathbf{T}_{i,i+1}$, ($i=1, 2, \dots, n-1$) denoting tolerance zone between features F_i and F_{i+1} ;
4. Transfer all torsors and their component interval bounds into one common reference system by applying equation (5.12);
5. Generate the unequation

$$\sum_{i=1}^{n-1} \mathbf{T}_{i,i+1} \leq \mathbf{T}_{1,n}.$$

5.2.1 Parallelism

Figure 5.1: Planes with parallelism constraints

The most common geometrical design constraint is parallelism. If such a dimension cannot be machined directly, another plane can be inserted into the tolerance chain to make it possible, which is done by the appropriate rewriting rules (see rules in table 4.1 and rules (4.4) to (4.5) as well as (4.8) to (4.9)).

5.2 Tolerance Constraints

For example, three planes with local reference frames attached to their nominal position, respectively, are shown in figure 5.1. These planes are attached to a common rigid body with reference frame $\mathcal{R}_0$; a convention used throughout in following examples. Suppose that the initial specification is $P_1 \parallel P_2$, and the specified tolerance value is t mm.

If rule (4.4) is selected to rewrite the design, and the variables in the rewriting rule are instantiated as: $a := P_1$, $b := P_2$, $c := P_3$, $s := \mathbf{I}$, then the alternative tolerance chain of this modification is $\parallel(P_1, P_3) - \parallel(P_3, P_2)$. Obviously, there are three augmented torsors involved in this chain, which are augmented torsors for $\parallel(P_1, P_2)$, $\parallel(P_1, P_3)$, and $\parallel(P_3, P_2)$. The calculation of these three augmented torsors is demonstrated hereafter.

As the parallelism constraint limits only two rotations of a plane [Bourdet and Ballot, 1996; Whitney *et al.*, 1994; Teissandier *et al.*, 1999], the D-torsor representing the allowable deviation zone of real plane P_1 from its nominal position comprises several undetermined elements. Here, only the rotations around the x- and y-axis are relevant for $\mathbf{T}_{1,\bar{1}}$, as shown in equation (5.18).

$$\mathbf{T}_{1,\bar{1}}|_{\mathcal{R}_1} = \left[\begin{array}{cccc} r_x & R_x & U & U \\ r_y & R_y & U & U \\ U & U & U & U \end{array} \right] \Bigg|_{\mathcal{R}_1} \quad (5.18)$$

where $\bar{1}$ is the nominal plane of feature 1, and $[U, U]$ here indicates undetermined values (see section 5.1.2).

Analogously, the augmented D-torsor standing for the allowable deviation zone of P_2 is

$$\mathbf{T}_{2,\bar{2}}|_{\mathcal{R}_2} = \left[\begin{array}{cccc} r'_x & R'_x & U & U \\ r'_y & R'_y & U & U \\ U & U & U & U \end{array} \right] \Bigg|_{\mathcal{R}_2} \quad (5.19)$$

where $\bar{2}$ is the nominal plane of feature 2.

In order to render the feature chains easier accessible to the reader, they are graphically shown. In these graphs, $\boxed{a}$ represents a feature a , nominal or real. For example, the feature chain in equation (5.18) is $\boxed{1} \xrightarrow{\mathbf{T}_{1,\bar{1}}} \boxed{\bar{1}}$.

In order to accumulate involved torsors, it is necessary to convert them into a common reference system prior to addition. In this case, $\mathcal{R}_1$ is chosen as the common reference sys-

CHAPTER 5. CONSTRAINT GENERATION FOR REDIMENSIONING

tem. The reference frame of $\mathbf{T}_{2,\bar{2}}|_{\mathcal{R}_2}$ is then changed from $\mathcal{R}_2$ to $\mathcal{R}_1$ by using equation (5.12), resulting into $\mathbf{T}_{2,\bar{2}}|_{\mathcal{R}_1}$. The augmented V-torsor denoting the tolerance zone of $/(P_1, P_2)$ is:

$$\begin{aligned} \mathbf{T}_{1,2}|_{\mathcal{R}_1} &= (\mathbf{T}_{1,\bar{1}} + \mathbf{T}_{\bar{1},\bar{2}} + \mathbf{T}_{\bar{2},2})|_{\mathcal{R}_1} \\ &= (\mathbf{T}_{1,\bar{1}} + \mathbf{T}_{\bar{2},2})|_{\mathcal{R}_1} \text{ (as } \bar{1}, \bar{2} \text{ are nominal features, } \mathbf{T}_{\bar{1},\bar{2}} \text{ is zero)} \\ &= (\mathbf{T}_{1,\bar{1}} + (-\mathbf{T}_{2,\bar{2}}))|_{\mathcal{R}_1} \\ &= \left[\begin{array}{cc|cc} r_{x1,\bar{1}} - R'_{x2,\bar{2}} & R_{x1,\bar{1}} - r'_{x2,\bar{2}} & U & U \\ r_{y1,\bar{1}} - R'_{y2,\bar{2}} & R_{y1,\bar{1}} - r'_{y2,\bar{2}} & U & U \\ \hline U & U & U & U \end{array} \right]_{\mathcal{R}_1} \end{aligned} \quad (5.20)$$

The four unknown bounds of two intervals in the above augmented torsor can be approximated as presented in section 5.1.3. In this case, four extreme points of the rectangular surface are chosen to calculate the bounds. The derived bounds for the rotation around x - and y - axis are worst cases.

Similarly, the augmented V-torsors for the tolerance zones of $/(P_1, P_3)$ and $/(P_3, P_2)$, and feature chains are:

$$\mathbf{T}_{1',3'}|_{\mathcal{R}_1} = (\mathbf{T}_{1',\bar{1}} + \mathbf{T}_{\bar{3},3'})|_{\mathcal{R}_1} \quad \boxed{1'} \xrightarrow{\mathbf{T}_{1',\bar{1}}} \boxed{\bar{1}} \xrightarrow{\text{zero}} \boxed{\bar{3}} \xrightarrow{\mathbf{T}_{\bar{3},3'}} \boxed{3'} \quad (5.21)$$

$$\mathbf{T}_{3',2'}|_{\mathcal{R}_1} = (\mathbf{T}_{3',\bar{3}} + \mathbf{T}_{\bar{2},2'})|_{\mathcal{R}_1} \quad \boxed{3'} \xrightarrow{\mathbf{T}_{3',\bar{3}}} \boxed{\bar{3}} \xrightarrow{\text{zero}} \boxed{\bar{2}} \xrightarrow{\mathbf{T}_{\bar{2},2'}} \boxed{2'} \quad (5.22)$$

where “zero” under an arrow signifies that the augmented torsor associating with two nominal features is zero, and $1'$, $2'$, and $3'$ represent real surfaces after redesign. Instead of $1, 2, 3$, different notations for real features are used, because real surfaces before and after redesign may have different D-torsors, and have to be distinguished.

Equation (5.23) constrains the tolerances zones: the sum of the two involved augmented torsors in the alternative chain has to be inferior to the augmented torsor representing the initial tolerance zone.

$$(\mathbf{T}_{1',3'} + \mathbf{T}_{3',2'})|_{\mathcal{R}_1} \leq \mathbf{T}_{1,2}|_{\mathcal{R}_1} \quad (5.23)$$

Provided this constraint is satisfied, the allowable deviation region can be allocated to the two V-torsors according to optimization criteria, as, for example, those proposed in [Dong *et al.*, 1994] and [He and Gibson, 1992].

5.2 Tolerance Constraints

Parallelism: axis to a datum plane

Parallelism can also be defined for the case that feature 1 is a line L_1 , and feature 2 is a plane P_2 , which is the design datum. And if this specification cannot be directly machined, plane P_3 is inserted to create an alternative chain.

Figure 5.2: Parallelism: axis to a datum plane

The tolerance zone of line L_1 is a cylinder. The diameter of the cylinder equals the specified tolerance value. The augmented D-torsor standing for the deviation zone of L_1 is

$$\mathbf{T}_{1,\bar{1}}|_{\mathfrak{R}_1} = \left[\begin{array}{cccc} r_x & R_x & U & U \\ U & U & U & U \\ U & U & U & U \end{array} \right] \Big|_{\mathfrak{R}_1} \quad (5.24)$$

The torsor for P_2 is

$$\mathbf{T}_{2,\bar{2}}|_{\mathfrak{R}_2} = \left[\begin{array}{cccc} r'_x & R'_x & U & U \\ r'_y & R'_y & U & U \\ U & U & U & U \end{array} \right] \Big|_{\mathfrak{R}_2} \quad (5.25)$$

The augmented V-torsor denoting the tolerance zone of $/(L_1, P_2)$ is

$$\mathbf{T}_{1,2}|_{\mathfrak{R}_1} = (\mathbf{T}_{1,\bar{1}} + \mathbf{T}_{2,\bar{2}})|_{\mathfrak{R}_1} \quad \boxed{1} \xrightarrow{\mathbf{T}_{1,\bar{1}}} \boxed{\bar{1}} \xrightarrow[\text{zero}]{\mathbf{T}_{2,\bar{2}}} \boxed{2} \quad (5.26)$$

where $\mathbf{T}_{2,2}|_{\mathfrak{R}_1}$ is derived from $\mathbf{T}_{2,\bar{2}}|_{\mathfrak{R}_2}$ by equation (5.12).

Assume that the same design rewriting rule (4.4) is applied onto the design with $a := L_1, b := P_2, c := P_3, s := \mathbf{I}$. Consequently, the underlying tolerance chain is $/(L_1, P_3) - /(P_3, P_2)$.

CHAPTER 5. CONSTRAINT GENERATION FOR REDIMENSIONING

The augmented V-torsor for $//(P_3, P_2)$ is

$$\mathbf{T}_{3',2'}|_{\mathfrak{R}_2} = (\mathbf{T}_{3',\bar{3}} + \mathbf{T}_{\bar{2},2'})|_{\mathfrak{R}_2} \quad \boxed{3'} \xrightarrow{\mathbf{T}_{3',\bar{3}}} \boxed{\bar{3}} \xrightarrow[\text{zero}]{\quad} \boxed{\bar{2}} \xrightarrow{\mathbf{T}_{\bar{2},2'}} \boxed{2'} \quad (5.27)$$

and the augmented V-torsor for $//(L_1, P_3)$ is

$$\mathbf{T}_{1',3'}|_{\mathfrak{R}_1} = (\mathbf{T}_{1',\bar{1}} + \mathbf{T}_{\bar{3},3'})|_{\mathfrak{R}_1} \quad \boxed{1'} \xrightarrow{\mathbf{T}_{1',\bar{1}}} \boxed{\bar{1}} \xrightarrow[\text{zero}]{\quad} \boxed{\bar{3}} \xrightarrow{\mathbf{T}_{\bar{3},3'}} \boxed{3'} \quad (5.28)$$

Thus, the desired unequation is

$$(\mathbf{T}_{1',3'} + \mathbf{T}_{3',2'})|_{\mathfrak{R}_1} \leq \mathbf{T}_{1,2}|_{\mathfrak{R}_1} \quad (5.29)$$

5.2.2 Size dimensions

A size dimension can be specified between any combination of planes, lines, or points. Figure 5.3 shows the most common case, as discussed in the following, two planar surfaces constrained by a size dimension. Suppose that rule (4.1) is used to rewrite the design in figure 5.3, where $a := P_1$, $b := P_2$, $c := P_3$ (see equation (5.30)).

$$\mathcal{S}(\mathbf{O}, P_1, P_2) \mathcal{S}(\mathbf{O}, P_2, P_3) \mathcal{S}(\mathbf{I}, P_3, P_1) \longrightarrow \mathcal{S}(\mathbf{R}, P_1, P_2) \mathcal{S}(\mathbf{O}, P_2, P_3) \mathcal{S}(\mathbf{C}, P_3, P_1) \quad (5.30)$$

Figure 5.3: Size dimension and tolerance

5.2 Tolerance Constraints

Constraints for functional size dimensions

When a size dimension is functional (compare section 3.3), the accumulated tolerance from the substituting dimension chain has to be inferior or equal to the original design tolerance.

As the functional size tolerance limits two rotations and one translation, the D-torsor describing the deviation zone of a plane P_1 is

$$\mathbf{T}_{1,\bar{1}}|_{\mathcal{R}_1} = \left[\begin{array}{cccc} r_x & R_x & U & U \\ r_y & R_y & U & U \\ U & U & t_z & T_z \end{array} \right] \Bigg|_{\mathcal{R}_1} \quad (5.31)$$

The augmented V-torsor representing the tolerance zone of dimension $\mathcal{S}(P_1, P_2)$ is

$$\mathbf{T}_{1,2}|_{\mathcal{R}_1} = (\mathbf{T}_{1,\bar{1}} + \mathbf{T}_{2,2})|_{\mathcal{R}_1} \quad \boxed{1} \xrightarrow{\mathbf{T}_{1,\bar{1}}} \boxed{\bar{1}} \xrightarrow[\text{zero}]{\mathbf{T}_{2,2}} \boxed{\bar{2}} \xrightarrow{\mathbf{T}_{2,2}} \boxed{2} \quad (5.32)$$

The alternative chain implied by rule (5.30) is $\mathcal{S}(P_2, P_3) - \mathcal{S}(P_3, P_1)$, and augmented torsors for $\mathcal{S}(P_2, P_3)$ and $\mathcal{S}(P_3, P_1)$ are:

$$\mathbf{T}_{2',3'}|_{\mathcal{R}_2} = (\mathbf{T}_{2',\bar{2}} + \mathbf{T}_{3,3'})|_{\mathcal{R}_2} \quad \boxed{2'} \xrightarrow{\mathbf{T}_{2',\bar{2}}} \boxed{\bar{2}} \xrightarrow[\text{zero}]{} \boxed{\bar{3}} \xrightarrow{\mathbf{T}_{3,3'}} \boxed{3'} \quad (5.33)$$

$$\mathbf{T}_{1',3'}|_{\mathcal{R}_1} = (\mathbf{T}_{1',\bar{1}} + \mathbf{T}_{3,3'})|_{\mathcal{R}_1} \quad \boxed{1'} \xrightarrow{\mathbf{T}_{1',\bar{1}}} \boxed{\bar{1}} \xrightarrow[\text{zero}]{} \boxed{\bar{3}} \xrightarrow{\mathbf{T}_{3,3'}} \boxed{3'} \quad (5.34)$$

resulting in the constraint:

$$(\mathbf{T}_{1',3'} + \mathbf{T}_{3',2'})|_{\mathcal{R}_1} \leq \mathbf{T}_{1,2}|_{\mathcal{R}_1} \quad (5.35)$$

Constraints for trivial size dimensions

If a size dimension to be replaced is trivial, each augmented torsor in the dimension chain should be inferior to the augmented torsor standing for the original dimension.

With the same example as in figure 5.3, augmented torsor $\mathbf{T}_{1,2}|_{\mathcal{R}_1}$, $\mathbf{T}_{2',3'}|_{\mathcal{R}_2}$ and $\mathbf{T}_{1',3'}|_{\mathcal{R}_1}$ are defined as in equation (5.32) to (5.34), respectively. The constraint for the modification by equation (5.30) is

$$\left\{ \begin{array}{l} \mathbf{T}_{2',3'}|_{\mathcal{R}_1} \leq \mathbf{T}_{1,2}|_{\mathcal{R}_1} \\ \mathbf{T}_{1',3'}|_{\mathcal{R}_1} \leq \mathbf{T}_{1,2}|_{\mathcal{R}_1} \end{array} \right. \quad (5.36)$$

CHAPTER 5. CONSTRAINT GENERATION FOR REDIMENSIONING

Dimensions	V-Torsors
Size $\mathcal{S}$ (parallel to the z-axis)	$\mathcal{PP} : \begin{bmatrix} r_x & R_x & U & U \\ r_y & R_y & U & U \\ U & U & t_z & T_z \end{bmatrix}, \mathcal{LP} : \begin{bmatrix} r_x & R_x & U & U \\ U & U & U & U \\ U & U & t_z & T_z \end{bmatrix}$
Parallelism $//$ ($\mathcal{P}$: perpendicular to the z-axis; $\mathcal{L}$: parallel to the z-axis)	$\mathcal{PP} : \begin{bmatrix} r_x & R_x & U & U \\ r_y & R_y & U & U \\ U & U & U & U \end{bmatrix}, \mathcal{LP} : \begin{bmatrix} r_x & R_x & U & U \\ U & U & U & U \\ U & U & U & U \end{bmatrix}$
Angularity $\angle$ (the interaction line is parallel to the x-axis)	$\mathcal{PP}$ and $\mathcal{LP} : \begin{bmatrix} r_x & R_x & U & U \\ U & U & U & U \\ U & U & U & U \end{bmatrix}$
Perpendicularity $\perp$ ($\mathcal{P}$: perpendicular to the z-axis; $\mathcal{L}$: parallel to the z-axis)	$\mathcal{PP} : \begin{bmatrix} r_x & R_x & U & U \\ U & U & U & U \\ U & U & U & U \end{bmatrix}, \mathcal{LP} : \begin{bmatrix} r_x & R_x & U & U \\ r_y & R_y & U & U \\ U & U & U & U \end{bmatrix}$
Position $\oplus$ ($\mathcal{L}$: parallel to the z-axis)	$\mathcal{L}[\mathcal{P}, \mathcal{P}, \mathcal{P}]$ and $\mathcal{L}[\mathcal{P}, \mathcal{L}] : \begin{bmatrix} r_x & R_x & t_x & T_x \\ r_y & R_y & t_y & T_y \\ U & U & U & U \end{bmatrix}$

Table 5.1: Augmented V-torsors for some geometric dimensions**5.2.3 Some augmented V-torsors**

In table 5.1, some common augmented V-torsors of geometric dimensions of interest are summarized. In this table, $\mathcal{PP}$ stands for the case where the target is a plane (the first $\mathcal{P}$) and the datum is also a plane, and $\mathcal{LP}$ for the case where the target is a line and the datum is a plane. In addition, $\mathcal{L}[\mathcal{P}, \mathcal{P}, \mathcal{P}]$ represents the case where the target is a line, while the datum frame is created over three planes, and similar for $\mathcal{L}[\mathcal{P}, \mathcal{L}]$.

5.3 Examples

5.3 Examples

In this section, the unit for the length is the millimeter (mm), and angles are expressed in radians (rad). Thus, the unit of the first two columns of an augmented torsor is rad, and that of the last two columns is mm.

5.3.1 The prismatic part with two holes

Figure 5.4 shows a prismatic part originally presented by [Xi *et al.*, 2001]. This part has two holes, where hole 7XY is located relative to a fixed design reference frame made up of three mutually orthogonal planes top surface 1Z, front surface 3X and side surface 6Y, while another hole, 8XY, is located with respect to a design reference frame made up of datums 1Z and 7XY.

To allow for these two holes being directly machined in one setup, the design datum frame has to be common to both holes. The easiest way to achieve this is to change the design datum frame of hole 8XY to the same as of hole 7XY, *i.e.* 1Z as the primary datum, 3X the secondary datum, and 6Y the tertiary datum. This can be achieved by applying the rule (4.24):

$$\begin{aligned} & \oplus(\mathbf{O}, 8XY, [1Z, 7XY]) \oplus(\mathbf{I}, 8XY, [1Z, 3X, 6Y]) \oplus(\mathbf{O}, 7XY, [1Z, 3X, 6Y]) \rightarrow \\ & \oplus(\mathbf{R}, 8XY, [1Z, 7XY]) \oplus(\mathbf{C}, 8XY, [1Z, 3X, 6Y]) \oplus(\mathbf{O}, 7XY, [1Z, 3X, 6Y]) \end{aligned}$$

To constrain the tolerances for this modification, five reference systems are created (see the 3-D model of the part in figure 5.4). The global reference system is $\mathfrak{R}_0$, with its origin arbitrarily at point O . $\mathfrak{R}_{1Z-3X-6Y}$ is the reference system established over datum feature 1Z (3 points), 3X (2 points) and 6Y (1 point), with its origin placed at the geometric center of the part. Note that, $\mathfrak{R}_0$ and $\mathfrak{R}_1 := \mathfrak{R}_{1Z-3X-6Y}$ may have different orientations, depending on datum features. $\mathfrak{R}_{7XY}$ and $\mathfrak{R}_{8XY}$ are local reference systems, whose respective origins are O_{7XY} and O_{8XY} , that is the midpoints of the axes of holes 7XY and 8XY. $\mathfrak{R}_{1Z-7XY}$ is established over datum feature 1Z (3 points) and 7XY (2 points) and is coincident with $\mathfrak{R}_{7XY}$. For the sake of simplicity, $\mathfrak{R}_0$, $\mathfrak{R}_1$, $\mathfrak{R}_{7XY}$, $\mathfrak{R}_{8XY}$ and $\mathfrak{R}_{1Z-7XY}$ are chosen to have the

CHAPTER 5. CONSTRAINT GENERATION FOR REDIMENSIONING

Figure 5.4: Drawing and 3-D model of the second example part with labels and reference frames (adapted from [Xi *et al.*, 2001])

same orientation.

Note that, given the properties of U defined in equations (5.4) - (5.10), these choices for reference systems simplify the problem, and the result derived from it is consistent with [Xi *et al.*, 2001]. If the U is treated as a “special zero” [Whitney *et al.*, 1994; Desrochers and Rivière, 1997], this simplification is not needed, but will complicate the augmented torsor calculation.

Based on these reference systems constructed, the V-torsor for $\oplus(8XY, [1Z, 7XY])$ is

5.3 Examples

derived as

$$\mathbf{T}_{8XY,[1Z,7XY]}|_{\mathfrak{R}_{8XY}} = \left[\begin{array}{cccc} -0.166 & 0.166 & -0.5 & 0.5 \\ -0.166 & 0.166 & -0.5 & 0.5 \\ U & U & U & U \end{array} \right] \Bigg|_{\mathfrak{R}_{8XY}} \quad (5.37)$$

and its feature chain graph is

$$\boxed{8XY} \xrightarrow{\mathbf{T}_{8XY,8XY}} \boxed{8XY} \xrightarrow{\text{zero}} \boxed{[1Z, 7XY]} \xrightarrow{\mathbf{T}_{[1Z,7XY],[1Z,7XY]}} \boxed{[1Z, 7XY]}$$

Similarly, the augmented V-torsor for $\oplus(7XY, [1Z, 3X, 6Y])$ is

$$\mathbf{T}_{7XY,[1Z,3X,6Y]}|_{\mathfrak{R}_{7XY}} = \left[\begin{array}{cccc} -0.1 & 0.1 & -0.3 & 0.3 \\ -0.1 & 0.1 & -0.3 & 0.3 \\ U & U & U & U \end{array} \right] \Bigg|_{\mathfrak{R}_{7XY}} \quad (5.38)$$

The constraint for tolerances of this modification is

$$(\mathbf{T}_{8XY,[1Z,3X,6Y]} + \mathbf{T}_{[1Z,3X,6Y],[1Z,7XY]})|_{\mathfrak{R}_{8XY}} \leq \mathbf{T}_{8XY,[1Z,7XY]}|_{\mathfrak{R}_{8XY}} \quad (5.39)$$

and the feature chain graph of the left side of the above unequation is

$$\begin{aligned} \boxed{8XY} &\xrightarrow{\mathbf{T}_{8XY,8XY}} \boxed{8XY} \xrightarrow{\text{zero}} \boxed{[1Z, 3X, 6Y]} \xrightarrow{\mathbf{T}_{[1Z,3X,6Y],[1Z,3X,6Y]}} \boxed{[1Z, 3X, 6Y]} \\ &\xrightarrow{\mathbf{T}_{[1Z,3X,6Y],[1Z,3X,6Y]}} \boxed{[1Z, 3X, 6Y]} \xrightarrow{\text{zero}} \boxed{[1Z, 7XY]} \xrightarrow{\mathbf{T}_{[1Z,7XY],[1Z,7XY]}} \boxed{[1Z, 7XY]} \end{aligned}$$

In unequation (5.39), $\mathbf{T}_{8XY,[1Z,7XY]}|_{\mathfrak{R}_{8XY}}$ is as given in equation (5.37). In order to determine $\mathbf{T}_{8XY,[1Z,3X,6Y]}|_{\mathfrak{R}_{8XY}}$, the augmented torsor $\mathbf{T}_{[1Z,3X,6Y],[1Z,7XY]}|_{\mathfrak{R}_{8XY}}$ has to be calculated. In reference system $\mathfrak{R}_{1Z-7XY}$, surface $1Z$ is the primary datum and constrains three degrees of freedom which are rotations around x - and y -axis and the translation along z -axis. $7XY$ is the secondary datum constraining translations along x - and y -axis. The total variation of reference $\mathfrak{R}_{1Z-7XY}$ with respect to reference $\mathfrak{R}_1$ is the stackup of that of $1Z$ and $7XY$ with respect to reference $\mathfrak{R}_1$. In $\mathbf{T}_{1Z,[1Z,3X,6Y]}|_{\mathfrak{R}_1}$, as surface $1Z$ is coincident with the reference system $\mathfrak{R}_1$, components of this augmented torsor standing for rotations around the x - and y -axis and the translation along the z -axis (with respect to the reference $\mathfrak{R}_1$) are all 0. Furthermore, according to [Villeneuve *et al.*, 2001; Park and Lee, 2001], in cases where the primary

CHAPTER 5. CONSTRAINT GENERATION FOR REDIMENSIONING

datum and the secondary can constrain the same degree of freedom, the primary datum is decisive. Thus, by applying formula (5.12), the accumulated torsor $\mathbf{T}_{[1Z,3X,6Y],[1Z,7XY]}|_{\mathbb{R}_{8XY}}$ can be calculated that

$$\mathbf{T}_{[1Z,3X,6Y],[1Z,7XY]}|_{\mathbb{R}_{8XY}} = \left[\begin{array}{cccc} 0 & 0 & -0.3 & 0.3 \\ 0 & 0 & -0.3 & 0.3 \\ U & U & 0 & 0 \end{array} \right] \Big|_{\mathbb{R}_{8XY}} \quad (5.40)$$

From equations (5.37) and (5.40), equation (5.39) resolves into

$$\left(\left[\begin{array}{cccc} r_x & R_x & t_x & T_x \\ r_y & R_y & t_y & T_y \\ U & U & U & U \end{array} \right] + \left[\begin{array}{cccc} 0 & 0 & -0.3 & 0.3 \\ 0 & 0 & -0.3 & 0.3 \\ U & U & 0 & 0 \end{array} \right] \right) \Big|_{\mathbb{R}_{8XY}} \\ \leq \left[\begin{array}{cccc} -0.166 & 0.166 & -0.5 & 0.5 \\ -0.166 & 0.166 & -0.5 & 0.5 \\ U & U & U & U \end{array} \right] \Big|_{\mathbb{R}_{8XY}} \quad (5.41)$$

From equation (5.41), it can be inferred that the maximal $\mathbf{T}_{8XY,[1Z,3X,6Y]}|_{\mathbb{R}_{8XY}}$ is

$$\mathbf{T}_{8XY,[1Z,3X,6Y]}|_{\mathbb{R}_{8XY}} = \left[\begin{array}{cccc} -0.166 & 0.166 & -0.2 & 0.2 \\ -0.166 & 0.166 & -0.2 & 0.2 \\ U & U & U & U \end{array} \right] \Big|_{\mathbb{R}_{8XY}} \quad (5.42)$$

In equation (5.42), from the permitted rotations point of view, the maximal tolerance is $(0.166 \text{ rad} \times 3 \text{ mm}) \times 2 = 1 \text{ mm}$, however, from the allowable translations point of view, it is $0.2 \text{ mm} \times 2 = 0.4 \text{ mm}$. Thus, it can be concluded that the maximal tolerance value of the new derived tolerance is 0.4 mm, which is the same result as [Xi *et al.*, 2001].

5.3.2 The bracket with one hole

The generation of alternative dimension chain and corresponding constraints is demonstrated by the means of the part in figure 2.3. The generated constraints are then validated by the means of tolerance stacking theory presented in [Lin and Zhang, 2001].

5.3 Examples

As stated in section 4.6, direct machinability of these three features in a single set-up can be achieved by applying two rewriting rules, however, only one is examined here. After instantiation, the rule (4.12) reads:

$$\perp(\mathbf{O}, 8X, 4Z) \perp(\mathbf{I}, 8X, 5Z) //(\mathbf{I}, 5Z, 4Z) \longrightarrow \perp(\mathbf{R}, 8X, 4Z) \perp(\mathbf{C}, 8X, 5Z) //(\mathbf{C}, 5Z, 4Z)$$

This rule replaces the original dimension $8X-4Z$ and adopts the alternative chain $8X-5Z-4Z$.

As argued in previous sections, this alteration must be complemented by unequations limiting the tolerances of the dimensions in states **O** and **C** on the right side of the rules.

The local reference systems of features $4Z$, $5Z$ and $8X$ ($\mathfrak{R}_{4Z}$, $\mathfrak{R}_{5Z}$ and $\mathfrak{R}_{8X}$, respectively) are as indicated in the 3-D model of figure 2.3. It can be inferred from this figure that the rotations and translations in equations (5.43) to (5.44) permit the adoption of a common reference system.

$$\hat{\omega}_{4Z \rightarrow 8X} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad \overrightarrow{O_{4Z}O_{8X}} = [3.5 \ 45 \ 60]^T \quad (5.43)$$

$$\hat{\omega}_{5Z \rightarrow 8X} = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \quad \overrightarrow{O_{5Z}O_{8X}} = [30 \ 3.5 \ -100]^T \quad (5.44)$$

Based on the information given and the procedure to construct augmented D-torsor presented above, the augmented V-torsor for $\perp(8X, 4Z)$ derived is

$$\mathbf{T}_{8X,4Z}|_{\mathfrak{R}_{8X}} = (\mathbf{T}_{8X,\overline{8X}} + \mathbf{T}_{\overline{4Z},4Z})|_{\mathfrak{R}_{8X}} = \left[\begin{array}{cccc} U & U & U & U \\ m & n & U & U \\ U & U & U & U \end{array} \right] \Big|_{\mathfrak{R}_{8X}} \quad (5.45)$$

where $m = r_{y8X,\overline{8X}} - R_{y4Z,\overline{4Z}}$ and $n = R_{y8X,\overline{8X}} - r_{y4Z,\overline{4Z}}$. And its graphical feature chain is

$$\boxed{8X} \xrightarrow{\mathbf{T}_{8X,\overline{8X}}} \boxed{\overline{8X}} \xrightarrow{\text{zero}} \boxed{\overline{4Z}} \xrightarrow{\mathbf{T}_{\overline{4Z},4Z}} \boxed{4Z}.$$

From the geometry of the surface and the simplex algorithm, it can be calculated that

CHAPTER 5. CONSTRAINT GENERATION FOR REDIMENSIONING

$m = -0.025$ rad, and $n = 0.025$ rad. Therefore,

$$\mathbf{T}_{8X,4Z}|_{\mathbb{R}_{8X}} = \begin{bmatrix} U & U & U & U \\ -0.025 & 0.025 & U & U \\ U & U & U & U \end{bmatrix} \Big|_{\mathbb{R}_{8X}} \quad (5.46)$$

In addition, the augmented torsors for $\perp(8X, 5Z)$ and $\parallel(5Z, 4Z)$ are

$$\mathbf{T}_{8X',5Z'}|_{\mathbb{R}_{8X}} = (\mathbf{T}_{8X',\overline{8X}} + \mathbf{T}_{\overline{5Z},5Z'})|_{\mathbb{R}_{8X}} \quad \boxed{8X'} \xrightarrow{\mathbf{T}_{8X',\overline{8X}}} \boxed{\overline{8X}} \xrightarrow{\text{zero}} \boxed{\overline{5Z}} \xrightarrow{\mathbf{T}_{\overline{5Z},5Z'}} \boxed{5Z'} \quad (5.47)$$

$$\mathbf{T}_{5Z',4Z'}|_{\mathbb{R}_{5Z}} = (\mathbf{T}_{5Z',\overline{5Z}} + \mathbf{T}_{\overline{4Z},4Z'})|_{\mathbb{R}_{5Z}} \quad \boxed{5Z'} \xrightarrow{\mathbf{T}_{5Z',\overline{5Z}}} \boxed{\overline{5Z}} \xrightarrow{\text{zero}} \boxed{\overline{4Z}} \xrightarrow{\mathbf{T}_{\overline{4Z},4Z'}} \boxed{4Z'} \quad (5.48)$$

And the constraint is

$$(\mathbf{T}_{8X',5Z'} + \mathbf{T}_{5Z',4Z'})|_{\mathbb{R}_{8X}} \leq \mathbf{T}_{8X,4Z}|_{\mathbb{R}_{8X}} \quad (5.49)$$

It follows from equation (5.46) and equation (5.49) that for rotations around the y -axis of the two augmented torsors on the left side of equation (5.49), the sum of their ranges cannot exceed $[-0.025 \text{ rad}, 0.025 \text{ rad}]$. The optimal allocation of tolerances to these two augmented torsors in the alternative tolerance chain depends on the optimization criteria selected, like, for example, the least cost proposed in [Dong *et al.*, 1994]. A possibility is an equal distribution, *i.e.* both are set to $[-0.0125 \text{ rad}, 0.0125 \text{ rad}]$.

To validate the constraints generated on the basis of the augmented torsor model, the tolerance stackup model basing on matrices introduced in [Lin and Zhang, 2001] is applied to the same example. It is assumed that the tolerance allocated for $\perp(8X, 5Z)$ is t_1 , and t_2 for $\parallel(5Z, 4Z)$.

According to [Lin and Zhang, 2001], the transformation matrix for $\perp(8X, 5Z)$ is

$$\mathbf{M}_1 = \mathbf{M}_t \mathbf{M}_x \mathbf{M}_y \mathbf{M}_z \quad (5.50)$$

5.3 Examples

where

$$\mathbf{M}_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{M}_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{M}_y = \begin{bmatrix} \cos \frac{t_1}{20} & 0 & -\sin \frac{t_1}{20} & 0 \\ 0 & 1 & 0 & 0 \\ \sin \frac{t_1}{20} & 0 & \cos \frac{t_1}{20} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{M}_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Note that $\frac{t_1}{20}$ is the rotation angle around axis y. The denominator in this fraction is the length of the plane 8X, in the orientation of the axis z (refer to figure 2.3).

Similarly, the transformation matrix for $((5Z, 4Z))$ is

$$\mathbf{M}_2 = \mathbf{M}'_t \mathbf{M}'_x \mathbf{M}'_y \mathbf{M}'_z \quad (5.51)$$

whereby

$$\mathbf{M}'_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{M}'_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{M}'_y = \begin{bmatrix} \cos \frac{t_2}{65} & 0 & -\sin \frac{t_2}{65} & 0 \\ 0 & 1 & 0 & 0 \\ \sin \frac{t_2}{65} & 0 & \cos \frac{t_2}{65} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{M}'_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Note that $\frac{t_2}{65}$ is the rotation angle around axis y. The value 65 is the length of the plane 4Z, in the orientation of the axis x (refer to figure 2.3).

The resultant tolerance stackup for the dimension chain $\perp(8X, 5Z) - \parallel(5Z, 4Z)$ is:

$$\begin{aligned} \mathbf{M} &= \mathbf{M}_1 \mathbf{M}_2 \\ &= \begin{bmatrix} \cos \frac{t_2}{65} \cos \frac{t_1}{20} - \sin \frac{t_2}{65} \sin \frac{t_1}{20} & 0 & -\cos \frac{t_2}{65} \sin \frac{t_1}{20} - \sin \frac{t_2}{65} \cos \frac{t_1}{20} & 0 \\ 0 & 1 & 0 & 0 \\ \cos \frac{t_2}{65} \sin \frac{t_1}{20} + \sin \frac{t_2}{65} \cos \frac{t_1}{20} & 0 & \cos \frac{t_2}{65} \cos \frac{t_1}{20} - \sin \frac{t_2}{65} \sin \frac{t_1}{20} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (5.52)$$

By applying trigonometric function formulae, the equation 5.52 becomes

$$\mathbf{M} = \begin{bmatrix} \cos \left(\frac{t_2}{65} + \frac{t_1}{20} \right) & 0 & -\sin \left(\frac{t_2}{65} + \frac{t_1}{20} \right) & 0 \\ 0 & 1 & 0 & 0 \\ \sin \left(\frac{t_2}{65} + \frac{t_1}{20} \right) & 0 & \cos \left(\frac{t_2}{65} + \frac{t_1}{20} \right) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.53)$$

The transformation matrix 5.53 represents a rotation around axis y of angle $\left(\frac{t_2}{65} + \frac{t_1}{20} \right)$. As stated before, the allowed rotation around y axis by the original perpendicularity $\perp(8X, 4Z)$ is $[-0.025 \text{ rad}, 0.025 \text{ rad}]$. It means that as long as $\left(\frac{t_2}{65} + \frac{t_1}{20} \right)$ is within this interval, the original dimension is respected. This is a constraint identical to the one generated on the basis of the augmented torsor model.

5.4 Conclusion

A methodology to create constraints for new derived geometric dimensions is presented and validated against existing models. The approach is exemplified for parallelism and dimensional constraints and uses the torsor model to mathematically represent geometric tolerance zones. Tolerance chains are manipulated and unequations are created to maintain the design intent. Through comparison with the existing methods, it is clear that the equivalent constraints are created, and the functionality of the changed design is maintained.

The proposed method has, though, advantages as compared to the one presented in [Xi *et al.*, 2001]. Xi *et al* consider datum changes only for a subset of location tolerances. Compared to this, the method proposed here takes into account alternative location tolerances as well as alternative dimensions for other size and geometric tolerances.

5.4 Conclusion

In addition, the model reported in [Lin and Zhang, 2001] only accounts for tolerance stackups and fails to present a methodology to systematically search alternative dimension schemes to improve the manufacturability of a design.

The constraint generation approach proposed in this chapter is a part of a design rewriting system, targeting a better manufacturability of a design through redesign. Notably, the method is independent from a particular machinability measure, as well as of the tolerance stackup model, and can therefore be considered generic.

CHAPTER 6

INTEGRATING AND IMPLEMENTING THE DESIGN REWRITING SYSTEM

The two major components of a design rewriting system were separately addressed in the previous two chapters. In this chapter, firstly, the manufacturability evaluation module is briefly discussed, while the adopted manufacturability measure, normalized tolerance, is highlighted. Then, the integration of these three modules with the interface module is addressed. In addition, in order to contribute to the automation of the proposed algorithm, a prototype is developed, which is called a DRS, the acronym of Design Rewriting System. Note that the interaction diagram of the system is depicted in figure 4.5. At last, an application and evaluation of the prototype DRS is presented.

6.1 Manufacturability Evaluation

To automatically evaluate the manufacturability of the newly derived design dimension graphs, a manufacturability criterion must be defined. In this system, the concept of the *normalized tolerance* is adopted, and a new manufacturability measure is defined (compare section 2.4.2 and section 3.7). Manufacturing cost (section 3.7), tolerance stacks (section 3.6) as well as operation element based efficiency (section 2.4.2) are not used at this moment. The main reason is that no proper computer-aided process planning system is available, which is required to accurately determine the above efficiency measures.

6.1 Manufacturability Evaluation

The tightness of geometric tolerances cannot be directly compared for different types of geometric dimensions. For example, even if a parallelism and a perpendicularity are toleranced as 0.5 mm, it is inaccurate to say that they are equally tight. Fortunately, this problem can be overcome by the method of tolerance normalization, as discussed in [Liu and Huang, 2000]. In this approach, the tightness of a geometric tolerance is measured by the maximum permissible rotational deviation of a datum from its ideal position in component locating [Liu and Huang, 2000].

Inspired by this idea, the author proposed a new manufacturability measure. The general idea is that the smaller the maximum permissible rotational deviation interval, the tighter the geometric tolerance, and therefore the harder (more costly) to achieve this specification. Furthermore, as stated in section 5.1.2 on page 84, a tolerance (including both GD&T and size tolerance) can be defined mathematically by an augmented torsor, from which this interval can be calculated from.

Based on the above two observations, the average and minimum of the maximal rotational deviation intervals of the targets of all tolerances in state **C** or **O** are two measures to assess the manufacturability of a design dimension graph. They are defined as:

$$\overline{NT}(\Xi^j) = \frac{1}{||\Xi^j||} \sum_{\xi \in \Xi^j} \text{maxrot}(\xi) \quad (6.1)$$

$$\min NT(\Xi^j) = \min_{\xi \in \Xi^j} \text{maxrot}(\xi) \quad (6.2)$$

where $\overline{NT}(\Xi^j)$ represents the average of normalized tolerances in the j th design version, while ξ , Ξ^j and are previously defined in section 3.4 on page 52. $||\Xi^j||$ is the number of dimensions in design dimension set Ξ^j . Function $\text{maxrot}(\xi)$ denotes the maximum permissible rotational deviation interval of a dimension ξ , and is defined in the succeeding paragraphs according to the type of dimensions.

Suppose the augmented V-torsor of a size dimension ξ is

$$\mathbf{T}|_{\mathbb{R}_2} = \left[\begin{array}{cccc} r_x & R_x & U & U \\ r_y & R_y & U & U \\ U & U & t_z & T_z \end{array} \right] \Bigg|_{\mathbb{R}_2}, \quad (6.3)$$

then

$$\text{maxrot}(\xi) = \frac{|R_x - r_x| + |R_y - r_y|}{2} \quad (6.4)$$

Only rotational deviations are considered, and the translational deviations are ignored according to [Liu and Huang, 2000]. In similar ways, functions for other types of geometric tolerances can be defined.

6.2 The Integration of Design Rewriting System

This section addresses the integration of the design rewriting system, where the rewriting module, constraint generation module, and manufacturability evaluation module are incorporated with the interface module. As mentioned in section 4.3.3, two modes, *i.e.* an interactive and an automatic mode, exist in the DRS. The reason for this is that the current manufacturability evaluation module may fail to give accurate manufacturability assessments for derived dimensional schemes due to its limited capacity (compare section 2.2.1 and 2.5), and at this time, human judgement can be used to determine which alternative is the best.

Interactive mode: (see figure 6.1)

1. The user selects a design dimension ξ_i^0 , which is impossible or impractical to machine directly.
2. Use the search-alternative algorithm as described in section 4.5 to search for all alternative chains of geometric design specifications for the designated dimension.
3. By user judgement, choose a chain with a better machinability (with yet unknown tolerance values).
4. Use the constraint-generation algorithm in section 5.2, create an unequation constraint, and add it into a set of unequations.
5. Repeat 1 to 4 until all design dimensions are directly machinable.

6.2 The Integration of Design Rewriting System

6. Add all constraints arising from the machining environment and practical aspects of machining to the established set of unequations. Solutions to this set are the desired tolerance values. It is very likely, that the system is under-constrained. Therefore, some balancing or optimization will be necessary. Should no solution exist, retry with different tolerance chains.
7. Replace the old by the new design specifications.

Figure 6.1: Activity diagram of the interactive mode

 CHAPTER 6. INTEGRATING AND IMPLEMENTING THE DRS

The interactive mode is illustrated in figure 6.1. Given the initial design and a set C of constraints on the machining environment (essentially best achievable tolerances for certain processes), the design is stepwise transformed towards a high manufacturability. Two procedures take place in parallel: while design dimensions causing a low machinability are replaced one by one, for each change constraints are added to C . These constraints are such that, if they are fulfilled, the resulting design fulfills the original design intent.

Automatic mode: (see figure 6.2)

1. The user specifies all design dimensions $\{\xi_i^0, \xi_j^0, \dots, \xi_n^0\}$, which are impossible or impractical to machine directly.
2. (a) Permute the designated dimensions, obtain a set of dimension lists.
 - (b) For each dimension list:
 - i. For each dimension to be removed, apply the search-alternative algorithm in section 4.5 to the input overall dimension graph(s) $\{\Omega^p, \Omega^{p+1}, \dots, \Omega^s\}$ ($s \geq p$) where the first input overall dimension graph is Ω^0 , output a set of overall dimension graphs $\{\Omega^{s+1}, \Omega^{s+2}, \dots, \Omega^{s+x}\}$ as the input overall dimension graph(s) of the succeeding dimension in the permutation; If the input overall dimension graph set is empty, apply the next permutation.
 - ii. When all dimensions in the considered dimension list are removed from the overall dimension graph, get the alternative design dimension graph set for this dimension list (with not yet known tolerance values).
 - (c) Repeat until all dimension lists are considered, and eventually, acquire all alternative design dimension graphs for the specified problematic dimensions (with not yet known tolerance values).
 - (d) Whenever a dimension is substituted by a dimension chain, use the constraint-generation algorithm stated in section 5.2, create an unequation constraint for this modification.

6.2 The Integration of Design Rewriting System

Figure 6.2: Activity diagram of the automatic mode

3. For each modified design dimension graph, apply step 6 of the interactive mode.
4. For each acquired design dimension graph, determine its manufacturability according to equation (6.1) and (6.2).
5. From the design dimension graphs which fulfill all above constraints, recommend those with the highest manufacturability to the user.

The automatic mode is shown in diagram 6.2, with activity “searching all alternative dimensions” illustrated in figure 6.3. The automatic-mode algorithm is based on the breadth first search. The advantages of this approach are that, since all possible alternatives are enumerated, the best dimension scheme is ensured to be found, as well as that it is easy to implement. However, it is not memory efficient. Probably a search based on an evolutionary algorithm is a good compromise, which is part of future work.

Currently, dimensions which cause the design to have a low manufacturability are specified by the user. However, provided that a manufacturability evaluation approach is available, these dimensions can be identified by the manufacturability evaluation module.

In addition, for simplicity and calculating the manufacturability in terms of normalized tolerances, when a dimension chain is used to substitute a problematic dimension, the permitted variation intervals of the augmented V-torsor for the original tolerance is equally distributed to augmented V-torsors in the chain (compare the first paragraph of chapter 5).

Also, derived dimension schemes are not checked for over- or under-constraining the design, which can be achieved by, as stated in section 4.4.6, the approach presented in Kandikjan *et al.* [2001].

6.3 Implementation of the Prototype DRS

The DRS is implemented in Haskell, a functional programming language, which is introduced in section 2.6.2. This prototype is written using Hugs 98¹, which is a functional programming system based on Haskell 98. Hugs 98 is chosen because it is a small, easy to

¹Refer to <http://www.haskell.org/hugs>

6.3 Implementation of the Prototype DRS

Figure 6.3: Activity diagram of part of the automatic mode

use Haskell interpreter. The prototype was developed under IRIX 6.5 on a SGI onyx2 workstation. In this prototype, the four modules and the program files are organized as follows:

- Rewriting module: Size.hs and Parallelism.hs. In addition, the initial overall dimension graph is stored in the OverallDimGraph.hs, which is initialized by functions implemented in InitGraph.hs;
- Constraint generation module: SizeVTorsors.hs, ParaVTorsors.hs, VTorsorFunctions.hs as well as the *Lp_solve*², an implementation for the *simplex* algorithm;
- Manufacturability evaluation module: Measures.hs;
- Interface module: MyMain.hs;

In the remainder of this section, the implementation of the rewriting module, the constraint generation module, the manufacturability evaluation module, as well as the interface module are introduced separately.

6.3.1 Implementation of the rewriting module

In this prototype, the rewriting rules for size and parallelism tolerances are implemented. In the file Size.hs, the rules 4.1 to 4.3 are realized, while in Parallelism.hs, the rules 4.4, 4.5 as well as those listed in table 4.1 are implemented.

The implementation of the search algorithm presented in section 4.5 is carried out in Size.hs as well as Parallelism.hs. The algorithm first searches all paths between the two nodes specified by the user. Then, it verifies the paths according to the encoded rewriting rules. At last, the verified paths are reported.

A breadth-first search algorithm from FGL/Haskell³ (acronym of Functional Graph Library for Haskell) is used for searching paths. As stated in [Erwig, 2001], its time complexity is $O(s \log s + t)$, whereby s is the number of nodes in an overall dimension graph, and t the average number of edges in overall dimension graphs. To estimate the time complexity of

²See http://www.math.iitb.ac.in/~rajesh/cs_project/csproject.html

³See <http://web.engr.oregonstate.edu/~erwig/fgl/haskell/>

6.3 Implementation of the Prototype DRS

the alternative dimension scheme search algorithm in the automatic mode as stated in section 6.2 on page 110, it is supposed that:

1. The number the problematic dimensions to be replaced is n ;
2. The average number of applicable rewriting rules for a problematic dimension is m ;
3. The average number of alternative schemes for a problematic dimension is l (refer to the step 2(b)i of automatic mode on page 110, and l is the average of x).

Then, the time complexity of the alternative dimension scheme search algorithm in the automatic mode is $O(n! l^{n-1} m(s \log s + t))$, whereby

- the factor $n!$ originates in the permutation of the n problematic dimensions as stated in step 2a in automatic mode on page 110;
- l^{n-1} is due to the number increase of alternative overall dimension graphs along with number increase of problematic dimensions which has been replaced. Given assumption 3 above, the numbers of alternative overall dimension graph follow a geometric progression series, whereby the common ratio is l , the first item is 1, and the last item is l^{n-1} ;
- $s \log s + t$ is the time complexity of the adopted breadth-first search algorithm.

Data structure of overall dimension graphs

The overall dimension graph and design dimension graph are implemented as inductive graphs which were developed by Erwig [2001] and implemented in his FGL/Haskell. Alike for all inductive data types, this makes graph algorithms amenable to inductive function definitions. The inductive view of graphs is as follows: a graph is either the empty graph or a graph extended by a new node v together with its label and with edges to those of v 's successors and predecessors that are already in the graph. The representation of each edge contains the successor/predecessor node and the label of the edge.

 CHAPTER 6. INTEGRATING AND IMPLEMENTING THE DRS

The following is the inductive data structure for overall dimension graphs in the prototype.

```

type Node                                = Int

type LabNode                             = (Axis, EntityType, TransformationMatrix,
                                           [ExtremePointsHomoCoordinates])

type LabEdge                             = (TolType, EdgeState, DatumPri,
                                           TolValue, AugmentedVTorsor)

type Adj LabEdge                         = [(LabEdge, Node)]

type Context LabNode LabEdge = (Adj LabEdge, Node, LabNode, Adj LabEdge)

data Graph LabNode LabEdge = Empty | Context LabNode LabEdge
                                & Graph LabNode LabEdge
  
```

where

- A *Node* represents a geometric entity of a design as introduced in section 4.1.
- *LabNode* is the label for a node.
 - *Axis* is the axis associated with this entity represented by the node, which can be *X*, *Y*, *Z*, or combinations of these as the second part of the labeling scheme introduced in section 2.3.4;
 - *EntityType* is the type of the entity, which can be a point, a line, a plane, or a hole;
 - *TransformationMatrix* is the 4×4 transformation matrix of the entity with respect to a common reference system;
 - [*ExtremePointsHomoCoordinates*] is the list of coordinates of extreme points of this entity, which will be used during the conversion between tolerances and augmented V-torsors as described in section 5.1.3.
- *LabEdge* is the label for an edge, which represents a dimension as stated in section 4.1.

6.3 Implementation of the Prototype DRS

- *TolType* indicates tolerance types, which is either a size, parallelism, perpendicularity, angularity, position tolerance or a basic dimension;
- *EdgeState* gives the state of a dimension, which is *O*, *I*, *C*, or *R* as defined in section 4.1.1;
- *DatumPri* defines whether the entity represented by the node in $(LabEdge, Node)$ is a primary, secondary, tertiary datum or not a datum at all;
- *TolValue* contains information like
 - * whether it has a tolerance, for instance, a basic dimension has no tolerance at all;
 - * if it has, the tolerance value is stored here;
 - * whether the shape of the tolerance zone is cylindrical;
 - * what is the material modifier (see section 2.3.1 and 5.2).
- *AugmentedVTorsor* represents the augmented V-torsor associated with the tolerance represented by this edge.
- for *Graph LabNode LabEdge*, it can be found that a graph is defined as an algebraic type with two constructors *Empty* and *&* as described on page 115.

6.3.2 Implementation of the constraints generation module

Three files, which are *SizeVTorsors.hs*, *ParaVTorsors.hs* and *VTorsorFunctions.hs*, as well as the *Lp_solve* library make up of this module.

The *Lp_solve* is an implementation of the *simplex* algorithm (see section 5.1.3). *Lp_solve* is a free linear programming library written in ANSI C by Michel Berkelaar⁴. It solves mixed integer problems using branch and bound method. Reasons to adopt *Lp_solve* are:

- It operates on both real and integer variables so that no special constraint is imposed on variables;

⁴See http://www.math.iitb.ac.in/~rajesh/cs_project/csproject.html

- It provides simple and efficient functions to add or remove constraints incrementally to a linear program.

In `VTorsorFunctions.hs`, common functions for augmented V-torsors are realized, like equations 2.2, 2.3, 5.12, 5.14 as well as 5.15. Furthermore, when the *AugmentedVTorsor* item in an edge label needs to be updated as this edge lies in the chain to be used to replace an edge representing an explicit dimension, the intersection between this existing augmented V-torsor and the newly allocated augmented V-torsor is calculated. This function is coded in this file as well.

In `SizeVTorsors.hs`, functions related to augmented V-torsors for size tolerances are realized. Unequations for size tolerance replacements are generated and written into a file in a format specified by the *Lp_solve*. *Lp_solve* then reads the information from this file. After results are available from *Lp_solve* and written into a second file, a function in `SizeVTorsors.hs` reads them from the second file, and updates the V-torsors it holds correspondingly. Similar processes are applied in `ParaVTorsors.hs` as in `SizeVTorsors.hs`, but to parallelism tolerances.

6.3.3 Implementation of the manufacturability evaluation module

Manufacturability evaluation measures as defined in equation 6.1 and 6.2 are implemented in `Measure.hs`. The manufacturability rating of an overall dimension graph is calculated upon needed.

6.3.4 Implementation of the interface module

An interface to the prototype DRS is provided in `MyMain.hs`. As addressed in section 4.3.3, two modes, an interactive and an automatic mode, exist. When running the DRS, the user is first asked to select a mode. Depending on the choice, different interactions are required to proceed as depicted in figure 4.5.

Given the interactive mode is chosen for replacing dimensions (see section 6.2), the user will be asked to enter a dimension. Then, the system reports alternative chains, if any, which

6.4 An Application of the Prototype DRS

may be used to replace the specified dimension. At this time, the user chooses one chain, and the system updates the overall dimension graph according to this choice. After that, the system prompts to ask the user if he/she would like to commit to the new design or to go backward to an earlier version of overall dimension graph. Then, if desired, a previous version is extracted and subsequently proceed to the next problematic dimension until all dimensions reducing manufacturability of a design are replaced.

If the automatic mode is selected (see section 6.2), the user is requested to enter all problematic dimensions at once. Then, the system performs steps from 2 to 5 as presented in page 110 (see also figure 6.2 and 6.3), and reports all alternative overall dimension schemes together with their manufacturability ratings, respectively. On the basis of the manufacturability ratings, the system recommends the top 5 alternative dimension schemes.

6.4 An Application of the Prototype DRS

In this section, the part shown in figure 2.3 is used as an example to demonstrate the usage of the prototype DRS. The common reference system, $\mathcal{R}_O$, from which the transformation matrix of each entity is calculated, is established at the left bottom of this part. The example part's design dimension graph and its overall dimension graph are shown in figure 4.1, 4.2 as well as 4.3, respectively.

In this prototype, the size overall dimension graph (see figure 4.2) and partially geometric overall dimension graph (see figure 4.3) are manually constructed, and saved in OverallDimGraph.hs. Note that in OverallDimGraph.hs, only one overall dimension graph exists, which contains both the size overall dimension graph and the partially geometric overall dimension graph.

In OverallDimGraph.hs, the sequence to add nodes to construct the overall dimension graph by using the two constructors *Empty* and *&* as introduced in section 6.3.1 is: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 (compare figure 2.3 to see corresponding entities). As it is too lengthy to present the whole overall dimension graph, only two nodes of the graph (they are node 1 and 2) are shown below:

CHAPTER 6. INTEGRATING AND IMPLEMENTING THE DRS

graphTwoNodes :: *Gr LabNode LabEdge* — the signature of the function

graphTwoNodes = (((((Size, O, NA, (Just((Just20, 0.25), 0, 0)), Nothing), 1),
 ((Para, I, NA, Nothing, Nothing), 1)],
 2, (“Z”, Plane, listArray((1, 1), (4, 4))[1, 0, 0, 31, 0, 1, 0, 75, 0, 0, 1, 90, 0, 0, 0, 1],
 [listArray((1, 1), (1, 4))[26, 60, 90, 1], listArray((1, 1),
 (1, 4))[36, 60, 90, 1], listArray((1, 1), (1, 4))[36, 90, 90, 1],
 listArray((1, 1), (1, 4))[26, 90, 90, 1]]),
 [((Size, O, NA, (Just((Just20, 0.25), 0, 0)), Nothing), 1),
 ((Para, I, NA, Nothing, Nothing), 1)]))
 &(([], 1, (“Z”, Plane, listArray((1, 1), (4, 4))[1, 0, 0, 37.5, 0, 1, 0, 75, 0, 0,
 1, 110, 0, 0, 0, 1], [listArray((1, 1), (1, 4))[0, 60, 110, 1],
 listArray((1, 1), (1, 4))[0, 90, 110, 1], listArray((1, 1), (1, 4))[75, 60, 110, 1],
 listArray((1, 1), (1, 4))[75, 90, 110, 1]]), [])
 &empty))

The prototype DRS is started in a unix shell by the command “hugs -98 +o” and the command “:l MyMain” is then issued in Hugs. At this moment, the prototype prompts the user for the mode (interactive or automatic) to choose. In the following two sections, these two modes are demonstrated separately. Before the MyMain module is loaded, the *Lp_solve* must be started in another shell (the DRS and *Lp_solve* communicate via a pipe).

In this example, it is supposed that $\mathcal{S}:\{(1,5),(8,9),(11,13)\}$ and $//(5,4)$ are problematic dimensions which reduce the manufacturability of the design⁵ (see figure 4.1).

⁵As only size and parallelism are implemented in this prototype DRS, it cannot perform replacements for tolerances other than these two types. However, once rewriting rules in chapter 4 are fully implemented, other types of tolerances can be replaced and procedures for those replacements are quite similar with these illustrated in this thesis.

6.4 An Application of the Prototype DRS

6.4.1 The interactive mode

In the interactive mode, the system will ask the user to enter a dimension to be replaced.

Suppose $\mathcal{S}(1,5)$ is entered first, the alternative chains proposed by the system are⁶:

1. $\boxed{1} \xrightarrow[\mathcal{S}]{\mathcal{O}} \boxed{2} \xrightarrow[\mathcal{S}]{\mathcal{I}} \boxed{5}$
2. $\boxed{1} \xrightarrow[\mathcal{S}]{\mathcal{I}} \boxed{3} \xrightarrow[\mathcal{S}]{\mathcal{O}} \boxed{5}$
3. $\boxed{1} \xrightarrow[\mathcal{S}]{\mathcal{I}} \boxed{4} \xrightarrow[\mathcal{S}]{\mathcal{O}} \boxed{5}$
4. $\boxed{1} \xrightarrow[\mathcal{S}]{\mathcal{O}} \boxed{2} \xrightarrow[\mathcal{S}]{\mathcal{I}} \boxed{3} \xrightarrow[\mathcal{S}]{\mathcal{O}} \boxed{5}$
5. $\boxed{1} \xrightarrow[\mathcal{S}]{\mathcal{O}} \boxed{2} \xrightarrow[\mathcal{S}]{\mathcal{I}} \boxed{4} \xrightarrow[\mathcal{S}]{\mathcal{O}} \boxed{5}$

At this time, the user is asked to select an alternative chain. Suppose the first of the chain is chosen. Then, the states of related dimensions are changed, and the constraint unequation is generated. The unequations subsequently are resolved by the interaction of the DRS and *Lp_solve*. The revised overall dimension graph is updated on the basis of the newly calculated augmented V-torsors. After the modification of the overall dimension graph is complete, and if the user accepts the changes, the system proceeds and prompts for the next problematic dimension.

Suppose $\mathcal{S}(8,9)$ is entered next, the output of the system is⁷:

1. $\boxed{8} \xrightarrow[\mathcal{S}]{\mathcal{I}} \boxed{10} \xrightarrow[\mathcal{S}]{\mathcal{O}} \boxed{9}$
2. $\boxed{8} \xrightarrow[\mathcal{S}]{\mathcal{I}} \boxed{6} \xrightarrow[\mathcal{S}]{\mathcal{O}} \boxed{10} \xrightarrow[\mathcal{S}]{\mathcal{O}} \boxed{9}$
3. $\boxed{8} \xrightarrow[\mathcal{S}]{\mathcal{I}} \boxed{7} \xrightarrow[\mathcal{S}]{\mathcal{O}} \boxed{10} \xrightarrow[\mathcal{S}]{\mathcal{O}} \boxed{9}$

Suppose the first alternative chain is chosen again and the resulting overall dimension graph accepted.

⁶The prototype only reports alternative chains but without boxes, arrows, types and states of dimensions in the chain.

⁷Refer to footnote 6.

Then, if subsequently $\mathcal{S}(11,13)$ is replaced by $\boxed{11} \xrightarrow[\mathcal{S}]{\mathcal{I}} \boxed{12} \xrightarrow[\mathcal{S}]{\mathcal{O}} \boxed{13}$ (the only choice) and $/(5,4)$ by $\boxed{5} \xrightarrow[//]{\mathcal{O}} \boxed{1} \xrightarrow[//]{\mathcal{I}} \boxed{4}$ (out of 31 alternatives), the design dimension graph from the resultant overall dimension graph is as shown in figure 6.4⁸. Note that, as addressed in the first paragraph of chapter 5, tolerance values for dimensions of type C are not allocated, although constraints for such allocation are created and resolved.

Figure 6.4: Modified design dimension graph for the example part

As stated in section 4.3.2, the order to enter dimensions can be important. However, as surfaces 1Z and 5Z (controlled by $\mathcal{S}(1,5)$), surfaces 8X and 9X (controlled by $\mathcal{S}(8,9)$), and surfaces 11Z and 13Z (controlled by $\mathcal{S}(11,13)$) are mutually perpendicular, the order is irrelevant in this example. In addition, as size and geometric dimensions are implicitly considered independent by the small displacement torsor approach, the order in which size and geometric dimensions are replaced is unimportant.

6.4.2 The automatic mode

In the automatic mode, the user is asked to enter all problematic dimensions at once. For this example, the input⁹ is

$$[[(1, 5), (8, 9), (11, 13)], [(5, 4), (8, 9)]]$$

⁸In the prototype DRS, no graphical output, only in the format like the one in the function *graphTwoNodes* on page 120.

⁹One more parallelism, $/(8,9)$, is added in the automatic mode.

6.5 Evaluation of the Prototype DRS

After about 80 minutes, 83700 alternative overall dimension graphs (including duplicate graphs) are reported. To eliminate duplicate graphs, at least another 25 hours are needed.

A more detail evaluation of the automatic mode is provided in the following section.

6.5 Evaluation of the Prototype DRS

The performance of the prototype in the automatic mode is shown in table 6.1. The time in the rightmost column is averaged over 10 runs. The example part, its design dimension graph and its overall dimension graph are shown in figure 2.3, 4.1, 4.2 as well as 4.3, respectively.

<i>Dimensions to be re-placed</i>	<i>No. of Alternatives with (without) duplicate graphs</i>	<i>Time for re-placement</i>	<i>Time for graph comparison</i>
$\mathcal{S}:\{(1,5)\}$	5 (5)	0.1 s	0.5 s
$\mathcal{S}:\{(1,5),(8,9),(11,13)\}$	90 (15)	68 s	174 s
$//:\{(5,4)\}$	31 (23)	0.238 s	49 s
$\mathcal{S}:\{(1,5)\} //:\{(5,4)\}$	155 (115)	4 s	154 s
$//:\{(5,4),(9,8)\}$	930 (345)	42 s	620 s
$\mathcal{S}:\{(1,5),(8,9),(11,13)\}$ $//:\{(5,4),(9,8)\}$	83700 (? ^a)	80 mins	at least 25 hours

^aAfter 25 hours running, the system still cannot finish the graph comparison and report the number of overall dimension graphs without duplication.

Table 6.1: Evaluation of the prototype

In table 6.1, the time taken are divided into two parts. The first part is for replacement, whereby all possible alternative chains are found, the initial design is modified, and all new overall dimension graphs with duplication are generated. The second part is for comparison, whereby duplicate graphs are removed from the overall dimension graphs generated in the previous step. Suppose n is the number of alternatives with redundancy. Then, the time for replacement is almost $O(n)$, and the time for graph comparison is about $O(n^2)$.

CHAPTER 6. INTEGRATING AND IMPLEMENTING THE DRS

The main target of this prototype is to show the feasibility of the design rewriting approach. Obviously, the current runtime performance is poor. However, it can be easily improved by

- Better Haskell compilers, like GHC¹⁰, which can be around 60 times¹¹ faster than Hugs 98, and makes the worst time for replacement to be 80 seconds;
- A better data structure for graph representations to reduce the memory used during graph comparison¹², and a smarter graph comparison strategy to reduce the time to eliminate duplicate graphs;
- More time efficient program languages, like C++, and
- Other search strategies, like a depth-first search or evolutionary algorithm.

¹⁰See <http://www.haskell.org/ghc>. GHC could not be installed under IRIX 6.5.

¹¹See <http://www.cs.kun.nl/~clean/download/FPdag/RvanHaaftenFPdag2002.ppt> (accessed in June, 2004).

¹²The current graph data structure is adopted from [Erwig, 2001].

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

This thesis presents a review of some aspects of manufacturability evaluation approach and computer aided tolerancing technique. Current manufacturability evaluation approach focus on the effects of geometry and topology on the manufacturability of a part, while the influences of dimensioning on the manufacturability of a design are not sufficiently considered. Bridging this gap is the target of this research. Four important aspects of computerized tolerancing, including tolerance representation, tolerance specification, tolerance transfer, as well as tolerance analysis and synthesis, are examined. In particular, a graph based process planning approach, the datum-hierarchy technique, is highlighted and identified as a good base for this thesis.

As there are strong indications that a major gain in terms of cost reduction can be achieved by changing the dimension scheme of a design, a new systematic redimensioning approach is proposed. This approach modifies an initial design to achieve a higher manufacturability through substituting problematic dimensions by alternative dimension chains while still respecting functional requirements. To this end, firstly, rules for modifying a design are presented. These rules allow to determine alternative dimension chains. Secondly, a mechanism to produce constraints for tolerance values of involved dimensions is introduced. Thirdly, a module for manufacturability evaluation is discussed; and eventually, an algorithm to integrate these three modules together with the interface module is developed.

CHAPTER 7. CONCLUSION AND FUTURE WORK

To automatically rate the manufacturability of a design, three manufacturability measures are proposed: the operation element based efficiency measure, manufacturing cost, and normalized tolerance. The first measure is discussed in [Thimm *et al.*, 2004a], along with an application of this measure to the manufacturability of process plans and designs. The second measure is briefly examined together with the benefits of considering alternative design specifications. The third, *i.e.* the normalized tolerance, is applied in the proposed system to assess the manufacturability of derived designs. The first and the third measures are suitable for all cases, while the manufacturing cost can be applied only when sufficient information on the cost-tolerance relationship is available.

In addition, a prototype of the design rewriting system is implemented based on the proposed methodology to demonstrate its feasibility.

Although an automatic manufacturability evaluation module is investigated, this thesis is less concerned with the choice of alternative design specifications (*i.e.* which alternative is better), but on providing a tool to transform a design dimension graph into another without compromising the design intent. A possible target of such a transformation is a design permitting direct machining. Also, it is not intended to re-specify tolerance values to newly derived dimensions, but to create constraints which should be respected during re-specifications.

The approach proposed in this thesis is validated and compared with existing methods in literature. The result of these comparisons indicates that this approach can maintain the functionality of the changed design and can be applied to more types of geometric tolerances than existing approaches.

7.2 Future Work

The work ahead includes:

1. To integrate the approach with existing commercial CAD softwares, so that the drawing (or solid model) information can be loaded directly from the CAD, and the modified design dimension scheme can be instantaneously reflected in the CAD drawing

7.2 Future Work

(or solid model).

2. A computer-aided process planing methodology should be developed and integrated into the system to make the manufacturability evaluation more accurate.
3. Other tolerance models should be considered to constrain modifications to certain tolerances of dimensional schemes.
4. The automatic-mode algorithm based on the breadth-first search should be replaced by a more memory efficient approach. Probably an evolutionary algorithm or some variety of depth-first search are good alternatives.
5. Geometric aspects of a design should be taken into account together with dimensional specifications to enhance a design further. For example, in some cases, a necessary tolerance stack (see section 3.6.1) with respect to a version of a design cannot be removed unless the geometry of the design is modified, like, for example, adding some surfaces whose only function is to serve as a location and/or datum to eliminate this necessary tolerance stack.

BIBLIOGRAPHY

Alting, L. and Zhang, H.-C. (1989). Computer-aided process planning: the-state-of-the-art survey. *International Journal of Production Research*, 27(4):553–585.

Anderson, D. M. (1990). *Design for Manufacturability: Optimizing Cost, Quality, and Time to Market*. CIM Press.

ASME Y14.5M (1994). *Dimensioning and tolerancing*. The American Society of Mechanical Engineers, NY. American National Standard.

Baader, F. and Nipkow, T. (1998). *Term Rewriting and All That*. Cambridge University Press.

Bjørke, Ø. (1989). *Computer-aided tolerancing*. 2nd edition. ASME Press.

Boothroyd, G. (1994). Product design for manufacture and assembly. *Computer-Aided Design*, 26(7):505–520.

Boothroyd, G. and Dewhurst, P. (1983). *Design for Assembly: A Designer's Handbook*. Boothroyd Dewhurst Inc., Wakerfield, RhodeIsland.

Bourdet, P. and Ballot, E. (1996). *Computer-aided Tolerancing*, chapter 8 Geometrical behavior laws for computer-aided tolerancing, pages 119–131. Chapman & Hall.

Britton, G. A., Thimm, G., Fok, S. C. and Jiang, F. (2001). Efficiency measures of process plans for machined parts. *The International Journal of Advanced Manufacturing Technology*, 17:819–824.

BIBLIOGRAPHY

- Britton, G. A., Tor, S. B., Thimm, G. and Jiang, F. (2002). A graph representation scheme for process planning of machined parts. *The International Journal of Advanced Manufacturing Technology*, 20(6):429–438.
- Britton, G. A. and Whybrew, K. (1997). CATCH: Computer-aided tolerance charting. In Zhang, H.-C., editor, *Advanced Tolerancing Techniques*, pages 461–488. John Wiley & Sons, Inc.
- Chase, K. W. (1999). *Tolerance Allocation methods for designers*. Technical report, Department of Mechanical Engineering, Brigham Young University.
- Chase, K. W., Gao, J.-S., Magleby, S. P. and Sorensen, C. D. (1994). *Including geometric feature variations in tolerance analysis of mechanical assemblies*. Technical report, Department of Mechanical Engineering, Brigham Young University.
- Chase, K. W. and Greenwood, W. H. (1988). Design issues in mechanical tolerance analysis. *Manufacturing Review*, 1(1):50–59.
- Chase, K. W. and Parkinson, A. R. (1991). *A survey of research in the application of tolerance analysis to the design of mechanical assemblies*. Technical Report 91-1, Department of Mechanical Engineering, Brigham Young University.
- Chen, B. S. (1994). *Characterization and Representation of Functional Requirements and Functional Tolerancing for Concurrent Design and Manufacturing*. Ph.D. thesis, The Ohio State University.
- Chen, C.-S., Chow, J. K., Lee, S.-M. and Swif, F. (1993). Concurrent product design evaluation and decision support. In Gu, P. and Kusiak, A., editors, *Concurrent Engineering: Methodology and Application*, pages 235–247. Elsevier Science Publishers.
- Chen, Y.-M., Miller, R. A. and Sevenler, K. (1995). Knowledge-based manufacturability assessment: an object-oriented approach. *Journal of Intelligent Manufacturing*, 6:321–337.

BIBLIOGRAPHY

- Chu, X.-N., Tso, S.-K. and Tu, Y.-L. (2000). A novel methodology for computer-aided process planning. *International Journal of Advanced Manufacturing Technology*, 16(10):714–719.
- Clément, A., Rivière, A., Serré, P. and Valade, C. (1997). The TTRS: 13 constraints for dimensioning and tolerancing. In EIMaraghy, H. A., editor, *Geometric Design Tolerancing: Theories, Standards and Applications*, pages 122–131. Chapman & Hall.
- Creveling, C. M. (1997). *Tolerance Design: A Handbook for Developing Optimal Specifications*. Addison Wesley Longman, Inc.
- Das, D., Gupta, S. K. and Nau, D. S. (1996). Generating redesign suggestions to reduce setup cost: A step towards automated redesign. *Computer-Aided Design*, 28(10):763–782.
- Davidson, J. and Shah, J. J. (2000). Geometric tolerances: a new applicaiton for line geometry and screws. In *Proceedings of A Symposium Commemorating the Legacy, Works, and Life of Sir Robert Stawell Ball Upon the 100th Anniversary of A Treatise on Theory of Screws*. University of Cambridge, Trinity College.
- Desrochers, A. and Maranzana, R. (1996). *Computer-aided Tolerancing*, chapter 2 Constrained dimensioning and tolerancing assistance for mechanisms, pages 17–30. Chapman & Hall.
- Desrochers, A. and Rivière, A. (1997). A matrix approach to the representation of tolerance zones and clearances. *International Journal of Advanced Manufacturing Techonology*, 13:630–636.
- Dissinger, T. E. and Magrab, E. B. (1994). A fully integrated design-for-manufacture system for powder metallurgy. In Gadh, R., editor, *Concurrent Product Design, ASME Winter Annual Meeting*, pages 47–58. Design Engineering Division, ASME, New York.
- Dong, Z., Hu, W. and Xue, D. (1994). New production cost-tolerance models for tolerance synthesis. *Journal of Engineering for Industry*, 116:199–206.

BIBLIOGRAPHY

- Erwig, M. (2001). Inductive graphs and functional graph algorithms. *Journal of Functional Programming*, 11(5):467–492.
- Fathi, Y., Mittal, R. O., Cline, J. E. and Martin, P. M. (1997). Alternative manufacturing sequences and tolerance buildup: a point of view and a case study. *International Journal of Production Research*, 35(1):123–136.
- Greenwood, W. H. (1987). *A new tolerance analysis method for engineering design and manufacturing*. Ph.D. thesis, Department of Mechanical Engineering, Brigham Young University.
- Gupta, S. K. and Nau, D. S. (1995). A systematic approach for analyzing the manufacturability of machined parts. *Computer-Aided Design*, 27(5):323 – 342.
- Gupta, S. K., Regli, W. C., Das, D. and Nau, D. S. (1997). *Automated Manufacturability Analysis: A Survey*. Technical report, Center for Integrated Manufacturing Decision Systems, The Robotics Institute, Carnegie Mellon University.
- Gupta, S. K., Regli, W. C. and Nau, D. S. (1994). Integrating DFM with CAD through design critiquing. *Concurrent Engineering: Research and Applications*, 2(2).
- Hal Daume, I. (2002). Yet another Haskell tutorial. URL <http://www.isi.edu/~hdaume/htut/>.
- Hayes, C. C. (1995). P^3 : A manufacturing planner that reasons about efficient plans to achieve positional tolerance specifications. In Thangaraj, A. R., Gadh, R. and Billatos, S., editors, *Concurrent Product and Process Engineering*, volume 85 (DE), pages 279–288. ASME, The American Society of Mechanical Engineering.
- He, J.-R. (1989). *Computer aided dimensioning and tolerancing for process operation planning*. Ph.D. thesis, School of Mechanical and Industrial Engineering, The University of New South Wales.

BIBLIOGRAPHY

- He, J.-R. and Gibson, P. R. (1992). Computer-aided geometric dimensioning and tolerancing for process-operation planning and quality control. *International Journal of Advanced Manufacturing Technology*, (7):11–20.
- Hong, Y. S. and Chang, T.-C. (2002). A comprehensive review of tolerancing research. *International Journal of Production Research*, 40(11):2425 – 2459.
- Hong, Y. S. and Chang, T.-C. (2003). Tolerancing algebra: a building block for handling tolerance interactions in design and manufacturing part 2: tolerance interaction. *International Journal of Production Research*, 41(1):47–63.
- Huang, S. H. (1998). Automated setup planning for lathe machining. *Journal of Manufacturing Systems*, 17(3):196–208.
- Huang, S. H. and Zhang, H.-C. (1996). Use of tolerance charts for NC machining. *Engineering Design and Automation*, 2(1):91–104.
- Huang, S. H., Zhang, H.-C. and Oldham, W. J. B. (1997). Tolerance analysis for setup planning: a graph theoretical approach. *International Journal of Production Research*, 35(4):1107–1124.
- Hughes, J. (1989). Why Functional Programming Matters. *Computer Journal*, 32(2):98–107.
- Huh, Y.-J. and Kim, S.-G. (1991). A knowledge-based CAD system for concurrent product design in injection molding. *International Journal of Computer Integrated Manufacturing*, 4(4):209–218.
- Ishii, K. (1993). Modeling of concurrent engineering design. In Kusiak, A., editor, *Concurrent Engineering: Automation, Tools and Techniques*. John Wiley & Sons, Inc.
- ISO 1101 (1983). *Technical drawings – Geometrical tolerancing – Tolerances of form, orientation, location and run-out – Generalities, definitions, symbols, indications on drawings*. International Standard Organization.

BIBLIOGRAPHY

- Jacquel, D. and Salmon, J. (2000). Design for manufacturability: a feature-based agent-driven approach. *Proceedings of The Institution of Mechanical Engineers Part B, Journal of Engineering Manufacture*, 214(10):865–879.
- Jayaraman, R. and Srinivasan, V. (1989). Geometrical tolerancing 1: virtual boundary conditions. *IBM J. Res. Dev.*, 33(2):90–104.
- Ji, S., Li, X. and Du, R. (2000). Tolerance synthesis using second-order fuzzy comprehensive evaluation and genetic algorithm. *International Journal of Production Research*, 38(15):3471–3483.
- Jietong, L. and Thimm, G. (2004). Constraint generation for alternative dimensional specifications. *Computer-Aided Design & Applications*, 1(1-4):675–682. ISSN 1686-4360.
- Jietong, L., Thimm, G. and Britton, G. A. (2003a). Alternative design specifications of prismatic parts for manufacturability. *Journal of Engineering Manufacture, Proceedings of the Institution of Mechanical Engineers Part B*, 217(4):569–572.
- Jietong, L., Thimm, G. and Britton, G. A. (2003b). Alternative geometric design specifications for manufacturability. In Qin, Y. and Juster, N., editors, *Advances in Manufacturing Technology - XVII: Proceedings of the First International Conference on Manufacturing Research (ICMR2003)*, pages 265–269. University of Strathclyde, Professional Engineering Publishing Limited, Bury St Edmunds and London, UK.
- Jo, H. H., Parsaei, H. R. and Sullivan, W. G. (1993). Principles of Concurrent Engineering. In Parsaei, H. R. and Sullivan, W. G., editors, *Concurrent Engineering: Contemporary Issues and Modern Design Tools*, chapter 1, pages 3–23. Chapman & Hall.
- Kandikjan, T., Shah, J. and Davidson, J. (2001). A mechanism for validating dimensioning and tolerancing schemes in CAD systems. *Computer-Aided Design*, 33:721–737.
- Karinthi, R. and Nau, D. (1992). An algebraic approach to feature interactions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(4):469–484.

BIBLIOGRAPHY

- Karinthi, R., Nau, D. S. and Yang, Q. (1992). Handling feature interactions in process planning. *Applied Artificial Intelligence*, 6(4):389–415. Special Issue on AI for Manufacturing.
- Kaul, A. (1992). Minkowski sums: a simulation for CAD/CAM. *Computers in Engineering*, 1:447–456.
- Laperriere, L. and Kabore, T. (2001). Monte carlo simulation of tolerance synthesis equations. *International Journal of Production Research*, 39(11):2395–2406.
- Lee, W. J. and Woo, T. C. (1989). Optimum selection of discrete tolerances. *Journal of Mechanisms, Transmissions, and Automation in Design*, 111:243–251.
- Legoff, O., Villeneuve, F. and Bourdet, P. (1999). Geometrical tolerancing in process planning: a tridimensional approach. *Proceedings of The Institution of Mechanical Engineers Part B, Journal of Engineering Manufacture*, 213:635–640.
- Li, R.-K. and Huang, C.-L. (1992). A frame work for automatic DFA system. *Computers in Industrial Engineering*, 22(4):403–413.
- Lin, E. E. and Zhang, H.-C. (2001). Theoretical tolerance stackup analysis based on tolerance zone analysis. *The International Journal of Advanced Manufacturing Technology*, 17(4):257–262.
- Liu, Q. and Huang, S. H. (2000). Geometric tolerance normalization and its application. *Transactions of NAMRI/SME*, XXVIII:305–310.
- Michael, W. and Siddall, J. N. (1981). The optimization problem with optimal tolerance assignment and full acceptance. *Transactions of the ASME*, 103:842–848.
- Moore, R. E. (1966). *Interval Analysis*. Prentice-Hall, Englewood Cliffs, New Jersey.
- Ong, S. K. and Chew, L. C. (2000). Evaluating the manufacturability of machined parts and their setup plans. *International Journal of Production Research*, 38(11):2397–2415.

BIBLIOGRAPHY

- Park, S. and Lee, K. (2001). Mathematical modelling of geometric tolerance propagation, considering effects of datums. *Proceedings of The Institution of Mechanical Engineers Part B, Journal of Engineering Manufacture*, 215:1191–1201.
- Pasupathy, T. M. K., Morse, E. P. and Wilhelm, R. G. (2003). A survey of mathematical methods for the construction of geometric tolerance zones. *Journal of Computing and Information Science in Engineering*, 3(1):64–75.
- Pennell, J. P., Winner, R. I., Bertrand, H. E. and Slusarczyk, M. M. G. (1989). Concurrent engineering: an overview for autotestcon. In *Autotestcon Proceedings'89: the System Readiness Technology Conference*, pages 88–99. Philadelphia, PA.
- Qiao, L., Wang, X.-Y. and Wang, S.-C. (2000). A GA-based approach to machining operation sequencing for prismatic parts. *International Journal of Production Research*, 38(14):3283–3303.
- Ramaswamy, S. (1998). A survey of DFM methods. URL <http://asudeSIGN.eas.asu.Edu/Education/MAE540/sanjayweb.htm>. Department of Mechanical and Aerospace Engineering, Arizona State University.
- Regli, W. C., Gupta, S. K. and Nau, D. S. (1995). Extracting alternative machining features: An algorithmic approach. *Research in Engineering Design*, 7:173–192.
- Requicha, A. A. G. (1983). Toward a theory of geometric tolerancing. *International Journal of Robotics Research*, 2(4):45–60.
- Rivest, L., Dupinet, E., Fortin, C. and Morel, C. (1994). Analysis of product tolerances for process plan validation. *Manufacturing Review*, 7(4):312–331.
- Roy, U. and Li, B. (1998). Representation and interpretation of geometric tolerances for polyhedral objects. I. form tolerances. *Computer-Aided Design*, 30(2):151–161.
- Roy, U. and Li, B. (1999). Representation and interpretation of geometric tolerances for polyhedral objects. II. size, orientation and position tolerances. *Computer-Aided Design*, 31:273–285.

BIBLIOGRAPHY

- Salomons, O. (1995). *Computer support in the design of mechanical products, constraint specification and satisfaction in feature based design for manufacturing*. Ph.D. thesis, University of Twente. URL <http://www.opm.wb.utwente.nl/staff/otto/thesis/>.
- Salomons, O., Poerink, H. J. J., van Slooten, F., van Houten, F. and Kals, H. (1996). *Computer-aided Tolerancing*, chapter 4 A Tolerancing Tool Based on Kinematic Analogies, pages 46–70. Chapman & Hall.
- Sermutsi-Anuwat, Y. (1992). *Computer Aided Process Planning and Fixture Design (CAPPFD)*. Ph.D. thesis, University of Canterbury, New Zealand.
- Shankar, R. and Jansson, D. G. (1993). A generalized methodology for evaluating manufacturability. In Sullivan, W. G. and Parsaei, H. R., editors, *Concurrent Engineering, Contemporary Issues and Modern Design Tools*, chapter 13, pages 248–263. Chapman & Hall.
- Speckhart, F. H. (1972). Calculation of tolerance based on a minimum cost approach. *Journal of Engineering for Industry*, pages 447–453.
- Srinivasan, R. S. (1994). *A theoretical framework for functional form tolerances in design for manufacturing*. Ph.D. thesis, The University of Texas at Austin.
- Srinivasan, V. and Jayaraman, R. (1989). Geometrical tolerancing 2: Conditional tolerances. *IBM J. Res. Dev.*, 33(2):105–124.
- Stage, R. W. (1996). *Manufacturability evaluation using resource based flexible form manufacturing features*. Ph.D. thesis, Arizona State University.
- Sturges, R. H. and Kilani, M. I. (1992). Towards an integrated design for an assembly evaluation and reasoning system. *Computer-Aided Design*, 24(2):67–79.
- Swift, K. G. (1981). *Design for Assembly Handbook*. Salford University Industrial Center, UK.

BIBLIOGRAPHY

- Teck, O. C. (1996). *A dimensioning and tolerancing method for product design and process planning*. Master's thesis, School of Mechanical and Production Engineering, Nanyang Technological University, Singapore.
- Teissandier, D., Couétard, Y. and Gérard, A. (1999). A computer aided tolerancing model: proportioned assembly clearance volume. *Computer-Aided Design*, 31:805–817.
- Thimm, G., Britton, G. and Jietong, L. (2004a). Operation element based efficiency for process plans and designs. *The International Journal of Advanced Manufacturing Technology*, 24(5-6):370–375.
- Thimm, G., Britton, G. A. and Fok, S. C. (2004b). A graph theoretic approach linking design and process planning, part 1: design to process planning. *The International Journal of Advanced Manufacturing Technology*, 24(3-4):261–271. ISSN 0268-3768 (Paper) 1433-3015 (Online).
- Thimm, G., Britton, G. A. and Whybrew, K. (2001). Optimal process plans for manufacturing and tolerance charting. *Proceedings of the Institution of Mechanical Engineers. Part B, Journal of Engineering Manufacture*, 215(B8):1099–1105. ISSN 0954-4054.
- Thimm, G. and Jietong, L. (2004a). A computer-aided re-dimensioning system for improved manufacturability of machined parts. *Journal of Manufacturing Systems*. Submitted Aug 04.
- Thimm, G. and Jietong, L. (2004b). Redimensioning parts for manufacturability: A design rewriting system. *The International Journal of Advanced Manufacturing Technology*. ISSN 0268-3768 (Paper) 1433-3015 (Online).
- Trego, A. (1993). *A comprehensive system for modeling variation in mechanical assemblies*. Master's thesis, Department of Mechanical Engineering, Brigham Young University. AD-CATS Report No. 93-2.
- Tseng, Y.-J. and Kung, H.-W. (1999). Evaluation of alternative tolerance allocations for mul-

BIBLIOGRAPHY

- multiple machining sequences with geometric tolerances. *International Journal of Production Research*, 37(17):3883–3900.
- Ullman, D. G. (1992). *The mechanical design process*. McGraw-Hill.
- Veitschegger, W. K. and Wu, C. (1986). Robot accuracy analysis based on kinematics. *IEEE Journal of Robotics and Automation*, RA-2(3):171–179.
- Venkatachalam, A. R., Mellichamp, J. M. and Miller, D. M. (1993). A knowledge-based approach to design for manufacturability. *Journal of Intelligent Manufacturing*, 4:355–366.
- Villeneuve, F., Legoff, O. and Landon, Y. (2001). Tolerancing for manufacturing: a three-dimensional model. *International Journal of Production Research*, 39(8):1625–1648.
- Wade, O. R. (1967). *Tolerance control in design and manufacturing*. Industrial Press Inc.
- Wade, O. R. (1983). Tolerance control. In Drozda, T. J. and Wick, C., editors, *Tool and manufacturing engineers handbook: a reference book for manufacturing engineers, managers and technicians*, volume 1: Machining, 4th edition, chapter 2. Society of Manufacturing Engineers.
- Wang, H.-P. and Li, J.-K. (1991). *Computer-aided process planning*. Elsevier, Amsterdam-Oxford-New York-Tokyo.
- Weill, R. D. (1997). Dimensioning and tolerancing for function. In Zhang, H.-C., editor, *Advanced Tolerancing Techniques*, chapter 13, pages 329–354. John Wiley & Sons, Inc.
- Whitney, D. E., Gilbert, O. L. and Jastrzebski, M. (1994). Representation of geometric variations using matrix transforms for statistical tolerance analysis in assemblies. *Research in Engineering Design*, 6:191–210.
- Whitney, D. E., Mantripragada, R., Adams, J. D. and Rhee, S. J. (1999). Designing assemblies. *Research in Engineering Design*, 11:229–253.

BIBLIOGRAPHY

- Whybrew, K., Britton, G. A., Robinson, D. F. and Sermsuti-Anuwat, Y. (1990). A graph theoretic approach to tolerance charting. *International Journal of Advanced Manufacturing Technology*, (5):175–183.
- Wozny, M. J., Turner, J. U., Graves, R., Dong, X., Sodhi, R. and Dixon, J. R. (1993). A unified representation to support evaluation of designs for manufacturability: Phase III. In *Proceedings of the 1993 National Science Foundation Design and Manufacturing Systems Conference*, pages 741–753.
- Xi, M., Lehtihet, E. A. and Cavalier, T. M. (2001). A mathematical optimization approach to a subset of location tolerance transfer problems. *Journal of Computing and Information Science in Engineering*, 1:180–185.
- Youssef, M. A. (1994). Design for manufacturability and time-to-market: Part 1, theoretical foundations. *International Journal of Operations & Production Management*, 14(12):6–21.
- Zhang, H.-C., Huang, S. H. and Mei, J. (1996). Operational dimensioning and tolerancing in process planning: setup planning. *International Journal of Production Research*, 34(7):1841–1858.
- Zhou, F., Kuo, T.-C., Huang, S. H. and Zhang, H.-C. (2002). Form feature and tolerance transfer from a 3D model to a setup planning system. *International Journal of Advanced Manufacturing Technology*, (19):88–96.
- Ziemke, M. C. and Spann, M. S. (1993). Concurrent engineering's roots in the World War II era. In Parsaei, H. R. and Sullivan, W. G., editors, *Concurrent Engineering: Contemporary Issues and Modern Design Tools*, chapter 2. Chapman & Hall.

INDEX

- 6σ , 31
- C_f , 31
- O , 54, 55
- P , 54
- S_ξ , 55
- T , 30
- T_i , 30
- Z , 30
- Z_i , 30
- ΔU , 30
- $\Re_i$, 26
- Ξ^j , 51, 105
- $\Xi_{\check{T}}$, 51
- $\oplus$, 55
- $\mathbb{F}$, 42
- $\mathbb{O}$, 42
- $\mathbb{S}$, 41
- ω , 26
- Ω , 65
- Ω^1 , 65
- $\overline{NT}$, 105
- $\tau(\xi)$, 50
- $\text{maxrot}(\xi)$, 105
- $\check{F}$, 50
- $\check{T}$, 50
- v , 26
- ξ , 50, 51, 105
- ξ_i^j , 51
- f , 30
- $t(c_{ij})$, 55
- x_i , 30
- $\boxed{a}$, 90
- $\mathbf{T}_{a,b}(l, m)$, 86
- ASME Y14.5M, 24
- assemblability, 21
- assembly datum, 52, 53
- augmented torsor, 7, 105
- basic dimension, 46
- bilateral tolerance, 47
- bilateral tolerance
 - equal bilateral tolerance, 47
- breadth first search, 110
- breadth-first search, 125
- CAPP, *see* computer aided process planning
- computer aided process planning, 18
- computer aided tolerancing, 23

INDEX

- confluence, 69
- critical factors, 48
- cross-functional team, 19
- datum, 47
- datum feature, 47
- datum-hierarchy tree, 32, 37
- design datum, 52
- design dimension tree, 38
- design for assembly, 21
- design for manufacturability, 7, 17
- design for manufacture and assembly, 21
- design rewriting system, 7, 44, 104
- deviation, 47
 - lower deviation, 47
 - upper deviation, 47
- DFA, 21, *see* design for assembly
- DFM, 17, 19, 21, *see* design for manufacturability
- DFM wheel, 17
- DFMA, 21, *see* design for manufacturability and assembly
- dimension, 46
- dimension limits, 47
 - maximum dimension limit, 47
 - minimum dimension limit, 47
- direct approach, 19
- DRS, *see* design rewriting system
- evolutionary algorithm, 110, 125
- explicit dimension, 7, 60
- FBM, *see* feature based model
- feature based model, 23
- FGL/Haskell, 113
- finishing machining operation, 38
- finishing sub-graph, 39
- functional design dimension, 50
- functional programming, 44
- geometric tolerance, 47
- Haskell, 44
- implicit dimension, 60
- indirect approach, 19
- initial overall dimension set, 65
- least cost optimization, 29
- locating datum, 52, 53
- Lp_solve, 112
- machining feature, 41
- manufacturability evaluation, 17
- manufacturing cost, 16
- ME, 17–19, *see* manufacturability evaluation
- measuring datum, 52, 53
- minimal manufacturing cost, 57
- necessary stackup, 54
- normalized tolerance, 16, 104

- operation element based efficiency, 8, 16, 41
- operational datum, 52
- operational element, 41
- overall dimension graph, 7, 60
- overall dimension set, 65
- pattern matching, 45
- pre-finishing operation, 38
- pre-form sub-graph, 39
- pre-forming operation, 39
- principle of coincidence of datums, 53
- prismatic parts, 15
- proportional scaling, 29
- radian (rad), 96
- reference and datum , 51
- root node, 38
- root sum square, 31
- rooted tree, 34
- state of a dimension, 62
- state transfer mechanism, 69
- stock removal allowance, 34
- stock removal set, 34
- surface label, 34
- Taguchi's approach to tolerancing, 29
- term rewriting system, 44
- termination, 69
- tolerance, 46
- tolerance analysis, 29
- tolerance chart, 34
- tolerance chain, 56
- tolerance representation, 23
- tolerance stack, 47
- tolerance stack set, 55
- tolerance stackup, 33, 47
- tolerance stackup model, 30
- tolerance synthesis, 29
- tolerance transfer, 32
- tolerance zone, 24
- traditional tolerance chart, 33
- trivial design dimension, 50
- type-1 operation, 39
- type-2 operation, 39
- unilateral tolerance, 47
- weight factors, 29
- worst case, 30, 31